TRANSCRIPTS

KERI Conference April 2026

A complete transcript of every recorded talk from the KERI Conference April 2026.

42
TALKS

19:07:04
RUNTIME

November 26, 2026

CONTENTS

Full transcripts of all 42 recorded sessions, in the order they appear in the archive.

01	**Alan Davies**	ACDCs for Enterprises	29:11
02	**Ari Argoud**	KRAM IT! with Ari	29:13
03	**Ari-Argoud-Interview**		01:01
04	**Cole Davis**	Contract Credentials & Verifiable Identity	35:42
05	**Cole-Davis-Interview**		01:51
06	**Daniel Hardman**	Applying KERI to General Evidence Problems	39:53
07	**Daniel Hardman**	Open Verifiable Communications	41:39
08	**Daniel-Hardman-Interview**		01:16
09	**Evan Asakawa**	Mobile Wallet	26:16
10	**Fergal O'Conner**	Signify Security Architecture	42:28
11	**Fergal O'Connor**	KERIA & Signify Path to Production	39:46
12	**Fergal-O'Connor**		01:57
13	**George McEwan**	How KERI Enables Better Private-Public Partnerships	50:58
14	**George McEwan**	The Landscape of SEDI	44:35
15	**Jared Jefferey**	FHIR and CMS	39:59
16	**Jared-Jefferey-Interview**		02:59
17	**Jay Alexander Elliot**	Open KERI Infrastructure	22:43
18	**Jonathan Rayback**	KERI Bridge to the DID World	40:00
19	**Jonathan-Rayback-Interview**		01:26
20	**Keanu Pahio**	KERI Developement & Deployment	23:37
21	**Keanu-Pahio-Interview**		00:59
22	**Kent Bull**	KERIA Tutorial	41:15

23	**Kent Bull**	KERIpy Async Architecture	39:06
24	**Kent-Bull-Interview**		01:07
25	**Ned Smith**	Secure Asset Transfer Protocol	39:57
26	**Ned-Smith-Interview**		01:37
27	**Nicholas Racz**	Investigating the Game of Trust	43:27
28	**Nicholas-Racz-Interview**		01:29
29	**Phil Feairheller**	ACDCs for Authorization	26:36
30	**Phil Feairheller**	Infrastructure for Security	36:59
31	**Phil Windley**	Digital Identity Needs a Legal Foundation	40:07
32	**Phil-Feairheller-Interview**		00:41
33	**Phil-Windley-Interview**		01:23
34	**Ryan Hansen**	Witnesses & Watchers	39:26
35	**Ryan Hansen Pannel**	What KERI allows us to do that we couldn't before	32:51
36	**Ryan-Hansen-Interview**		01:21
37	**Samuel Smith**	SPAC & TSP	40:49
38	**Samuel Smith**	State of the KERI Suite	58:17
39	**Samuel Smith**	The Digital Identity Tradespace	37:05
40	**Samuel Smith**	The Future of KERI	1:03:31
41	**Scott Stuewe Mark Scrimshire Jared Jeffery**	Provenanting Patient's Medical Records	48:46
42	**Thomas Mayfield**	KERI Infrastructure & the 5 W's	33:24

TALK I

ACDCs for Enterprises

Alan Davies

KERI Conference April 2026 · talks · 29:11

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Alan Davies argues enterprises need simplified messaging about KERI and ACDCs as trustability beyond passwords and detection. He traces trust from face-to-face deals to rampant remote fraud, proposes incremental ACDC adoption while rearchitecting toward user-controlled identifiers, and contrasts weak bank logging with cryptographically attributable actions.

... and thank you all for being here. I've been accused many times in the past of oversimplifying. I can say here and now that I never oversimplify. However I do like to simplify things and the world out there needs to understand what KERI and ACDCs can deliver to the world and we need to find ways of simplifying our messages to them to make sure that they get it.

I thank Thomas [Mayfield, red.] for his great presentation explaining in high-level terms what KERI is all about and then diving into the lower level detail and I'm going to a certain extent follow that track. Let's get going.

I want to start by talking about how trust used to work. It used to be all about personal reputation. This in the days when you wanted to carry out some kind of transaction, buy something, you would go along to the person who was selling it or providing it. You would talk to them, you would make a deal, you shake hands and you would decide whether you're trusting them or not. If you trusted them, you would give them their money. They would give you the goods and off you went. Trust depended on physical presence.

All of the verifications that took place were pretty well immediate and you would decide do I really trust this person? If so here's your money. I'll take the goods now or maybe later. Fraud did exist. Of course there would be people who would promise to do something and then not deliver it but the scale was limited to I guess the person that they were doing the deal with.

That was in the distant past. Now things have changed and in fact they changed at least a couple of hundred years ago when the ability to communicate in different ways came along. But it they changed in a very big way when global communications arrived firstly with the telephone and then Fax machines and of course now the internet.

What the internet enabled us to be able to do is execute transactions remotely which is a great thing in many ways. It enabled international commerce to be able to scale up in so many different ways and also allow regular humans to participate in that kind of commerce. We can go online to Amazon, we can buy what we want when we want and it turns up the next day. It's very convenient.

In order to support that, when the internet came along, it was a almost a natural knee-jerk reaction to say, well, if people are going to use online systems, they need to authenticate to be able to access it. And usernames and passwords was the way that we did that.

You log in, you get a session token and the system that you're talking to uses that token to decide what you're allowed to do. The end result so far has been rampant online fraud and we've done what we can to limit that. Mostly by trying to detect fraud which is in my view a losing battle. The fraudsters have so many tools available to them to be able to fool the general public into thinking that they are someone legitimate.

and detection is we're always going to be losing that battle to try and detect fraudulent activity. This is my own personal view of the distant future in which everything that we do will be signed. It's a signed intent world and I think the way we'll end up if you think maybe four or five years in the future We will all have an agent on our devices that does things for us.

You'll ask your agent AI agent to do something. It'll be maybe move some money from your checking account to your savings account. the AI agent on your device because it knows you've authenticated with that device biometrically. It'll wrap that request in an ACDC, sign it with your private key.

It'll just use your face ID or whatever other biometric it's using to make sure it really is you. It may add additional signatures if necessary. If you're a child using your device, it may be that whatever you're asking your agent to do requires your parents permission.

You may get an additional signature added to that request. That request may include relevant credentials, your SEDI credential for example, or who knows what other entitlement credentials that you may have. That request will then get sent to some kind of execution service. If it's your bank, because you're moving money from one account to another, then it'll be your bank's online system.

It'll verify the trustability of that request. It'll see where it came from. It will see your signature. It will see your maybe your parents signature if you're a child. It will do the usual KERI ACDC stuff. Verify it hasn't been tampered with.

They it they know it's you and the bank will then execute that request and make a record of it. And I don't think there will be a need to have a username and password. You won't need to login for this because basically you've sent a signed sealed request to your bank that they know it came from you. They'll execute it.

They'll record it and that's the end of it. I'm exaggerating here. I'm sure there will be passwords, sessions tokens, and 2FA for all sorts of stuff, but in the long term, they will be at least less necessary. And in the example I gave you, you can imagine a world where everything you do is signed intent.

I think that this is going to be how we reduce the rampant online fraud to .., there's always going to be fraud, unfortunately, there always going to be someone who will find a way of fooling someone else into doing something they shouldn't be doing. But this is the long-term vision and this is all going to be delivered, in my view, with implementations of KERI and ACDCs, and everything that we've been talking about today.

However, there's going to be a lot of disruption to be able to enable all of that. I mean the example I gave you about the bank they need to rearchitect their systems so that they are accepting signed requests from their customers that they identify through their their online identity that the customers own and that's going to be a great world of course everybody will be their own identity provider which is how it should be if you think about it. I'm the only person who knows that I am me. And if I control a digital identity that asserts that, then I'm by far the best person to be able to prove who I am.

And I don't like the idea of delegating that to some other federated identity provider. However, they're not going to like it at all. The federated identity providers are having lots of information from people every time they log in.

There's going to be a lot of resistance. Information harvesters are going to hate all of this because everything that Sam [Smith, red.] tells us that we should be in control of is not what they want us to be in control of, sadly.

As I mentioned, all sorts of online services will need to do a lot of rearchitecting to implement that final vision. And this isn't going to happen in the next 12 months. There will be movement towards it but In my view, we need to do something sooner to start getting KERI and ACDC adoption in a small way and then move on. This is where we are now. We have rampant online fraud.

Autonomous AI agents are doing more and more stuff and working out how to do things on their own. Mostly because they think that they are doing what we want them to do but not necessarily. The current way of recording what happens is really online logging. The the bank will keep a record of what you asked it to do moving money from one account to another. But if someone gets my password and asks my bank to do something on behalf of me then as far as the bank logging is concerned it was me doing it which just is not real. It is it's just weak.

What we really need is what I call trustability. And trustability is the ability to be able to make a good trust decision on any kind of action that you might be Intending to make based on some information that comes at you over the internet. And it can be something as simple as a news article.

Did this journalist really write this? Do I trust this journalist? The trust decision might be very weak. I'm just going to develop my own opinion about whatever is being written about based on that journalist's reputation. But other things are much more important.

If I see a request from one of my kids to send some money to a bank account because he's in trouble, then I need to be able to trust that is really from my son. And I need to make a trust decision. I might need to make it quickly. How do we get trustability into the world of online transactions?

And what we need of course is KERI and ACDCs and we need .., I hate to say "a little bit of KERI and ACDC" because we need the whole thing but we can't very quickly implement the entire or reimplement the entire online access infrastructure to support the the long-term vision of how the whole world of signed intent is going to work. And where digital trust foundry is starting, is saying that what we need is something that we call execution trust.

Immediately before making some kind of Decision to execute some kind of action based on information that's coming to us remotely. And that might be we're a supplier receiving a purchase order. We might be an employer being presented with someone's academic credentials before we hire them for a job. Before we make that decision, then we need a way to be able to make a trust assessment of what's being presented to us.

And this is where I think that what the world needs is to be able to start slowly and to be able to implement what we call execution trust into existing systems and of course start with high value high-risk transactions. For example wire transfer fraud. It is way too easy for bad actors to be able to make changes to wire transfer requests as they move from place to face before they get to the bank that's going to execute that wire transfer.

Taking a wire transfer request, wrapping it in an ACDC, getting it signed by the owner of the originating account means that when the bank gets it, they can verify it, make a trust decision and execute it and then keep a cryptographically verifiable record of what they were asked to execute for any time in the future. What we're talking about is inserting a level of trustability into existing applications at execution time, hence the term execution trust.

And this can apply to any digital artifact. Verifiable credentials was really all about academic and professional credentials but ACDCs are really capable of handling any kind of data content that might be delivered from one trust domain to another.

Even better in the short term, it makes sense even with a single trust domain to be able to take some kind of action. Let's say, it's updating a production system. We have an update that we want to make to a production system and we want to make sure that it's fully authorized before we execute it. And of course, that update may have been generated by AI. It's kind of important that a human gets involved in that process. A workflow application that's going to apply that update to the production system could take that description of that update request, wrap it in an ACDC, send it to the CIO to say, "We're about to update your production systems with this. Do you approve it?" He would sign it, return it back. Then that update application would say "Okay, I have approval to go ahead with with this update, apply the update, keep a record" and that all happens within a single trust domain. What you've done there is applied a little bit of KERI and ACDCs to a particular high value use case to make sure that in that particular example the update was being made was fully authorized by someone who is the appropriate authorizer of that update.

The same thing applies across trust domains where maybe a business generates a legal document that needs to be signed by the CEO before it gets delivered to a partner for them to execute. You can build that kind of functionality into an existing application that's already generating these documents. You just say, "We already have the document. It's a PDF.

We're just going to wrap it in ACDC, get it signed, and send it off. And then whoever gets it, because they know how to verify an ACDC, we'll run a verification. They know that it was signed by the CEO, they can act on it.

I think that's basically what I'm talking about here. And of course, multisig comes in very useful in situations like this. The application at execution time can verify and either decide to reject it if it's not happy with what it sees or executed and keep a permanent record of its trust decision and what it made that trust decision based on.

The benefits of this approach which is a little bit of KERI and ACDCs injected into existing business applications for high value transactions. There are so many business use cases that this can apply to across all industries.

I know from personal experience that if I use my bank's online system to do a wire transfer even though I'm logged in, it'll get to a certain point where it says "We're going to send you a code to your mobile phone and want you to type it in. Two-factor authentication to make sure that even though I'm already logged in that it really is me again, sort of really me.

This would replace those kinds of weak two-factor authentication with what we call a hard gate. There will be no need to rearchitect existing systems. If you take for example the bank's wire transfer request system, they already have that ability in their system. It's part is the functionality of their banking system anyway. The additional step would be just to make an API call to a platform that knows how to wrap that content into an ACDC, send it to an authorized signer, and return the signed ACDC back. Then that application just makes another API call to verify it, looks at the results, makes a trust decision, either rejects or continues. No need to completely rearchitect existing systems.

This will drastically reduce online fraud for high value business transactions and it can be done really very soon. I like to talk about this as getting used to KERI and ACDCs one bite at a time. You introduce something like this to a high value transaction that's already being executed by business applications Today and they see the immediate benefits and then once businesses start using this technology in this

limited constrained way then they will be able to see how that the basic technology can be applied in all sorts of other situations. And this is how we get the world starting to use KERI and ACDCs without having to make huge changes.

And also this is another good reason for businesses to get a vLEI of course. Digital Trust Foundry is a platform that enables businesses to do exactly what I've described. It's multi-instance multi-tenant platform that fully implements KERI and ACDCs with our witnesses and watchers network.

(let me see if I'm getting ahead of myself here) We've implemented what we call a trustables API. The business application can say "I have some digital content that I want to make trustable." It makes a call to the API specifying the AIDs of the signers or the roles that need to approve whatever the digital artifact is that's being processed.

The end result is, it gets back an ACDC that it can verify and that's where we're starting. We're saying that we need to provide what we call execution trust to existing applications so that they can start using the power of KERI and ACDCs now in a small way but then to see understand the value of it by practical experience.

We're looking for partners. Our business model is "Yes, we will deliver our a platform" for use by businesses generally but also by partners who have existing applications. It might be workflow applications that need stronger approval processes for example. And they can work with us to build execution trust functionality into their applications and deliver it to their customers. In this way we build a network of businesses who are implementing the power of ACDCs and KERI without having to build the whole thing themselves. They just make use of our platform. In some ways it's similar to if you're a business wanting to Manage your sales process, you can either use spreadsheets and build your own application to manage all of your sales processes or you can go to Salesforce and sign up for a CRM with them. They provide the functionality all of the low-level detail and you just get the benefits.

We have a list of interactive examples of our functionality that you can go and play with and just to give you some ideas of the sorts of things that can be done with that. There's about 30 different use cases in different businesses. One of my favorites is verifiable résumés in which an individual who's collected digital credentials or professional credentials can actually take them all and generate a verifiable resume that has references to all of each of those credentials. And they can provide their digital verifiable resume to a potential employer who can actually click verify on that resume. It will go to the issuers of each of those credentials, verify them and then return a trust report to the employer.

That's all I have to say for now. Thank you very much for your time and if you have any questions now is the time. – I got one question because of the term that you coined "trustability." I use that in a promotion on LinkedIn and it got a red dotted line under it. It doesn't exist in my translation. It got trustworthiness.

I heard your explanation of trustability but could we use the word trustworthiness as well in your… Of course, yes. The ability to be able to make a trust decision on something is my definition of trustability which may not be a real word. But I think it should be.

We're making trust decisions all of the while. But when I receive some kind of digital information over the internet that I want to make a trust decision on then I have, at the moment, only two choices. One's called blind faith. There's three actually.

Blind faith. The one is ignore it completely. And the third is spend quite a lot of time doing some kind of verification manually to decide whether this is genuine or not. If it's an email coming over from me, I look to see where it actually came from, for example. If you can trust that.

About a week ago, my wife had a phone call from the Bank of America saying that they noticed that someone is trying to initiate a wire transfer from our checking account. They said we we can help you fix this. They sounded very genuine and they said this is my name and if you go to the Bank of America's website go to a particular branch and you scroll down you'll see a picture of me and you'll see my phone number and if you look that the phone that I'm calling from is in fact that phone number.

This is me. My wife said, "Mmm, well, I'm going to call you back anyway." And that was the last we heard of them, except we called back and spoke to somebody completely different. We could have got caught out very easily. Because lots of people assume that if it says that phone number and they prove that the real person controls that phone number that's where it came from. Thankfully, we have Provenant who is solving that problem for us using KERI and ACDCs but for now we're in the world where we have to make either blind faith, ignore it or spend some time doing verification and and that's how we get caught.

Implementing execution-time trustability is where we think things should start. Thank you. Any other question? If not, then thank you very much, Alan, for your presentation.

TALK II

KRAM IT! with Ari

Ari Argoud

KERI Conference April 2026 · talks · 29:13

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Ari Argoud introduces KRAM (KERI Request Authentication Mechanism) for non-key-event messages in KERI v2. KRAM adds datetime stamps and receiver-side timeliness caches to block replay attacks on legitimately signed messages, sitting atop the non-KEL pipeline after basic allow-deny filters.

Hi everybody. This is KRAM It! With Ari. Important exclamation mark! placement there. KRAM is KERI Request Authentication Mechanism and this is for all non-KEL events or non-key event messages that occur in the KERI protocol and some examples include encrypt sender sign receiver which if you were at Phil's [talk] you'd know is a good way of doing Secure APIs to a given identifier essentially a replacement for shared-secret-based account management.

You can do exchange transactions for negotiating contracts or other negotiations. You can do credential presentations. And so the fundamental problem that KRAM solves is ensuring that a message is timely and authentic. Without KRAM, an attacker can intercept a valid signed message and then replay that message at a later time and you could have a ..., or the receiver of that message would have no way of distinguishing it from a fresh one which could cause all kinds of problems, like if you were signing on to an account or something, you could have someone who is able to sign on for you.

KRAM works by enforcing a timeliness cache window. Each message includes a datetime stamp and the receiver is going to check whether that time stamp falls within an acceptable [time] window relative to its own clock. If the message has been seen before, if it falls outside the window, it's just going to get dropped.

KRAM exists at the top of the message processing pipeline. All of the non-key event message flow through a non-key event messages flow through a single method and KRAM is applied immediately after some basic 'allow deny' logic if in case you want to filter for given identifiers, in case you want to not let specific people in. And I should note here that KRAM is designed for KERI v2 which is not out yet. [update July26: v2.0 is out] But it represents a change from a prior approach wherein before we would escrow messages when authentication information wasn't immediately available like if we didn't have the Key Event Log whereas now we just drop them and notify such that the Key Event Log can be retrieved via something that is out of scope for this talk.

Before diving into KRAM's mechanics I just want to make sure that everybody is familiar or largely familiar with these few key terminology points. Does anyone have any questions about KERI? Well, okay. Questions about KERI might be a little broad, but we all know what AIDs are, KELs, Key State, we've talked a lot about it today. SAID is maybe worth touching on, which is the Self-Addressing Identifiers, which are digests computed over the contents of a data structure.

And basically, this is an ID for a given message or in this instance, it is or referencing it as an ID for a given message. – It's a context identifier based on your payload? – That's that seems correct. – It's actually the whole message, right? – It's a digest of the message.

– It's a digest of the message? – But it's embedded in the message… […] protocol. That there's one universal identifier for that message. – It's unique per message even if applied in a thread. – Yeah. – Yes. – Could you repeat the conclusion of this because you are the only one with a microphone.

– Oh yes. Basically We have SAIDs as unique identifiers which are digests of a given message such that no two messages that are the same would have different sets. I.e. if two messages have the same set then they must have identical contents and time stamps. Then Seals, briefly is a cryptographic commitment which can anchor arbitrary data to a tree of hashes or to a particular event in the key-event sequence. And we're assuming with KRAM that the receiver will already hold the copy of the sender's KEL. Which is again kind of gesturing towards that fact that that retrieval mechanism is out of scope but it does exist or will exist.

In KRAM we have three authentication types. We have an Anchoring Seal Reference which is a message that's authenticated via a seal in the sender's KEL. And this works because you can preemptively authenticate your messages as a sender by digesting them into a seal with the authentications already created and then adding that to your KEL, and then continuing your Key Event Log. I think this would be done via an interaction event and then other people can just look up that seal and confirm that is a pre-authenticated message. We have Single-Key Signature which is just one key in the key list. I should mention that, let's see… when we are determining this type of authentication we're looking at key list cardinality.

Sometimes an identifier will have multiple signing keys but it won't be a multi key signature and for the purposes in .., eventually that matters, it is covered in KRAM because KRAM does authenticate but what we're doing here, why we're breaking these things out is because… … we're trying to do Amplification and DoS (Denial of Service) mitigation. It's important to do the cheapest method of dropping messages before we get into the more expensive authentication-based methods of dropping messages.

First we would… first we're going to look at the key list cardinality. We're going to say is it a single signature? Is it a multi key signature? Does it have a seal reference? And then if it's multi key and it has a seal reference, we're going to try and look at that seal reference first before we go on to see if that multikey has some validity to it, if it has signatures that are valid and validating those.

All right. We'll talk briefly about multisig. You guys have probably heard it all already today. But this is when multiple signing keys are required to authorize an action. These would be distributed signing keys. In practice, that's when a multisig group sends a message and not all the signatures may be attached at once. The different members of the group may sign at different times and from different devices. The receiver needs to collect those signatures, needs to collect those signatures, sorry incrementally until we meet a threshold which will validate that message if the message is able to collect all those signatures.

This is where we have different KRAM [time] windows. KRAM uses a long lag window for multi key signatures wherein as opposed to a short lag window for single keys and seal reference messages. The long lag might be minutes, might be hours or days whatever is appropriate for the operational tempo of the multisig group and the short lag only accounts for network latency and initial processing times. This is that window that I was mentioning earlier.

It's going to be different depending on the authentication mechanism that we suss out in the beginning of the KRAM-It method, which is why this [talk] is called 'KRAM It with Ari.' Let's talk about what a Replay Attack is I mentioned this already but I wanted to go through it again just so that we can create

some context for the next things that we're going to be talking about. Replay Attacks are fundamental threats in any authenticated messaging system. The concept is that an attacker will intercept a message that was legitimately created and signed and then is able to resend that replaying it to the receiver at a later time. If the receiver has no mechanism to detect that the message is a duplicate, it may process it again as if it were fresh. And depending on the message type, this could grant unauthorized access, trigger duplicate operations, or otherwise cause harm. It's important to note that the attacker doesn't need to forge or modify anything in the case of a replay attack. The message is genuinely signed by the real sender and the attacker can exploit the fact that message has a valid signature on it. But that doesn't mean that the message is provably delivered right now.

This is why timeliness is a core mechanic for KRAM. A message could be properly signed but it must also fall within the timeliness window that is able to correctly filter out any messages that may be sent by malicious bad actors.

– In current-use protocols, how is that mitigated? - OAuth2 or other examples – I don't know. I've only studied KERI, so I couldn't tell you how OAuth 2.0 does it. [inaudible] approach in most secure channel – is it a nonce – by TLS is you do a challenge response.

It's you […] with three messages. You send a request, the host sends you back a nonce, you sign the nonce or encrypt the nonce, send it back and then they decrypt it or verify the signature and that nonce is basically not replayable as long as they don't ever generate the same nonce and accept it. That means they either assume an ephemeral secure session so they don't have to save the nonces but if you want something like multisig that can span multiple sessions then your host because now you have to store all those nonces forever get replayed.

When you look at Replay Attack Protection, when you go away from ephemeral interactions to things that have to last for long periods of time, most of the existing mechanisms just don't work. They just don't scale. You have to remember things forever or you have to start to build provable caches.

– Statistically random with large – Yeah. – That makes sense actually. – Okay. – I want to ask to summarize. To summarize… other authentication mechanisms like OAuth 2.0 do multi- message authentication via nonces which are not scalable as such.

Now, talk about some fundamental protections of KRAM. The first thing that we're going to do when well, we're ordering these by the cheapest means of making sure that your message is authenticated per the given point in the protection that you're at. The first thing that we want to do is we want to look up this message. Like we said, it's got a SAID on it which is unique. We're looking it up against a cache. And if that message is there in that cache already, then we know it's a duplicate because no message can have the same set as a message that's already existed.

you've got something funny going on if you're… or well it could be it could not be something funny. It could also just be someone who's legitimately trying to send a message again. But the point being that the first thing we do cache lookup cache lookup and drop if we already have something there. We are also checking to see if these are multikey messages because we can have a message that has the same SAID but has additional signatures that must be collected towards a threshold. There is a little bit of nuance with respects to multi-key messages.

Next, you're going to do a timeliness check which only occurs for uncashed messages. And you're going to do this nice little formula here. This is the real magic behind KRAM or at least some of it which is a receiver date time minus drift minus ML which is going to be a… this is a variable, this could be

different depending on whether you have a multi key, it might be a longer lag or whether you have a zip wherein you're going to be waiting for an even longer time to do different negotiations of contracts or multiple messages within within a given very long time window.

Right, do you see here ML equals SL, i.e. we're doing for seal reference and single key we swap this out for a short lag, long lag for multi key, etc. Next you're going to try to authenticate. You're going to resolve the auth seal type by checking the seal reference first because it's cheap and then you're going to go on to signatures and with single key you just have to validate cache and accept. Multi key, again, you're going to verify the available signatures and then you're going to take a look at that threshold And see if it's met at that point.

And if not, then you're going to you're going to well, you're basically just collecting those and waiting for more messages to come in until that threshold is met. It's important to note that the prune [time] windows always must be greater than the accept windows which is going to prevent gap attacks during configuration changes. And again we're doing… Yes?

– The window is a measure of timing. – It's a sliding window. A given window will be an offset in one direction and another. Right. Like if it's what time is it? Three. – Or wall-clock time? – Say that last part again. – Is the expectation that it's wall- clock time or something else like tick counts?

– It's going to be wall-clock time, right? Network time. – Network time. – Yes. – Is there a dependency to have secure time? – Yes. No. Okay. – No. No. No. – What's secure? What do you mean secure time? – Time is always relative to the host. It doesn't matter what an attacker does. If they're not synchronized with the host time, they can't lie about the time, right? And and if the host records the timing, they can't do a clock replicate attack on it. So that when they spin back up, you They can't like push the time back. – Time, but the security is the the host itself. And the assumption is the attack [inaudible] – To be clear… – I don't know what you mean by secure clock but those [inaudible] – Yeah. But in this case server you just you are the server.

– Yeah. You are the server. – Okay. You are the server which is the security essentially to answer your question. Okay. That for the microphone. Let's see. Did I miss anything on here? Oh, right. Key State Protection. It's good to be aware that this is for a single given key state. And if sender key state changes in the duration of a given messaging exchange, like if you're collecting signatures on a multisig and the sender key state changes at some point during that, then that messaging operation is going to have to happen again.

– You said gonna have to what? – It's gonna have to happen again. – Oh, yes. – Yes. Okay, good. You're looking at me. I was like… – I didn't hear what you said. I didn't hear the… – Thank you. Okay. Now we get to talk about the cool stuff. It's all cool stuff. I'm just kidding. These are kind of… they're not introduced by KRAM, but having KRAM as a Replay Attack Defense introduces other modes of exploitation. We have gap replay attack and gap first play attack.

Essentially… I'm actually going to move on to the next slide because I have some nice visuals here for you. For a gap replay attack. Here we have our timeliness window. We have our message being sent. It's accepted and cached and then the message gets pruned. You have an old timeliness window right here. And after the message is sent, before it's accepted, a bad actor is going to go and grab that message. Right?

Next, the timeliness window expires. The message is pruned, but then you as the server or the receiver is going to change your timeliness window and increase it. Here after the message is pruned a bad actor can replay that message in this little vulnerable gap right here. We're going to talk about how we solve these with KRAM but these are complications that are introduced that need to be addressed.

– You dynamically change [inaudible] – If you dynamically change windows – Yes exactly. This is when you have a running system and you want to change your timeliness window for a given message type or other configuration.

For a gap… hold on… sorry this one should say gap first play visual. This is a gap first play attack. Basically what happens here is you have a message sent. A bad actor is going to intercept that message, but it's going to come through with the receiver of the message never having touched it or looked at it because it comes through after your timeliness window.

Post that. This is a little bit of a confusing time scale, but post that you're going to increase your timeliness window. And then that bad actor who's saved that message can first play it into your new increased timeliness window. – What's the intuition for why they increase third time in this window?

– Could be any number of reasons like let's say you're doing a multisig exchange or a multisig and the group participants are not getting their signatures in on time for given messages and you say I want to give these people a little bit more time. You increase the window but you also open yourself up to this… – Just on a network with lots of packet loss.

– Exactly. – The network starts to have congestion and now your latency [inaudible] – [inaudible] could simulate that using the Denial of Service attack somewhere in the network – and then [inaudible] it into the window.

– Yes, so to reiterate another reason that you could encounter this or that you could encounter needing to change your timeliness window is if you have a lot of network latency. Perhaps an attacker is DDoSsing you or maybe you're just scaling up and you realize that you need more time to process.

Anyways, now we're going to talk about how we address these. … oh… now this is the right slide. We have three cases. If you're decreasing a [time] window size, that's safe and you can apply that immediately. If you're increasing a window size, you need a two-step process. And if you're changing cache type granularity, I'll talk a little bit about configuring cache types in the later slides. That's more complex and would require a worst case analysis, but that's a little bit out of scope for this. Anyways, the two-step processes for increasing it is you're essentially going to want to have a prune lag wherein when you do an increase, you have an extra lag where you're not pruning windows until you've exceeded the time delta between your old timeliness window and the new increased timeliness window.

And that actually covers both gap replay attacks and gap first play attacks. Then we have a little bit about the configuration wherein this is designed for incremental adoption. Not every deployment is going to be able to run KRAM for all messages types overnight and the configuration system reflects this. We have up here the 'enabled true' which just means your KRAM is on. Then you have denials and these are each keyed to KERI version. We have version one and version two right here. And these are going to be message types. And these are going to be… These are going to be… route. Sorry.

And the denial systems supports wild carding through empty fields. These one comma zero KERI version one wild card wild card just means we're going to deny all KERI v1 messages for KRAM or we're not going to use KRAM on KERI v1 messages, right, and for these we're not going to use KRAM on

KERI v2 messages with a pro message type and then any route and for the third one we're not going to feed messages through KRAM for KERI version two reply messages with a “/end” route.

And the denial logic is applied at the top, before we even get into any of the message processing stuff that we discussed earlier. That is pretty much all I have. I was also gonna say that a lot of people worked on this. Sam wrote the white paper. Several other people from the KERI Foundation have put a lot of effort into getting the implementation to where it's at now. And I know that many other people are going to work on it before it's complete.

I give them a round of applause. Does anyone have any other questions? I run a little short. – Yes. – Where do you see this being put into application within, let's just say the healthKERI platform. – This is going to be like for say for example for our account communications with ESSR they could use this. For any credential presentations they could use this, right.

We were talking a little or we haven't talked about this yet, but actually… I'm going to abstain because I'm not sure if I'm supposed to say that yet, but those two use cases are pretty relevant to us as healthKERI. – Absolutely.

– Yeah. [inaudible] – Yes. – Yes. IPEX EXN messages, all covered by KRAM. – Yes. – I just want to understand the problem a little bit better of why we need to keep this just I think that's what he had asked. – Oh, sure. – I understand it replaces the nonce, but why is that beneficial? Like what?

– What? – Yeah. Just so if you have a nonce I assume it's kind of like a temporary token. – Yes. – That's what I've used in the past. That then it would increase the database size or maybe you have to have connectivity back to… I'm trying to figure out why.

– It's like an elegant way of solving replay attacks without attaching nonces to every message because the time itself functions as the nonce, like the time attached to the message. It's not that we don't have a nonce the time is the nonce right?

– [inaudible] interactive authentification right so if you think about network latency we're doing at scale requests every request has to create a TCP connection because it has to be a secure channel otherwise you can't control it, so all asynchronous messaging is off the table. This will work just as well with asynchronous messaging as synchronous messaging.

It'll work just fine where the sender and receiver use a completely different channel. Doesn't matter. Because there's no interaction, right? Now you can get rid of all of those constraints and then the latency for a request is zero because the request either goes through or it doesn't.

Whereas if it's interactive, you send a request, you get a reply, you send a response, and now you've got three passes across your network, which if your network's loaded or congested, you triple your latency always. And when you go to scale, that doesn't scale well. And So that's a reason not to use interactive.

Two reasons, one asynchronous messaging …and remove the latency at scale. – To reiterate for the microphone: this is asynchronous which traditional nonce-based authentication is not and this also works at scale without having to rely on the traditional mode of exchanging several messages in order to meet the same functionality.

– [inaudible] statement, right? You know what's what was the difference between KERI not having KRAM and now KRAM existing and I'm just maybe restating but […] restated for the replay attack but so was that a vulnerability and existing vulnerability right now?

– Yes. – KRAM has been proposed for quite quite some time but nobody thought it was important to… – In our authentication server, we have a custom build replay [inaudible] we can now get rid of that once we upgrade KERI 2.0.

– Yeah. That was traditionally not covered by the KERISuite or KERIPy… We want it to be something that everybody gets for free. – Awesome.

TALK III

Ari-Argoud-Interview

KERI Conference April 2026 · interviews · 01:01

Hi, my name is Ari Rgoode. I am a full -stack Python developer with HealthKERI, and my presentation was on KRAM, which is KERI Request Authentication Mechanism. I think what I do and see differently is I'm really engaged with moving the healthcare industry specifically towards a repudiable, individually owned mode of data governance and transmission. If KRAM disappeared tomorrow, we would lose a huge step towards replay attack protections in the KERI ecosystem.

Previously, everybody had their own bespoke methods for doing this when implementing with ShareAPI, and it is very beneficial for the ecosystem to leave everybody into the same modality of protection that is proven and trusted throughout the different implementers.

TALK IV

Contract Credentials & Verifiable Identity

Cole Davis

KERI Conference April 2026 · talks · 35:42

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Cole Davis introduces contract credentials, marrying legal contracts to ACDCs so holders prove contractual authority beyond identity. Using a commercial lease example, he shows how Switchchord binds prose, schema, and issuer AIDs into derived credential chains for assets and access.

Hey everyone. I'm Cole Davis. I am the founder of Switchchord, which is a music, we're legal tech and digital identity company building products in the music industry. That is one brand under sort of an IP holding company we have called Steamroller Systems that is industry agnostic to apply machine learning and decentralized identity techniques to legal workflows. We kind of blend the worlds of legal tech and digital identity primarily using decentralized identity. And I'm going to talk today though not about music because I always talk about music and I wanted to change it up. And today I'm going to talk about, I call these things, contract credentials. I've for years advocated that legal contracts and and verifiable credentials should be sort of married together like a legal… like I wanted to create a dual root of trust like a legal root of trust and a cryptographic root of trust and I want them to be inherent in everything you do when you prove you have contractual authority. I spent years looking for solutions starting in the blockchain world and then the W3C DID VC world discovered KERI in 2021 and finally was like that's what I need. Primarily because it doesn't require like an underlying ledger. I can get into why the Switchchord stack is built the way it is but for now I'm just going to focus on the general concept. And then I did a demo in the real estate industry because it tends to be easier to understand and you don't need a whole lot of like domain specific knowledge about music to really appreciate it.

Because you apply the tech anywhere, anywhere you have a legal contract or contractual authority or contractual relationship you can apply this framework to it. That's what we're going to talk about today. Credentials a lot of times at IIW and the W3C and the KERI community, we talk about credentials proving who you are. You're a human. You're an organization. I don't often see them proving what you have the contractual authority to do. It's like not a concept that comes easy to people. But when I explain this to the music industry, I often use the example and this actually did start to resonate of a mobile driver's license cuz some states have mobile driver's licenses and they sit in their Apple or Google wallet. I explain, driver's license is a credential issued to you from an issuer. It has claims about what you can do and it's kind of digitally signed and you can prove to third parties that you have the authority to do something. That's easier for people to make that mental leap of .., Okay. The contract credential is kind of like this, but it tells me I can sign documents over 10 million dollars cuz I have this authority from the board of directors or I can enter this building because my employer has a lease with the building owner and I am an active employee with the right to enter the premises. What I wanted to do was the way a lot of people talk about creating like verifiable presentations or capturing claims from different credentials and presenting them about you. Oh, I'm Cole.

I graduated from TCU. This is my GPA. I have this job. Okay, cool. I'm more interested in the commercial applications of this, as far as commercial transactions go. I want to be able to say "I'm this organization. I have the contractual right to do this.

I have contractual right to do this." I want to be able to pull select from different contracts that give me the authority to do transactions on a market and I want that to be instantly verifiable to third parties who want to maybe do business with me. Because it gets into that Ronald Coase world of like transaction cost. How can we reduce transaction cost? Verifying counterparties and their reputations and all that and their authority to do things is like really important for transaction costs. If you can reduce some of this technology, you can have more transactions on the market.

That's what my concept of a contract credential is like "What if they're the same thing. What if the contract is the credential? Like what if you can hold your lease up to the smart lock or what if you can [make it the same thing, red], the stock option agreement or this stocks purchase agreement is the stock certificate?" Stop taking the contract here and then the authority or the thing it represents over here and like "No, make them the same thing and have a shared cryptographic and legal root of trust." That's basically what this is. There's an authority gap like credentials they don't model contracts they live in separate systems.

You have a contract in your rights management system or contract life cycle management system. And then you have the operational aspects of the contract, what you can do or what you're required to do or can't do. Those tend to live in different systems. What if we collapse them into a single system? What if it's all the same? Like it's a really weird concept, but it's really powerful. And we've been… Did you have a… saw hand for just a second?

– I have a question [inaudible]. What's the difference between a credential, a role and the terms that the ACDC can define? Is that kind of combined or [inaudible] there as well? It's flexible enough to where like the vLEI, ECR and role credentials and stuff are sort of derived from this whole general technology but it's a little more specific. This is more general. This is any employment relationship, any board granted authority, anything can be modeled into this framework using ACDC. You can do it with W3C VCs too, but there's just like limitations there that ACDCs give you. Ability to chain credentials and they give you a better ability to pull in legal pros and the Ricardian concept. And ACDCs are way more powerful to the concept of a contract credential.

You can do it with W3C VCs. You can do it with whatever you want as long as it's cryptographically verifiable data that is derived from a contract. But it needs to all be like one big signed object. The way I think about it is okay, you have a legal prose .., I can't remember if I actually have like the three .., I do. I'll focus on KERI ACDC's what the what GLEIF came up with how you can take that. Taking the legal relationship that is embodied in a contract that's usually a bilateral thing. It can be multilateral. It's usually a bilateral thing. You can model the relationship: I, landlord, give you, tenant, the right to enter premises for one year for this rent and you can do x y and z. You can model that relationship between counterparties and assets. It doesn't have to have an asset subject to the contract, but you can model it. In the music industry, model it as far as what you can do relating to copyrights. Copyrights have in our system DIDs. They could have AIDs.

It doesn't matter. You're modeling legal relationships between organizations, people and things through contracts. The contracts are like the the routing mechanism for the data. And you pull in the AIDs or the DIDs into the data model that is behind the legal pros. You got like, it looks like a PDF or a Word document but you can use a markup language you can have a data model, you have JSON… Party A is whatever Acme systems. Acme systems can have their AID or DID in the data model and you're modeling…it looks like a English name but you're pulling in the identity the decentralized identifier with all you know the key state and stuff and then that whole object is what's going to get signed. You're attesting to this is our relationship it is embodied in this thing and this thing can then generate these

credentials that has a one to one mapping to the legal contract and a one to one mapping to the identities and the one to the counterparties and a one to one mapping to any assets that are like the subject of that thing. That's what we do is model legal relationships. Why KERI? You guys probably already know all of this, but AIDs, it's a short talk, I can't go into like why I stumbled upon KERI and what I was looking for, but I didn't want… we were modeling the legal state of a copyright. Let's say four different music companies own a copyright. It's very common. Okay, cool.

We're going to model the copyright as a DID. Okay, who are the controllers of the DID doc? Hm. This should be the owners. We can model that through the contracts that give those four music companies ownership. They can all have DIDs, controller properties. Cool. Oh, shoot. They want to rotate their DID. Aah, we're all using the same DID method that relies on a ledger. Now I have to explain to like these big well-known music companies: I need you to interact with like the Ethereum blockchain. It just doesn't work. And I wanted all of them to be able to manage their key state separately. And that's… I discovered KERI and I was like, that's what I want. Like everyone needs to manage their key state separately, but you can still compose a full chain of title as far as assets go. It just also happens to work really well for contracts, too. KERI AIDs, the base layer. ACDC's have a lot more power than W3C VC's. You can do it either way but ACDCs again sort of the the rule, the Ricardian part, the attribute section and then the edge capability to chain credentials is really important. The the TELs and the KELs like being able to anchor a credential to a TEL which is anchored to a KEL is really important because if the landlord, if the lessee stops paying the landlord, it's a breach and they revoke… they terminate the lease. You want the credential to automatically revoke, so that the tenant can't get in or the tenants authorized delegated employee can't get in you want the whole thing to break at once when there's a breach of the contract. The contract should be like the actor in the system. And vLEI, like GLEIF's already done so much work here with sort of bringing this into what we consider legal workflows.

This stack is a great way to do this. This is sort of legal prose. Let's say in the R section of ACDC you can have the actual prose or you have the digest of it as a SAID. You have the structured data. The attribute section which is just JSON like "term: one year, rent: $100,000, square foot: 4,000." And then you can have the edges. You can be delegating contract like contractual authority. Which is really important because that's how things work in the real world is a lot of times like the actions you take are from a couple different contracts. Maybe they're in a chain, maybe they're not.

Either way, you can sort of like compose it into a proof using this stack. I pulled just a random corporate or commercial lease building lease from the SEC's Edgar platform. This is an actual commercial lease that was used by big sophisticated companies. What you can do is, you can have let's say they were negotiating this lease. They can negotiate in Word using a markup language. You can sort of put it into the data model that would then populate the ACDC attribute section or it's the same thing as the JSON schema of a W3C VC.

You're tying the schema to the prose and then from there. Let's say we're doing it for an office lease. This is kind of what the ACDC fields would look like. You've got the issuer AID. You can have the attribute section which is the deal points. You can have edges which I'll show you for this one. We can create the access credential from the lease credential. You can have rules all the technical aspects of an ACDC.

One way to show how this can become a chain. Let's say they sign that data object. Cool. And now we've got a credential that is derived and cryptographically married to that contract. Any asset can have an identity. Assets should have identities. The building can have an AID. In our in Switchchord all the

copyrights have DIDs, DID:web DIDs just for simplicity. We did an implementation with DID:Webs to show that KERI can be used in that system.

All of our assets should have decentralized identifiers and then the controllers of those assets should be the controllers the owners in the real world either ownership or like administration rights. All of this can be modeled. Within this example, this building where the commercial lease is has an identity, an AID. And then you've got a lessor an AID, the lessee with an AID. You have the lease contract. And you can derive the access credential from the lease itself.

The lessee can issue it to itself as an access credential. There's lots of different ways you can do this. This is just one example to kind of show what it would look like. Again, lease credential .., the contract credential that's derived from the contract using Sam's I2I versus the NI2I. That would be the issuee, the lessee issuing it to themself and they can now say like I have it from this contract and it can be then shown to a smart lock. This references the building identity, the issuer like in this case this is just a lessor creating it themselves but, let's say, I build a building with someone else then we can actually have a contract that says all right I own 50% you own 50%, and that contract can then create like the building ID there's different ways you can do this there's not like one set way it's really flexible: you've got the building identity that gets pulled into this that gets pulled into that derived credential that you can show to a smart lock. This is that the I2I operator just one example of what you can do here. This would be the lessee of the lease contract granting themselves an authority contract. You could probably do it without that. I was just trying to show a simple example of how you can have chained delegation from a single contract.

You could probably just have this be the lease credential too. But if you're issuing this instead of issuing your employees like entrance badges, you just issue it as a credential their phone and then they're they're basically the ones doing a delegated access credential from their employment agreement credential which then goes to the lease credential where your employer is the tenant. That's how you can build these chains that can automate legal processes.

The rule section you can, like Sam's done, very forward thinking here with sort of putting the legal prose in with the credential itself. This is kind of optional if you were to still do the same framework of a data model in a legal pros contract and the whole bundle is signed. You kind of have it already, but you can get even more granular. And this is what is really helpful for selective disclosure.

You don't want to show the whole contract. You just want to show I have the right to do this one thing. Like I don't need to show you smart lock how much rent we pay. I don't need to show you any of that. It's just like selective disclosure for a human proving attributes about yourself. You don't want to show your home address. You don't have to just show that you're over 21. It's the same thing. You can selectively disclose or graduated disclosure your contractual authority which is extremely powerful if you're building towards trust with a counterparty. This would be the office AID. Creating a digital twin is represented as an AID. Again this one is issued from an ACDC that the lessor AID like attests to the building. I own the building. This is the information about the building and here's the building AID. You have a smart lock that knows I control access to the building AID, you can build logic into that but it knows I am here solely to let people into this AID. If you show me something that does not line up with his AID, you're not getting in. If you show me something that is to this Aid but is revoked, you're not getting in. If the smart lock becomes .., you don't need like a central registry anymore. It's like the credentials from the contract and the smart lock just knows it. It's a mental leap for a lot of… If I were to take this to market as a product for say commercial apartment like apartment managers or commercial building managers they'd be like what?

But it's really powerful. It cuts down on much infrastructure that's currently required. Graduated disclosure [is] super helpful. You can, for this example, you could just say to different verifiers you can disclose different things.

Like the smart lock, you don't need to disclose the rent term or the .., well maybe the term, but not the amount of rent. But to your financial auditor, maybe you need to do the whole thing to a third party. The graduated disclosure of contractual authority is incredibly powerful.

Showing them the whole contract. That's not good. You can just show what you need to prove. – Litigation, you might want to just say even if it is a litigation. Oh, totally. – Could you repeat the question because… – In litigation .., there's litigation over a contractual provision, there's all sorts of what's evidentiary rules. I took evidence in law school, but I'm not a litigator. I'm a corporate lawyer, corporate entertainment lawyer.

There's lots of rules for that. The judge can order things redacted, but this would be a bulletproof way to redact things. You don't have these boneheaded, the PDF is redacted with a black bar, and you can like delete the black bar, you could actually have cryptographic opening up that .. And then you can really dial in who has the ability to do that; it's crazy what you can do. This could certainly be applied. If there was a litigation over this commercial lease, like you go to arbitration instead of litigation. The arbitrator, a mediator, okay, maybe you don't want to show the arbitrator. It it's really cool that you can selectively disclose things and know that it's true like cryptographically know that that's the state of this contract. Revocation would kind of use the KEL and the TEL framework to say all right lease terminates the lessor would create revocation event in their TEL that's going to tell the smart lock that at this time period with our key state we revoked this credential that was issued with this signing key in our KEL, that's how the lock would .., the smart lock just walks the cryptographic operations all the way back and it gets to the point of "Sorry, this thing's been revoked."

It's really powerful to be able to present contractual authority. The easy thing to think about is like proving it to a human or an organization, but it's like no, no, no, how about a software system? How about you're trying to get into a certain secure database in your organization and only certain engineers have that? Like "that could be done through the employment agreement."

It's really powerful when you start thinking about it in these granular ways and then as soon as you're fired you don't have to do anything else like it's done, it's like "Everything breaks" versus "Oh my god, we got to log in to this system and revoke his authority and this system revoke like everything once you start to view the world this way .., I can't undo it like I view the world differently that this is interaction between contractual authority and your rights as a human and it can all be bound up and it would be cool if you could pick and choose from all of them. I work here, I live here, I am employed here, I have the contract right to this and I can just take what I need to prove anything. I focus on the commercial transaction elements. The vLEI, you can model this based on the GLEIF route to the QVI to the legal entity to the role credential and the ECR.

That's a little more specific to the GLEIF environment. This is an abstraction of that. Anything can be turned into this. We could even abstract the GLEIF one to say "Okay, the board of directors there's five individuals they all have AIDs they all agree in a written consent to give Karla authority to sign contracts up to ten million dollars. They digitally sign this prose looking consent that creates the board consent credential. The credential itself can be chained to the employment credential of Karla. Karla's credential gives her certain her employment relationship ship gives her certain rights. The board has now given her certain rights. We can pull those together and now when she tries to go into Docusign, she's asked to

present her authority credential, which can be a derived credential. The contract is 20 million. Karla needs to get more authority. It's a much more abstract and granular notion, but it's the same general framework. And that's powerful because we know it works like GLEIF is in production and it works.

I'm trying everywhere I can. I'm trying not to reinvent things that already work. I didn't have time and I don't think I really can. Like I've gotten pretty good at vibe-coding legal products. I built it myself as a lawyer to automate things, but I tried to build this and it was just too hard to do the actual KERI stack.

I built like a simulated version, which I'll walk you through. You got the AIDs of all the parties. The lessor will create the building AID. You then sign the lease. The lease then populates the schema of the hybrid legal contract populates the scheme of the ACDC. The attribute section. You then derive the access credential from that. It's an issuer the eye to eye transaction from lessee to itself and then lessee can present that. The smart lock would then verify back.

I vibe-coded this. This is based on the ACDC like IETF, whatever it's called specs. Everything's like very much to spec. I didn't have the time or capability to spin up all the infrastructure to make this real. You've got this contract is like I said, this is a real contract. I found it on the SEC's website. I was like, I want something; it's 45 pages, really dense, really complex. And I used that for this demo. This would be the inception event for let's see, this is the lessor. The property owner, and this follows, everything that you technical people in here are familiar with all of Sam's fields. Same with this one. You've got the AID in there, somewhere. I think it's the I and then you've got this building. This digital twin with an AID, public key and then it shows it shows where ..

I don't have the ACDC white paper IETF spec memorized. I get kind of lost when I'm looking like which one is the actual issuer, but it would be AO6. There it is. Okay. That's the one that's the lessor. Now we've got AIDs for the lessor or the lessee in the building. Then we would issue the building identity credential.

This would be issued by the lessor that attests to the building and gives the attributes about the building. This is establishing the idea of the building that can be pulled into this concept of contract credential. We've got the lease. I don't know where I put the PDF if I had it.Tthis is the actual language from the lease. This is how you could take a markup language to then the prose version. But if you look at it with X-ray vision, you're seeing this behind the scenes.

And then this is what's populating the ACDC schema or for a W3C [credential, red.] it'd be just like the JSON in the VC schema. That's how you're having a one to one mapping between the contract and the credential. And then that whole data object can be signed. You've got ideally the same public private key pair is signing the PDF as it's signing the data model and the credential. It should all be the same. That's how you can like bind everything together. And then you've got, let's see, lessee receives the credential now acts as the issuer. Oh, this is where .., just to show the delegated and the chaining mechanism, this is where the lessee can create its own access credential. Let's say this is for an employee. They issue it to the employee.

– I'll ask a question, but please repeat it as well for your mic. – Definitely. – What, it's a bit off topic, but how much time do you think will we have to endure before this becomes all graphic like with puppets? Lessor, lessee… You can click on them.

– That's what it should be. Totally. – What do you think? What kind of… Please repeat the question. – Yeah. – How soon? How much time? Yeah. This was me. I built this I built all of this in about 4 hours late at night when I needed to get what was due. I forgot what the due date was for the slides, but I was like,

"Oh my god, I still haven't done them yet." I did everything using Claude and Cursor in including building this in about 4 hours.

I could probably turn this into a much prettier UI. This is basically a one-shot attempt. I mean Opus 4.7 did all everything you're seeing here is like me working with Opus 4.7. – I'm just unaware are there already toolkits to make it more graphically [attractive, red.]? The question is on UI/UX of all of this. I don't know.

Decentralized identity tends to have a pretty bad UI/UX in general. It's hard. Some companies have done a much better job than others. I would prefer to see it. Again, I'm really just trying to show just like how it works, the nuts and bolts using the actual ACDC spec. There's no way I would show like build a product that looked like this to like an apartment manager. There's zero way. This is all behind the hood.

– I think that's the solution, I think, to what you're saying is I don't think there's going to be a one-way-fits-all way to view all this. I think it's going to be individual apps. That are encapsulating it all. Yeah. We built this amazing platform for the music industry.

It was five years ahead of its time which is why it was hard to get off the ground because no one knew what we were talking about. We're taking songwriter agreements. We're dropping them in. We use AI to extract everything about the legal relationship between a songwriter and a music publisher.

Populates a W3C VC, issues the songwriter. Now we've got this verified connection from the writer and the publisher. Writer writes a new song, drops in the information right here on their phone. Boom, shows up instantly on the publisher side. It's tracking from the contract. It's almost like the data is getting stamped by the contract as it goes through the pipe.

It's amazing, but no one understands it. The identity thing, no one cares that we built this cool thing where you can like click the credential, it flips over and shows the JSON schema. Everyone's like, how does that help me? That product, I don't want to say it failed. It was like a great product and it actually this one feature of that product is now what is getting a ton of interest and is working for us and we will Trojan-horse the identity stuff back in, eventually, but UI/UX is just hard with this stuff.

– More comment than question at this point On this model from a physical security standpoint you said this is a management company for an apartment building. Most places today have a keypad out in front that you put in and it calls up or whatever lets you in.

They're made by two manufacturers. You can buy the three keys that open that box everywhere in the world. For $18 and there's a button inside that overrides the system and opens the door. To your point, if it was on the phone, you could completely do away with that!

You have biometric. You marry biometric to cryptographic to legal… – That would save these folks… So much money cuz they they have to go pay insurance. You got to pay insurance. Anytime there's a risk, you got to insure it and that's a transaction cost that doesn't need to be there. That's what I'm saying. Like this magic cryptography can just like really reduce transaction cost everywhere you look. And like I said, once you see the pattern, it's really hard to unsee. Now everything I look, I'm like, "Oh god, why do we do it that way?" But his would be great. And this would be instead of most apartment complexes don't let you do Airbnb, but people still do it because you can find ways to trade the key fobs.

Like this technically kills that. Or you can put in place rules and governance that says you may rent out your apartment, but the other person must be verified and have a credit score over X. Like you can build in the logic. They get the credential, they buy a metric, they show it, they go in. It's really powerful what

you can do with this stuff. – You have five more minutes.

Okay, cool. This is the example of showing it to the smart lock and this is what's happening is, it's checking this is like checking all the KELs, the TELs, traversing the nodes, Again, that didn't actually just happen with the true KERI stack, but this is true to the ACDC spec and like how it would happen and access granted this is the tenant, that was the tenant with that AID, they can get in 24/7 to street 105 open it up and then this would be what happens if the tenant doesn't pay their rent and the lease is terminated. Therefore, all building access credentials of employees are automatically revoked. That's what this one would be. Let's see. Attempt access with revoked lease. It kind of does the same thing. It's running backwards and checking the TEL, seeing that that edge on the lease has been this TEL was revoked. It has the registry AID, the TEL sequence. You can, check the signing keys at the time. Access denied. Reason, upstream credential revoked, lease contract, lease is terminated.

That's to me, much easier to understand than the music stuff, and it's fun to apply this to other things. That is that's the demo. That's that. And I think that is kind of the end of the presentation. That's what we're doing.

I guess we have five minutes for questions. – Yeah. Yeah. What's the… Have you studied you got a lawyer [inaudible]? Have you studied the enforcability angle of this? There's no enforcability issues cuz you start with the legal prose contract as the base.

It's all enforceable because you have it all like you've got it, you Different states. In Wyoming's been pushing the boundaries on this and others are smart contracts are legally enforceable like blockchain based smart contracts. That's fine but we're not going to get rid of human communication and legal prose.

it's not going away anytime soon. I don't think we're going to have some domain-specific coding language where the language you write is the contract and it is the code. I just don't see that happening. I think you have to take this sort of hybrid approach of there's disputes. It's going to have to go in front of some sort of judge or arbitrator or mediator or something. And they can't read the case, they can't read the ACDC spec. It's not going to happen.

You still have to have that original legal prose. You can pull it out, you can hash it, “SAID” it. I don't know how it's Sam likes to pronounce it. It's different every time. Said it. Said said. Okay. You can said it. It's just a hash digest. You can do whatever you want, but just don't think you'll ever get around needing a contract.

– Is it just me … KERI is incredibly powerful, but it's very hard to explain to people. You see, have you seen that? – Yeah. KERI especially. Like when I explain to the music industry, I can't bring up KERI. I just have to talk about basic JSON and W3C.

make it easier for consumption. I think it just needs to be in the back-end of products and I think you just can't even really explain a lot of it. It's like most people don't know how the internet works. – It's got to be user experience.

– Yeah. Like 99 people% of the people don't understand how the internet works. – [inaudible] That's exactly where I am. If they can connect to the business aspect of things, they go. Yeah. That's what we're trying to do. We're getting closer and we've been working at this for four or five years now and we're finally getting somewhere.

– We have to stop Totally. - Just do it. Just do it. Just use it. Make people fight you about it. Yeah. [applause]

TALK V

Cole-Davis-Interview

KERI Conference April 2026 · interviews · 01:51

I'm Cole Davis. I am the founder and CEO of SwitchCord. I'm a lawyer and technologist and I spoke today on what do we call it? Contract credentials and verifiable legal authority. Modeling legal relationships stuck in paper-based agreements, PDFs, Word documents into KERI, ACDC spec to be proving your contractual authority to third parties. I see things differently now after living in this world for the past six, seven years. That contractual authority and your ability to go through life and do things depends on legal contracts and authority from the state and you can't get away from it. I stopped viewing these things as separate and it's all sort of this interrelated mix of identity and legal.

And that led to me also seeing finance accounting, legal finance and accounting is not separate disciplines like I was taught and have degrees in, but it's really a spectrum. And identity, I found, is the one thing that ties all of them to together. It is the backbone of commercial transactions. And I think people are starting to come around to that idea. I've been saying it for a long time and it's exciting to see people, their eyes light up when they realize what I'm talking about. If my work and research were to just disappear, I think it would be I've been working to push this society towards Ronald Koest's magical world of zero transaction costs. I'm using identity and AI and every tool at my disposal to reduce the cost of doing commercial transactions to bring shared economic prosperity. If it were to go away, I think we'd be stuck with centralized databases, centralized processes that don't scale and are not they're beholden to the existing gatekeepers and it does not allow us to push forward as a society. Thank you.

TALK VI

Applying KERI to General Evidence Problems

Daniel Hardman

KERI Conference April 2026 · talks · 39:53

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Daniel Hardman argues dossiers extend KERI/ACDC beyond typical verifiable credentials, enabling one-way, multi-party, and out-of-band verification of rich evidence bundles. He describes curation workflows akin to loan applications, where assembled proof supports investigative or regulatory review without entitlement-style presentations.

Okidoke. I suppose everybody's pretty tired at this point, but I hope that I can keep your attention and keep you awake through this presentation. This is about Dossiers. I think that this is actually one of the more profound topics that this community could be .., thought leaders and world experts on there's a lot of untapped potential in this and I don't think we realize how cool that potential is. I hope I can convince you and get you guys excited about this.

This topic represents kind of the synthesis of a bunch of stuff that I've been working on in the telco space, but it really is not a telco topic. I remember when I first heard Sam talk about ACDCs circa 2019. He came to a Trust over IP working group meeting and started talking about ACDC theory and Sam mentioned This phrase that ACDCs are a verifiable data graph and at the time I thought oh that's cool and it didn't really like impinge on my consciousness that deeply but I have since come to feel that's actually quite a profound statement and there's more to it than we give it credit for and we could use this to solve some problems that people can't solve elsewhere.

Let me describe the problem as I encountered it in telco. I came to telco from a background in verifiable credentials. And in verifiable credentials the assumption is that you prove something to a verifier and you the party who is proving something are having a direct interaction with the verifier.

Right? The verifier sends you a proof request or something and the proof request arrives at your wallet and you get you know it gets your attention and you make a decision and then the proof flows back the other way. Well, guess what? That does not work in telco, at least in the use cases I was exploring, because for one thing, we wanted it to be the case that the party that emitted a communication and signed something Could have that thing that they're trying to prove verified all along the way.

Not only that, the party on the other side doesn't have a channel to go back yet. We needed basically one-way multiple-party verification and not only that I needed verification to be possible Completely out of the out of band as well like having an auditor or a regulator pull a transcript of the conversation and verify it a day later a year later or whatever to see if we were taking the right actions.

And again that's not really what verifiable credentials are built to do. This is all setting aside the more fundamental problems which are, you Verifiable credentials have mutable schemas and all those other things. But anyway, if you set all that aside, I still couldn't solve the problem, right?

The other thing that became very obvious is that we needed a much richer evidence model than the typical verifiable credential would provide. Almost 100% of the verifiable credential workflows that you see demoed or discussed anywhere out in the community are single credential workflows: "Show me your driver's license" or "show me some subset of the driver's license, " right? Or "show me your passport" or "show me your COVID thing" or whatever. But there are all kinds of scenarios in real life every day that call for very rich collections of evidence.

I listed a few up here and I put them under kind of headings because I see these patterns showing up. There's all kinds of processes in the real world that are evidenceoriented that deal with applications, right? You're going to apply for a grant.

You're going to apply to go to college. You're going to submit a regulatory filing. You're going to apply for a job. All of these are processes. They're typically multi-step processes where a large body of evidence has to be assembled and submitted.

That's not the same pattern as this nice tidy request response Of verifiable credentials. There's also research activities that happen. You're conducting a scientific study and you want to compile a whole bunch of evidence and if you'd like to solve the problems that we have in the world of academic research with a nonreproducibility, you'd like that evidence that you compile to be verifiable, right?

Investigative journalism, Criminal investigations, all of these fall into this category as well. And notice here, the evidence that you're putting in a lot of these things accretes over time. It's not just evidence that you just have in one issuance event and then you're done.

When you're trying to document something, maybe you're trying to prove Some things about the reasons why your company was valued a particular way to regulators or investors, or you're trying to capture the results of a forensic accounting audit. Or you're trying to submit documents for a court case, A court filing. All of these are cases where evidence is quite important and verifiability is quite important.

But the simple trust triangle that we talk about in the verifiable credentials world is really quite different from what the patterns of behavior are. Another thing that became really clear to me is in realworld hardcore ambitious use cases.

Lifespans of evidence, is a big deal. It's surprising .., now that I've become sensitized to it, it's almost shocking to me that this is never said, but you'll hear somebody analyzing you know can we prove 'xyz' with let's lsay with certificates or with credentials or whatever and they'll go through a whole deep analysis about whether it can be proved and they'll completely omit the fact that they've got four different kinds of evidence that they're trying to prove with. And the evidence has different lifespans, right? One piece of evidence that they're building from has a year-long lifespan.

Another piece of evidence has a 20 minute lifespan. And when you combine those two, what's the combined lifespan of the thing? 20 minutes. But nobody ever analyzes it that way. They're the very naive views about lifespan of evidence.

And I just kind of randomly picked it: Three different categories. You could group evidence into very different lifespan categories than these ones I picked. But the point is we need to start thinking about the lifespan of evidence a little bit more carefully. And if we do, we're going to realize that the simple way that we've been trying to solve certain problems is really quite inadequate.

Another thing is that the purpose of the different constituent pieces of evidence that you might assemble in a collection could be quite different and have very different assumptions. You don't see this when you're only doing kind of .., it's dismissive for me to say "Mickey Mouse" demonstrations, but when you're doing demonstrations are really quite simplistic and you're using one credential and you're saying I can either show you all of my driver's license or part of it, you don't realize this, but as soon as you start assembling Evidence to solve a real world problem, you find that the different kinds of evidence that you're grouping together introduce some interesting complexity. In this particular slide, I'm

imagining what's the evidence that a loan officer has to assemble to get you qualified for a mortgage. Okay? And I just profiled four different kinds. One piece of evidence that a loan officer has to assemble is your last three paystubs.

Right? Now, how those last three pay stubs are verified is pretty different from some of the other things up here. And how long those paystubs can be useful is different from some of the other things up here and so forth.

That's the last three paystubs. And the purpose of putting it together in this collection is, you're trying to prove that the person has a certain kind of income. And that means (my red text there) these have to be three consecutive paystubs and they have to be recent all these other rules apply to that right then how about proof of employment that's a whole different kind of evidence and proof of employment is probably you want quite recent proof of employment maybe a different recency standard than you have to meet with the paystubs and the kind of credential you're using to prove this is quite different as well and it has a different issuer, right?

How about the credit report? Yet another issuer, yet another standard, yet another governance, and so forth. If you're really trying to solve a serious use case with rich evidence, what you have to model about that evidence and prove about that evidence to achieve what Sam said years ago about a verifiable data graph is really quite sophisticated.

It's not as simple as "Oh, show somebody a credential" and you're done. You also kind of need to know what's the purpose of the collection of the evidence you're putting together. You're not just stuffing evidence in a drawer somewhere so you can call upon it. In the case of the loan, going back to that, all of these pieces of evidence that I was talking about are going to be evaluated at funding.

That's the time of the evaluation or the verification. Now, actually, in the case of a mortgage, they're pre-evaluated once or twice before that and re-evaluated at funding. After funding, they don't get evaluated anymore. You have to kind of know when you're building up a collection of evidence what you're building it up for and when you plan to have it be verified.

Ideally, you want to be able to verify it forever, but in the context of the event that was important. You're never going to ask the question, would I give somebody this loan today, a year after they got their mortgage, but you might go back a year after they got their mortgage, and say, were they qualified for this loan when they got it, which is a different question.

As we wrestled with issues like this, and I used the mortgage example here just so that you wouldn't think I'm only thinking about telco, but again, it was telco that was forcing me to think about this. People were asking, well, how can I prove and then fill in the blank with a bunch of telco related questions and I was grappling with all of these issues. And this is where I kind of ended up. And I feel like I kind of came back to "Oh, KERI has a superpower that I wasn't giving enough credit to before."

First of all, we started calling this construct of a bunch of evidence that you've assembled a Dossier, for lack of a better word, seem to resonate with people. I could have called it a file, but that's so generic that I just picked dossier.

You notice that I'm using these icons here to represent the fact that it's a collection of different evidence types and you have some kind of a process that feeds all this evidence in and then you go through this curation process. You basically get the evidence to the point where you like it. This is like going to the loan officer and saying, "Okay, I've got all my documents in. Now you can evaluate and decide whether to give me a loan." Right?

Now you actually might have more than one of those points. You might evaluate it one time or seven times. Or you could evaluate it for a very long time to come. But you need to know what your plans are about that. The idea here is that this dossier is intended to be shown to somebody but not presented in the sense of a verifiable presentation.

You don't have to prove that you are the issuee of everything in the dossier. Let's say it's an investigative journalism dossier. you're not proving entitlements with this thing but you have to prove that all of these interviews that happened with certain sources really took place or whatever. It's not the simple model that you get with verifiable credentials where you're saying here's the thing and I'm proving it for myself on behalf of myself and I'm entitled to something.

Once you have a dossier, this is the other thing that I became very comfortable with in the telco context. The world's telco systems are built to move very small amounts of metadata. Let's take the case of SMS. When you send an SMS message, in theory, you're sending 140 characters or less.

Nowadays, phones will send stuff over RCS instead of SMS. You can exceed that limit. But the original S SNPP protocol wanted to move 140 character or less things. And now imagine that you were trying to move the equivalent of let's say a verifiable credential or even an ACDC in the headers of that.

You have headers that are like five times as big as the message itself. That's not very effective. What we can do though, is we can derive, and this is part of KERI's superpower, we can derive from a dossier. If it's really a verifiable data graph, that means it has a SAID.

Which means, I can just send the SAID in this really light form and somebody on the other side who understands that it's a SAID can say, "Oh, I bet there's a whole bunch of data behind that." And go fetch it and it works.

And we can make that a little bit more convenient by turning the set into a OOBI. Basically now I can take something and here I wrote 'token,' this is a word that I don't like a lot, I kind of roll my eyes when people talk about security and tokens. But in some contexts that's the way security is enforced. In the case of voice calls, for example, every VOIP system in the world carries a JWT with every .., well it can, it's built to carry a JWT with every SIP invite.

When you pull out your phone and you dial a phone number and you hit the dial button, your phone is causing a SIP invite, which is a small message to flow over the world, the internet. And typically it uses the real internet for parts of the transmission and then it flows over specialized telco circuits for other parts of the conversation. But anyway, somehow your SIP invite ends up on the other side and that's when the phone starts ringing is because the SIP invite arrived at the other location.

That SIP invite looks very much like an HTTP request and it has headers. The SIP protocol and the HTTP protocol are kissing cousins. They look so similar. They even to the point where 400 range errors in SIP are 400 range errors in HTTP and everything - Intentionally?

They very deliberately copied HTTP when they built SIP. All that you're doing really with that infrastructure, if you want to decorate your call with evidence, is you're putting a header .., and think of it like an HTTP header, It's almost identical. It says identity colon space and then a bunch of stuff that's B64 encoded. And that B64 encoding is that B64 encoding is the JWT that I'm talking about.

And that JWT can have a field in it, a claim that is the URL to your evidence. That's basically all we're doing. But this same technique can be applied well outside of telco. And that's why I'm interested in this. You can derive any number of tokens that keep referring back to the same piece of evidence. And

that's what we're doing over and over again when you make phone calls. If you're Microsoft and you're making outbound calls, every outbound call that you make carries a hyperlink back to the same evidence, but a new signature, right?

There's a new signature on the JWT, but the evidence behind it, the stable evidence doesn't change. What does a dossier actually look like? For those of you who are ACDC people, this is a trivially simplified version of an ACDC, I have JSON and in it I have a field called A and a field called E.

You know what those mean, attributes and edges. All that a dossier is really is a particular kind of ACDC or maybe I should say a flavor of ACDC where the A fields are used to carry curation metadata like who built and assembled this thing when was it assembled under what governance rules was it assembled what are the assumptions about how it's going to be used that kind of stuff but none of the credential content is in the A field.

All of the credentials are then in edges. You're just saying "I assembled .., I'm applying for a loan. Here's my mortgage qualification packet." That's all up here. And then every individual piece of evidence that I submitted is down in the edges.

The edges just point to another ACDC. That's right. The edges just point to another ACDC. And this is where we get verifiable data graphs. There's a lot of different kinds of evidence that you might want to put in a verifiable data graph. The obvious one is credentials. All of us think about credentials a lot. But in fact, the more I have worked on this, the more I think these other ones on the left side of this slide may be more compelling in some ways. I want to put A signed PDF in my evidence. I want to put an MP3 file in my evidence because I'm a journalist and I interviewed somebody. I want to put a movie in there because I captured some video. I want to put an AutoCAD drawing in there. I mean anything you can imagine we can embed In an ACDC simply by using the attachment mechanism and that's quite important. Some of these might be so big that we don't actually want to put them in an ACDC.

We just want to put a hash. Like a genome, your full DNA. I don't think you'd want to put a 40 gigabyte genome inside of an ACDC, but you could certainly put a hash of your 40 gigabyte genome inside the ACDC along with a URL of where to find it or something like that.

Anyway, these two techniques, embedding and wrapping are what I'm contrasting on this slide. The idea with embedding is you have data that's not credential oriented, but it's still data that's important. Scientific readings all these other examples I went through.

You want your assembly of evidence to reference it. Let's suppose you're an NTSB crash scene investigator and a plane crashed and now you're trying to take photographs of all these different parts of the plane and how the wreckage was spread out and all this other stuff.

you would be embedding JPEGs or hashes of JPEGs in your dossier when you're filing your accident report. Wrapping is relevant when the thing that you want to put in your verifiable data graph is another form of verifiable data, but it's what I would call foreign to the ACDC world. If you have an X.509 certificate and you want to put it in your verifiable data graph or if you have a W3C verifiable credential and you want to put it in your data graph, you can do that. The tech now you can do it by doing an embedding but I don't think that's a good idea and I'll explain why in a second. What I would recommend is this wrapping. What you're essentially doing is you're verifying the evidence with its native stack and then you're issuing a new ACDC that says on such and such a date I so and so verified this credential and it was legit on that date and you're making an explicit claim one way or the other about whether you take any responsibility for revocation status.

You can say, "I explicitly disclaim any responsibility for telling you .., in the ACDC land, I'm not going to reflect the revocation status. All I'm telling you is it was valid on this date." The end. Or you could say, "No, I'm a really super fancy service and this kind of evidence is really important and I will just [tell you], within one day of latency, I'll tell you if the other thing gets revoked, I'll revoke the ACDC wrapper." That's kind of up to the person, but the wrapping is the name of the technique that I'm talking about. And I could imagine doing this because it builds a very nice bridge. You can stick an ISO mobile driver's license in this. You can stick a verifiable credential W3C generation 1 or generation 2. You can stick in a nonredit presentation in that, whatever you want.

I anticipated that at this conference there might be some people here who know a lot more about verifiable presentations than I do and that they might be asking, well, isn't this the same thing as a verifiable presentation In VC land? And the answer is it does have some similarities. That top line of this table, both of these are cryptographically verifiable containers of evidence.

But there are some profound differences too. And these differences lead to a very different set of addressable use cases. Without belaboring the point too much, I'll simply say that the life cycle is really crucial here. A verifiable presentation is intended to be verified when it's generated and then it's thrown away.

It's ephemeral. A dossier is not right. You don't throw it away. Yes, you make a decision about whether to fund the loan for the mortgage on a particular date, but I guarantee every one of you that has a mortgage your loan officer has a file somewhere in their back room or whatever that has all the data that you ever submitted and they can look it up whenever they want because that's important data to keep on file.

And the same thing would be true if we're talking about investigative journalism. If we're talking about crash scene investigations or if we're talking in my case about the reputation of Microsoft and proof that it can use the world's telco infrastructure in a particular way, it needs to be on file and kept.

The other line that I want to just call out here is the bottom one. Really the meaning of these is a little bit different. In the case of a dossier, what you're saying as the person who assembles the dossier is here's a bunch of data that I put together that I'm signing over it, but I'm not necessarily asserting that everything in this data is something I believe. Like if I'm an investigative journalist, I could well have interviewed a bunch of people who lied to me on camera or while I was taking notes. And I want to document that they lied to me. And I'm not necessarily asserting that they told me the truth just because I signed over it.

I'm asserting when I sign that I was the collector of this body of evidence and collectively it represents some purpose that's described in the metadata about that dossier. Right? Maybe I'm assembling all of the evidence from somebody who lied to me because I want to take him to court later. It could be full of lies. Everything that I put in the dossier on purpose.

The again the metadata of the dossier tells you the purpose. Whereas a verifiable presentation is always about proving enough to get me the entitlement. I feel like I'm just talking and talking. Everybody's listening quietly.

I want to shake the room up and do something, but I don't know what to do. Anybody have any questions? Richard. - When we talk about a verifiable presentation, the protocol includes the ability to make claims across the different credentials ination.

Does the dossier I don't know if it's going to be a protocol. We haven't got to that part of the conversation, but does it include the ability to say this name on this one is supposed to reference name on that one? But it doesn't do that because of the dossier. It does that because that's a native feature of ACDCs. When you declare an edge in an ACDC, you can say this edge must be related to that edge in the following way. There's an operator that you can declare the dossier spec by the way to kind of prefigure your question about protocols.

A dossier is not really a protocol, but it is a data format. It's a specialized form of ACDC and there's enough kind of extra rules that we want to attach to a dossier I think that It needs a spec. We're writing a spec about it. And the spec right now introduces a couple of new operators to enrich what edges can say about one another and things like that. Sam, I saw your hand go up.

Oh, Phil. Two questions. This is one I think is an obvious yes, but can a dossier point to another dossier? For sure. And then I thought the use of an OOBI in the SIP header was an elegant way to AID itself. Have we considered doing that in edges either in the dossier or in the ACDCs?

I have not thought about that. But as soon as you said it, I was like, wow, why didn't I think of that? That's a really great idea. I love that. What was your idea first? Edges are actually a block. You can add as many properties in an edge as you want. It's just a matter of saying you're going to add an UI to an edge.

Okay. Cool. This is an example of the kind of thing that the dossier spec needs to say is here's a really useful way to make this more powerful. Since your voice is quiet, why don't you use the [mic]? Anyway, no, but he's only recording his thing.

For the camera. The gent speaking to his name. I got I got to talk. I'll be No, don't pick up my sound. Okay, pick me up. It wasn't that I said verifiable data graph 20 years ago or 2020 or whatever it was just came out of nowhere. It is comes out of the theory of automated reasoning. Automated reasoning you build verifiable data graphs with aggregation operators. The only difference is that you're trying to do often live control of something where you're saying I've got 50 sensors, I've got data process, I got a database, I got a map, I got all of this stuff and I have to make a live decision with the best available data to me at the time that I do it. And all the data is aggregated differently. You have a whole family of aggregation operators and you just have connections between all of your sensors sensors aggregate to aggregate and then and it just looks like this big complex graph and all I had to do was say "If I want to make a trust decision that's no different than any other automated reasoning." We should be able to map ACDCs directly to how you actually do the processing.

You shouldn't be building some artificial thing that is divorced from the actual way you make you make decisions. ACDCs model how you actually make automated decisions. And that resonates for me because of that earlier slide that I was showing about how to make a decision for a mortgage. You're actually .., this is the real way that the real world works, right? You're assembling a bunch of evidence and evaluating it and .., yeah.

Go ahead. – Very different approach to all of this. But from the legal standpoint, we're now starting to move to modeling where the fair information practices are actually going to be mandated. Your original definitions solved a number of problems because the the data collected is the minimal data necessary to fulfill the purpose.

it is a specific purpose that even though the information may be kept in the dossier, it isn't ending up in some giant data lake that then can be used for other purposes later on. And I think the other piece of this that's really elegant is that if you have, in government, social programs that require three or four agencies to interact right now they claim they need a data lake.

This allows you to decentralize that none of them is in effect the steward. They are just the steward of the particular piece of it that they need to integrate for the programmatic utilization. I love it. And this actually segways directly to what I was going to say next. Traditional verifiable credentials and in fact most ACDC use cases that we think about assume a single issuer, right? But one of the things we're introducing in the dossier spec is the idea of joint issuance.

And the idea here is not that you have joint issuers all using a single multi-IG even. You just have joint issuers. They're all individual parties representing themselves with different AIDs. But in the ACDC, in the dossier, you can say this dossier takes effect as soon as I have five of the following 18 people who've signed.

And it's just a completely asynchronous thing. And I put a picture of the signing of the Declaration of Independence up here because you had a bunch of people in a room. But actually, in some ways, that picture is bad because it suggests everybody got together right at the same time. In fact, what we're suggesting here is you don't have to do that.

You can have one party sign and then a week later in a different place under different circumstances you can have another party sign and another party and so forth. Sam, – I just want to clarify you're using sign as a euphemism for actually anchoring into the registry that is of stuff like that but it's the same thing. You're anchoring the same thing with the different AID, but everything else is the same. It's a joint issue because everybody issued the same facts.

Yes. In order to verify, we would need the verifier to do a little bit of work that it doesn't currently do, right? It's going to have to go look up signatures in five different places or something. Unless this bottom line an optional finalizer can say, "Look, I've already done all of that work. Here's a a thing in my KEL that just lists everybody that already signed and you can go verify it." That would be a nice thing. But I really like this model. I think that joint issuance is going to create a lot of useful flexibility for us. You can also have models, by the way, that aren't M ofN models. You can just say this is a petition and as soon as we get a thousand signatures on it, whatever.

There's no N in it, right? It's just this is the threshold is M. Anyway, there's some really neat things about that and I can see how there's certain legal behaviors that this facilitates very nicely. Now I wanted to say something else.

This is another way that some of the superpowers of ACDCs come into play in the telco use case that I had. I had a case where I wanted parties all along a chain to be able to verify but the amount of data that they needed to see varied.

Okay. A regulator needs to see more and an intermediary on a transmission chain needs to see less. And the beauty of it is I can have a dossier that is sparse for one audience and full for another and it's the same dossier and they can both refer to it the same way which means you have the option of assembling this dossier but having many different variations on the verifiable data graph that expose lots or little amounts of what's needed.

In telco when you have the rule about lawful intercept for example law enforcement agencies require that they be able to see certain things about Your communications. Anywhere where there's a jurisdiction like that I can solve that problem with a nice rich dossier view and anywhere where I can show less I will do it.

Are you using selective disclosure to mean the category of controlled disclosure or just the specifically selective disclosure as opposed to any of the other variations of disclosure? The original title of this slide When I wrote it a month ago, it said graduated disclosure.

And then I thought, oh, I could play around with this. Oh, I could stick it, I could take contractually protected disclosure. There's all these variations on it. I don't know which is the exact frame that you want to use in a particular context. The point though is that I can suppress data or I can show data using variations that all have the same root hash and I don't know what you would call that control this thing is called partial disclosure but essentially it is creating a nonbinary Merkle tree of the data so that the fully compact is essentially the root and then he's just tracing it out, but the hashes along the way are generated. You can do a proof of conclusion the same way you would a Merkle tree just by walking down and then expanding it at that point saying, "Oh, here's what it is." It's a non-binary hash graph that you can do some sort of booster.

If I had to use a different word, I'd call it contextual display. Okay. That's a good one. Okay. I'm basically done with my presentation at this point. This is this is where you can get more information. The spec is currently at version 0.6.

We'd love to have people contribute or if you don't feel like contributing because you're way too busy like me. I get it. But at least keep track of it. I'd love to have some people just show up and fix a typo so that I can at least put more names on the spec besides me.

TALK VII

Open Verifiable Communications

Daniel Hardman

KERI Conference April 2026 · talks · 41:39

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Daniel Hardman presents Open Verifiable Communications for telco, addressing scam calls and texts that push users to abandon phones. Through a GSMA Foundry-backed demo, Provenant shows how transit networks prove call attributes from credentials, satisfy regulators like Ofcom, and react in real time to revocation.

- Daniel, you will introduce yourself? Sure. I'm Daniel Hardman and I'm the CTO of company called Provenant. You've heard a couple of references to work that we're doing in telco. And I'm here to explain a little bit more about that.

All of us have been impacted by scamming in telco. Usually a day doesn't go by that I don't get some kind of a message. This morning it was somebody running for Congress in some state that I've never lived in telling me how important it was that I donate $3 to his campaign. But all of us get these kinds of things. And many of us who work in cyber security sensitive areas are pretty well aware of how scammy and fraudulent a lot of it is.

Even people who don't know anything about tech know that this is scammy and and full of fraud to the point where this is the answer they're coming up with. Just don't answer your phone, right? Or just stop sending texts.

That's actually a pretty bad indictment of a technical infrastructure when the best advice we can give is just stop using it. But that's kind of the state of affairs right now. And that constitutes an existential threat for an entire market vertical.

There are billion-dollar companies that make their money off of phone calls or make their money off of texting. If you have entire influential voices, segments in an industry that are saying just stop using that tech entirely, there's panic going on.

Microsoft is an interesting example of this and so I'm going to use them because I know some details about their world. One of the common scams is maybe you've had a grandparent who says, "Oh, I got a call from Microsoft tech support."

The reason that scam is dangerous is because Microsoft does have a tech support staff and they do sometimes call people. They don't spontaneously call people, which is the way the scam usually works. But if a person calls Microsoft tech support and they get disconnected, Or if they call and they have the end of the current conversation, but there needs to be follow-up, Microsoft does call people.

The problem is people don't want to answer. And that's a serious pain point for Microsoft. And regulators are getting involved in this now because Microsoft has another characteristic that's kind of complicated that would be true of anybody who's a giant global conglomerate, which is they put their people in the most strategic places they can for cost reasons and so forth.

Particular case in point in the UK Microsoft has plenty of customers that they interact with but Microsoft doesn't put their call center staff in the UK. They put their call center staff in Bulgaria or some other lowcost center that can make a cheap call into the UK. But Ofcom, which is the equivalent of the FCC in

the UK has some problems with that because they don't want a phone call to be claiming that it's coming from the UK and from Microsoft when it's coming from Bulgaria.

They have said, you may not attribute this phone call to a UK number unless you can prove it. You have to prove that it's really coming from Microsoft and this is creating a lot of pain for Microsoft. What Microsoft wants is a transformation of the customer experience from the phone on the left to the phone on the right. They'd like it to be the case that when the phone rings, it shows the Microsoft logo. It shows the Microsoft name and it shows why they're calling. Oh, they're calling because they're customer support representative and they want to be able to prove it.

It's not hard to make a phone show logos. What's hard is to be able to prove why you're showing the logo. And this is the requirement that Microsoft has. What I'm here to talk about what I'm here to talk about is .., I'm just going to let this narrator take over for a minute and we'll watch this demo. This was presented at Mobile World Congress about a month ago, […] a three person consultancy who wish to make outbound calls to their customers.

We then move to a larger organization, Microsoft. Their outbound contact centers need to be able to make calls to customers, but increasingly find that their calls are impacted by incountry regulation, get spam marked or simply not answered because customers do not know it's Microsoft calling.

What you will see looks like a branded call. However, it is a little bit different. With OVC, the information displayed on the handset is based on a set of credentials which prove certain things about the origin of the call.

They prove that Microsoft has the right to the telephone number that is being displayed, that it is Microsoft Corporation that is calling you, linked back to the Axel legal entity responsible for the call using a globally unique legal identifier, and that they have the right to use the brand that is being displayed to you. These credentials have all been issued to Microsoft and are in their control.

Microsoft could use these three credentials to originate calls themselves, but there is a fourth credential which allows Microsoft to delegate the right to sign on their behalf to another platform. For the project, we've worked with Infobip, Twilio, and Vonage as platforms that could be used to sign traffic on behalf of an enterprise and they do not wish to make their own calls.

The call you see will leave the platform onto a network where it will be verified by a transit operator before it has arrived at the handset and the outcome of the verification is displayed. Let's go and make a call. For this example, we're going to make a call from Infobip.

We are logged in as Microsoft and we'll send a call from this US telephone number which has been allocated to them. We will call this Spanish number which is a prepaid SIM we bought in Spain for the demonstration and is in the handset which is mirrored to the right. Let's go ahead and make a call.

You can see here the name, telephone number and brand along with a verified tick are displayed. These are displayed based on a successful verification of the evidence provided in the communication. What we'll do next is have a look at the governance model that underpins this evidence.

One of the biggest challenges with global verifiable communications is ensuring there is thorough and consistent vetting of original parties that can work in many different circumstances. OVC has a rich governance model and provable chain of evidence to support this. It starts with a local governance authority that sets the vetting standards for OVC.

We next have policy administrators who appointed by the government's authority to apply these vetting standards in a given jurisdiction. It's important as there are many differences of regulations and vetting processes in different countries. They're responsible for certifying vetting agents to be able to vet in these jurisdictions for different communications channels.

Once credentials been issued to vetting agents, they're able to vet enterprises in specific regions and for specific communications channels. In the trial, we've had brand numerical and [inaudible] enterprises. The vetting agents verify enterprises to prove rights to telephone numbers, that they are the legal entity they say they are, and that they have the rights to use the brand, and then issue the enterprises with appropriate credentials. This process can support support both small businesses and large corporations.

It's important to note that these credentials are chained all the way back to the governance authority providing a provable chain of trust. It is also means that it's possible for credentials to be revoked at any point in the chain.

If for example a vetter does not follow the rules and vets bad actors, it's possible for a policy administrator to revoke their credential and have a cascade impact on the credential set issue. The credentials are held in wallets by enterprises to whom they have been issued. While the focus of the demonstration has been on voice, these credentials can also prove other communication channels such as RCS, SMS or APIs. The final credential which can be created if needed is a delegated signed credential which we discussed earlier.

What we'll look at now is how the evidence looks for a real call. First, we're going to make another call, but this time from another one. One of the credentials has been removed by Microsoft. We will call the same number.

As you can see, in this instance, the call fails. This is because the transit operator verifying the network attempted to verify the call and found that one of the credentials had been revoked and therefore rejected the call.

We will now go and look at the evidence detail for this call. We can see the call here. Evidence detail records refer back to the evidence and provide a permanent record of the evidence accompanying the call. We can see the three credentials here.

The identity credential, the TN credential, and the brand credential. And we can see here that the TN credential has been revoked. We can also look into the evidence in more detail and see who has vetted Microsoft. In this case, it was Aegis and even how they gained their authority to be a vetter. We can also then see the policy administrator and the government's authority for the trial.

These were just placeholders. We'll now move on to look at the actual call flow for the call that you've seen. We started with a set of credentials that have been issued to the enterprise and delegated to an originator so they can make calls on their behalf.

When the call is made, the originator calls out to a signing service from Provenant, the technology provider for this project. The service returns a compact passport, a STIR passport and also replicates this passport from out-of-band cache. The passport is not a shaken passport but a verifiable voice protocol passport.

Once signed, the originating party sends the phone onto the PSTN. Here it is received by transit operator Wavecrest, it takes the passport and uses a verification service to verify the pool. Verification can happen anywhere in the call flow. The ideal scenario is that verification takes place in the terminating network. In the case of the good call, Wavecrest will pass this call to the terminating operator. And in the case of a bad call, Wavecrest rejects the call.

When the call arrives on the handset, it does a look up to an out-of-bound passport cache. If a passport is found, a verification again takes place and if successful, the result is returned to the handset. The outcome of the verification is then displayed as the phone rings.

In our trial, we relied on verification on the handset using a modified version of the standard Android dialer provided by Google for this project and a lookup was to a public out-of-band service. In an ideal scenario, the verification would take place in the terminating operator's network with the cryptographic evidence being passed directly to the handset or to the out-of-band service to be deployed in the operator's network.

This concludes the demonstration. This project has been supported by the GSMA Foundry and has had participation from the following companies. If you would like to find out more, please follow the QR code. Okay. There was a lot of detail in that and I'm going to try to pull out of what you saw a few points here.

The first one in the view that you saw that was like this: anybody who has seen that telephone call transit their network has the ability to prove certain things about it. This will satisfy a regulator. It will satisfy an IT team that's worried about regulatory compliance.

It will satisfy an end user and all of those stakeholders matter. Just making the end user comfortable isn't the only goal here. We had to make Ofcom comfortable. Be being able to drill in and say for this call, what are the credentials that were available and what can I know about the party that made the call at that time and I need to be able to have real time reaction to revocation.

If Microsoft is doing something funny and a regulator says don't let that kind of call through anymore and we revoke one of the credentials the system has to react in real time to that. That's one of the things I wanted to call out.

Okay. Being able to drill in deeper so it's not just a badge but you can you drill into the individual pieces of evidence and look at their status and you can do this historically too. At the time the call is made you can look at it but you can also go back a week or a year later on a particular call and say what was the state of the credentials at this time? What was a justified conclusion based on what the evidence looked like then.

Now Sam's talked about tipping point. I really hope that the telco thing is a tipping point. It ain't over till it's over but we have made a lot of progress here. I want to just highlight a few of the things that were kind of touched on briefly. VVP is an acronym. It stands for Verifiable Voice Protocol. That's an IETF draft spec right now. For those of you who don't work in telco, I've gone through this learning curve recently. STIR is a technology that is used in VoIP.

Voice over IP and basically a STIR passport is a JWT. So many of you know what a JWT is and how it works. A JWT carries a small amount of information and the world's VoIP systems already know how to carry these JWTs. What VVP does is it says we're going to put a little bit of extra information in that JWT. And I'm going to show you how little I really mean when I say a little bit. It's actually remarkable how rich we can make this with a very tiny addition. And because the world systems already know how to carry

this information, we don't have to modify the infrastructure. The narrator in that video mentioned just in passing that we were calling a Spanish SIM card that we had bought off the street, literally we flew into Spain for this conference, which was a month ago in Barcelona. We flew in, we walked into a shop on the street, a random one, and we bought a SIM card and stuck it in, and we got a call to work. And that's important because we're not talking about proposing to the world that they go deploy a whole bunch of new infrastructure.

The networks that carried this call with this information had no notice that we were doing this. We just did it and it worked. That's really important to the adoption story. The events that are listed here are kind of in chronological order.

The GSMA is the world's… basically the arbiter of mobile technology in the world. This is important that they convened this project and those names that are listed there are all big names. They all participated in one degree or another.

Google was the handset manufacturer who got their dialer modified fastest and that's why we used the Google handset to do the demos. But Apple We had good conversations with them. Samsung was there, Cisco was there. All of these companies participating together and Microsoft is going to production in right now as we speak in various parts of the world. It's not a single big bang. It's a series of going to production things in different jurisdictions and with different kinds of evidence. In the UK, they don't need the branded calling right now. They just need the name Microsoft to appear. They don't care about the logos. They're going live in the UK first because they have to satisfy Ofcom in the UK. Then they're going to other places.

Anyway, there's some real live in production stuff by companies that you know that are doing this and the GSMA completed this OVC project in March, just a month ago. Now they've announced that they're going to have some follow-on projects. One of them is going to focus on governance.

And this gets to the whole LEI and vLEI topic. During the proof of concept stuff that we did, we were using an LEI to identify the companies and everybody agreed that was vital. Now they're looking to extend that and fit into the vLEI family.

That's part of the follow-on governance work. The other thing is that they explicitly want to make this technology usable in multiple channels. The channel that we focused on in this first experience was voice calls. But they care very much since the world's mobile technology providers. They care about texting just as much as they care about phone calls. They care about rich chat and how iMessage works and bunch of things like that. When I say multi-channel, that's what I'm talking about. Now, I never explained the acronym OVC. Let me just spend a minute. OVC originally meant Open Verifiable Calling.

But during this foundry project, people said, "Wait a minute. There's no reason why we couldn't take this evidence that we've built into these other channels." Right? OVC has now become in the minds and in the communications that go on in GSMA channels like their mailing list and the meetings that they have. They've started calling it Open Verifiable Communication and that's what OVC now means with this check mark. They're explicitly pursuing communication as a general goal, not just calling. I have a little bit more information about that in a second.

Sam made a point in his presentation that we don't want to have a single vendor solution. Provenant was kind of the enabler of this whole thing. And of course Provenant is a for-profit company and we hope we can sell lots of people on what we've built. But it's important for it to be open and we believe in

that.

Here's some kind of proof points about how open verifiable calling and the verifiable voice protocol work and thank you to this is a shout out to the Trust over IP folks who've let us build a Dossier specification at trustover IP and I'm going to talk about what a Dossier is in just a minute. It's it's important to this whole concept and Also to the folks over at IETF that we're working with.

Okay. Let me talk a little bit since this is a more geeky audience. It's not just pure business people. I want to talk about what's going on underneath here. When we looked at what we had to build, these are the kinds of requirements that we felt we were getting from the business side and the regulatory side. And I'm not going to read all of them out loud, but I hope when you look at that you say, "That's a pretty ambitious list because it felt kind of overwhelming to us." Building half of those would be pretty easy, but when you start having to build all of those things in or address all those requirements, as Sam said, you get into a very hard trade space where you're making trade-offs and some of them are hard. And we didn't want to trade away any of these requirements. And because we're here at a KERI conference, I will just say I don't believe we could have built this with any other technology. There were a number of reasons why you end up at a dead end if you try to solve it any other way.

Without belaboring that, let me say also it was important to identify some things we did not have to do. And this is important because there are calling solutions that deliver significant value that have some of these requirements and we were not asked to address some of these. The most interesting one I'll just mention the bottom one here.

Somebody said, "Oh, well, this is all well and good, but how do I know that an AI is not sending me words and pretending that it's coming from another person?" And the answer is we're not trying to solve that problem. The reason we're not trying to solve that problem is because for us, the issue is about accountability.

If Microsoft connects to you, then it's on Microsoft if they once they're connected with you you're really dealing with Microsoft who made the call. If Microsoft then sticks an AI on the call, that's their problem and your problem, but it's not the problem of the technology that connected you and proved that Microsoft was accountable in the first place. It's about accountability.

Not about whether it's human or not. However, we could certainly prove that the party who dialed you was an actual human being versus an AI if we needed to. And I'll show you where that fits if that became interesting. These are some of the things that were the places where maybe the trade-offs manifest and it would be hard to address some of these requirements.

The top one is the maybe the biggest. Okay, how do you do all of this but you do it with a commodity SIM card over networks that haven't been modified. If you can do that, that's a big deal. Anyway, there another one, the third one there, I just want to mention, it's possible to prove things between a caller and a callee using classic trust triangle that we've talked about for years using verifiable credentials.

However, In that kind of a model, the parties have to know who they're talking to in a sense before you can prove because basically you have to have a connection with the other party in some kind of an out-of-band way before you can then prove with your credentials. And the specific characteristic we needed for this solution was that a stranger could call you. You haven't had any prior knowledge about who these parties might be that are reaching out to you.

Okay. What does this look like? If I gave you a really high-level architecture diagram, it might look like this. You have a party like Acme Corporation that wants to make a call. Sometime near when the call originates like right at the same time or within milliseconds afterwards if they're putting traffic on a network and they're were you using like a cloud provider or whatever as the call originates, a signing event occurs the signing that we're talking about is a signing of the JWT.

And the JWT is this thing that's called a passport That term passport is a term of art in the IETF specification for these JWTs. This passport contains .., and the blue with the underscore here is supposed to suggest this to you, contains a hyperlink to what we're calling a dossier.

The dossier is the collection of rich evidence. All we're adding to this passport that it didn't have before is a field that has a hyperlink in it. The relationship between the signer of the passport and the passport itself is crucial and we could talk about that offline. I probably won't go into it in my talk that much. Suffice it to say that you have to prove that the party signing has some relationship of delegated authority with the party that is originating the call. It could be the one in the same party, but it doesn't have to be. And that's important because you have to support things like call centers That are other businesses. The call center is not Microsoft and a regulator needs to be able to tell those two parties apart. But you have to be able to prove that the call center is acting on Microsoft's behalf. They can't just assert it.

There's a whole story about how evidence is built and I'm not going to go into it in detail, but essentially you start out with a cryptographic identifier for the company and a control mechanism. Those two things go together. And once you have those two, then you build an identity credential. And for those of you who know how vLEIs work, this is straight out of Karla's thinking.

It's exactly how vLEIs are built. Basically steps one through four are the journey of getting a vLEI. And then you add to that a bunch of other credentials. Importantly, the credentials are issued by different parties. You have a collection of issuers.

They all have their own revocation criteria and their own revocation SLAs. We can't put all of this in a single credential. And then you bundle all the credentials together in a Dossier. And the Dossier includes a last thing which has to be a credential issued by the actual company that we're talking about declaring their intentions to make this kind of traffic flow.

And then you can use it in ongoing use. Now some of you may be wondering, here we are at a KERI conference, can this technique be used without ACDCs without KERI. No, it can't. You can't build this whole solution but you can interoperate with other kinds of credentials quite nicely. If a company has a credential and the format of this credential is a format that's required by the EU to prove something about the company and they want to use that to prove one of the logical propositions in the dossier, you can do that. You could accept a credential that is a W3C verifiable credential. You can accept an SD jot. You can accept a signed PDF if you want as evidence, but you have to wrap it somehow because unless you can put it in a verifiable data graph and you can reason about it in a particular way, the solution falls apart. You can't end up generating that view that I was showing you in the evidence that has the badges and stuff like that. There's two basic ways that you can deal with that. You can either use wrapped mode, which is where I take a foreign piece of evidence and put an ACDC around it, and literally the ACDC has the content of the other piece of evidence right inside it.

And it's a reissuance essentially. You have an issuer of an ACDC that's saying, "I looked at this other piece of evidence and I used its native technology, whatever it is, W3C verifiable credential stack or something, and I verified it on date X."

Now I'm issuing an ACDC that says on date X, this credential was verified and it was true. That's wrapped mode. And you can see that wrapped mode has some good things about it, and some bad things about it. The good things are that a party on the other end doesn't have to know about any other stacks or technology types. It only has to verify ACDCs.

But you have a revocation issue. Does the party who's doing the reissuance sign up to also track the revocation? To be determined, right? The other one is you can have embedded mode and in that mode the party that's creating the ACDC doesn't take a lot of responsibility. They basically say here's an ACDC envelope around data but what it means is up to you to determine and so you have to have a stack that can look and understand this other evidence format and decide whether it agrees with the guarantees that it provides. I don't think that embedded mode is a good idea because it fragments the verification space.

But anyway, enough about that. I won't go into great detail on this diagram. I could talk about it for a very long time, but I'll simply say this. The way that passports in STIR were conceived of is that a passport was supposed to include an X5U field that was a hyperlink to the certificate of the signer of the passport.

Okay. And when you went and looked up that signer certificate, you would see that the certificate included proof of a cryptographic identity, that would be their public key and it included information about the attributes of this party. That'd be like the common name of the company and their organizational unit or you know all that other stuff. What we've done is we've split that apart and said information about identity and information about attributes need to be handled differently because identity can be permanent.

On this model. How often do you have to replace this? All the stinking time? I don't know exactly what number you're going to come up with, but I guarantee it's a number I don't like. Right? If it's once a month or once every three months or once a week, SHAKEN in the United States mandated by the FCC calls for three-month rotations in their guidance. I think they enforce a higher number than that, but okay. Anyway they're only doing it right now. The certificates that SHAKEN use are only for big companies like Twilio and whatever.

They're not for the millions of little companies around the world. They said, "Let's take this thing and we'll start using certificates on all the little companies around the world." Can you imagine if every little company around the world has to replace proof of who they are? Let's say there's 30 million businesses in the United States alone. You're going to be maintaining a fabric of 30 million certificates that rotate once a month.

It's completely impractical. Here, how often do we rotate this? Zero times. I really love that answer. You can rotate the keys as often as you want, but I don't have to replace the identifier ever. And I can't tell you how profound the difference in cost is and the difference in infrastructure lift when you don't have to replace anything. It's huge.

Now, we do have to change attributes, but we only have to change them as often as the attributes change, right? If your company doesn't rename itself, you don't need a new proof of what its name is. And if it doesn't lose the ability to use a logo in a particular jurisdiction, you don't need to replace that either. Again, the maintenance is dramatically different.

This is how it actually looks on the wire today. On the left is a standard SHAKEN passport. This is what actually flows over the wire right now. And the reason that we could deploy this with no changes in infrastructure is because this is how different it looks over here.

None of the infrastructure cares. I'll highlight these two lines. This is the hyperlink in a traditional SHAKEN passport that says here's how I can find the certificate of the party that signed this. And here we have a link. This is an OOBI to a KEL.

Okay. And down here we have an unsupported assertion from somebody saying, I think that this is an A-level call. I believe in this call. Here we have hyperlink to evidence. It's really simple. I'm not going to… I'm about out of time.

I'll just say that Dossier can prove lots and lots of logical propositions, not just one. And the list of things that you can prove under the previous model had one bullet on it. You can prove the opinion of the company that signed the JWT about whether it was a good call.

That's not a very long list. I mentioned this that OVC is coming to mean something different. There's active work now going on on a variation of this for SMS, a variation of it for rich chat. Proving who sends you an iMessage or the equivalent in the message app on Android and web meetings. Wouldn't you like it if when you showed up to a Zoom meeting you actually had proof of who you were and everybody else did too? And there are some other transports that I won't announce today, but there's cool stuff going on with those two.

If you want more information, the spec for the Verifiable Voice Protocol is at this one and the demo that I showed you is at that QR code. You can watch it again. Any questions? – We got time for two questions. – Yeah. You mentioned that you were using the Google dialer. Was that [inaudible] – Google's dialer was actually signed by Google and so forth, but it it's not in the mainstream distribution.

Google is still debating whether they want to put it in. Another question – Question about the security of the URL that goes to the Dossier. Obviously if it's wrapped and signed then the URL is safe. The URL itself is safe but at the target that's where something can change and be malleable. How are… what are you doing to secure the endpoint where the URL goes?

– The URL is an OOBI. It's not a generic URL, meaning that when you go there, you must see a signed sequence of events that proves the its evolution from its origin moment, its inception moment. You could have somebody who tries to replace the content at that location, but if they did, they would have to come up with content that still evolved from scratch and was still related to the identifier in question, and that would be detected then by the watcher network and so forth. KERI prevents a bunch of monkeying with that. And interestingly enough, this is one of the debates that I've been having in the STIR working group at IETF because people said, "Oh, this is a great idea. We'll put a hyperlink in there. We'll just accept any hyperlink."

And I said, "No, that's not quite a good idea." For the very concern that you're talking about, we'll see if I can convince them. They might just modify the spec and say any hyperlink will do. In which case… – …that immediately becomes the sensitive attack surface.

– Exactly. – One more question. Could you grab the mic? – In your architecture, you showed a couple of phone home paths. What are the implications of privacy on that? – The phone home path is not a phone home to another party. It's a phone home to the originating party.

Basically the party that made the call might have somebody call them back, Or it call their watchers and witnesses back. But these are entities that they control and have chosen. It's not calling home or phoning home to a central registry. It's phoning home to… it would be the equivalent of on the web calling the name server for Microsoft.com that Microsoft chose. You're tracing data back to a server Microsoft set up and empowered.

– It doesn't have to be Provenant in every case. Oh no, no. Provenant implemented this system. But the spec is open and anybody could implement the system. There's no lock-in about that. – Okay, thank you very much once again. [applause]

TALK VIII

Daniel-Hardman-Interview

KERI Conference April 2026 · interviews · 01:16

My name is Daniel Hardman and I'm the CTO of Provenant, which is a small startup in the telco industry. My presentation was on open verifiable calling. That acronym OVC is changing into open verifiable communication and it's all about proving who is making some kind of a communication flow to somebody else. The big thing that we're doing that I think is crucial is we're providing evidence for the first time about who is involved in communication. Up until now, most efforts to eliminate fraud have basically proved somebody's opinion, but now we're actually able to prove hard facts about the parties involved. I think without the OVC initiative, the entire telco industry is missing an important insight and an opportunity. Fraud is driving down the usefulness of all of our communication technologies and unless they come up with an answer, people are going to stop answering their phones. They're going to stop texting and we need to have strong evidence so that these technologies remain viable and deliver real value.!

TALK IX

Mobile Wallet

Evan Asakawa

KERI Conference April 2026 · talks · 26:16

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Evan Asakawa describes a browser-native KERI wallet running KERIpy via Pyodide/WebAssembly with local keys and persistent storage. He recounts porting cryptography, replacing LMDB, and demoing witness/watcher plugins connecting to KERI Foundation infrastructure from the browser.

We have been developing a web wallet for KERI essentially just as you know more wallets more platforms better. Initially Sam [Smith] set out these goals where we wanted to create a fully web native wallet because currently the Locksmith application is a desktop application.

Core wallet, we wanted it to run entirely in the browser with no like server or anything running KERIpy on the back end. Which, proved to be a challenge in fact. Again, we wanted everything to happen in the browser, so you can have all of your keys locally created and stored on your device. Don't have to worry about them being stored on some server. Then we could just send you this application and you would get your in-browser wallet and we also wanted it to be persistent because that would be a little bit useless if it wasn't.

Some challenges: Running Python in a WebAssembly was the first big hurdle. because currently Python is run on CPython which is a C-written interpreter. The cryptography had to work in the browser and this basically meant that KERIpy couldn't drag LMDB into the web path because LMDB relies on MMAP which is not supported in WebAssembly.

And, again, we want this to be not a one tab, one reload type of wallet. We want it to actually be like a production grade wallet that we can deploy where you can just take the files, serve them yourself, even if you wanted to, and be able to have your own identifiers that you can manage from a browser and not have to download an application on specific architectures. Getting Python into the browser was a interesting challenge. We found Piodide which is essentially the Python interpreter compiled to WebAssembly which allows it to run in the browser which means that you can then run any Python native packages and for all the non-Python native packages that was a medium challenge And then we have PyScript as like the browser bridge so you can launch off web workers in the browser and it interfaces with like the browser database so you can have again that persistent storage which is exactly what I put on my slides.

To do this we had to again go through the dependencies of KERIpy get them to work in the browser which started with pysodium which is a python wrapper for Libsodium which is a C library. I had to take Libsodium compile it to target WebAssembly and then I changed some of the code in Pysodium so that it loaded the packaged Libsodium that I created, called it pychloride because it's like you know sodiumchloride… that was Sam's idea.

I'll give credit where credit due. Blake3 is written in Rust. I had to compile this WebAssembly and also in an attempt to get ready for future proofing we took this pyoqs which is an open quantum secure safety library for key creation and made it WebAssembly compatible. That will be coming in the future hopefully before quantum computers.

In KERIpy itself we had to make it WebAssembly safe, which meant creating a second database backing which used indexedDB which runs in the browser. We created 'webdber' instead of 'lmdber' which means we're no longer relying on the MMAP LMDB And we also had to make it so that the imports don't randomly import lmdb so that it works in the browser. And again, this webdber allowed us to create like a persistent storage in the browser that can survive when you press reload on accident and then you lose all your keys because that would suck.

– Can I ask you can I ask you a question? Is that from a technical perspective? Is there an abstract class for that or is that not implemented with an abstract? – We created a base spacer Or wait not for basers. No, for the database we made a second class, just for this one.

– Okay. It's like what is it called, duck typing or something? – We duck typed it all the same functions just with indexedDB. – I mean maybe we can come back to this but do you mind if I ask a question? Go ahead. – I'm not familiar with KERIpy so I understand none of the words on Oh, okay.

– Except for like just from like a more of a web-centric mentality like how is it storing data? – We're using PyScript's API, actually which is one of the reasons why we chose to use it. They have a storage API that exposes a couple useful functions that we can then use to store our data persistently.

– What is WebDBer? WebDBer is the ducktyped class for LMDer and KERIpy. KERIpy is the core KERI repository that implements the protocol. – We can talk more after. This is like basically the nitty-gritty of… – This is some of the KERIpy internals of the library itself. – WebDBer is a KERIpy concept.

– Is HIO. – HIO is a separate library. It's an asynchronous library that [inaudible] Sam had to extend HIO actually for this to be possible because the browser and JavaScript uses asynchronous calls and HIO did not before and so Sam had to add an asynchronous interface into HIO we can then use to persist our database.

– Are all these changes now today committed into KERIpy? – It's on a branch. This is how it works. We have the UI. It's just like a very basic JavaScript shell which launches our PyScript web workers that then call into KERIpy and HIO which allow us to run Python in the browser and those call out to indexedDB which stores it persistently.

Keys are created in browser… survives. No server side key management. Don't worry guys, here comes the demo. – Before you get the demo guys… Shoot. – What you guys do about Keeper? How are you encrypting keys? Made a web keeper that interfaces with… [inaudible] the encryption before putting it in indexedDB.

I think so. I could look at the code. I can tell you what I did. I don't know. I did a lot of this like a couple months ago. – If I wanted to make my own database implementation, I would make a ducktyped class for another data store.

– Correct. And then you'd have to create like a baser for that class and a keeper for that class. – Really? Probably. Anyway, here's it running in the web. I just copied Locksmith's UI because that felt intuitive. Again, this is just being hosted locally. Eventually, I think we're going to want to deploy like a the KERI Foundation will want to deploy like a, basic rate-limited version for the features that require witness and watcher access. I have some vaults.

See, I was testing some vaults out. All right, let's create a vault. Open the vault. We can create local identifiers. I'll make one for Evan. Oh, those are emojis.

And I was trying to get this working last night. Those might work. I created the plug-in page. Let's see if it actually works. It's spelled incorrectly. That's okay. The KERI Foundation plugin. In Locksmith which Phil demoed yesterday which they then donated to open source.

The idea is that because it's open source anyone who wants to provide witness and watcher infrastructure can create like a plugin and then you can connect to these different providers plugins and get oh it worked nice – It works you can get witnesses in the browser. – That plugin is pointed to some external infrastructure or… – It's pointed to the witness and watcher service that healthKERI also so graciously donated that is running locally on my machine right now.

– The same thing he demoed yesterday. – He demoed this yesterday just spinning up your own local infrastructure. The KERI Foundation is going to provide I think like a basic API which will allow you to interface with our plug-in and then you can grab our deployed witnesses and watchers on our service. That sound right?

– Close enough. KERI Foundation will have a plugin. Other providers will have plugins as well that they can provide. We're going to host all of our services or we have hosted our services on Digital Ocean. I'm doing this all running locally because it was faster for testing.

It's working in the browser. The next thing (I have a lot to show guys), I've also been working on trying to get Locksmith able to have multiple plugins so that we can show that in this open source ecosystem, anyone can come in and provide their own witnesses and watchers as a service.

Here's the KERI Foundation. – This one's desktop? This is desktop app. I'll add my Evan identifier. That's not my name. With an emoji. No. What? No emojis here, guys. I'm sorry. This is Locksmith, the open sourced version.

Okay. On the desktop. On desktop. This is the desktop app. – We just doing this for comparison with the web one. The web one is not as far along. I have plugins working on the desktop one, to demonstrate that you can have multiple providers giving you witnesses and watchers. We have our KERI Foundation plugin.

We create like our account identifier. I think Ryan [Hansen] showed this yesterday, it's basically just talking to .., – It works a lot faster when it's local. [Yes, it] Does work a lot faster when it's local. And then I have a second plugin here.

Actually, that wasn't anything impressive. I can add one of my local identifiers in the KERI Foundation plugin. Then I can grab witnesses from the KERI Foundation infrastructure which is running locally. And I have to provision and register them.

It's like kind of similar to healthKERI because they left some shared components. I used a lot of them. And then I can grab this. Rotate them in. All right. Selected to rotate in the witnesses. Okay. Now my local identifier has one witness from KERI Foundation infrastructure, ideally once it's deployed. And then say you as an individual or corporation want to also provide witness infrastructure. You would create a plugin for Locksmith.

I copied and pasted the folders so they look very suspiciously similar. – Can I ask a question here? Yep. – Creating a plugin, do I have control over the UI? Yeah. – If I want to have some enterprise -whatever-specific thing or any service I want, I can integrate that as a plugin?

Totally. And you probably want to again because this is fully open source to get it like contributed to the… you could either contribute it like upstream to like the one in the KERI Foundation repo or you could just clone it and deploy it as your own application.

– Okay. As an example, when we first did the plug-in concept, we were thinking just SaaS providers would put a plug-in here so that you'd get watchers and witnesses from anyone. But then we found usecases for the enterprise credential issuance platform that we're building.

The user interface is a plug into Locksmith and it just made sense because if you're building any kind of user interface that could use KERI, it's a great place to put it. As long as you're like writing Python and PythonScript.

Or writing Claude [laughter] – As long as you can. I don't like it. – All right. I'm creating my test infrastructure account. I'm not going to scan this one. And then I can go to my identifiers. I can grab the same local Evan identifier. You can see and it has the witnesses. It doesn't have all the the fancy schmancy tooling, that healthKERI showed, quite yet.

Now I have four witness servers running for this one. I can add four witnesses here. Okay. And then I can rotate in these witnesses. I can also rotate the other witness, but I'm not going to do that. My authenticator has two seconds left.

I'm using Google authenticator on my phone, so no onePassword copying like Phil was showing yesterday for me. I missed a digit. All right, there you go. They're in. Now I have again it took it sequence number is two because I rotated in two sets of witnesses from two different providers and there are five total witnesses witnessing my local identifier. It works. [applause] Sweet. And then, I've been trying to make it easy for anybody who wants to create a plugin, to create their own plugin. I was thinking, I've had many different approaches as to how I best thought this possible. My newest approach was to create just like a templating folder basically that would kind of just give you like a one UI page that you could use as a template so that it would Claude could actually be smart when he does it.

– Sam introduced the concept of what was it called? Observer registrar. I still need to figure out what all that is. Similarly here the plugin you could I suppose manage those kinds of things as well? – Eventually, yes. I think the observer and the registrar stuff.

– Are they just extensions of witnesses and watchers? I guess I don't know… – They monitor a different thing. – Those monitor your credentials. – But if I wanted to make a plugin for that, I could mention… – Probably. – I don't we haven't gone down that road yet.

– Wait, sorry, what was the question? If the plugin could be used for helping implement the registrars and observers. – Absolutely. It would just be the user interface into those. If you stand up a SaaS platform that has APIs to allow you to stand up registrars and observers that absolutely you could add it to your interface here in your plugin to reserve those for you.

– Which is similar to the witness and watchers you have to services out there. – Exactly. – It's just UI to connect manage the relationship. – That's exactly right. – Yep. Sweet. – Okay that was most of the things I wanted to show. I mean this is cool because it shows the how we are, KERI Foundation. We're open source accepting multiple providers if possible.

– What's the status of the mobile because we did web and desktop… – The mobile is, Jay do you want to take this one? – I'm sorry? – How's the status of the mobile? Jay has been working on mobile – It will be deployed sometime in the near future as soon as it gets the account info from Sam for the iOS and

Android stores.

– It's the same code though that runs the web wallet. – It's the same code. – It's just being deployed. – It the iOS and Android code are just in wrappers that run almost like a pseudo browser and we just deploy pretty much the same code onto it.

– It's the web essentially version. – And, it's pretty much it's just the web app. – It's doing all the same stuff you said, the Wasm and all the WebDBers mobile. – Yes. – Is there a mailbox as part of this? – No, not yet.

Probably. – On either the phone or or a browser page, right? Let's say I'm an issuer, issue you a credential offer or friend. And your phone is off or your browser is not on at the moment. Will there be a mailbox in between to pick it up?

– I would assume it would go to your witnesses or something. – [Phil Feairheller:] That's a common misconception. When we originally deployed witnesses, we did make mailboxes a part of the witness deployment simply because we were moving very fast and had nowhere else to hang them. Okay, we didn't want to create a whole new service at the time. We have since moved away from that. HealthKERI does provide mailboxes as a bespoke thing you can reserve and I was going to demo that yesterday but ran out of time and then and then you just launch a Locksmith turn it on say activate this mailbox and now you can get messages on it and if you miss any messages they're stored forward so they sit here.

– Are those on your platform like you're storing? – Those are platform. Then … – It's a very simple small service. It's it's basically if you were to look at the code inside of KERIpy you just pull out the mailbox stuff. Because there is mailbox up in the right – That could be hosted anywhere.

– Absolutely. It's just like any other resource. – Yep. Again, we got this code recently from HealthKERI. – Awesome work. There's a long list of features we're aiming to add. Just hit the ones that were priorities for the conference first.

Again, you can always go into the open source repos, create issues for features you want to see, and we can address those as they come in as well. – Or fork and build. Yeah. - Absolutely. – [Karla Mckenna:] did I understand you correctly that you said five witnesses were monitoring? Five witnesses were associated with your identifier? [Yes] Okay. That's the same amount of witnesses that GLEIF maintains in the infrastructure and the same that we require from QVIs to contribute to with only a small overlap for shared resources in effect. Should we be also perhaps providing guidance to wallet holders? Credential holders?

Guidance as to how many witnesses they should be addressing? – True. Well. If there's particular requirements for it, then absolutely. – Sam's recommendation was – Was five? – It was either one or four cuz he says when you have three or four as your signing threshold for witnesses you can't be eclipsed and then if you have one well it's less secure because one could be compromised.

– [Phil Feairheller:] I think a mission of the KERI Foundation is going to also be to provide documentation with recommendations and I would guess it will grow to span ecosystems so like if you going to become a QVI here's what you will need to consider here are the recommendations you're going to have to follow if you're going to be doing this in healthcare, or use other things. I suspect that KERI Foundation will be a central location for that stuff.

– [Karla Mckenna:] I've already got those kinds of requirements in the ecosystem governance framework for QVIs. Because obviously we issue credentials to the QVIs. But I could put in the credential frameworks for the legal entity one, the QVRs and OORs, a recommendation of how many witnesses should be associated with.

That would be the right place to put. - All right. Thank you. – Anybody have other questions, things you want to say, comments, concerns? – I just have a general question. I'm not very familiar with KERIpy and I'm I'm doing my own like very lightweight kind of educational JavaScript implementation and as soon as you try something like that you notice how extensive the spec is.

It's like one step five. You're doing this in the web but you're not reimplementing with JavaScript you're wrapping KERIpy. You basically get all the… like how much of the spec does KERIpy implement like pretty much all of it or?

– [Phil Feairheller:] KERIpy as it stands right now in… there's three branches 1.2x and 1.3x those implement full KERI of KERI and ACDC of 1.0 and KERI has since evolved significantly into 2.0. What these guys are… well so one of their missions besides just put out these tools is to take KERIpy and completely upgrade it to 2.0. That is still work in progress and it is the current main branch on KERIpy represents that work.

It will of course eventually implement certainly the entire KERI spec and I would guess the first milestone is as much of the ACDC spec as is needed to satisfy SEDI. I'm suspecting that's probably Sam's big goal for getting KERIpy to where it needs to be.

– But it seems like with this strategy whatever the feature completeness is of KERIpy you kind of get that for free. You get that from this except is there a distinct UI in the browser like do you have like could you have a situation where the desktop is ahead of the browser because the browser hasn't written the HTML templates to… – Yes. – Okay.

– [Feairheller:] Yep. Those are separate UI implementations. That could come slightly opposite, and I would suspect they'll eventually grow to be somewhat separate simply because of the paradigm of running something in the web and mobile versus a desktop app.

Just like if you look at one password looks similar but there's quite a bit of difference functionality between their plugin and the desktop app. – The Locksmith desktop app is that also Python everything? – Yes. – And what's the PySide6?

– PySide6. It's pretty good to work with. – And that works cross platform. Mac and everything. – Yeah. – Mac, Windows and Linux [inaudible].

TALK X

Signify Security Architecture

Fergal O'Conner

KERI Conference April 2026 · talks · 42:28

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Fergal O'Connor details KERIA and Signify-ts security architecture for Veridian: cloud agent with edge signing, encrypted key storage, client and agent AIDs linked by cooperative delegation, and boot flows that provision isolated agents while keys remain at the controller.

Hello everyone and welcome. My name is Fergal O'Connor. I'm the identity solutions lead architect at the Cardano Foundation. I've been there for the last four years now helping out build out our open-source identity solution known as Veridian.

Last year we launched the first mobile wallet in the KERI/ACDC ecosystem to production. This is a production-grade mobile identity wallet. It's gone through security auditing and pentesting. And in general we put a lot of effort into the user experience of this wallet and how we could bring KERI and ACDC to the masses and make it more accessible.

The wallet itself is based on KERIA and Signify-ts. And throughout the years, we've put a lot of effort into contributing to KERIA and Signify-ts in the open source community. Particularly because we were trying to bring in this extra user experience.

There was kind of gaps in the existing software that was required to deliver that user experience we wanted. And also some kind of security edge cases kind of operationally that we had to add in, to align with our pentesting.

Today I'm going to be going over kind of the security architecture of KERIA and Signify-ts. This is in the advanced track so we'll be quite technical. I think it's generally okay if you have kind of a background in into KERI but some parts might be a bit low level.

Now, the first tagline I want to start with is: KERIA is a cloud agent In the KERI ecosystem. And kind of in line with everything that Sam [Smith, red.] was speaking about this morning, I put the tagline here of “Agent in the cloud, controller at the edge”. Just because you're using a cloud agent doesn't mean that you give up control of your keys.

If you have a mobile wallet which is using a cloud agent based on KERIA your keys will always do the signing at the edge on your phone. And I also took this snippet directly from the KERI specification. Sam defined that a controller application needs to do five functions.

This is keypair generation and storage, key event generation and signing and key event validation. Of course this is at the lowest level of KERI. It doesn't talk about higher level protocols such as ACDCs or OOBIs or IPEX and so forth. But if you have that controller application with those five functionalities. You can, if you wish, split it across two kind of software deployments of a controller application on the left and a controller agent on the right. KERI is your controller agent. The split here is that your controller application, say your mobile wallet will have to do the keypair generation, key event generation and signing. And then your controller agent in the cloud can do keypair storage and key event validation. Now the keypair storage of course has to be encrypted because your agent cannot access those keys because that would break everything. And I think also with this diagram you could put the keypair storage in either side that works in the setup of the deployment.

But critically the key event validation is pushed to the agent in the cloud. Okay. Particularly when you start adding in other protocols such as ACDCs and so forth, you're pushing all of this complex validation logic that must be spec compliant and so forth to the cloud and abstracting away the details in a way that your controller applications can be a bit more lightweight at the edge. All you have to do is sign some events, essentially, and not do all of the complex validation.

Okay. Today we're going to cover kind of three areas. First and foremost the kind of service architecture of KERIA and Signify How it works in general. Then we'll delve a bit deeper into the trust relationship between the cloud and edge. What keys live where you know what must be signed to establish this trust relationship and so forth. And then finally to wrap it all up we'll see and explore how to use KERI to interact with the rest of the world once you set this up.

On the service architecture… As an overview before we get into the diagrams: KERIA is of course a KERI agent in the cloud. It's based on KERIPy directly. Is the Python service that uses KERIPy as a direct dependency. Most of the complex validation is from KERI core. It's essentially an orchestration unit wrapped around it. It's a multi- agent service. It allows you to provision many agents for many controllers and kind of to summarize it's a set of HTTP endpoints essentially for controlling these agents.

Provisioning them utilizing them and to interact with the rest of the world. Signify on the other hand is your edge client library. This is what your controller application would use. Again the main idea here is to sign at the edge. I believe in the early days people were going to call Signify 'KATE': “keys at the edge”. They stuck with Signify however. But it's essentially just basically like an API wrapper around KERIA. Except they can do this key event generation and signing as well.

It understands bits of CESR and so forth. The relationship is: for every edge client there is one cloud agent, a direct one-to-one mapping. And it's available in multiple languages. Within the WebOfTrust the two complete implementations are Signify-ts in TypeScript and SignifyPy. I don't believe SignifyPy is used in production but Signify-ts is used by most people in this ecosystem, to be honest, that aren't building directly on KERIPy. In the Cardano Foundation we've also built a full Java implementation of Signify. It's fully feature complete. That's actually been used internally at the Cardano Foundation and a number of other projects that we're working on which have actually gone to production.

And in general one nice thing about Signify is that right now it gives you a higher level API that is more accessible to actually utilize KERI. If you were to build directly on KERIPy there's a steep learning curve to understand how it works internally. And it would be very easy to go wrong. Whereas with Signify the API is a bit more like: create this identifier, create this ACDC registry, issue this credential.

I think having that higher-level API also brings much more accessibility to a wider audience for adoption and because it's available in multiple (programming, red) languages that also makes things easier. I'm not going to say it's super simple to add new languages for Signify but it's much more trivial than porting KERIPy to various languages.

If we to look at this overview diagram of KERI is here in the center as I mentioned it's a multi- agent service. We have agents A, B and C. Each agent has its own fully isolated database. These databases are mostly provisioned by KERIPy. They're a set of LMDB databases but they're fully separated and for each agent you have exactly one Signify client and they have a secure API between them. Okay. Client C cannot talk to agent B.

All of your state will be held in the cloud. Signify is completely stateless except for the key material that you give to generate and derive the keys. All signing happens at the edge. Signify will push signed events up into KERIA and your agent will do whatever with that. It'll make propagate it to witnesses send an exchange message to a verifier or so forth.

And all interactions with the rest of the KERI network. Agents, witnesses, watchers, controllers and so forth, all happen via KERIA. And these will all be direct CESR inbound and outbound streams. The Signify clients will never talk directly to the other agents, witnesses and watchers.

It's always through your agent directly. And if I take that diagram and maybe zoom in a little bit on KERIA. Again we have the three clients and three agents. Each agent has its isolated database, and KERIA itself is a set of HTTP endpoints. There are three endpoints 3901, 3902 and 3903.

The last one 3903 is used to provision new agents. Okay. If you come along if say Signify client D installs a wallet on their phone which uses Signify and they go through the onboarding process, they'll be able to provision a new agent through this endpoint which will set up all the database and so forth. And that endpoint uses the the “agency” it's called internally, and the agency has its own database to track you know which client relates to which agent, which identifier that was created by a client relates to which agent.

It's pretty small, to be honest. This port 3903 in production shouldn't be left open because otherwise you could just write a loop and keep provision new agents with more databases. In Veridian we completely block that port from public access and we have one-time boot URLs that are sent by emails to protect the infrastructure.

Once the client is onboarded in and they have their agent up and running within KERIA in the service they can then use the 3901 port on the top left which is the admin API. This is just a set of REST endpoints, create identifier issue ACDC and so forth.

It's completely authenticated. Of course only one client can talk to one agent. We'll see more on that in the coming slides. And then your agent, it might be talking to other nodes in the KERI network to say "propagate events for witness receipts", and so forth or send an exn message. Like everything in the KERIA implementation, everything's async. The response will often come back in on a different port.

When it comes in, it'll come on to the message router. And the message router will based on the headers it'll look up in the agency: "okay, someone external is trying to talk to this AID which agent should I route to", and it'll route to the correct agent and they'll parse the incoming CESR stream using that database specifically.

That's kind of generally operationally how it works and again just to recap there are three different ports. The boot port is used to provision new agents. The admin port serves all the rest endpoints for the clients and the external 3902 port is the CESR endpoint for inbound interactions; okay?

If we to delve a bit deeper into the trust relationship between KERIA and Signify. You've onboarded into the system but how do you actually authenticate on that API? I created this sequence diagram here. It's probably going to be the most technical part of the talk. On the left we have the Signify client and here we have the two ports that the client uses: the boot port and the admin port. I don't know how clear it is to see there but there's a blue box there in the top left which is the boot sequence to provision a new agent Signify itself is completely stateless except for a passcode that you must give it; okay? This passcode is used to derive key material. It's a 21- character salt essentially, in the CESR base64 URL variant.

Signify has an API to generate these passcodes or they can be generated externally. It doesn't matter. You just pass it to Signify when you're connecting, booting or connecting to the API. This 21-character passcode, which is a salt, then gets stretched in a hierarchal-deterministic way using a path and the Argon2id function.

This allows you to drive keys, ed25519 keys specifically. At index zero, you get your first signing key. Index one you get your rotation key. These two keys together you can use to generate an inception event for your client AID which I've called 'CAID' here. This client AID will be used by Signify to communicate with KERIA and to authenticate on the endpoints. It's KERI 'direct mode', there are no witnesses because this is directly between client and agent. This client AID is not used to interact with the outside world. It's only used to authenticate on that endpoint in KERIA. Once you've generated the inception event for the client AID (you of course sign it with your signing key) and then you send the signed inception event on the boot port.

Once KERI receives this if it's a valid inception event it will provision a new agent. It'll set up all of the databases. It'll generate a salt for that agent in the cloud because the agent itself also needs its own AID.

And this AID which is a AAID here, agent AID is actually a delegated identifier. Okay. It's delegated back to the client AID. There's this kind of tight connection between the two AIDs. And it'll send that back in the response and because it's delegated and KERI uses cooperative delegation. The Signify client will need to approve of the delegation and send that back the approval event which is just an interaction event essentially which contains the delegation seal. The main reason you would do delegation here is because Signify is completely stateless. It can't remember on its own what agent the ID was talking to that it set up with initially. But if KERI can give you back all of your Key Event Logs and it can verify it, it has proof within its own Key Event Log that it the agent AID was the one it created this kind of trust relationship with. I don't think Signify does that yet, but it could.

Yeah. One thing I've conceptually been confused about is that and you probably just explained it, but I needed to hear it again. Is who's delegating to whom and what is being delegated? You probably just said it, but I need to hear it again. And is for just that event or is there delegation used later, beyond ...?

When you create a delegated identifier in KERI, it essentially means this identifier is not valid until the other identifier approves it. They put essentially the AID of the delegator inside their KEL in a seal in a new interaction event. And now every time the agent ID would want to do a rotation event, the rotation event is not valid until the Signify client approves it. Okay, it's like a two-step approval. GLEIF use this with QVIS for example. If a QVI rotates its keys, GLEIF has to approve it for that rotation event to be valid.

Cooperative delegation and KERI is just another security mechanism essentially. But once that delegation is done, then the agent can sign for the controller for something. No, no, no, no. It's not a delegation of authority. That would be more like issuing ACDC to someone.

Cooperative delegation is a security mechanism essentially for rotation events. Okay. It also for superseded recovery where you can like step back in the KEL and fork like recovery. It there's the mechanisms in which that's allowed are much stronger, much more powerful in delegated identifiers. Delegated identifiers are for security essentially. It's cooperate delegation. But in this case, it also has the nicity of the Signify client knows it should be talking to this agent AID. It has proof that's who it should be talking to. At this point you essentially have your agent fully provisioned and fully ready in the cloud. And you have your stateless client which just has your 21-character passcode which in the code

is also called a brand there's different naming everywhere. At this point, this is only needs to be done once, you can now use the other endpoint to do whatever you want to do create identifiers, issue ACDCs, and so forth.

Now the first thing you need to do because of stateless is to fetch the state of the KELs for direct mode. It fetches both the KELs of each of them And the sequence number of the derivation path that way it knows how to derive the keys again for its current signing key. On your client AID if it did a number of rotations once it starts up again it needs to initialize itself with the state and this is an unauthenticated endpoint because it doesn't know what key to generate yet.

Once it has done that and derived its current keys it can create HTTP requests on the REST endpoints And fully sign them with the client AID and the responses will be fully signed by the agent AID. We're not using bearer tokens here.

The HTTP requests are signed back and forth. Does that make sense? Any questions? I won't explain this, cuz I might run out of time, but the passcode that you use to boot you can rotate that full passcode. You can switch to a new salt essentially. This functionality exists in KERI and Signify already.

The reason you might want to do that is because you might think your passcode's exposed. Now, if it's fully exposed, you're already gone. But let's say you have a time window to recover from that. You can actually rotate the passcode. How that's done, I won't explain right now. I'll do it at the end if I have time. This QR code, suppose you might be able to see that work is a link to the markdown file from Phil [Feairheller, red.]

in the repository. Which explains everything I just explained this and everything I'm about to explain quite clearly. If you want to look at it after finally one other thing that I want to mention is so I just said that the HTTP requests between the client and agent are fully signed. They use an RFC 9421.

Okay, this is an existing standard outside of KERI. It's just been slightly adapted to KERI, which just means the key that you should be using for signatures is the latest key state of the KEL essentially. This is the in-vanilla Signify-ts and KERIA; this is how you do it.

I have an open PR for ESSR support. I think Phil mentioned ESSR in the last talk. On our fork of Veridian. We've been using ESSR for I don't know the last like 14 months or something. The problem with the signed-header approach is that you have your authenticity because everything's signed by the AIDs but you don't have your confidentiality. You have to lean on TLS. If the TLS connection was compromised, you lose confidentiality. Whereas with ESSR it's basically your best way of combining authenticity and confidentiality.

Sign -, combining encryption and digital signatures. It's what is used by SPAC and the trust spanning protocol. Phil [Feairheller, red.] implemented it in KERIPy. We've extended it down to the API between Signify and KERIA. It looks something like this. I won't spend too long on this, but it's designed with Sam [Smith, red.]. It's like a HTTP request tunneling kind of thing. The green at the top left is the original request Which we would send say HTP request for put 'slash' identifiers with the body and the headers and so forth.

We serialize it into a HTTP string and then we do a cryptobox seal with Libsodium to encrypt it to the agent AID. Only the agent can decrypt it. Cryptobox seal is a form of hybrid public key encryption. For those of you who are aware, asymmetric keys are incredibly inefficient at encrypting data. It's just not really feasible.

Symmetric keys are much quicker. The crypto box seal from Libsodium is a form of hybrid public key encryption which uses your asymmetric key to derive a symmetric key essentially. I don't know the internals of it, but it's what SPAC uses.

This sealed box then can become the body of a new request which we send to this post 'slash' endpoint. When we do this, every request that's sent from Signify client, nobody knows where it came from. They don't know what we're even trying to do. They don't know that we're trying to update an identifier because it's going to post.

Only the agent will be able to decrypt it. And in the headers, we put a signature over this, essentially. Now, the signature needs to be over the digest of the body, a timestamp for replay-attack protection, and the sender and receiver AIDs.

And additionally in encrypt sender sign receiver, you also have to encrypt the sender. That's why the sender is also a header of the original request which was encrypted. Only then are you fully secure from all the attacks that Sam (Smith, red) mentions in his SPAC paper.

Okay. Yeah we've gone over the general architecture and also gone through the trust relationship. Now, finally, the last kind of piece to tie it up will be how you use KERIA to interact with the rest of the world. Up until this point, you've basically installed a wallet and onboarded onto your cloud agent and done nothing else.

The first thing you'll probably do is create an identifier, maybe resolve a OOBI, but probably create an identifier. In the codebase these are known as managed AIDs because the client AID we created earlier is only used indirect mode to talk to your agent.

It's only for that communication. It's not used to receive credentials, issue credentials or anything else. We can use the '/' identifiers endpoint on KERIA to create managed AIDs. And these are, well, they may be private or public bur these are AIDs you would use to interact with the rest of the world or use in your use case, they're more user facing.

But once again the keys remain at the edge. Now this is a bit technical but for those of you who know KERIPy: In KERIPy you have a 'Hab', which represents an AID and you can incept a Hab and you can sign with a Hab and so forth. You can have group Habs. Also within KERIPy there are Signify Habs and Signify group Habs. It's just so the code knows that the keys are not readily available. The keys exist somewhere else at the edge.

If you see that in the code that's why it exists. Now, you might be asking where did the keys come from? Maybe a naive approach would be to use the passcode that we used to create the client AID and derive more keys. That would be problematic because of passcode rotation and it also would go against what Sam said this morning of being able to use just your KEL to derive exactly which sequence number on the derivation paths is required.

Actually for for each new AID you actually create a new salt. They're fully separated. It's of course randomly generated by Signify at the edge. And it's stored encrypted on KERIA. This is the default mode in Signify. It's of course encrypted with the client AID. And you have a salty keeper which tracks for this AID, what encrypted salt does it have, what derivation pad is it currently on, how many rotation events is done, and so forth.

The main reason you would want to split these is if you did a passcode rotation and your one passcode had all your keys for all of your identifiers when you want to rotate that passcode for whatever reason, you would have to issue a rotation event for every single AID. And that would get very cumbersome as well if you also had group AIDs. You had to coordinate that. This way there's a kind of a clean separation kept from them.

You can also .., I think Phil had this in his last presentation as well, you can also use' Randy', which means, just instead of a salt, which is used to be stretched with argon 2 ID to generate keys, you can just randomly generate keys directly and store them encrypted in KERIA; same idea. this is kind of goes back to that first slide I had where from the spec that your keypair storage can be done by your agent.

Now in case you don't want to sign with Signify.

In case you don't want to sign with Signify or expose your salt over the internet like that, you can also use an external source for signing such as a HSM. This, I believe there's a current open PR in the community to fix that on KERIA. But you can use HSMs and directly sign all of your key events externally.

Now once you've created an identifier so you would generate that inception event generate the salt whatever post that to KERIA, to communicate with the rest of the world you would need to generate an OOBI. Each managed ID can now set up their OOBI. This is probably going to get a bit technical now but if you know how OOBIs work this will make sense. If you don't, this is probably too much information, but in KERI nodes or whatever can have various roles in the system such as witness, watcher, agent, controller.

If you were to build a wallet with just just KERIPy and not have the split of controller, agent, your OOBI would look something like /OOBI, your AID/controller. Okay, you have the controller role by default over your AID. But there are other roles in the ecosystem.

One is Agent of course. You may As the controller of your managed AID authorize your agent in KERIA to act on your behalf in the agent role essentially. This indicates to everyone else in the world if you want to talk to me, direct all traffic to the agent. The agent will handle that essentially and this is called an endroll authorization.

Again, this is a bit low level, but the agent itself needs to say where its endpoint is which is the two 3902 port. it needs to create a signature over that to commit to say "I'm at this URL" Veridian.id, or whatever. That's called a location scheme in the code And additionally another thing that came out of our pentest was actually securing the communication not just between the Signify client and the agent but agent to agent.

What we've implemented on our fork is we've added full ESSR support that's also in line with SPAC and the trust banning protocol. All of the CESR streams inbound and outbound essentially wrapped in the most basic form of the trust spanning protocol.

The work for that in KERIPy I merged a while ago with Phil that's done. The KERI work I still need to open a PR for but we'll be contributing that back soon. And finally, this is probably the most technical part, but this is only if you understand OOBIs. If you don't, then it's probably too much, but just to kind of show the relationship between all of the AIDs that we mentioned today. There's kind of six slots here. The top one is a managed ID I created using the admin endpoint. This is its key event log. It ends in two 'z's that's its AID. It has the three demo witnesses from KERIPy. These are the signatures and the witness receipts.

If you look at the very bottom I give the end-roll authorization from my controllerAID that ends in two 'z's to this endpoint AID that ends in 'XA'. I gave it the role of agent. This over here, see the cursor?, This is my managed AID. This is my cloud agent 'XA'.

I can also see that my cloud agent here. It signed a location scheme to say you can find me at this URL and no other URL. That's how OOBIs work. It's very very particular. If you look at the one up here, this is the Key Event Log of the agent. You can see it's a delegated AID.

It's delegated to this AID here, which is my client AID from the initial sequence diagram. The second one is my client AID KEL. And here's the interaction event that approved the delegation. You can actually look in the KEL and see the full kind of sequence of what I mentioned end to end essentially. That's all I had. I think I have seven more minutes I will very quickly go back to this.

When we created the client AID, it had a signing key and a rotation key on sequence number 0. If you were to drive more keys and do a rotation event on that same passcode, it would just look like a normal rotation event where the next key becomes the current key and there would be a new next key that's blinded for postquantum security. But if you want to switch from an old passcode to a new passcode, okay, in such a way that you don't need to remember the old passcode when you're done and you don't want to because someone's going to compromise it, then you need to generate your new signing key and new rotation key on the new passcode. All right, so here's the rotation key hashed from the new passcode. Here's the one from the new passcode as well. And here's the unblinded one from the old passcode. If you issue a rotation event, you have to unblind a threshold amount of next keys from the previous event. This one from the old passcode has to be here. Otherwise, it's not a valid rotation [red.] event. And this or a rotation event, sorry. And this rotation event has to be signed by both of these keys. Okay? Because each new event has to be signed by a threshold number of new signing keys. And it also needs to be signed by a threshold amount of previous rotation keys.

That's why it needs to be signed with both of these. And the trick here is if you look at the threshold, the threshold is splitting away 1 and 0. The old passcode even though the key is inside the rotation event, it has no authority from this point on. It has a weight of 0. It can do nothing essentially. But it just fits the validation logic of rotation events but completely gets rid of the old passcode.

Okay. It's kind of a it's a partial rotation, essentially, for a single-sig identifier, which is kind of unique. Yeah that's it. Any questions? Have you mentioned anything about retiring an AID? Could you discuss that if that's okay?

I mean it's not really specific to KERI and Signify but If you want to abandon an AID, as Sam (Smith, red.) calls it, You have to do a rotation event which just sets all the thresholds to zero I believe. At that point there's no more signing authority of also of the the signing keys and the rotation keys essentially. It's just dead. It can't do anything. It's called abandoned at that point. But I don't believe anyone's done abandoned AIDs.

It's probably in KERI core and KERIPy but it's not exposed in Signify. – I have some questions that may be tough for you to answer and because I've struggled with these over a long period of time looking for an answer. The one might be simple the first one might be simple is what could work offline in a wallet without the agent being online.

Can you sign things? Can you issue credentials? In the current model? No. You could sign them and cue them up, but they wouldn't actually be effective on the network. They need to be on the KEL or TEL or witnesses, etc. Yeah.

You need some infrastructure. But you might be able to prove .., you might be able to sign something, but it couldn't be logged anywhere. Could you sign something, right? You could sign something. Okay. And if you're using direct mode KERI where there's no witnesses and watchers, then you don't need the infrastructure level. You you could sign something and send it.

Yeah. Okay. One of my concerns with KERIA in general is that to my knowledge it doesn't support high availability and disaster recovery kind of modes today. Right. If the hard disk crashes or sorry: solid state disc drive crashes, you're out of luck on your KERI service. Is that correct? And what is the roadmap, if any to make it high available or disaster recovery of KERIA I have a whole talk about that tomorrow about the road map. – Oh awesome. I know that's a tough question, but it's an important one, I think.

And then there's a similar question is Sort of with vendor lock in. You host your own KERIA or use another one from one of the QVIs or providers. And you have some vendor lock in at that point, what can you do to move to another KERIA, hosted by someone else? Has that been discussed?

Again, that's something that's more on the road map. Like you can do that. You have control of all the keys. More of a problem would be the data. You would have to have, the previous provider would need to be okay with moving that unless you're going to sync down to Signify, but it's not all available on the KERI interfaces, right?

Yeah. But you could make it available, right? That doesn't exist yet. What you're exact asking for it. Okay. Yeah. Okay. Yeah. But it is something that we've been thinking about. It's just again on the road map. One problem I have with the if it's with this Is if you lose your KERI instance, you lose all of your AIDs.

The passcode that you have, that you're entering only gives you access to KERIA essentially. If you lose your encrypted salts, you've lost your AID. I didn't realize that was that those things were "salty". I thought those were .., the encryption key was also derived from the passcode, but no, it's the encryption key is, but having the actual cipher text to decrypt, it's stored on KERIA. You do use your client AID. Oh, you use using asymmetric encryption or symmetric encryption.

Asymmetric. Oh, it doesn't need to be fast. Okay. It's just salts, I believe. Anyhow, but these are stored encrypted on KERIA. If your KERIA provider went .., disappeared, you've lost all your AIDs, unless you have backups to these.

Yeah. the combination of .., without redundancy of data, ideally across regions, and so forth, and without this recovery, it really makes KERIA vulnerable in production today. Yeah. Well yeah there's the path to production is kind of long It's not really production ready at scale or at any trust level in my opinion; is that a fair statement?

It depends. I mean we've done mitigations on that. I mean for example these encrypted salts are also in our wallet and we have our own backups of those because I mean okay the data is one thing like if you lose all your ACDCs you can go back to the issuer. Okay.

Right. The main thing you need to keep is your keys. It would be annoying to have to do that and build it all back up, but And of course, our deployments have backups every few minutes. And I know you're in on the same channel, but just for everyone's ears, it be nice to see road maps from WebOfTrust to say, "Yeah, we're intending on doing these things this quarter or whatever."

I know. The same. Thank you. three minutes. Any more questions? Maybe one thing to mention as well is what we are working on with regards to migration for the road map discussion is how the Veridian wallet can request the cloud agent to provide a backup file. Then you can migrate to your own KERIA or a different KERIA provider. You can actually trigger that exactly export and then as long as a KERI provider is able to accept that import.

I mean this is also an element of the GLEIF ecosystem, right, you can move from QVI to QVI. There are ways to migrate but it does require there to be it's also not trivial because you have to update all the OOBIs and everything like that and so forth. There's some work to be done.

It's definitely something to think about. Cool. But this is why we put a lot of cost and resources into actually how we back up the KERI data. We assume that there will be a catch failure and we pay a lot of money to have high availability of that data.

For the production services that we use in our own organization foundation and for any clients a lot of cost goes into high available to data center (inaudible, red) .. disaster .. But that is an implementation choice and how you deploy. Are you describing how your cloud provider and your EC2 kind of things are set up from a data storage How we're making sure KERI does get wiped Frequent snapshots.

Yes But that isn't really a protocol-based thing. Right There's no protocol issues with KERIA, really, it's more of an operational thing you know that's where the gap is immaturity. Snapshotting is not necessarily the most elegant way to provide that high availability redundancy, especially.

No, but that's disaster recovery. That's not for high availability. Yeah. Like horizontal scaling is a different thing. (Aplause)

TALK XI

KERIA & Signify Path to Production

Fergal O'Connor

KERI Conference April 2026 · talks · 39:46

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Fergal O'Connor reviews Veridian's production path with KERIA and Signify-ts, noting maturity gaps versus KERIPy. He advocates outbox patterns, idempotent APIs, and eventually CESR streams instead of bulky HTTP embeds between edge clients and cloud agents.

My name is Fergal. I'm the identity solutions lead architect at the Cardano Foundation. I've been there for the last four years building out our open source identity solution known as Veridian. Some of you may have seen that this morning in Thomas' presentation on the 5Ws.

Over a year ago, we launched the Veridian wallet to the public app stores. This is the first KERI/ACDC mobile wallet in production. It's based on KERI and Signify. We invested a lot of time and effort into bringing a nice user experience to the user. Abstracting away the details of low-level KERI and giving something that's actually usable to the end user. In doing that we had to actually contribute a lot of things back to KERI and Signify.

Yesterday I spoke about the security architecture of KERI and Signify. But today I'll talk about the path to production. I think there are some maturity gaps in the implementation of KERI and Signify that should be addressed for adoption of the KERI suite. It's starting to lag behind the maturity of KERIPy. We'll kind of step through that today.

I mean yesterday's talk gave kind of an in-depth explanation of how it works in practice. But kind of the tagline that I had was agent in the cloud controller at the edge. KERIA allows you to deploy cloud agents which essentially handle all of the verification of KERI/ACDC/CESR up in the cloud and allow you to use an edge client library known as Signify at the edge to just assign key events essentially or ACDCs this keeps the edge layer very thin.

For example, Veridian wallet on our mobile just uses a Signify-ts to sign things essentially and all of the complex validation logic is pushed to the cloud kind of like a black box essentially. In many ways it becomes a more easy to use API that's higher level which is desirable for adoption It gives something that's a bit more tangible for people to develop on.

And very quickly, this is taken from my deck yesterday, but doesn't look so good on the four screens there but if KERIA is this kind of block in the middle is the multi- agent service it's based on KERIPy. It's a Python service. It allows you to spin up various agents in the cloud that are fully isolated. They have separate databases. And for every agent in the cloud, you have exactly one Signify client at the edge. This these could be various applications from various people sharing the same cloud agent. This could be on a mobile device for example. And there's a secure communication between each client and agent. Which I delved into yesterday. I won't do that today.

And the agent is also responsible for interfacing with everything else on the network. Such as witnesses and watchers and issuers and verifiers and so forth. Signify only ever talks to the cloud agent. – Quick question if I may. Why must there only be one Signify client per agent? For example, if I had a Veridian wallet and I had a browser wallet. Why couldn't I both connect to the same agent? What would be the problem with that?

Do the Signify clients have different key material? – No, same password passcode. For me that's moving around the passcode a bit too much. It's the same user though, right? I have my phone and I have another two phones or two devices. What's the problem with that?

There wouldn't actually be a problem with that if it's effectively the same client. If you're using the same passcode, it is effectively the same client, just a different instance of it. It'll derive the same key material and authenticate in the same way.

But if your applications keep local states you might it might cause inconsistencies essentially. – Okay. But it might work. It would work. Although it might then .., how it actually stores data on KERI needs to be the same. You've seen this when you've tried to connect your wallet to what was our wallet. It works but you see some fuzzy stuff that's related to our wallet only.

This is generally the setup. And today I've kind of split into three different sections of what I will cover on how we can mature KERI and Signify. First is focused on the developer experience. The second on the operational resilience and third on scalability. I think the first two are the most important to do in the short term. To really you know increase the usability of the stack.

Scalability is more relevant in the long term. A lot of people think about this now but these first two points are more relevant in my opinion and scalability will be very relevant for SEDI though you know if we're launching KERI instances that want to serve thousands and thousands of people with wallets and scalability becomes a big concern. And just to preface as well, I mean this is going to be a talk probably explaining all the bad points of KERI and Signify, but it does work.

It is in production. It's just maybe a bit nuanced to work with. It's very unless you understand it, it can be hard to work with because there's a lack of feedback in places. But all of that is really just traditional software development points that are missing.

Just some extra maturity. It's not like it's not anything related to the protocols as such or the security. And in places where I say that, oh, and in places where I say that 'Oh, there's not enough API hardening in place', the reality of that is just something happened. The KERIA crashes.

It's not a security bug. For as so much as I can see, it's more about a resilience thing; okay? On the developer experience the first point on this slide refers to 'At the edge.' Everything in Signify essentially. We've been working quite hard over the last at least 6 months I think to strongly type the edge libraries so I mean if you're using Typescript you should be using types essentially but when Signify was originally written It didn't have any types in it This is problematic if you're refactoring code and so forth things start to break. We've put a lot of time and effort to strongly type Signify. It's not fully there yet, but it pretty much almost is. And instead of adding types directly into Signify, we've actually taken a kind of A different approach because as I mentioned yesterday, Signify essentially is an API wrapper around KERIA. It can do extra things like generate key events and sign them, but it's otherwise an API wrapper. And since you have an API on KERIA, you can let KERIA decide what it wants and what it will respond with through its open API documentation. It will say at this endpoint, it will give a response in this format if it's a HTTP 200 or whatever.

And this is common practice, to take open API documentation and autogenerate types in languages. That's what we've done: We've updated all of the open API docs on KERIA and generated types in Signify-ts and Signify Java because they propagate to various languages now as a single source of truth and if you wanted to add in another Signify like a Rust library you could just automatically generate them. Now you might have to do some reworking in places.

Java for example is not as expressive in its type system as TypeScript is. It didn't play nice with a lot of the semantics of CESR. But we've kind of got it working now. It took some changes there but it's quite useful and also we're not hand rolling the open API docs. We're actually taking the data structures in KERIPy that Sam has written saying "Here are the top level fields of an ACDC, we've converted those into Python data classes automatically generated so they can be reused in KERIA and those data classes are converted to the open API docs" and you what that gives you is, if there's an update you know downstream, it propagates upstream and the open API docs don't go out of sync. This has been a huge lift to make it more usable.

Internally, I think Signify still needs some work. Probably we can get away with not doing that for a bit. There's maybe some other points that are more relevant to that point. I would say we'd maybe tackle this when we're upgrading to CESR2, which would be quite a bit of work as well.

Quick question if I may. Some of the WebOfTrust developers refer to Signify meaning ..? Is it what Signify-ts has in it? I always thought of it as like the protocol that KERIA exposes. Is Signify a component or is it a protocol or is it all things or I don't ... How would you define what Signify is? Sorry it's a simple question, but it's a language question mostly.

An 'Edge client' that you can ... – Any edge client of KERIA? Yeah. – Okay. Got it. Yeah. There's a KERIA protocol essentially, – Right, it's defined as a protocol, but it could be Signify JS or Signify Java or Yeah. whatever other language.

How does it look like? Is it UX stuff or is it command line stuff or library? Library. When you're using Signify-ts, it's like having a compiler error if you type in the wrong formats, essentially. If you try to create an ACDC or something, it will know what the structure should look like and you can reference it directly.

Whereas before you're kind of blind, essentially. Maybe you receive an ACDC back from KERIA and you were going to present that in the UI and you would say I want to get the top level 'I' field as the issuer. If you were to reference some field that doesn't exist, like 'T', before it wouldn't complain and you just get a blank space in the UI whereas now TypeScript would say there is no top level T field in an ACDC.

I'm a strong believer in strongly-typed languages. The payoff is very high in terms of keeping the bug count low and especially when you refactor. To your question: It's a library you need to be able to build that you want on top of sort of a hole in the wall to service food. Like you give to the other side.

The delivery desk. Yeah. you have the back office which is KERIA and then you have this (service window, red) and then you can deliver it out to the UX (User Interface, red) "Serving window" or something? Yeah. Okay. And with this you know what's going to come out, what shape it will be, what comes through the serving window whereas before it's like you're blindfolded.

No, I'm good. Another one which is a bit technical. Currently with in all of the Signify implementations When you want to do something like issue a credential in one command on Signify it will construct the credential Try to create the interaction event to anchor it in your Key Event Log. It will pick the sequence number for you and submit it to the API all in one go.

Absolutely. Which automates workflows of several steps. Yeah. Which is not a good thing because if you lose the HTTP response you've now lost data; okay? It's okay when you use Signify completely stateless in smaller applications but when you want to build enterprise applications where you need mission critical reliability you need things to go through. If you want outbox patterns, you need to be able to decouple those things. You can create and sign at the edge and then submit.

And the API should be idempotent. Yeah. On that particular note, I want to make I want to hear more what you have to say on outbox patterns. That's actually something I was looking into. It seems like that while we're considering this, we should consider the API, the fundamental design of the API between Signify and KERIA because one of the reasons why we have so many different HTTP payloads instead of just CESR streams which if we were to just had streams the outbox it would be very straightforward is because we didn't have a fully a complete parser and cryptographic primitives implementation in TypeScript when Signify-ts was being first created.

We ended up doing, a bit of what you would say is, a bulkier API from Signify-ts to KERIA because really we should just be passing CESR strings around. And instead we have these and embeds that we put in HTTP messages. I it's sort of a something that I've wondered about as part of maturing the Signify and KERIA. But even before we even get there, I do think we should architect everything the way that you're saying about outboxes: save everything locally. That makes a lot of sense.

But I'm just sort of saying if we're going to do that anyway, maybe we should reconsider what we even send per endpoint. That's a big refactor as well or change rather. But you could put helper functions, also in Signify-ts, sorry, that either give the structure or raw C.

Okay, going I'll finish because I have like 10 sides of this. This is on our fork. We've done this already with identifier creation. We create the identifier locally and then submit it and it allows us to get reliability in our wallet, which we couldn't get before. But we probably need to apply this everywhere.

Can you say that again? You create the identifier and then submit it. The post call is decoupled. We can put it into an outbox locally and push it up. We've gone to a lot of lengths for reliability in our wallet, but you need to do things like that for it. And you also get better control in the sequence number for anchoring.

If you're trying to assign multiple things at once, especially in a group multi context, having better control will be better. – How is this how is that just different from Signify right now? Is that what the 10 slides are example, right now in Signify, your IPEX is decoupled. You create the IPEX message and you submit it.

It's two calls. Everything else isn't, but it should all be like IPEX. I see. Yeah. Wants to go through. Maybe I should leave the room. All right. Okay. It's just an example of the TypeScript generated, a snippet of it. It's a very long file.

And in Java there's like hundreds of classes generated but .. This is what we're contributing. The other developer experience issue, and this is actually probably even a bigger issue, And you see this all the time on Discord for new developers is Silent Failures where you submit something to KERIA and nothing's happening essentially.

Now with KERI when you submit things that take time you get an operation back and you can track the operation completion such as if it's waiting for witness receipts and so forth. But it's so common to have operations that are stuck in pending forever and you have no idea why and you can't trace it down and it's because you had a different SAID in this place of this .., and there's very little feedback to why that happened.

– You want to put more weight on it? You just need more you just need better feedback patterns essentially. It's probably a case by case basis to do that. This is at kind of the API layer. It needs work. – But there could be like a 'result key' kind of pattern applied to the call so that if there's a failure in any call, it's handled in a consistent way/ Yeah. But this is not the call. The call is like it accepts it and it puts

it onto a parser and then the result of parsing that doesn't result in a fully signed something because something was wrong and you have no idea why and you're tracing through the logs and there's a million logs.

I don't have a I mean this is case by case basis so you'd have to step through the code and find the biggest issues but even if you historically look at the GitHub issues or Discord you'll see what pops up most. This one I think we fixed this one and actually caused a different issue or not sure.

We made a change. Cuz it wasn't anyway. But in general it just needs work. I would say it's a barrier to entry. I've seen this too often where the operation just times out and you have no idea why. I would think they're probably the two biggest things you could do for the developer experience. It's actually easy to build on because if you get stuck wondering why this isn't working for hours and hours, it's If you finally convince someone to try KERI Yeah.

and they start developing and this is experience. Yeah. But this these are easy things to fix. It just takes a little bit time to go through certain edge cases. The harder one to fix is the typing which we've almost done. That's the good news. The second so that was developer experience. The second kind of section is Operational Resilience you know after you deploy it what that looks like. Some of these might bleed back into developer experience though. Related to the last one is API hardening. They're kind of overlapping. But additionally not just you know KERIA getting stuck in the state where things are pending. It's also a bit crash prone from certain bad inputs.

What happens if it crashes? If it's in the Docker setup, it just comes back up. That's fine. But we even see in our developer KERI deployment that we use at Veridian and you'll have some random person in the community trying to do the previous slide sending the wrong input and they crash our instance and all of our wallets are flashing to offline mode because KERIA has gone down and up, it needs a bit more resilience, at that point. It's not on the parser. It's actually just on things like the granter and things like that. All the dexter and the agent in file.

It's the development, not the production. Yeah. He says our wallets are going up and down. That's the development environment. Yeah. They don't have access to our production. And, of course, KERI is multi- agent. Okay. like I just said, you're now denying service to other people just because you're submitting the wrong thing. And it bloats the logs with lots of errors. That could be nice, information back to the user.

You submitted the wrong thing. I believe this was the one we fixed recently. It doesn't crash anymore if you point your IPEX Admit to the wrong ..., instead of pointing it back to the grant, you point it to a different string. It caused a crash. Now it causes a an operation that's pending forever.

You just pushed it down one layer. Another one is application layer reliability. I think there's kind of a misconception that the escrows in KERIPy are meant to provide application layer reliability, but they're not. They're more they're more so to handle out-of-order processing; okay?

In reality, if you want full reliability at the application layer between two wallets, their agents will need to continuously resend information Until they get a acknowledgements back. For example currently if I send an IPEX grant and my KERI sends it to the port of another KERI instance and it's accepted 202 all that's done is: put it onto the parser.

It's not going to give you any feedback if it crashes. The parser is in memory right? If their KERI instance goes down at that point in time when you just got the 202 back, you have no idea that they didn't parse it at all.

They've never seen it essentially. For application reliability, we need retries on that. – When you're talking about “they”, you mean the other controllers agent maybe not even related to your agent other than it's on the same KERIA.

I was experiencing the need for retries that I didn't understand because I have a good network. But it could just so as a defensive programming when you get the 202 or other certain other areas, you need to retry. It's not necessarily a network error.

Yeah. Is that what you're saying? It's the KERIA is busy or dropped it. Yeah. On the 3902 port, the CESR endpoint of KERIA, it will just take in the CESR stream and put it onto the queue of the parser and give you a 202 back and then parse it. But if it gives you the 202 back and immediately crashes, you've now lost that data.

They have no idea that they need to parse something. They can't parse it. It's lost in memory. But the 3902 port isn't used by Signify-ts, right? It's I'm not talking about Signify-ts here. I'm talking about entire KERI here, but agent-to-agent communication.

But also from Signify clients. can use it if they want to? 3902 is open. Yeah. But Signify-ts doesn't do anything with it, out-of-the box, right? No, it doesn't out-of-the box. No. But also, if your agent accepts, an IPEX message, it should send it, right?

Especially for groups. For example, if you have a group set up based on KERIA and Signify Let's say, the one person proposes an IPX message, sends it to the others, it starts collecting signatures. Before it collects enough signatures, it goes down, comes back up, receives the other signatures, nothing happens. It doesn't send it to the other person. Because the granter deck is in memory, the queue.

It's lost it. And the lead of the group currently is the one that sends. A lot of the in-memory decks that we use for cues in KERI need to be made durable. Is there an issue? If there's not an issue with that, I'm going to make that one right now. That's That's bad. That one that one's bad, actually. That needs to be fixed soon. But you don't see these in your normal setups like because it's just "happy paths, " everything just works. Everyone clicks at the same time.

It's all good. Yeah. And it'd be nice to have some more feedback on the async stuff because everything in KERI is async. If we have better ways of receiving results that are more rich on what's happening, that would be useful. But that's a bit open-ended.

If you already have a plan to put these into issues, I don't want to duplicate it. But is there a plan to currently make an (GitHub, red) Issue on that because if there's not, I'll capture it immediately. I have a joint (GitHub, red) Discussion but I don't know if it has everything from this. It was written a while ago. It's either in Signify-ts or KERIA (GitHub repos, red).

that's some information from Sam on application layer reliability. If you want to read it, it's him saying that KERIPy is not going to provide this essentially. What is him saying again? About achieving application layer reliability end to end.

Okay, cool. And the need for retries and idempotency, so forth. Cool. Finally, well, not finally. Especially when you're using group multisig, and if you're doing something wrong, it can be quite hard to understand what you're doing wrong because it's a multi-agent setup. All your logs are intertwined. There's no way to filter on the logs. And even if there was a way to filter on the logs per agent, there's still too many logs, there's no way to filter on the action you were trying to do. And data again, everything's based on HIO.

Everything's async, everything's moving various ways with little observability. It's again fine, once you've developed it's fine, but sometimes it can be very difficult to add new features to KERIA and try and figure out where the data is actually moving and what's actually wrong and why it isn't completing and so forth.

Should the KERIA requests have a request ID added to them? Probably. But it'll need some work with KERIPy as well because most of the complex logic of KERIA is KERIpy. It's a direct dependency. It gets pushed onto KERIPy.

I need to do some work there. Some instrumentation would be useful at that. You see you go through the logs and you're just trying to figure out what's going on, because you've put the wrong ID somewhere, and it's ..., You can see Kent added some stuff here. But we probably need to go a bit further so we can actually filter on IDs of agents and so forth.

A structured log. Yeah. Request ID or client ID. Yeah. I'll do this one quite quickly. Probably KERI and Signify should have some semantic versioning because it's an API. And I think you raised issues on this. There should be compatibility enforcements between them.

I don't think this is big issue now, but in as you move towards being a more mature ecosystem and getting more developers in, it'll become more relevant. Also versioning. Right now you have to version on the KLI per agent and iterate over them.

There needs to be a migration framework in place. We've been avoiding this for ages because there hasn't been migrations, essentially, but when there is in production it would be problematic. That's resilience. And again these are traditional software problems Just put some resources in and it'll make it a more mature app It's not a protocol thing. These are all very achievable.

Finally on Scalability And this one is also a kind of developer experience I mentioned earlier that because a lot of the things are async, you need to submit like create an identifier, create a registry, issue a credential and you get an operation back. And then you need to pull on that status of the operation until it's done while it's waiting to collect signatures and so forth. That's kind of a pain.

Typical modern systems are usually event driven. Where you would get some notification that it has completed rather than constantly checking. The same goes for notifications. You have to constantly check if there's new new notifications.

In any general application like a wallet, it's going to have to constantly check because it doesn't know when to expect that you're receiving a credential. – Single-threaded environment, it could completely block the rest of the application.

Yeah. We've made event driven in our wallet in every application we've built because we've just put it in a corner in the box and made it event driven from the perspective of the rest of our application. But it's kind of annoying to have to build that every single time.

But this is very doable, if we were to take the query reply approach of just using the the WSGI response stream, we could open up the server ... connection per agent. Yep. To Signify and do this today. I might try that over the weekend.

Huh? No. Tell me about it. Before I get to that. The reason I put this in scalability is because right now you have 10,000 wallets using one KERI nstance and they have 25 pending operations that they're pulling, pulling, pulling. These are all full HTTP requests. It's way too much traffic hitting KERIA.

It won't work. Moving to something like Server Sent Events is much better from a developer experience, but it's also much better from a load experience. And it isn't..., I've never implemented them, so I don't know what it's like, but for notifications, it wouldn't be too difficult because it's just the next notification coming out on the event. It's more problematic for operations. I explain that in a moment.

I also Server Sent Events are true HTTP, so they're compatible with load balancers. If you start to do horizontal scaling and so forth, they're compatible, which is why Sam would prefer that over websockets because that isn't pure HTTP.

Now the problem for integrating with operations is currently when you perform an action that requires a long run operation to track that state of the operation. It just stores the related SAIDs and metadata that refer to that operation.

It doesn't have a status of pending or failed or anything like that. And when you use Signify to check that status of pending or failed, it's not stored in the database. That's what I mean. It doesn't flip in the database from pending to complete actually if you look at longridden.py in KERI, it takes all of those SAIDs and compares it against the database of the agent to see if it's effectively complete. Okay, it's not checking is this operation complete in the database.

check in based on this operations metadata are all the signatures in place for this to be considered complete for example you can see here for the delegations using the 'swain.complete' every time you check Basically like a diff on the database – This has been quite a mental challenge for me for like two years with KERI is the notion that the concept of state and state modeling is missing and that's intentional I think from Sam IPEX for example, right, is .. I tried to get a state machine diagram of all the possible state transitions and IPEX of Spurn, Grant, Accept, and all that and I had a challenge getting to the precision but this is similar: You're wanting to query the state of an operation and it has to be derived and has to be requeried to compute the state.

That is what I think I heard you say. Yeah. ...of an operation, right? It'd be nice just to get the state back from KERIA like 'Oh, it's in Pending'. Yeah. Basically while currently we can't do event driven with Server Sent Events.

To date? Because it's you're checking against the database state. Okay. Instead of checking the database state or the diff when this completes for the first time inside KERIPy, it needs to emit an event that gets sent on server the Server Sent Event thread.

Okay. Because I'm thinking what we could do for each operation type is that there could be there's already a server event connection from the agent to the Signify client. It would essentially just be a channel to communicate. I'm thinking on completion it would just send a completion.

Yeah. But there is no completion of an operation in KERIA. It's a state check. It tells you it's effectively complete. There's no point where it's in the database .., – Iit's inferred It's inferred. If you never ever connect it again, nothing changes on the database to say it's complete. It's just you coming back later and saying this is the status.

Okay. Because there's no point at which that happens This is a problem all the time, right? Essentially we need to store the completed state, not just derive it. All of this happens in KERIPy so KERIPy needs to tell us because this is a KERIA thing operations aren't in KERIPy – .. locally in KERIA per agent so because once we derive it once even though we're reading it yeah but what triggers to derive it other than you're going to move the polling inside internally – What triggers what?

What triggers 'Check and it's complete.' in KERIA. This is where I think what we should do when we have an operation There should be a Doer for each operation once it completes at the end before that Doer finishes it should write the complete state.

Yeah. But now you've just moved all of the polling internal to KERIA and you'd have so many doers for 10,000 Signified clients. It would be a lot better if internally in KERIPy It could just let us know when it's effectively completed because there's already something inside KERIPy tracking this I would say.

Then it can admit that event or whatever and we can capture it and update our operation in the database and send the event back and Server Sent Events. I think we're slightly talking past each other but I think I get what you're saying. I think fundamentally when an important event happens in KERIPy, it should be admitted saying "this and this happened."

Yeah. Rather than having new Doers that are also tracking it again. Cool, just one more slide. It's actually good. No, actually one; so not too bad. And horizontal scaling. People have asked about this before I would put this on the lowest priority.

because other things just need to be fixed first to get adoption. But for SEDI it would be important. Okay, right now you could do something like sticky sessions. It would be a very naive approach. Where you deploy 10 different KERIAs and you have a load balancer and once a client connects and gets routed to a KERIA instance they always have to go there, forever. That's like a sticky session. It would work now, but it wouldn't be great.

To do true horizontal scale, I think KERI is actually already quite suited to it. Because all the databases of agents are fully isolated and no agents are shared. They're just a bunch of LMDB files. If you had to ... [reformulates, red.]

The easiest thing you could do is have a shared volume in your cloud deployment that all of the instances can access and they would just go to the specific LMDB files that they're currently serving. And if it's done correctly they won't try to access the same LMDB file of someone else and you could have the agency coordinate that. If you wanted more isolation, I guess you could separate the volumes, but I think that would be a little bit complex, not necessary.

You'd be moving data across volumes. I don't like that for now. At least I don't think it's necessary. Unless there's a legal constraint on data or something. The agency can become the load balancer, like it currently is basically for the message router in KERIA. But instead it would actually be outside KERIA and route into a more light-weight KERIA. You kind of decouple it a bit; be different services.

I think that's more down the line. That's why developer experience and operational resilience are highest priority, I think. Done! – What's this up there? That's the file system. The agency database is an LMDB database. You can see that there was two new agents booted. There are the config files that get copied in the JSON config files with the OOBIs.

And for each agent they have a subdirectory in all of these directories. Like there's the key store, for example. Probably it would be better if you restructure it, so that the agent is at the top level and it has its databases internally so you can more easily take it. But if it was a shared volume across many instances it wouldn't matter.

TALK XII

Fergal-O'Connor

KERI Conference April 2026 · interviews · 01:57

– Hello. My name is Fervil O'Connor. I'm the Identity Solutions Lead Architects at the KERON Foundation. Over the last few years, I've been helping build out the open source identity solution known as Beredium. Today, I gave a talk on the KERIA and Signify TS security architecture. We delved into operationally how KERIA works to be a cloud agent in… I should go back there. We delved into how KERIA and Signify TS can enable stateless signing at the edge both operationally, we delved into some architecture diagrams, and also from a trust relationship point of view, delved into which keys to the where, who signs what, and so forth, and how you can use KERIA and Signify to interact with the entire world. I think one thing that we do differently is crunch. We're trying to bring KERIA to the masses. It's essentially a set of cybersecurity protocols that are very low level. What I've been trying to do with KERIA is to take low level KERIA and make it accessible to individuals and organizations alike in a way that's super streamlined and highly accessible. I think my talk today on KERIA and Signify kind of relates to that in the sense that Signify can provide a higher level API to developers as more accessible. I If KERIA and Signify were to disappear everything we're doing is built on open protocols. You don't have open protocols. You don't have any lock-in to particular software stacks. I'm sure the community would find other ways and there already are existing other ways to create your KERIA identifiers, interact with the world. This is just one particular implementation and one flavor of that. If it disappears we would see different flavors popping up from different organizations and different open source communities. This is how sovereign at the end of the day.

TALK XIII

How KERI Enables Better Private-Public Partnerships

George McEwan

KERI Conference April 2026 · talks · 50:58

SUMMARY

AI generated summary, if a statement needs explication, see the video.

George McEwan recounts decades in healthcare IT, from enterprise patient matching to national health alliances, and argues unreliable identity resolution remains the core problem. He describes how DirectTrust and open dialogue with KERI, mDoc, and W3C communities build trust for honest public-private partnership on digital identity.

While we let Chris find a chair, I'm going to go ahead and start so we can stay on time. One of the things I want to ask the panel and each one of you will get an opportunity to answer this question. What I want to know is if you in 2 minutes or less, could you tell us a little about your background and what was the specific lightbulb moment that drove your passion in trying to fix digital identity?

I don't think this is on I turned it on, but it didn't connect anything. This, however, does. My background is actually kind of a bizarre. I started [inaudible] about an hour, so we're not going to do that. but I think my healthcare IT background began actually at the very beginning of the exercise we're all involved in, right? My first health… my first job actually was in 1986, which is when the 8088 was the processor du jour. And fast forward about 10 years from that point when I went through a couple of life cycles of career, I ended up being transferred back and ended up in the healthcare IT space. Actually got the… My first healthcare IT job was actually the bass player in my band got me that job, so but actually my original notion of why I identity is important is really far back. It is really actually back to the notion of my first implementation of an EMPI, if you know what that is. This is the notion of a an enterprise master patient index. The idea that you would know who you were talking to reliably at that point was a question of probabilism with very little assuredly about who those people were in any way.

Pretty early on I realized that the quality of the data that was being collected to establish this was poor. That their ability that actually the organization I was working with at the time was Kaiser Permanente Northern California. They had a about a 30% duplication rate in their system. Your ability to actually resolve identity was the first thing that was my light bulb moment. My next big opportunity then was like, I don't know, 20 some years later when I was responsible for a national network deployment that was the Commonwealth Health Alliance. And again, when that first launched, its focus was not on healthcare information exchange, but on identity first. Because the expectation was this was the problem you had to solve. You had to figure out who you were talking about at this point and that point. What I quickly learned is that we were still talking probabilism and we were still talking 20% duplication some 20 years later. I mean, this stuff had not gotten even a little bit better.

The notion that identifiers would be important, I think for me really began to click when we couldn't resolve identity. Getting to identity assurance wasn't something I was thinking about. I was thinking about the fact you just couldn't match people or records with people. The my light bulb moment really was around identity resolution and that's really my was my raison d'être. But then, now in this setting, what we're trying to figure out is how to make it possible for us to be sure that people are who they say they are when they're contracted over the internet. This is very different problem than identity resolution, but it has the same core problem, which is that people go through processes to do this exercise, this initial exercise. The initial exercise is the first transaction. The first transaction is the failure transaction. That is the fact. If you can't get the first transaction right, all other transactions beyond that will fail.

This was my big aha moment. I've been saying the first transaction for about a decade. I think it's the most important transaction of all. And if you can't get the right stuff in the door the first time, then you're trying to match to it, all bets are off. Not going to work.

That to me was my light bulb moment. It's maybe, I don't know, 1993. It's a good time to have it. We're just going to do this like a slow stadium wave. We'll just keep rolling on down. Okay, well, I'll give it a try. I suppose that light bulb moment for me probably came back in 2015. I'd started with CMS. We were working out how we were going to make data available to patients, to beneficiaries. And it was going to be done via third-party applications, health applications that we're seeing pretty typically these days.

And I just asked this the really the simple question is there a way that we could automate the registration? But if we're going to do that, I really need to be able to have some confidence that the organization that is presenting themselves is who they say they are.

I started this the way I seem to do quite often. I wrote a blog. It happened to be for HHS, and it simply asked that question. And that led to conversations. And I think, Scott, we even had conversations with the DirectTrust around is there a way I could call an API and check with you guys whether you know this organization. That really evolved from there more directly around healthKERI.

Back last year, again, I wrote an article because I'm thinking how do we operationalize these APIs that a thousand health plans are going to have to implement and tens of thousands of provider organizations and I don't want to do all this work on defining what these APIs are and have them sit there and do nothing because that doesn't benefit the patient whatsoever. How can you operationalize that?

And realizing that the manual methods we used for patient access API which takes days, weeks to go through is totally non-scalable. I asked the question how can we have an organizational identity that we can check? And that sort of led me down the path to KERI.

Enough said. Coming at it from a completely different angle. My background is in financial services, particularly in transaction services, which at the time that I started wasn't a very common or what would I say attractive part of financial services to go into.

But I've spent most of my career around identifiers and identification. Which is quite different than identity. Although a good foundation and a good basis for the way that my career at GLEIF and my involvement in this community has has led to.

come the financial crisis as I explained before, my background in standards led me to be involved in the development of the LEI, of course, another identifier. And then it was its usage, especially guided by its initial usage in order to solve an initial problem. We want to know who these organizations are who are responsible for a particular function in a particular context, just to generalize it, basically.

And then as we started to to operate for a few years, in our strategic discussions at GLEIF, we started to talk about we have a high assurance identifier that's global in nature, that can identify a very broad list of entity legal forms or legal entities in effect. Because of the effort that we put into, or our partners put into validating this identifier and the effort that GLEIF puts in managing the global LEI system and the quality of that system and its operations, why don't we look forward in order to be able to see how we could make this relevant in the digital world, which is the way that the world is going?

This is when we first started to get involved with the identity community and I'll call it the identity industry, which of course I had no idea up until this particular point even was a thing. Luckily we were able to get connected with and be welcomed into the community by many of the experts that we see

here today, who introduced us to the importance of identity and not only identification but verification, which of course led us to wanting to be able to develop the vLEI and then use the vLEI.

I have to say that it was an eye-opener for me. My aha moment was actually not as I was day-to-day trying to be able to figure out delivering on the strategy that my CEO had given me in order to be able to say, "Okay, let's make a verifiable credential containing the LEI and let's make it a success."

It was how clueless I was to how I was exposing my own identity every day in in the everyday things that I did just as a natural person and a citizen and how I naturally accepted anything that people told me was secure and was protecting my identity. I took it just as face value.

Scott, you said your lightbulb moment came in 1993. I think mine came at the exact same time. I was two. At and I'm a little bit tongue-in-cheek, but the truth of the matter is that for my generation we were born into… Now, I do remember when we had dial-up installed in the house. I'm not that young. But in a real sense, I grew up in the digital world. Like we had digital identity before we had a term for a digital identity. And we just accepted, you In the halcyon days of yore MySpace was the hot snot and there was none of the concerns that we had around it. But for my generation, there's never really been a demarcation between my identity in the real sense in the real world and my identity in the digital world.

For me it was I don't know that I ever had a lightbulb moment. It was a lifetime of experiences where you XYZ got hacked, your credit card gets stolen, your know, all of these things and it wasn't until I started to understand some of the things that were possible with digital identity that I went back to this whole life that I'd lived and said, there has to be a better way. We have to do something different than what's being done now. And I mentioned this earlier, but, I guess, the coup de grâce on that system was when I just realized the fragility of the system we've built off of poor identity structure and how that could impact me as a patient.

I feel like a baby compared to this entire group. I'm a newcomer here. I think I had three things that have got me to this current point that have I guess made me a part of the identity movement. When I first got into my role as chief privacy officer, it was one of the it was the first time the state had one. And I got calls all the time from people that I would sit on the phone for I think the longest one was probably two or three hours where I felt like I was like a psychiatrist of they're just saying, I've had this horrible thing happen and I have no control.

And I've had cases where women would call and say, my psychiatrist literally like gave my data to my ex-husband and now it's being used against me in court or my data's been breached and XYZ has happened. And that made me realize: 1: people had absolutely no control and 2: they had nowhere to go. Their data's been lost and nobody was listening to them.

The next thing that built upon that was George and I had offices close together for years and between George and the rest of the team, we knew government had a big problem. If you don't know the problems in government right now, there is really no data governance and with what's happening with technology and the transition the last 30 years, what used to be due process rights and ensuring you had processes that were not arbitrary, that were not capricious, that eliminated bias, that is not what you have now with technology. It became very clear that if we truly believe that the basis of government came from the individual, and then they're the ones authorizing government to do what they do, but government is now just doing whatever, you have got to have some anchor of trust. And that was where my first constitutional basis came from of it's the individual. And that's why I hold so strong to that in the SEDI approach. And then lastly has come of as we work to redesign government, not just SEDI as a point at which you engage, but what do all the other government processes look like for identity, for

organizational identity, to have end-to-end verifiability, to have that provenance, and to identify, where is automated decision-making going to fit in and these artifacts government has to create so you can hold them accountable, you can't do any of that without identity to know who are the accountable parties and who's been wronged. All of that together last five years has for me created a, essentially, an undeniable truth of we have to fix identity or we are not going to have a trusted government in the future.

As you can see we have a very passionate, well-qualified group to talk about the elephant in all of its various facets. What I want to do for the about the next 15 minutes is I want to talk a little bit about the topic area. We're going to start with Chris for the government perspective and then anyone on the panel jump in to add additional perspective. We can acknowledge that public and private partnerships are not new. Historically their value is questioned and often debated. We don't know if they deliver on what they say they're going to deliver on. Describe for us, Chris, how you view SEDI's engagement with the KERI community and how that's going to benefit all the parties involved.

Does anybody know how we started the engagement with the KERI community? How many actually know that inception event? Okay. Let me tell you this. It started with, Sam and Timothy and Mike Lahey, sat us down in a room with Alan Fuller and myself and a couple others and they were like, "We have the solution for you."

And we were like, "You are so full of it." But just like I think you'll find the whole Utah team, we are willing to sit down and listen to anyone. If you have a good idea, it doesn't matter who you are, we sit down and listen to everyone. We listened and we listened and before long, It was a “We think there's something here”. You learn things that you can't deny.

That passion from the the KERI community and their persistence in bringing facts to the table, we've been very fact-driven. That's what's impressed me is you come to the table, you bring facts, you've educated us cuz we weren't experts, we verify and validate on our end, but then we've created this reciprocal relationship of what are the facts, let's have an honest discussion. I can assure you every time we meet with any of you, we bring in skeptics as well to challenge it and we've had meetings where we have Sam, we had Phil Windley and we had Steve McConnell all in a room kind of battling out in a way to get to first principle truths and we've gotten to the point now where I feel very comfortable in just having honest conversations and we've said everyone else, bring mDoc, bring W3C, let's have that conversation of how it fits together and to me that's resulted in how we have engaged with the KERI community and then the basis of how we're engaging going forward of just what are the problems, what are the facts.

And then because of that trust built, I feel loyalty because it was partly created in Utah to help that community thrive cuz there's not many like it. And if we can scale that and that type of engagement and truth finding, you can do really amazing things when you have passionate people.

Anyone else want to jump in? DirectTrust as an organization was born out of a groundswell of basically volunteer effort initially if just in fact, I don't know if you were involved in the Direct Project initially or not, but watching closely. The Direct Project was like 150 different organizations and individuals who were involved in this exercise.

The idea was we need to be able to do secure identity assured communication in healthcare. That was the goal, right? That was the goal. This is 2012. This is coming out of the HITECH Act, which is basically legislation that says all of the electronic health record companies are going to get a big boon as all of

the doctors can basically get paid to implement healthcare systems to move from paper, to actual electronic systems.

And that all just really happened very quickly. The Direct Project was born, kind of, in that crucible, but it was a public private partnership. The first thing that we did was they they actually were stood up within the ONC as the as a part of the national coordinator's work. And then the national coordinator gave a couple of years of support economic support to actually stand up the organization. Since 2014, DirectTrust has had no economic support from the federal government. But that short period of time where we had federal relationships, actually the relationships have continued, but the economic relationship has not continued.

We do have one aspect today, interestingly, we've talked about this UDAP exercise. We're talking about the Root of Trust that's associated with that is a cryptographic Root of Trust that is an X.509 based PKI framework. And let me tell you the challenge that we have with it. It is not extraterritorial.

It's bound to TEFCA. It is invented and used intended only be used in the TEFCA area. The opportunity with this eLEI exercise for us is to take something that is good anywhere it's presented for the same benefit of identity assurance that a PKI framework can be used for.

We're intensely interested in this exercise and very excited about an ongoing relationship with GLEIF and also hopefully with SEDI. You had Ryan Hamilton here yesterday and he was one of the very early organizers, along with people like Aneesh Chopra, around the CARIN Alliance. That has now also evolved into… It's an accelerator within HL7. And I think HL7 itself, those accelerators are great example of public-private partnership. Really what happens… I'm retained by HL7 to help write some of those standards for the DaVinci project, for the accelerator. We're focused around payer-to-provider. I've written payer data exchange. What you see happening there is CMS is watching. In fact, CMS has a contract out there where there's a whole bunch of contractors that they have embedded into HL7 to help write implementation guides to help develop testing protocols, to help develop the educational materials, because the secret is Congress will go and set legislation, they'll write a law.

The agencies then have to regulate. You can't regulate what you can't point at. They really need to come to organizations like HL7 and say let's help you get those standards written so we can point to them to then get people to implement. This it's been a really it's almost an early warning signal as to what are they going to regulate on next because you see where the pressure is on to get something published.

– I just wanted to add that GLEIF is an ongoing part public-private partnership. We are under regulatory oversight and continue to be throughout our entire existence. And the engagement with the public sector being a private sector individual and myself started long before that, actually in my my standards role at the international level.

And I think that being involved in public-private partnerships and making them work that it's important in order to be able to realize that bringing viewpoints and sensitivities and important points together from the private sector and the public sector actually produces better conversation, better discussion, better debate, and hopefully in the end better solutions than if it was a completely one-sided discussion. Also, it's sensitized me over the years that I need to believe in exactly what I'm putting forward and I need to speak facts and I need to understand what I'm talking about and it really sold us at GLEIF in order to be able to decide to develop the vLEI when the people who approached us and said that there's a better way than the traditional way of being able to do this were actually the people who had built the previous solution and were ready to move on.

– I would maybe just add a quick thought here which is that with regards to public private partnership and I think we'll get into this from a healthcare perspective. I've seen what happens when the public part of that partnership goes awry and they just run at something on their own and they don't enact a partnership with the private market. I think we've also seen the alternative to that where the private market starts to dictate terms and I think in both cases it is better if you get all of those parties aligned. I think that's one of the powerful things around doing things like KERI that is so open source, open standard. It invites the collaboration and invites the participation and it actually defends both sides from being able to just run amok.

– Thank you for all those answers. One of the things that I've noticed with this public private partnership approach that we're taking is there's not a person we don't talk to who realizes they're ultimately going to be the beneficiary of some of this technology. We really appreciate the fact that everyone's putting forward the effort to say well wait a minute.

Ultimately this is going to impact me. How do we do it the best way and that open discussion and that dialogue has been super helpful for us. Well we are at KERI council, let's take this directly to a KERI question and this one I'm really interested to see where everyone goes on it. Regarding the healthcare industry what is the biggest problem you see KERI solving in the time horizon of the next 5 years.

Anyone jump in. – Why don't I go first? We have to solve organizational identity and the thing I see around digital identity particularly for individuals is it's all about context. Right? You've got to be able to tie those threads together.

And that's where I see how we can implement the vLEI, the ACDCs, these here. All of the this technology to basically say I'm part of this organization. I'm allowed to do these things. This is the type of transaction I want to be able to take on.

I think some people already said that. We exist in many different facets. I'm in Utah in the winter, I'm a ski instructor. I have a an employee badge for that and I also have one for my day job. They're totally different, right? And give me different permissions based on the context of am I working there or not. We have to get this problem of being able to clearly identify and prove and be able to root out bad actors at the organizational level and relate it to individual identity as well.

Can I interject just really quickly? 73% of all cyber attacks that happen in the US healthcare ecosystem come from third parties first. The bad actors, one, we're the most targeted industry. Two, they know exactly how to get in and that's through the insecurity of the identities established between parties in those connections. I think if there is anything that KERI is going to do in the next 5 years, it's going to take that 73% and plummet it to the ground. I think we'll be able to solve the largest problems in that connectivity. – And something like 60% of providers that are hit by a ransomware attack go out of business. – I can tell you also just from… We also accredit healthcare organizations for their cybersecurity capabilities. And as we review the activities of folks in post-breach world, what happens what you find out is most of the breaches are caused by the use of a credential that is actually a valid credential. You're talking about people who are using someone else's credential for nefarious purposes. And once they get in that initial door, they're able to then use AI, for example, to get to traverse the entire network once they're inside. From my perspective, being able to move the entire healthcare ecosystem from weak credentials to a model that is actually password-free would be the biggest thing we could do for the workforce in healthcare. I think that would be the best outcome.

I also think that in general that the world needs a veracity engine and we don't have one. There needs to be a veracity engine. When I appear online, when I appear online in any context, like when I'm talking to you over Zoom, you should be able to see who I am, that I am who I say I am, reliably. And that should be, I think, and that's if I want my identity to be shared. And like these things I think are so important to establish. I think the policy focus is really important though. If the policy doesn't happen right, I think this is what Utah has done so well.

first principles, policy first. I think that has a huge impact. – I'm going to give a slightly different answer here. It's going to be controversial. This is from the policy perspective of what needs to be done in 5 years to actually get lots of money put into the ecosystem and to align interest with policy makers to get them to run legislation and actually get the market going cuz we've had breaches and security and lack of trusted identity and provenance for decades and it hasn't fixed the issue. What is the one topic everybody's talking about now and pumping money into and it's AI?

If you want to solve all of these problems, you've got to tie it to some AI use case which is where all the money is going. Utah did something really unique and this is just for everyone to think about as you're working on the problems.

They have been in national news because they did the first automated prescription issuance from AI, coming out of our regulatory sandbox. Utah's created some really unique things to allow AI to be used. We know in this room what KERI and ACDC is going to bring to the AI ecosystem in terms of provenance and then associated the data or to AI agents having being delegated with authority to act on somebody's behalf.

I think it's critical to tie in some of these unique capabilities to AI and not show that we're reducing fraud or losses but to show some efficiency gains or things that directly impact individuals. If you can start saying because we have proven, verifiable delegation for this agent acting on another doctor's behalf to start streamlining prescription issuance and lowering cost, that's going to get people excited. And if we can prove the use of this technology over the next, I don't think it's 5 years, I think it's the next 3 years, that is going to start turbocharging the digital identity and verifiability approach.

– Okay, who in the room just registered KERI.ai? Did that just happen? Okay. Jared, go ahead. – I agree with what Chris is saying and really my thought there is that this is a problem that can't wait and this is a problem that people should be thinking about and working on already. And what I guess what I'll say is while we haven't moved some of the work we're doing at Health KERI around AI into open source, I We probably won't in the near term, we definitely want to entertain conversations with folks that look at that and see it as a problem the same way Chris does.

– And I will say to do what I just said, you already have to do everything you just said. It's just what's the story you're going to tell and that's an amazing story to tell that will get people to put more money into it.

– Karla, I'm going to lean on your organizational experience that you've had. Given what you've seen in creating verifiable organizational identity, what do you anticipate is going to be the practical friction or challenges that healthcare organizations are going to encounter going through this process?

And then, panel, listen up. My question to you would be is how do we help mitigate those issues? To begin with, I think that the first thing is that people are used to doing things, or not doing anything at all, or doing things a different way. There's going to need to be a change in behavior.

And the reason why the change in behavior is going to need to be explained to them. And that's a challenge for us. Not to get into the technical details. To focus on the user experience and what the benefits are in order to be able to target whether it's doctors, whether it's patients, whether it's payers and the plans. As we've been speaking with healthKERI and we've been speaking with DirectTrust.

I'm learning more and more about the healthcare industry and the fact that there are data inconsistencies, there are efficiencies that need to be exploited and identity and verifiable identity is at the core of all of these things. And for us to be able to leverage what we've already created and to see this applied in this way, to be able to take a problem and say, we can look at our process that we have here and we can explain to our members that they're going to want to see their credentials registered in the provider registry so that it reflects the real situation in the world, I think is an improvement. In order to be able to to also apply the the credentials to the plans.

You know who is associated with what plan and also to just disambiguate the providers themselves. I go back to the example that Scott gives to say which St. Mary's is it. And we had I guess the answer kind of like sitting there in the LEI repository and we are glad in order to be able to know that there is another industry and another domain that can benefit from the same kinds of principles that we wanted to promote in the first place.

– I did want to say, you asked the question about barriers, like so what's going to keep us from being successful? I think I'm new to KERI. I think KERI is incredibly interesting and exciting, and I'm a kind of a propeller head, but not not my propeller is kind of just drawn in crayon. But I what I've got to tell you is that that KERI's going to have the same challenge that the X.509 community has, which is y'all speak Eldish.

You're going to have to translate this into English, and really quickly, because if you're going to try and bring this to market, we're going to have to start using words that people understand. And we're going to have to de-acronymify because in this day and age, I can tell you, we merged with an organization called ENACT, and when we did, we went and we de-acronymified everything, cuz all their programs were four-letter acronyms. I'm telling you, this is really not a 2026 kind of a thing. That's a 1995 thing, right? Acronyms are big in technical space, because you can't do these things without them, because you can't describe something technical in short words. Germans, of course, just put words together. But I'm telling you that this is, I think, our biggest barrier.

What I heard some people explain here, actually, I think it was a legislator, it's hard to explain. And I think that's probably our biggest barrier is that we have to be able to describe it very succinctly, and with value proposition in mind, and with words that people understand. If there's a barrier here, that's the biggest one.

– I agree. I think clarity in language is the biggest hurdle. If we can get to clarity of language, we solve a lot of problems. I think the second biggest hurdle, And then the solution, honestly, is money. And I wish that wasn't the case.

I wish we could talk about healthcare being this folderol, and everybody wants to be for the patient. The reality is this industry is not going to move unless there's money attached to it. And that comes in two forms. One is if companies like HealthKERI can come and improve out a model that KERI drives dollars for companies like us, then a lot of different ecosystem players are going to come in. It's going to be new startups competing with us on our level. It's going to be people that are established in the space that want to come in and start doing what we're doing. But if we can show money, we can move the market.

The other side of this is for the customers, the groups that will be paying for these new services, they need to see money as well. For them likely, they will see that in two forms. It's going to be a reduction of cost or an increase in business. If they can prove that they're more trustworthy, they can capture more customers. If they can do this automatically, they will save on COGS. I think it's both the biggest barrier and if we can solve it, we'll be the biggest activator because people will forgive your language if there's a big pile of money behind your acronyms.

– One industry that does not get rid of their acronyms is the legal industry and those terms and conditions don't get any smaller and I do think we can move the industry and I said this at the SEDI summit is what's the regulatory framework to incentivize all these experts, technologists, and the legal departments to move this way and so some of our tools would be like the reciprocal verifiability. You want organizational identity, we're going to require them to do reciprocal verifiability. Who is the org? Who is the individual? That sounds like well, it's a cost. Who's going to do this then? But then we say, well, if you do this, here's the limitations now on liability. Here's the safe harbor to function and we give them the incentives to move toward reciprocal verifiability and to use tools that have the capabilities KERI provides and that should take away enough liability or provide enough incentive for them to at least start in their transformation to move this direction.

And especially when we start tying that to AI, I think this is a perfect example. We say, "Your AI agent must have some verifiable form to say that they're acting on behalf of the entity and who provided that delegation." And if you do this, then you can be in the sandbox in Utah. Those are the incentives that I think regulators can help with to move things.

I want to pull a few threads together here. The yes, we need to speak in very plain language to folks as to what we're trying to do. The problem is yes, legal and government love their acronyms. We've got to also meet them there.

What we need to do is really think about don't make it too complicated. And also try and tie it back to things they understand. I jokingly tell a story about how I actually got blue Button Built at CMS and got it out through ATO, their authorization to operate, so that it went out and into the marketplace.

And I say, "I basically designed the project so nobody had to make a decision, right? That you always could point to precedent so that they could say, 'You've already done that.'" And if you were listening to what we were saying this morning, a lot of what we've done there is to say, "This is just a small change on top of what you're doing already. It's the same type of technology. It's just a different cryptographic key, right?"

that you help them make the intellectual jump. – One other thing to consider is what is the use case you're building for? We've had many verifiable credential companies come in from KERI and every space and they're selling a product for a use case which is not the low-hanging fruit that's needed. There are many entities with really smart people if you simplify what you're doing enough but hit that right use case and there's some out there. This isn't the healthcare industry but we had Maverick in here on the first two days of the SEDI summit and they actually said, "Hey, if you want, we've already made SEDI legal for purchase of alcohol and tobacco." And they said, "If you have the implementation guide, we could have this implemented in 6 months and we'll start accepting SEDI." And that's cuz there is a product market fit there that perfectly aligned with the protocol we're looking at for SEDI.

What are the right healthcare use cases as well when and we don't need the whole platform. I think it's what's that to get you in the door to show value and if you do that, then you can build from there. Fantastic point and you're right. This is all tying together. I love when a panel just leads towards that

natural conclusion.

Let's talk about the ultimate stakeholder in the healthcare industry and that's the patient. We know that the trust is broken in terms of the personal data and the protections around it. We say HIPAA a lot but that doesn't mean anything when you're lying in the hospital bed and we do have some representation on here from people who are in government or who've done some pretty remarkable things with government regulation trying to help the industry move forward. If you can give me your closing remarks individually and if you had the magic wand that you could change one regulation or add something at a state level or a federal level that could really kickstart everything in the right direction, what would that be?

– I'll tell CMS is the is the organization that has the biggest hammer. I mean their hammer is actually is it a spoon with sugar in it or is it hammer? I don't know which, but it's very, very, very authoritative and powerful. Their two opportunities are reimbursement and regulation. When they regulate, if for instance, they were to require provider organizations to all have vLEIs in order to get into the national directory, it would be an instant driver.

And this is a national directory they're trying to build that is failing on the subject of organizational identity. This would be a huge opportunity for them. They haven't learned to say GLEIF yet. They call it GLEEF. They are confused. I I'm trying to bring them forward and say once they can pronounce it, I'm hoping that they can also support it.

– It's a tough one. I I think I've moved beyond this, but back when I was with CMS, sort of 2017, 2018, there was a lot of talk about Medicare for all. And I was saying, "No, no. Why don't we just start with Medicare registration for all? That you got a standard ID that you could take." And I I've sort of moved beyond that to say what I would love to see is probably regulation that says, "If I as a patient present an identifier or an identity to a practitioner, to an organization in healthcare, you should damn well put that into my record.

As they start sharing data and you go to more and more places, you can actually patient match. I had a colleague in the ideal lab at CMS who was there to do patient matching. You'll know who he is, if he ever sees the video.

Every time I went to see him, I'd say patient matching is so much easier when you involve the patient. – I haven't become a healthcare expert overnight. I'm going to take a little bit of a different tack here. I would like to be able to see any law going forward that involves identity in order to be able to come at it from what are you trying to accomplish? What benefits are you trying to gain? What features then do you need in order to be able to have in a solution?

Or to have the law be principle-based. If you have need for to be able to protect for example children or elderly parents, do you need delegation? Are you trying to be able to instantiate or give credentials in order to be able to know the real situation of authority within a company in the area of organizational identity then you're going to need a feature in your in your identity solution that gives you that. What level of assurance to know that it's really the person who is supposed to be able to be authorized to get that credential? Do they have that role? What level of security am I comfortable with?

Do I need automated verification? Those types of things. And when you answer those questions the law then gives you a guideline or a road map in order to be able to pick a good solution. And for me, I think that changing the dynamic of being able to come at the law in that way would be an advantage.

– I've been looking around the room and I'm pretty certain that our friends from Libertas are no longer here. I think I'm safe to say this without getting a water bottle checked to my head. I would at this point if I could wave a magic wand I would get rid of section 510 from the HHS labor bill. That section of that law made it essentially illegal for the federal government to issue a national patient identifier.

And I agreed with all of the reasons that law was in place in terms of privacy and the concerns for patients up until very recently with some of the work that's been happening with SEDI and other initiatives. Now today knowing that we have a way to do that national patient identifier that is privacy preserving, that manages to keep the patient in charge of themselves digitally, I think that should be repealed immediately. And again, if Connor or Jason from Libertas see this, we can talk later about the implications.

– This is an unfair question for me. – What would you like to do? – Cuz there's this bad thing that happens when I say what we want to do in regulation then I have to make recommendations. It ends up happening. Now some of this I've said big picture we know privacy and security is broken in the country. The amount of correlation happening with your data and how that profiling and the scoring and predictive analytics is occurring in every sector, private sector, public sector, healthcare, social services, education. You would be amazed in this room and me and George have seen these systems. And so I'm coming at it from a different perspective of what is coming from the private sector industry and how that would be used to exploit for monetary or the public sector side to try to influence you to make different decisions.

I had seven years of experience in privacy and security with human services in Utah and I know HIPAA does not work and the levels of non-compliance for both privacy and security. There's maximum use of data, but these other things are viewed as just hindrances or there's not the expertise to do.

And I think this was alluded to on the legislative panel on the SEDI summit. What I am working toward is I think there needs to be a total rewrite of the approach. Our goal is that starting with identity end-to-end verifiability and zero trust.

Move toward the duty of loyalty model where anybody with your data should be acting in your best interest and this crosses all industry. This crosses healthcare, this crosses financial. And if you're aligned with that end-to-end verifiability, we modernize the laws for provenance requirements, we figure out how to track these terms, Then we will be creating also some extreme safe harbors for entities to act… to work with knowing mistakes are going to happen, but I think that will create a more balanced free market and set us up for coming in the age of of AI and automation, but anything absent… a total overhaul for both public and the consumer side, it's going to be a very fragmented disparate market and the public is going to be confused and businesses are going to struggle to maintain our economic dominance that we have here in the US.

– Since Chris pulled me in by saying I'd seen some of this stuff, too, I'm just going to say, yes, that's true. I It's there have been many times when we've effectively done our architecture reviews and opened up a system just to find that it has a digital hamster wheel still spinning and we have needed to change and some of that's going to be in behavior, as we've talked about, and the incentives need to be there to help us with the change.

As we started this day out, Sam had made a bold statement that this is the potentially the year of KERI. I think from what we've heard from this panel, that's been corroborated. Sam, put one more in the win column. Can we have a round of applause for our panel?

Thank you. [applause]

TALK XIV

The Landscape of SEDI

George McEwan

KERI Conference April 2026 · talks · 44:35

SUMMARY

AI generated summary, if a statement needs explication, see the video.

As Utah's privacy architect, George McEwan outlines the SEDI program's split between government frameworks and private-sector control. He frames digital credentials against constitutional rights, warns that unregulated markets erode accountability, and stresses delegation and guardianship as critical relationships KERI infrastructure must support.

Everyone, I'm George McEwan. I'm the privacy architect for the state of Utah. There's a lot of synchronicity that happens when you go to these various events. And amazingly enough, I'm going to talk a little bit about healthcare in the rural healthcare setting and why we look at it the way we do. One of the things I'm going to make very clear is that the state is going to declare some boundaries in what it does in the KERI infrastructure and what the expectations are for the community because one of the things we learned as I was the IT director during the COVID era at the department of health, there was definitely some resistance to where government had offered the technologies or applications or worked with platform providers to provide things like exposure notification and the backlash was very intense. That's probably the best word I can use.

We realize that where government can help and we can set regulations and we can set frameworks in play, there's definitely a spot where the government has an edge and we say this is private sector and for the individual to handle. What I'm going to do is I'm going to run you through a little bit of a presentation that we do with we'll call them non-nerds, normal people.

Because we can talk in shorthand in the SEDI community and in the KERI community and cyber security. We have words they don't understand. What we did is we put together kind of a generic presentation that we add slides to depending on who we're talking to.

Realistically, we hit them on these points. Digital identity is here. It's coming. It's going to be something. Now, one of the, I would say, the beauties of the SEDI program is Chris Bramwell said, "We could take this and put this against what are the constitutional rights that someone has."

And by doing so, you get past all of the vendor needs to make money. You get you get right down to the root of the problem, which is there needs to be accountability and that your rights need to be preserved and that privacy is not impossible when you are talking about digitally issued credentials. And of course then we get into the stakes.

If the government doesn't choose or regulate, the market does it itself. And the goals of a open market are a little bit different than what a government needs to be providing the services that a citizen would want. And of course, we know that besides governments, there's also organizations that put together specifications that may or may not be rooted in the same value system that we're trying to maintain. We're very adamant that we want to make sure that whatever we do in the SEDI environment, it's reflective of the values, goals, and rights that are held here in America. Now, it is not hard for this statement to be believed by all of you, right? Whoever has the data has the control. And the model right now, everything to the left or sorry, my right everything over there is how we work today. Your identity is owned by other individuals. It's granted back to you with limited rights. And then if you've ever looked through all those long user licenses, in effect, you've given up all control for the services that you have.

And this isn't limited to just, hey, I got myself an email address for free. Your financial institutions do this, your hospitals and your HMOs, your care organizations, they're holding a lot of the data that belongs to you. And because it's set up that way, you are basically powerless to do anything about it. And in a lot of cases, you'll find that incorrect data may be used in profiling you to create an improper picture of who you are.

Really what we want to accomplish inside the SEDI framework is the idea that you are the individual who controls that information because wherever the information sits, it is fair to say without contradiction that impacts the financial outcomes. It impacts your opportunities. There are a lot of things that get impacted because you don't control what should rightfully be yours. And this is not to say that everyone did it for a nefarious purpose. Profit-seeking motive.

We're a capitalist society. That's understood. But the problem is we built everything to be so exploitative of you that we forgot that you are the individual who ultimately gets hurt if something goes wrong. In the state of Utah, inside government, there's a very good way that we look at data. When we make a data error in government, it's a negative outcome for a real person. Classic example, oh the Medicaid eligibility system, which is a real time system, goes down for a couple hours. Now, if you're standard IT, you're like, big deal. Things happen.

We'll get it back up. No big deal. But then when the Medicaid director marches in with their flesh hook and rips a pound off of you and says, "Right now, somewhere in Utah, there is a single mother who took two buses two hours out of their day to go to the pharmacy to get medication for pain management." And the pharmacy directors at that facility said, "We can give you one pill because we can't confirm your eligibility. You'll need to come back."

That's the sort of situation we deal with when we talk about our data errors. And of course, we try very hard in government to be good stewards of that data. But realistically, if the individual has a lot more control, who's a better advocate than you? You are definitely going to be your best advocate.

Now, when we get into the concepts of state endorsed digital identity, we really try to keep this as simple as possible. And this is probably way too high. And everyone's like, "I get this slide immediately." But what we try and explain to them is those core concepts that we put into the bill of rights that identity is yours.

It's not issued by the state, but the state has a role to endorse it for certain legal purposes. And of course, we get into the concept of what is a digital identity? How does it get created a little bit and then we go into the whole idea of how it becomes your state endorsed digital identity through this process of vetting. Now there are some potentially debatable questions on here. We don't get into the politics of this, but we do know that there are some markers that need to uniquely identify you to be part of that digital identity. And this isn't to say that these are forever attributes. For example, your address, your biometric of your face, those are going to change.

And we have existing laws that handle that. For example, if you are getting a driver's license at 16, that's not valid for more than 5 years because we know you're going to change by the time you get to college. And of course, with the residents here in Utah who have driver's licenses who are over 18, it's a 10-year window. There are opportunities for those things to be changed as part of the credential or to .., for lack of a better term, lose the endorsement.

Now when we talk about healthcare data there are a ton of different places where this applies. Now we had healthKERI talking about the provider payer relationship. That's just one of a myriad of relationships that exist between you and your healthcare interactions. Whether that's the ownership of your individual data, your desire to participate in some sort of demographic mapping. Today, we don't do that. Today we have systems that basically, at government control, take your data and aggregate it for research purposes.

And for some people, that's not shocking. But when you go to a state and you say, "Hey, do you have an All-Payer Claims Database?" If the person's in the no, that means that every encounter you have that's part of your insurance program in that state with your physician and the notes associated with all your ICD-10 codes, everything that happened during that medical encounter is now in a government database for the purposes of looking at healthcare outcomes. Did patient A and patient B who received different medications, which one had the better outcome? There's a lot of good intentions in terms of why these data sets exist. But I will also tell you the most dangerous thing in government is a well-meaning bureaucrat. It's not that they're trying to do something nefarious, but sometimes they try and get hold of data and put it together to create a picture. Where you don't have any interface to this and you don't have any control over that data, there's the possibility that that inference, that conclusion may help you, it may harm you. This is part of the reason we want to start looking at how we switch this model around so that you have more executive control and how your digital twin story gets told.

Now when we talk to them about the state endorsed digital identity, we also like to help people understand this concept of perpetual identity. And this is where we differ with the mobile driver's license community and they're saying you can use it for everything.

I'm like, you can use it for everything that you don't mind having burned down if you move to a new state. That's totally fine. But in our particular instance, we want to be able to say that there are certain documents that you may associate with your SEDI and they could be your medical records your digital driver's license. Any of these items that you have associated with your SEDI can be created and you may have information that you are going to selfassociate like, hey, here's my claims history. Here's some of my MRI scans, things that you have in your possession that you want to get into your digital records but currently don't have. And this is where these functionalities like Blue Button are fantastic is you have the ability to say this is categorically my MRI. I'm going to send it to you electronically. You can pull it directly into my record set. That's the level of control that we want to have. Not: go to doctor office A, pay $50 for a doc fee to have them really, it still cracks me up, fax information to doctor B, and then hopefully they didn't lose anything in translation and the whole packet's complete, which you generally find out on your next medical appointment that it's not somebody dropped pages 8 through 12, and now you got to go through this whole cycle again. We have these inefficiencies that are in the system the way we run it today. And we're telling people, no, if once you're in control of this, you can start to aggregate your data as necessary. And of course, we talk about the concept of perpetual ownership. If I move from Utah to Texas, there are certain privileges that go away. I don't keep a Utah driver's endorsement anymore. I have to go get a Texas one. But everything else that I had associated with my SEDI is still valid. And in fact, because I have a SEDI that's authorized by Utah or endorsed by Utah, Texas can say, "We know that's Utah. We're going to accept these baseline facts that you've asserted. We can bring you right into the system and we can move on." And hey, guess what? You can go get your Texas driver's license endorsement without all the reproofs that we have to do along the way.

When we looked at SEDI as a functional item, we wanted to build into the framework some very important principles. And of course you can see them up here. We break them into some general categories. Now this is where I want you as KERI experts to start ticking off in your mind which one of these items are solved by KERI, which one of these ones are actually enhanced by KERI, and also when we get to the end, I'm going to give you a little here's what we're not going to do at the state level. Okay? Privacy and autonomy this is pretty straightforward.

We want you to have interactions with your digital identity that the government does not monitor has no capability to and also third parties. We do not want crosscontext correlation. We want you to have individual relationships and as such there are technologies that will do that. Funny I think there's one in the room here that might do that. We do want you to have this capability of selective disclosure. I was telling Mike a story where I went to my grandkid, picked them up for a fun day out and we went to a trampoline place and the first thing they did was get grandpa to sign a liability waiver. And as I was looking at it, the first thing I thought is, you do not need my birthday. You just need to know that I'm old enough to enter into a release of liability. And I kind of sparked because not only would that be a problem, I don't actually have permission for my daughter to release liability for her children. There's some other things that could get cleaned up in that situation, but this is part of the community education. This is the open sourcing of privacy we hope to have happen is I'm not going to have every single interaction where I'm going to be able to ask why, but we do want to get people and just across the world saying, "Do you really need that for this transaction?" And that's part of where I say in keeping with what Chris has said, a 10-year journey. Now, if you think back in cyber security, how long did it get people to understand the Nigerian prince scam before we could go on to that's a fishing email and then that became part of the lexicon that people use? It took a long time. I call it about a 15-year window just going back over my history and saying, "It's about when we got people to understand it." And still we fall for these things. But this is the same journey we're going to be on with selective disclosure. We have to get people understanding "No, no, no, you're in the driver's seat. You don't have to give up this information to be able to receive services." Now, there's a major medical provider that I'm not going to mention them by name, but when you go in to receive services, they give you this great set of disclosures. And one of them is basically a blanket that says you're seating all your rights for all your data that we may use in research purposes including biological samples.

We don't have to notify you. And by the way, if you don't agree to this, we're not treating you. That is scary that they have that capability. But as we get a community that's becoming more savvy about these things, not just small groups like us, but the larger community, we can affect change and make those policies flat out illegal. Okay?

And of course, one of the things that I do want to mention is personas in the term that we're using it in is basically helping people understand. You don't have to stand there at the counter and pick element A, element B, element.

There are certain transactions you're going to have that are going to be standard. If you go to a bar or you might have a pub crawling a persona that says, "Here's my name, here's my face, and here's when I want you to cut me off and call the Uber for me." That's going to be something that we would expect the wallets to start supporting is how do I map these things very quickly so I can get to through the interaction as frictionally frictionless as possible.

We get the next one going. Okay, I talked a little bit very fast about delegation and guardianship the other day. Probably faster than I should have, but we'll talk about it some more. Delegation and guardianship is one of the most critical relationships that we see both from a government and a societal

capability. From a government standpoint, there are multiple programs that operate based on the relationship that a parent has to a child a caregiver to a foster child. You can you can go down the litany of them and you'll find there's a lot. In the story I told yesterday about getting a durable medical power of attorney for my father while he was basically passed out on the bed. He doesn't really remember that. And there are situations where he may want to rescind that. How does he do that? All we've got is a scroll on a piece of paper that I can guarantee you is not his signature, but it will hold up in court because a notary came, took 400 bucks in hash from me and said, "Yep, he had a positive acknowledgement of what happened."

when we have these situations of where we have vulnerable individuals who need assistance or you have an aging adult who quite frankly need you to help them with their finances and other items, it's really cumbersome today. And I've had to do this for one relative where we had to bring the court order down to the bank and then they went through this whole trust and verification process. They called the courts. They wanted the make sure that the all the numbers matched up on the case and everything. It was ridiculous. And it and fortunately, they only had one bank. I was thinking, what if they had like an investment portfolio completely separate and that investment company isn't even in my state? I got chills on how long that would take me to help them out. And with the technologies that KERI is providing, there's the ability to really do that quickly through these the these designed and delegated guardian relationships.

And as I mentioned in that little vignette yesterday that all my brothers and sisters were sitting around the table arguing, I realized there's six adults here that could come to some consensus on what is best for the situation. And it would be in the interest of my father or my mother to have us have that sort of capability. Multi- signature to be able to authorize these activities.

Totally important. Now we talked a little bit about agentic AI. It's coming. I'm playing with a lot of them. They're still in my little sandbox, but some of them are getting more useful. And one of the things that I would find extremely useful is I was playing around with Open Claw and I decided I wanted to implement Gemini's or Google CLI and it's like well I usually ask Grock how to do that but I thought I would ask Gemini directly and it came up with this big fat don't do that is a violation of our policies and terms and conditions and we will shut you down no ifs ands or buts "Bad you!"

And I realized as I thought through what would have been the implications of losing my I call it my burner account, but I have not dozens, I have hundreds of logins that are tied to that. Hundreds of logins. If that company decided to take it away from me, I would break my digital life in more ways than one.

Here's your product placement idea. If anyone wants to add this, I would love that once my SEDI goes live that I have the capability to go and break all those relationships automatically and reassociate them with an identity that I physically control. That's where I want to end up because at the end of the day, if we leave all those accounts and logins out there, we're seeding our power to control our own destiny. And of course, we've talked about the permission and consent the demo for KERI blocks.

some of you might have seen at IIW [#41] was fantastic. For those who didn't, and I'll just give you the two-minute overview, the goals are the same. Parents do not want their children in environments where there are predators.

And we know that there are online communities that despite their best efforts, wink the profit motive is still there. They're not investing in the technologies that will protect children. And in this particular example, we had a parent credential, a SEDI-based credential that did a derivative credential for their

child and said, "This is my child. This they are 13 and they can be on this system from 5 to 7 PM, two nights a week, and they can only talk to other children who have a validated SEDI credential."

And it's fun to watch the chat window and all the people disappear who are not able to talk to your child. Now, some people think that's overly restrictive. You're going to have your genius child get around it anyway. That is where we do have that interface between technology and law. Especially in Utah, we've talked a lot about the harms to children through social media and other ways. I don't think it's going to be too much of a stretch once the technology is ubiquitous to have the law catch up to help parents manage their children's online environments. Okay. Security and recovery. One of the things that when I read Sam's paper for the very first time, I was like, "Okay, why is it that we've been doing PKI this way for like 25 years at this point and nobody figured out this really obvious thing, which is if you can't tell what the key state is and where you're going next, you're back to that whole trust model and all the things we did to make CA's work and the intermediate certificate and by the way, get rid of this one."

And I was like, Sam, it was genius how simple it was when you looked at it, but it was one of those things that, until somebody came up with that answer, we were going to still suffer from the same problem. Where KERI has done a lot of things to help us out and of course like Sam and I have talked about this over the last three years about detection of compromise, the KERI specification continues to get updated to meet the real world demands that we have. And one of those is this whole idea of the mutual authentication. We've just talked about that with the phones. We've talked about that with healthcare providers. And I will predict something that is going to be in the future. We have health information networks all across the United States. And they make a lot of money brokering the transaction or uplifting it from one EDI format to another. What a world it would be is if I'm the provider and I can reach out directly to a discovered payer interface. We have this mutual authentication and I can just move that significantly changes the overhead that a provider office has to have. I mean I I know that my dentist for example he has about nine administrative personnel.

He's one dentist with two hygienists but he does a lot of work in social services programs and he has to have that level of people to continue to keep up with the paperwork. And of course that's kind of reflected in his pricing.

Every couple years he goes off plan and I have to pay more to use his services. But if we could get those administrative burdens lowered, think of where we could go. No credential works unless there's comprehensive utility. One of the long-standing jokes about the mobile driver's license is that it works at the TSA. It doesn't. Their terminals are down. They're like, "Hey, do you have your physical license?" And I've heard this one myself because that interaction will go a lot faster than trying to play around with the terminal.

It doesn't have a lot of usability and the models were wrong. One of those models was is we tried to monetize in the wrong way or at least vendors did. In our case the our we're an alcohol control state. You have to go to a state liquor store.

But when you are the director of the state liquor store and a vendor comes to you and says, "Hey, by the way, I need you to pay X number of dollars per month per store to be able to accept this digital credential." And they're like, "Wait a minute. Isn't that issued by public safety?

Why do I pay money to public safety or to a vendor to accept a credential that the state issued?" And of course, their answer was so logical. I can look at the I can look at the state ID or I can look at the driver's license for free. Adoption is a problem. There are a numbers of areas where we hope to see

functionality come along like I talked about replacing username and passwords or taking over accounts. We want to make sure that this area which is where the ACDC comes in great is offline verification. I can't have my environment completely breaking because I don't have online access at all times. Now, Joe Jackson during our our study panel, he talked about our offhighway vehicle OV certificate that we did. And when Joe and I talked about it, I said, "Joe, this is fine for X.509 technology. Completely fine. It's a lifetime endorsement. It's never revocable and you have to supply a primary credential with it. Just because you show the QR code and it has your information embedded in it, you have to show a driver's license." We said, let's go down the road with this one because we want to make sure that there's an appetite for these credentials and that it would be extremely low risk. Joe told you a little bit about and things we've learned along the way there. But I'll tell you what, the funniest thing that happened is once we release those, the boating community, everyone who's owns a boat, I've heard this story like 40 times now. They're like, I own a boat and you guys have this OHV thing that I could have on my phone. Why do I have to have a printed PDF? And they always say paper on my boat. We're in the water.

Why do I have to have paper? We kind of smile and say things are coming. The the appetite is there for these credentials. We have no loss of ideas and use cases of where we want them to go. Key to the whole success is the open standards, no vendor lockin.

We have been very clear to the community that security by obscurity is never going to be a solution we're ever going to be interested. Somebody brings me a closed source package and says it's magic, I don't believe them. I want to see under the hood and in some cases I lack the time these days, but if I can get down into the code and look at A plus B does equal C, I get a lot more comfortable. And I try to point out to non-technical people, the reason Apache is such a successful project and how has the high level of security it does is there's lots of people looking at it and that's where we need to be in identity systems and identity credentials because I can tell you across this room and I know some of you by face, some of you by name, there is a lot of expertise that can get into a room and say A and B are not equaling C and here's why. And as a state, I don't want to have to broker broker that and be able to say, "Yeah, we made that decision on our own. We want the community to support us because ultimately the beauty of this is, it's your data." And by golly, you're going to care about your data. I'm going to get some really good feedback as we go along in this program, which we have.

And we enjoy the fact that you can do that. And as Chris has said, if a standard requires you to pay to be able to understand your interface between you and your government or how your identity works, that's a non-starter for us. We just we just won't go there.

Now, this part about any app becoming a wallet, what we mean by that is we do have some digital credentials in the state already. For example, if you download the Department of Natural Resources app, you can get a QR code for your fishing license and your hunting license. That's great. But that's only inside their little tiny ecosystem. We want to be able to broaden that out. But that doesn't mean they have to throw away their app because they've invested some good money in it. They also have some poaching features.

You can take a picture of a carcass and a GPS and it will send it to you DNR enforcement officers. We don't want to say "No you got to go through the wallet only, "but there are going to be instances where we will see extension of SEDI credentials inside existing applications in the state. The other thing that we are very aware of is the first mover advantage for a lot of people is not going to be going through the platforms.

They're going to be doing their own trusted apps. To me, there are many Roots of Trust that can come into your relationships. For example, I may trust my bank and you say, "Hey, we're going to help you out with your Wells Fargo wallet and we know about these pre-key rotations and we provide a service where you can escrow those keys with us and using your multisig capability. Us plus you plus whatever you know key material you have, we can unlock your next key for you." That's fine and that's perfect. And I want to harp on this just a little bit. The state's credential is the SEDI. We are not the wallet. We're not your key escrow. We're not your key recovery. I kind of smile and my wife says this a lot. She's a children's librarian. She works in a public library. She has a little counter. It's almost her personal drinking game where she says, and I quote, "I'm sorry, I do not know your Gmail password. I can't help you with that. Do you have somebody you can call to help you with that?" And she has people who argue with her. No, no, I used this computer yesterday. You should know my password. From the state, I suspect we're going to get something like that where people are going to be like, well, you issued it.

You should know how to recover my keys. Nope. That's part of this whole idea of, hey, I'm turning it over to you. Now, with the wallets, one of the things I do also want to say is that the state definitely is going to be looking for third parties who will go and audit wallets and provide some we'll call them seals of approval. What I anticipate happening is when you go to the state to get your endorsement for your study is we're expecting a cryptographic hash and a signature of the application you're using. And I can say "Yeah, that's one of the ones we recognize as meeting the standards of the state. We will go ahead and finish up this issuance process and give it to you."

Steve McCown's going to take the source code directly and he's going to compile his own wallet and make sure it meets his requirements. When Steve comes to the state and says "Here's my wallet!" And in the presentation he provides us a hash and a signature we don't know we'll say "Okay, Steve, we're happy to issue to this to you but we also want you to affirmatively consent that the state doesn't know anything about the security of what you're doing."

Yes, you want it. Here you go. Because we do want that freedom of choice and the ability for people to operate, but we also want to have that guardianship that the state has, which is the public interest to say, we know you're not all techies and if you bring us something that you got your.ru wallet and it seems like it was the right idea.

We want to make you aware of what that potential security risk would be. That's something I expect to see coming in the ecosystem. Now we at the end of our slide deck always go back to what is the key points we want and we want you to be in charge. We want you to determine who you share data with, when you share it with them, and why that matters in the healthcare environment is we know that data can be weaponized against you in the health space. And if you have no visibility to that and AI models are starting to be used and I'll say quite frankly in inappropriate ways to come to like billing and costing decisions, you have to be made aware of that. You have to have control over that.

Interoperability. You've heard this if you've been here the last two days. What's the interoperability? We are going to have to be interoperable. We can't expect that every state is going to say we believe exactly how Utah believes and we think their technology choices are great and we are going to go that way.

I don't know if you noticed during the panel with Joe and our representative from California, Joe had said we have to agree that we're solving the same problem. If we're not, then interoperability isn't a question. And I completely agree with where Joe was at on that. We definitely want to make sure that we're solving the same issue and that we're moving forward in something that is rational for all of us.

Digital reciprocity between states is something we're going to have to deal with. We definitely want to make sure that, hey, if you've ditched all your your plastic, you jumped on a plane and you're suddenly in Oregon and you get pulled over in your rental car, your digital driver's license has to be something that can be considered valid in all jurisdictions. And that's quite frankly a format war that it's not my problem right now that I'm trying to solve. I'm trying to solve the core identity issues for Utah residents. Now, Of course, we have the Olympics coming up 2034.

There's probably going to be digital credentials that come to us from other jurisdictions even outside the United States. We are building this system around the concept of American values. And what it is no tracking, no surveillance, no no monitoring. In those cases, we may say, "Yeah, that's a valid credential from where you came from." We might issue a temporary one while you're here for the Olympics to allow you to interface and do the things that you need to do, but we're certainly not going to phone home to a foreign server and participate in state surveillance on behalf of a foreign government. We just won't do that. When it comes to the healthcare interoperability issues, we want to make sure that the blast radius is always 1. And this is the thing that .., I hate to say it, but we're finally getting to the point where data breaches, people are understanding that your data continues to get lost because they're in central systems; these huge honeypots. If we do this right, people are still going to get social engineered. But if the blast radius is 1, that's a much better situation for us to be able to handle than having a blast radius of, "Oh, 100 million records were lost. Would you like some free credit monitoring and hope things work out for you?" And when we look at the overall cyber security risk, there's definitely going to be some things that we are going to look at from the from the wallet provider community, key escrow and how that's done is going to be super important to us because it's not your identity if somebody else has full access to your keys and can perform rotations without your permission. You I got asked the other day, well, what happens if I'm at the border and the law enforcement agency requires my phone from me and they copy my whole phone and then they wander away and now I know they've got my current keys. Sam, you can put this on a t-shirt if you want, but I just said when in doubt, rotate out. You don't have to stay with that key pair.

When you get your phone back, move on to your next set. I'm going to give you a situation that happens in rural healthcare because we talked all about these things in kind of this really large buckets, but when I was down talking to rural healthcare, we had to have some specific examples of how rural healthcare is hurt by not having a good identity system that can provide instant validation and of assertions of credentials. If you look up there, 50,000 to 100,000 is the and this is across the nation, not just Utah. This peer-reviewed studies that they can bring in a doctor or say a critical care nurse and getting to the point of having the original documentation validated.

They're basically sweeping floors or doing something else or working in a non-risk position while those credentials get validated. 50 to 100,000 per provider they bring into the system. And the tragedy is when you look at these small healthcare systems in these rural settings, you may get hospital privileges for this rural hospital 50 miles down the road where you might want to work on a weekend or provide services. You're not accredited. You have to go through this whole credentiing process again. It's really hard for people who have made the commitment as a healthcare provider to be in a rural community to be able to service more than one. It becomes a nightmare and a paper castle. We're looking at SEDI and the subsequent credentials that get endorsed by the medical organizations and others. We want them to be able to show up with an ACDC in the middle of the night and say, "Here you go. Here's my whole chain of credentials." And all those boxes get ticked and that person goes to work doing the job that they want to do in that rural community. And of course, one of the things that we don't want to

Underplay is the fact that these rural healthcare systems compliance is a huge issue. You blow up one thing and you suddenly get all your payments stopped. In a lot of cases, that can be detrimental to the organization. I know for a fact that in the state of Utah skilled nursing facilities for long-term hospice care patients, their hash flow is about 2 and a half to 3 weeks. If you get an interruption in payments, you're calling relatives and saying, "You need to find somewhere else because I can't afford to keep my facility open." It's a real risk.

All right, let's talk about us. I don't know if I'm going to try and get this as good as phonehome, but I want to kill the clipboard. I am so sick and tired of going to an organization that has my medical records, and every time I come in, they want all this demographic data they already have.

It drives me insane and I hope it drives you insane, too, because it's like, look, you know who I am. Think if I attest to things haven't changed since the last time I saw you, but I'm here for a sniffle or a broken arm, I don't need miles of paperwork to tell you the things that I you already know about me.

And I will say I've been rendered functionally illiterate by computers. My handwriting is awful. Half the time when I'm in these places, they're like, "Is that an E or is that an A?" I'm like, "I don't know. I'm in pain. I'm in a doctor's office. I'm not really interested in a critique about my my handwriting. What we really want to do with this kill the clipboard is reduce these inaccuracies that come in. And for the love of everything, why do you have to get a copy a physical copy of my medical card when I provided it to you a month ago? And part of this is not the it's not the fault of the front-end worker. They have a set of procedures that are trying to do risk reduction. This is the way that we operate our systems. And from my perspective, if I could just go in and do a tap and say, "Yeah, let me just punch the items on the iPad of where I hurt and how bad, that's a much better experience for me as an individual." And I also know that the data errors aren't going to be because you couldn't read my spelling or you transposed a number on my medical card. It all came across in a trustable transaction or series of chain credentials that we can now move forward on the situation. That's the utopia I look for. Now as I said at the SEDI summit we look at this in government in in a lot of different ways but if you take any process whether it's government or whether it's business look at the pie chart of how much of that transaction is trust verification risk management and all the other things you actually do before you get to service delivery. It's inordinate the amount of time we spend in making these things work because we're resetting the trust model every single time. In government, where this hurts us so much is that we actually have to staff people who are part of that front office to injust this paperwork, make some value judgments, move things forward, and it's not something that is sustainable.

Utah has experienced record growth over the last decade. We're just growing, growing. We don't want to grow government at the same rate. I can't keep hiring people to ingest this physical paperwork. I have to remain in the digital realm, but I have to be accurate.

We are facing this dilemma everywhere in the United States. As populations growcivil servants retire, next generation may not be looking for that sort of work because generally it doesn't pay really well. In fact, it doesn't ever pay really well.

But you those are the individuals we want to be able to have a good experience with us. But we're not just going to keep adding and adding. We have to get the digital systems working to our best effect. When we look at overall the situation that SEDI can solve, it's not a situation of where it solves only one problem. It solves a number of problems. When you're thinking about your specific marketplace that you work in and how KERI is going to make things a lot better, We see it as an underpinning that breaks a lot of the traditional models and lets us move on.

I look forward to a world endpoints where I can identify and move things around without an intermediary. I know Chris and I have talked about this a lot is that when this is live and you have these items available to you in a functional format with utility, we're going to have a lot of people who are going, "What's my purpose in life now?" Because I've had my job as an intermediary for a long time. But I'll tell you what, there's always going to be something new for people to do. I love history and one of the things I find funny is when the automobiles were first allowed in London, it was a requirement that you had a signalman walk in front of your vehicle with a bell and he'd ring the bell to make sure as you put it away that the horses were not startled and they even extended that into the evening where he would carry a lantern. We don't have signalmen in front of cars today.

They do something else. And as we move forward with our new user centric sovereign environment, I think we're going to see some significant changes in marketplaces. But it's not just going to be driven by the state. We can set governance. We can set law. But as a larger community, we need to ask those questions why we do the things the way we do. And if we are not afraid of those questions, not only in business, but in government, we're going to see these massive improvements along the way.

Okay, we have like three minutes. It's always hard to be the one who's like backed up against lunch because the questions drop off. If you don't ask your question now, that's okay. Find me at lunch or over the next day or so. Do we have any questions I can answer in the next couple of seconds, minutes, whatever.

Can we get a copy of that? Absolutely. That was a simple one. Give me a harder one. - My question is about on the delegation slide. You mentioned several populations where it's super relevant and one of them is like unhoused population and migrants and what how does delegation look in that case? Who's the delegator?

In that specific case I'll tell you what we find with homeless population specifically in the healthcare environment. When we have an encounter with somebody who's homeless, the best we can do is put them on something like emergency Medicaid, which is super expensive for the state, and we can provide those critical services that keep them from dying, but that's not long-term maintenance. It's not prophylactic medications or anything of that nature. Those don't come until we have identified that individual. And given that we might have to query them like where were you born approximately when were you born all these items that we case work it's assigned to them. It is generally 6 to 8 weeks before we have documentation in hand that we can now say we have identified you and we can now move you off of the emergency Medicaid and put you on to a different service. The tragedy is once they hand them that packet of information within 30 days the majority of them have lost their documentation. And it's like well of course they're unhoused. They're not going to walk around with a manilla folder under their arm.

How do you solve that problem? Well, there's a couple ways you can do it. One of the .., I was at a conference and I was just being flip and I said, "Well, why don't we do a system called Nomad?" And they were like, "What do you mean by that?" I said, "National Online Assisted Documentation, or management of assist of documentation."

We are looking at opportunities in it that could do something like that which is basically there are some things you're always going to have with you and those are going to be some of your biometrics and if you are willing to work with the state to get you identified and put you into a national data store that's encrypted to your biometrics then we have means for how we can get those people on the system. There are care relationships that exist inside the state like Foster Care or we have Battered Women who

are put in like safe harbor shelters. There are some there are some relationships that we need to clarify in not only in law but how that works digitally because there are certain rights that get communicated to each group. As we move along there will be both a policy side and a software technology side to work on it.

But the interesting thing is we do think that the easiest relationship for us to solve right off the bat is the parent child relationship. That's pretty easy to prove. But when we get into some of these more complicated situations, so say for example, a Medicaid household is defined by the number of people living under the same roof and what happens if you have an acrimonious divorce. Party A has a lot of KYC data to try and infiltrate what party B is doing. The state's going to have to be very careful as we roll these out to make sure that when we do endorse something like a parent child relationship that we do have a glass break where we can go and turn that off so that we keep everyone safe.

Okay, I think you is we're at lunch. Thank you again. I look forward to talking to you individually. Reach me out to me at any time. I'm always willing to talk about both the tech and the policy. Thank you very much. Thanks.

TALK XV

FHIR and CMS

Jared Jefferey

KERI Conference April 2026 · talks · 39:59

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Jared Jefferey of healthKERI introduces the VITAL initiative as a verifiable identity and trust automation layer for healthcare. He surveys CMS interoperability policy, FHIR basics for the KERI audience, and CMS-0057-F, arguing a missing trust layer blocks patient access and that CMS scale makes compliance strategically essential.

All right. The title of the talk is FHIR and CMS. And we'll talk a lot about that. I actually want to spend a little bit of time today going over what the vital framework is. It's actually the vital initiative. I need to change the slide on this.

This is an old one and as you can see, it's glitching out. But I want to talk through this, what it means, why it's important what the acronym of verifiable identity and trust automation layer is. Here's just a quick intro. We're going to go through CMS and their standards process interoperability standards in healthcare kind of at large, how we got to what FHIR is now today.

What is FHIR? And we'll go very high level on this. You we're not going to go into kind of the different IGs within FHIR. We're not going to talk about FAST necessarily. Just this is FHIR 101 for the KERI ecosystem as they want to come up to speed on the applicability of the KERI ecosystem for different industries. Then we'll talk about CMS-0057-F and then we'll talk about The VITAL initiative. Just as a as a high level Jared Jeffery CEO founder at healthKERI, a startup that is in the industry of the missing trust layer for healthcare.

We have adopted some open source open standard protocols. If you're at the KERI conference, you can probably guess as to what those are. But for us, it is imperative that we see this trust layer be implemented into the healthcare ecosystem. As a little bit of background on myself: I am a fellow of the American College of Health Data Management. I'm former class research. Class is probably the biggest name in health- care's market IT research and convenience.

They've spent years and years getting all of the right groups into the room where typically those groups are very contentious. They'll put those people down in a room say take your hats off. Let's actually solve problems. They've done a lot of work like getting pairs and providers to come together and talk about solutions which is the fact that they did that and it never came to blows was I think actually a miracle.

And then previously my educational credentials come both from UVU. Happy to be back at the alma mater today. And then WGU, which is Western Governor's. Okay. Want to talk about just really quickly why it is we're doing this in healthcare. My background is one 15 years in healthcare IT, yes, but I'm also a brain cancer survivor and patient myself. At one point I was being treated by the single machine in the entire Rocky Mountain West that could provide me the care I needed to survive, to thrive. Without that machine I wouldn't be here today. The reality is that for many millions of patients that is a very similar thing but often for them it is not some bespoke proton therapy machine that is hard to build. Oftentimes it is something as simple as access to an ER doctor at the time they need an ER doctor. One of the things that has started to happen increasingly is that as these ransomware attacks which are targeting healthcare, (we'll talk a little bit about that in a minute), the targeting of this industry has real world implications. The why for us, why we're here, why we're doing this is because as we increasingly move

care into a digital context, we increasingly make the risk to patients explode. And why is that? If you shut down a hospital using a ransomware attack, the first response that hospital has to do from a care perspective is send every ambulance that's supposed to come to them someplace else. And especially in the US that ecosystem is incredibly fragile. If you're talking about for example an urban center, they may have to push hospitals out to hospitals in rural centers and those hospitals may not have the same equipment, they may not have the same training, they may not have a level one trauma center. If you shut down a large health system, you are actively threatening the lives of patients. In fact, NPR about two years ago put out a report that stated that a ransomware attack on healthcare should be considered as disastrous as a natural disaster. A tornado or hurricane, they are the same level of disruption. The difference of course is while we can blame God for tornadoes and hurricanes we blame Russia and North Korea for the ransomware attacks that are happening in the industry. A little bit of data on this. Healthcare is the most targeted industry by a large margin when you look at some of these other pieces. There's a few reasons for this. They're also the most expensive. This is part of what's driving that. You'll get the biggest payout if you're holding a knife to the throat of an organization that has to pay you or people will die.

Attackers know this. This is a persistent data source. This isn't just one year. This has been for the last decade. Healthcare has been the most targeted industry. One of the questions we get all the time is why attack patient data? What's the value proposition for an attacker to come into this ecosystem and to compromise all of this data? The reality is there's a few different things. The first is that on the black market, a patient record is typically 10x or more the value of let's say a credit card, which makes perfect sense, right? You your credit card gets compromised. What's the first thing you do? You call up Capital One or Chase and you say, "Hey, I lost my card." And within a week, they've got a new card for you, right? They've rotated you to a new credit card number. And you're back up and running. When you lose your medical record, you're losing your social security number, you're losing your date of birth, you're losing your name, you're losing typically one or two payment methods, and you're also losing all of your medical information itself. None of that can be rotated. As much as I'd love to call Capital One and say, "Hey I've got brain cancer. That's great. Can we rotate this to like diabetes?" I'd prefer that it just doesn't work that way. You have immutable data in a health record, which means that the value of that data lasts for as long as the data is valid, which is forever. With that data, there are a number of different things that they can do. One is just selling the records online but beyond that it's things like standing up fake clinics to execute fraud to CMS and it I can prove to CMS that I'm treating you and I can do that at a scale of hundreds or thousands of clinics.

There was actually a clinic, this was two years ago, I say “clinic”, there was a group in Florida that was found to have absolutely no patients, but they had a whole lot of data and they were billing CMS pretty aggressively to get money out of that system. Other things to note here is and this one I actually want to hit pretty heavily which is extortion attacks. It used to be that the primary attack surface that an attacker was looking for, the monetization that they wanted was if I hit a hospital, they'll pay me $7 million to unlock my data. And that's still true.

But these days, the attacker actually also couples that with exfiltration of that data. And because AI has made the cost per extortion so low, they'll now go after individual patients as well. They'll go after .., a recent article I read was that a bunch of mimography images were captured and they were actually going to individual women and saying, "Hey, we have compromising images of you from this this data that we pulled out of your health system. You're going to pay us $2,000 or we're going to put these out

on the dark web." It used to be that $2,000 wasn't enough to get anybody interested. These days, because it is so simple and easy to make that attack, and by the way, I already have all your contact information, we can just set up a script and the AI can extort for us.

Which is obviously not terrible but when we ask about 'Why is healthcare so compromised?', it's because it's insecure it's high value and the data doesn't change across time which makes it a really good find and can be resold multiple times in the future. The other problem here is, not only is healthcare's data incredibly valuable, it is also incredibly complex. You have health systems, you have health payers. These are typically the two big players in the space. Each of these ecosystems will on average have about 1,500 connections. The average health system has got vendors that are connecting into their system doing a number of different tasks for them. Each one of these connections represents an attack surface for the attackers. I showed on that previous slide about 3/4th of [inaudible] of actually originate in an external party. I think a good example of [inaudible] helps people understand [inaudible] happened, is probably five six years ago now.

All of the credit card information in Target got stolen. The way that they got in was through the HVAC vendor. The HVAC vendor had a connection into Target. Obviously HVAC, he doesn't feel like he's got to be all that secure, but simply by virtue of connectivity, they were able to do a privilege escalation attack, which is basically you compromise the first set of credentials. You move from those credentials laterally to somebody else that has better access into the system and then you can move up. Eventually, you're looking for a sysadmin role.

Once you have a sysadmin role, you have keys of the kingdom. You execute your ransomware. Within a month, the cyber insurance will have paid out the $7 million because the hospital's got to get back up and running. That's the attack in a nutshell and this is the risk.

Again thousands of connections coming into the system each one of them representing a potential attack surface for bad actors to enter in. We used to think and I think some people still do that if we've encrypted the channel then we have a secure connection and the reality is that the attacker will most often use that channel against you hunting for what we call shared secrets. This is a shared credential. This is username and password. This is a bearer token.

There's a number of different things that qualify as a shared secret. Our contention is that we probably should stop letting them steal these shared secrets. Impersonating your third party, moving through the channel that you trusted to be secure, and then exfiltrating your data. It's time for us to actually reduce the attack surface by taking this piece here and removing the shared secrets that come into the system so that those attacks are no longer valid, no longer reasonable for an attacker to execute. Now, we've talked about earlier today a lot of the value in terms of the connectivity problem with regards to onboarding, with regards to the manual lift on it. One of the other things that KERI does for us is this. It allows us to create an ecosystem of true zero trust where there is no compromising credential for participants to be attacked on.

In fact, we've actually shown this. We had a pentest done on a system that was basically running this. We stood the system up. We were running hundreds of thousands of FHIR transactions, (We'll talk about that in a second) through the system, we brought in some white hat hackers. We pointed them to the system and we said, "Go ahead and break it. Here's everything that you would need if it was a traditional system." They came back and said, "We can't get into the system. There's no serious flaws

that we see." It was humorous to me though that they did say, "Hey, there's no TLS on this connection. You guys are missing that. Maybe you should put it in there." And the reality is we believe that KERI-based connectivity is actually a replacement for TLS.

Tthat's kind of the background on healthKERI and why I think the vital initiative and what we're going to talk about today is important. Now let's get into what CMS is, what they actually do, why they matter. Very high level CMS is the Centers for Medicare and Medicaid. Remember Medicare is for the elderly, Medicaid is for low-income individuals. We care about our elderly, we provide aid to the poor.

That's the way I always remember it. And then you can keep those two straight. It's important to note that while not all care happens through CMS, they're managing what was it like a trillion dollars? 1.2… 1.8. Round up, you've got a couple trillion dollars flowing through the system every year, which means across years it's trillions and trillions of dollars. Even if you are not a Medicare or Medicaid plan, even if you're not accepting Medicare or Medicaid dollars this is the 800 lb gorilla in the room. When they speak, typically everybody listens. Mark – Amendment. Medicaid is not run by CMS.

– Right? It's state-run. – The states… the reason they have the some level of control is that they'll pay a 90 / 10 split for investments into Medicaid. – That's good clarity. But at the end of the day, the reality is that if you are looking to get money out of the healthcare system it would be wise of you to listen to and abide by the regulations that CMS puts out. Now CMS themselves typically do not originate standards. What they like to do and what they've done numerous times in the past is that they will work with an industry group like HL7 who who stands up the FHIR standard and they will basically say these are the things that we are looking for. There's deep collaboration there. FHIR will say, okay let's get the industry together, we'll stand up a standard like FAST which is what Mark was talking about earlier today. Once they have a an acceptable standard CMS will come back and basically bless that standard, put it into regulation and say this is how we want to see things done. For example, we talked about pretty extensively earlier today, CMS-0057.

It is a regulation. It's a requirement for payers. They will not get or will be penalized by CMS starting January 1, 2027. And that penalty is based on whether or not they have a FHIR API endpoint. Again, you start to see this converging of the federal regulators working with industry groups to establish these standards, then enacting them into mandate. And then often, but not always, though I would argue the most effective ones are when they tie those to payments.

If they say, "Hey, you're going to get penalties. You're going to get fines. We're going to cut off your Medicare Medicaid funding." That's when you actually see the industry move. And then they'll come in and they'll basically say by X date you want to see Y happen within the industry. Let's talk about kind of the history of these standards, how we got to where we are today. I think there's some instructive pieces here. It's important to note that we started with what I would call foundational interoperability standards and as we've moved across time we've moved up to, I think, much more useful standards which would be network interoperability standards. Now I want to give a couple of definitions here. A foundational standard is basically just what's the pipe look like? If you stand up the pipe: good, you've met the foundational standard. And there we'll talk about some examples here in a second.

Structural is really just: I have the pipe. I know that the data is coming through and the reality is that this is for a diabetic, let's say it's a glucose test. That's more useful than just: "Hey, data is coming in and out as you move up the ladder to like semantic data. This is where you start to get structure and coding

into it, "which basically lets you say things like: "Hey, this is a glucose test, it needs to go into the doctor's decision support tool." Right? As you build the standard on interoperability, you get more and more value out of the data that's flowing in and out of the system.

The network standard, and this one is .., these are I would say adopted nomenclature. I've been adding this. This is new to healthKERI. My argument is that we need to get to network standards where it is not just structured coded and scaled, which is important for these other things. It needs to be governed and that governance piece is the piece that healthcare is still I think currently grappling with.

Let's talk about a few of the ways that this lays out in terms of manual effort. If you have a foundational interoperability model, literally it's just the pipe; everything else is on you. Congratulations. You have data. Have fun normalizing all of that. Maybe you'll pass it into Mirth or Iguana or somebody else .., they can do the job for you. This goes all the way up to the question we should be asking, which is: "When data is flowing through the system is the standard built in such a way that we can answer the question affirmatively, will this improve patient care?" And that's what we're trying to get to with interoperability standards.

Let's talk about what some of these standards are. Back in the 80s which I was not around for ADTs came out. ADTs is admissions discharges and transfers, basically this was the building block of healthcare's interoperability. Can I see from a hospital perspective is the patient in my hospital, is he out of the hospital or has she been transferred to another hospital? And that was the foundational effort that came again back in the 80s.

We moved on to between 87 to 95 there was CDAs that were added in and the one that I actually want to talk about is HL7v2. HL7v2 is the precursor to FHIR. This is probably and correct me if I'm wrong here, Mark, the most adopted standard in healthcare today.

Most data flows on HL7v2. It's HL7v2 dot question mark… – It's standard. – standard adjacent. That was back in 95 that this came out. You can see that healthcare is… go ahead. – What's the differentiation there? What's what's the customization? Is it the data model that's consistent? Is it just what's the proprietary niches?

– It's extensions. It's wide open extensions. They call them Z segments. They just let you let you put whatever you want in there and define them for your local use. It's a local extension and they're allowed and a lot of them are used in fact even later versions of HL7v2 that actually support those capabilities later. People didn't change their Z segments because they were working and they were mostly were communicating inside the enterprise so communicating from the system that manages the encounters with the systems that are [inaudible] so from the [inaudible] system the lab system back. That's why they could be that's why they couldn't be not standard because they were for use in a – And the other thing is the v2 and CDA standards were effectively proprietary. In those days you have to pay to get access to those just like you do today with X12. If you want to do a financial transaction in healthcare it has to go via RX12 because cost you like 1,500 bucks a year just to be able to see that standard.

– [inaudible] – Yeah. Right here. It's all in here. And obviously not comprehensive. We're just trying to do a high level here. Then back in 2011… you'll be happy to note I was here for this one. I was here for this one. The FHIR standard was put out in 2011. Since then there's been a lot of work in the FHIR space. I think at this point.

When I got into healthcare we were just talking about interoperability standards and there was a lot of discussion around, is FHIR the go forward, are we going to do this? Are we not? Because a lot of people were happy with HL7v2 in the sense that they're lazy and didn't see the need to change and we're still faxing things anyway. Why do we care about a data standard? I would say that in the last 5 to 10 years that has really flipped and the industry has started to say yes FHIR is the go forward standard. Even if people aren't using FHIR today for all of their connections everybody has kind of accepted the fact that at some point FHIR is going to be coming into their world and again we'll talk more about what FHIR is in a second. I want to go up one level and talk about kind of the network level of interoperability standards. For things like TEFCA, which is the trusted exchange framework and common agreement, This is a federally backed program to do nationwide interoperability.

They are actually using FHIR in TEFCA. This isn't a morphing of this standard here. Sorry, they talk a big game about using FHIR. They anticipate that in the future they will use it. CDA or HL7v2. But they have acknowledged that this is the direction they're going to go (FHIR).

This isn't a new standard as much as it is a new set of guidelines. That common agreement is the governance framework that they're trying to instantiate on top of existing frameworks. The other one that I would mention here just kind of in passing is CMS aligned networks.

That's one that will become I think more relevant as we [indaudible] and FHIR in the ecosystem …the first being we don't actually have full adoption of FHIR up here yet and beyond that these ecosystems up here do not have, I would argue, an effective mechanism for governance and trust on the frameworks. And I think I've got a couple things. We we'll come to it later on why I say that. But let's switch just very quickly to talk about FHIR. FHIR is fast healthcare interoperability resources.

It's a lot of things. It's an acronym, but it's also the data structure. It's restful APIs, which is just modern API connectivity. It's a set of tool sets. It's developers kits. It's, validation implementation guides. HL7 actually does maintain public FHIR test servers. If people are looking to get into FHIR, they've made it really easy because you can go in and play within their ecosystem. And then there also is a community that's I would say relatively strong and vibrant, growing every day in terms of people in the ecosystem doing FHIR.

like I don't think I've talked to another healthcare startup that isn't starting with FHIR in terms of the connectivity that they're trying to drive into the ecosystem. The other thing that I think I should point out here is that the data structure piece is actually incredibly valuable. A CCDA HL7v2, they didn't really allow for discrete elements of the data to be called out, which meant that often times when you're transacting a data record, what you're getting is a stack of information and then you'd have to go through and kind of parse through that to get to value.

– How does that relate with various EHR type? – What do you mean? – I think electronic health or portable health records. I'm not sure if – Yeah. With EHRs it's going to depend on EHR, Obviously, most of them are transacting data on HL7v2.

– But does this say you have to use a particular type or aren't there multiple EHRs? – There's lots of EHRs out there. The ONC basically has really got the role of regulating the healthcare tech industry. – ONC being the office of the national coordinator for health.

– They, a number of years ago, defined FHIR set of resources known as US core. And basically had certifications so that if you were a certified health information technology platform you would have to certify against that and that required you basically to have a population health API based on FHIR,

patient access API based on FHIR, and using those calls that was on an older version of FHIR and what's happened over the last year is they updated to the same version the players are actually using.

It is a standard that lives underneath the EHR itself right because they're trying to get data in and out. Back to the data structure the model here, these data descriptions, and it's completely gone now, which is great. Is actually really important in terms of an upgrade here. One of the things that FHIR allows you to do that previous interoperability standards didn't, is to be able to call out discrete data elements from the FHIR record. When you're talking about a multi-hundres-page patient record, which I'm sure mine is with all the things I've gone through the ability to go in and say, I just want these blood tests, I just want this from this date to that date. To be able to parse a record at scale like that. That's incredibly valuable. And I think, all the rest of these things are essential. They're helpful, but really it's that data structure and the ability to make the data actually useful in clinical practice is what's driven FHIR.

– [inaudible] the rest of the API as well because traditionally what would happen is if you wanted data, you would make a request say basically send me a document. Yeah. Right. They're going to put that data into the document. I've been in meetings where there's been discussions of how much data should we put in a document if we're going to have a standard which is a little bizarre, whereas with the restful API and those data structures you can, if you've got the permissions you can go in and say give me just the A1C readings out of the observation resource.

– Yeah just give me the discrete elements from the data that I actually want. This is the objectives of FHIR that's stated by HL7 who's standards body that forwards FHIR. They wanted to be built for implementers. This isn't something that was thought up by people like me who would definitely get it wrong but actually by implementers that know what needs to go into a standard.

They regularly do connectathons because it doesn't do any good to have an implementer standard if you can't teach people how to get this done. I think that this is another piece that's very valuable is that they operate on an 80 / 20 rule. They're going to anticipate that they can get about 80% of all the use cases enacted into the standard. But that XKCD Comic I had up earlier is very true which is if your standard is rigid not flexible you end up proliferating standards. The other things that are I think important is it's open source. And the adoption as far as everything that FHIR tries to do is to make it incredibly easy. That's kind of the FHIR in a nutshell. Let's talk about CMS-0057.

This is the latest version of that synergistic loop that I talked about earlier, which is CMS doesn't want to set standards. They want the industry to set those standards through somebody like HL7 and then they come together and start to put them into mandates. CMS- 0057 basically says that by January 1, 2027, every payer that receives Medicare and Medicaid dollars has to stand up an API for these four use cases, which is payer-to-payer exchange, patient access, payer to provider exchange, and then a streamlined prior authorization process. Is anybody not familiar with what prior authorization is? We've thrown that around pretty loosely, but if you don't know, okay, great. I will say that when I went to go get the proton therapy, it took 3 months to get the prior authorization approved on that.

If we can streamline that, I would be very, very pleased. Let's talk about how the connections work right now. Mark mentioned a couple of these things earlier, but it's typically in terms of trust on the connection, it's certificates, it's tokens, it's very manual. And often the org identity is assumed not proven. They have some evidence of who you are, but it's not cryptographic evidence. It's not something that would necessarily hold up in court.

With this idea that healthcare is the most targeted and that most of those attacks are coming through third parties this is a disaster waiting to happen. CMS has come in and said you need to stand up these four types of APIs. In reality, what that means is you're going to have thousands, if not hundreds of thousands of new connections coming into the ecosystem. And given that we are already the most targeted, we're just going to make it worse for ourselves because again 3 / 4s of all the attacks are already coming through an outside connection that you have. Let's talk about what FHIR has done and then we can talk about what we think still needs to be done. Semantic data: super important that you're all speaking the same language on your data transactions.

Real time versus asynchronous. You want something that's lighter. I think they've done a great job with that. The developer access and the discrete element parsing. This is again, I think the biggest one that FHIR has solved. You may disagree with me, Mark. There might be things more important, but I think that's pretty foundational to the value that FHIR drives. Really FHIR fixes today the how. They do a really good job of how are we going to get data moving back and forth. (Sorry, that's a med- alarm.) But what it doesn't solve is the who. Right now in the industry, Mitch, what was the last count on Epic lawsuits? Are we up to 15 yet?

– It's climbing. – Yeah. Okay. We're somewhere between 13 and 15 lawsuits. And that is both Epic suing people and Epic being sued by people. – Epic's the largest EHR [inaudible] Yeah. They're the behemoth [the biggest, red.] in the industry. What happened, and I'll walk you through this because I actually think it's really interesting, is a small startup called Particle Health sued Epic and basically said, "You're blocking information. We have a right to clinical information through Epic's EMR. We want to be able to treat patients and you're not giving us that info." They kicked off this lawsuit that kind of rocked the industry. The response to this was, yes, obviously Epic's taking this with their lawyers in court and they're having discreet arguments there.

But in the court of public opinion, Epic decided to go out and sue one of the on-ramps to the networks. Particle said, "Hey, you have to give us the data." And in the court of public opinion, Epic said, "We can't give you the data. The networks aren't trustworthy." They went and sued Health Gorilla, which is one of these on-ramps into TEFCA, that national exchange framework, and basically alleged that they were letting bad actors onto the network without proper vetting. Epic now can kind of hold their virtue in the court of public opinion by saying, "No, we're not info blocking. We're doing what's right by patients because if we were to put patients data out on these networks, it would be compromised because the networks can't be trusted."

And then of course because lawyers get excited, there has been a number of other lawsuits that have all started to spawn out of this same thing. You've got class action lawsuits that are go now going after Epic because this is a big issue. States are suing Epic around this same thing. And at the core of all of these litigious contentions is: well can we trust the other side of the ecosystem?

Coming all the way back to the lawsuit that kicked this off, if we could trust the network, then we could actually hold Epic to account and say, "You're info blocking. You're just not sharing the data because you don't want to because data is the new oil and because you see a lot of value in owning all of the patient records." But without that trust, we have to get into this incredible and increasing web of litigation trying to solve that problem at scale. I said this earlier today, I'll say it again. The question used to be, can we connect? The answer is yes.

Now, the question is, can we trust those connections? This is a slide I also stole from Mark. I did beautify it a bit. But if we are going to get to a place where we can say "yes" to the question of trust and connections we really need a couple of things. We need first to have a organizational binding to the individuals in that organization, be they people or entities. You have to be able to delegate and prove delegated access and authority to those individuals. And then you have to be able to particularly in healthcare scope down their access to just the minimum needed the purpose of use for that particular connection. If you can do all of those things, you can bind them together cryptographically then you can get to: yes we can trust the connections.

Now how do we get to that trusted onboarding the connections themselves? It's these pillars down here. You need something that's verifiable, auditable, automated, secure and open. And if you think of a there may be an open source open standard protocol that was invented by someone in this room [Sam Smith, red.] may actually fit the bill for this. In fact this may be me right now. If you understand KERI I think you know where this starts to go. We want to build and we talked about this earlier today. A trust equity framework or connectedness at trust. This is a diagram I entirely came up with myself. Phil gave no commentary on this at all. He didn't No, I'm just kidding. This is Phil's diagram for really how we see KERI and in particular the vLEI participating in an ecosystem where we can start to answer that question of can we trust the connection. It starts with the trust anchors. Go ahead. Okay, I'll go super quick. It starts with the trust anchors which for the vLEI is GLEIF. We see DirectTrust playing a huge role in this in terms of translation of global standards into value for individual ecosystems like healthcare. And then obviously SEDI because this is all built on the same stack plays very nicely with the rest of this. The way that the vLEI stack works is they are the Root of Trust.

They create what are called QVIs that issue a credential to organizations. That credential is called the vLEI. Once you have your vLEI, you can start to issue credentials off of that vLEI to individuals or entities within the organization. And again, all of this is cryptographically attestable all the way back to the Root of Trust. I now know who you are, who you work for.

In particular, I know what you're doing, which is where DirectTrust comes in. And it's the right things to be doing. And that you are chained back to GLEIF as a Root of Trust. Super important. The end result is, we can move from an ecosystem where connections take you know weeks to a day or two, and like you said earlier today Mark, the actual technical lift on this takes minutes, it's going to be a day or two to get the emails you know coordinated right that's all the lift that has to happen on these and we drop those costs down by 75% or greater per connection the result is of course that If we take away the attacker's favorite toy in this industry, they will move on to other industries. They will find another place to go and do compromises because the third party attack that is so common and prevalent today, no longer an issue. What do we see? We see a future where the question of can we trust this connection is answered by KERI through the vLEI. As you incorporate the vLEI into the FHIR standard, as you start to build network interoperability standards that require organizational ID, then you get to a place where you can start to say "Yes, the connection is trusted." And more than that, you can start to automate connectivity across multiple organizations. The initiative that we've kicked off for this is called the VITAL initiative. It's backed by CARIN Aliance, DirectTrust and GLEIF.

This is a group that's come together and said yes, we see this as a problem. We want to get this solved. I'll stop there. What questions do you have for me? – I'm writing up and I'm making FHIR end points to Epic. Do you foresee this protocol or this trust layer being the the intermediary in that FHIR call or is it a… I'm trying to understand at what level do you actually implement this?

– Yeah. We would see it as embedded within the FHIR API, the call itself. Every time you reach out and call basically you're doing a credential presentation. You're saying this is my vLEI and any of the relevant engagement context raw credentials underneath that, that say, I am Etna. I'm the sysadmin there and I have a purpose of use for payer-to-payer connectivity. And all of that is cryptographically handed to you as you establish the connection and then the data flows through a typical FHIR API.

– What about from the clinical parts – The next step for us really… this is the simplest version, right, we're trying to get to just, can we prove to the payers that they can connect here the next step, I think you're absolutely correct, is to say can we get this to the providers right and that becomes a much more complicated. – [inaudible] individual identity but you end up having to… a provider can work at three different places so what's the context in which they're operating, so that's where the vLEI, with a purpose of use, tied to a digital identity would be able to .., say, because you're working for Beth Israel as a provider, that yes you can request data about this patient you're treating.

- Yeah and there's going to be, I anticipate there will be a large fruitful discussion in the future around how we do credentialing for providers using things like this. Right now there's a very very expensive rent seeking happening which is the American Medical Association gets $300 million a year just issuing credentials to providers to provide care. And like we heard earlier today oftentimes that can take a couple of weeks to get things done.

Okay. And we are at time. Thank you [Applaus]

TALK XVI

Jared-Jefferey-Interview

KERI Conference April 2026 · interviews · 02:59

My name is Jared Jeffrey. I'm CEO and founder at HealthKERI. Really excited to be here at the KERI conference today. We just did a session on using the VLEI in healthcare to basically solve the problem of organizational identity. With healthcare organizations, there's really not a good identifier to help you understand which organization you're talking to. As just a quick example, if you talk about St. Mary's, well, which St. Mary's? There's 17 of them. And which St. Mary's owned by which organization? Because several St. Mary's are owned by the same organization. Several are owned by other organizations. Really what we're trying to solve here is can we bootstrap rapid connectivity using KERI and the VLEI? I think one of the things that I see maybe differently than the rest of the world here is that everybody else has kind of adopted now 25-year-old security technology tokens, OAuth, OAuth, things of that nature that were really never designed for an ecosystem that exists here today. We've kind of carried this legacy architecture into the modern world where not only do you have way more organizations doing connections, real time, but you also have things like not just AI internally being used in LLMs, but agentic AI. You have kind of created an entirely new entity class for connectivity. And I think a lot of people are still trying to take this round hole and fill it with the old square peg of last year's cybersecurity protocols. What would the decentralized identity system lose if all of our work at HealthKERI had evaporated? I think the easiest answer is as of today, we put a ton of code into the open source. One of the biggest complaints we heard about people that didn't want to adopt KERI or were slow to move into a KERI-based ecosystem was that KERI was too complicated and the infrastructure wasn't there. We recognize that a rising tide raises all boats. We just recently put into open source a production ready wallet that can handle KERI identifiers.

We put into open source witnesses and watcher code as well. Those three components are three of the five Ws that are being asked for from the KERI that are being asked for from the KERI Foundation. By putting those into open source, we're really putting a commitment into this ecosystem saying we want all of the parties that are interested in seeing KERI be successful come to the table and collaborate with us. If all of that went away today, you would lose probably two years worth of solid work from one of the best developers, Phil Feairheller on bringing healthKERI or KERI at large into existence.

TALK XVII

Open KERI Infrastructure

Jay Alexander Elliot

KERI Conference April 2026 · talks · 22:43

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Jay Elliot presents Locksmith, an open-source desktop KERI controller wallet stripped from healthKERI code. He argues genuine openness requires keys and control to stay local, with witnesses, watchers, and providers connected through a real plugin boundary rather than a single integrated stack.

This version of the Locksmith app is an open-source stripped down version of what healthKERI has. If you guys want to follow along, I have a open source repo. It's https://github.com/jaelliot/locksmith-demo and you can follow along. There's some instructions in the readme.

Locksmith is the desktop wallet for KERI. This talk is about the controller side of the system where the keys stay local. Users manage their identifiers and control does not disappear into a single provider stack. users need a wallet surface where the keys in control.

The trust root lives at the edge and non-hosted services. In KERI the controller is the part that actually controls the AID. That means that it manages the keys in the key event life cycle. If that user-facing wallet gets absorbed into one vertically integrated service stack, the system may still look open on paper, but it becomes less open in practice. That's why the wallet boundary matters is where the control actually lives.

Many services describe themselves as open, but in practice, the wallet, the provider, and the integration path are all tied to the same stack. When that happens, builder choice starts to disappear for KERI to remain genuinely open. The wallet must remain separate, and that separation must be real and verifiable. With that in mind, here's where Locksmith sits in the stack.

Locksmith sets the controller side wallet layer. Witnesses stay outside of it. The watchers stay outside of it. The connectors… the providers connect to it but they do not become the wallet. And by "providers" I mean the real external services, witness and watcher operators, credential services, workflow services, and other KERI-adjacent integrations.

The first thing that Locksmith proves is the local state. It is a real local identity vault. It supports multiple vaults and local first storage and controller-owned state on disk. That matters because portability could come later. The first proof is local control.

– What are you defining as portability here? You could deploy it to any location, I suppose? – Okay. Because you say state ownership for portability. I imagine that means the key state and everything necessary for like interacting with the service provider.

Is that what you mean by state ownership? Yeah. The identifier credentials and real user workloads is where the wallet becomes real for the user. Locksmith is the desktop surface where someone can create and manage identifiers, track credential state, work with KERI operations through a usable interface. That is what the wallet layers for turning core KERI behavior into something a person could actually use.

This is where the architecture stays open. Locksmith has a real plug-in boundary and code, providers sit behind that boundary rather than replace it. That means that the wallet can integrate with outside services without collapsing into infrastructure. What is real today is the boundary itself. What is in progress is the broader ecosystem around it. More provider side integrations and the KERI Foundation's work to carry that pattern through more completely. The architecture for now is real, and the full provider story is currently being built.

This is our architecture flow. The user signs and stores locally in the Locksmith vault. From there, the wallet can communicate with provider side services via the plug-in boundary. Separately, events go out to witnesses and watchers and receipts come back.

Verifiers stay out of that path, which is the point. Verification stays available without turning the whole system into one provider-owned stack. if somebody wants implementation detail, I could go there, at the highest level that is the overall shape.

Let me draw a clean line between what is real now, what is next. What is real now is the desktop KERI identity vault, local vault and the identifier workflows, a plug-in boundary encode and repeatable setup reset and launch evidence. What is next is the broader foundation plug-in follow through mobile and web portability and the browser adjacent work around that… I wanted the room to leave knowing exactly what the demo proof say and what it's still in progress.

And then for getting started with the… [Can we ask question now or at the end?] [At the end] And then for getting started with the code I have a demo repo that I've forked and I have commands for Unix, it'll work on Mac OS or Linux and for Windows; that script is just a PowerShell script you could execute.

Demo bridge. The Locksmith gives KERI a real control side wallet surface. I wanted to show the a smallest life path of this setup can honestly support. I'm not going to try to turn this into a full network demo is the… The work that we had to do to remove the the healthKERI code kind of made that part still a work in progress.

In order to install this just run the script where it'll detect if something's ready there. If it is, it'll get rid of it and then it'll set up the environment and then I'll deploy it.

– I came in late. This Locksmith is the wallet in KERI? – Yes. – What's the relation between Locksmith and KERIpy 1 that was in the session description? – The KERIpy 1 is that this is the KERI Py one's just my fork or KERIPy is Wait, the other one's my fork Sorry. For whatever reason, I selected the wrong command.

Python scripts demo day. Come on, please work. Sorry, this is taking a while. And then we could initialize a vault and a lot of this stuff is still work in progress. A lot of the functionality is not there because it was stripped out from the healthKERI wallet.

When we have the full functionality built out, it'll have largely at parity with the healthKERI wallet and we could create our own identifiers. We could set the cryptographic strength. I don't have a remote identifier that I could connect to right now.

Same thing for a group identifier. Issue a credential. No schemas to .., I don't have anything for that yet. Add a schema. And set this. I could change the settings for the cryptographic Key strength settings or other stuff for the cryptography. But this is the open source version of the code that you guys can go and explore.

And sorry, I'm just not good at public speaking. That's all I've got. What questions do you guys have? Yeah. Essentially is what you've showed us the KERI Foundation using the plug-in API with Locksmith to run against their own infrastructure.

Yes. In this demo. That's not in this demo. Okay. This is the open- source wallet that we got originally from healthKERI. We've debranded it. This is available now for anybody to go and fork this, download it, use it, build out more if you want other features if you want. The plugin itself is not shown here.

You'll see that in my demo, but you and I are competing with each other. Now we're not now we're not gonna now you're not going to see it ever. thanks man. Hey Joseph, does so does this sorry this is a dumb question but does this wallet here have the KEL on in this whatever device is installing the wallet does is the KEL also there you're local yeah local yeah locally so this creates the manages the keys and the KEL in that environment and you publish the KEL to a witness yes to one or more witnesses right so you would configure the witness here You'll see that in my demo if you come to my demo.

When did you demo? 3:15. It's big – And then also the TEL stuff is that ..? No, that's ACDC related. We're not going to cover that in this. Would it be on his local .., That's a good question. I don't know where theoretically, I guess probably, but I don't know that's a little bit outside of scope of dealt with yet.

– I know they defined the TEL this morning, but I forgot what it stands for. [TEL =] Transaction event log - And that's for the recipient to detect That's for the holder to detect if their credentials are being used in the non Not necessarily, TELs are specifically related to a different construct in the KERI ecosystem called ACDC, which is more about credential issuing.

Yeah. The benefit of the TEL you're publishing the transaction event so that you can I'm going to get anchored to the key event log I can't keep track of the state I think a good way is the KEL is the state of your keys and the TEL is the state of your credentials Question what the state of because they can be in multiple different kind of states depending on what the credential is .., – Like revocation status state Yeah.

I would guess that in Phil's ACDC talk tomorrow morning, he'll cover this stuff. Tells everybody if he's doing one then. Yeah. He's talking about ACDC tomorrow morning. If you have more questions about that, he might actually have production use cases that you could guess.

About the repos again. The Locksmith demo one is this one. And what's in the Locksmith one? That just that's the full healthKERI that's the healthKERI is proprietary. Yeah. we had to strip out all their fun features when they donated to us. But we still have like all the core KERI stuff like identify creation, There's support for like witnesses and watchers but in order to connect to witnesses and watchers you need to connect to a provider such as healthKERI through a plugin. The KERI Foundation is going to create our own plugin, where you can connect to our infrastructure to get witnesses and watchers and then in theory you could also have, in the same UI, you can see KERI Foundations plugin and like healtKERI's plugin or any other provider who wants to create witnesses and watchers.

[It can] provide infrastructure for you to Basically what you want is creating a vault Encryption storage right it's just a place to store keys. But it is like for per wallet you have multiple wallets. What is the reason for it?

Just in case you wanted separation for some reason. It's like one password and bit warden. Most of these password managers have the similar idea of vaults where you can say this is my work vault. This is my personal vault.

Okay. It's just applying the same belong to a specific vault. Yeah. You can't move them across. Yeah. That's I don't know. I don't think you can move them across yet. Theoretically, I guess we could eventually I can understand I have them.

Yeah. But here I'm trying to understand because they're very tightly coupled with ID, right? The credentials are very tightly coupled with ID. And when you have walls, I mean how would you move .., you can't move the credentials across because different ID I would guess that you could probably .., like if you created two vaults I have so I don't know. I'm guessing if you created like a local identifier in one vault, you could OOBI with it from another vault as a remote identifier so you could watch your own vaults. But I don't know why you want to do that.

The I was looking at the KERI Foundation GitHub. There's 16 repositories and it looks like there's three different wallet implementations. There's the Fort Web the No, there's the I think those are just different deployment w targets.

There's one for iOS, there's a web app, and one for Android. Okay. But there's the fort for Android and fort for iOS and then there's the key the Locksmith. Are those how are those related? The Locksmith is just what we take and deploy to the different targets.

Locksmith is desktop version. Okay. The Fort Web and Locksmith are they're both desktop versions. They're just different implementations of the desktop version. Essentially, all of them are a wallet, right? They're all the same. They're all wallet. You make identifiers. You can UI.

My question is, why would you pick one versus the other? Whatever platform you want to use it on. Are you trying to deploy to a desktop or to an iOS? But Locksmith is web. No, lockmith is just desktop. It's Oh, this is native desktop.

It's a PIDE 6 applic. Okay, that makes sense. Are they coming from the same? It's multiplatform source bundle. It's it's all it the web or they they all are getting deploy the same stuff the same code is getting deployed to all those different platforms. For web is just the web SaaS app of Fort also have multiple local identifiers right in a single one have multiple local yes that's just talking about issuing credential to many identifiers some Josh was talking about Evan be talking more about the web wall tomorrow or whatever you want.

Wow. People have other things they want to hear about or whatever suggestions. Basically, he's just going to stand up there and let you ask him anything. Ask me anything session with that. Cool. I think that's kind of it.

Do you guys have any other questions or I guess we're kind of still kind of pretty early tomorrow. I guess I mean the question I have that might be answered tomorrow probably in your session. Is it clear, based on what code that's open source what a provider has to do in order to do their own plug-in implementation and what they have to sort of implement on the infrastructure side for the plugin kind of.

Yeah. Because I want to do I want I'm going to do a blog about it and sort of show people how to do it. I can make my session how to make your own plug-in demo. That'd be great. It's pretty cool. A good idea because I will definitely be doing it.

I want to help other people learn too. I mean it's all there in the repo so you could obviously learn how it's built but I don't think we haven't done anything formally yet to describe how to build a plugin. Okay. That would be a good thing.

There is healthKERI shipped like a plug-in based flask. They basically like define the set of functions you need to define in order to make your own plugin. Awesome. That's what I want to know. Time to go through that sort of thing tomorrow.

Was interesting. There are other open source KERI wallets besides the Fort and Locksmith one, isn't there? Yeah. From so I'll talk a little bit about this in my session, but there's heavyweight wallets and lightweight wallets. Lightweight or Signify wallets.

This is a heavyweight wallet. It's the most complete one in the world by far. Health KERI's is HealthKERI's is I guess this is sort of a stripped down slightest. I guess health carries is the most complete and then this is stripped-down version of but it's the furthest of all but that's your session tomorrow not your session today.

No I have two sessions today. One is the last one is on the KERIA stuff about lightweight wall. The first one is going to be on the GLEIF Root of Trust. Cool. Wow. Well thank you. Do you guys have any other questions? – Who wants to speedrun building a provider Right now?

Let's do it. Evan can coach you through it. All right. I just copy our Done; 40 minutes? No. Sorry that this wasn't very long. Good. See, We got a little time, Jay, you can launch the iOS thing if you want. Sure. Just to show it.

Yep. This is the iOS app. Suppose I should It's functionally speaking just it's the same code just it's deployed to a the mobile apps are just wrappers that will run the Locksmith code it's just this functionally the same experience on any of the devices and that's right I have not fixed this yet.

I'm noticing that this looks very similar to the flat UI. It's And I know it's Pyite 6. Mhm. Yeah. HealthKERI basically just translated the flat app into Pyite 6. Okay. The flat app is, right? I just wondered if it was I guess did they lift the CSS styling when they This is CS. I just I left it as close as it to what it was.

Okay. Good jokes. Nice work. Thank you.

TALK XVIII

KERI Bridge to the DID World

Jonathan Rayback

KERI Conference April 2026 · talks · 40:00

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Jonathan Rayback, co-chair of the DID:Webs task force, explains how DID:Webs bridges KERI and W3C decentralized identifiers. Discovery uses familiar web DID documents; cryptographic truth comes from validating the aggregated KERI event stream, separating discoverability from verifiability.

Okay, well, let's get started everybody. Thank you for coming to this session. Hopefully, you'll find it interesting. Let's see if I can get my cur- whoa, my cursor. There we go. My presentation is on DID:Webs. My name's Jonathan Rayback.

I am a co-chair on the DID:Webs task force underneath the KERI stack working group at TrustoverIP. My other co-chair is Kent Bull here, is here with me. And I'm going to be talking about DID:Webs. I also am a chair of the DID methods working group at Decentralized Identity Foundation. I am squarely in the DID ecosystem and community.

And the idea behind this talk is really to talk about DID:Webs, which is the KERI entry into the DID:Webx genre of decentralized identifier. The talk today is going to be I would say pretty technical, and I'm going to really assume that the audience is fairly familiar with KERI. I'm not going to assume that the audience is familiar with DIDs. I'm going to sort of that's kind of the point of the talk is a little bit to describe DIDs and why we have DID:Webs and how it relates to KERI.

That's kind of I guess my prelude. Let's see if I can actually get this to… We're going to go through a little bit about DIDs. We're going to do a brief discussion on DID:KERI itself cuz there is DID:KERI came before DID:Webs and it is another DID method. We'll talk about DID:Web, the original DID:Web at W3C. We'll talk about DID:Webs and why it exists why DID:Web exists.

Then, we'll do a little bit of a conversation about the relationship between DID:Webs and DID:web. We'll talk about some other DID:WebX methods, the status of DID:Webs from a standards point of view, and then I'll do a little demo.

Decentralized identifiers or DIDs. It is a W3C official recommendation As of several years ago. There's a 1.0 core spec and I think they're working on a 2.0 spec now. You can click through to the actual specification there at on that hyperlink.

It's a type of identifier. It's formally a URI identifier that allows an entity to communicate things like cryptographic material keys, Verification methods, and or service endpoints like say an authorized agent. And this communication happens by resolving is the term the DID. You resolve the DID like you would resolve a URL. And when you resolve a URL, you get back a usually a chunk of HTML like a webpage or something like that, some resource. When you resolve a DID, what you get back is a DID document and the DID document will contain information like what are your current public keys, do you have any authorized agents, etc. etc.

The scheme for a DID is you have the mandatory URI scheme prefix, which is DID, the colon, similar to a URL. Like in a URL if it's the HTTP, you And then whatever. This is in this case it's DID, whatever. The DID method name, so in the case of DID:Webs this would be Webs. And then each DID method has its

own method specific identifier that's determined by the designers of the specification for that DID method. This is an example of an imaginary one.

You have, in addition to the DID document, the DID method itself describes how do you create, update, delete etc. the DID documents, and every DID method has its own mechanism for that, along with its own method specific identifier.

There are currently hundreds of DID methods out there in various states of standardization. Many are associated with specific blockchain technologies. There are lots of other creative ones. There are a few that deal specifically with web and web protocols and web infrastructure, of which DID:Webs is one.

If anybody has questions, by the way, at any point, feel free. Oh, there'll be time for questions at the end, hopefully, but if you just want to ask a question, ask a question. Okay, cool. Decentralized identifiers, as I mentioned, there are four method operations that every DID method must explain. If you're going to create a new DID and then have a new DID method, you have to explain at least how to do these four things. How do you create them? How do you resolve them?

How do you update them? How do you deactivate them? Then the DID document itself is always called did.json, so it's a JSON format following the specific DID core JSON scheme. And every DID that's compliant with the DID spec is going to have a DID document did.json, and it's going to follow that particular scheme, right?

Including DID:Webs. DID document has specified properties, including and I listed these out cuz these are the ones that are particularly relevant for DID:Webs. The subject of the DID is in the ID field. The DID can have other identifiers that it calls out as equivalent IDs basically, "also known as" You'll see that in a minute. It has Verification methods Which is basically the list of the keys that the DID controller is publishing the DID document. It can have verification relationships, for example, authentication saying this key is authorized to authenticate for me or this key is authorized to sign on my behalf in these two cases.

You can also list out various services that are relevant to you as a controller. DIDs .., DID identifiers do support parameters like URLs do, so you can have a DID and then like a question mark and a parameter on the end that allows you to adjust the kind of information you're going to get back from the resolver.

And there's also metadata that comes back when you resolve a DID. You'll see that later when we do the demo. This is an example DID document. This is pulled from the core DID spec. You can see in this case the subject is this DID.

And this particular the controller of this DID has authorized this key which is an Ed25519 key. This one in particular. they've authorized it to serve as authentication method. It's basically what this DID document says.

Okay? Questions so far? Yes. – There's no “also known as”… In this one. Also known as is a optional field in the core spec. Technically, authentication is also an optional field. In DID:Webs is not an optional field, and that's why I called it out earlier cuz DID:Web asset does makes has certain additional requirements above and beyond the core spec, okay?

I just wanted to cover this really quickly. I added this morning cuz like if I was in a KERI conference, I thought someone would ask what do I DID:KERI. There absolutely is a DID: KERI, okay? There's a specification for It hasn't been touched in 4 years.

It's mostly stub, I would say. It's like got all the sections, but each section kind of says, "Need to fill this in, "right? Like it hasn't really been filled in yet. It's very early draft, very draft, right? A DID:KERI is very simple. It's just, the prefix, the name of the DID method, which is KERI, and then the AID. That's a DID: KERI.

The thing about DID:KERI is, it is entirely silent on the topic of discovery. How do you discover the KEL essentially? How do you .., cuz like if you know the KEL, this is totally sufficient for you to go do whatever you need to do, right? You just you have the ID, you go to the Key Event Log, you can do whatever you need for validation. But the DID method itself doesn't give you any hint on like how am I supposed to get this KEL? And I think this spec was actually written pre-OOBI world.

Like they didn't have the idea of OOBIs even yet. Now you would say you're probably just going to get an OOBI or something, right? But in the day we just don't have this. It has limited practical use is what I'm saying, right?

In the real world today. Because you are going to have to negotiate how to discover the verifiable history outside of the method itself. Okay, DID:Web going to the other side of the world. In the W3C, right, it's the web specification, web standards organization, they thought, "If we're going to have these DIDs, we should have a DID:Web. That would be amazing, right?" Yes, there is a DID:Web and it's an unofficial draft right now at the W3C, but it's used and it's used in production. People use DID:Web. When when I was at Gen Digital, the solution we were developing was built around DID:Web.

The thing about DID:Web is it's all web technology, it's all web infrastructure. The way that you resolve a DID:Web is you turn it into a URL and resolve the URL in your browser, with Curl programmatically, with a headless browser, whatever, And you get back did.json. And that's amazing. It's a very very extremely limited practical use because there's no history of that DID document.

That DID document could change at any time. It's subject to any of the exploits that are common on the web, man-in-the-middle attacks, hijacking, DNS, like any of these things. It's completely subject to all those problems. When you get a DID document, you have no way of knowing that the key that's published here now, frankly, is the key. But you also have no idea of knowing this controller at some point in the past may have had other keys, may have signed things with other keys.

Don't know. None of that. You know none of that with a DID:Web. Yes. – I wanted to plus one and add a bit of bit of history. I was one of the co-creators of the DID:Web spec and it was always meant to be training wheels.

Right? This was like, "Hey, you're just dipping your toe into DID world. You're going to try to do key and then try DID:Web like to resolve it over the network." But, it has all of the downsides that you just mentioned. Right? There's no key history.

There's no cryptographic binding of that identifier to the key, all that stuff. The funny part is, so I've at least always meant to add that to it. But, what happened was… I was one of the authors of this spec. And I opened an issue like, "Okay, now that we've gotten it out there, let's add cryptographic binding. Let's add key history."

And the other developers were like, "Nope." And I was like, so basically there was a lot of pushback against that. They're like, "We're going to keep it super basic. We're going to leave the binding and the history to other methods."

This is… That's basically what ended up happening. Hence, DID:Webs and all that stuff. – Thanks. Appreciate the history. This is a the main thing I wanted to show here was kind of just give a bit of background, but then if you look at the syntax of a DID:Web, this is like relevant as we move into DID:Webs and other DID:Webx methods. You're going to have, basically the method specific identifier for a DID:Web is looks very much like a URL. You're going to have your host information.

You can have an optional port information. It has to be percent [%] encoded like this with URL safe. And then some path to whatever the document is. This would then translate into a URL that looks like this, right? Your host is example.com on port 3000, path to document did.json. You resolve this, you're going to get back that JSON DID document.

Because of DID:Web having the shortcomings that it has, a number of designers have proposed solutions. DID:Webs is one of them. I think it was really the first, honestly, and then some others spun off, but DID:Webs has a specification. Kent and I are responsible for it. We're working on it.

But what DID:Webs does is it separates discoverability from verifiability. We use the web to discover the information about the identifier, and you can use the DID document to get the keys and all of that. But, if you want to verify the information in the DID document, you do that via KERI events through something we call the KERI Event Stream, the KERI Event Stream, which is kind of an aggregation of various KERI events from various sources that may be relevant to the controller, and that's aggregated into this KERI event stream. To resolve a DID:Webs, you actually are resolving two resources, two endpoints. You resolve the did.json and get the DID document, but then you also resolve the KERI CESR, and you have to validate that just like you would any other KERI Event Stream. We have a question back here. – I just wanted to ask you is it possible to get the copy of the slides? – Yeah. There's a Yes. There's a little QR code at the end if you want to just scan it.

– All slides all presentations. – I must have missed that in the beginning. – It's more widely adoptable than DID:KERI because, you It's got existing web infrastructure. It solves the problem of how you do discoverability.

And it keeps the security properties of KERI. It provides way better verifiability than DID:Web certainly, but also than other DID:Webx methods. And we'll talk about DID:Web versus other DID:Webx methods a little bit at the end. But there are properties you get from DID:Webs that you don't get from any other DID:Webx method.

This is a DID:Webs identifier. You can see it looks super similar to a DID:Web identifier. The only difference is the last segment of the method specific identifier is always the autonomic identifier. It's always the AID of the controller.

"Da da da", path, you've got the host, you've got the port, optional port, you've got the path, and then at the end you've got the identifier. In order to resolve this, you're going to turn this into essentially a URL up to the path, and you're going to go get the slash did.json and the slash KERI.cesr, and the very first thing you better see in that KERI.cesr is the inception event for that AID.

If you don't see that, you're already no go. No bueno. Right? – That path is not is just to a web server that's serving the redirect. It's not necessarily the KERI watcher. – It's not the KERI watcher. 100% not the KERI watcher. It is where you go to get the Ultimately, it's where you go to get the KERI.cesr stream.

But it's also where you can for convenience get the did.json file, because it's a DID. By specification, you have to have a did.json file there at that endpoint as well. – And then with that DID.keri stream, that's where you're going to confirm against the watcher to make sure that it verifies correctly.

Sure. KERI.cesr, you're going to take the information from the KERI Event Stream With the file is called KERI.cesr. It's a CESR encoded stream of KERI events and you're going to use that, sure, against the watcher or whatever validation you're going to do. What's going to happen first is your DID resolver is going to validate that. That's what a DID:Webs resolver is supposed to do is check that.

– I mean I'm not hearing so I'm trying to understand. The CESR file is the source of truth for that Key Event Log? – Yes. – Okay, it's a – The CESR is the credential encoding format used by KERI. Correct? – CESR, it's a streaming it's a streaming protocol.

Which includes encoding, yes. Given that we are reversing we're giving responsibility to the controller for his/her keys? – Yeah. – Is there a protocol that, given this exception log, gets pushed to the domain of some DNS host or I mean a website?

Or… – Yeah. – I'm looking at it from the custodian perspective. I have my customers, I assign them a DID:Webs DID. – Mhm. – But then it breaks the whole purpose of KERI, right? Where my controller is… – No, we'll talk about that in a minute, actually. But theoretically you could provide a hosting platform for individuals to have a DID:Webs. You could certainly do that. – And the individuals have to push their own?

– You could do that for them, but there's a mechanism that DID:Webs has that is intrinsic to KERI that allows the controller to authorize that endpoint as the definitive endpoint for their DID. And we'll talk about that in just a minute. And that's actually one of the unique features of DID:Webs that other DID:Webs methods do not have. And this is maybe a good point, too, to bring out the fact that in DID:Webs, and it may seem like a minor kind of detail, but it's actually functionally has some pretty important ramifications.

DID:Webs, the DID document is always a derivative of some KERI Event Stream. Always. There's no idea of like I'm going to go write this like type out my DID document or something like that, right? The DID document should always be generated from some KERI Event Stream, and the current key state and all of the other events that have happened in that have a consequence on what shows up in the DID document. This is reversed from other DID:Webx methods like DID:Webvh, where it's kind of the other way around.

The DID document and the state of the DID document creates the verifiable history as hashes. It's the other way around. And there are some ramifications for this that come up. It's a less nuanced differentiation than you might think.

– Are you going to cover those ramifications? At the end, yeah. We'll talk about it for a little bit, yeah. – You just piqued my interest. DID:Webs has some features that just come because it's KERI based. You get multi-party signatures.

You get delegation. I want to be clear this is really clear for people who are in KERI land, there's kind of there's two kinds of delegation we can talk about, right? There's ACDC-based delegation that you get through like vLEI, where you get this hierarchy of credentials that go back to a root credential. That's not what we're talking about here. This is AID delegation. This is where my identifier is specifically saying, "And this identifier is allowed to act on my behalf". Okay? That's a different kind of delegation. We can encode that into a DID:Webs DID document. It supports services that are relevant to KERI like OOBI endpoints, witnesses, agents, mailboxes, and it supports designated aliases through ACDCs. This is an

ACDC delegation concept, but this is where for example it's required in the DID:Webs spec that if I'm going to have my DID:Web… I, as a controller, am required by the spec to authorize the host and path information that's going to be in my identifier with an ACDC credential or it's not a valid DID:Webs.

Again, that's another differentiating point from other DID:Webx methods because they give you no way, other than implicitly, to say this is the correct host and path information, like when I give you my DID:Webvh and it says example.com in it, implicitly I'm telling you I've authorized this. In a DID:Webs when I give you Moisey's service.com or whatever, it is you're hosting for people.

The person who's verifying my DID gets to go through my KEL and gets to see that I actually did at one point give you the authorization to host that at this location. This is the only DID:Webs method that provides that capability.

– Sorry. The DID controller's KEL? Where does it live? They live in… the authorization? Are you asking where does what live? – The controller's KEL live. – The controller's KEL lives wherever the controller's KEL lives cuz it's KERI, but the information from that KEL that's relevant to the DID:Webs is aggregated into this KERI Event Stream and encoded in a KERI.cesr file which lives on your server cuz you're hosting it.

Kent. – May I add something to what you said earlier? Earlier you said, and this is important for those that are new to DID:Webs, you said the DID:Webs DID document is always derived from the KERI Event Stream. And I think it's worth highlighting the fact that based on what you just barely said about the path information being in an ACDC, that the DID document will always be derived from an ACDC as well. You've got the KERI Event Stream with the KERI event logs, and actually the TEL and ACDC.

Cuz when I was first learning the DID:Webs spec, I was like, oh, is there a… Does it make sense to just have a KEL cuz I that was simpler. I thought it was a good idea at the beginning. Then as I got more familiar, I thought, you know what? Actually, cuz my goal was maybe we can have a DID:Webs version that only depends on KERI, not KERI and ACDC. I tried to simplify it.

But actually, when you look at the security characteristics that you're going to get into in just a minute, Jonathan, you want there to be an explicit authorization of the path segments of the whole DID. It's important to realize that the KERI Event Stream I think it bears repeating that the KERI event stream will always include at least one ACDC.

It has to have one ACDC. You cannot have a DID:Webs unless there's at least one ACDC because at the very least, you have to authorize the host and path information for the DID:Webs. Period. That's all. – Why is DID:Webvh inherently unsafe? – It's not inherently unsafe. It does a lot of things fine. One thing it does not do is it doesn't let you as a controller explicitly authorize host and path information for your identifier. It's always implicit because like I'm going to give you my DID:Webvh and it's going to say rayback.com and because I gave that to you by whatever mechanism, you're going to assume implicitly that I've authorized that.

KERI says no, I'm going to actually validate that because I'm going to check to see if you issued an ACDC authorizing this cryptographically. That's an explicit authorization versus implicit. – That makes sense. I just want to add Webvh implicitly authorizes, it not just by handing over but by signing. Right? There's an… The identifier.

– The identifier. There's an implicit authorization signed with the [inaudible] that serves as a… Yeah. True. And I might as well just get to the point cuz I may not get to the slide cuz I realize I'm running out of time.

But like the other thing that DID:Webvh does not do as well as DID:Webs is the long-term long-lived multi-decade identifier problem. The spec really tries to make sure that it's possible to do that but it gets quite awkward. If you're going to change like hosting, you're going to change path websites over time.

And this is where I think that stems from the fundamental difference. Like you're essentially going to publish a new DID document with new location information in it and then you have to kind of go and make sure that your verify history. A: it got transferred from one domain to another, B: that it like has all of the steps in it, that nothing got missed. Like that's all really important. Managing that over time is much less practical than a DID:Webs because it's the other way around.

In DID:Webs all of this is happening by stuff in the KEL or the TEL or one of the KERI artifacts which ends up in the KERI.cesr and therefore your DID document is always going to be like reflective of the actual current state of things cuz it's derived from the KERI event information. – They tried to do KERI light and now they have to bolt on.

They have a whole section on witnesses and watchers now, This is the also known as I want to get into this really quick. The spec requires there to be an “alsoKnownAs” section in every DID:Webs document. You must list the equivalent DID:KERI as part of the “alsoKnownAs”.

At least there's going to be a DID:KERI in every DID:Webs document cuz that's considered an “alsoKnownAs”, it's always by definition any DID:Webs is also known as DID:KERI whatever the ID is. You may, if you choose authorize a DID:Web endpoint.

That would happen by ACDC, the same ACDC that the DID:Webs identifier shows up in. You can authorize a DID:Web endpoint, which means you're authorizing a DID:Web resolver to go resolve the DID:Web version of the Identifier for interoperability or for whatever reasons you want to you want, but like you have to explicitly authorize that and if you don't authorize it nobody should assume that like it's DID:Web resolvable. You have to look for the actual identifier in the DID document to know that.

Verification method. Okay, this is what a verification method looks like in a DID:Webs DID document. This is all standard one of the various options that's in the DID standard. We support JSON web keys. We support Ed25519. There are two or three other .., Whatever key public key are supported in KERI, we support in DID:Webs.

verification relationship. This is optional. It's optional to have this in your DID:Webs, but the DID:Webs spec does say specifically if you are going to include verification relationship any key that you publish as a public key must be authorized both for authentication and assertion. That's the one thing. If you're going to have it, you have to have both.

Again, because KERI itself any keys that you publish are for either authentication or assertion. We support these services, these KERI services, which would look familiar, I think, probably. Looks something like this in the in the DID document. This service is a witness service. You can get to it at this URL or this TCP endpoint.

Parameters, as I said, you can have parameters on the end of a did identifier, just like any other URI. For example, this one has a version ID parameter on the end. This lets you tell the resolver that you want version one of the… You can go back in time and derive the DID document that existed after event with sequence number one occurred in the KEL, right?

We also support this transform keys extension, which just… I don't actually think we have this implemented in the resolver right now, but like theoretically, the spec says we should support this. This basically just means if you pass this in as part of the parameter, we're going to transform all the keys in the document to that format for convenience. This is for convenience for people who when they resolve the DID.

– What other formats would you use besides CESR? – They're like the native format that we support is this JWK. – Okay. – JSON web key, right? – I opened an issue on this. There's public key JWK and public key multi-base. – those are the two. – I'm already using public key multi-base in the W3C cross walk, and so it's going to be added to the resolver. – And the spec says we should be supporting public key CESR as well. But, then there's document metadata.

This is where you get into some like I'm not going to get into much detail here, but the main one has to do with this equivalent ID. There's a bit of a distinction, but the idea is anything that should show up in the equivalent ID in the metadata. This is not in the DID document itself, it's in the document metadata.

Should be other DID:Webses that you've authorized. And again, this is going to happen via some ACDC. But, it's possible you have a DID:Webs example.com, but there's an equivalent DID:Webs foo.com. That's fine. You can have that. It has to be authorized by ACDC. Mhm.

– Why “equivalentId” and not put it in “alsoKnownAs”? This goes back to the DID core spec. The idea is equivalentId are equivalentId of the same DID method. If I have a DID:Webs identifier, any other DID:Webs identifier that I want to call equivalentId goes in equivalentId.

If I have a DID:Webs identifier, it might also be known as a DID:KERI DID method, it might also be known as a DID: Web DID method. You have It's more inclusive. The alsoKnownAs section is meant to be more inclusive. The equivalentId is meant to be within the scope of that DID method because it's meant to be processed by a specific resolver. That's going to be a DID:Webs resolver, right?

It could be that. It could be for any other any number of reasons, but… – I can give an example there. We like to do some work that's called up with DID:Iota and the twin. The DID:Iota Foundation. And the integration point, the initial integration point between DID:Iota and DID:Webs was just through mutual alsoKnownAs. From their DID document, they put the DID:Webs And from our DID document, we put the DID:Iota in the alsoKnownAs sections.

– I'm going to go through this really quickly cuz I want to do the little demo and I only have about 5 minutes. This is important though because this is something we had to like deal with because of KERI cuz there was no way to deal with this in the DID document.

When you have multi-signature. This particular verification method type, ConditionalProof2022, there's a spec for it at W3C. That's what it looks like. This is the… This is actually the KERI Event Stream. You can see here that there's a threshold. It's a two out of three threshold involving these three keys.

When you get to the DID document itself, it looks like Oh, this is delegation. When you get to the DID document itself, it looks like this. You get a verification method, ConditionalProof2022. Says there's a threshold of two, here are the keys, and then here are the keys listed. We'll look at that more detailed later.

And then this is a delegation of the AID itself. It just basically, you have to provide the delegator OOBI so that the person can go look up the delegator's KEL and validate that the delegation event is in that doc. Okay.

I'm going to really quickly. Specification status, there's a bunch of stuff we've done. We've registered it all the right places. There's a bunch of like. There's a final draft version that's being reviewed at under the KERI stack working group. There's a new IANA media type registered. We also are in the final review stage at DIF to become a DIF recommended DID method.

Okay. We're going to do the demo here now really quick. – Five more minutes. – That's why I want to do this. This is pretty. This is more like an illustrative demo. Why? Oh. Okay. What we're doing here is we're just going to kind of start doing some KERI stuff. The first thing I'm going to do is I'm just going to accept an AID, right? I've created this AID.

It has that public key. This is what the KEL looks like. This is what the KERI Event Stream looks like. Okay? You've got the the AID, you've got the key. At this point, I cannot have a DID:Webs. Why? There's no ACDC authorizing the host and path information. That's the next thing I have to do. I have to issue the the ACDC.

Here's the ACDC that I just issued that's authorizing for this identifier these two identifiers. For this AID identifier, these two DID:Webs or DID identifiers. One is a DID:Web In this case, I'm authorizing a DID:Web endpoint as well in the correct format.

Now I have this Key Event Log, this KERI Event Stream that includes the ACDC right here. Right? That has my host and path information committed to, right? Now, I can create a DID document. This is what DID:Webs DID document would look like. This is almost the most simple possible DID:Webs DID document.

I don't have to have a witness service if I'm working in direct mode. That's optional. I don't have to have a DID:Web identifier. Beyond those two things, this is about the most simple possible DID:Webs DID document you could have.

Okay? Now, let me show you a multisig example. I'm going to set up a multisig group. And they've got to do all the chatter back and forth to get each other's KELs. Okay, now I've created a multisig group. It involves three identifiers with each having one key.

You can see here that the keys are given each one half weight. This is fractional weight using KERI fractional weights. That means that there are three keys and each key gets half a vote. In order to have a full vote, two of the three have to vote, right? Two of the three have to approve. That gives you quorum.

That produces a DID document that looks like this. I've got my my identifier. I have my alsoKnownAs. Now, here's my verification method ConditionalProof 2022. It correctly determines that two out of three, so two of these three keys, which are listed in detail here, have to sign in order for something to be valid, right?

And then we get our cute pony. CUTE PONY. There's that. And then let me really quick just show you in the last minute or so. We have We have the… This is the Universal Resolver and [inaudible] the Universal Resolver. This is mostly Marcus and his work that he's done on this. It's pretty amazing. Kent wrote the driver for this. We have a DID:Webs driver that goes here.

Let me grab my DID:Webs identifier here. I'm going to resolve this DID:Webs DID at the Universal Resolver. There it is. Tells me where the witness is. You can look at the DID document, see the DID document. You can see the document metadata where you should get the equivalent IDs.

Okay. And this is where you cannot because DID:Webs has the unique property of the KERI.cesr that's not available on the Universal Resolver page. If you want to see that. I can resolve that. Just by Curl here. And you'll notice too that this CESR that's getting printed to the terminal.

Looks really great. This is a tool that Kent wrote recently here that's a really nice that annotates and prettifies the CESR stream. This is the aggregated chunk of KERI events that we call the KERI.cesr, the KERI Event Stream that created that DID document.

You're going to go all the way back you're going to see an inception event here. Somewhere in here you're going to notice that there's going to be an ACDC. It's very bottom. Right here. That authorizes the endpoints. There you go. That's what I got.

Time up. [applause]

TALK XIX

Jonathan-Rayback-Interview

KERI Conference April 2026 · interviews · 01:26

Hi, I'm Jonathan Reibach. My presentation was on DIDWebs and the connection between KERI and the decentralized identifier ecosystem. I think KERI offers some unique qualities that really help DIDWebs be the strongest version of the DIDWeb family of identifiers. Specifically, it allows for the controller to explicitly authorize information like host information, path information. And it also allows for much more convenience when dealing with very, very long-lived identifiers like over the span of decades, a lifetime. I think those two advantages are really strong for DIDWebs. I think if the work that we're doing on DIDWebs disappeared tomorrow, it would be too bad. I think that there as long as there's a decentralized identified identifier market and infrastructure out there, having a bridge between that ecosystem and the KERI ecosystem makes a lot of sense. There are going to be consumers of KERI information that want to get that via a DID document because their infrastructure is set up that way. And I think if we lost DIDWebs, KERI would lose a foothold in the interoperability space. That would be too bad.!

TALK XX

KERI Developement & Deployment

Keanu Pahio

KERI Conference April 2026 · talks · 23:37

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Keanu Pahio walks developers through local deployment of Locksmith, witnesses, and watchers using Python and Libsodium. He covers wallet basics, the KERI Foundation plugin for onboarding and witness registration, and how PluginBase enables custom Locksmith extensions.

I'm gonna start and talk about KERI development and deployment [checks] [Understand the architecture for deployment understand the role of each part Deploying local witnesses, watchers and wallet And to give an idea where you] can start for development on your own and that includes maybe doing your own plugin for Locksmith and developing your own API for witnesses and watchers. I want to talk first about the KERI infrastructure that we have.

[inaudible - We have the Locksmith wallet for signing, witnesses] and the watchers which is pretty much you've seen this the diagram a lot in the past few days, but for the requirements, the only thing that we really need is Python. We recommend the version 3.13, [because] 3.14 had some issues with some repos.

3.13 should be fine. And then [we need] Libsodium because this is what KERI uses for its cryptographic library. First, we'll go with the wallet. It's a Locksmith. The role of Locksmith on this wallet is to generate the AIDs to sign it, to have the key pairs secure and to store and verify through your KELs and finally the OOBIs.

Nothing more, I think, to expand on, It's the basic function of a wallet. And then we're going to go through deploying it. We have just three parts that we're going to go through individually and first I'm going to show you the repo and how you can set up the environment with its dependencies and how to run it.

Everything, every deployment that we're going to go through are going to be local. I can explain that for a while later. First you have the public repo for the Locksmith which is on the KERI Foundation [github repo]. Note that what I'm going to use in this presentation for the demo is from a branch of this repo which is the KERI Foundation branch. It's not the main branch that I'm going to use because the KERI Foundation branch has the KERI Foundation plugin that we're going to talk about later. I'm also going to try to get your attention for the docs that are provided by the repo and I'm just going to show you a bit through those docs. The developer guide in those docs explain what I explained before. If you need a reminder, you can just go to the repo and then read it. But first it's going to be Python 3.13. And then to set up your environment, you just have to follow the commands that are on the developer guide. Pretty easy.

And then finally to run it, you have the Qt resource regeneration. And once you copy paste and run those commands, you can run this one. Then hopefully a lot of initialization logs will pop up and you will have your Locksmith.

I'm going to run it myself, and show you. I'm going to activate the virtual environment [preparing his demo] Here we have our Locksmith. I just created a vault, but we can… I make a new one… call it presentation. No passcode and then, here you go.

Here… – A vault is just a logical grouping of of AIDs or ..? Yes, if you're familiar with the KERIpy code, it's the same as a Habery which is where you host your AIDs and which is where you have your key pairs and all that stuff.

– Okay. Here you have the KERI Foundation plugin which we're going to talk about now. The reason why we have a KERI Foundation plugin is because of a boot server that we have. The boot server, what is the boot server? Well the boot server is kind of the in between between Locksmith and the witnesses and watchers. And the role of the boot server is to provision the witnesses to your wallet. And now, maybe a question that you might have is why can I not just talk from Locksmith directly to the witnesses and watchers? The reason for that is the witnesses and watchers API.

When you look here, for example for the witnesses API, you have actually two servers: you have the boot server and then you have the witness server. The boot server, if you pay attention to the API, this is the one that actually creates the witnesses and deletes them. And the problem is that the boot server is private. It's not publicly accessible. Even if you try to send a request to it, it will not respond.

But the witness server and the watcher server, those two are public. Once you have your witnesses and your watcher you can publicly access those however you want. And same for the watcher API this is also private. The reason is that we have our boot server which serve as the one who will call the private APIs to get the witnesses and watchers and the KERI Foundation plug-in acts as a bridge between the Locksmith wallet and the boot server. Currently the KERI Foundation plugin is in charge of onboarding, witness registration, watcher resolution and auth management. And when we're talking about onboarding we're talking about setting you up so you can have your own witnesses and watcher.

Now an important thing to note is that for, at least locally, the boot server has to be on the same machine or server as the witnesses and watchers so it can talk locally. As I mentioned before because the APIs are private.

Let's go through the plugin before we go through the boot server. I want to go to the plugin and how you could, if you wanted, make your own plugin. If you go to the Locksmith repository and if you go to “Locksmith/plugins/base.py” it has a bunch of classes that allows you to create your own plugin. If you look at the “PluginBase,” it has a bunch of methods that you can override to create your own plugin.

For example, here is the KERI Foundation plugin that is in the plugin folder. You can also look it up on the KERI Foundation branch. It is using the plugin base, the witness provider plugin and the account provider plugin.

Then it has u for example here you can see that it's overriding the plugin_id function and the initialize function. I'm not going to go into too much detail as to How you can .. or What you should put into into those methods, but at least know that it's there. And if you need the preferences you can just look at the KERI Foundation plugin in the repository to get some inspiration.

Now I want to go to the boot server. To remind you, the boot server serves as a public API to provision your witnesses and watcher. The kf-boot is also available publicly. If you want to look at it, you can. It's on the KERI Foundation repository. You can look at it and see how the code works.

To run it, very simple, create your virtual environment. And then here are the variables that your KERI Foundation boot needs to talk to the witnesses and watcher. I'm going to just show you a demo. Here I have my KERI Foundation boot server. I'm going to activate it. And then just like in the slide that I showed, I'm just going to copy .., I'm not going to install the dependencies since I have already done it.

I'm going to go here, paste these, and then run it. Now you can see that it's running here. And if I go to my Locksmith [needs to refresh]

I also need to give Locksmith the address for the boot server. Here and then if I run it back [then] my Locksmith can get in. [Keanu checks and prepares] I actually can't see it, but usually if you look it up on the KERI Foundation plugin, it should show that you can service an AID and connect witnesses and watchers to it.

– You could try deleting your local KERI “home/.keri” and then start fresh and then should recreate on that. Oh, I make a whole other repo? I could… – Go ahead. I could do it at the end probably. I just want to go through the whole slide and maybe I'll do another demo. Let me finish the slides first. Go to the witnesses, of course you have the role of the witnesses and what they do to anchor the events and trying to keep the authenticity and make sure that everything is correct of course, because they're the witnesses.

Same thing, we're going to go through the repo, the environment and how to run it locally. First you can get the code on the witness repository in the KERI Foundation. It also has docs. I have the developer guide. Same thing just make sure that you have Libsodium.

How to set up your environment. And then one thing about the configuration: If you were here last time Ryan (Hansen), you talked about it and how it works. I'm not going to repeat it but surely the curls here will be the address that the witness server is going to use.

To run the witnesses, you can use the first one or I usually use the other one. Let me show it here. I have it here. Let me activate the ..,

You have my witnesses running and if you look at here you can see the [private, red.] server address, I mean the port which is 5631 .., Sorry, the private one which is internal and then external which is the public one at the 5632 port which is the default one unless you change it in the configuration and then you can see your witnesses and their own AID.

Now to develop with witnesses you have the API which is given and shown in the readme file of the repository for the witnesses, you have different, of course, action POST [action] DELETE but I'm going to go into the POST and here basically we have the the whole POST function it's a Falcon app so it's in under the witness collection class. And if you look at it closely, you can see that when it receive a post method, it will first check for the AID, check that it's a valid AID, and then it just creates a witness, gets the OOBIs and sends the data back. That's all it does. If you wanted, you could check for, I don't know, check for more things. But if you want to start developing witnesses and changing maybe how they behave, this is where you would go. This is just an example of the 'on_post', but there's also the 'on_delete'. There's also all the the public actions and API that you can change which would also be under under this. It's in ”src/witopnet/witnessing.py”. If you want to change anything go here.

Now the watchers, same thing that we're going to go through: public repo of the watchers in the KERI Foundation [github repo]. This one doesn't have docs yet. I suppose we might either add it later. But that's from the readme.

I don't think we can install from Pypy yet. Correct me if I'm wrong. We can't install it from PyPy yet. Just clone the repo and then install the environment or hopefully from a virtual environment. Same thing for the configuration. The curls is also the address from which the watcher server is going to respond.

[Keanu prepares] But here you can see the same thing as the witnesses. It shows your address, public address for the watcher server, private and also the watchers running. I can probably show it better on here. You can see they have the watchers here as well as their AIDs. You can see that it's going up because the witnesses are running. They're all checking the witnesses.

A quick note that is important is that the witnesses and watcher are supposed to run together. They're not supposed to be individually running. They have to be together. First would be the witnesses and then the watchers have to be deployed together.

Same thing for the watchers. You have the same API and you can find the functions for them in this file here. I took the POST for the watchers that provision the watcher. To finish: what we have is this architecture technically that we have the wallet that can make public API requests but mostly if you want to just have witnesses and watchers provisioning it would be through a boot server that then talks privately to witnesses and watcher server in their own environment.

I think that's all I have. Unless do you guys have any questions? Yes. – Thank you. Given that KERI is a protocol and again protocols not companies. Why would you want to make changes to witnesses or watchers? I thought that was like common infrastructure or am I wrong?

– Like for example in this? – Yeah. – For example, you can add some more checks or you can have… we were talking about weight limiting for example on how much a person could provision witnesses. You could limit it or you could expand it.

Also another thing that you could note for example for the plugin if I can find it. I think it was here. For example, for the KERI Foundation plugin, all it does right now is just onboarding. It's pretty simple. But if you really wanted to, you can make your own plugin where you include payment. People go through the payment first and then it would provision your witnesses and watchers.

– And your watchers can make sure the money was transferred. – Yeah. It could go through the auth authorization for some account account management and approvement. – Approval. Approval. That's the word. If you want, I think what would be more interesting is probably doing your own plug-in first because that's when you could decouple that from the witnesses and just handle everything in your own plugin.

– One thing to note also that this witness service that we're talking about here doesn't necessarily, if you're manipulating in that 'on_post' handler, you're not necessarily manipulating how the witness code itself works [inaudible] you're just manipulating the layer, basically the API layer of so you can expand… – You're really not modifying any other… – Just adding extra functionality, auxiliary functionality. – Of course being open source if you wanted to go in and manipulate the witness protocol some way for your own version, that's on your own.

If you look here for example… This is all it does, just calls create witness but then everything after, you can probably add more functionalities for yourself for example for tracking or for like stats if you wanted to but that's up to you. I'm just trying to give you an example of where you would start and how you could start.

– That “createWitness” method weaves down into the KERIpy. – Yeah. – Great. Thank you. – No problem. If you don't have any questions, then I think I'll give you back 20 minutes.

TALK XXI

Keanu-Pahio-Interview

KERI Conference April 2026 · interviews · 00:59

My name is Keanu Pahio. I'm one of the developer of the KERI Foundation and my presentation today was about deploying KERI infrastructure. Including witnesses, watchers, and the wallet. I think it's about making connections together. Through my presentation, we can see how you can deploy both watchers, witnesses, and also the wallet, but also the boot server that we developed and how it all works together and how this infrastructure is relevant for a production or also development environment. I think it would lose the connection and how it all works together, how it's related. And I think it's a good step for beginners because it's really easy to understand. I think new people who come into KERI could benefit. If we lose it, it might be harder for people to get into it. I think it's really easy to understand.

TALK XXII

KERIA Tutorial

Kent Bull

KERI Conference April 2026 · talks · 41:15

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Kent Bull demos Signify and KERIA's signing-at-the-edge architecture used by qualified vLEI issuers in production. He maps the agent API surface for developers, showing how encryption keys and identifier salts stay in the browser while the agent stores encrypted state.

Okay, we are getting started. All right. This is primarily developer focused. We're getting down into the nitty-gritty nuts and bolts. And I imagine since it's developer focused, we'll probably have a smaller audience. I'm going to go through and get to know each one of you.

Some of you I've never seen before. Who? Oh, thanks. Great. I don't think I've ever met you before. Will you tell me your name? This is everybody. This is Joseph Hunsaker. Everybody knows Karla McKenna. One of my bosses at GLEIF. And I do have to remind me your name. Matthew? Yes, – Matthew. – Matthew.

Matthew Hillstone. – George McEwan. – George McEwan, okay. Mitchell with healthKERI. Mitchell with healthKERI, okay. I don't think I've met you before. That's cool. Mark Scrimshire ONYX. What was the name of the company? ONYX?

ONYX, okay. And then the name, Matt? Mark Scrimshire. Mark Scrimshire, okay. Nice to meet you. What this tutorial is for you if you're wondering what is KERIA, what is Signify, what is this signing at the edge architecture that many of the qualified vLEI issuers in the GLEIF ecosystem have embraced.

For example, Provenant, Finema, I would say there's a CFCA in China. And we've got a number of others that are Trade Go. Certizen, that's correct. The platform you're seeing today is used in production by a number of our qualified vLEI issuers and other people .., there's the Veridian wallet, they've built on top of this architecture as well. Now, I should caveat that the wallet you're seeing today is a demo wallet. It's not Provenant's wallet. No QVIs use the wallet. It's a demo wallet to walk you through the features.

Now, the goal of today's presentation is so that you can walk away as a developer knowing how do I configure and run this thing? What is this API surface area? What are the capabilities? What can this thing do for me? We will actually have a demo. And if you look at this wallet right here, this is something yes, I use Codex to help me make this.

I actually made one of the first wallets in the world for DigiCred for student credentials based on Signify-TS and KERIA. This is way better than that. It's more of a developer wallet, I'll admit it. Their DigiCred wallet was way better user experience which geared towards students for non-technical people. This wallet is geared for you all. If you want to look inside how does this work as a developer? It's geared towards your mental model. Cuz quite honestly, for a customer for a non-technical person, this is already way too complicated. This is a developer user experience. It's for you to categorize and understand what are the subject matters of API surface area for me to worry about. Did you have a question, Matthew? – This is basically is it meant to be a web application? Is it meant to be an extension or a mobile app or all any of the above? – I'm going to repeat that question because it's a very good grounding question for what is this architecture. He said, is this intended to be a web app? Is it a mobile app?

What is this intended to be used for? Let's actually Or a browser extension. Or a browser extension. Let's actually pause and jump back to the presentation cuz I answered that question. Very good question. I can tell you're thinking.

Okay, then we ask what is KERIA? When is it useful? KERIA, the architecture, let's just jump straight to the nuts and bolts. It's this architecture right here and we're going to answer your question right away. That web app that you were seeing is a browser web app.

Most people use it as either a browser app or like on a cross-platform application basis, Ionic Capacitor or React Native. Anything that uses TypeScript that's compatible with Libsodium in the browser, specifically Libsodium wrapper Sumo, that can work with React Native, you can run the Signify and KERIA architecture. Why would you want to do that? Let's understand a little bit more about the "what" and the "why" will become evident.

I'll preface this by saying there's two broad categorizations of wallets in the KERI ecosystem. The terminology that I'm about to use is mine. I started using this because it's it's a way for me to categorize how the different wallets are.

You can use whatever terminology makes the most sense to you. I call the Signify client wallet, Signify and KERIA architecture, I say lightweight wallet. And I call what Locksmith is, Philip Feairheller from healthKERI, Locksmith is a heavyweight wallet. Why is that the case? I'm about to describe why and we're just you're going to understand it from the very beginning of the presentation.

Lightweight refers to what the user has in their hands. With Signify, all they have is an edge signing client. They don't store their credentials. Keys only ever exist unencrypted in memory in whatever browser or web view instance they have, but that's all that's ever stored here is just the unencrypted keys and just enough state to do whatever your operation you're doing, whether it's issuing credentials, doing an interaction event, doing a key rotation, doing an OOBI resolution. A bare minimum of state is kept here, which is why I call it the lightweight client.

The heavyweight client essentially takes, you could say this edge signing architecture, and your key store, heavyweight wallets put all that together locally. You could think of local first software where all of your keys .., if you'd have a heavy weight wallet, you'd have everything in in one app on your phone, and you would not have a server agent. The lightweight wallet has a server agent, and you might say, "Is that okay from a security perspective? What about the keys? Where are the keys? Are we custodying keys here?"

We're going to get into the nuts and bolts in the later slides. The answer is that there are keys, key seeds that are stored here, but they're only ever stored in the server encrypted. The way that it works, and we'll get… there's some more slides… is the root master key here in the Signify lightweight wallet client is an encryption decryption key.

It's also a KERI AID. It performs a delegation from the in-memory identifier to the agent identifier. If you're familiar with the concept of a witness, a witness is a non-transferable identifier, right? And but it's a regular KERI identifier. You can get a key event log. You can resolve an out-of-band identifier to get that Key Event Log with that one event plus some location scheme records and endpoint rule records for a witness. An agent is a little bit like a witness, and there's a common capability between witnesses and agents at least in the most commonly deployed form. HealthKERI actually did witnesses the right way in the sense that it's only a witness.

But, just because of an artifact of accident of history, witnesses in the beginning of GLEIF's implementation of the the KERI spec, witnesses and mailboxes were combined. Mailbox is a store and forward mechanism. Cuz what happens if your phone goes offline and it's a lightweight client? You still want to be able to receive messages when that phone comes back online, right? You don't want to not be able to receive messages because your phone's off or it's dead battery.

What you do is there's a mailbox. A mailbox runs in a persistent process that has an IP address, a way to essentially send it messages, which is similar to a DIDCOM relay. Now, I see that a couple of people came in here later. Before I sort of finish off the mailbox topic, I'm just giving you a warning. I'm going to come in to ask your names and sort of introduce you to everyone else cuz we're all developers here, so we get to to know each other. All right, so I'm talking about this concept of a mailbox.

An agent functions as a mailbox. It's a store and forward mechanism. It's a persistent internet presence. The primary responsibilities of an agent, it's a mailbox, it's a delegation communication proxy, which we'll get into later, and it's the primary data store.

Your ACDCs are stored here, not here. Your contacts, your out-of-band identifiers, your Key Event Log, it's all stored here. It's all signed here. Nothing is signed here on your identifiers. With a small caveat, remember I said it's a mailbox, it's a store and forward mechanism, it's a delegation communication proxy.

As it performs those functions, remember, we're in zero trust. It's a signed everything architecture, no shared secrets architecture. When the agent itself, which is an identifier, it's an actor in this ecosystem, when the agent is sending communications, it does have its local key that it'll sign messages so that everything coming from the agent can be signed and verified.

But, the agent does not have any of your keys. That's the thing. Go ahead, question. – Is this strictly a delegation model? – Strictly delegation. KERI delegation. Okay. And what's interesting, so how it works and this is… architecture slides are fun because you can spend a lot of time on them and then we'll breeze through the rest of the the slides.

The way that you provision an agent, so when you instantiate it in memory, the way that you boot an agent there's it's two-phase process. You do boot and then connect. When you do your boot up (I need to touch my computer more often, I guess.

Oh, it's thinking that we're not paying it enough attention. Let's give it some attention here. Change. I love this picture. I got to go there someday. This is in France. It's the one of the most beautiful places I've ever seen in my life.

I'm going to take my wife there.) We're back up. process, you're actually signing a delegation event. You send it over here. You request from the agency. You actually send the initial boot to the agency. It provisions the agent and starts the delegated inception process in the agent identifier. It's first event in its Key Event Log is a delegated inception. It accepts the delegation request and then when it responds on the connect, it actually completes the delegation handshake.

This is a strict KERI delegation relationship between your edge signer and your agent identifier. I have some other graphics here to show you that the agent has the responsibility of managing the database for your identifier, any credential registries, and credentials that you've got. Now, the keys that are stored here, remember they're only ever encrypted.

And they're encrypted with the key that is derived from the passcode you have at the edge. This passcode never moves over here, ever. It only ever stays at the edge. That's why it's called a signing at the edge architecture.

Not quite. This together is the lightweight architecture. The heavyweight wallet combines both of these responsibilities into one thing. This runs on a server. This runs on a client. Makes sense? (This is leaking. That's really weird.

Just got to drink it all. I wasn't abusive to this. I don't know why it's leaking all over there.) Let's move on. Is there anybody that has any lingering questions on the architecture before we move on? – The agent on the right side is the whole thing. It's not just the one component?

– This is multi-tenant in the sense that the agency can have multiple agents. I expanded out what you're going to have inside of an agent. – There's multiple clients. – There can be multiple clients. Yeah. – All using the same agency, but they have a different… – Yes, this is the orchestrator for the multi-tenancy for a given server process.

– Therefore one agent per Signify client. – Correct. That's a good point. It's a one-to-one relationship between Signify controller and KERIA-agency agent. – And all of the components underneath the agent are what's inside of it.

– Yes. And there's a dedicated LMDB database for agent. Actually a set of files. Now that we beat that to death, KERIA means KERI agent. That's what it means. It actually originated something called the simple KERI Web Auth Protocol and that evolved to Signify, sign at the edge. Where you have your link to the original hackMD document. I remember when this is being first talked about back in 2021, 2022. There's a current WebOfTrust repository with the original simple-KERI-for-web-auth (skwa). It's actually some of the best documentation of the different alternatives and ways to think about it if you want to really sort of do a some software archaeology.

But the basic idea encryption key at the edge and the identifier seed salts in the agent. Now you have the encryption key at the edge and what happens is you will pull those encrypted salts from your agent locally to the edge in the browser and use them to do signing of events for any of your managed identifiers.

Okay, so… what's the agent? We sort of beat that to death. Let's go ahead and move on. I don't think there's anything [more to discuss, red.] We haven't really talked about delegation communication proxy. We'll get into that a little bit later, but once again, it's your storage server for ACDC's contacts, OOBIs, challenge response, and all that.

At a high level, what are the features we have in in the Signify/KERIA architecture? It's just the same things that you're going to have in the heavyweight wallet architecture. It's just all the events are signed at the edge. We've got client creation boot and connect like we talked about. Client creation, that's actually something specific to the lightweight wallet architecture, but all the rest is applies to the lightweight and heavyweight. Identifier creation, rotation, interaction, I should have put signing on there. Out-of-band identifier generation and resolution, challenge creation and response verification, credential issuance, grant admit, and the full IPEX prop flow, if you're familiar with the other verbs.

And we're going to get to the KERIA configuration. That's sort of what what we've got the capabilities of this architecture here. You will leave here knowing how to configure a KERIA server. We're about 20 minutes in. We'll get to that. I'll make sure we get to that. Question.

– Credential issuance. Is for the AID [inaudible] they're issuer Is that? – No, that's a good question. Let's actually go back. Which identifier does the issuance? Is it the Signify controller that does the delegation to the agent? No.

What's called a managed AID is what does the issuance. The Signify controller delegates to the agent and it says, "Agent, make me an identifier." And the agent says, "Okay, well, you have to give me a signed event." You create the key locally that's going to end up as the stored encrypted seed.

When you have that in memory, it's decrypted, you create the inception event for a managed identifier, then you send that to your agent, and your agent says, "Okay, great. I'll create a managed identifier record in my database."

And this is the level at which you're going to start to issue credentials. It's actually one layer deep below the Signify controller. Does that make sense? Or enough for now? We'll have a demo later. It'll probably clear it up a little bit.

– Why are you generating the key and sending the key at that client and not generating it and sending it at the agent itself? It feels like the generation of the key would be more secure at the agent side. – It's not because the server can be compromised, right?

We don't store any keys here unencrypted. The keys only ever exist at the edge, which is why it's called the signing at the edge architecture. – I see. You're encrypting the key, and then that encrypted key is stored there. – Yes, the encrypted key is stored in the agency, but remember, the agent never has access to the unencrypted key ever.

All signing occurs at the edge. This has implications when you use HSMs (hardware-security-module). Because you have to send your event to the HSM to be signed. HSMs are another signing at the edge architecture. It's just the HSM is the edge.

Question. - [Why store encrypted key in the agent?, red.] Think about it. If you lose your lightweight client, you don't want to lose your seed, right? All you have to remember, the reason why it's a lightweight architecture, all you have to remember is your passcode.

If everything fails and you lose your phone, well, you've got a server agent where you can, as long as you have a password manager and you have your root passcode, well, you can reconnect to your agent and reconstitute the keys locally by decrypting them. Storing them in the agent is actually very handy.

Yeah. You could store them in a mobile app if you wanted to do that and change the storage model, you could totally not even store them on the server. They're just stored on the server from a convenience perspective. And really, you could build an agent replication or state migration architecture if you really wanted to on top of that.

Yes, 21 characters. But, it's actually exactly 21 characters. If you put in more than 21 characters, it chops it off the 21. And if it's less than 21, it doesn't let you do it. It doesn't do any stretching right now. It has to be 21 characters.

And I can we can go into the dirty details of why that's the case sometime. Yeah. Okay, we went into the features outline. I'll show you a demo. [hardly audible confirmation] [that this has nothing to do] [with witnesses and watchers] Correct, let me clarify. Let me give you a caveat.

Whenever you create an identifier, you can specify witnesses as Backers. It's not the agent. What I'm saying is that the agent is not a witness. The agent is not a watcher. Well, Anything that stores Key Event Logs acts a little bit like a watcher.

But technically in the way it's commonly thought of this is not a witness. This is not a watcher. That's a separate service that's run. Is that what you're asking? Great. Okay. Did you have a question? A quick demo to answer your question to make it clearer what does the signing? Let's actually run through a wallet. This is the wallet that I made. Let me actually pop over to the browser here. What I'm going to do… If anybody knows about the Signify React TS example repository in the WebOfTrust community you'll probably recognize this button.

This is the only thing recognizable from back when RootsID originally contributed the wallet. I did heavy coding with Codex over the last 3 days to get this in this state. We've got a heck of a lot more features and it looks way better than the old ugly one did. I think this is still kind of garish, but we'll connect.

I'm going to connect. Now what what I'm going to do here is I'm going to paste in my passcode. (It's kind of hard to see I guess with some of those the screen separations there) I'm going to paste a passcode in. There we go.

I'll just go ahead and show. This is a 21 character passcode. Oh, that's fine. This is a throwaway passcode. But thank you. Good security consciousness. Oh, thank you. Good point. All of that is luckily just dev stuff. But thank you.

You're making me think about something I saw on X about a person who They had a a policeman took a picture of somebody's Bitcoin private key and it was it was on their… not even a picture, but it was the body cam footage and then somebody got like million dollars taken out of their Bitcoin wallet or something like that. But thanks for the security consciousness. Luckily, I don't have anything worry worrisome in my clipboard.

There shouldn't be. You should double check for me. All right. If you find some, let me know. Now, this generate here. By the way, this is open source Apache 2 stuff. This will generate… I'll just make a new one. And then I'm going to copy and paste it so I can reuse it.

When I connect here, so what's going to happen this pass code will instantiate in memory a Signify edge signing client. I'm going to click connect. It's provisioning the agent. Now, I've got the agent. This is sort of a dashboard. I'm going to go to the identifiers.

I have zero identifiers right now. The Signify client AID in memory is not assigning identifier. It's not an identifier that will issue credentials. I have to create an identifier. I'm going to click create identifier. And I'm going to say… – Is this creating one of those [inaudible] – This is what? – This is creating one of those managed identifiers. – Managed identifiers, correct. This is creating a managed identifier. I could change from salty which is deterministic… This is if you saw the healthKERI demo they had keychain versus random. Salty is essentially random. I'm surfacing some terminology that really should be hidden from the user. I just went really fast.

Salty is deterministic, Randy is random. We could make a delegated identifier if we wanted to. I'll go ahead and put some witnesses in there. And if we would look at the advanced options, we can sort of specify signing threshold and things like that, but I like to hide those options unless you actually know you need to use them.

We created an identifier with three… Let me just sort of pop into the details here. It's a little bit hard to reach from the back. But this is our AID. This is our managed AID. This is not the Signify controller identifier.

The Signify controller identifier just connected to the agent. And actually, I can I have a page for that. If we go to the client, let's scroll up a little bit. This shows you some details about what we've been talking about so far.

The controller AID, which is derived from my pass code, is this AID right here. It delegated to the agent AID. This is a KERI delegation relationship between these two identifiers. And here's some more interesting stuff about… Okay, the sequence number of the… Technically, as you can see, since it starts with E and if you know CESR, they're both transferable identifiers.

You could rotate keys for either the controller or the agent. And you can see the sequence number that the controller's on. We're on sequence number 1. You as you see, (remember how I said delegated inception?) the first event for the agent is delegated inception. You can see the first message it's showing in this here is "dip", which is delegated inception.

This is the little control panel that I added to show both the Signify controller and the delegated agent. I don't think there's anything else that we want to show here. We created an identifier. And we can go in there. You can see there's a couple interesting things you can I can copy the agent out-of-bound identifier, which if we curl that, let's actually show that.

I'm going to copy that. We're going to go to… You can already kind of see where I'm going here. Let me zoom in a little bit. What if we wanted to see what the Key Event Log is for that identifier we just created? We copied its out-of-bound identifier, what does it look like? If we curl that, this is running on my computer now. This is a big old honking CESR stream. It looks like the Matrix.

Let's format it so it's a little bit easier to read. Let's do pipe it to a little colorizer. I went kind of like JQ I wrote for CESR. 2FA annotate colored pretty Okay. It's a little bit colorized. I'll walk through this kind of slowly.

You'll see this is the inception event for the first managed identifier I created. It's an inception event. Notice it's not a delegated inception event. This is a dedicated identifier. That's a managed identifier where you use… The root pass code that you made is like the root seed.

You generate another root seed for this managed identifier. It's encrypted and then sent to the agent and you pull that back when you're going to do a rotation for this managed identifier. Is that making sense now? Perfect. Okay. Any lingering question on the relationship?

– Yeah. When I think of the AID, I think of entity of some sort. I'm the controller. The AID is essentially me as a controller. The AID of the agent is the agent AID Those two make sense, but I don't understand why you why I would have a bunch of managed AIDs.

Like is it In the case of a university, am I making… I don't know. Like what is the use case where… I Why would I have to manage several AIDs? – It could be that one of your [identifiers]… Let's say you're a legal entity and one of your identifiers is for your legal entity itself. And another one might be for you as the CEO.

Another one might be for a different role that you're playing for engagement context role credential. And it there's a variety of different reasons why you might have more than one. But generally speaking, it's not going to be very many, depending on what you're doing.

– They should all be under the the controller's jurisdiction [inaudible] – I mean it could be. Maybe you're working for a corporation and they say we want you have an account for this and an account for that and have different identifiers. It really just depends. – Okay, the root controller could be an employee.

And then this company makes a KERIA server for their employees. And then they [inaudible] people or something like that. They have roles that might need to communicate with other entities or something. You wouldn't think of the corporation as just one Signify controller and then managed AIDs per employee.

You would think of each employee at that corporation as having their own Signify controller. And maybe one, usually probably one, one or more managed identifiers. Right. And the company would have one or more KERIA instances.

That's right. Question. – I was just going to make a comment on the managed ID discussion. We discussed that in a SAID perspective as we try to, how do you do, prevent cross-context correlation. We've been looking at managed IDs and like all issues.

I think that's the same scenario we're discussing here. We have individuals that have multiple managed IDs that they can use in different contexts. – Oh, for sure. Like you could have somebody that says, I want my whatever SEDI credential with under this identifier… – Or that the state may issue multiple SEDI credentials.

– I see.

But a credential is not an What he's getting at is that he's… if I'm understanding you right, Chris, is that SEDI may allow issuing the same SEDI credential to different managed identifiers that you have cuz you might want to segment, for your own privacy, the usage of a given identifier. – You can both issue your AID, managed AIDs [inaudible] state for endorsement.

It's effectively your steady credential duplicated to whatever level you want to go. – [inaudible] your ID from being a main Okay. It's like sharding your identity. Like effectively if I might want to have a managed ID for my university or another one for my financials or another one for my whatever. Just like so sharding pieces of me.

As an individual, but as an employee, I might shard my identity in different ways that are custom to my company or my organization or what I'm doing. – Yeah. That's right. – The infrastructure provider for the agency configuration: What prevents this individual to mess with this with the signer keys or I guess with the agent because it's not [inaudible]. – Good question. Encryption.

Yes. The administrator of the multi-tenant agency server has access to zero keys. Because the keys are always encrypted at rest when they're on the agency. The only time the keys are unencrypted is after they've been pulled to the edge client and your master decryption key decrypts them.

Even if somebody compromised the server, they could still never use your keys. – Got it. The signing goes up in at the edge always. Oh, I thought it was at the edge. Okay. – It took me a couple of little whiles to understand it when I was first learning.

And what's cool is, you learned more quickly than I did. It took me a while to really track this down and sort of ask the right person and sort of figure it out. And I didn't feel comfortable with it until I read through the code to see exactly how it works and I could show you exactly where. But all signing happens at the edge.

Now let's go through some of the slides, see if we've got some more .., We've talked about this: edge controller signs events. Sends to the delegated agent, which we talked about is the agent. The agent is in KERIA. We've got it handles identifiers, all these other capabilities that we saw.

Okay, we have two signing libraries. I recently updated SignifyPy to be on par with SignifyTS. Yes, the latest version of SignifyTS was released yesterday, 030. And the latest version of SignifyPy is 0.2.0. I should re-release that just 0.3.0 to be sort of consistent. But SignifyPy is updated in actually, no, it's 0.4.0. Even worse.

I got to align the version numbers. SignifyPy is updated in PyPy SignifyTS has been updated in npm as of yesterday. And what I don't have on the slide here, so there's KERIA, the most recent version of KERIA has been released at 0.3.0.

There's going to be a 0.3.1 very soon because while I was preparing for this presentation, I found a delegation bug or missing feature. And we fit we patched that this morning. As soon as it gets the PR gets approved, we'll patch it in.

Anyway, that's what we've got: two libraries right now, SignifyTS and SignifyPy. And Cardano has Signify Java. They have some clients who wanted a Java implementation, so they implemented the entire surface area in Java, in case you want to know. Now, the Cardano implementation has only ever so slightly diverged from the WebOfTrust implementation. It's very, very close.

It's if you needed the Java implementation, it's very doable to, if you wanted to, use the WebOfTrust standard implementation. These are the gotchas. This is what's going to save you time. And I was okay, we got to be done in 9 minutes here. I want to get through all of these and some of the config.

We've been answering questions throughout. Please continue to stop me. This is where you're going to need to come back to me. You or someone on your team will have to come back to me or somebody from the KERIA community cuz you're going to hit these problems.

What is the passcode? Where are the keys stored? And we've cleared a lot of these up cuz you're going to have to re-explain this architecture to every team member. And they're going to ask, "Where are the keys?" They're going to ask exactly what you asked, "What if somebody compromises the server? Is that a problem?" You're going to have to explain to them, "No, the keys truly are at the edge."

It's like it takes some individual missionary work to get it between minds. And you say, "Okay, what is encrypted and decrypted and when?" We've already taken care of the major misconception or misunderstanding that people often have with KERIA. You're already ahead of the curve.

And some people ask, "Okay, what about storing the passcode? If I'm making a web app, can I just store it in indexDB or localStorage?" That's a big no-no. And the reason why is any same origin JS code from you can read and exfiltrate that passcode.

Now, if you want to use something like Ionic Capacitor, they do have a secure storage plugin that'll work with the trusted execution environments or other hardware you have on a device. It's called the Capacitor secure storage plugin. It does not save to indexDB or localStorage.

Now, for a development time convenience, sometimes to speed up development, you might want to say, "Okay, I want to be able to refresh the page." Maybe you don't have hot module reloading figured out, for whatever stack you're working with. If you want to store it temporarily as a dev thing to make it easy to refresh and not have to recreate your state cuz it's annoying to have to… I'll show you exactly what I'm talking about. I'm going to refresh here.

And boom, I have to put… I'm logged out. I have to go and put my… reconnect, which this can be annoying if you're doing a very rapid iterative development loop. As you might you want to say, "Well, when I refresh, I want to deep link and go back to that deep link so that I can quickly develop."

If you want to add that in a development time, that's okay, but don't let it go to prod. Do not let that go to production. And use the secure plugins if you're going to use something like Ionic Capacitor. Where are the keys stored? We beat that one to death. We know that Signify controller key is the edge encryption key and delegation key only.

AID root salts that are only ever unencrypted at the edge, which is where all signing happens. You can just think of it as an HSM. And when they are stored in KERIA, they're encrypted. They're encrypted before they're even transmitted.

Okay, the other biggest one, you're going to forget this. I forgot this a couple times in the last couple of days. You need “KERI_AGENT_CORS=true” so that it properly responds to preflight requests, the preflight options requests.

Make sure when you're viewing your “keria start”, whether you're starting in a Docker container, whether you're running it local… just the regular raw Python virtual environment, you need “KERI_AGENT_CORS=true”. You put that as environment variable in your in your CI scripts, in your deployment scripts.

We really should just make this the default behavior. I need to do a PR to make that the default behavior. I mean, I don't know anybody that doesn't deploy this way in production, so we'll probably make that the default so that you don't hit that over and over again.

Okay, multisig. How is multisig handled? Multisig in KERIA is just like it is in heavyweight wallets KERIpy with the exception of what transmit the messages. What's the mailbox? When you're doing multisig, you need to know how to communicate with each member, which means you need to know where their witness or whatever's acting as their mailbox is so that you can send them your multisig notifications.

In KERIA, that's your agent or their agent. Other than that, it's the same. There was one I said there's a delegation but there let's see. Let me think about There has been some I'm trying to think of some bugs we've had over the last year. For all intents and purposes, it's the same except for it goes through the agents instead of a witness.

Okay. Okay, this is the only the other thing. We're looking at 5 minutes. Good. We'll be done with KERI config. Okay. I do have a blog post that goes through this pretty exhaustively. If you go to kentbull.com /posts/configuring-keripy-keria-controllers-witnesses/ You can go there. It's just kentbull.com right there that post.

And I go in detail, but we'll quickly walk through the most important parts of a config file. If you want to start up a KERI agent server, you need a config file that has at least the date property. And one other thing you must have this block right here with the curls, which is controller URLs, cuz what happens is it takes this value and binds to that port.

And this name right here matters. Let me go forward to that. This name has to correspond to the name .., when you do “keria start --name "keria",” these two names need to match up. That's the hardest part. And then the other hard part is that you must have a value here so that it can pull out the host and the port to bind to that port. That's what Python reads to do the port binding.

As long as you've got the port open, the host right, you can use 0000 if you need to. And these names match, then you're good to go. Now, the other parts of the config, you might say, "What would I use the other parts for?" IURLs are essentially identifier or AID URLs.

Who are the contacts that I want to resolve on startup for every single agent? Cuz this config is the agency config. The way that config works in KERIA is you configure the agency, and then it copies its config to every agent. If you want to bootstrap .., Let's say you have preferred witness infrastructure and preferred credential schemas that you want every agent in your agency to know about, you put them in your IURLs section, and there's actually a DURLs section specifically designed for credential schemas and partial ACDCs. Like if you have a trust chain like the GLEIF trust chain, and you want your agents to pre-resolve it, there's an IURLs section for identifiers, and you'll notice these are all witnesses. What I'm saying is I want every single agent that boots up here to resolve these three witnesses so that I don't have to through the steps of resolving them every single time.

Anyway, there's some tocks. This is some more advanced stuff if you need to do some CPU tuning and IO tuning. This is where you're getting more low-level into the async. Tomorrow I have a talk, HIO async frameworks. It's even lower level than this. This is where we'll get more into tocks, but this is more into the KERIpy async framework called HIO, hierarchical IO that Sam [Smith] designed. That's what these are refer… As long as you know what it's for that's enough to sort of start to ask her off and that sort things. Okay.

We kind of understand why to use KERIA. Use KERIA when you need a lightweight wallet. KERIA is signified when you need a lightweight wallet. When to not use KERIA? When you need a heavyweight wallet. And we talked about configuring. Now, go build your AI agents with security behind them and your steady agents.

TALK XXIII

KERIpy Async Architecture

Kent Bull

KERI Conference April 2026 · talks · 39:06

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Kent Bull teaches KERIpy's HIO structured concurrency model for contributors. He contrasts cooperative coroutines with threads, promises, and callbacks, emphasizing that child tasks cannot outlive their parent lexical scope—a design choice that shapes how KERIpy runs.

This is a developer focus description for the async, the runtime foundation of KERIpy. And what this is that's essentially what's the process, what's the execution model, how does this run? And if I'm going to read the source code, what do I need to know in order for me to effectively contribute to the KERIpy code base.

And to learn it or to extend it. And hopefully by the end of this presentation you will have enough of a mental map and a mental model to know how to effectively talk to Claude or Codex or another AI in case you need to go and relearn this later to concretize your own learning, to anchor it in or to potentially extend it.

We're going to go over a lot: What structured concurrency is, why it matters, how it's superior to promises async/await callbacks and goto. And then some common gotchas for when you're trying to use KERIpy and why things are done the way they are.

All right, so hello is it's Evan, right? No, Keanu! You probably know a fair bit of this already. I know I've talked to you about this before, Joe. I was just probably talking to you when we say stuff about structured concurrency.

How many would say they're familiar enough with structured concurrency to be able to explain it? Keanu? You work at the KERI Foundation. – I think, HIO, the only time I've really dove into the code, was when we had to port into the web base there. Because we had to use HIO because it was async. It was asynchronous so we had to adapt it because in the web it was sync, so we had to convert it and that's what we did in the tutorial, but I don't want to say something that is completely wrong, so ..

Okay, I see. This will be valuable for you then because maybe you haven't had as much experience with the HIO async runtime. Great cuz if I even have one more programmer that really deeply understands HIO after this, that will be good because I got to get this knowledge out of my brain.

Who's going to ask a question? I think structured concurrency is just multi-threading and then how you actually come to a finality? It's a good question. You're going in the right direction. It's not multi-threading. Okay. Are you familiar with Go co-routines?

– Not in Python. It's a lightweight co-routine construct from Go lang. I just said the Go co-routines. One of the reasons why Go became so popular is people like the simplicity of the mental model with Go co-routines. Python co-routines are actually better than go co-routines in my opinion, the way Sam uses them in HIO.

If you want to think about what the unit the base fundamental unit of concurrency in a structured concurrency framework is, it's a co-routine that is cooperative with the other co-routines. Usually a thread .., it's like a fundamental construct where it doesn't matter if you block within one thread and the

reason why you use threads is because you want to have a main thread that's not blocked by work that will block that you put to another thread.

If you want to adopt the mindset of structured concurrency, you need to think from a non-blocking mindset. Often it's called a reactive mindset. Zero of the co-routines should block. And you might wonder, how is that possible?

The way this is implemented at the IO level with network IO and file IO is in a fundamentally non-blocking way. And how do you do that? You essentially chop up your IO reads and writes into chunks and then you… it'll be blocking ever so slightly to process one chunk and then it'll move on.

But it's fast enough, right? Exactly. It doesn't like it's like it doesn't really block. And not only that, but if you look at the low-level socket implementation in HIO, it's it waits for a chunk and if there's an error, it just waits and then schedules itself for the next cycle of the root scheduler to schedule that co-routine again.

Okay. We're still at the first two slides two slides, so you're good. We're talking about what structured concurrency is. The fundamental unit of structured concurrency, which we will beat to death in the rest of this as a Doer, it's a co-routine.

A non-blocking co-routine. And if you look at the readme of HIO that we're diving into, it says, a waitless hierarchical structured cooperative concurrency runtime. Now, you might say, let's dive deep into structured concurrency. Let's go all the way back to 1968.

You might have heard of Edgar Dijkstra. He's very popular in the computer science space. There's a Dijkstra pathfinding algorithm and lots of other things. He wrote an article back in 1968 that says, “goto statement considered harmful”.

And it's actually not that long. This is the whole article, this page and this page. And he says a key part statement, the unbridled use of the goto statement has an immediate consequence that it becomes terribly hard to find a meaningful set of coordinates in which to describe the process progress. He's trying to say this is very difficult to understand from a procedural standpoint, where I'm at in my procedure if I've got goto statements that could take me to arbitrary places.

And we'll give you a bit of a… I'm going to give you a visual here. This is flowmatic. It's like Basic. It's like some of the early programming languages and this is a jump, but a jump is a goto. And it's saying, "Okay, when you get down here, if this set of conditions happens, go to operation two."

That seems like it might be okay till you get a lot of goto statements. And now how do you understand what the boundaries are here? Have you ever heard of solid programming principles? Highly cohesive, loosely coupled? Where's the loose coupling?

That looks like tight coupling spaghetti code. Right? That's where the name spaghetti code came from. Right. We don't like this. This is bad. This is hard on humans. It's hard on LLMs. Nobody likes it. Now, the reason if we go back to couple slides, so goto, the go statement in Go for go promises, callbacks, they're all variations, if you look at them closely, of a goto statement. And the problem is they break abstractions.

We as programmers endeavor persistently to have clean abstractions that let us reason about the complexity of a system that we're modeling. Like Sam said earlier yesterday, all models are wrong, but some are useful. All of our models of abstractions of the real world are wrong, but they're useful to accomplish a certain end. They're abstractions.

We try really hard with software architecture to have invariants about abstractions, conditions that are true no matter where our system is at. And we create state machines to ever so specifically describe the behavior in the states of our system.

That only works if our abstractions are stable. Goto breaks abstrations. There's a very good article by a man named Nathaniel J. Smith, not related to Sam Smith, where he talks about structured concurrency in the context of a goto statements in in the context of structured concurrency. And he talks about… That's where this picture comes from.

He shows you how effectively a promise, whether it's a JavaScript promise or a which is async/await in JavaScript, or in Python, where you have async/ await, they're goto statements. They really are. And I'm not going to go into the deep… We have so much to cover today.

You just go read the article. You're going to be so happy you read the article. Let me go to… There's actually a couple of articles cuz he talks about what about timeouts and cancellations. How do we thread that through the code without making it explode and blow up and look ugly. This right here, https://vorpus.org/blog/notes-on-structured-concurrency-or-go-statement-considered-harmful/ You can take a picture now or get it with the digest that Kim's going to give everybody later [site keri.foundation]. This link is in the notes.

What's fascinating is that Samuel Smith and another excellent engineer, Charles Lowell, who wrote Effection, which is a structured concurrency library in JavaScript, also TypeScript, they both cite this article [Nathaniel J. Smith] as their inspiration.

I highly recommend, and in the wallet that I put out in preparation for yesterday's talk on KERIA, it's a job TypeScript application. It uses Effection structured concurrency. And we'll see a little bit of that code later in this presentation.

You If you want to understand in precise intricate detail why promises, callbacks, the Go channel, Golang, Go channels are different kinds of goto statements expresses concurrency constructs, read this article. And at the end of it, you'll be disgusted with the different kind of goto statements. And you'll say, "How can I repair my abstractions? How can I have clean abstractions?" Highly recommend this article. You're just going to love it. It's kind of long, so you get a nice refreshment or give yourself an hour or two.

What is structured concurrency? Let me jump forward to a couple… This is a borrowed diagram from Sam Smith. What's interesting, you might say, “Let's talk about root Python, the async/await.” What it is the async HIO has an event loop, and the event loop runs any function that has async in the function, right? The way in Python you use their native async support is it'll just run a generator for you. And the way you declare a function as a generator is you mark it an async function.

The async function is the only way you can run This is not HIO. This is just regular Python. The only way you can run an await statement inside of a function is if you declare it with async as async def, right? That's the signal to the Python runtime to plop it into the async HIO event loop, so it'll get scheduled. If you don't put async in front of it, it won't get scheduled, and then it'll never get run. Now let's take that kind of that mental model, and let's do the HIO version of it.

The root scheduler is called a Doist. The Doist is what will run all of your async tasks. There are two kinds of async tasks in HIO. There's the DoDoer and the Doer. Now, the DoDoer is an aggregation construct. You can run many child tasks, and it can schedule them. The DoDoer is like a little bit of a

combination of a scheduler and a task. It has a function where it'll perform work and it has the ability to add more tasks to itself. You'll see a little bit of similarity between HIO's structured concurrency design and other async frameworks like Trio and Curio as and Effection as we go on.

At a high-level architecture diagram, this is what we're dealing with. Primarily three concepts. The scheduler, a task aggregator that's also a task itself, and a task. This is the fundamental unit of execution in HIO. The Doer. This is your task.

You can see this like a task group. Schedulable task group. The guy who wrote this article about fixing this problem. Why would we use structured concurrency? This slide says a couple of benefits. The biggest benefit is my first bullet point here. Program flow or task flow follows lexical scope. Now, what does that mean?

It means that the execution of the program will follow your intuition as a programmer. You're going to write a function. You expect control flow to go from the beginning of the function to the end. And when that function is over, you expect operations you've created in that function to have completed. Unless you're doing threads or processes, in which case you might return a handle from the function to sort of go later, but structured concurrency, usually the mental model is if I have a function that fires off a bunch of tasks, all of those should complete by the time the function returns. Okay. What you're doing is you're turning the function into a bit of a almost like a scheduler of a bunch of subtasks. Are you familiar with Erlang and a process supervisor?

No. Okay. That's a good mental model. Erlang or Elixir? No. It's there's a process super supervision model. But I'll leave that terminology aside. It'll be clearer as I get further into the code examples, but essentially the idea is that if I enter into a function and that function has tasks in it, there's a guarantee that structured concurrency provides that all those tasks will finish or be canceled by the time that I leave that function.

What that does is now you don't have any… The promises break this because what happens if my outer scope cancels and returns and then my promise returns later. Now it's a different scope. It's a different function scope. You might use closures or other things to try to transfer state to callbacks to promises as a kind of a callback.

But, promises break this. What essentially what this is saying is that your function is a kind of abstraction. That function you're not going to break the boundary of that function cuz by the time that function returns, everything in it will be completed and you'll see that in code in just a minute. Another thing, so child tasks cannot outlive their parent scope.

A function is like a parent scope. All tasks you spawn within that will finish before their parent returns. And HIO enforces this with the dot done attribute. Go ahead. – When you say lexical scope, that just makes me… my brain, I go to the program task flow can't outlive. Like, when the variables that you declared in that scope go out of scope, the task outflow is also complete.

Yes, exactly. Precisely. That's right. Yes. Now, now that's a common structured… – Closures and stuff. Like, you basically say you don't have to deal with any of that because everything happens in context with stuff that's in context with it.

Yeah. You don't have to pass any of that state around cuz it's you will you just avoid that problem. – Yes, that's correct. Okay. – This make sense, this seems much more simple. Exactly. That's why people use it. That's the whole selling point for structured concurrency.

– But it feels more complicated for some reason. Venkat Subramaniam, he's one of the most popular speakers in the JVM space. He says: it's unfamiliar. It's not that it's difficult, it's unfamiliar. Cuz once you learn structured concurrency, you're like, "Oh, that maps more onto how my brain thinks."

There's a little bit of upfront learning curve, but then you're like, "Oh, this is so straightforward. I like this." – What do you call that concept in the Java world? Is that a like a task? They're trying to do I think there's a fiber, if I remember right.

There's JVM 22, I believe, is trying to do some integration with some structured concurrency, but it's ugly. It's not very nice. And Pythons and JavaScripts are very, very nice. HIO is the best Python structured concurrency library out there. It's not the most usable, but as far as the most intuitive, by far the most intuitive. David Beazley, a big name in the space has one called Curio that you'll see referenced later. And then Nathaniel Smith has one named Trio.

And then there's Sam's. I've looked at all the other ones. Sam's is the best. It's not the most usable as far as like there's nice wrapper functions because the other ones have so much more adoption, they hit a bunch of problems, and there's a little bit more wrapper functions to adapt them into the async/await stuff. The async/await is a dumpster fire. – Does Java have async await?

Or – They better never get it. It's a dumpster fire. It's too bad that it infected Python. – It's future. I think that's where it is. Aasync/await, the JavaScript form of async/await, was forced upon the world because of the limitations of the browser execution engine. It was a mistake to put that into the Python runtime itself. They should have done structured concurrency.

If there's any mistake that Guido made, the benevolent dictator for life until he resigned in Python, it was putting async/await into Python. They shouldn't have done that. It's really ugly and garish. It works, but it's overly simplified.

Because people were complaining about generators. They were saying they're too hard. But you can make frameworks that abstract the hard parts and still preserve the structured concurrency guarantees. Python had an opportunity the whole language ecosystem had an opportunity to do adopt structured concurrency and they missed it. And now there's libraries.

Okay, other important guarantees of structured concurrency, if you spawn in a lexical scope, you will join before that lexical scope returns. And in HIO that means if you have a DoDoer and you spin up some child processes, those will get to the done state before the DoDoer finishes. The Doist will own all of its children will complete before the Doist returns.

And some nice consequences of this is then we have straightforward execution handling. Failure cancellation propagation throughout the task tree is very intuitive from child tasks up to parent tasks. And now you can intelligently respond to both failures and cancellation signals.

And there's actually a sister article to the one that I mentioned earlier, the note the notes on structured concurrency goto statement considered harmful. There's one on timeouts and cancellations. And then there's one on control C and interrupts.

You want to read all three. They're so well written. Okay, and exceptions are carried all the way up through the scheduling tree. This is one of the major benefits of structured concurrency, is that you can have error propagation all the way up to the root scheduler and handle it there or in any of the parents. It's an intuitive, straightforward way to handle exceptions.

Consistent, predictable exception behavior. Very awesome and amazing. There's Trio and Nurseries from the guy who wrote the article that you're going to go read. And this is an example from his article… He uses “async with”, his async with is essentially a Doist or a DoDoer, just think of it as a Doist.

You start one task, which might have a child task, which might have a child task. And all of those complete before the outer bounding lexical scope of the async with returns. And because it's a context manager, that means you can have startup and teardown logic that will complete predictably for all these tasks, for any resources that you need to open or close. It's kind of Java started to go in this direction with the try with resources, with the auto-closeable interfaces, but they didn't go to fully-structured concurrency like this.

Okay, I talked about HIO and Doers a bit. We went here. We'll sort of breeze past this. You already know the Doist, Doer, DoDoer. Let me double-check our time here. Oh, we've only got 19 minutes left. Now we're getting into the HIO internals. You get some of the high level and an object level, what the benefits are. Let's anchor this in some code.

We mentioned the Doist is the root scheduler, the code routine tasks are the Doers and DoDoers. We won't get into the details of time. It's really fascinating. It's really interesting. I highly recommend you read the source. I had to read it a couple of times.

And this is before AI came out, by the way. Now you can ask AI and it'll walk you through it much faster than I learned it. But you need to understand the outer scheduler has its own concept of time. And it accumulates time.

You'll see the one sleep in the entire system. There's only one "sleep", and it's in the Doist, the root scheduler. And it keeps track of its own cycle time, so that .., and you don't need to worry about completely understanding this, but just anchored in your brain somewhere, the Doist keeps track of time so it can have deterministic scheduling of its array of tasks.

In order to make it deterministic, it has to keep track of its own sense of elapsed time from the start. You always start at zero and increase. If you keep track of your own time and your own sort of talk, essentially your your next… how long you're going to sleep before you advance to your next execution of your array of coroutines, you have to have an internal concept of time. This came and bit me and cost me 40 to 80 dollars in in AI tokens with Codex.

Because it wasn't able to model reality and realized that "Oh, HIO has its own concept of time" and I actually hit that as a bug. You realize LLMs are going to struggle with structured concurrency. And if you don't know how to tell it to do the right thing, you might waste tens of dollars of tokens like I did. I'm glad it wasn't hundreds.

Remember, Doist has its own concept of time, but we won't get into that here. It's another presentation. We're going to anchor this in with some actual commands. And we're going to go through the KERIpy code base. If you've used KERIpy, you've done KLI incept.

What this does is it calls into what's called kli.py this main function. This is the entry point for the command line. The important part is it says runController. That's important because it calls the root scheduler. Here, runController has these three lines, defines the tock, which is the run rate of the cycle time, how quickly the root scheduler iterates over all of its tasks.

Which means this is 1/32 of a second, so it'll 32 times a second it'll run its entire task list. Unless there's like something that takes some time to execute. Here's the instantiation of the root scheduler. And then we say root scheduler.do. This is a life cycle function. This is the life cycle function that runs the system.

This is the entry point of the HIO runtime. Okay, we don't have to worry about those other parameters just yet. Let's get through this. Here's the whole class outline of the Doist. Here's the Do. The Do calls all of these other functions.

These enter, recur, exit. Think if for enter think resource allocation and exit resource closing. That's what the enter and exit context is. There's an enter and an exit for every task in the HIO ecosystem. Every Doer has an enter and an exit.

Now, that means it can be used as a context manager. What does a context manager do? What you're saying is "Give me a before and an before to allocate resources and an after to clean up." That's why people use context managers or try with resources in Java or languages like that, so that you can have automated cleanup [of processes, red.]

What the enter and the exit do is it allows you to make arbitrarily smart your coroutines. Python coroutines by themselves are dumb and stupid. You can't add this nice cool logic to it. This is a big selling point of HIO is that your tasks have an enter and an exit.

Just about everything else in the space only has a recur. Go ahead. – Can you implement handlers to those… They're essentially events, when you enter or when you exit then you can attach handlers and then react to it? – Think of in terms of subclass. You subclass the Doer. You override those functions. That would be equivalent of handlers. Okay, now let's go deeper into the Do function. You'll see.

The red arrows are the life cycle functions. Remember I said enter, exit, recur? Enter. Where are we? Recur, I guess I made this one special cuz I'm going to go there next. There's enter and exit, startup, teardown. We've got the recur, which we're going to jump into later. Here's the only sleep in the system. There is one sleep in the entire Doist run time. It's this right here. This is what accumulates time, so essentially starts to track the time elapsed to know when to schedule the next co-routine.

One thing to be aware of, that self.that enter, what it does is it takes all of your tasks and runs enter on every task. By the time that finishes, all your resources have been provisioned for all the tasks that are going to run. – After that one line? – After that one line, yes.

And what it does, it advances every single co-routine to be primed to be ready, essentially. That readies all the tasks for execution, and then recur begins the infinite while loop to run all of your tasks until you get a an abort an interrupt signal, which will then trigger exit. Like control-C will trigger exit, and it'll exit all of the child tasks, and then their nested tasks, and then it'll finally exit the Doist.

That's one of the cool things about structured concurrency. Okay, we're going to jump in. Let's jump in. Now this is recur, remember? We're going into the blue arrow, or the teal green arrow. This is the recur function. The main What actually runs… We're still in the scheduler. When are we going to get to the task? Right here, this send. And that send right there, that's the Python generator API.

Cuz remember, the co-routines are just generators. And the syntax to run a generator is this "send" right here. You can ask Claude or Codex about it. There's also a really excellent… Luciano Ramalho, the guy who wrote the Python reference manual, his 2016 version of the book has a really great chapter on co-routines. And then when the whole Python language community went the wrong direction to the oversimplified async/await, for the 2022 version of the Python reference manual, he did not include that

awesome content on co-routines, but it's on his website. You can go to I think it's [https://www.fluentpython.com/] and you'll find the chapter 16, I think, on the co-routines. It'll go through that in exhaustively in-depth.

It'll teach you everything you want to know about Python co-routines. High-density, high-value content. Anyway, this is what runs it. We won't worry about the rest of the stuff. This interest interestingly enough, last note on the root scheduler, this tick advances you forward that next 1/32 of a second. It starts the next time it schedules the time sleep, after which the next iteration of all the tasks will occur.

Okay, now let's go into the individual task. What it does, that ".send" calls .., you might be familiar with other Dunder methods, the underscore underscore methods in Python. It calls the call method of a Doer. And what the Doer does is it says self.do. What that send does, this the and it's dog, I won't get into the syntax here. Dog stands for Doer generator. There's some Sam-isms here that it takes a little bit of time to get accustomed to, but this is essentially the generator that is the subtask.

The "dog". We're doing the send and that eventually results in self.do. Now, why is that important? The Doer.do is the entry point into the life cycle of one task. And let's look at that. Here's the Doer.do. Look at that. We've got an enter, exit, and a couple of other life cycle functions for different reasons that we don't have time to get into, but there's a clean, there's an abort, there's a cease.

And the important thing is that Okay, you see the enter and the exit, startup teardown, and you see the recur. Self.recur. This will actually execute the self, but there's another important thing that you need to know a little bit about Python generators, to know how this works.

See how we say isgeneratorfunction, Self.recur. What that means is if you have a yield statement, this is how Python works. If you put a yield or a yield from in a function body, it turns that into a generator. If you're going to do anything with IO, the network, files, that's usually where you're going to have to wait. And usually you use some sort of yield or yield from or async/await. You need to drive that in a way to where it can be paused and resumed later.

That's what generators are used for in Python. You need to use yield from syntax to drive the generator function. The generator function will stay inert and it'll never execute unless you have a "yield from." If you have a recur function in a task that needs to call some business logic that's going to execute like a network call or a file call, you need this yield from. This is what runs .., you can put synchronous logic in here, in your recur, and it'll execute this line, but when you have async logic, it'll call the yield from.

Now, let's we're going really fast and we I know we only have 10 minutes left, so I'm going to do my best to just keep on going. If it's uncomfortable, if you don't aren't quite tracking, it's okay. Let's get to the end and then we'll get to some questions.

We went to the from the Doist.do to the recur of the main event loop or scheduler. And now we're going to the individual task do. And the individual task do is taking us to the recur of the task. And you'll see it's empty except for return false. You have to subclass it if you want to do work, right? The the main Doer construct is almost like an abstract class in the sense that you need to subclass it, override the recur, and that's where you call your business logic function.

And the false there is interesting. If you return false, it'll keep running until you return true. Because the way that you end something is you return true cuz it's the done or not status. Return true sets done to true in the root scheduler for that task.

That's why the default is to return false. Now, this is an example from the KERIpy code base. If you've used KERI it all, you probably have to deal with database migrations. We have a Doer that does database migrations. We have a recur.

We open up the database. And then after that, so really this database stuff should be in an enter function. Whoever wrote this wasn't trying to fully embrace the life cycle functions. This is set up that really should be in a What do you call it? And you notice, where's the cleanup?

We need an enter and an exit function. This is not as good as it could be, but it's sort of sufficient for now. We run all this business logic, we run database migrate, return true. Runs the database migrations, return true, and it's done.

This is a basic task. If I do "kli migrate run" this will trigger this. It'll migrate your database version. Now, if you're going to look through the rest of the KERIpy code base, you might say, "Why do we use so many DoDoers, but not many Doers?"

And it's because of one function, doing.doify. That's kind of funny. Sam has a really fascinating naming policy. But what doify does, it essentially allows you to use yield from in the body of a function. Remember how I said structured concurrency says, "Program flow follows lexical scope."

The reason why doify is used is because it allows that sort of intuitive program flow to follow lexical scope contract to be met. And it's easy to essentially schedule a lot of subtasks in one lexical context that makes intuitive sense.

I'll give you an example here. Oh, so and actually there's a difference. This is the DoDoer function. You'll notice there's a critical difference. The reason why you use doify is because there's something that I think is my opinion it's missing from DoDoer.

I think Sam argues differently. He argues for a different or way to arrange things. Personally, I think that the way that we use doify is evidence that we should add the missing part here. See here we have self.recur, but we don't have the generator check.

Remember the generator check? If I go one slide over, here's the Doer. We say if is generator yield from. What this means is a DoDoer will not allow you to use yield from in its recur. That's why we use doify all the time. And Sam has this argument about hierarchical state machines is the right place to use state to essentially manage multiple sets of Doers, and then you have a scheduler Doer that manages other Doers as a set of Doers within a DoDoer. There's an argument there, but I don't think it's very intuitive.

I understand the argument. Maybe I just need to maybe I'm just unfamiliar and I need to learn that. But to me, the fact that we use doify so much is evidence that we should add this is generator in the Doer and see now we're in DoDoer right above here. We should put that is generator check and run it that way in my opinion. And you'll see why. Now let's get to the code. But notice the DoDoer has the same life cycle functions.

Enter, exit, clean abort sees, and we call recur. The DoDoer And you'll notice this return self.done for when a task is done, true or false. The difference between the DoDoer and the Doer that is the DoDoer you can add it's extend or remove. You can add tasks to it or remove tasks to it. It's an aggregation construct. The Doer is not.

You can see how the DoDoer is a bit of a combination between a scheduler and a task, but the Doer is just a task. There's an example. Back to the [inaudible] incept. Inside of the init function for the Incept Doer, what you'll notice is a DoDoer. We call doing.doify on the main function that runs everything.

What that does Here's the incept do. You see we have a yield from. It's pretty intuitive to co-locate a lot of that logic. Rather than create a complicated state machine and worry about the state of this Doer and that Doer and sort of schedule them at the right time, we just locate them in the same lexical scope.

That's pretty easy for humans to understand that program flow. That's why it's so handy. That's why in my opinion this really should be a recur function, and the recur should allow you to use these yields and yields from in the body of the DoDoer's recur.

That's not how it works. I actually did go through the trouble in the DID:webs repository in our tests to do it the way that Sam recommended, and it's kind of interesting. It does make you clean up your code a little bit… There's an argument there. And what it does is essentially the mental model is if you're going to do things the right idiomatic way according to Sam's idea, every place where you have a yield from should essentially be a separate Doer.

You have at least one, two, possibly three more here. And what you do is you would add all of those… See how we have a list of Doers here, and then we send it when we say call the constructor we init Doers equals Doers. What you would do instead of having all those yield froms in one function, you would make separate Doers for that sub part of the task, and you'd have that be part of the task array that you send to yourself when you're scheduling yourself.

You can see it sort of force you to lift things out, and make dependencies more explicit. It's kind of like smaller functions versus larger functions. It is more hygienic in some senses, a little bit more convoluted. It's a bit of a trade-off.

Now I'm just mention Effection. Effection is a JavaScript structured concurrency library. Now that you understand the yield from syntax and the yield and whatever, now you can see what it looks like in JavaScript cuz JavaScript has a construct that's like yield from.

If you see yield star in JavaScript, it's the same thing. It runs a co-routine from that spot. This is using the Effection structured concurrency library just like how we did doing.doify. This is part of why I lean in favor of making DoDoer.recur do allow yield from because in Effection in the structured concurrency library in TypeScript / JavaScript this whole function is like a doified function.

You can have as many yield stars, which are yield froms, as you want and just sort of naturally proceed forth in an execution chain in one function. You'll see I'm saying yield from list identifier service and we'll look at that one layer deeper.

This actually it wraps the Signify client. You'll see it has a Signify client right here. This is the wrapper. This is another thing that I think is sort of missing from HIO that we could add to it. Wouldn't be that hard to add it. Sam has actually been experimenting with the wrappers.

More recently with trying to integrate async/await into HIO. It's already done in the Effection. We say call promise and turn it into a generator. This allows you to preserve the structured concurrency guarantees even when you use promises in JavaScript. You'll have promises at sort of the edge of your architecture and all the nice stuff in between you'll have your structured concurrency guarantees.

– Just call a library you're importing that you can't control. – Exactly. Precisely. And if you look at the wallet that I demoed yesterday for my… I need a really need to wrap up. Oh. Okay, let me just run through the end.

Okay, the last thing is… Look just look at these in the slides later. You'll see this weird convention used everywhere in KERIpy the self.wind, self.talk and yield. And what it is there's something with all generators in JavaScript and in Python where you need to prime it. Prime it means advance to the first yield. You don't want the priming of your generator to run business logic.

You don't want it to come down here, so you put this yield at the front. I think that should be handled by a framework in my opinion. It's kind of weird to force people to manage that, but that's why this exists. And Okay, and I would say there's Oh, we'll cut it there.

That's HIO.

TALK XXIV

Kent-Bull-Interview

KERI Conference April 2026 · interviews · 01:07

I'm Kent Bull and my session was KERIA tutorial talking about how to set up a KERIA server, what a Signify client is, and how to make a basic lightweight wallet with Signify and KERIA. What I do differently now as a consequence of being in the KERIA space is I aggressively adopt zero passwords, no shared secrets. And I challenge myself to say, when I think I need a password, how can I replace that with the signature? If Signify TS and KERIA disappeared, what the self-sovereign identity community would lose would be a lightweight wallet implementation that makes it easy for a low barrier of entry and people to adopt this sort of architecture. We'd have to re -engineer a server architecture with an agent and a lightweight wallet implementation. It's very useful for certain, I would say constrained contexts useful technology, we'd have to re -engineer it, but that's what we'd lose if Signify and KERIA were to go away today.

Thank you.

TALK XXV

Secure Asset Transfer Protocol

Ned Smith

KERI Conference April 2026 · talks · 39:57

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Ned Smith surveys IETF Secure Asset Transfer Protocol work for tokenized assets and stablecoins. He walks the asset reference token issuance use case and stages one through three of cross-network transfer, noting how credentials such as KERI KELs may anchor trust alongside conventional certificates.

Welcome to the session on Secure Asset Transfer Protocol. I'm Ned Smith. I happen to be Sam's brother. I've been introducing myself as Sam's brother for a long time. I'm the younger one. But I have a 30-year career at Intel and then another eight years in industry. All of it related to security. Sam's the newcomer in the group to security, just so you know. And I've also been a prolific inventor. I have over 500 granted patents and I've been interested in doing some work in the IETF. Most recently I have been looking at the SATP protocol in one of the working groups in the IETF. I'm going to just give you a quick overview of what the IETF is as a standards organization and how the SATP group fits in there.

IETF is the acronym for internet engineering task force. It's really a standard setting body that's recognized internationally. It is the body that defined the internet protocol. It is all things internet originated in the IETF.

The organization itself is structured around different areas. There's a internet area, a routing area, security operations, general applications and real time. There's a few others that I didn't mention here. And within the applications in real time area, also called ART, there's a working group called Secure Asset Transfer. And they're the ones that are doing the work to define a secure asset transfer protocol.

And I'm going to talk about that the rest of this talk. To dive in here, there's a what you might think of as the canonical use case in SATP to drive the technical portions of it. I'm going to walk through this use case. It's essentially a token issuance use case.

And this token use case, it's built around this something called the asset reference token or art not to be confused with the working group. But it's a sort of a reference token for for working through the challenges associated with building systems around tokenized assets and stable coins. In this use case the token issuer in step one issues and declares an asset that is then given to an artifacts registry to make it persist there so so that the asset is known to exist. In step two, the issuer mints a token. And we're using the acronym network one or NW1 to refer to the token issuer side of this exchange.

And they mint the token and in this token, there's sort of the value part or part one, which is the part that is on the chain. And then there's something called context data, or part two, which refers to any artifacts that are not on-chain or maybe partially on-chain but in some way represent some stability to the stable coin asset.

The importance of that linking will come in a little bit later in the context of the asset network. In this case, they've minted the minted the coin and then there's an entity that is the interface to the asset management network. We're calling this, network one, asset network one. And that's where you interface to whatever is your distributed ledger technology, whether it's a public, private, or even a monolithic Network.

And then in step four, the gateway reads a token from network one, which we're calling the SATP gateway one, and that then in step five interfaces with the artifacts registry to collect the additional data that's there and it validates that prior to the asset transfer. And then in step six, this is sort of the jumping off point for actually doing the transfer over the SATP protocol. And the idea is you're transferring an asset from gateway one or the token issuance domain into another network which is receiving that asset.

What are the SATP entities? There's something called the originator which is the entity that owns the asset and then the beneficiary which is the entity that's receiving the asset. There's a set of hosts that are in the model for how that exchange happens.

There's the gateway which are the endpoints of the protocol. There's an asset network or an oracle that is the controller of all the domain assets. Anything that's relevant to the context in which that gateway operates is represented by this notion of an asset network; could be one or many hosts.

Then there's the user application or orchestrator which is the kind of the controller or the entity that's driving the asset transfer that's coordinating the resources and and giving the instructions to the gateways to present their assets on the protocol. There's another entity which I call the host owner. It's not in the first iterations of the protocol. It's not really called out directly, but it's implied in the some of the message exchanges that appear in stage one of the protocol. I'm calling it out as it's a real entity and we'll call it the host owner for now just to make the slides more clear, but it's these entities that are responsible for administering the infrastructure and it represents the entities that are responsible for that are legally responsible for the asset.

There's also something called an artifacts registry. Think of it as a directory that maintains asset transfer context and makes it u persistent for whatever additional Your investigation that's needed on the asset. It's something that provides the audit trail for the protocol.

Here is a view of the architecture for SATP. One of the main goals of this architecture is to essentially implement a an atomic two-phase commit protocol for the asset exchange. It's fundamentally decentralized. In other words, there's no assumption that the originator and the beneficiary share a trusted third party relationship.

There's a tokenized and non-tokenized asset. Not everything is on-chain per se. But it can exist and it is relevant to the exchange of assets. It needs to be auditable and and the assets need to be durable. There's also the notion of a canonical identifier, which is to say that the representation of the identifiers and the information that are known to the originator don't have to be the same as the representation of identifiers that are known to beneficiary B. In other words, the attributes that are in common have to be negotiated or created as part of the protocol.

But it's not requiring that the asset identifier has to be morphed into a namespace that's defined by the protocol or the artifacts registry. There's a need for asset versioning, pinning and roll back. That's part of the two-phase commit requirement. And There's also requirements for provenenced asset transfers. In other words, you should be able to unwind this asset transfer and be able to identify all the stages in the transfer after the transfer has happened.

As you can see in the diagram, the protocol is divided into four stages. Stage zero, stage one, two and three. And and you see the artifacts registry kind of in the middle. It's not really a separate entity. And I'll talk more about why that is in a second.

One thing to note is that the authority model from the perspective of the architecture in the first series of drafts, there is an assumption, that's implicit, that it's X.509 certificate based and so this is a bias that finds its way into the specifications and there's no normative text that says it has to be X.509 but the way that the people are thinking, their bias towards X.509 thinking, and so that tends to be an uphill battle as you contemplate integrating your KERI concepts into this. Part of the initial investigation that I did into this environment, was to look at what would it take to integrate vLEI authorization into a SATP architecture.

I sketched out how you might do that. We have the the standard GLEIF and QVI and LOU entities represented and then we have the within the domain the SAT domain you have something I'm calling the SAT domain which is whatever is the appropriate way to identify that organization. Here I've just coined the term originator.example.org or beneficiary.example.org to capture that idea.

The point is it becomes an LEI holder and then there's the OOR and the ECR entities; those are also represented. These entities aren't defined in the architecture or in any of the specifications but there is somebody, the CEO for example, that can that has the legal responsibility for the asset and there is something that is responsible for you know provisioning or issuing credentials to the hosts.

And then on the asset side we basically have some requirements, maybe the legal requirements for the asset, but the asset authority is someone that you know has legal title or ownership of the asset. They have the right to disposition the asset or to transfer it. It needs to be a non-repudiable authorization.

And there's this notion of express delegation and recorded disposition as well as effective date and some other legal requirements that have to be met in order for this protocol to work but it's not defined yet in the set of specifications of how to do that. One other note is, from the perspective of LEI, that actually the hosts are not legal entities.

They're devices and there and in LEI there isn't the notion of a credential that's appropriate for a device. But it could be anything: It could be a certificate, there could be a DID but you could also have a KERI KEL as the credential, but it's not specified in the specification that says that it can't be anything any of these, But like I said before, there is an assumption that it's probably a certificate.

[To next slide, red.] In 2023 this working group was formed and the working group sat down and said "okay, what do we want to work on, what charter should we bring to the working group" and of course one suggestion was let's start with stage zero; that's the first part of the protocol. But it became clear that people were not really clear about what should be in stage zero they said let's start where we're least confused which was stage one.

The first set of protocols begin with stage one. And I'm going to just briefly walk through stage one, two, and three. If you have questions, chime in. But it's the easy part basically. The important aspect of stage one is there's a there's sort of a handshake that says, "Hey, I want to do a transfer." You propose your transfer and then the the recipient then acknowledges the request and then you begin the transfer. Part of that proposal is to provide all of the contextual information necessary in order to perform the asset exchange. What you see here is a snapshot of the transfer protocol message and then kind of the body of it, which is the transfer claim which is where all the ugly data shows up that provides the context for the being able to transfer the asset.

The interesting thing though is the transfer context identifier is supposed to be negotiated in stage zero but we haven't defined it yet. It's just a placeholder, a string. 'tstr' means string. Then there's these other things that are showing up which refer to domain bindings. There's the verified originator entity ID and

the verified beneficiary entity ID. These are strings again undefined as to what is in the string.

Then there's a bunch of information about the context of the various devices. There's the originator's public key which presumably is there to be able to verify the identity of the originator. You got the identity the originator's identity plus the originator's public key which can be used to authenticate the identity.

And then there's some of the other entities here. There's the gateway. There's the network. There's also identities and and credentials for both the gateway, the network and the identity. And then down at the bottom there's this notion of the owner Which is not well defined but it asserts that there is some entity within the gateway that is the legal owner of the asset.

in one of the recent meetings there was observed "Hey, there's in this transfer init claim claim, there's all of this information but as we contemplate looking at stage 0 it seems like a lot of that information already exists somewhere in stage 0; do we need it all here" and that's kind of setting the stage for the next slide but basically My interest in this started with a draft that I did called SATP vLEI binding and it's essentially populating some of these values with the assumption that if I had an infrastructure that looked like KERI underneath how would I integrate that and that was the genesis of that draft. One of the other outcomes of doing the draft was to define media types for the various payloads that are CESR specific. It proposes Additions to the IANA registry for CESR payloads.

Moving on we have the stage 2 protocol which is essentially locking. Before we can actually transfer the asset, we need to lock the asset. Very simple kind of request-response locking. It's a it's a not fine-grained locking, it's a global lock around everything.

And then we have a two-phase commit protocol which includes in this the notion of minting a new asset in the beneficiary domain and burning or destroying the asset in the sender domain. There isn't this notion of "Hey, I transferred the asset." It's just bookkeeping in some sense.

Right? All right. Now the working group is almost done with the specification for the architecture and for the protocols from stage zero to stage three. They're in review by the working group and the area. And the way that the process works is when the working group believes they're done with something, they issue a last call on the specs which basically says, that the period for providing feedback is ending. Please, give us your feedback because we're going to move it to the next phase which is to deliver it to the area directors and then on to what's called the internet engineering group. IESG is their acronym. What it means is If all of those reviews are successful, then it will progress to RFC status.

Okay? Since they're kind of done with The stages 1 through 3, they're now rechartering to look at stage 0. And then in the context of that, what are the objectives or requirements for a stage 0 portion of the protocol. One of the important aspects is that this notion of a transfer context identifier needs to be defined. What does it really mean? What are the properties of an identifier for the transfer context?

How do we represent asset identification? How do we represent asset ownership and validation? How do we represent the authorization to transfer? There's also the exchange of travel rule requirements in financial contexts where it's basically saying "How do we know what information has to be included in the transfer in order for the appropriate context to be represented equally on both sides."

How do you represent gateway ownership and validate that? How do you negotiate the transfer? There's also information about the asset itself. How do you classify it? And then there's something called mutual attestation which is a future requirement but it's essentially saying "We want to be able to

get information about the environments in which the hosts operate to better minimize or mitigate risks associated with potential attackers.

Here's the stage zero protocol. I'm not going to go through this in huge detail but I want to draw your attention to step one which is a request to transfer a context and there's a tokenized artifact record which correlates to the part one of the token that gets issued. There's some information in the token that is used to to generate an identifier which we'll call the transfer context ID.

And then there's a binding of the transfer context ID to other assets. within the context of the domain, there could be a variety of digitized artifacts that are off-chain or domain specific, that are tied to the token in some way; but not in the token.

There's a binding step and then there's a propagation step in step four which is to say now that I have this context for transferring an asset I need to notify the beneficiary and say "Hey, we're going to progress with this asset transfer. That happens in step four and five. Step six is basically giving the beneficiary the opportunity to bind some of its assets to this exchange as well. We have the DR2 from the beneficiaries side that's binding assets and then it returns with its propagate in the other direction to say in the context of the current transfer transfer context that we're exchanging here are my attributes that I'm adding and what happens then in step eight is we now have the senders set of assets, the beneficiary set of assets and context and then we update the registry with the transfer context. This is saying now that we've established the context we're going to make it persist in the registry.

And that happens in step eight B. One thing to note is that the registry doesn't have to be a single registry. It doesn't have to be in essence a trusted third party. It could be separate registries. Network one could have its own choice of registries and network 2 could have a different choice of registries. They would have the same information in them because of the protocol, we've negotiated that, but they don't have to be the same registry.

That's the point I want to make here. And then we go through the rest of the stages one through three transfer the asset. Then in stage zero there's a step nine which is sort of after the asset is exchanged there's a finalise message to the registry that basically says yes the whole thing happened correctly and we're done with the transfer.

My observation here is that the stage 0 is a lot like an inception event for an asset transfer. I think some of the creative thinking around this is to take a look at the registry architecture in KERI and say what can we leverage from that architecture to influence the direction of this SATP protocol, in terms of the registry design.

And then of course just by the way since the registry contains all of this state and context can we simplify you know stage 0 protocol: you take some of the information out of there; it seems bloated otherwise. What are the desirable properties of an artifacts registry?

This is not a complete list. This is sort of current thinking and so there's an opportunity now to influence where this goes. If there are other desirable properties we can entertain adding them to the list or working to have them added to the list.

But the general sense it's append only. This seems to align with the assumptions that we make for KERI logs and transaction logs availability which is to say "Hey, I can have replicated registries." The spec is not trying to limit the structures in any way to prevent replication or redundancy auditability, canonical

preservation (we talked about that), durability, which is database concept. As well as discoverability, domain isolation, extensibility, privacy, provenance and verifiability. These are all the illities there may be some others. They're open to suggestions for what other properties you think you need.

Let's talk a little bit about future of SATP. I mentioned earlier that they are anticipating integrating of a tested hardware and and context. A little bit to observe the motivation for how that for why that's important to them. If you think about it, in an asset transfer, if there's a failure Early in the transfer, the cost of recovery is much lower than if it happens late in the transfer. You get this sort of exponentially increasing cost curve as you progress through that protocol.

One way to help to prevent failure is, at least from the perspective of attacks, is to introduce trusted platform hardware. This idea of a trusted platform blueprint which has at its core a trusted platform hardware component.

Think of the blueprint as identifying a set of hardware, firmware, software, system software components that all work together and can attest to their nature Using industry-standard attestation protocols. The blueprint says "Hey, I've got this trusted environment and I can prove to you that it's a trusted environment."

Some of the properties of this are: I have a hardware isolated execution environment. My gateway runs within this environment that is walled off or sealed off hardened against a variety of different attacks. That's a preventative approach.

The hardware that's built on top of isolated containers or what's called confidential container technology, that a lot of the vendors are providing. It turns out that all of the major silicon vendors are are supplying to the market their acronym soup of confidential container technology. Question how much do I got left? But t– Then you have there's a window of time for people to transition out.

Okay. – Sorry. All right. I got I'm almost done. We got about 10 minutes it looks like. The important point here is: the industry is building hardware supported isolated execution and the IETF expects for things like SATP gateways to run in those kinds of environments.

Further down, there's something called attestable hardware Root of Trust and something called Dice. This dice is technology that I've worked on personally. I'm one of the primary authors of the dice specifications in the industry and all of the major industry players have adopted dice.

Nvidia, ARM, AMD, Intel and then standards organizations like the TCG and OCP have all built around the notion of a dice hardware root of trust. It isn't that you have a variety of different technologies based on different vendors. They're all working to have a common open hardware representation for a hardware root of trust.

What is attestation? This is maybe the most abbreviated explanation of attestation that I've done. But in a nutshell, the way attestation works is to improve risk management. You think of it from this perspective of flow of information. You have like the SATP gateway which is the attestor.

It's providing evidence to an attestation verifier who produces a result which is an assessment of how trustworthy the platform is and then the SATP gateway, which is the the beneficiary which is referred to as the relying party in this architecture, writes their information to the attribute registry. In addition to all the other stuff that we went through in the protocol, in the registry there's also device attestation information that then could be used, a long with the other information in the transfer context used by an actuary, to make an assessment of what the risk is that's associated with this asset transfer. And if you

think about how value that could be is whenever there's an asset, whenever you're transferring assets, there's risk of something going wrong and who's responsible for for compensating something that goes wrong. Chances are in large amounts, high value asset transfers, there's going to be an actuary there that's providing some sort of compensating transaction to offset the risk.

Last slide: What are the opportunities for SATP that are sort of essentially KERI aware? I think there's three of them. One is what does it mean to have a KERI-enabled trusted platform? You might say "Hey, one of those layers should be like the KERIpy library."

And we might take a look at the KERI controller and say "Shouldn't the hardware Root of Trust and the layers that are at the fundamental bottom end of this stack be KERI capable, such that when a verifier is verifying not only the KERI-related information but also the device certificates, and so forth, that are behind the attestion and verify the infrastructure associated with those credentials are also KERI enabled That's one thing. Another is enabling the artifacts registry with ACDC technology. If you were attending Sam's talk earlier this week on the registry architecture, you noticed that there was the notion of a set of the witness-watcher pattern associated there. Which is to say: I could have a registry a bifurcated registry where one registry is acting on the behalf of the gateway way 1 and then another one is acting on behalf of gateway 2.

and then the third one is the LEI enabled asset transfer. This builds on work that I've already started in that sense where you know essentially what's what we need to do here is define how to provision the other SATP hosts, like the gateway and the network and so forth, with KERI credentials We can have an entire hierarchy that's built around the the KERI credential and the vLEI ecosystem.

That's it. Any questions? – Do you think let's say hypothetically .., I'm being brought in with a group of advisors to advise on a cap 'n trade program where they're looking to have the emissions allowances in a digital form. There are some groups advocating for a tokenized blockchain based approach. There's me arguing for ACDCs and GLEIF This country is interested in having a liquid market because allowances could be traded.

If you can reduce your emissions more, you can sell your allocations to others who cannot. They need to buy in the market. Could this potentially help facilitate a market for trading and represent allowances? The major motivation was the observation that hey, there's a bunch of blockchains out there, they're all proprietary in a sense that they have different consensus algorithms and different formats and whatever. But people want it to move their coin or their tokens from one to one network to the other and there's no industry standard way to do that.

SATP was designed with that motivation at heart. If a cap 'n trade is trying to essentially move one kind of asset from one kind of blockchain to another kind of blockchain and maybe backwards the other way, then that's just an asset exchange, which is just an extension of the asset transfer. You just do another asset transfer in the other direction and then it's an exchange.

I think it would work. It's there's no design I don't see any design goals that would prevent it from working but that particular use case might not be in the forefront of the minds of the designers. – It's probably not.

Yeah. – I have a question. Yeah. – The bias toward X.519, is there any sense in the community that that the risk of the security vulnerabilities of X.509 are materially problematic? The question is given the bias towards X.509 is there any sense of the problematic aspects of using X.509 in the forefront of the minds of the various other designers? And the answer is no. I don't think they thought about it.

Okay. That's a conversation that will come up in the context of stage 0 discussion. It's a great timing for this group to engage with that effort. That's what it boils down to. – We're having conversations with the certificate authority about integrating vLEI into group certificates.

Would that make this implementation easier if the actual certificate was there was being added if you still needed to be integrated down to the gate? It would make it easier if somehow the working group decided that they didn't need to do the extra work to allow extensibility where extensibility is needed on the credential type because right now there's some high-level wording and then some precise examples based on [inaudible] something in the middle that says "Hey, this credential structure is extensible and here's how you extend it and here's how you say that it's a different type of credential and here's the type.."

And that's something that I'm trying to get in there but they said "No, you're not going to do it in stage 1; it's already working group last call. Right? But like I said stage 0 might say "We're going to put all that information into the registry and all you really need is.. [abrupt end]

TALK XXVI

Ned-Smith-Interview

KERI Conference April 2026 · interviews · 01:37

Hi, I'm Ned Smith and I am a researcher in trusted computing. I've spent 30 years doing hardware trusted computing and I'm also interested in secure asset transfers based on protocols that are being defined in the Internet Engineering Task Force.

And my presentation today was to overview where we're at with the definition of the SATP protocol. In particular, how a SATP protocol is tied to authorization mechanisms that are important for legal responsibility when assets are transferred. One of the things that sets me apart as a researcher in trusted computing is my experience working with trust both from a hardware side but also from an infrastructure side. And focused a lot of my career around this notion of device attestation which is providing evidence about the environment in which applications or workloads run so that a remote entity can verify that the environment is trustworthy not only that the workload is trustworthy. I'm incorporating that kind of thinking into to not only the work that's being done in the ITF in the RATS working group which I chair but also in other standards and architectures that I influence.

TALK XXVII

Investigating the Game of Trust

Nicholas Racz

KERI Conference April 2026 · talks · 43:27

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Nicholas Racz reports six years investing in KERI's trust-spanning layer, framing KERI as the internet's next essential protocol. He describes capital's role in making that future real and reviews portfolio bets on industry adoption, including Provenant and healthKERI, with founders reflecting on healthcare's slow security uptake.

Okay, it is 3:15. 3:16 now. I'm going to get started. Welcome everyone to my second talk. It's going to be a lot less complicated. I can tell you that because I don't want to delve into any of that anymore. We're just going to talk about the history of Key State. We're going to talk about our philosophy, investing in the trust spanning layer, what it's been like to be at the ground floor and pushing it forward going forward into the future.

We've been here six years. We've got a partner right over here, Will Dahlquist. Thank you for making it. And this is a field report about, what we've done in this space. And thank you all for being here, everyone who's been part of our journey from the beginning.

Okay, the idea really is simple. Evan and I actually had a conversation about this earlier. Capital creates reality. It's pretty profound when you think about it, but it can simply be said that when you place money in something you believe in, you bring that thing from the future into the present. And when we saw KERI, when we found Sam out of the wreckage of the sovereign debacle, we saw a future that was worth believing in. And it was profound.

We're going to talk about the rest. This is the protocol moment like what is KERI, right? Every internet wave needed a protocol layer first. The identity layer is being built now into the open under the name KERI. And the protocol is not a product. It is a service, public service. Every wave of the internet value needed this protocol. It starts with TCP/IP. It emerges with HTTP and URL. Then we have SSL and TLS.

And finally, we have really the epitome of security in the internet as we know it in the form of KERI. In this we're making trust portable and the operators are being funded right now in this room. What is KERI for those that don't know? It's a self-certifying identifier that travels, cryptographically verifiable, portable across jurisdictions, rotatable without losing history. The protocol is stable, the governance is real, and both are open.

In essence, when we first started, there was… this is really just a spec. Sam came out with a white paper which had every bit of the ingredients that needed to establish the trust spanning layer but no one had yet built out the essential infrastructure.

And when we started actually and [via] Timothy Ruff (thank you for introducing us). We got pitched the idea of Provenant. Provenant was an idea in the mind of Randy Warshaw. He wanted to bring verifiable communications natively to SMS, RCS, calling, Voice over IP, etc. And it's only now six years later finally coming to fruition for example. Provenant is one of these three levers that we have in the KERI ecosystem.

The three actual approaches we want to take in terms of deciding what's actually investable are the following. The first is industry adoption. You have to identify a vertical which distinctly can be owned by the company we are bringing into reality. And then the second is actually technical development. We

want unique features that's only possible to be created in KERI that no one else has created before. And then finally we want to emphasize the human talent emphasizing human resources that we can attract to actually make this real.

You need all these three things. If you have tech without industry, you essentially just have nice theory, right? There's nothing to apply to. You can't help anyone's problems and you just have nothing to fund. If you have industry without tech, you essentially have synergy, synergy, synergy. You choose logo and you can't ship integration and you lose the edge. And if you have both of these, but you don't have the right human resources, you essentially have a beautiful idea that completely flames out. I've seen this personally where visions of really advanced technical ideas pushing forward Proof of Work. For example, I visited New Zealand twice to fund a specific company that was pushing the edge of what was possible in blockchain, but the team just wasn't right. They had the ear of the New Zealand government but they could not bring it home because the leader did not have the appropriate direction to anchor in real economic realities. Right now within our strategy we've identified three key Roots of Trust. The first is GSMA.

And within each one of these entities we've actually funded, we really wanted to direct them to actually colonize and leverage the trust that a Root of Trust has within a given ecosystem. The GSMA route within telco is a clear showstopper, right? When you have a phone number that's verifiable, that means you have an OOBI introduction at your fingertips and that KERI is not just a figment of the internet. It actually starts to span multiple different communication layers. It's a great utility there. Secondly, as well, we have DirectTrust and Vital Alliance being emphasized by healthKERI.

– Vital Initiative. – Was that changed? – DirectTrust. – Okay. Vital initiative. Excuse me. Essentially DirectTrust leading this were Jared and team are… well maybe Jared do you want to speak about it a bit? – Yeah I'll tell you the thing that I found most interesting about the work we're doing with DirectTrust is how [inaudible] it is to work [inaudible] If you think about certain evolutionary tactics in nature, there are different paths like lobsters and crabs and claws. They don't have the same ancestor, but they came to the same use case at the end of it.

– You kind of do have the same ancestor in Key State [inaudible] but it wasn't like I ever worked with ratings and how are you guys tackling problem in your industry, which I think gives stronger validation to both missions. The way that you win with a B2B focused KERI play is to go from the regulators, the standard bears in the industry and convert them to the project and then they will bring the rest of the industry into – Right. I mean I say this a lot. I think the goal for me is promoting harmony within different investments in our ecosystem. We want canopy shyness. Provenant doesn't tread on healthKERI's toes.

healthKERI doesn't tread on a yet unnamed third company we're going to fund in defense's toes. I can't wait to bring you all to show off this next team. They're going to be fantastic. And the idea is really getting back to those previous principles, you We have this beautiful network effect in the form of KERI. We can share in the things we build, the learnings we have. And you could even hunt together, but what you catch is what you catch, right? We want independent revenue streams that are converting independent users and adding to the overall network effect of KERI as we see it. Which builds into also our final Root of Trust which would be the SEDI, state-endorsed digital identity. Now that we have organizational identity via the vLEI and personal identity endorsed by the state of Utah, we have both ends of what essentially amounts to a trust ecosystem. I think we can actually start really creating new trust services that previously weren't possible.

Let's go first into Provenant. This was the hardest. We had to break really tough ground. There was like, in Texas, right? I actually was born in Lubbock, Texas, right? And the Spanish actually colonized that and they called it Llano Estacado, which means the stake plains. And the reason why they called it is because there was this thin layer and by thin it's like several inches thick of capstone that covered the entire west. Breaking through that capstone is tough. But once you did, you could put a stake in the ground and know that it would be there in perpetuity.

And for us, Provenant was exactly that. We've been hammering down on this stake for like literally six years now and with GSMA we finally have it in the ground. Building out open verifiable communications did not just come overnight. You Initially we actually thought we were gonna have an exit within a year, right?

And the reason being is that we had a precedent. Timothy Ruff had come just recently out of Credential Master selling to Salesforce. A beautiful exit within nine months, wasn't it? 18. Oh, okay. Sorry. Excuse me. But there was a really quick turnaround.

A lot of the members of Key State itself actually thought, okay, great. We'll do the same play. We're going to appeal to one of these regulatory bodies that needs the services Provenant is bringing and they'll snap us up and we'll have a nice turnaround. But it didn't turn out that way. We actually had to pivot our expectations on what Provenant was into a much deeper technical play in colonizing the GSMA Root of Trust.

And really also we funded it quite heavily but they also had not only to pierce telecom but they also had to build the first QVIs, the first wallets the first means of I'm sorry – I don't mean to interrupt but I'll just add a comment when you have a break – Right, I mean just go ahead – Think about Provenant as often happens with startups and by the way the co-founder of Credential Master sitting back there in the corner, Allan. And we went from zero to exit in less than two years, which was amazing. But if that was your expectation?

– We and this was a naive expectation. I'm being quite honest with you, actually. – But Provenant pivoted twice. – Provenant pivoted from original telecom focus to thinking that being the world's first QVI was a business.

It wasn't. It was a good thing. It was an important thing. They broke important ground business and then pivoting back to telecom. And it's dangerous and hard to make pivots and some investors will evaluate a startup as to whether they've made any pivots and if they haven't that they haven't been listening to the market.

– I actually don't think that's quite right. I think that they did pivot away from .., oh my goodness this is failing me, but anyways but, they did pivot from their first approach but when they had to become the first QVI it was a necessary step they had to take to actually convince the regulators that existed about the legitimacy of the vLEI and essentially the leverage to convince them to use KERI – I'm in agreement with that the pivot was where they would make money They thought there would be a business in one place, pivoted to that.

Decided it was and decided to pivot away from the business model. – That's right. Cool. Thanks. Exactly. And going into this, they had to build alongside their product, wallet infrastructure, vLEI and the first QVI. They had to build Java SDKs for KERI, these KERIpy libraries to be usable. And then ultimately what amounted to the Verifiable Voice Protocol, which is what OVC itself is using to have branded calling, verifiable branded calling.

And telecom as an industry is a pretty substantially in need of what KERI provides. You're seeing a 25% decline year-over-year from 2024 to 2025 in terms of SMS trust. 72% of US consumers, including myself, I don't answer phone calls that I don't recognize. I'm sure. Do if you guys get a phone call that you don't recognize, do you answer it?

– Only if it's from China. – Okay. They might have a cookie for you. You never know. Really it says to how much this public utility has been abused and how much it needs a trust infrastructure like KERI. There's 80 billion in fraud losses in 2025 and STIR/SHAKEN as well was just insufficient.

It's built on X.509-based infrastructure. It's not enough to actually solve the problem of spam. Mr. spam guy here is nodding his head in approval. Cool. Now let's go to healthKERI. HealthKERI was actually our second investment.

Sam came to us. He said hey I've got a brilliant proposal. Phil Feairheller is going to be the CTO. Jared Jeffery is going to be the CEO and we're going to take healthcare headon. And Jared maybe you could talk about it.

What was your initial expectation founding healthKERI? What were your thoughts there? – We've pivoted. We're on our third. Right. We were wrong twice. I think we'll probably come back to some of those concepts later. But one of the things that we initially thought was that the the pain of insecurity in this industry would be sufficient to drive adoption of new technology. What we found very quickly was the industry gives a whole lot of lip service to this issue of patient harm by way of cyber security. And in the boardroom that all goes [inaudible] for us as we kind of initially started this I was like I believe that KERI can solve this problem.

But I've always tried to temper the expectations of my team to know that… listen, healtKERI runs on cold glasses the industry moves very slow [inaudible] technologies and that gets worse when you're talking about technologies that identity – Do you think, and this just occurred to me actually, the notion of data loyalty could actually whip these guys into shape? – Yes probably, it's going to be a long road to get them to fight on something like that, it would have to be forced.

– But that's what I mean. Enforced from SEDI down, regulators down. – Yeah. It would have to be regulatory, – Anyways. Oh, sorry. Go ahead. – See I spent years a few more understanding [inaudible] the issue is the consumer of what to ask for unless that really gets in front You don't know what to ask for push these companies to move that fast – Right which is why I think the data loyalty framework network if applied could actually incentivize right.

– What happened to the data by right? How much of privacy to give up between just two weeks of March 2020? Just look at that before COVID during COVID after COVID. If you go through those parameters and understand how people are interacting and secondly how fragmented the digital touch points are.

That's another area to look into and say okay where is the commonality getting in where is the critical mass? These are all critical inputs because even if you look at from their angle the digital engagements has been completely different.

There is opportunity here but it comes with letting the consumer know what to ask. – Very good feedback. Thank you. I appreciate that. Well that aside, excuse me for bringing us off track. Bringing this back healthKERI. We're still high on contributing to Provenant. There's a lot of liquidity milling around in the macroeconomic space. We place with healthKERI as well. Jared, like Jared said, you had to pivot several times. I think Jared epitomizes actually a tenacity that I think is admirable because the amount

of funding we actually gave Jared as opposed to Provenant was remarkably less. And I think what they've done to produce is substantially more proportional to the amount of capital that we've actually contributed. Jared is an exemplary showcase of what a founder should be like in order to survive tough times and we'll go over like what I learned from that later.

healthKERI just the overview: quantum ready, phish-proof network security for healthcare data layer. You've got production watchers, you've got witnesses, you've got cryptographically signed and transit gateways. The vertical is healthcare obviously, the layer is like an overlay essentially and the idea is to sign every transaction.

Again they built production witnesses you have watchers at the edge. In-transit gateways and as well I mean the Locksmith stack that Phil has demonstrated several times is extensive. This is a full end to end credential life cycle management for the healthcare ecosystem.

– And can I just put a comment on that one? One of the questions that Phil got a couple of times he's been demoing the the wallet is the question of do you think this is going to be the end UI for the wallet? And the answer is no.

Absolutely not. But we can't worry about UI challenges until this exists. Part of the reason we put that into the open source is because I mean I'm operating on a very resource constraint – Shoestring budget is what he means.

– We produce because we focus on the things that matter most. Part of the reason we put that into open source understanding that right now the UI is very KERI developer oriented is to invite this community to come together and say okay let's do the UI work on this right let's make this thing sing for all the different use cases that will exist in the ecosystem.

– Do you think that the UI itself has substantially been influenced by the more you learn about KERI as time goes on? – Yes and no. I think really what we found is as we grow we need to hide KERI to everyone so that for them it feels like everything else that they're used to. I so during SEDI conference Phil's demonstration of how to log into a payer portal with Senn Yep. I don't actually think that any of the legislators that watched that presentation had any idea the magic that was happening there. And that was by design, right? We wanted them to look at this and we wanted them to say, "Yeah, I logged in with Google this exact same way."

Exactly. Perfect. All of that to say that the stuff that we put into the open source, the watchers, the witnesses, the wallet is an invitation. It's an open invitation to the industry to say go build on top of this. We've given you the production ready code to go pay – And didn't you also have several feedback from potential customers and adopters of your technology that they wouldn't even adopt the technology if it wasn't open source?

– Yeah. one of the things that everyone in, entrepreneurship, particularly tech, has to contend with now is that the stock standard mode of defensibility for companies has completely dissolved in the last year. And it used to be that you were going to stand a proprietary code, you would get a patent on those things, and then you would basically rinsing of that patent with your code.

Claude makes that completely useless. - I have watched videos of people just walking through the UI of a platform with an AI agent and the agent can then on the back end code that for you. This defensible mode of the last 20 years is no more. Which means I think that you need to find other modes. And in so

doing, the code that you used to hold on to so dearly now becomes only valuable if you put it into open source and let the community, let the, not the AI brain, but the human brain and collective make that code more valuable.

Jared's pointing at something very essential that Key State understood getting into this, that most of these funds that we put into these companies would be going to open source development. And there's a fine balance. Not everything should be open source.

There's a time and a place for IP to be leveraged. (IP = Intellectual Property) – Our gateways are not open source. – Right. And exactly like the open source is to invite people to participate in the ecosystem and you can provide what is uniquely yours on top.

– Just wanted to follow up to that source code is not patentable. It's copyrightable. It's a big difference. – I think you are the foremost authority in making that declarative statement. – Anyway, the issue is that's how open source works is by copyright.

Because since you assert the copyright, you can decide what to do with it. – Sure. And it depends on the license you use for said copyright, right? – That's your choice. – But you're asserting ownership in order to give it away. But you have to assert the ownership first.

– I want to throw something out in an assertion that I would love to have challenged if someone disagrees with it and it's about where open source is appropriate and where capturing IP is appropriate where the line is and I think there's actually a new line a new bright line because of AI and I think it's bright and let me just throw it out there as an idea and that is if your code, your protocol is something that you need multiple parties to adopt. There's an issuer holder verifier or there's two sides. If it's a two side, if it's multi-sided at all, I don't think there's a choice anymore.

But you have to open source – From strategic necessity. – Otherwise, you're trying to be the platform, what I call EUM: “Everybody Use Me” and I will be in the middle and everybody use me to be the bridge of everything. But that's only if it's multi-sided.

I think it's 180 degrees different if it's single-sided. I think if you do not protect your architecture, you copyright code, but you can patent architecture, right? You can patent architecture and if it's single-sided architecture and you do not patent it, you're going to be destroyed in a New York minute by every AI copier… it's too easy for them to just take it and do it. And I used to feel that everything was open source and make fun of everything IP, But now I'm starting a company to take advantage of the SEDI momentum and I started doing product design and I invented something and I'm like, "Holy cow, this is single-sided. If I come to market with this product without any IP, I'm going to get exploded in a minute."

I just defensively, apologies to you, Steve, because I've been criticizing you for your very pro-IP perspective, but I realize the distinction. It's whether it's a single-sided product or multi-sided. Multi-sided, open source, single-sided, you got to patent it any way you can or you're going to get raped as soon as it gets public.

– Robert. – Just a question to your challenge, and I'm not saying it's either way, but would you say the FICO score has to be open source now since that's proprietary yet multiple sides? – No, it's singlesided. Each one of those credit rating authorities has their own proprietary way of doing it single-sided to them in their own environment. That's why they keep it proprietary.

That's why there's three. If it was open source, everyone would use the same one, but they each have a proprietary one. That's a good example, – Steve. – Let me follow on to answer your question, but I agree with Timothy.

– We agree. – Yes. But the issue is: the reason it's good to have KERI open source is because your goal is uniform adoption. – Multi-sided. – Yes multi-sided. I'm just trying to put it in a different language that you want everybody to use it because then you become interoperable and that's different than coming up with secret sauce to do something and you want to keep that protected. And so both the app world and the open- source world are indispensable tools. We use a different time.

– Right. Exactly. The timing matters and to what degree. And honestly, it's still something that I think we're working on in terms of our understanding, but Jared and the healthKERI team, I think, have the right approach. A lot of this sort of SEDI support for example with the watchers the witnesses, the wallets, it needs this level of open-source licensing to allow for that network effect to propagate and again we knew that coming in that this was going to be a little bit of this sort of allowing the fruits of our labor to be freely appreciated.

On healthcare in particular to speak on what the opportunity was. If breaches were a contest, healthcare would be numero uno. The third party ransomware origin is pretty extensively documented. There's hundreds of millions of US patient healthcare records stolen. And the most serious of attacks actually affect real life care environments. Jared, I think didn't we have the first case of someone dying from ransomware?

– There have been a number of cases where they said this probably was because of ransomware. This last year BBC finally put out an article and said, "No, we can clearly state that this person died to the hospital they were supposed to go to but couldn't."

They would have survived. Yes, ransonware, and it's worse than that. Most of these attacks are not individuals looking [inaudible] Most of these attacks are nation state actors that look at healthcare as a critical infrastructure.

It's the cyberwar. – And this is a key thing as well. One of the big and we're going to talk about macroeconomics in a bit. One of the big investment thesises that's emerging as we speak is defense. We are literally as we're in World War III right now. It's not completely physical. It's not ballistic necessarily. It's more information based, state-based actors have been performing these attacks for decades now. I mean North Korea, Iran, classic examples this is not just like fanning the fear flames but it's actually these are real outcomes that we have to protect against.

Here's a table so to speak of sort of where the capital has been moved. We still have some gaps that we've identified. I think we're addressing the wallets at the edge with Locksmith in particular. I want to see more use cases emerge that feature wallets and credentials being used in production. I honestly think the gap in supply chain is massive. Verifiable.Trade Foundation, led by Stephan Wolf has been spearheading this sort of approach now for three years now. Ever since Stephan Wolf left GLEIF and what they're building to will be a multi-billion dollar opportunity. It just needs to be realized and the problem of verifiable bills of lading is one in particular that's quite hairy.

Getting that recognized across different domains of trust. And then again having capital allotments to future SEDI initiatives is something that will open the floodgates for startups and also KERI as a whole. Utah is just the start.

We're going to go into what this taught us. When we first started most of the members of Key State Capital were very asset-rich. The crypto boom had just finished in 2020. We were sitting on our laurels. We have this incredible technology in form of KERI and we actually did not properly allocate resources in a way that was considering the fact that this flood of capital was going to subside and it did.

It did in a way that was substantial. We're talking about an order of magnitude less capital available just in terms of depreciation of assets and how we allocated risk. For me that's unacceptable going forward. It's not something that I can do for myself, I can't do it for my family, I can't do it for my founders and the industry as a whole. When capital actually creates opportunities and we're allocating capital irresponsibly, it creates danger for our entire ecosystem.

We need to actually have a means of measuring the risks we take in an appropriate fashion. Another thing that we've learned is that before the contract is set, no one's your friend. It's really easy, especially when you're in these sort of highly intellectual pursuits where you have to share ideas in order to validate and verify their authenticity. But when you're in a business negotiation, all all hats are off. It's a matter of survival in many ways. And you two are feeling each other out in real time as to the integrity of each one.

That respect is established in the negotiation process. If you can't stand up for yourself and have, as Timothy says, the crucial conversations, then there is just no basis of trust. And when the deluge becomes the desert, then people are pointing fingers. You're like, why is this happening? There's a lot of confusion, and you don't have a relationship that can actually stand the test of time.

A lesson there. Knowing your founders. This is something that I've really learned personally. It's amazing how distinct every single one of our entrepreneurs are and understanding their nuances, understanding what makes them tick, understanding their strengths and their weaknesses both while you're evaluating and also after you've placed capital. You don't just go away after you place money.

Having equity in a company means being part of that company. If you're not actively contributing to the value generating process, you're not boots on the ground necessarily selling this stuff, but you can open opportunities, you can introduce new capital. And you can learn how to speak with your founders. Communication is so key. And so maintaining that dialogue and checking in and keeping things accountable is the way we've kept this ship afloat. We've kept this thing on track. Whenever something is needed, we always have an ability to call each other and that's essential.

Knowing your founders, communicating. We heard this like two distinct times. Obviously you invest to allow people to create something new, but you cannot be attached necessarily to the first iteration of that vision. You have to be able to pivot what you've created and formed an emotional attachment with when you get enough signals from the marketplace that you're not going to make it or this is not working.

Essentially when you have creativity without flexibility it's just an attachment and this is… sorry go ahead Jared – One of the things that I, one of my mantras is that creation is a destructive process by virtue of creating you will tear down and try again, creation is a destructive process. – Right to make an omelet you got to break a few eggs.

And finally, I think the key thing is patience. When you have a vision for something that is special, when you've you've tested it's true, you've done what you can. Having the wisdom to know when not to do things, when to just wait for things to mature over time, and to let your founders do what they do best, I think is the key. We're only just now after five years of investing, seeing fruits from this labor. And I think

it's going to be accelerated growth. I think we're in our teenage years, so to speak. And we'll see how far our core companies go but I think being patient is one of the best virtues you can have as an investor.

And these are the three principles I'm looking at, right: identifying the right market finding the right technical ecosystem and then having the right team. And so when you have all these three things, you have a recipe for success, I think, in terms of finding an investable opportunity in this space.

We've been funding the teams from the front lines. We're seeing that KERI becoming the trust spanning layer of the internet. And the job is to keep filling in the gaps, recruiting new Roots of Trust and sharpening our filter, making sure that we can understand what is a good founder, what is a bad founder, a good opportunity, a bad opportunity, and so on and so forth. We're still looking for investable opportunities. If you have any, please come my way.

I'll give you a card and we'll talk. Thank you all for your time. Any questions? – You work with other try and help them understand what the import of the digital identity ecosystem is. What are some of the biggest things that you hear in terms of reticence to invest?

What are the complaints you hear from other investors when you bring to them identity based deals like… – Well I think the uncertainty the technical barrier is big, having an ability to communicate what this is in plain language to someone that isn't technical is one of the biggest barriers, I think also liquidity is an unconscious limiting factor in a lot of this right right now, it's not the best time to raise money. If you have adoption, if you have proven revenue streams, you can raise, but it's this is not a zero interest rate environment any longer. I think a lot of this comes down to trust.

You have to build a relationship actually with the person you're introducing the idea to. If someone knows that you're a smart guy, they trust you. They may not know what you're talking about. They may not know what a key rotation is, what a KEL is, any of this stuff, but if they have your trust, then I think you can convince them to place. And that's how literally what it comes down to. But… what's up, Will?

– I was going to add to that for Jared's question here, too. From my experience, conversations and trying to persuade investors to take a leap into this industry and this ecosystem, the largest hesitation has been just the vast scope of the problem that is trying to be addressed and solved and convincing them of the idea that there is a technology, or a protocol that has the ability to address that issue that they see as just almost unsolvable.

– Right, or they just presume that it's always going to be this way… – And then there are so many others trying to do it right the fact that there are so many that have tried and failed or have tried and succeeded at various levels. That's been the largest amount of hesitancy that I've seen.

– Two questions that probably are related, a lot of times investors during a pitch or whatever will ask about exit strategy. Meaning when are they going to get paid back as an investor basically, and this is the kind of bet that may take a while to pay off [inaudible] …investor perspective and so be … your color on that. And then the other one especially you know perhaps with healthKERI and Provenant that are starting to gain some traction.

It's not going to be too long before they're looking for A rounds. How do you see your role in helping secure A rounds? – Well I think one of the interesting things that… In a traditional equity market, exit is a binary, right? You either have an exit or you're waiting for one. You can be bought out. In private markets… – Also a valid exit.

– Not really, honestly. Dividends are a bad, they're a poor choice. Every time you take a dividend, you're taking away from the reinvestment opportunity of the company to actually push it forward. – Going public is a long shot for many companies. What do you look for an exit strategies pitch?

– In this case, it's like just building as much value as you possibly can as an acquisition target is smart. I think there's a lot of potential innovations we can actually leverage around tokenized revenue that I think are fascinating.

In this case, if you actually have market adoption in terms of infrastructure that has a longterm productive yield and that's an asset and if you can actually sell that up front, right, you can have liquidity without necessarily having an exit. And I think within five years there will be plenty of these sort of new sort of levers that companies can actually pull that previously they would never even dream of doing, right? I think it's going to be less of a binary thing and more of a… it'll be more of a liquidity friendly equity investment market.

But that's out of my scope. I don't know whether or not that's coming. – And then with the A round, if you have just a thought, or we can take this offline if we're out of time. I'll get back to you on that. – Okay, cool.

TALK XXVIII

Nicholas-Racz-Interview

KERI Conference April 2026 · interviews · 01:29

Hi, I'm Nicholas Rotz, Senior Partner at KeyState Capital. We had two talks. The first was about data registries and the governance. And the second talk was investing in the trust spanning protocol, which is KERI. The thing we do differently that differentiates us from the marketplace is the fact that one, we're a series of dedicated investors who have a passion and a vision for what the space is going to be like. This is shown in our products like the web of trust map, verifiable smart contract, as well as potential designs on governance that this space desperately needs. You're not going to see a more engaged investor in the KERI ecosystem. If we were never here, so we're talking about it's a wonderful life George Bailey, which she was never born, that sort of reality.

There would be no QVIs. There would be no VLEI. There would be no open verifiable communication. There would be no provenant. There would be no healthcare because there would be no one to actually place the capital for these things to happen. I'm not saying that we are the sum total of everything in this ecosystem, but it did need the catalyst of contribution that we were willing to put up or shut up about. Without us here, we were a necessary but not sufficient part of the KERI ecosystem.

TALK XXIX

ACDCs for Authorization

Phil Feairheller

KERI Conference April 2026 · talks · 26:36

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Phil Feairheller demonstrates healthKERI's Locksmith wallet using ACDC credentials for OAuth-style login on a mock healthcare portal. He shows credential offers with signed receipts before disclosure, seamless KERI key rotation across sites, and how vLEI-based profiles can augment UDAP client registration beyond X.509 certificates.

Hi everyone. For those of you who don't know me, I'm Phil Feairheller, CTO and co-founder of healthKERI. And those of you who are presenters will know that we had to submit our talks ahead of time to get the slides approved. I submitted a talk on ACDCs for authentication. But I'm going to pull a little bit of a fast one and switch it up because there I think is a much more important topic that we need to discuss and that is HIO. Why Sam? I'm just kidding. I'm not going to do.

Totally kidding. I'm actually doing ACDCs for authentication. Okay, this talk is based on the demo that we did for the SEDI conference and I have it here. I'm just going to run the video of it real quick and talk through what we did.

Let me get it started. In this demo, we start with our Locksmith wallet. We show that the wallet is holding a couple of credentials, a vLEI credential and a fake Utah SEDI health identification card credential and the various fields that are in it.

And then we switch over to the site which is a fake healthcare portal and demonstrate how we could use those credentials to first create an account and second to authenticate and log into the account from there on; go through the fact that you could allow the user to type in the information and suffer through typos and everything else or you could use the SEDI credential and get accurate data and much more secure connections. In addition to that, we went over the fact that we could introduce at the beginning of the OAuth flow an offer stage that allows me to offer the credential without sharing the data and require a signed receipt back from the site that they agree to my terms and conditions before I give them my credential and all the data in it.

That's what we're doing here. We're doing the offer. We also demonstrated that the site is built so that anyone using any other KERI tool can create an offer and upload it. You don't need the plugin. We just use the plugin make it work a little more smoothly. And then here they have returned the signed receipt.

You can view the receipt in the accepted offer section and then submit the full credential and now you're logged into the site. You've created your account. All the data is extracted from the credential and you are now signed in. And then you can go through the process and sign in again. And the point here was that we made it such that the process of using a any kind of ACDC credential for this form of authorization can be just as easy as using a passkey, for example, but of course much more secure because it's based on a KERI identifier through the stuff I'm going to show next. Key rotations work seamlessly. You can do them anytime you want. And if you use this AID in this account, (how do I go next? There we go.) If you use this AID for multiple sites that you've logged into, you do one key rotation and they'll all get your new keys. You don't have to worry about changing a password, remembering passwords for many different sites, etc.

That demo was something that we kind of backed into because of the work that we've been doing in healthcare. What I want to cover today is how we got to that by starting to work with integrating the vLEI into the OAuth protocol for a specific purpose related sorry specific per purpose related to these protocols. But then we realized that if we just keep going and extend it a little bit further into OAuth, we could do end user authentication. These protocols I realized this morning looking at my slide you've all heard about them now I think at least three different times in three different sessions. I'm going to go over them very quickly but to make a very specific point about them. HL7 FHIR think of it as REST APIs for healthcare very simple but extensible data model specific to healthcare. It supports real-time APIs documents and messages.

It is actually written into HSS regulations and it was informed by all of these other standards that have been used in healthcare for 30 years, Mark. 20 years. The old way, Mark explained this in another session yesterday. The old way was is that as they were starting to exchange more and more data, they were just exchanging documents over these protocols and the documents were just getting bigger and bigger because they had to send the whole thing. FHIR, what it did was allow people to send distinct sections of records that were much easier to transmit and of course modernized because they're REST APIs. The FHIR modules are broken down into these different categories. You have your foundation and base resources and then of course you have the specific resources that are defined for things like clinical, financial, specialized resources. And you can see all the categories here. They're not particularly interesting to us or KERI, but I wanted to point out that what we're trying to accomplish is the ability to do authorization at a level that's as low as this. Very particularly, we are working on something called payer-to-payer access, which allows payers to, for example, if you switch from one payer to another, get your information over to the other payer much more seamlessly than happens today.

But we only want the two companies to be able to exchange data for payer-to-payer, not for anything else. We're able to dig in at this level and create authorizations very specific to the purpose of use. Okay. And what got us here? What got us here was CMS-0057-F final rule. Easy to say and it mandates that Medicare Advantage, Medicaid and CHIP plans.

Anyone that interacts with these people have to implement FHIR-based APIs to streamline prior off and accelerate decision times. You have the impacted players here as I said anyone who works with these people. The one that's probably most interesting to folks in the industry is the Prior Auth because that's always been a sticking point. Takes forever. It's manual in some cases. They want this to be automated much more quickly.

Patient access, so getting into patients being able to get a hold of their own data at the FHIR level. But then the one that we focused on because it's the easiest to get pilots going, it's the most constrained use case was the payer-to-payer data exchange. This mandate is forcing by January of 2027 health plans to implement this. That was our opportunity and what gave us the idea, actually Mark the idea, that we could integrate the vLEI into this, to accelerate the connections that are now going to be needed between these payers and then down the road between payers and providers and then down the road even further between patients and payers and providers. You're starting to see the number of connections that are going to be needed. Making those connections quickly and securely would be a limiting factor without what we're trying to implement.

OAuth 2.0, I don't really need to go into this, most of you probably are aware of what it is, legacy, kind of like using the word legacy, authorization authentication protocols, essential for interoperability and that's the key so I think the biggest lesson learned for healthKERI here is meeting the customer where they

are. A little history when we first started out we created a product called RACK: Routing Authenticity and Confidentiality with KERI. I like making acronyms, too.

And what that was, is a API gateway that wrapped every transaction in an ESSR packet. Signed and encrypted to a specific destination and provided full security between those two endpoints. No need for TLS, no need for VPNs, anything else. And it also created a cryptographically verifiable audit log of every single transaction.

Nobody wanted to use it because it was not interoperable. What we have done over the course of the last six months is slowly back our way out and meet the customer where they are by integrating with OAuth because FHIR requires or mandates or includes the ability to use OAuth for authentication a protocol I'm about to talk about next.

UDAP had taken OAuth and added five or six, I only mentioned four here, new profiles that allowed the use of X.509s embedded into these different flows to do authentication. Mark Scrimshire came to us and said that seems to work, but what happens when everyone hits their 47-day limit and they have to rotate all their certs? What happens? As the companies we were talking to, they're like, "I have to manage my certs, but now I got to worry about everyone else's certs. How what do I do when they all have to rotate? How do we reestablish these connections?"

he identified that the vLEI would be the perfect opportunity to introduce true organizational identity and KERI to solve this problem in a very similar way that UDAP did, but replace X.509s with the vLEI. And there's quite a bit more detail underneath. I'm not going to get into it, but we have other credentials that chain off of vLEIs to define the purpose of use to allow them to be that more specific authorization I was talking about, but we don't need to go to that level of detail. These are the four protocols or profiles that were defined in UDAP that we are going to be augmenting or replacing.

The first one and this is where we were going to stop before we decided to do something for the SEDI presentation and that is dynamic client registration. In OAuth is what is it RFC7591 dynamic client registration that allows people without having a developer go to a developer portal and register your client ID get a secret or enter a secret get it back so that your client app can now make token requests. There is a RFC that allows for that to be dynamic.

What UDAP had done was say "Okay, if we insert a signed statement from the client and a reference to their X509 certs if we have an ecosystem of CAs that we trust for this purpose, we can allow a client to be registered dynamically and and automatically." We saw that and said "Okay, cool, works great, just replace X.509s with vLEIs and now we're on to something." That's the first thing we did. We automated the ability, payer-to- payer, remember that's what we're focusing on. We automated the ability for one payer to submit a credential to another payer, and without any human involved, be registered to start interacting payer-to-payer APIs and from that point down, the rest of their stack stays the same. They're using they're using FHIR APIs. They're using token. They're doing the token requests because this is client authentication, so there's no authorization-code step first.

That worked great. Then when we decided that okay, that authorization server works great. There's a real good opportunity here to insert ahead of, or after client authentication, after client registration and then client authentication, to do user authentication. They had JWT-based user authentication which allowed for a cert to be used to sign a JWT that would then assert who the user is. You could do the user authorization flow which involves getting a code first then it redirects back to you and then you use that code to get the token. We replaced that with vLEI as well and that's what I demoed here. The last step was, after having conversations with the fine folks in the state of Utah, George and Chris, we

realized that there was still one problem left to solve and that was "Okay, if we're using these credentials, how do we still know that our data is being protected?" And that's when we added a new step that's not part of OAuth, the credential offer.

Doing the credential offer allows us to offer a credential based on a schema. The structure of the fields, a cryptographic commitment to what the data will be and require the site to sign your terms and conditions with that cryptographic commitment before you reveal the data to them. Then once you reveal the data, they can verify that yep, it matches the schema. Yep, that's the data they promised to give me without revealing the data. Now I can register them and allow them through.

I pretty much just talked about this but the client credentials flow with UDAP and vLEI involved, like I said, dynamic client registration with vLEI, we do a sign inserted inside of the client registration request, it's slash register is an IPEX presentation in the OAuth registration request itself the flip side of this, one of the things that UDAP had that we borrowed from, is a .well-known for the server so that we could identify the server. We can avoid having a server that we get a man in the middle or whatever. In our case, we just have in the.well-known an OOBI that identifies the AID of the server that we're going to be interacting with.

Pardon me. Oh, so then finally, and most importantly in this case, the client ID is mapped to the SAID of the presented credential. The SAID of the presented credential. And there's no client secret. Every token request from here on out is a signed request. You eliminate a shared secret from the equation and you have fully signed token requests. Now of course our hope is that once they see the power of this they'll say okay now that I can sign these token requests can I sign all my other requests and then we'll start selling them RACK.

I just touched on the client credentials flow. Every other request after that is a IPEX presentation signed and the client and server key rotations are supported by the watcher network and I'll talk about how we implemented that to enable connectedness as trust at scale specifically for purpose of use and these extensions to the OAuth protocol.

Then the logical next step authorization code flow. Let the user now leverage what we've done to log in with their vLEI credentials. Okay, good. We extended the model to support user authentication with data privacy of doing the offer ahead of the actual credential presentation. This is a typical OAuth flow. The user the resource owner goes to the client application. The user gets redirected to the authorization server. They say approve. It comes back here. The client then says, "Okay, here's the code. Can I have a token?" You get the token and then you send that over to the FHIR resource server and you get your data because the resource server is able to talk to the authorization server behind the scenes and say, "Is this a valid token!" That's how it works. This is a… I stole this diagram. Actually I made it but I looked at someone else's.

This is what it looks like with KERI and the vLEI. KERI and ACDC. In this case the resource owner has a KERI database. The client application has a KERI database and the authorization server has a KERI database. In this case when he is redirected over there he submits a credential instead of username and password. You've all done this with log on with Google, right? This is the exact same thing but it's now log on with your ACDC credential.

Then you get redirected and everything works the same. But the really important part here is these arrows. Everybody is watching the watcher network and each other's witnesses so that when he decides to rotate his keys, he can do it every hour if he wants to. We actually did that in a in a pilot we

were running. It was really fun. He can rotate every hour if he wants to. It'll get propagated to his witnesses. Everyone else's watchers will pick it up and now it's still fully secure and you have new keys as frequently as you want.

Yes, you can hyper churn your credentials. And Sam, your voice is sounding better. [laughter] IPEX issuance and presentation exchange protocol. It's a protocol built on top of KERI EXN messages and it allows this flow… I stole this table from the the ACDC spec.

It allows this flow of messages to negotiate credential exchange between two parties. We are only starting at the offer phase and you can see where you can start. That's the Ys in here. And you can see after an apply you can spurn it. I love that you use the word spurn. (I don't know why I just do.)

You can offer it. You can either spurn or agree to the offer and then you can grant the credential. And then once you grant the credential, this is an important step, you admit it, which sends the admission back to the person.

You have a receipt that they accepted your credential. What we've leveraged with IPEX, one of the I think the driving factors for IPEX was graduated disclosure, chain link confidentiality, and contingent disclosure on them accepting your offer, assigned acceptance of your offer. And then once you embed the rules inside of your credential, the metadata credential that you exchange back and forth during these exchanges, you get contractually protected disclosure as well. And that's what we're leveraging here and I just said that. This just details the contractually protected disclosure. The user offers the terms and conditions, no data exposed. You get a cryptographic commitment in the offer to the structure of the data hash of what the data will be and then terms and conditions which are being enforced. I I think in the future we can see sites offering a couple of .., just like when you create a GitHub account you can pick your license for the new GitHub repo. I can see a world where sites are offering different terms and conditions and the user can select which one best suits their needs for giving the data to the site. And if none then they don't give the data.

the site agrees, returns the signed agreement, and then the user saves the receipt and grants the full credential. And that's what the flow that we showed in the demo. This is just further detailing. The credential offer comes back or I'm sorry, the site comes back and says, "We'll accept an offer, you make the offer, you get to see what terms and conditions are there before you agree to it."

And then the signed agreement comes back from the server. We use our plugin to validate it. But again, at any point if you don't have our plugin, you want to do it yourself, you just download the signed agreement, you send it through KERIpy, if you want to or any tool that supports this flow with KERI and you can then upload your offer or your full grant, sorry, you upload your full grant, which we do with the submit full credential and now the site gets your information. And when we did this, it really did hit home that the user benefits greatly for all the reasons I just talked about, but the site benefits from it is something I think George said in one of his talks.

They get cryptographically verifiable data. They're not going to get typos. That's going to halt that's going to slow down the process of users signing onto your site and getting access to the data that they need. Okay, this is how we did it. We built an ACDC authorization server. It acts as both an OAuth 2.0 authorization service as well as an OIDC Identity Provider.

We're doing it with ACDC schema configuration. You can configure both the client registration schema that you'll accept and it can be a list and you also specify the user authentication schema that you'll accept and it can be a list.

Each of those supports a base schema specification which allows you to enforce what issuers you will accept at any chain in your credential hierarchy. For example, an oversight we did with GLEIF. I can issue vLEI credentials. They're not valid because they don't come from GLEIF root, but you can enforce that it has to come from GLEIF Root in here by putting the issuer of the GLEIF Root AID.

This was a fun part. ACDC payload to OIDC claims mapping. We used a pretty simple template syntax (Jinja2 for those of you who are familiar with Python) to create complex field mappings between the credential payload and the OIDC claims that come out the other end.

When you put all this together, what you're really doing is you're creating a truly distributed identity provider appliance. I can put one of these up, specify a list of credentials, I'll accept, specify how I want the data to look in OIDC.

Someone will submit their credential to me and I'll let him in. Sam can do the same thing. We never have to talk. Our systems aren't federated. We don't share data. If he specifies the same schema as me, he'll let the person in the same. It looks exactly the same to the user. The data comes back to the client application, of who that user is. But there's no federation. You're not relying on Google to hold your Gmail account. You're doing it with a decentralized credential issuance from anyone.

Then these are the flows I've already talked about that we worked with, dynamic client registration, secret lists, the client authentication flow, this is for our payer-to-payer model so that the payers can then just start as soon as they do the DCR the payers can start exchanging data by getting the token and then user authentication. The last part. I talked about integration into the watcher network and the witness network. For those of you who know me, I do actually own a Great Dane and a Bull Mastiff, but these aren't my dogs.

We called it HK Tracker and it's what Sam has been calling forever a local watcher. Anywhere where we deploy a server or a system that needs access to the KERI watcher network, it's going to be resolving AIDs and it's going to need to keep track of key state. Any enterprise system we can put as a sidecar. Think of a sidecar pod inside of Kubernetes or just running along in a VM. We put a local watcher service which is configured to our SaaS platform to be able to ask our platform to start watching new AIDs. The situation where someone authenticates for the first time and then continue to keep track of their key state changes.

It adjudicates the key state changes locally and integrates with the KERI watcher network. It uses our ESSR secure connections to our APIs to get the updates for the AIDs that it cares about. And the cool thing is it doesn't have to do it for any ids that have registered with Sam's server because I don't care about those only the ones that I have authenticated against. And so we use, borrowed very heavily from Tailscale if any of you are familiar with that VPN company, t hey have a model for when you want to get a headless server on the system you go into the site and you get a authentication code onetime authentication code that you use. And so that's how we do it down there.

These are my actual dogs. I just wanted to show this. That's Zoe, my Great Dean. That's Jackson, our Bull Mastiff. And that's Remy rescue Pit Bull. This is the diagram that we've created. I think Jared already showed this in one of his talks, but this is the diagram we created when we were first trying to conceptualize this.

And the idea here is that when you look at the convergence of what's happening here, DirectTrust is working very closely with GLEIF to help facilitate bringing the vLEI ecosystem into healthcare. You can have these credentials issued into a payer and a provider wallet, dynamic client registration, client

authentication.

They can issue vLEIs down to their members. State of Utah can issue SEDI credentials and then all of them can now authenticate against the site and get the data they need using this exact same method. That's what we're trying to get at here.

And that's it. Any questions? Yes. Do we have a microphone? – Just a quick question. Can you say more about the difference between the payer and the provider wallets? Like what they're meant for. Oh, like the purpose of use, like why a payer would be accessing another payer versus why they'd be accessing a provider. Is that what you mean?

– And what the wallets do; what's their provider mean? In this case this was just meant to represent the fact that you have this the same type of wallet. – Got it. The payer would be holding his vLEI credential and use it to access anyone else in the ecosystem. Literally anyone else in the ecosystem access their data.

– Got it. Those are just roles, not different wallets. Okay. Got it. I misunderstood. – My question is about the authorization server flow. [adjusting volume] The authorization server waits for an ACDC presentation. Does that make the ACDC essentially the identity provider?

Yeah. That's why I call it a dynamic identity provider appliance because it's just you install anyone anywhere and you say trust these credentials. – Okay. And it just works. – Cool. – Is it limited to person to business or can also be used in API situations business to business?

The client authentication flow, dynamic client registration and then client authentication flow supports business to business. In OAuth you can do just client authentication where you skip the authentication flow you get you don't get the authentication code that's just payer to payer that's business to business so it does support that as well.

That's why I guess you have a client ID because the client can get the data themselves not on behalf of a user. All right, thank you very much. [Applause] – Thank you.

TALK XXX

Infrastructure for Security

Phil Feairheller

KERI Conference April 2026 · talks · 36:59

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Phil Feairheller maps KERI's security stack from CESR and Key Event Logs through delegation and multisig, then demos Locksmith alongside open-source witnesses and watchers donated to the KERI Foundation. He explains wallet-held identifiers, thresholds, and remote or group AIDs as deployable infrastructure for healthcare trust.

Hi everyone. Thanks for showing up today. My name, for those of you who don't know me, is Phil Feairheller and the CTO and co-founder of healthKERI. We are specializing in bringing KERI technology, as the name would suggest, into healthcare, in particular data exchange in healthcare to help facilitate trust across the healthcare ecosystem. I realized I forgot a intro slide, so I'm just doing it without a slide.

In my job, I end up teaching KERI very often to a lot of different people, in particular people who have never even heard of it before. Sometimes maybe they've heard of PKI, sometimes they've heard of X.509s, but they've never heard of KERI. And this is one of my favorite slides to teach KERI because it starts at the very bottom of the layers that make KERI the security protocol that it is. And when Sam first asked me to do this talk on KERI infrastructure, I immediately thought of this slide because the infrastructure that I'm going to talk about is very clearly spelled out in the layers of security that make up the KERI stack. I won't do the whole spiel that I usually do for this, but very quickly, starting at the bottom of the self-certifying identifiers and CESR, for folks that who know PKI, I always like to say that KERI solves the two hard problems of PKI. One is creating a cryptographic binding between the identifier and the first set of keys, and the second hard problem it solves is key rotation, creating the cryptographic binding to every other set of keys. And we are able to do that with CESR, which is our encoding format for exchanging cryptographic primitives at scale.

The next layer up is pre-rotation Key Event Logs. Already touched on what pre-rotation is, and the Key Event Logs are of course the data structure, append-only data structure that allows us to communicate changes in our key state that anyone else in the world can verify without ever having to phone home. Layer three delegation distributed group multisig, that is something that we only talk to I only talk to when I'm dealing with more advanced audiences, talking about KERI's ability to do delegation both at the identifier level as well as attenuated delegation with ACDC credentials and distributed group multisig is one of the more fun technologies in KERI. I love working with it. I love building it when we were at GLEIF, but we don't really dig too much into it. Then if you think about what I'm going to talk about today, the wallets and then the watchers and the witnesses, the first three layers, one through three that I just talked about are all in the wallet. I'll be able to demo most of these, actually all of these capabilities today with the wallet. It's our wallet called Locksmith. And then when you get to layer four, witnesses and watchers, those are the other components that I'll talk about today that we've donated it to the KERI Foundation for open source.

And those are of course the layer that allows us to make our Key Event Logs available to other people and keep an eye on anyone else we're doing business with, keep an eye on their key state. The Locksmith wallet is an application that we've built over time.

It actually started as a pet project of mine because I was tired of using the command line. Most of the work that we did at GLEIF was done building out the command line and then actually creating the entire GLEIF-authorized representative system with the KERI command line. And at one point I just got tired of using the command line over and over again particularly when trying to demo KERI to people. I just started playing around with what it would look like to create a wallet and took the KERI command line from KERIpy and just started building little features one at a time into this and eventually it became a tool that I was using all the time and we figured it would be very useful and started making it a production quality wallet.

The core of it is open source and what I mean by core and I'm going to be demoing it here in a minute, you'll see the core is basically all the KERI functionality. Just core KERI functionality for working with your key state locally, creating what we call vaults which if you know KERIpy they're called Haberies. (Sorry Sam, couldn't call them Haberies in my product) Creating identifiers inside of those vaults, doing delegation, multisig, Provisioning witnesses is actually on our SaaS platform, but working with witnesses, doing key rotations, etc. We also have sections for all the work you do with a credential, so uploading schema, issuing credentials, granting credentials, doing the whole IPEX exchange, as well as receiving credentials and using them in various ways. There's no server required for the core Locksmith wallet. You can create all of your stuff locally for everything that you would need to potentially share with someone, we have an export feature. If you create an identifier, you do a couple key rotations, you want to share it with someone, you can just export the CESR of the Key Event Log and email to someone.

And then finally, it's extensible by design. We'll get into that quite a bit. When we first started adding features for our SaaS platform, and the healthKERI SaaS platform is really just KERI infrastructure. We're allowing folks to provision witnesses and watchers and use them with their identifiers for doing business with each other, we realized actually it was just because of where we put it on the user interface, it looked like a plugin. And we thought, well what? If we do really want to donate this to open source, we're not going to donate that part of the product to open source, so why not wrap the interaction with our SaaS services into a plugin API and make it so that anyone can provide plugins to that system. I believe later today or maybe tomorrow we'll see some of the gentlemen from the KERI Foundation who have taken this open source core and integrated their own plugin to deal with their own SaaS services. And Two presentations this afternoon. Great. On just that?

Excellent. What we've also done with it over time is part of the business that we're getting into, you heard Jared talk about today, we're starting to bring the vLEI into healthcare. We're helping folks with the issuance of credentials, particularly vLEI credentials. We've added a new component underneath our plugin for our SaaS platform that is a credential issuance management protocol or plug-in. That will interact with a server component, really wrapping all of the hardest part of of issuing the vLEIs is managing the multisig identifiers required and the coordination between participants and how they see the credentials and how they interact with each other, notifications, etc. when issuing credentials across an enterprise by various parties who may not be sitting in the same room all the time.

(Let's see where to go.) So the KERI core already talked about it a little bit and I'm honestly not really sure where to go to the demo, so I may just tell on the slides and go right over soon, but this is just all the basic stuff you would do with a KERI system. Create your identifiers, manage them, do rotations, like I said, export them, share them with other folks.

We've broken out group identifiers into its own section. We initially experimented with having it as just another type of identifier you create and that was a little bit unwieldy, so we have our own separate group identifier section. The ability to do delegation with any identifier that you create, you can send

challenge responses. I already mentioned the credential schemas and then working with witnesses and watchers, as a matter of fact. And then kind of tangential to what you get with a normal KERI wallet, but much needed is a full notification system integrated with KERI messages that go back and forth. If you receive KERI messages like IPEX messages for a grant that you were given, they turn into notifications inside of the user interface, kind of separate from the actual IXN message itself. (What's next?)

The plug-in arcitecture is the part .., Down at the bottom here, this is the healthKeri plug-in and this is what you would access to be able to reserve resources on our SaaS platform in much the same way that you would go reserve a VM from Digital Ocean or from AWS. We tried to keep that comfortable for people who are used to dealing with cloud services to say, "Oh, I need another witness. I can just go rent him, if you will."

When you go into that Plug-in, you will see the ability to access identifiers, witnesses. We give you the opportunity to publish credentials, we're going to do a lot more with that in the future once the KERI Foundation starts working on observers and registrars. This is just kind of a lightweight version of that. We have a connections feature which steps kind of outside of OOBIs if you just want to do quick connections with each other the other person is using Locksmith at our SaaS platform. And then watched identifiers is our watcher network. And I'll talk more about that later.

Mailboxes so that you can get these messages sent back and forth to each other. And then finally servers and profiles I'll touch on when I do the demo. Okay, we're going to stop there and I'm going to switch over To the demo. This is Locksmith.

Again, I have two of them running here. (I don't know how awkward that is to see on the screen. Maybe I'll just center them both.) (With the overlap there, that might be a little bit awkward.) I have two Locksmiths here so that I can demo some of the features of going back and forth between them.

As I mentioned, we call [them vaults, red.] Your database environments which you can separate out groups of identifiers within a KERI system, we call them vaults. We have the ability to create a new vault. You can give it a name. I'm going to go with Alice.

And create a vault. One of the things we focused on here is making it very simple to do very simple things. When you're creating an identifier, there's a lot of things you can do and it can be very complicated and complex to understand what all the knobs and buttons do on creating an identifier.

Same with creating a vault here. We tried to make it so that in the very simple case, if you know what you're doing, you just need to do something quickly, type in a name and go and you're done. That's what we did with creating in a vault you could creating a vault. You can do two-factor auth with it if you want. Not really that meaningful or important, but you can cuz someone requested it from us. Now I'm going to open the vault.

And here I am in a KERI environment. Over here on the left-hand side, I have my menu which gives me the ability to work with all the items I talked about before. Identifiers are my local identifiers. Remote identifiers are those with other people who I'm doing business with. Group multisigs, credentials, proxies, and the settings. I'll just touch on the settings very quickly inside of here. These for those of you who have worked very closely with KERI under the hoods, these are various knobs and buttons you can turn to configure your environment whether you want a temporary data store where if you want to change the directory where we store things by default, etc. We have a plugin that I'll be demoing

tomorrow that interacts directly with Locksmith for gaining access to getting things signed for you Submitting IPEX grants so that you can log onto a website, that sort of thing. You can configure that here as well.

Going back to identifiers. As I said, we try and make things very simple. I'm Alice and I want to create an identifier. I just did. Now I have a KERI identifier. It's got a Key Event Log, only an inception event. And if I look at it, you can see that I created it. It has a public key.

No witnesses or anything like that yet. And there's my Key Event Log. That was really easy. And that's one of the things that we really want to emphasize here is how quickly we can do things. Now, if we want to get more complicated with it, you can go into our advanced configuration and you have all of these options available to you. You can we can do a keychain if you want to do a hierarchical deterministic keychain. You can create that here. You can have a salt which you can look at, copy and paste, and save in your one password so that you could recover this identifier and all of its keys in the in the case of a catastrophic event.

If you're a little more concerned about security, you can just do a random key so that you could never get it back if you lose your keys. Underneath that, we have our signing thresholds. You have the number of signing keys in the signing threshold.

This is only the M of N support. We haven't built in the support for fractionally weighted thresholds yet. We will eventually. Some of the other features when you create an identifier, you can go with establishment only or do not delegate. And then if you choose delegation, you can delegate from some other identifier, whether it's a local or a remote identifier. I'm not going to create any of that right now.

We have our remote identifiers. (That's actually a problem. I'll come back to that in a minute.) And then in our group identifier section, if you wanted to go ahead and create a group identifier, you give it a name, you select your local identifier that you want to use to join that group, and then you would select your other participants.

And again, a group identifier is exactly like any other identifier in KERI. It's just that the keys are shared across different machines. All the other options for specifying your thresholds, whether it's establishment only or do not delegate, all apply in here as well.

– Quick question. (Yes) Is there a discovery built into that drop down for adding group? Yes, so the other participants would be coming from your remote identifiers here. Any identifiers that you OOBI with and you've connected with become candidates for joining a group multisig. You already have to know about their Key Event Log, and that's in that's because of the protocol used to create a group multisig. It involves taking keys from individual identifiers from all the participants involved. And only the public keys, we don't share the private keys. That's where they came from.

– Is that a different concept than like a connection, a more general connection, or is it specifically scoped to Multisig group identifier? When you say that, are you referring to remote identifiers? Yeah. Remote identifiers are just ones you OOBI's with. In KERIpy, I think they're called 'contacts' in KERIpy, that's what the remote identifiers are here.

It could be anything. Issuing a credential to receiving a credential from, anything at all. (real quick, I realized I forgot to do something, so you're going to see a little bit of behind the scenes here.) (Because I realized I don't have…) – [Inaudible] What do you mean?

We're going to restart my two Locksmiths. (And I knew I was going to make that mistake.) We should be good to go now. I'll re-center and size them a little more. We were in the middle of creating Alice, so we will open up her vault again, and we'll go over to remote identifiers. Good, there they are. Those are the two identifiers for our SaaS platform. They are configured to load when I And these are configuration options you can add when you package and ship your version of Locksmith. Those are OOBIs that he will automatically resolve when he starts up, so that he has access to them here. For in our particular case, verifying the signatures against our SaaS platform for gaining access.

Let's go over to the healthKERI, SaaS platform. This is our 'getting started' screen, creating an account within healthKERI. And the reason why I'm doing this, it won't take very long, but this gives me the ability to provision witnesses, so we can talk about the witnesses that we have, and then also to start watching identifiers inside of the healthKERI watcher network so we can see the interplay there. And then I can talk more about what we what's going to be available for all of you folks to start working with. I am in Alice, so I will very quickly create Alice and this is one of the key differentiators of what we do with our SaaS platform. There are no usernames and passwords. There are only usernames and KERI identifiers. You aren't uploading a password here, you're selecting a local KERI identifier .., (Hold on.)

(Create a new one.) I'm going to select my healthKERI identifier to use and that will be the identifier that the public key and Key Event Log of that identifier will be stored on our services and anything that I send to the SaaS platform API will be signed and encrypted and anything that it responds with, will also be signed and encrypted from their AID.

We have all of our communications between our local Locksmith plugin and our SaaS platform secured with what we call ESSR which is 'encrypt send sign receiver' which is a protocol that's built into KERIpy. I'm going to continue into the account creation. This is Alice.

She's on the internet. 'Alice at healthkeri.com' Create an account. Now when we first created our SaaS platform, we based everything off of an individual user and you would provision your own witnesses and watchers and that immediately was revealed to be very much too limiting because people would want to share accounts.

We switched over to a model where we have teams and if any of you are familiar with using Digital Ocean, it's very much modeled off of their approach for how you share resources across teams. And this came in handy later when we started working with provisioning services inside of enterprises where you have maybe a cloud service that needs an identifier and it also needs things like witnesses and watchers, how you bring that server into your platform so he can start doing KERI things and reserving resources from your KERI account. This is team 'Alice .. [@] healthkeri.com' and we'll continue to access code and as of right now we don't have billing built into our system so we just base it off access codes. I hand out access codes when people want to get an account on our system.

I've created a couple for this demo. – There's no vLEI integration where you're collecting their role? No, not in this. No, we don't require a vLEI to gain access to our to our platform. Okay, I've gone ahead gone ahead and created an account Inside of healthKERI. I've added a team to that account.

And now I'm going to add an identifier. And this is a little bit of an awkward step but one that we were required to do for all of the other things you would want to do on our SaaS platform. And what I mean by that is when we provision witnesses, slightly different from what many of the other witness deployments are right now, when we provision witnesses, we provision them only for a single AID. Every witness that you get on our service will only ever witness for the AID you gave it.

And that's a security measure and also it helps us keep the scale of all the witnesses very low so that we can have a lot of witnesses because we know each one will only ever witness for one AID. In order to do that, we have to give it the AID to witness for.

That's an example of why we had to create a first step that when you want to start using our system, you have to give us an identifier to start working with. – Are there any special caveats for group identifiers? No. – They have separate witnesses?

Correct. Absolutely, yep. In addition to that, if I have, for example, maybe an identifier on a phone and I want witnesses for it, I can actually do that as well from here (and I'll show you how you integrate with that later).

When you first upload an account identifier, it can be a remote identifier that you OOBI'd with so you can get resources for some other identifier or a local one from this wallet. In addition, we added a feature of making this identifier publicly available and that's just the healthcare platform global OOBI system so that whenever you upload an identifier, we immediately create a new OOBI for you that you can hand out to anyone and it'll work for all roles within your system. And for those of you who know the details of KERI, if you OOBI against a specific role, you only get back the responses for that role like for a witness.

This just gives you everything back with one single OOBI. I'm going to upload Yep, got to select it first. I'm going to upload the Alice identifier. And you can see that now our SaaS platform knows about this identifier.

It's associated with our account. It has our sequence number. And broken out on this grid is both the local sequence number that I have on my machine and the sequence number that the remote system has for me. And those can get out of sync if I do a bunch of rotations and don't come back and update healthKERI except for when watchers are involved.

Now it's also public. From here I'm going to jump back real quick to the wallet side of this application, not the healthcare platform side, and talk about our credential management system. Inside of here, we have the ability to issue credentials. We broke credentials out into those that you have issued, those that you have received, and then accepted offers is something very particular to the demo I will be doing tomorrow.

And for those of you who were here at SEDI and saw me do it yesterday, it's where we have integrated IPEX protocol into our OAuth servers so that you can do a credential presentation of an offer without the actual credential itself and require the site to sign your terms and conditions before you give them the full credential. Those signed receipts, signed offers show up here.

And then finally we have the ability to do schema. We can upload schema from any source. I'm going to go ahead and grab the QVI schema just cuz I know it doesn't have any .., Doesn't have any chains on it. We put the OOBI in there and it automatically pulls out the SAID for you just so you see you have that tuple of the identifier as well as the OOBI. And then this is another use case where we looked at it and said, "Okay, credential registries make sense to me, " but that's cuz Sam has explained them to me about five times, maybe probably more than that.

It was really hard for us to explain to people what they're doing in that first step before you're allowed to issue credentials. You have to create a registry. What's that? What name do I give it? "Where does it live?" And so instead of doing that we tried to hide all that and this is open for debate. This will now be an open source.

If folks don't like this it can be undone, but we've decided that instead we've added a step to uploading a schema that says, "I want to use this for credential issuance." When you select that, you select an identifier who's going to be the issuer for that particular type of credential. And when you load the schema, not only does it load the schema, but it has now created a credential registry behind the scenes.

That is completely hidden from the user and we think that's probably a good thing, but again open for debate and we're willing to accept feedback on that. Did you have a question? Did you issue a credential registry per schema type?

Yep. It was just a an allowance that we made to try and make it a little bit easier for the user. When you come into our issue credential screen, (I'm not going to issue any credentials but I just want to show you really quickly), you select the type that you want. We dig into the schema and present fields for you. These are as typed as we can make them based on the JSON schema in each of the credentials cuz you cannot specify formats and stuff in schema if you don't want to.

Calling out required fields and stuff like that just to make credential issuance a little bit easier for the user without hardcoding into this app knowledge of any specific schema type. Okay, now we're going to go back to the healthKERI plugin.

And there we go. Perfect example. This this identifier is now out of sync and that is because I created a credential registry with it which had to create an anchor into my Key Event Log and now he's out of sync, so you can see that the remote server doesn't have my most up-to-date key state.

I'm just going to go ahead and update that. This tells me that I'm going to be taking the local key state where I'm at 1 and pushing it up to the remote server who is at key state 0 for this and updating now we're in sync. He now knows about my most recent key events.

Now we're going to move on to witnesses and this is .., (Does it make sense to go back to the slideshow?) I'll just go over this slide very quickly. This is witness as a service and as I mentioned earlier this gives you the ability to reserve individual witnesses for your AID. Any number of witnesses you want. We have geographically disperse witness infrastructure so you get to pick where you want your witnesses deployed, how many you want deployed and then you get to pick your threshold that you want of those witnesses and our user interface will default to the most logical threshold based on the What's the name of that algorithm?

The Kawa algorithm. The number of witnesses that you should have in your threshold versus the number of witnesses that you have. As I said reserve a witness as easily as any cloud resource. The core of this is open source. I know we're going to see demos of the KERI Foundation folks who have already put it into production in their systems.

it is a multi-tenant system, so you can put it on one VM and reserve as many witnesses as you think that VM can support. And then what we've done behind the scenes for us is that we spread that out across multiple hosts and we have a round-robin, and I'll talk about how we deploy all this later, that we can then pick from for any particular region.

We have added, actually I added, to KERI core, at the very beginning of my work at healthKeri, the ability for a two-factor auth for witnesses. We have a one-time passcode that you select that you authenticate with the witness.

He gives you back the one-time passcode, then you have to enter that anytime you do a key rotation. And that's to make sure that your witnesses won't just accept the Key Event Log for you from anyone. You have to provide that one-time password with it. There are other ways to do this. We could use our our proprietary API with the ESSR to accept events for your witnesses, but we thought that was more vendor lock-in than we really were comfortable with, so we went with this one-time password solution. They are non-promiscuous. They will only witness for the people for the one AID that you reserve them for. Each witness has its own dedicated key store, so there's no mixing of keys in like a MongoDB or anything. They each have their own private LMDB and we manage that across multiple servers. This makes it fairly scalable, pretty easy to deploy, and designed for and in our case implemented with geographic distribution.

(Let's just see the screen. I don't need to talk through this.) Here we are. Let's add a witness. When you open up the create a witness screen, it will of course make you select an identifier, so I'm going to select Alice. In our system right now we have four regions deployed. We have New York, Frankfurt, Singapore, and Sydney. And you get to choose where you want your witnesses deployed. I've already chose I've already selected Alice as the identifier that I want witnesses for. I can add as many as I want in each region. It gives you a suggestion of a name for that witness in that region. You can name them anything you want.

If you want to, I'm just going to go ahead and reserve a witness in each of the regions. And then I'm going to choose an authentication method. Trade-off here between security and friction, ease of use. We've given you the ability to have an individual one-time passcode for each witness that you reserve. At four, that's really onerous, so I wouldn't even do it at four. At 12, I couldn't even imagine it. We also give you the opportunity to do a batch one-time passcode. One-time passcode for all of your witnesses for this AID, and that way you can only enter it once, only enter one code once when you do a full rotation.

We're going to select batch, create our witnesses, and you can see here I've now spun up four witnesses. Pretty fast. This is actually witnesses spinning up in our SaaS platform. It's very light touch for getting a single witness to spin up.

Each one has its own identifier. You can see them here. They're non-transferable identifiers, as you would imagine. Here are the OOBIs right here that you can copy and paste if you wanted to. You can also toggle on QR codes for those OOBIs. If you're doing this for a device that's on a mobile phone, and you need to scan in these so that you can now gain access to them for that device, you can do that here with these, QR codes. And then finally, we provide the opportunity for you to right here grab those OOBIs out of this screen and just introduce to all of them. I'm going to go ahead and do this. I'm now OOBI'ing with each witness individually and saving him into my database. And then we did the authentication step, where I called into the witness, and this was really tricky to get right because this like sort of a trust a TOFU (trust on first use) case, where that witness wasn't authenticated anybody, but he would only accept things from me. I was able to sign one request to him to get the one-time passcode back from him, and it was encrypted between the two of us. Now I have the one-time passcode here. I can, again, I could scan that into the one-time passcode feature on my phone or in this case I'm going to copy it and put it in one password so I can actually do things with these witnesses afterwards.

I don't know if any of you knew this, but What's that? – We need your one password? No, I use my fingerprint. You don't have that again. – I just wanted you to type it out for us on screen Nope. I don't know if you knew this, but when I discovered this like wow, that's cool. They have a one-time password

support.

Absolutely. They have OP command line, very cool. Two words or two letters, I love that. Then finally, the last step would have be of course rotation this in. I could just go do this manually, but we've added the ability to do a rotation right here on the screen. It brings up common defaults that you'd use. You want your new your next keys or signing thresholds. We select the TOAD, the TOAD = threshold-of-accountable-duplicity accountable duplicity of three, which makes sense for four witnesses.

And these are the four witnesses we just created. You can add more or remove here if you wanted to, but we'll just go ahead and rotate them in. And of course now it's asking for my one-time passcode so I'm going to go over here.

(I'm very paranoid cuz I'm going to wait till it rolls over to the next 30 seconds.) 1:30 There we go. Now I'm going to grab that. Paste that in here and do a rotation. And we've rotated. Now I have all four witnesses rotated into my identifier. You can see over here on this side. I'm now at sequence number 2. I was at sequence number 1.

All four witnesses gave me receipts. There's my threshold. There's my new public key. And because this is a remote identifier and we know that we're given the opportunity to just go ahead and upload the new event that you just used to rotate in those witnesses into your upstream server. We'll update that. I'm now done with this. You see here I've created these four witnesses. These are my account witnesses. You're now paying for them.

They're a monthly service, but very very cheap. The controller is Alice. This is each of the AIDs for those witnesses, and these are the regions that they're on. I now have witnesses spread out around the world for this identifier. Someone would immediately now have to attack four different places around the world to eclipse [attack, red.] my witnesses. And that was pretty easy, right? I think it was pretty easy.

Thank you. And then if we come over here, you can see that we have updated to our new key state to this remote server To our remote SaaS platform, and we're now at sequence number 2. If you click on this, you can see this is the public OOBI I was talking about cuz I said "Hey, make this guy public." I 5 minutes left?

Oh, man, that went fast. I'm not even close to done. All right, well all right, let's see if I can do this really quick. All right, so let's get to watchers really fast. And to do that, I'm going to create a whole new account. Let's create Bob.

Create. Open. Add identifier 'Bob'. Now you see how fast this can actually work, right? All right. I think I copied. This is where you get to see remote identifiers. I grabbed Alice's OOBI, so I'm just going to paste her OOBI in here and connect, and now I have Alice. Now I'm connected to Alice. I'll go over here. Oh, no, I got to create an account ' Bob' I 'boob'-ed, oh, jeez, sorry Bob!

(That was a zero. It wasn't a second O) Continue to account access ' Bob' Wonderful. – Want to use the healthKERI identifier for that? [Meanwhile, Phil is typing and speaking out loud] These are one-time access codes, by the way, if I didn't mention that before.

We're going to grab Bob, paste. Paste, continue to invites, no invites. Go to identifiers. Let's add Bob an identifier. Although, I don't think I need to do this, but anyway, you can just see how quick this is. And now, let's go to the last step, watcher network. Very quickly, I want to mention that we initially implemented our watcher network the same way we did witnesses. You have to rent individual watchers and assign them to watch people. If I wanted to have like five watchers watching Sam's AID, I'd have to

go and get five watchers and I'll say watch Sam's ID. That way, it was it didn't work well. I talked with Sam about it. He said it's no more secure giving them VMs inside of our infrastructure than if I just created a global watcher network. That's what we did. We created a global watcher network. Instead of having to worry about renting your own watchers and assigning them and managing them, we just tell you tell us an identifier you want to watch.

We'll distribute it across our global watcher network for you at random in different locations, and we'll keep an eye on it for you. That's what we've done. Getting a watched identifier is as simple as selecting one of your remote identifiers and say watch. He immediately pulled the key state for that watcher down, and now telling our system to go watch that identifier. The value of that, of course, is that now I can come over here to Alice, and in the regular KERI side of the plugin (lock that) (Hm? 2 minutes? I did that pretty quick then!)

I'm going to rotate Alice's identifier, do a key rotation. (I got to get my a one-time passcode.) Grab that. 15 seconds should be plenty. Rotate. There, I've now rotated. I have all four witnesses Pulled down for him. And if everything works correctly, Bob will now get a notice that Alice has done an update. This takes a lot longer than I wanted it to doing a demo, but that's because I have all my numbers kind of spun down. There it is, it just showed up. Just so I don't burn my CPU out. I just received a notification.

This is again the integration of notifications into our system. The notification is available in the open source plug-in, and you can integrate it anything you want. Whatever you add to the health KERI if as a plug-in, you can integrate into the notification systems to drive that as well. Now that I see that I have new key state for Alice, we made the decision not to automatically pull that key state down for you. I would just go into my watched identifiers. You can see that he is now Alice is now out of date, and I want to do a sync, and now I'm up to date with Alice.

One last thing I'll point out is that's all well and good, but I'm doing it manually. Where this really comes into play is when you're deploying servers into enterprise infrastructure, and you have a server that has an identifier and that is let's say for example an OAuth server that's validating vLEIs coming in, but needs to keep up to date with the key state of any of the identifiers that it is now interacted with. You can provision it so that it can connect through this authorization server feature inside of our SaaS platform, so that it now has the authority to ask our platform to watch identifiers on its behalf, and it will and I'll talk a lot more about this tomorrow. It will then have a side car running that is its local watcher that integrates with the healthKERI we watcher platform to keep up to date on key state changes so that anyone that you're using towards anyone that is using you to authenticate into a system can do a key rotation anytime they want and automatically get updates.

I just real quick I'll go to questions I guess. [applause] Thanks everyone. – How diversified is the watcher set that you're running? Right now we're in four regions, and the same regions that we are in the witness in the witness network that are on different data centers than the witnesses are.

. We have I think three servers in each of the regions. And we just pick them at random when you ask us to watch. We do four watchers. Pick them at random, one in each region. And that's easy to extend. (I didn't even get to how we're deploying the services.)

No, it was only 40 minutes. I know. I realized that when you told me I had 5 minutes left. All right, great. Thanks, everyone. Dismissed. [applause]

TALK 31

Digital Identity Needs a Legal Foundation

Phil Windley

KERI Conference April 2026 · talks · 40:07

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Phil Windley argues adoption needs legal and institutional foundations, not only cryptography. He discusses Sankarshan Mukhopadhyay's "proof gap," where people depend on centralized institutions to verify attributes, and connects Utah's SEDI model—state-proofed credentials on KERI identifiers—to scalable verifiable identity beyond paper filing cabinets.

Okay, I'm going to start since we're over time already. I'm glad to be here today. Title of my talk, as you can see, is "Digital Identity Needs a Legal Foundation". If you were at SEDI summit with the first two days of the week, this is related to the things I said there and just kind of goes into a little bit more detail on how the legal foundation actually helps us out. We spent a lot of time talking about the cryptography and key management and all of those things.

But I think the hardest part of digital identity is actually not the technical things but legal and institutional. I spent a lot of time talking about governance and sometimes it's can be kind of boring but ultimately I think it's what gives us the foundation for making all of this work. Think about where we're at. We've got decentralized key management, authentic chained credentials, Serialization formats, some implementations, open- source code.

The technical foundations are largely there. That doesn't mean they're done. Doesn't mean they can't be improved, but we've made real progress. We also have things that we're still working on like tooling and ecosystem and developer experience and all of those things that make it work.

But we need to get adoption. And in order to get adoption, we have to get rid of some of the fundamental blockers. Now, I'm going to talk about a couple of them today. Not necessarily all of them. I mean because there's no end to blockers when you're trying to do something this big. But there's a couple of them that I think we can help solve. Sankarshan How to say his name Mukhopadhyay I think is how you say his last name. I haven't heard it said so I'll say it that way.

Recently wrote a book about what he called the a book a paper what he called the proof gap. And for him, the proof https://thetrustgraph.substack.com/p/the-proof-gap gap was the idea that we still rely largely on centralized institutions to be able to prove things about ourselves.

Our education, employment, whatever licenses we have benefits we're entitled to, all of those things. When proof is needed, we can't generally present it directly, especially digitally in a way that somebody else can easily verify. Even worse, I If we lose that proof, we have to essentially go back to those institutions one at a time and confirm something about ourselves, right?

That could be a real problem. Back 50 years ago when everything was based on paper and kept in filing cabinets, that might have made sense because it was really all you could do. But I think everybody in this room acknowledges that we can do better than that in a digital world and we should do better than that, not just take that old paper filing cabinet model and move it forward.

Sometimes people think of the proof gap as a technical problem and it certainly is that. There are technical problems with moving this centralized filing cabinet model into the digital world and trying to decentralize it and create something that's better.

But I think we can ask ourselves about the problem of governance. There are lots of ways that we have worked out how to transfer trust in the physical world, right? One of those is very familiar to everybody and that is credit cards.

60 years ago, we didn't have this problem because we didn't have credit cards really at all, right? There were bank cards and maybe had a card from your department store, right? I remember people had Sears cards that you could use at Sears and nowhere else, right? They were all centralized, single domain and Visa first and then others followed came up with this idea that we could have a coalition of banks put together ways that they can work together. This involved technology, how are we going to technically transfer information? It involved rules and processes. It involved legal agreements. Involved all of this. And collectively we call that thing a trust framework, right? All of those technical and legal and regulatory rules that make all of that work. But even today if you're a merchant, you've got multiple ones of these that you have to interact with. Now, it turns out somebody saw that problem and there are all kinds of companies. There's actually between this guy and those guys, there's like four layers of companies that make this all work. In order to make it easy for him and for you when you go in, right? He doesn't have to have six different terminals and all of that because somebody's solved that problem. But still, it's a problem.

Now, imagine what happens when we've got not five different banking system or credit card systems. We have thousands of different credentials. Now we're trying to navigate the sea of islands where each of these is a separate trust framework with their own rules, their own governance. Maybe they share some technical underpinnings at least we could hope for that. But it's still a problem. If every category of credentials requires its own private trust framework and every verifier has to somehow be inside each of those trust frameworks, it's a nightmare, right? Instead of four layers between the merchant and the credit card companies, we'll have a hundred layers between you and all of the different things that you might need to verify in your life. Now, private governance can solve some of these problems, right? And we've seen lots of efforts over the years. IRA is one of the most recent that tries to create a coalition of related or like-minded maybe organizations who can come together and create a trust framework that is interoperable and I think they all kind of have this vision of we'll just get big enough that we'll solve this problem. It won't be a problem because we'll just get big enough that everybody will be part of us but Visa is not part of Mastercard and they never will be.

And that's always going to be a problem. A different way to solve this problem is to have something underneath all and connect them. Okay, but that's just one problem. There's another problem. I talked about this in my talk on Monday and that is an economic problem.

In the mid 1990s, Netscape had essentially solved the digital identity problem for at least the web. They had pretty good ideas, right? We'll have servers that will have certificates and we'll put certificates in browsers and your browser certificate will identify you. We'll proof your identity with these certificates. In this day and age where you can essentially get server certificates for free, you may not remember, but back in the day, certificates were really expensive because they were all proofed.

Right now, they basically just link a domain name to a key. That's all they do. But if you want a real proofed certificate, they cost money. Not only because it costs money to proof you, but because the company is putting themselves out there and saying “this information is correct”. They're liable. They cost money.

Turns out, people were not willing to pay money to shop and to bank and all of those things. They wanted it all to be free. We standardized on "Servers will have certificates and people will have passwords" and just kind of waved our hands and hoped the problem would go away because there was money to be made.

Here's the two problems. How we scale trust. How do we create trust frameworks that are more universal? And second, who's going to pay for this. Who's proofing the person using what authority, what's the legal standing, all of those questions.

This is where I think SEDI can help us with both of these problems and I'll talk about them in some detail. As far as scaling trust, SEDI doesn't provide any kind of universal trust framework. That's not the answer. What SEDI provides instead is a foundation that trust frameworks can sit on. We have something that is firm underneath, a firm legal foundation for all of these other things that are going to happen. The second help that SEDI can provide is on the economic problem because turns out that the state already has lots of infrastructure for doing this. They already proof you, part of if you look at your driver's license, we have a couple of people from Europe here so they may not be familiar with this idea but if you look at your driver's license which is the only thing we have to prove who we are, there's a little star on it. You know what that star means?

“REAL ID”. It means the state proofed your identity. This was started in 2005 when… earlier than that, maybe 2002… It was right after 911. It happened before 911, because I remember I was CIO when they started talking about this.

This was early 2000s, right? Proofing your identity. And at the time, the motor vehicle division, the driver's license division said, "We don't want anything to do with identity. We're not an identity provider. This is not us." Let's see how that worked out for them.

Anyway, you have the old star. They already proofed your identity. Let's talk for a minute. I think most people at this point probably have an idea of what SEDI is, but let's just make sure that we're all on the same page. SEDI doesn't give you an identity. In fact, the SEDI legislation specifically says: "An individual possesses an identity innate to the individual's existence and independent of the state. Which identity is fundamental and inalienable." SEDI doesn't give you an identity. They're not an identity provider, right? That's a nice term. Instead, what it is, you create your identifier or identifiers.

The state proofs specific attributes about you, then links those attributes to you in a cryptographic way, not to you, to this identifier (that you control). In a cryptographic way. You can kind of think of this like a notary, right? The state is endorsing your identity, not giving you an identity. They're just saying, "Yeah, this identifier, we looked at the person who controls it, and that really is the name and the birthday and the address and whatever other attributes might be part of your SEDI."

SEDI not just one credential. Like I said, you create an identifier (or identifiers), and then SEDI is a credential which links those attributes (that the state has proofed and endorsed) to that identifier. Now there could be other government credentials that sit on top of that.

Things like driver's licenses. We heard Joe talk about offhighway vehicle operator licenses, fishing licenses. States are veridible credential factories. They just love giving credentials to people for different things, right? States have lots of them. Now, some of these will involve guardianship, things like birth certificates and school records, and the statute includes that idea.

Others will be just things that you get. And, like I said, I've got six up here, but there probably a thousand of them that states have. Most of us don't care about most of them. But also, other institutions can give you credentials. You could use your SEDI to prove who you are when you go to the bank to get a credential from your bank. Or you could use your SEDI to prove who you are when you're registering for school and you would get a school ID. All of these other credentials are essentially resting on this foundation that SEDI has provided.

And you can even issue your own credentials. You may not imagine why that would be interesting now but I mean with personal agents, AI agents coming along this could be a lot more interesting. But it's not just people. There's also organizations. Now this is not SEDI. This is different than SEDI but it's related to SEDI and important for what we're talking about.

Organizations already have firm legal foundations in the sense that you can register a business and the commerce department gives you a registration number and the IRS gives you an EIN and all of these things that come along with being an organization. We don't care as much about the organizations creating and controlling their own identifiers, but they will, right?

State Department of Commerce generally in the United States at least is who is responsible for endorsing that so you could imagine a SEDI-like credential or it might just be their business registration but anyway it's analogous to the SEDI. Organizations like GLEIF are indispensable for businesses and other organizations who need some sort of global reach.

I got an LEI just because I wanted to see how it worked. I've never ever used it for anything, for my business. But that's important. Because there are some things like government credentials and banking credentials, business licenses that would just depend on whatever organizational identity you have, whatever identity or credential the organization has from the Department of Commerce, but many might require this global interaction.

Local restaurant might not ever need an LEI and all of the things that go with that, but a bank might. Now, let's talk for a second about the legal underpinnings of SEDI because this is important. I mean first of all SEDI includes a bill of rights. I read you the statement that it says about your identity belonging to you and being inalienable.

These are legal rights, not terms of service. This is not a click-through contract that you don't read. And, if you want to enforce whatever small rights they might have left you in this 60page document, you have to go to court. These are things that are in statute. They're in law. It also includes selective disclosure. The idea that you don't have to disclose everything in a credential or in your SEDI credential in order to use it. You can just disclose the parts that is necessary. Requires that the system be built on open standards.

That it's not a proprietary system MDL. No vendor lock-in. Anyone can build a conformant wallet verifier, whatever. Speaking of wallets, it mandates a choice of wallets, right? There's not just one app that the state gives you. And if you don't have that state app, you don't have a SEDI. You can choose different wallet providers. You can put your SEDI in any of those, move it between them, whatever. It includes a duty of loyalty which is like a fiduciary requirement for identity data which is a first. This is revolutionary in the world of digital identity.

And finally, like I mentioned earlier, it supports the idea, recognizes that many people are wards of a guardian, and therefore the guardian has to have the rights and the abilities to carry out functions, operations on their behalf. I don't know what the right word is. Carry out what?

What are they doing? Acting on their behalf. There we go. Acting. That's the word I was looking for. Nice simple word, not carry out duties. Okay. These are all part of the statute and important to why SEDI creates this kind of legal foundation.

Now, as I mentioned earlier, the problem in this proof gap is that when institutions make errors, you basically are left to navigate the correction process all on your own. Now, SEDI doesn't mandate that every institution has to do things a certain way, but it does provide you with a legal foundation. And it does say that you have enforcable rights related to your SEDI. You have the right to obtain SEDI credentials. You have the right to have its attributes amended as they change, or need correction. You have protections against arbitrary revocation. There are only certain circumstances where this can be revoked.

There's a complaint, an enforcement mechanism for how you can deal with problems that come up and that includes both the digital… what's it called? A digital privacy ombudsman and the attorney general. There are both of those that give you the ability to seek redress if something happens. … – Just real quick, but it does not include a private right of action. The attorney general has to act on your behalf.

— we tried to get that in there, but they decided that was reach too far. Reach too far. These aren't just like consent form click-through consent forms and support calls that you make. These are legal remedies that you have the right to … They are legal remedies you have the right to access if these things will happen.

Now… like I said earlier this doesn't mean SEDI doesn't replace all of these trust frameworks that exist. I spent my career largely in academia. Academics is going to have trust frameworks about what an accredited institution is for example.

Visa is still going to exist as a trust framework for how money gets transferred inside their network. Those are all still going to exist. But what does change is the idea that they can sit on top of this legal foundation. They can have attributes that they know, someone they trust, proofed.

Now along those lines, I want to talk about two different ways of establishing trust. This is a conversation I had with Sam, at Smoking Apple six weeks ago, or so, right? Because I had just written a blog post about reputation for agents. And how all that can work. But it's equally important here. There are two different ways of establishing trust.

One is by observation. You look at someone's behavior and you say, "Sam seems like a nice guy. I guess I'll get to know him a little better", and then you have more interactions and you build up that trust. The second one, which I'm going to talk about first, is reputation by reference and the reason I'm going to talk about it first is because it's essentially how we bootstrap things. The top of this have a diploma, so if you want to know things about me related to my education I can give you a diploma.

That is essentially reputation by reference. Some institution is saying that they did specific things and are vouching for the fact that I met their standards with respect to this course of study. But who are they? And does this really belong to me? And what standards are they actually holding me to? Now, if I tell you I have a PhD from the University of California at Davis, most people would say, "Oh, University of California. I know that school. I guess that's okay." But if I tell you I have a bachelor's degree from Norda, whatever, what is that thing called?

Norda Medical Thing. You guys know? Down in Provo. Anyway, are they good? Are they acceptable? Or is that a good institution? I honestly don't know. But who does? The way you do it, is this diploma can be a credential and it can be chained to my SEDI credential which now means that I can prove that this

diploma was actually linked to me or given to me. But it also can be chained to the university's credentials and identifiers. They probably have an accreditation credential or should have one that came from their accrediting body. And there's not just one, right?

There would be an accrediting body for the university. There would be an accrediting body for the specific degree program I'm in. All of those… I mean there might be dozens of these out here that the university has. You can also check to see is this a legal business? Is it even recognized as a business?

That would be your commerce department or GLEIF vLEI credential. And importantly when I present this diploma as a credential all of this can be checked automatically, these chains can be followed. We can check all of the credentials and you can get a green light on your system that tells you: yes this credential is issued to this person, it came from this institution, that institution is accredited by this body and this body and this body and it's organized in the state of Utah and all of the things that you might want to know about it.

That's reputation by reference and SEDI provides a nice foundation for that, as do the organizational credentials that we talked about. There's another kind of reputation that I mentioned and that's reputation by behavior. Now, in the physical world, we just interact with each other, and that's kind of how we figure out whether or not we like this person. We trust them, right? The fact that I'm going to let you fix my plumbing doesn't mean I'm going to let you babysit my kids, right?

There's different domains of behavior that we look for. We use reputation by reference to essentially bootstrap this. I needed someone to come in and fix an oven at a at a rental property I have. I didn't know anyone, so I just used reputation by reference, right? I looked at reviews online to see, okay, they've got a pretty good thing. Timothy… – Isn't reputation by reference also an accumulation of behaviors?

It is!, but it's at a distance. We go back to this. All of these could be many layers away from the actual thing that I give you. There are accumulated behaviors that lead to that diploma. But it's not me who watched you complete the degree. It's indirect. It's the university who watched you complete the degree and you're believing the university. There's an indirectness, not a direct link. – But either way, the quality of reputation can't be separated from an accumulation of behaviors over time.

Correct. – That's true. – If it's a good restaurant, it's done well. It's had good behaviors which got that reputation. And what you're doing is you're collapsing the history because it's so well known. I'm going to collapse the history and just say I just need to tell you the name of the organization.

– In reputation you explicitly trust the reputer. If you're observing the direct behavior, the only person you're trusting is yourself. The verifier doesn't have to trust anybody. If they truly do reputation by behavior, they can verify all of… Here's a different way of thinking about it.

Trust always, always requires vulnerability. Trust requires vulnerability. If you aren't vulnerable, you don't have to trust. The question here is, am I trusting me? I've eaten at that restaurant before and it was pretty good. Or am I trusting the reviewers, which is an accumulation of behavior, but I'm trusting someone else, in one case I'm more vulnerable to mistakes than I am any other right because it's indirect rather than direct. That's the difference.

– You're trusting a third party's opinion of the reputation versus your own. – That's right. It's not whether it's based on behavior, it's whether it's indirect or direct, so a different dimension. Right now, the way reputation by behavior works, one example of that is credit bureaus. Credit bureaus accumulate all of

your transactions and whether you pay your bills on time and whether you were late and all of those things, right?

And they use some algorithm to determine a credit score which is then paid for. I mean other places other banks or credit card issuers or employers and others can then pay for that credit score and get that for you. That's one way that we do behavior. Different way is, say every time I have a transaction with company X or company Y, I get a receipt that I paid on time, all of these things. Now, this may not matter as much for individuals as it does for organizations. I mean you can see the organizations could really benefit from something like this because it's big, lots of money but anyway you get these as credentials and now you can prove to other people in combination or apart whether or not you have behaved properly. Now I have to say there's one big problem here.

I can admit bad behavior. Doesn't mean that these things are ever going to go away. I think there's always room for things that are looking for bad behavior. – One thing though that the decentralized model allows is that the verifier can pick their own algorithm.

If I go to get a credit score, there's like five algorithms. And I was going to mention that this model allows for a marketplace of algorithms. And for transparency in algorithms. Somebody might say, " We use this algorithm for how we do how you do scoring." Yeah.

– I don't know if you were talking about it, but you said at the bottom in one case, you don't own your data. You do own your data. And I'm gonna say that this is the sort of issue, legally, that is going to slow us down dramatically because if you look at what the Supreme Court has always said, it's that no one owns facts. Okay, data is facts. You might own the database that the facts reside in, but no one owns the underlying facts. The solution to that is to treat those facts as an unownable commons. Okay? And like all commons, navigable waters or the air we breathe, someone has to be responsible. Going back to the fiduciary duty of loyalty and other fiduciary duties, you are the best candidate to be the trustee over your facts because you're more interested in their accuracy.

– I care about them. – Right. And the government has a secondary fiduciary duty to make certain that your rights in those facts aren't abused. But the point there is I think that's what's going to slow us down because if you look at these 80page agreements where you just click “I agree”, none of them say “you own the data and you're giving us ownership”. They all say “you transfer whatever rights you have and we accept whatever rights you have” and then the company goes out and says “we own this”. It's a legal fiction.

– I probably shouldn't say “own” there. I mean control is a better word. Own is a problematic word in this word for the reasons that you have pointed out. And in fact, if I'm in a transaction with someone, we both are parties to that transaction and we both could have a credential [inaudible] to that transaction and we both control that. And of course, that's just two-party. It could be multi-party.

It's about you have access to and can control who you share it with, right? – And just to your point very quickly about transactions we always sort of say, well, the newspaper costs a dollar. If I'm buying the newspaper, I value the newspaper more than the dollar and the publisher values the dollar more than the newspaper.

Otherwise, the transaction wouldn't occur. – Yeah. There's cost, price, and value. And they're not the same. – Even the word “your” implies ownership. – “data about you”. Okay. This whole idea of reputation has lots of implications or ramifications for how we might approach agentic AI and how agentic AI works with us. I mean this talk is not about agentic AI but I just wanted to point out that this

same foundation and this idea of building reputation around solid identifiers for organizations and individuals helps us in this space. If you want to know details how I think about it like I said there's a blog post.

One thing I thought of when I was doing this actually is this idea of an evaluator. You could imagine agents that you employ that are evaluators. You We talked about having a marketplace of reputation evaluation algorithms. You could imagine a marketplace of reputation evaluation agents who are working on your behalf for various things. Things that you care about and ones that you trust, and “I'm going to use that agent because I like the restaurants it recommends”.

What's the technology opportunity here? If you're building decentralized identity infrastructure, SEDI is an opportunity, but it's not a guarantee. SEDI sets out these requirements: Open standards, decentralized control, that represent a high bar for how this all has to work. And like I said at the start of the talk, there's the technical side of it, which is not in any way a small thing, but doing that at scale, meeting the user experience requirements that we might have, finding the right economic models for the various parts and pieces that have to work. Those are all extremely difficult problems to solve. It's not a no-brainer that “Oh SEDI came along good, it's all solved” and we still have lots of work to do. Timothy.

You've got whole talks about this. – If you in [inaudible] have a legal basis in their giant centralized digital system, use a lot of cryptography as well component missing and that's the decentralized nature of SEDI. It's trying to have [inaudible] a legal verifiable basis but also decentralization meaning that you can have different vendors in different states and different systems and they can issue in one place and verify in another because they're speaking a common [inaudible] We hope!

– Those systems that's obviously the… But that makes it hard, right? – In contrast with Aadhaar in India and in contrast with what they're doing in Estonia because they do not have that property and they're not even trying.

It's a giant centralized system and everybody phones home to the mother ship. – Absolutely. – My point is just having those two components up there describes Aadhaar. That's my point. – Okay. – And it does… going back to your point about control because decentralization is about autonomy and control ultimately.

– (Lawrence) Lessig famously said that “code is law” and there's a lot of people who take that to heart and it is true to the extent that code controls what you can do and what you can't do but I don't think for identity that idea is quite sufficient. That we need more than code, we need this underlying legal foundation that SEDI provides for individuals.

SEDI answers questions about who has authority to revoke, when can they revoke? That's just one example. Who has the right to get one of these? Code doesn't solve those problems. Those are all things that we need legal infrastructure which is steady, at least on the road to providing. I don't think this will be the last SEDI statute. I think there will be more but but nevertheless we're on the road to getting there.

SEDI provides this legal foundation like I've said. We've got code, we've got stacks that are maturing. And what we have now starting in May is a legal foundation for all of that. We've had some discussion going back. I've got like two or three minutes I think if there are questions. Two minutes I'm told. Is there questions? Happy to answer them.

Otherwise, I will turn you over to the next one. [Applause]

TALK 32

Phil-Feairheller-Interview

KERI Conference April 2026 · interviews · 00:41

– Sure. Hi. My name is Phil Fairheller. I am the CTO and co-founder of HealthKERI. And today I will be presenting on KERI infrastructure, focusing mainly on the components that HealthKERI just donated to open source to the KERI Foundation, that being the wallet witnesses and watchers. Sure. If our work in HealthKERI disappeared tomorrow, the decentralized identity system would lose its foothold in healthcare. Right now, we are making a big difference in bringing this technology into healthcare to help secure data exchange between providers, payers, and patients. And we would miss out on that opportunity.

What we at HealthKERI see differently is the impact that this technology can have to facilitate rapid connections that are needed to make healthcare move more smoothly. Thank you.

TALK 33

Phil-Windley-Interview

KERI Conference April 2026 · interviews · 01:23

My presentation was about why decentralized identity, first-person identity as I like to call it, needs a legal foundation. And I spoke to the idea that SEDI, recent legislation that has been passed in Utah, SEDI stands for State Endorsed Digital Identity, can provide a lot of that legal foundation that we need to make digital identity work. I don't think it necessarily would surprise many people at this conference, but one thing I see differently than many is the need for first-person identity, for identity that is controlled by the individual rather than provided to them by some other organization. We've been working on self -sovereign, decentralized identity, first -person identity for since 2015, so 10, 11 years at this point. What has been missing is the legal foundation that makes it possible. I believe that what I talked about, the legal foundation for digital identity is not just a nice to have, but a necessary component of a functioning digital ecosystem where people have dignity and the right to act for themselves.

TALK 34

Witnesses & Watchers

Ryan Hansen

KERI Conference April 2026 · talks · 39:26

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Ryan Hansen of the KERI Foundation walks through deploying witnesses, watchers, and boot services from Foundation GitHub repos. He summarizes an Ubuntu-based setup guide with Nginx and Certbot, discusses Sam Smith's cautions about Docker in production, and shows a wallet provisioning and connecting to live witness infrastructure.

All right. How's everybody doing? My name's Ryan. I'm one of the team, the KERI Foundation development team. We're all pretty new to this gig, so it's kind of a fun little trial by fire to be up in front of you talking about some of these things that we're only getting into ourselves.

I wanted to kind of take the temperature of the room. Generally speaking, I know we've got at least one person in the back that wants to see specifically kind of about deployment. How do we get really started? We'll cover some of that. What else do you guys want from this particular session? Anything in particular? Do you want or should we just get into it and see where the questions go?

Sounds good. All right. The objectives here for this session, we're going to do some overview of three of the five W's. The witnesses, watchers, and boot service isn't really the one of the W's, but secondarily, it's the web in general, deploying to the web. Three of those five W's.

We'll talk about all of those things a little bit. We'll do a walkthrough, high-level walkthrough of a deployment guide that I created that's on the repo. When we get to that point, There will be a link and a QR code if you want to follow along. And probably would be best to or easiest for you if you want to see how it actually works to pull that up in your own laptop and just follow along. I'll just be kind of taking high-level snapshot overview of that. We're not going to actually do the deployment here because I've already done it. But when we're done with all that, kind of walking through what that those steps are, I'll give you a quick view of the wallet launching and then actually connecting to some witnesses and provisioning witnesses and stuff. We'll go through that. Here are the repos we'll be dealing with. They're all under the KERI Foundation GitHub account.

If you go to KERI Foundation, these are the three that you're looking for that we'll be talking about today. And just for a I don't know that you're going to follow along on your phones or whatever, but that's the QR code to the setup guide specifically, you can just go to the KF boot repo itself, and it's just right there on the at the top level. You can click on it and pull it up to kind of follow along with that.

– A QR, but – What's that? – Doesn't show a QR, but – Oh, sorry. I hadn't got that far yet. Next step in .., my bad! I've got my double view here and I wasn't looking at the right one. But we've got some new people coming in. These are the repos we're going to be dealing with in this particular session today. For those who want to just pull this stuff up, I would recommend pulling up the setup guide and just at least following along. I'm not going to go through it like actually run the code in this the setup guide because I've already done it. But we are going to go through it at high level. It'll be easier for you guys to see what's really going on by taking a look at that and just following along as we go through it. And it's all broken out into steps.

Should be pretty easy to follow. Everybody got that? Okay, we'll go on to… I want to just… I I probably don't need to do this with this group, but just to remind everybody the core concepts of what KERI's all about. First, it's about keys. Above all, it's about keys.

And there's two different kinds of keys. Who wants to answer? What kind of keys are we talking about? [inaudible] keys. Public and private. Yes, but there's two different kinds. Signing and private. Signing and… – Rotation.

– Rotation. That's it. Signing and rotation keys. Those are the keys we talk about when we're talking about KERI. You got to remember that there's those are different things. And always just kind of remember that's how it works. The next thing that's one of these core concepts is Key Events themselves. Anything Events related to those keys, those various different kinds of keys. And within that sort of context, there's a lot of ways we could break it down, but I just for the sake of this particular presentation, I separated it into single and multi- signature types of events.

And finally, the Key Event Log, the log that keeps track of all the events. Those things taken together are in effect I mean it's probably oversimplifying but in effect are the canonical core of what KERI is all about. Everything else is sort of in support of this set of things.

Keep that in mind as we talk about KERI ecosystem which this is a slide from Sam's white paper as you guys saw earlier today in Sam's presentation. This describes what the different roles of each of these different components are.

The best place to go and get a real deep understanding of that stuff is Sam's white papers. And if you go to (https://) keri.one, he's got links to the white paper, and he's also got a link to his slide deck, which is related to his white paper. I definitely recommend going and looking through that stuff, too.

There's the three core components we're going to talk about, Keanu. Now, I'm getting back to where your question you asked earlier. Those three core components are and we're sticking with the movie theme. For those who missed it, my first slide was a screenshot from a movie called Witness.

The first core concept or component is a boot service. Dumb movie jokes, right? Second one, witness. We just talked about that. And the third one, there it is. Watchers. Watcher in the Woods. Keeping that movie theme alive.

Those are the kind of the core components we're talking about here to deploy this infrastructure that HealthKERI has so generously donated into the open-source world. Just in time. Come on in. You're going to want to .., probably if you can kick that bag out of the way and just toss on the ground or whatever.

– Thank you. – Yeah. For a lot of what we're going to do today, I'm going to do a high level walkthrough of some deployment stuff. You'll want to go back. Let me just back up and I'll show you. You can get to For those that are just coming in, you'll want to go to this setup guide on the KF boot repo because I'm going to do a high-level walkthrough of that.

That's very detailed, with commands and everything, that you can use after this, but we're going to kind of step through it and I'll talk about it a little bit Just so you guys can follow along. And whether you want to do it live here, probably won't have enough time for that, but you can certainly walk through the setup guide and actually do it on any infrastructure you've got and it should work for you great. Anyway, let me get back to where I was. The movie theme. Okay, a couple of quick disclaimers. First of all, this is a this guide is a very opinionated guide, meaning that it's one way to deploy this infrastructure.

As you get more familiar with how this works, you can start to recognize that there's a lot of ways you could actually do this. This one's going to pick one track and be very opinionated about how to do it. Doesn't mean that's the only way to do it. Keep that in mind. Second of all, all these repos and all the services that they run are Falcon apps, which is just Python – HTTP; in the end.

They're all Python processes. It's all Python under the hood. There nothing really magic about that. It's all Python stuff. It's all early work in progress still as well. I mean the witness and watcher code is pretty robust in terms of that's the stuff that healthKERI open sourced, they're using it in production.

It's pretty proven in that sense. The boot service itself is a little newer and we'll certainly go through iteration and all of this stuff will continue to grow and we'll iterate on it as a team and as a community and it will certainly get better and better.

Don't be too harsh in judgment of how it is right now. It'll keep growing. And finally, this is a demo-only version of how to do this stuff, meaning it's all going to be on one host. We're going to do a multiple-witness configuration on one host. Which is not a production sort of deployment.

We would not say do that in production. It doesn't really give you much help. As an example of what a production and deployment might look like, again with an asterisk, “ish”, productionish, this is a high-level sort of idea of what it might look like in production where you've got your wallet, you've got these providers that provide witness watcher service, deployed on cloud environments across the vertical lines here or separations or maybe zones and and, or sorry: regions, and then availability zones or the horizontals. In theory anybody who's providing these services including a private data center of your own where you deploy this kind of a situation where it's distributed across multiple zones and multiple regions and all that kind of stuff is really the ideal way to do that. And then from the wallet side, the AIDs when it gets to that point in the wallet, the plug-in could provide multiple providers. You could choose from either one provider that has a good distribution into their cloud environment or multiple cloud environments or you could choose from multiple different providers to spread your witness surface out as far as you can and have as much availability and redundancy and all that as possible. That's the kind of idea of how you would want to deploy this in production at scale.

There's a lot of nuance and a lot of different ways to do that, but that's that's what it might look like. With all that preamble, let's get into the actual setup guide itself. Base requirements step, which is step one of this setup guide. Our opinion of this, in this guide, is we're using Ubuntu 24.04.

The latest LTS. It does depend on Libsodium one of the libraries that's underlying KERIPy and KERI you know the KERI code in general. Nginx for, in this case, for TLS termination and reverse proxying and certbot, just to set up certificates and stuff like that.

That's just the base guide setups. (I forgot to mention that .. Oh, no, I did mention it. Never mind. Sorry.) No real magic there. If you're looking at the guide, it's pretty straightforward to get these things set up. If you have any experience with infrastructure at all, Any particular cloud provider, whatever. Our we set ours up on Digital Ocean. I did want… – Stupid question.

Do you throw that in a Docker container or? No, you could. Don't tell Sam though because he does not like Docker for this kind of a deployment for security reasons, which is… – For development? … valid. For development, sure, you could. I run Docker locally.

And you could certainly put this all in Docker and deploy it in that way and into Kubernetes or any kind of orchestration system, but Sam's pretty adamant that there's some major security concerns about that approach. Just be aware of that.

This is pretty much bare metal. No production. Yeah. According to Sam, he would not do that. This guide is bare metal. It's not really bare metal because we're still in a cloud environment and so it's not really bare metal, but it's as close as it gets these days unless you got your own data center with your own hardware.

Any questions so far? Everybody good? All right, step two. Set up a service user that has we're just going to call that user KERI. And there's no login. We're, so the service user, no login on this one. And we're going to set up some directory structure. The setup guide goes in a little more detail about this, but generally everything's under /opt/keri, witness pool watcher and KF boot.

There's some… the guide as you'll see has a broader structure and it'll give you the command to create all that stuff. Just copy and paste and it'll just work. Oh. What time are we ending this session again, Henk? 4:00. Okay, I'll keep that. Keep in mind of that.

– Five minutes to 4 – In five minutes [Ryan misheard, red.]. All right. Quick. Next. We set up uv and Python. This particular version of Python 3.14.0 for this purpose. There's some dependency there for that particular version. We just kind of pinned to that and uv is the easiest way to manage that dependency these days. Anybody that's dealt with Python much… uv is kind of the way to go.

Now we get into a little more of the gut. That's just basic setup. Just to get the environment in place. Now we get into a little bit more of the guts of this install. The witness The repo is called witness HK. That's the healthKERI open- sourced version of the witness infrastructure. The witopnet is the name of the package that will go into PyPy. It's not there yet.

The install in the guide is just an install straight from GitHub over HTTP. It's just a uv install from straight from GitHub with the latest version of the witness HK. But shortly after, I'm sure sometime shortly after this conference, we'll get it all into PyPy and then you can install it from… just with pip straight up. And then of course the next one is the watcher Operational network or I don't know what the op means exactly, but we'll say that.

And then the boot service which acts as kind of a control surface for the witness and watcher provisioning and deployment all… or not deployment but the provisioning and communication and stuff. Anybody following along actually doing it?

Nobody's actually installing it. I wouldn't figure you would. But any questions yet? We'll I'm kind of going a little bit quick through this to get to more of a demo and then we can get to questions about anything related.

Okay, next part of the setup is just the config. And there are various different config files that we'll have to set up to make this work. The witness config is called witopnet.json. The paths are all there on the setup guide itself. In this particular instance, the wallet will allow for a one-witness configuration where, with your AID, you're going to set up one witness or if you've got your configuration set up right on the server, it'll allow a three of four, meaning that it'll the threshold, it'll have four witnesses available and it'll require three of the four for it to be considered fully witnessed. By doing this four times, four different instances of the witness service, we're kind of faking on one host being able to have that three or four configurations and test it out without actually having to spread that across multiple different servers or data centers or whatever.

And that's what the configuration file looks like. That curl is that URL will be the base URL of what the witness communicates back to the wallet for OOBIs and things like that, as we get into that point. As the guide describes, you'd have a different copy of that with a different port for each one of those.

Whatever ports you want really, but in our case, we did 5632, 5642, 52, and 62. Those are the ports that we're running on our infrastructure.

watopnet.json is the watcher config. It's very similar pretty much the same structure just slight differences in naming and the different port. And in theory you can .., I have not done this yet, but in theory the design is such that you can run a network of watchers as well we won't talk about that in this particular session because we I haven't actually done it so we're not quite to that point but you can do that.

And then next we'll set up some launch scripts for each of these services, these are just shell scripts that invoke the various different packages. In this case, it's just the witnesses with the setups. Again, pay a special attention to the HTTP and boot port.

Those will have to coincide with what we put into the previous config. Make sure that those ports align. And that will be used for the public and the internal private communication ports that it'll use. The 5631 in this case is the internal local loop-back communication port that it uses.

And then the watopnet similar, watcher network same thing. Same kind of stuff. Set this up to your liking, but for the most part that's the pattern. And then the boot script, the KF boot boot service has a bunch of environment variables that we have to set up to make this system work, make it all communicate well and keep track of itself and all that stuff. The one that we want to pay special attention to in this particular thing, this particular demo is the boot witness backends. This is a string of multiple witness IPs with the following pattern: witness ID, internal IP and port and external IP and port. As you can see at the end of this string here, it starts after the comment starts into the next one. By doing that with the the all four of those configurations, that's basically what's going to tell the wallet to allow a three of four configuration because we're telling it there's four different witnesses to choose from. We've met that requirement for the wallet to be able to enable it.

– Those IDs are AIDs or? – They're just arbitrary witness IDs. You can call them whatever you want. They're just an identifier internally. They're actually like literal wit wit one wit two, that's the literal name or ID for this purpose.

And then next step is Circus install. Now technically in the guide I think the Circus install is at like set step four or five and then the Circus configuration is at step eight or nine. For this presentation, I kind of combined those. Circus install you can do whenever you want before you configure it. And Circus, anybody know what Circus is? Anybody? Okay.

Circus is a process manager. It's the thing that's going to actually manage the launch and the state of the Python processes themselves, the witness and watcher and boot service processes themselves. We're going to put together a config file as it shows in the guide there. And any file that has all the run scripts for all those different things and Circus is going to manage all that. All you have to do is run one Circus command to start them all up and you can get status from it and all that stuff as opposed to running them all independently.

And then along with Circus we're going to set up 'systemd' unit so that systemd can manage Circus so that it boots on startup and you know keeps the state and the logging of the Circus process itself. It's a little bit two layers deep when it comes to that.

You can restart these services either with Circus Or at the global level with systemd and then finally Nginx setup to be able to… if you want to have TLS termination… in our case we wanted TLS for the initial communication so that was all encrypted and then subsequent communications all signed and all

that stuff anyway. You don't have to set up the TLS stuff, but we did in our case.

And I've done this on two different machines. I've done it on… KERI Foundation is providing the one I'll show. I also have my own that I set it up on as well that is fully TLS, all the endpoints are encrypted, running over TLS.

And that's pretty much the long and short of the actual setup. If you do want to try this out, by all means, go ahead and follow through that setup guide. If you've got servers of your own you want to try it on. You're going to have them, obviously, or a Docker container, whatever. You can just follow along through that. And it should get you up to where you're ready to try it out and see how it all goes. That's pretty much it for the setup guide itself. And there's a couple more steps in the guide, but they're mostly, you Setting up the firewall.

That's because my slideshow's over.

You guys aren't going to see this yet, but… Any questions so far? Anybody have any… – What is the boot doing exactly? It's it's going to… it will create the [inaudible] – It's basically the control plan for the witness and watcher network themselves. When we first launch the… I'll kind of walk you through it. Let me… oops wrong thing.

Let me get my display in the right place… There we go. Let me open up the right project here.

In theory, I think I cleaned myself off enough, my environment off, that we're starting entirely fresh with a new deployment. We're going to initialize a new vault. We'll just call it 'test'. I'll give it a passcode. … a new AID. Now I have the new AID.

Nothing special yet. Now all that does is the local key, the local AID and in the inception event in the KEL locally. I'm not talking to anybody yet. It's just a local KEL, has an inception event in it. You're not going to see that here but that's where we are at this point. And then from here I can go into this KERI-Foundation plugin. This is an adaptation of what healthKERI provided for us. Evan did a bunch of good work to turn this into something more useful for us.

Good job! And for all you to use. You can see it's connected to this particular host which is the KERI Foundation host. I'm going to choose my AID and the profile that I want, which in this case is “three of four”. And then when I click this "begin onboarding" to answer your question, Joseph, it's going to hit that boot service to basically make the introduction and initialize the process of provisioning witnesses, sending back the URLs for the OOBIs and taking it through that whole process of introducing the witnesses, sending a rotation. Since we've already done an inception, it's going to do a rotation to rotate the witnesses in based on what the boot service is returning back from the witness service. Does that make sense?

Ish. Well enough. Get into the code, you'll you'll get it. Now fingers crossed. If I did everything right, this is the big moment of truth when I'm going to click a button and hope it does something. It's going to give me the beach ball… for some reason. I've literally never seen that.

Looks like it's trying to do something. Oh, it failed.

Am I connected to the internet? Make sure that all works. The Wi-Fi here isn't super fast, so… We could try that. You can see it's doing some stuff, but it's bailing. I don't know what that's about. Oh, I might know what it is. Sorry.

I'm gonna kill my session there. I was running in debug mode. I wonder if that's just slowing it down too much.

There we go. Getting a little bit better. Can see all the stuff that's happening here. Does take a minute for this to finish.

This is gripping… watching this work, right? Everyone's on pins and needles like, is it gonna work? Shazam. – What made the difference? – I was running in debug mode on my IDE, which was just puts too much overhead into the process. It was just too slow.

What I can do now with this QR code is I'm going to scan it into I'll use Authy for this one. Any authenticator app should work fine. Okay, now it's… I've got it in my authenticator app and then I can go back into the plugin and this is just a demonstration but I'll go to the credentials. I have a schema that's a test schema that we use for issuing credentials and I will … local ident… oh I got to upload the schema first. Sorry.

Downloads I think. I'm adding this particular credential schema. Going to use that for issuance. – Is that's an ACDC? – It's just a credential just a schema for a credential. It's not necessarily an A… or is it an ACDC? That's a good question.

I think it's just a credential, but that is a good question. I have to look at it. It's just a test one that we've got. And now I have to authenticate with my authenticator. And now I've issued a credential, which is nothing more than a little test just to show that it worked.

And there you go. That's a basic rundown of setting up the wit… using the code that healthKERI has open sourced providing it to you guys to all of us to start to build on, deploy on, test with, all that kind of stuff. We're fairly new to it, ourselves on the KERI Foundation team.

It's been a cool exploration and a cool process. Any questions… any… Anything else you want to know about witnesses, watchers deployment, any of that stuff? Yeah. – How is this whole infrastructure different from what the packaging was three months ago when we done doing the hackathon?

packaging. – Oh, I honestly don't know because I don't know what was going on with the hackathon. This all this code came from healthKERI. It could be the same stuff to be honest, but I don't know what the hackathon was doing.

– I mean the witness is from KERIPy. There's a witness inside KERIPy but that was added by Phil in the early days. It's wasn't like, as you said, production grade. A separate service because KERIPy is just [inaudible] core.

Right. This is a separate deployment a whole separate project that is for deploying a witness network being able to put up a whole… As I went back to… in my earlier slides in a production demo or a production environment that variable that I was showing you, that had all the different configuration in it, that wouldn't be all on the same host those would be pointing to ..., the boot service itself would probably run independently pointing to various different endpoints across cloud and whatever other infrastructure you want.

– And you can run with a single witness, but the wallet can't connect to a single witness? – It can. Actually that's one of the options in here. I'd have to create a new one to go back to it. You can choose one witness and deploy it that way if you want. The ultimate KERI idea is that you want to have a lot of witnesses, but if you want to just run one, you can certainly do that.

And it works just the same way it does here. It did in the demo anyway. Any other questions? Is this useful? – What does Circus add instead of just using systemd? – Nothing really other than it's just a nice little wrapper for all the Python stuff, but you could do it all with systemd for sure.

Handle it that way just as easily. – Circus is just the Python native way of doing systemd type stuff, right? Service management… – Kind of. It's not service management, process management really. I guess it's service management.

It's kind of scoped at the project level. Circus gets installed with the project as opposed to at the system level. Generally, I guess you could maybe do it that way if you wanted to, but generally at least what I've seen, Circus is usually kind of part of it's a Python thing that gets installed with your dependencies as opposed to being part of the server infra or server toolkit.

And now I would like to do some tap dancing for you. Is that good? Anybody questions? Is this useful to you? We want to see everybody get out there and start trying to use it. Get it deployed, give us feedback, make suggestions, whatever. Like I mentioned, this particular infrastructure that I connected to is the healthKERI… sorry is the KERI Foundation deployment of these services which right now is for just demo purposes.

You're welcome to actually use it. I guess I'll show you one .., I don't remember if Jay covered this or I might have missed it in his demo, but as it is right now, when you launch… the Locksmith code, there's a few environment variables that you have to set up so that it knows how to go and discover the witnesses that are avail… or the witness services that's available.

And those are right here that I set up in my configuration for running Locksmith locally. That's the host if you were going to try to use this. That's the host you would want to use, to use our infrastructure. We'll put that out there somewhere at some point publicly.

– That means that's part of where the OOBI… – Yeah. The first phase of it when you first create an AID and click the button it's going to hit this URL here for onboarding to make its kind of initial introduction. It'll use a an ephemeral AID to make the first connection and say that it wants to connect and then from then on it'll use the account after it's set up what's called basically an account relationship.

It will use the account URL for subsequent ..., some of the subsequent calls, some of them go straight to the witnesses, some of them go through the boot service. Yeah. – A little tangential, but [inaudible] between the web GitHub or the KERI Foundation GitHub work?

– it's a thin… it's sort of a blurry line at this point, but the KERI Foundation one came along later because the KERI Foundation itself came along later. And probably if I were to guess eventually KERIPy will move into the KERI Foundation org if it's worth the trouble of moving it.

Maybe not. They're both Sam's… that created both of them. The distinction is more logical than physical. It's not much in other words. We created these ones in KERI Foundation just because now that the foundation exists, we want to start kind of centralizing some of this open source stuff related to KERI into that particular org.

Cool. I mean I have nothing further unless you have any other questions? All right. Thanks everyone.

TALK 35

What KERI allows us to do that we couldn't before

Ryan Hansen Pannel

KERI Conference April 2026 · talks · 32:51

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Ryan Hansen moderates a panel asking practitioners why they pursue KERI and what it enables that prior systems could not. Participants share aha moments from sovereign identity failures and blockchain limits, debate self-sovereign identity, and weigh risks of partial “KERI theater” versus implementations that deliver full security and privacy.

Just really quickly as we get our conversation just started, I wanted to give a couple of quick ground rules. First of all, some of my questions might be fairly open and broad and lead to some pretty long-winded potentially long-winded answers. Just be conscious of the time and the respectful of everybody and make sure we give everybody a chance to chime in. Also, all the questions, even if I direct maybe at one person, they're all open to all of you. If you feel like you want to say something about it, by all means, jump in anytime. With that, our the theme for this conversation, this chat tonight is in effect, what does KERI make possible that isn't possible without KERI? In that frame, the first question I have is a little bit of an existential question, I guess. Why are you here?

And what I mean by that is kind of what's the story again being respectful of the time and all that. What's the story that leads you to sitting in this chair at this time? What's your aha moment with KERI? What brought you to this point? I'm going to just kind of open this one up as a bit of an introductory question for everybody.

Alan, I want to start with you. I guess I should start by saying that I I genuinely believe that we should all be our own identity providers and I don't see how you can do that without either KERI or something very similar and that's why I'm here to see that happen.

Great. And did you have a particular aha moment with KERI when you fi founded and or was it more of a gradual thing for you or it's been a gradual thing and a continuing education by Sam it's a continuing process I can tell you that's good Thomas I would have to say for me the aha moment coming from the blockchain space was realizing that what Sam has invented is all of the things that we value about blockchain without the negatives.

Okay, cool. And when did you actually first get introduced to Sam or to KERI in general? It's going to be about four years now. Yeah. Okay. It's been a while. All right, Nico. Yeah. This kind of brings me back to Sovereign, right? I was introduced by Timothy Ruff randomly messaging me on Twitter as a pseudonmous account whether or not I wanted to invest and the opportunity and failure of sovereign was the exact catalyst for what led me to KERI.

Obviously like the opportunity of decentralized identity was massive and not realized there. And I knew there was something beyond a blockchain when I was posting, I remember on the Telegram of Sovereign about something like a microleddger and Timothy actually messaged me in a sideh show like who told you about microledders? It's like it's like no, I just came to it on my own idea and we started talking about the implications of what could be beyond how do you scale a chain beyond this sort of one chain to rule them all. And then subsequently when things are looking their bleakest Sam comes out with the white paper and Immediately it strikes me like, "Oh my god, this is it. This is the endgame of

how identity needs to be created at scale in a decentralized way." And I was pleading for the rest of the sovereign people to recognize what this was. And no one else saw it. We created key state Capital to solve that problem.

I started working in the telco space only a few years ago and not very deep Into it and I'm a little bit of an outsider in those circles. But the thing that was shocking to me is that large companies have invested billions maybe tens of billions is a better estimate in complying with regulations like from the FCC related to shaken in the United States.

And after 10 billion dollars of investment they haven't solved the problem even a little bit. It's like almost as bad as it was to start with. The only way to solve the problem is with KERI. I agree. I want to ask a little bit about you all come from it seems pretty different problem spaces. Maybe there's some overlap in some cases but what solutions have you evaluated or seen attempted to some of the problems that you're particularly interested in?

What solutions have you seen attempted and how have they failed? Now let's not get into vendor bashing and stuff like that you know we'll try to be nice as much as we can but you know what have you seen as failures and we'll go from there and this is open to all of you so whoever would like to jump in first I I think for me I in particular when I was working primarily in the blockchain space was focused on consensus systems Thinking about again how do you scale recognition of an event globally right and we went across many different various obscure consensus systems some that like are more than proof of stake you have DAGs you have parallel crystal streams it's some crazy structures out there but I think the bottom line is that you need a system that is expressive enough to if someone wants to define the limits of consensus they can do so in such a way that it can compose with another consensus system because it all it is governance right I'm going to push back what I'm going to push back I think with KERI coming from the blockchain space and consensus I think with KERI you don't actually need consensus all you need is one honest watcher to tell you someone's lying.

I Wait, so hold on. Maybe I misinterpreted what you said. No, no, no. I was leading there. Okay. Okay. Right. In the sense that like you don't need consensus. What you need is convergence. Convergence on the truth. To your question.

For me the failures that I have seen in the decentralized identity space is when the logic of digital identity was applied to decentralization of finance where Blockchain networks tried to do the same thing they did with a token with your identity. Then you end up just locking an identity into another network with varying actual decentralization And relying on purchasing of a token for that identity to work and then not really being able to use it outside of that ecosystem. For me the failures are trying to apply what we've applied to decentralized finance to identity and that just does not work.

And why not like give us a little more detail on Well, well, because effectively your identity is once again siloed into a network. It's relying on a token. It's also non-existent once that network fails or that chain crashes. And it has no meaning or authenticity outside of that network.

I need multiple identities across multiple chains of multiple tokens that have multiple levels of security. It just isn't the right way to go about decentralized identity. I'd like to comment that one of the failure modes that I see and that I'm deeply concerned about right now is the idea of business wallets.

I think this is one of the most dangerous and worthless efforts that the world has ever come up with. That's strong words, but I think the whole EU business wallet initiative is destined for failure. And part of the reason why is because it's not respecting the first principle which is you have to have transparency.

If you wrap a border around an organization and say we're going to think of everything inside this organization as being atomic. We can't decompose it in any way and we're just going to say the organization projects its identity outward and everything inside of it is irrelevant.

you end up no matter what your key management strategy is, no matter what your credential formats are, whatever, you end up in failure mode. In some ways, this is a shout out to Carla. I think the fundamental principles of VLES are deeply true. Those principles are that an organization doesn't just have authority because it exists. And people who belong to an organization don't have authority just because they belong to the organization. They actually have authority because they got delegated authority from somewhere. And if you can't track that down, you end up with a massive failure mode for the whole world.

Alan, I see something that is a continuing failure in the world generally, which is something I call identity theater. And I call someone up and they say, "Oh, what's your email address? What's your mailing address?" They ask me a bunch of questions that somehow are supposed to prove that I am me. And I'll say, " did that really prove that I am me?"

And they just laugh at the other end, but they continuously do it. I went through an airport the other day, and they checked our passports five times on the way through to the gate. And why it's why it's just Security theater, identity theater, whatever you want to call it, but it's The SAD thing is that's the that's the best that the business world has at the moment. And they seem to think it's okay.

Better things are coming. Good. Okay. Speaking of theater, I'm glad you brought that up because one of the questions I wanted to ask and I'm going to direct this specifically at Thomas and Allan to start with just because we heard your presentations earlier today and both of you talked about in different ways. Allan, you mentioned kind of taking a bit of a step-wise approach to getting KERI out into the world and Thomas, you mentioned a version of KERI maybe that's not fully implemented as KERI as it is defined. What do you see as the risks of what you might call KERI theater where where it's not fully KERI but that's out there in the world but some form of it? What do you see as potentially risk that we need to be aware of in that sort of situation?

I'll go first. I think it should be Really as much of caring as is necessary to make it fully secure and privacy protecting. But there's nothing wrong with that being built into existing legacy applications to add this layer of trustability onto existing transactions that are at risk.

But no it really if it's not as much care as is needed to make it secure and safe then that's not going to be good enough. Yeah. I would agree. I think there are scenarios perhaps whereby You can take on the risk of not using all of KERI. The KERI light method that I mentioned that the Swiss government are using from my perspective is actually where you shouldn't be taking those risks because you're dealing about with digital identifier people. I also think that the KERI Light approach is just The byproduct of someone on their journey through KERI. I was about to say that it's almost part of that learning roadmap to some extent from my perspective but I would I would emphasize that for digital identifier for humans for people for governmentsized systems Anything but real full KERI with all of the five W's is a risk.

Fair enough. Nico, you looked like you had something to Oh, no. He took everything I wanted to say. How rude. That's great. Daniel, anything to add here? I once proposed a KERI light posture in our ecosystem and there's certain conditions under which you could operate with less than what KERI provides, but I don't think they apply at an ecosystem level. I think that was a mistake.

Okay, good. My next question, what well, I'm going to open this up a little more. I was going to say one thing, but I'm not going to limit it to one thing and put too much of a boundary on it. What what have you taken what will you take away I guess from this conference that either enhanced your understanding of KERI will change the way maybe you approach or describe KERI or what are you taking away with you from being involved in this conference for a couple of days the first inaugural KERI conference I would say first and foremost If it's not perfect, still put it out there. I think the KERI Conference is a little bit of a microcosm of that, right? It's our first step at making a carryon only conference. I think we did a great job considering, but there's definitely ways we can improve on it. There's definitely ways that it can be expanded beyond this. And the same thing with all of the open threads that Sam's designs has allowed us to pursue now this blue sky of opportunities that exists now in the world because of this architecture.

The level of coordination and communication that's necessary to optimize our attention about what really matters is one of the most important choke points I think of our ecosystem. That and money. We need more money too.

Indeed I'm going to take away how delighted I am that the KERI community is as not only as big as it is but growing so fast and that's must be very encouraging for everybody here but especially Sam because it's his baby and it's getting to a difficult teens age I think but It's maturing nicely. Yeah. And I'll just personally add on to that I'm I've been super impressed how deep so many of you guys already are in KERI and understanding of I feel like I'm certainly a little bit of imposttor syndrome even sitting in this chair in the position that I'm in. Anyway Thomas, you want to One of my takeaways from some of the conversations that I've been privy to and observed is there's almost a need to take all of the technical contributors Behind the various libraries and lock them in a room so that they can have effectively a technical standup and have these conversations because I've seen just actually through the hallways from one session to another devs talking to each other and finding a path to solve a problem that they've been dealing with and that's just happened walking from one room to another so I think not only is KERI conference important to talk about the specifications to talk about the opportunities where revenue streams may exist and what everyone is building but also give the devs that are typically only engaging in an online environment that opportunity to lock them in the room. Maybe not lock them, but stick them in the room and allow them to have that same dialogue that I've seen happen just between rooms with a with a focus kind of agenda. I think that'd be extremely productive.

I like that. Daniel, I'll just say that I'm super excited about the is it five engineers, Sam, that you've got working on your five Ws or maybe it's five W's but three engineers or four. Anyway those five W's are really awesome and I'm looking forward to it.

That's maybe one of the things that I'm most jazzed about. Wait, also I didn't get a chance to see the mobile wallet yet. You didn't? No. Where's Keanu? Or no, Jay. I want to see that. But also, and you'll arrange that regrettably as well. With Sam's crippled voice right now. It was that's why we recorded all the sessions so you can see all of them, right?

And additionally, I think the transfer registry primitive that's going to unlock a lot of new opportunities. It's very interesting. I'm curious to see how that fits with the rest of the stack. Cool. The theme that as we introduced it is what's possible now.

That wasn't possible before. And let's flip that for a minute and talk about and some of you or maybe all of you have had a chance to talk about this to some degree. Let's maybe unpack it a little bit more. What isn't possible right now with KERI and what needs what do we need to change to make KERI

possible? Let's start with Daniel and move work our way down or anybody else want to chime in if you don't.

Well I'd like to see the the developer experience be one where a developer says I heard that KERI was interesting. I'm eating a sandwich on my lunch break. I wrote a KERI online or I asked chat GPT and three minutes later I have something that you know plugs up end to end or whatever. That to me feels like a utopia. We're working hard on that. But it will be a while because there's just so much to learn and there's so many things you have to glue together. And I remember Back in 2017, I saw the first demo of a wallet that showed a proof request and a response In decentralized identity circles. And it was such a from the outside it was a really boring demo. It was like somebody holds up a wallet and they click a couple of buttons and then they get a green light and they showed it. And I freaked out when I saw it because I knew there were like 2700 things that had to be right for that demo to work and that's kind of how I feel today.

Nice. And maybe applying this to your specific space like what still needs to happen yet for KERI to be fully realized there? In telco, the demo that I showed everybody when I kind of showed a canned video included Google a Google handset verifying something for KERI. Google built that. They signed it. They showed it at this worldwide conference, but it's not in general distribution yet. I would really love to see that happen. And Apple, who started working on it, I'd like to see them finish what they started.

Nico. I it's kind of a specific one actually. But I really am looking forward to when someone builds an ambient watcher network cache where you can just propagate key state to there and then just repeat that when that comes online and it's sufficiently decorrelated I think the network effects of KERI and the possibilities of multiple Roots of Trust and their credential sets interacting in the combinatorial explosion, right? The Cambrian the KERI explosion, right?

Is that's when the value proposition of KERI really comes alive for me at least in a theoretical sense. I don't know there's so much that is possible to be built with these data structures. It you could talk about this all day. Please continue.

Thomas, go ahead. I'd have to second what Daniel said. I think one of the biggest risks is there's a snowball effect happening with KERI and developers are hearing about it. They're going to explore the ecosystem and maybe they're going to shy away because it's not the most ideal developer friendly experience, especially when it comes to troubleshooting and tracing out errors within your code. And I mean, Fergal actually ran a whole session on this.

There's that element of the experience to the developer. I mean I think a lot of us see that as just a pain we have to deal with for the time being but we understand that the journey is worth it. Outside of that for me I would also like to envision the ability to download the wallet and for there to be a gossip protocol that allows me to just choose which agent I want to connect to. I'd love to see that level of adoption and the infrastructure roll out and I think SEDI is actually one of those initiatives that will drive that type of infrastructure roll out as well.

There's a scenario that may be solvable and may have an answer that but I don't know what it is yet. I'm going to put this out as a question to everybody. The scenario was that the u the governor of the state of Utah has himself a SEDI credential and a watcher network detects duplicity.

What happens next? Impeached. It's a good question. I'm looking for answers. We can't have any political leader who says two different contradictory things, right? That doesn't happen in politics. Yeah. If anybody wants to try to throw an answer out there, I think you stumped him, Alan.

His people will do a rotation for Should we revoke or revert? Oh, no. Revise. Is it revise? Actually, it's a it is a bit more of a general purpose question though where you have a a regular human out there and u some watcher network somewhere detects duplicity on their key and this is just some person who's going to inform them what to do and well you need to your wallet you should have a local watcher that is informing your wallet of any changes to your key state that you didn't source And so it can compare the two and see that your witnesses don't agree with what you did.

It'll tell you, hey, there's bad stuff just happened. What would you like me to do about it? I can do a rotation, repair it. Do you want to call? You All of those are things are part of the user experience that needs to be developed to make it. That's what the wizard in the five W's is for is to figure out so that you can suggest to people the right thing to do. One of the things in critical physical security that we had to figure out is that when bad stuff is happening, the security people that are monitoring the cameras get overwhelmed very quickly.

You can't give them a menu of options. You have to give them like one button, press here, and then good stuff happens. You have to figure out what that press here does so that when there's an emergency, they don't go, "Crap, there's duplicity. My life just ended." They need to be able to say, "Oh, everything's okay." Right? You have to figure out. You have to have enough intelligence to be able to say, "Okay." Right? And rotations are safe.

You can rotate a lot. It doesn't hurt you to rotate. It's just friction. And so you can try to reduce that friction but make it so that it and you know it's like that what's the worst thing that can happen right you know is that you rotate when you didn't need to right but as long as you know I mean rot rotation too much becomes a security problem in and of itself but you can manage that but of course the pre-rotated key then has to be stored somewhere else and so yes that's why I said there's friction there so you have to the regular human.

Yeah. But that's why that's why there's various things that we have to work on to figure out all the different friction points and figure out how to solve those and it's not those aren't trivial problems. Yes, it's a hard problem yet to be solved.

Maybe the answer eventually is we all have a little chip and that's where we keep our pre-rotated key. I want to throw in one other thing. Go ahead. Movie with Keano Reeves in it. That's a really dystopian view of putting stuff in your brain.

I want to throw in one other thing I'm really looking forward to that we can't do today. And that is I'm looking forward to the postquantum support in KERI. I want Falcon to be in. Part of why I want it to be in is because I can see the train wreck happening everywhere else.

And I know that we have a great story, but it will be even greater when the train wreck starts to have casualties everywhere else and we say, "Come on over." I we're ready. It would not be that hard to add it in. That's just a little bug in everybody's ear. Let's get that in quickly. Something that just occurred to me that I would love to see is a login with KERI button just everywhere.

Wouldn't we all? Get rid of get rid of the login with Google and GitHub and all the other stuff and right login with KERI. All right. I actually don't have any further questions. I wanted to open up maybe to the group to ask questions of our illustrious panel here. Those who haven't had a chance to maybe ask questions or hear from some of our guests.

I guess not. I guess we've covered it all. We are coming to the end of or I guess we are at the end of our conference. We've had a chance to hear from pretty much all of you at one point or another and it's really been an amazing experience. For me personally, as a kind of new to the KERI ecosystem, the KERI world, it's really encouraging to see how many other people have been thinking about this for a long time and have been working on this for a long time.

I've got a lot of catching up to do and we're excited as a KERI Foundation team to get in to start addressing some of these problems that you guys have brought up. In the last just maybe few minutes, I'll give you guys each a chance to kind of give your closing statements and your last little messages to the KERI Conference are our first ever KERI Conference. No time limit.

Dangerous. I'll go first by just saying that all of the thanks goes to Sam Smith. None of us would be here without Sam. And I'm looking forward to next year. Indeed. I think one thing that's kind of come up in some conversations I've been having with fellow colleagues From the W3C space that I just wanted to share is There's been a lot of work there in standardization and for me there was kind of a moment that shifted across the last couple of months where what I what I can acknowledge from the W3 three spaces they have standardized the data model and that's been useful for a lot of development work but I believe that what we standardized here with KERI is the trust model and from my perspective that is something that for me is what's keeping me here keeping the team working hard and I look forward to continuing the journey.

Just in the same way building off of Allan's sentiment I just want to thank you, Sam, and also everyone else here who's put their blood, sweat, tears, and time into this protocol and into the people. You all make it an absolute pleasure to be on this stage with and to share in this sort of co-creation of a what I view as a beautiful new future.

And it's a joy. It's a joy to be in this space and I couldn't be happier. The duplicity game is a foot. I'll just say that I think we all need to protect our own identities and reputations urgently. If I were malicious I would, Point AIS at a bunch of different communities and have them wreak havoc.

Yeah. And we have the tools to protect ourselves, but we're not using them yet. It would be great if we toolled up GitHub so that you couldn't make a contribution unless you could prove who you were. It would be great if you know our remote interactions were protected and I'm seeing I don't know how many of you are on the various W3C mailing lists but they like us are vulnerable to the assumption that well-intentioned humans are the ones that are going to show up and they're having AIS show up in the in the comment streams and the email threads and so forth.

and they've shut those down. But this is just going to get worse. I think it might this is kind of a an invitation or a shout out to people that maybe it's time for us to solve our own community's Reputation and trust problem before any havoc occurs.

And to the extent that's true for our community, it's true for every other community out there, too. The world really needs what we have. Let's go give it to them. Well said. And a big thanks to our panel today, our guests, and a big round of applause for all our guys here.

TALK 36

Ryan-Hansen-Interview

KERI Conference April 2026 · interviews · 01:21

My name is Ryan Hanson. I'm one of the core developers for the KERI Foundation development team. Presentation today was about witnesses and watchers and our boot service in the KERI infrastructure that we're deploying and making open to the world. Witnesses and watchers are a key part of a crucial part of the KERI ecosystem the way it's designed to be used and without them there's some value in key management at least in terms of being able to have somewhat self-sovereign keys but without the witness and watcher infrastructure to prove that those things are still valid, that your keys are still yours, the kind of the whole principle underlying what KERI is all about falls apart. For me personally when I first got introduced to KERI, the witness and watcher idea was the thing that really hooked me in because it doesn't really matter what assertion is made if there's not something that can prove that that assertion is true, then it ultimately doesn't matter. In practice the witness and watcher network infrastructure is critical for being able to assert and prove that something is true and therefore to make the KERI ecosystem actually work.

Thank you.

TALK 37

SPAC & TSP

Samuel Smith

KERI Conference April 2026 · talks · 40:49

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Samuel Smith presents SPAC and TSP frameworks for privacy-preserving interaction. He contends systems must let first parties hold second parties accountable inside confidential contexts, using contractual or regulatory liability, and explains partitioning interaction contexts with high-entropy identifiers so routing metadata cannot correlate parties across disjoint relationships.

We need to so accountability is about imposing those things. Since any since you can be correlated anytime you share data with a second party eventually with repeated interactions they're going to know who you are and then they can share that with anybody they want unless you impose on them some sort of liability that says I'm afraid or I'm worried about the cost. I'm worried about something that you've imposed on me some externality that says I'm not going to share this data. I'm going to respect your data rights.

Any interaction system that does not enable first parties to hold second parties accountable is fundamentally incompatible with privacy. It's not going to be privacy preserving. It's going to incentivize them to share your data and violate your privacy rights.

How we do it? If we understand that the only thing we can do is to create confidential contexts which are protected from third party surveillance and within those contexts impose liability on the parties to the confidentiality that changes their behavior. That means that we have to either use a combination of economic -, legal, civil, contractual means or regulatory means and potentially criminal means. I think one of the Utah guys talked about whether or not sharing your data was a form of treason. He says, "Oh, that's a really great predictor, you guys are all treasonous, we can we can shoot you, because the the law for treason is death." Anyway, it was a joke, but I thought it was fun to talk about that. We want to create control-safe relationships where the interactions are protected from surveillance by third parties and second parties are held accountable for any information shared with or by them. It's not just shared with them. It's not just one way disclosure.

We also want to hold them accountable for what they do, right? Because safety is about what is incoming as much as it is what is outgoing. If I go someplace and I want to be in a safe environment, I don't want people calling me names, being vulgar, sharing stuff with me that I don't want to see treating other people badly in my presence. That's all part of the environment. You need to hold everybody accountable for their behavior. That seems very nanny state, but actually it's what a refuge and a sanctuary is. It's the ability for you to decide what interactions you have versus what interactions the internet forces on you. We use this model.

Basically two parties are sharing information. It's always from the view of the first party. You're the first party. What else did you care about? Third party observers may obtain information in this model indirectly metadata because they observe metadata because this is going over the open internet. The open internet has to be routable. That means routing metadata is public. That means there is no privacy on the internet because if routing metadata is public then everybody can correlate who the parties are at either end. It just takes time. There is fundamentally no ability for you to have that interaction without it eventually being correlated. What you have to understand is that metadata is the problem. It's not the

data that's in here. It's the metadata about here and the Supreme Court has defined it that way. You have protections for content data and no protections for non-content metadata. I can encrypt the content.

I can't encrypt the metadata and therefore the metadata will always be leaky. It'll always provide correlatable information. What I have to do is I have to build an artificial construct via tunnels so that I tunnel stuff across the internet so that the tunnel itself is content data relative to the internet so that its metadata is content at another layer. And if I do enough layers of tunnels, then I will have a layer where the metadata is sufficiently well protected that it can't be surveiled from the outside. We can make a highly surveillance-resistant communication channel, right? If the sharing is leaky, there's enough metadata, right? Or they're using intermediaries, then they can surveil that. If there's any kind of collusion, right? The way we're defining privacy is when third parties have no knowledge of who is sharing information, then we have privacy with regard to third parties.

And confidentiality is when third parties have no knowledge of what is what was shared. If the second party shares with the third party, they broke confidentiality. If the second party shares with the third party who they are or who's in the party, then they broke privacy. That follows the Supreme Court definition of privacy and confidentiality in terms of how we reason about it. And it's important because if we lump these all together under one term, then we get confused when we start talking about it.

I try to be really clear and not get confused about it. This is the the TSP and I leveled SPAC because the name of the protocol when I wrote the white paper was secure privacy authenticity and confidentiality and the header to the introduction of the protocol says the goal here is to get the best privacy we can, given we make no compromises on authenticity and confidentiality.

We start with the strongest possible authenticity, confidentiality and then work backwards to figure out what's the best we can do. And the reason for that is, that in my experience, as soon as you compromise on authenticity and confidentiality, you end up with more leakage pass more opportunities and the net result is you have less privacy overall. Whereas if you start with privacy, you make hard trades that you can never recover from in on the confidentiality and authenticity part. And that means fraud, that means identity theft. And if you're worried about somebody knowing who you are because your metadata got leaked and tracking you, then you have to make a judgment decision saying, "What's more important? They know who you are or they steal all your money, "right? For most people, stealing all your money is a much bigger harm. Now, there are corner cases where knowing who you are can be really bad, especially if you're a criminal and you're worried about going to jail, right?

When you're a dissident and you're worried about being repressed. The core of the protocol is a triple tunnel. We have a tunnel between A3 and B3. Oh, here I can do it over here. Can you guys see the cursor showing? No cursor. It was there a minute ago. Oh, I guess it disappeared. All right. Anyway, so it's a triple tunnel. I have three pairs of identifiers. A3 B3 A2 B2 A1 B1. Endpoint A has three identifiers. Endpoint B has three identifiers.

Those identifiers are not mutually correlatable. They're independent identifiers. What would make them correlatable is if they were exposed in such a way that the context correlated them, but they're high-entropy identifier. They don't intrinsically convey any information that says they're related.

The fact that it's B1, B2, and B3 doesn't mean anything. They're just ran, pseudo-random numbers, right? The other important distinction is to understand that clandestine surveillance and counter surveillance for clandestine stuff is purely ephemeral because clandestine says, I'm not detectable. I'm going to minimize my detectability as much as possible so they can't tell what's happening. Covert says

I have an overt detectable identity that confuses the surveiller into not looking underneath the cover of what I'm doing so they don't detect the thing that I'm doing. And for persistent counter surveillance, I want to I want to be covert.

If Im clandestine, usually that means it's a very short op [portunity]. It's only going to happen for a short period of time cuz eventually I'll start to leak stuff. And once they start to leak stuff, there's nothing stopping them from come getting me. Whereas if I covert, they're confused. They go "Wait, well, no, he's not a problem, " right? "We don't see that."

And the covert stuff allows me to hide things in ways that I can't hide it if I'm clandestine. The idea is that we're basically creating two cover identities to hide this guy. They have to pierce two covers. They have to burn two covers to get what we're doing. And the reason it's two covers is because of the intermediaries. What happens is, I create ..., I'll go to the next one, I'm going to introduce the concept of relationships. Remember I have three three pairs of identifiers. I have relationships between two identifiers and the relationship can be bidirectional or unidirectional. And this is important because A can talk to B using a different triple tunnel than B talking to A. And that removes a form of correlation of metadata. We don't necessarily want it to be bidirectional. We want to be completely independent and be unidirectional or not. What that does, is that we can then define a relationship graph of all of our identifiers and everybody else's identifiers and every pair of identifiers forms a relationship. And in that relationship we have routing data and we have identifiers on both ends of each of those relationships. And what we're trying to do is make it so the metadata in those relationships is hidden so that it's not surveillable.

And when I say not surveillable, not easily surveillable. Everything is surveillable, but you want to make it hard enough. And if you think about this, this is a self-identity graph. This is all of my identifiers and my data and attributes. And a relationship graph is just the intersection of multiple identity graphs.

Right? When I go when I was here, I got controller A and notice it he's got multiple identifiers A 1 through 4. Right? His identity graph is that subset of this graph, right? But when he then forms relationships with B, then then those two graphs intersect.

If you haven't figured out .., I love graphs. My brain just works in graphs. Everything I think about is just like "Oh I can graph that!", right. And then I can reason about it. Relationships properties. I know this is a little small, there's a lot of them [properties]. A cryptonym is a cryptographically-derived pseudonym with at least 120 bits of entropy, so we don't have any ability for anybody to discover anything about it because it has enough entropy and AID is a cryptonym that is also securely attributable to one of our key pairs right you guys know all this stuff but some people that see SPAC, especially when we present the TSP, which is like sort of the TrustoverIP's generic version of it. They don't understand these things.

Relationship is just a pair of two AIDs. Two AIDs because they're cryptonyms with high entropy are uncorrelated. There's no .., from an information theoretical sense, there's no knowledge that one AID provides about the other.

People say things like, "Oh, they're super they're super cookies or they're super correlators." From an information point of view, they're not. The only thing that correlates them is the context of their use. They themselves have zero correlatability. They're totally random. That's what high entropy means. If I don't use them in the same context, they can't be used to correlate me. The only way they can be used to correlate is if I use them in a context and the context leads to information that allows the correlation.

It is the context that is the supercorrelator not the identifiers. A context is a set of events. We got to define that two contexts are disjoint with respect to an AID when the AID appears in one or more events in one set but does not appear in the event in the other set. If they're disjoint, they're not correlatable.

Right? The goal is build disjoint context. We want to partition our context. A relationship is not a communication channel but the event sent over communication channel may be a context for the AIDs in relationship. As soon as I start sending information I create a context and that context may correlate me. Interaction graphs are all about correlating using context to correlate things.

A partition is a set of contexts with mutually disjoint relationships. I'm being kind of mathematical here. Any set of relationships using one relationships identifiers may form a partition. That's how we get disjointness. We only use an identifier in one context.

And if that context could potentially form a partition, therefore I'm not correlatable outside of the context. Partition relative to other contexts. Any two member contexts of a partition may be correlatable due to other information associated with those contexts, but the partition relationships by themselves provide no correlatable information. Partition relationships are by themselves not mutually correlatable. The relationship isn't the problem. It's the context.

Think context when you're worried about correlation. Don't think identifiers. We sort of have inverted that when we talk about linkability because we say cryptographic unlinkability is about identifiers. No, it's about how you use the identifiers in what context. That's where the correlation comes. It's a subtle nuance. It's subtle. I get it.

But it's vitally important if we're designing protocols to understand that subtlety. An AID common to any set subset of events within a context provides a perfectly correlatable feature across those common events in that context. Now why do we want perfect correlation within a context?

We want perfect correlation within a context so that we have accountability. That's how we get accountability. You can't do something and not have me know that you did it. I have non-repudiation. I have absolute ability to point the finger at you and say you did that. If you leak information, I know you leaked it.

As soon as we weaken this and use identifiers that are not perfectly correlatable, we lose accountability. The whole goal of what we're trying to do is to create safe contexts we just threw away. We threw away before we even built anything. We started with something that's broken and we can't build it from there.

We have to start with this and then we have to isolate the context. Within a context secure attributability together with perfect correlatability across that context ensures reputational trust. I just said that in a different way. Accountability, right? If I put a gun to your head, I know what you're going to do. Right?

That's reputational trust. Yes. Do you see any situation when you would want to not have accountability? There's lots of situations like that. I'm not precluding those. But when I'm talking about internet safety and I want to have valuable relationships with value to me, then they're not valuable unless there's accountability. Yes.

The Signal people really emphasize deniability. That's a really good question. I can look up two papers for you. The BYU security research group did two several papers they published and they asked the question, is deniability actually what they say it is? And they and the answer was no. People want accountability, not deniability.

I' I've heard that as well. I've also heard studies that no jury has ever used this alleged cryptographic property to do. That's right. That's the other thing they found out is just that they claim it, but they won't the judges and the juries won't enforce it because they don't see it as something that they can adjudicate [rule on, red.]

and I believe it actually breaks down the device case, but that's a separate Yeah. It doesn't work in any sense. Yeah. But the conclusion was It's a false concept. It's fundamentally designed as a way to advertise the fact that you don't have the good properties you want but you have other properties that if you're criminal sound like they're good and actually they are for criminals, right?

But it doesn't protect you, the noncriminal, it's actually harmful to you, the noncriminal, because you're the one who's going to be exploited by the criminal who has deniability, and you're expecting that you have some recourse or protection, and the law says, "No, no, we're not going to give you protection because they don't believe they should protect criminals." And deniability says, "Oh, you're a criminal then, so we're going to go after you anyway."

I'm summarizing, but you should read the papers. They're pretty good. Just look up BYU and deniability. There's two papers. Yes. In use cases where you might need deniability like voting, you wouldn't use this. You'd use a different protocol.

Yeah. Because one of the thing one of the core tenants of voting is that you don't want to allow repression of voters by exposing their votes. You want to have a blind vote. If you can tell what a voter did and what they said and hold them accountable, then that has all kinds of bad effects.

You would use your KERI for the ticket and then you exchange the ticket. Yeah. You want you use KERI for the voter ID, not for the voting. Yeah. Partitions balance. This is the reason why we talk about partitions. They balance these two things. Strong authenticity and strong confidentiality within a context. That's one side. We get accountability when we have authenticity and confidentiality.

Confidentiality isn't really about .., we actually get accountability with strong confidentiality when it's a symmetric with sufficiently strong privacy between contexts. That's how we do it. We say okay I want context with high accountability. I want to sacrifice that because that's where most of the value is. But I want to have a partition.

What can I do to make the partitions as strong as possible? I can't make them impervious [Impenetrable, red], but I make them strong enough that the tradeoff between account value from accountability and the loss of value from surveillability or trackability or something else is on the side of what I do. And we can do really good stuff here. And often I get in conversations where I criticize the privacy community and people get defensive and they think that I'm just throwing the baby out with the bath water. No, I'm going to great lengths to do everything I can to make those partitions as strong as possible, right?

I'm doing everything I can, but I'm never going to sacrifice these two just to make it easy on me to make those partitions strong. I'll have to do more work to make the partition strong. Yes. You mean privacy or confidentiality in the last statement?

No. Privacy. Confidentiality is easy. Privacy is what's hard. Privacy is knowing who's who's the party. Explain what it means to have privacy between contexts. The parties in each context you can correlate. If I go to a bank and I'm interacting in my bank context and then I go to Walmart and I interact my Walmart context, Walmart and the bank can say, "Oh, you're the same person that they broke the partition.

There wasn't a partition there." tracking and surveillance is about aggregating context because that's how you gain value, right? I need to know how you use, what your behavior is. If I want to advertise to you, I want to profile your behavior. That means tracking you across contexts.

You're saying "if you think about all of those interactions as building a a connection graph, " [Sam corrects:] interaction connection graph "the ability to define those connections, I want to be in control of the connection graph." Yes.

that I want to make it so that there is no correlated build metadata whenever I interact, right? That they can't build the interaction graph. I'm basically looking at what NSA funded in the post 911 world. They spent lots of money developing software that allowed surveillers to build interaction graphs.

They would know that if you're talking to this person, they're talking to that person that somewhere in that network they're talking to somebody that that is a bad person. When you do a FISA warrant, one of the things you do is you say, "How many hops is it?

a one hop fight a warrant, a two hop or a three hop?, "right? One hop is your immediate connections in that interaction gap. Two hops is two hops out. Seven hops is the whole world, right? A three hop FISA warrant is a pretty broad warrant, right? And people go, "Oh, well, you Right?" It doesn't take much, but it takes a lot of work to to fight that. Let's see how we do that. All right. Here another acronym PARK portable authentatable reputable contextual cryptonimity that doesn't roll off the tongue very well.

All right. First we have to start with how we make the strongest possible authenticity and confidentiality. We're not going to sacrifice that. And this is called ESSR encrypt sender sign receiver. What it looks like is: I create a message. I have cipher data. Inside the cipher data, I have the source identifier.

That's encrypt sender. Here's the data. I then have a destination identifier that's within the block that's signed. That's the sign receiver part. And then in order to verify the signature, I need to send the source identifier. The source identifier shows up twice because I need it to verify the signature.

What that does, is it satisfies all of these and I'm just going through these, you can go read these papers, but basically there's been this tension and this goes all the way back to my rate days of what should I do? Encrypt then sign, sign then encrypt, sign and encrypt, right? These are the properties I want: I want third party unforgeability plain text, third party unforgeability cipher text, receiver unforgeability plain text, receiver unforgeability cipher text. The only one that gives me all four is ESSR.

And signed and encrypt only gives me one of these. It's really bad. What I want to do is ESSR. And this is so weird because this is like a paper from 2001 that nobody paid attention to, but it tells you exactly [how]. And now, 18 years later, everybody's worried about key compromised impersonation attacks. And the IETF 9180 Hybrid public key encryption standard basically says, "you can't do it with encryption. You can't do it with chems you can't defy helman you can't solve this because it's it's vulnerable to key compromise impersonation attack." But ESSR isn't and so they say you got to sign. If you're not signing, you're vulnerable.

Every protocol on the planet that doesn't sign is vulnerable to one of these so let's go back I already did this yeah that's just a copy of that slide okay so let's look at the .., this is a little complicated. I'm going to skip through this. I don't think you want all this.

Yes. How many P and Q's do I have to mind in order to get privacy? How many P's and Q's? Minding my P's and Q's. How many of them do I need to? I don't know what you mean by is that a joke? It's a bit of a joke. Okay. How many? But you have one P and one Q. Is that all I need? Because that's not

very many. I can surveil that right one. No, no, the triple tunnel. I'll have to explain how the triple tunnel works. I – Sam, I think I need to Oh, there I here we go. Seeing a single tunnel. Yes.

– One of the things you said a bit ago that I think everybody misses in this is, is your connections to me, at least visually, are loosely analogous to unidirectional onion routing. Yes. But the other part of that is: there's more than one AID.

Yes. And that's a nuance I don't think I've ever heard you say before, but it's an important one because the correlation that people complain about, including myself, is reusing AIDs. And your other diagram showed that you don't do that.

No. That's important. That's why I said it's covert. I have two AIDs that are cover identities, – right. What is a covert op? A covert op says I have a cover identity. Let's say I'm going to go to some foreign country and I'm going to dress up and act like a shopkeeper.

Actually, a good one for World War II is I'm going to get a job at the train station as the ticket taker at the train station. I'm going to spend years building that cover identity. And during the time that I'm building the cover identity, I'm clean.

I'm spotless, I don't do anything bad. Right? Now I have this cover identity and everybody trusts me. I have this reputation that says I'm the ticket taker at the train station. But guess what I can do now? I can surveil people that use the trains and if I have a dead drop someplace, I can share that surveillance information and nobody goes, "Wait a minute. Who's that guy standing at the train station just watching everybody?" And they're going, "What's he doing there? Why is he there?

He's surveilling." That's right. But if I'm a ticket taker, I'm there every day. It's my job to be there. Give me your ticket. Oh, okay. Who are you? Who are you? Okay. I know who is everybody getting in and off the train, right?

Nobody suspects me as surveilling because I'm supposed to surveil as part of my cover identity. This is great. What we want is a cover identity that says we're doing something that is innoculous, that doesn't reveal any information, but we're going to use it to cover something else we're doing that the surveillor can't see through. Right?

The first cover identity is that I have an endpoint and I have an intermediary and I have a pair of AIDs that allow me to communicate to the intermediary. It's perfectly surveillable. It's from the privacy point of view.

These two are used to route information between them. Right? I can encrypt the content. I have confidentiality but I have no privacy. That's okay. This intermediary has a cover identity P and Q that allows these two to talk together. Right? And then I've got a Q to B1. Here's the path. My packet's going to follow this path. Now what's not shown on this graph, which makes it really complicated but is vital, is that an intermediary .., what does an intermediary do? It routes. If it's if it's an intermediary of any value there are 100 or a thousand or 10,000 Q's that P talks to.

When Q when P sends out a packet, it's got to hurt privacy of a thousand because any packet it sends out addressed to any of those Q's is uncorrelated with a packet I send to P. Somebody can observe this pack and they go, "Yeah, well it went here and then a thousand packets came out here. Which one is it?" I don't know. Right? That's that's essentially how onion routing works.

The problem is that intermediary P has to know which Q A wants to send to. He's now a problem. And intermediary Q needs to know how to send to B, right? He's a problem. I can't just tunnel once because the intermediaries can disclose the tunnel. What do I do? I add a second tunnel. I've got A2. Now A2 is encrypted.

Intermediary P doesn't see A2 because it's content data relative to the pink. Now inter[mediary] P doesn't know where the packet's actually going. It just knows that I need to forward it the next hop. And that's all it knows, right? And Q, right, he is the one that has to unwrap this one to know to get it to B2. He's still a problem. Now I do the triple tunnel and A3 and B3 are not known by any of the intermediaries. They can't see it.

The reason I need three tunnels and only three is because I want the intermediaries to not surveil it either and they're second parties, right? I have third parties that that can surveil this. I have second parties that can surveil this, but nobody can surveil this. I don't need onion routing. I don't need 20 onion routers. I need two intermediaries with high fan out and high fan in.

That's it. It's inherently easier to build. Any internet service provider can build this. And they're not a target for law enforcement because they're doing onion routing. They're just doing what they're supposed to do efficiently route lots of packets.

– It seems like the subtlety is that we're talking about routing at an application layer versus routing at an IP layer. No, it's both, right? Because I have to route at the IP layer. – But the IP layer routing is inconsequential.

No, it's not because this one here, the IP layer routing correlates to this and so it defeats me. But at this layer, then yes, IP's gone at that layer. No more IP. I've done two things. I've gotten rid of IP because it's in this layer and now I'm in a an application layer and now I'm in an application layer inside the application layer. And that and so it took three. I took me a while to figure it out. I had to li that's why I have these diagrams with all these 50, 100 symbols on it because I'm keeping track of every piece of metadata and I'm going, who knows what metadata? What can they surveil? And after I got to three tunnels, I said, "Oh, wow. I don't need any more. I didn't know. I thought I might need four or five."

– If the routes were isomeorphic, then it all collapses. Right. What do you mean by that? – If the route for .., if the pink route is the same route .. That's why I said you have to have a fan out. You have to have high fan out.

It's not just fan out. It's that there's randomization of the fan out. No, you just have to have white noise of the fan out. There's two ways to do that: One is you can try to randomize or two is you make sure that the probability is uniform. You have a uniform distribution. And the way we do that .., because we can't manage the randomness, what we do is we make sure that there's no correlatable metadata in the packet itself. That means the intermediary, if they're a good intermediary, will send data on all of their fan out to maintain a uniform traffic level so that you can look at and see a burst of traffic from here and then see it populated.

That's practical. Yes, it's absolutely totally practical. Every TCP connection does that anyway. It does that anyway. We're not adding any net traffic. Huh? – Can be bursting. It can. But, then you do scaling. You do elastic scaling to handle the burst. What we're saying is that you design for a certain level and then you can either decide here to look at your own statistics and decide to delay or not send based on your statistics, right? And every packet is the same length. Yes, if you're trying to build optimized

routers, you don't like this. But if you're trying to build for privacy, you want uniform. You want white noise and so what is more efficient? Two intermediaries with high fan out and uniform distribution or 20 nested tour routers three minute warning.

Okay, – Just real quick. On that diagram, P is aware of A and Q. Oh, sorry. And Q is aware of P and Q. In order to route the packet from endpoint A to endpoint B, you have to send the route with the packet, right? No, I get that.

My question is what if P and Q collude? If P and Q collude, the only thing that they can do is DDoS this route. That's all you can do. They can't expose it. They can just break it. That's you're always vulnerable to that.

If you're worried about that, then you send the same packet by a different route. You do multi-perspective routing so that you have multiple intermediaries. If two of them collude, it still gets through. And that's why having it not be channel-based but be totally asynchronous and unary instead of by you know birectional unidirectional allows you to do that. And since all the messages have cryptographically secure unique sets you can you can figure it out at the other end that they're .., you can do duplicate detection triply with with duplicate detection replay attack KRAM. Where's the KRAM guys?

There you go. Sorry. I was looking for the first I was looking for the first shirt like this. Yes. You do things like that to protect it. But those are good, those are all good questions. We added to the protocol. We just need that that would be all that's required. But we added another protocol because we found that it would be useful and that's a relation formation subprotocol. What that allows us to do is that once we establish a route and I won't go through all this. Okay, I'll go back. Once we establish A3 and B3, we have a secure private channel.

We can use that channel to create other secure private channels. And all we're doing is saying, I've got an A4 and I've got a B4. And we'll form a relationship with those two. Now within this context I can create other secure contexts and when I create those I can exchange routing data that says move A4 and B4 to a different set of intermediaries.

Now I can use one of these. I can stand this one up and then I can do Do this on steroids because I say okay the first five packets we're going to use this one we're just going to we're going to create A4 and B4 and part of the information we exchange is a new set of new pair of intermediaries that we're going to use and every time we send a message every day we just create a new one. Every day we have so we're having one-time use contacts and onetime use contacts are highly uncorrelatable because you're minimizing the amount of correlatable data because you're creating a new pair of non-correlatable identifiers within a secure context that is a partition. You can create partitions on the fly as many as you want which satisfies all the usable properties that people have for privacy without sacrifice security or confidentiality or authenticity.

We're done. All right.

TALK 38

State of the KERI Suite

Samuel Smith

KERI Conference April 2026 · talks · 58:17

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Samuel Smith reviews the state of the KERI Suite and the journey toward adoption in 2026. He highlights early milestones—Linux Foundation standards, funded open infrastructure, SEDI, telecom and healthcare pilots—and relates them to tipping-point dynamics. He also sketches a brief history from RAET and OpenReputation to today's conference moment.

All right. I want to introduce Henk van Cann. Right, the KERI Foundation (https://keri.foundation) would not exist without him. He came out of retirement. He likes to play guitar and sing on the streets in the Algarve in Portugal, and he could be doing that instead of being here with you guys.

Give him a big round of applause. And he's also a financial sponsor of the KERI Foundation. There's no amount of thanks that is too much for him. I want to thank everyone for coming. This is KERI Conference 2026. Get a t-shirt cuz there'll never be another first-ever KERI Conference.

And I think they're downstairs, right? Yeah. All right. And then… one more shot for the for the sponsors. I'm going to talk about the state of the KERI Suite, the journey to adoption. and I look at 2026. We've been doing KERI for a lot of years now, and it feels like this is going to be the year of KERI. Mark your calendars. This is where the world starts to understand and and know that KERI exists.

Some early feats that have already happened in 2026. The KERI Suite of protocols released as Linux Foundation open standards. That was a four-year process starting in 2020. And we've got TrustoverIP. Drummond [Reed, red.] who is the chairman is here today.

And then we have a couple of members of the steering several member at least three members of the steering committee here today. They've been helpful; Karla, Niko and Drummond. We started a funded project to build open infrastructure for KERI.

We received a grant from the Utah Governor's Office of Economic Opportunity to build a digital identity utility for SEDI, based on KERI. the Telecom Industry's OVC Alliance is doing proof of concepts to vet use of the KERI Suite of protocols and the verifiable voice protocol. And later today, you'll have Daniel Hardman is here. He'll talk about that.

We have Utah State passed the SEDI implementation law funded law to build SEDI. There is production implementation of using those for stopping call spam. Healthcare's vital Initiative. We'll have a talk later today about that with the vLEI, KERI and using the KERI for healthcare data provenance.

We just completed the SEDI Policy summit the last 2 days. How many people are here that were here the last 2 days? All right, almost everybody. Fantastic. You so I don't need to tell you any more about what happened there.

And first-ever annual KERI conference That's just this first 4 months of 2026. We're just getting started. One of the things I want to look at is what are the chances of broader adoption? What are the sort of rules? I went back to Malcolm Gladwell wrote a book called The Tipping Point.

And he had rules for when adoption reaches a tipping point. And this is a curve. And the tipping point is this inflection point down he- down here where you sort of go from flat adoption to where you start to get a significant grow to a steeper curve. And the the reality is down here, it's not a smooth curve. It's

really noisy down here. Really the tipping point is when the noise starts to dissipate and you start to see month over month, week over week, year over year growth. And where does that tipping point happen? How do we get to that tipping point? His three rules are the law of the few, the stickiness factor, and the power of context.

The law of the few. What does that mean? If you have a sufficiently influential set of early adopters, those are the few. That means you have to have people that recognize the technical suitability. And then they have to try to get other people to adopt it.

You guys are the few. You're the ones. Let's see what's happened. You have to first overcome the seven stages of adoption resistance. I know there's some people in this room that [are in there:] shock, denial, anger, bargaining, testing, and acceptance, right? Most of you are well along. There's still a couple that maybe they're in the bargaining stage yet. They're still trying to well, do we really need KERI or does it really solve what we want? We'll see.

What are some adoption vectors that the few start; sort of like an epidemic, right? What are some of those? I sort of hinted at those. We have the telecom sector. That could be a massive adoption vector. We have the healthKERI sector. That could be a massive adoption vector.

And we have state entitlements, which Could be a massive adoption vector. Any one of those could cross the tipping point. And since I'm a fault- tolerant person, I believe in fault- tolerant systems. I got to have three, right?

You got to have three massive adoption vectors to make sure one works, right? That's where we're at. Widespread bootstrap vectors. The difference here is in order to have adoption that is we had a saying in Florida, "What comes in over the transom goes out over the transom.

As a wave washes fish in, it's junk fish, you don't want it, right? But what we want is to have adoptions that we didn't plan on. We didn't know that they were going to happen. And they they just spin up spontaneously. We need something that will bootstrap those without us forcing them. These up here are the result of years of work.

We started working with SEDI, three years ago. I know how long Provenant and healthKERI have been working in their sectors. It didn't happen overnight. It took lots of time, right? But we want to also have vectors that happen quickly.

We have two bootstrap vectors, organizational Root of Trust, that's GLEIF. And that's been the thing that has been enabling these other ones, right? They went to production with the vLEI in 2022. That's now been 4 years. Seems like it was yesterday, but it's been 4 years.

We're starting to see the fruits of that. But one of the things that is new this year and in the next year is a personal Root of Trust. That's never happened before, not an organizational root of trust, a personal Root of Trust. That means lots of businesses that wouldn't normally consider using digital identity, centralized digital identity, are going to get for free bootstrapped identity assurance that is highly trustworthy.

That means every industry has the potential to leverage that Root of Trust, which is Utah state. And then all scale enablement vector, that's open standards, open source, and open infrastructure. I'm going to talk about what I mean by open infrastructure a little bit later in this talk.

There's the law of the few. I feel like we've got that. The next one is the stickiness factor. What makes it stick? It needs to be compelling and contagious. We have to think about things that make it, right? We need to have some sort of pathology that's endemic that isn't going to go away and it's going to stick and it's going to keep driving adoption, and that is fraud.

The world is filled with fraud. Every business is suffering the cost of fraud. And fraud includes identity theft. It includes, all of the security, exploits, ransomware. But, it includes just normal day-to-day business, factoring fraud, PO fraud [Purchase Order, red.], those sorts of things.

And they're getting worse. And one of the things that are making them worse is the fact that the tooling that enable fraud has now reached the point where it is better than the mechanisms that protect against fraud. That's going to continually push people over the edge. They're going to say, "Do I really need KERI?" And you go, "Can you fix fraud without KERI?"

And they go, "No, not really." Okay, then you really need KERI. Regulator imperatives to repress fraud and abuse. Those are happening in several industries. For example, in the telecom industry, FCC. In the healthcare industry .., you see Scott [Scrimshire, red.] nodding his head.

In state endorsed digital identity. You're you now have regulators that are saying, "Hey, we need to we need to we have an imperative that says we've got to get rid of fraud and abuse." And by abuse, I mean exploitation of citizens in the case of personal identity.

The malaise of digital exploitation. Do any of you feel like you're sick of being exploited online? Only a few of you. Okay. Do you know of anybody that feels like they're sick of being exploited online? Okay. [laughter] Do you have a desire to control your digital presence? Yes.

Absolutely! The That's a stickiness factor. Is that desire going to go away? Do you know anybody wants to have less control? I know some. Yeah. There are few. But most of the .., everybody in this room, I'm pretty sure, wants more control over their digital presence.

The other thing that's happening is that the technology that enables KERI is key management. And that means the devices, the software, the systems, operating systems, have to start moving in the direction of educating people to manage their own keys. And that is happening.

When we started KERI and when I started doing digital identity in 2015, 2016, the people were really saying, "There's no way you're going to get people to manage the private keys." The blockchain community had already gone there. They had to manage private keys to be on the blockchain. The technology got a big boost. Now you're seeing it in every system.

This endemic pathology is pushing people to better key management. As people become familiar, the friction to adoption reduces. I think we have stickiness. The power of context. What that means is timing and externalities. Is the timing right?

I think the specter of AI is making an unprecedented context that says the timing is right. We are faced with the potential for the annihilation of trust. There's an impending doom. I don't If you guys don't see it, if you aren't, go read OWASP 100-page report they just published on how to deal with AI threats to security.

And it is chilling cuz they say it has never existed before, it is a different type of threat, and we don't have a cure. All we can do is mitigate some of the symptoms. And they're only going to get worse. And it's 100 pages long.

Right. And it's just starting. They're going to add 100 pages on a regular basis because the threats are increasing at an exponential rate. Impersonation, been around for a while. AI impersonation, deep fakes, those, they just barely happened.

When was the ChatGPT moment? 2023. It's only 3 years around, and we're already seeing those effects. Exploitation, you thought somebody could exploit you? Wait till an AI exploits you. Surveillance, you thought you were being surveilled online?

Just ask an AI to surveil you. I've done this test for people where I went out and said, "Hey, I'm working with X, and he needs somebody to manage his online presence and reputation, and I'm doing that for him." And it says, "Okay."

And after about three prompts, it says, "Would you like me to create an interactive dashboard that tracks all of the things they say and do online?" And I always say "No." It takes literally 10 minutes for anybody in the world to say, "I want to know everything about everything that anybody that they know of that they have a name for has done.

And that's just starting. We're not even close to the capability yet. Sufficient political will to legislate new civil rights. I think you saw that the last 2 days. Individual control over digital ID data and context. That is what we need is legislation that gives us back control over our internet presence.

Duty of data loyalty is probably the most significant legislative initiative that we can put in place to enable control over over our digital ID presence online. We can't do it with anything less. It inverts the liability equation. And for those of you who don't understand the difference between consumer harm legislation and fiduciary law, the standard of proof is different.

if I have consumer protection, I can't punish the person abusing, me unless I can show harm. The standard of harm is pretty high. And it's pretty hard to show harm in a diffuse internet online where your data gets spread out all over the place and you can't point to somebody and say, "You harmed me." They go, "No, I gave the data to this person. They gave it to that person." And then you can't show harm.

But fiduciary duty of loyalty says, "If the fiduciary is self-dealing, then you can punish them because your interests have to come before the fiduciary's interests." That means everybody that collects any data about you online in Utah now has to understand that if they self-deal, if they benefit from your data and you don't, you can go after them.

That has never happened before. That is the way place where it has to be. That's the stickiness factor cuz that's now in law. And that stickiness, that's the context. And then the last one, the inevitability of KERI. You have to have true believers who are going to fight through the fights, who are going to go to the meetings and put up with the arguments and make the case. And I think we have cadre of people that are willing to do that.

People that are willing to do that. And the thing is, we have a severely constrained solution trade space. And what I mean by that is, I spent years doing the trades. What can we do to solve these problems? What tools do we have available? And everything every time I go, "I could do this, but wait, if I do that, then here's the tradeoffs, and I don't like those tradeoffs. And I do that, here's the tradeoffs. I don't like that those tradeoffs." And once you do the tradeoffs, you realize that there's only a very small space for a solution, which means, as far in my opinion, if it's not KERI, it's going to be something that looks, smells, walks, and quacks like KERI.

They just changed the name. Because the under the hood, there just isn't any other solutions. We have to .., cryptography only gives us a handful of things that we can use. That means that the timing is right. Anyway Persistent problem cascade that continually exhausts anything less capable.

It's just every time you look at some other system, you look at DNS/CA, you look at OAuth, OIDC, you look at these systems, and they keep failing for the same reasons. People are starting to recognize that.

What does that mean? It's a matter of when, not if. If it's not this year, it's next year. If it's not next year, it's the following year. Because none of this is going away.

I'm going to predict that before the end of 2027, KERI adoption will reach its tipping point. All right. The journey begins. What I wanted to do on this first ever conference is give a brief history of how we got here. Sort of like the foundations.

I think most of you have been there through most of the journey. But, I know there's some people that haven't been, and this talk, will be made available online for other people. I thought this would be a good opportunity to sort of talk about, where it came from.

I started down this path back in 2013, 2014. I wrote a protocol called RAET. It's reliable asynchronous event transport. It is an end-to-end, peer-to-peer, encrypted, signed, and it was used in production at scale, and it was open source.

This was years before most of you have even heard of Signal. Back when Signal wasn't called Signal, it was called Whisper. But, it was the start of that. What was the lesson learned? I could solve all the problems but key management with RAET.

I said, "Okay, that's not a solved problem." I already knew that there was a big problem that had to be solved sometime. OpenReputation I wrote a white paper. It started using self-certifying identifiers. It had the concept of identity ledger and identity graph.

It defined reputation as a modulator for trust for online interactions. And it brought up the idea that we wanted to have individual control over the context in which we interact online. Our internet presence. And lesson learned from that is that I needed persistent identity. I can't have a good reputation without a persistent identity, and that's the hard problem of identity because nobody had a solution for it back then.

Identity System Essentials, this white paper. This is the first draft. Introduced cryptonyms, built on the SCID model. It provided a formal identity model, primary secondary IDs. Talked about identifier management with HD key chains, three degrees of privacy, and it introduced the concept of self-sovereignty for identity. And the draft paper was shared in the community, and lots of people amplified and built on the concept of SSI, but as far as I know, that was the first. And I stole it from Phil. Because Phil Windley had written some blog posts on self-sovereign data, and I said, "Well, we don't just want data to be self-sovereign. We want our identity to be self-sovereign." That was back in 2016.

The lesson I learned, there's a hard trade space. If you want to trade off security and privacy and self-sovereignty and control, you're making some pretty hard trades. Sovrin Foundation. That really changed the environment for decentralized identity.

The idea was there's going to be one ledger for digital identity. One ledger to rule them all. And it had persistent identifiers. It had prophylactic [preventive, red.] key rotation, not recoverable key rotation. That was going to come a while later, but it built on but it established the foundation of managing your

key state so that you'd have persistent identity.

Hard problem, shared governance. That was a hard problem. Spent a lot of time trying to solve that problem. 2017, the ledger wars. Okay, a lot of you fought in those wars. What happened is SSI was such a great concept that everybody wanted it. Every ledger wanted to have their own SSI Identity.

There was every time somebody would have an application that they wanted decentralized identity for, the very next thing was, "Okay, which ledger are we going to put it on?" All the people for all the ledgers would come to the table and say, "No, my ledger. No, my ledger."

And that's what it looked like for several years. And then GDPR came out in 2018 and that made it difficult to have identity on a ledger. The whole idea of persistent identity using a ledger is a problem for GDPR. The ledger wars, so the insight was that blockchain is too centralized for truly portable identity.

That was a really hard lesson to learn. It took several years of fighting the ledger wars to realize that. The way of KERI. Really the first paper that sort of opened the path was this one in 2018. The three R's of key management and one of those R's is Rotation and this is where I defined the concept of pre-rotation.

It's in that paper. It says, "How are we going to do key rotation? If we do it the normal way, where we use our signing key to rotate to a new signing key, if our signing key is compromised, then we lose control of our identity and that's all of blockchain, by the way.

And that's all of blockchain identity. If you lose control of your signing key, you lose control of your identity. That's kind of a problem. The other approach is have a rotation key that you use to rotate your signing key.

But then what happens if your rotation key gets compromised? Then you have another rotation key to rotate the rotation key that rotates the signing key. And I had several cryptographers says that's best practices. What happens when that one says two is usually enough. You only need two rotation keys. They'll never get there.

Good idea, but maybe we could do better. Pre-rotation is one-time use only rotation keys. You rotate your rotation key every time you rotate your signing keys. Pre-rotation and then that also introduced the idea of having a log of your key state.

Didn't call it a KEL event because we hadn't coined the term KERI, so it couldn't be a KERI event log, a KERI Key Event Log, but it was a log. Had multisig individual source and provenance data. The idea is make key management fault tolerant. I started down that path.

In 2018, a little bit later, paper called. A DID for everything. It introduced the idea of Verifiable Provenanced Graphs of Cryptographically Chained Data. That's the core idea that is now in the ACDC standard. Yeah.

In 2019, KERI version one white paper. Based on those, I said, "Okay, let's see if we can figure out a way to do it without a blockchain." And the idea is that you have a cryptographically verifiable append-only event log for each identifier.

There's no shared governance. You get rid of shared governance, you get rid of the ledger wars. Introduced CESR, key rotation, witnesses, deconstructed blockchain. It solves the hard problem of cross trust domain transfer.

Lesson learned: we don't need a blockchain. And I spent a couple of years arguing with people that said, "No, you have to have a blockchain. You can't do it with a Key Event Log." But now I hardly ever have anybody tell me that you need a blockchain for identity. Does anybody Is anybody arguing that still in the room?

You don't have to raise your hand. I took that paper on the road. I went to IIW several conferences, shared with everybody that I knew. Tell me what I'm doing wrong. Got lots of really good valuable feedback. There's a list of of acknowledgements in the back of the KERI V2 white paper because there were some really significant meaningful enhancements that came from that. One of those is the watcher network, CESR streaming, hierarchical delegation, layered threshold structures in more detail.

Basically, it solved all the hard problems of key management and so it was ready to go and that was 2020. And then the same year, while we were still refining the version 2 white paper came across the idea of quantum-secure DIDs.

I co-wrote a white paper on this. Using hashes to protect the pre-rotated keys means that we've pre-migrated along with the cryptography agility. We already have a story for post-quantum secure and Daniel [Hardman] and I, Daniel did most of the work, co-wrote a recent paper about the post-quantum position for KERI. If you're not familiar with that, go read that white paper. And then you can maybe do a reference for it.

2020, GLEIF Got religion. They said, this works. We don't like the ledger wars. I think that was your main motivation, Karla? Yes. The ledger wars. They didn't want to fight those anymore. so lasting peace from the ledger wars.

important insight. It solves organizational ID. Organizational identity is a new concept. Came from that because now you have a organizational Root of Trust. It's a bootstrap for any organizational ID problem. And now people in this room are starting to get that. I wrote a white paper "Universal Identifier Theory" just to extend the identity model, unified model for identifiers, bootstrap trust, be a reputation by reference, multi-valent key management, starting to talk about how we build infrastructure at scale.

That went into more detail. And then in 2020 we formed a task force at Trust over IP to start standardizing this stuff and ACDC became a thing, at that point. Authentic Chained Data Containers, and I think, We'll talk a lot about those in the next 2 days.

And then 2022, GLEIF vLEI in production. I won't go into most of you are familiar with that. Now where we're at? There was a story missing in KERI. KERI is about authentication, proving that you control an identifier. But, there's more to the story and had lots of conversations about how do you deal with privacy and confidentiality, and wrote the white paper, the SPAC white paper which is now a task force within Trust over IP working group at ToIP, and it's like draft version 3 or something, To solve, protected communications online.

It uses a three-layer tunnel, and the idea is, that if I have layered confidential contexts, then I can solve surveillance and privacy using layered confidential contexts. That's the basic idea. And what I mean by "It's exploitation, stupid, " what I'm saying is that the real question is, how do we protect people from being exploited, not how do we help people hide better. Hiding is not a very fun place to be. If you've ever tried to really hide, you don't have a life.

I know people that live off grid. They don't have a life. You want to be able to live with a life and not be exploited, and that means having control over the context of your data, not having a context, right? That's the main difference.

Do you have a context that you control, that you share data within, or do you not have any context? Most of the community says, "Don't have any context." That doesn't work very long because you live longer than a week, and you can maybe have no context for a week or a month or 3 months, but you're going to run out of places to hide after a while.

You need a fortress that you can hide in. You need multiple fortresses you can hide in and have context. 2023, Utah State Senate work begins. Wrote a white paper called Sustainable Privacy, talks more at length about the regulatory things.

Fiduciary data loyalty is in there. The idea is you need a comprehensive approach. Technology by itself won't solve the problem. You've got to do it .., That meant as a technologist, spending a lot of time in the room with regulators and politicians, which were not my favorite people, but now they become my favorite people. I'm looking for Chris Bramwell, but I don't see him.

He's late today. ToIP, we got our own working group, which helped speed along the things. We had some Cardano Foundation made a big investment in open-source software. Utah State started passing legislation. KERI Foundation we established last year [Dec 2024, red.]

No, that was that's wrong. That's 2024 12 02. It's in just the end of 2024, but it's really last year, to try to foster KERI throughout the world and get adoption. And then we're back; we're now at 2026. And we already did what happened in 2026. That's the history.

And you know what? I left a whole lot of stuff off. There's been a huge amount of work by people in this community pushing KERI along. But the core ideas, where they came from, and then the milestones that got us to where we are, I put on there.

What's the mission of the KERI Foundation? Our mission is to foster open infrastructure. What does that mean? It's the I in KERI. I had somebody ask me, "Why Is that a typo? You made the I in KERI red." I said, "No, I'm trying to emphasize that I stands for infrastructure."

KERI doesn't exist without infrastructure. It's not a system that works without infrastructure. And if you don't have infrastructure, people can't adopt it. They can't practically use it because nobody wants to build their own infrastructure.

KERI is a decentralized key management infrastructure. Right? It's It was designed to be web scalable so that we get an ultra low cost floor. Right? That's why blockchain ledgers were a problem. The idea is .. [tokonomics, red.], and I spent time working in the blockchain world, I think there was somebody here from the Wyoming delegation yesterday that used to work for Consensus when I worked for him.

But he had a beard and had a different lifestyle. He was all clean-shaven and had a suit on and I didn't recognize him and he said, "No, it's me, really. We worked together." And I go, "Oh." It took me a minute. But, the idea is Tokenomics. You build incentives into the governance structure so that people work together to build to to make your infrastructure alive.

Another term for that is a cooperative model. You can be in competition. You can make money. Everybody's making money. But, they're helping everybody else out to build the infrastructure that makes it all possible for everybody. And that's the idea.

We want to incentivize multiple vendors. No single vendor lock-in. We don't want that. That's bad for KERI. If you think about it, if you have pre-rotated private keys, you can and we have this in the code in the Keeper, you can reconstruct all of your keys if you're using HD keychain from any copy of your KEL because the HD path is in the KEL. It's the sequence number and the offset into the into the key list. You

don't need any other information. You actually don't have to worry about losing all of your information as long as one copy of one legitimate valid copy of a KEL exists somewhere in the world, you could reconstitute it.

We don't want wallet vendor lock-in. We want multiple wallets. We now have several wallets that are open-source and we're going to demo open-source stuff. The KERI Foundation team, like they've only been doing this since January, but you can ask them whether whether it was hurting their brain to try to learn KERI in 3 months.

Some of them had a head start. We've added another wallet. We've got a mobile wallet. You'll see some talks today of people that are building wallets and and infrastructure for KERI so that we have multiple vendors because adoption requires multiple vendors.

No big adoption wants to have a single source of the technology. They want to see they want to see multiple vendors. We have in KERI the concept of Non-Cooperative Key Custodianship. You can have key custodians that are custodians of your signing keys.

They can have the keys. And they can go away. They can hate you. They can lose them. And you can still reconstitute control over your identifier because you just have to retain your pre-rotated private keys. The major friction of people managing their keys and managing their signing infrastructure, you can rent out.

But you're not locked in because at any moment in time you can pull your pre-rotated keys out of air gap cold storage, publish your rotation, signing 'See-ya' to your custodian. And I think there have been lots of money lost in the blockchain world from key custodians who have .., what do they say? "Not your keys, not your coins." Not true in KERI.

Not your keys, still your identity! Not steal, still. Users benefit from witness pools that have no common mode failure sources. If your witnesses all come from the same vendor, the vendor is a common mode failure source. You really want to tell your customers, have witnesses supported by multiple vendors.

The watcher network, every watcher that is an honest watcher benefits from other watchers sharing what they see in duplicity. There's a huge incentive for watchers to cooperate. What we're doing for, what we're calling open infrastructure is "The Five Ws" or One-click KERI. The idea is people would come and say, "Okay, I want to use KERI. How do I do that?" Well Come to some meetings, go read these repos, go do this, build a bunch of infrastructure, then you can use KERI. I'm sure I see some people in here that have walked that path. That's not a really good story for adoption. The idea is we want to be able to tell somebody One click, it'll install KERI infrastructure for you. You got a wallet, you got witnesses, watchers, and now you can start to build stuff with it. That's the goal. The five Ws, wallet, witness, watcher, web, so it's all on the web, and wizard. And what wizard is the catch-all term. We want user interfaces that hide the complexity of key management from the users.

And KERI gives you the ability to do that. I mentioned some things that KERI has built in that allow you to do that. People building user interfaces need to understand those so they can build user interfaces to take advantage that. The other wizard is agents.

You want interfaces to agentic AI, and you want to control those interfaces, you want to use KERI to do it. This is what KERI infrastructure looks like. We've got a wallet on one side. Whoops, go back. [organizing slides] We have entity A has a wallet.

Entity B has a wallet. Entity A has their witnesses they control. They have watchers they control. Entity B has witnesses they control, watchers they control. There's a shared watcher network. And now entity A and entity B can communicate authentically.

But, they can't do it without the watcher network and the witnesses. (Now I can move on.) These are early screenshots from Mobile wallet that that you'll see demo-ed today. If you look at the sessions, KERI Foundation team is doing demos of the KERI stuff.

HealthKERI had developed Locksmith wallet, watchers and witnesses for their proprietary products. And earlier this year, they donated those to the KERI Foundation and allowed us to debrand them and rebrand them so that everybody that wants to build, can build on proven production-quality infrastructure. And we're hosting those repos and we have spun up we'll be providing some witness and watcher networks on our own servers so that people who want to test it out can test it out running in the cloud. That's one of the big things that we're doing for the community and and I hope that makes a difference.

There's two other pieces of infrastructure that are new; that we're building this year: Registrar and Observer. And the reason these are important is that they are how we do this concept what I call control over context.

I have issuances like SEDI issuances. They're ACDCs and you have a registrar and registries for like revocation. And you have observers they get bulk updates. The key here is the bulk update. Bulk update means that the observer and the issuer can't correlate back to the point of validation because that's the point of use the observer sees it but a change in state by the issuer can't be correlated forward because of the bulk update and the observer's not allowed and doesn't need to communicate to the issuer because they get a an update of all the information that they need to do verification. The verification happens at the observer, and observers can't collude between each others if you're using bulk issued credentials. We'll talk more detail about that.

This enables bulk issuance. It takes advantage of Sparse Merkle trees. That's some new infrastructure. Another new Use case, and these are in the spec, these are in the ACDC spec, are things called a user presentation registry.

One of the challenges in verifiable credential world is that if I have a proof that a holder has that they're the subject or the holder of a credential, and that proof gets stolen, and they steal the user's keys, then that user can be impersonated, and they can present that credential, and the user can't know that's happening or doesn't have any way to know it. But, we can use the same ...

(Go back.) We can use the same registrar mechanism so that a user in the ACDC, when they get it issued, can designate that their own registry Is going to keep track of blinded presentations. And that means that a verifier won't be able to verify the presentation unless the presentation itself is anchored in the registry. And this is all blinded so that it's not correlatable presentation to presentation. But, that means that issuees or users who want to be ultra secure can have fault-tolerant detection of compromise of their keys with regard to their own credentials. And they'll know, just like: you know that your keys for your KEL been compromised because you can watch your own witnesses. You can watch your own registry to see that somebody else is presenting your credentials and you didn't do it. Now now you can take recovery action.

What is KERI really? You've heard the phrase Keys at the Edge. Security First, Always. Minimally Sufficient Means. These are mottos, phrases; [another:] Signed Everything. Sort of ways to look at the elephant from the multiple blind men. [Don't assume one viewpoint captures the entire situation, red.]

All of these are features of KERI, but if you only thought of it as any one of these, it wouldn't be KERI. And lots of people have copied parts of KERI, but they've but not all of KERI. And if you don't have all of KERI, I guarantee you there are vulnerabilities that you'll be exposed to.

Heard a lot about that yesterday. No more half measures, guys. It's It's all KERI or no KERI. KERI is truly an identity revolution. There are concepts in KERI that don't exist anywhere else in the world in identity. And I'm going to give you a short I love creating acronyms because I'm old and if I don't have a mnemonic, I won't remember what I thought of yesterday. This is how I remember what I think from day to day.

Reputable Autonomic Pseudonymity. That That encapsulates KERI in the smallest set of words. Right? An identifier's a pseudonym. Is an autonomic pseudonym. What does autonomic mean? All of the features of true individual control over autonomic means self governing individual control over your identifier that's secure that's persistent that's perpetual all of those things right and reputable means it doesn't go away.

There's something to stick to a reputation isn't good unless it sticks to something it has to stick to an identity right? And then the other mnemonic Control over Confidential Context in which you use your identity. You want to control the context that you use the identity and that means that you want the context to be confidential. Now in the dictionary there are 20 definitions of the word privacy or private.

About a third of them are synonymous with confidential. When I say I'm sharing a secret with you and I say it's just private between us right? That's not private that's confidential right? Because there's other meanings of the word private that don't mean that.

The core private word means intimate means only you know it. Right? A private key is never shared. A shared secret isn't private. Shared data over the internet is never private in that meaning of the word private but it could be confidential.

You could make the information confidential. If you're going to interact with people you want to interact in confidential context and so you want to make those contacts you want to have control over that confidentiality. That means control over what you disclose control over what they do with it once you disclose it to them.

Here's the thing RAP-C3 and if you say it really fast it's a rhapsody* and that's KERI. [*an enthusiastic, emotionally expressive outpouring, red.] That's it. And we have a minute for questions. – I think it'd be useful to explain the difference between watchers and observers.

Watchers watch the KEL. That's all they watch. They're looking for duplicity in your KEL. Observers are watching the state of issuances that are anchored to the KEL. KEL? Via the TEL. It's a TEL observer. Yeah. – Why do they need to be separated?

Well, they have different purposes. Separation of concerns means you design protocols so that everything has Yes, it's layered. It's a layer, right? They depend on watchers, but they aren't watchers. Any more? I guess we're out of time.

No, one more? Four more minutes. Okay, I guess we we started a little bit late. More questions? – Please be gentle. I'm Moises. Be gentle. I'm a newbie at KERI. Watchers witnesses. You mentioned the term KEL or TEL. Key event log.

No, TEL. But TEL is transaction event log. – Since there's an exchange of two parties of information and keys, how do the watchers attach to confidential a confidential transaction? Watchers aren't part of a confidential transaction. Your key event log is public.

You're publishing your key state so everybody can verify any attestation or assertion you make relative to an identifier. And to do that, you sign the You sign things, and you have to verify the signatures, so you need to know the public key to verify the signature.

The Key Event Log publishes your public key and then the witnesses and the watchers, the witnesses make the public key state highly available and also give you a detection mechanism. And if you were you at the SEDI summit?

– No. Oh, okay. One of the things that happens is that you want to be able to detect that your keys have been compromised. The witnesses are a detection mechanism. Because if somebody wants to publish something on your KEL, they have to get your witnesses to witness it, which means you get to see it because what the witnesses witness is public. You can see that somebody's compromised your keys in order to publish to the witnesses. And then the watchers are watching the witnesses to make sure that the witnesses don't cheat.

– Where is it published, if not in ledger? A KEL is a ledger. It is a micro ledger controlled by one identifier. It's not a shared ledger. It's not a shared ledger. It's a non-shared ledger. But, technically under the hood, it is a blockchain.

– But, how is it made public? I guess that's my point. I probably can't answer that in this room. Okay. But, if you're here for the conference, you got 2 days of people who'll answer that for you. With that open-source infrastructure Okay.

Yes, that's right. Okay, thank you. One more Yes. – Maybe you'll answer this later, but witnesses are selected by the controller, and then the watchers are selected by the verifiers. What's your vision of how a controller actually selects a witness?

Usually they will hire .., They'll either, If they're technically savvy, they'll spin up their own witness, but most people won't do that. They will just rent witnesses from witness services, just like you would rent a cloud host for your website.

– Okay, do you ever see individuals being or employees being controllers, and would they have to go hire someone? The vLEI, if your work for company has a vLEI and you get issued, an ECR and OOR credential, you'll have a wallet and you'll select witnesses that you probably your company spins up for their employees to use.

– Okay, and then at the individual level, do you ever see grandma having to go select a witness? One of the things in the discussion that we've had with the state of Utah is how do you pay for disadvantaged populations to have infrastructure?

There would be that would be an entitlement that somebody would pay for your infrastructure. But, it's going to be the cheapest form of infrastructure, which is web infrastructure, not really expensive blockchain infrastructure.

– I'm curious to know more about the transactional friction there or the just the friction there. a really good conversation to have because another conversation. Yeah. – A real quick one on the registrar observer network, do you see benefit for the registrar putting random noise into the stream so that they can't tell it's a correlatable event on a revocation? Absolutely. The blinded state TEL, which is the thing that we use, not the current version 1 does not have that, allows you to do updates because they're blinded that don't change the state, they're just there to introduce to whiten the signal. You can as a registrar, you can whiten your signal.

You can issue random updates to to registries that don't change the state. That means that a state update doesn't necessarily mean the state changed. It just means you did a state update and most of them are ... You could create you could roll out 4 million registries for birth certificates and update them. And then when somebody's born, the fact that there's an update to a registry doesn't correlate to the fact that they were born and you then started using that registry for their birth certificate.

Thank you very much, Sam. Time's out. Running out. Do you want to introduce the next speaker? Yes, That's Daniel. Do I need to get to move my Please a big hand for Sam.

TALK 39

The Digital Identity Tradespace

Samuel Smith

KERI Conference April 2026 · talks · 37:05

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Samuel Smith maps KERI's digital identity tradespace, arguing that security-first design and keys-at-the-edge self-authentication reject weak privacy tradeoffs. He explains how bulk issuance, blindable registries, and sparse Merkle trees enable control over confidential contexts and precise disclosure scope within the ACDC stack built on witnesses and watchers.

My name is Sam Smith. If anybody can't remember my name, it's because you're more identity challenged than my parents thought I would be than they gave me that name. Anyway, that's all right. Great. I'm going to talk about the trade space of KERI, not in detail, but I'm going to talk about why we solve the privacy problem using bulk issuance instead of other mechanisms. And I'll give you the shorthand to the trade space.

KERI's policy is security first always, which means if we're ever in a situation where we're going to make a trade, and we're going to trade for weaker security or stronger something else, we're not going to make that trade. We're going to always trade for stronger security. We're not going to say, "Well, that security's okay.

We can do this, "because the trajectory of technology for security is that every weak trade for security is a bad trade. May not be today, but in your lifetime it'll be a bad trade. Let's try to be creative about solving the other problems without starting off by saying, "It's easier to solve those problems if we just throw security under the bus."

Actually say, "You know what? If my constraint is I'm not going to throw security under the bus, what's the best I can do? And is it good enough?" And that's really the the difference in the ethos. For those of With KERI, each AID controller is its own identity provider.

We use this phrase Keys-at-the-Edge. Each edge self-authenticates. What that means is there are no third-party IDPs, no relying parties, no trusted intermediaries. Every interaction is end-to-end or edge-to-edge, depending on your your mental model, right? If I have a network and I have people at the edge trying to get in, it's called an edge.

If I'm trying to communicate across a network, it's called an end, but they're the same thing from KERI's perspective. In my plenary this morning, I ended with this idea of Reputable Autonomic Pseudonymity in Control over Confidential Contexts.

The way we solve privacy is with this one, control over confidential contexts. And bulk issuance gives the user precise control over confidential contexts. They get to decide what the reach and scope of their disclosure are in terms of their correlatability.

We understand that every piece of information that we share over the internet is correlatable. There is no such thing as no correlation. Unlinkability is not the same as uncorrelatability. We called it Rhapsody [RAPC3]. Let's see how we use ACDCs for this.

ACDCs… And I think most of you are familiar, but I recognize some faces in here that maybe haven't actually sat through, something about ACDC. I'm going to give a couple of just short introductions cuz I've got a lot of detail to go through, and if if you thought this was going to be a high-level talk like the

plenary this morning, I'm really going to disappoint you.

This is going to go fast. Think of an ACDC as a graph fragment that has edges that point to other graph fragments. If you take graph fragments and connect their edges, you get a graph. That's the point. But, the edges are part of the ACDC.

And the reason for that is the authority and the authentication for each graph fragment includes its edges. You authenticate the attributes which are the node and its edges, and that is an ACDC. And now I can compose arbitrary data structures from graph fragments because all data can be graphed. Right?

And the edges (It's called a labeled property graph) because the edges have properties. The edges can be blinded. They can be private edges. They can have operators. They can be grouped together. They can be aggregated. They can be used in various ways. The edges have their own set of individual attributes that are metadata about the way you compose the information in a secure way.

And other approaches to composing information don't allow you to securely compose the information from multiple sources. They put it all in one data structure that under the hood is actually a graph, right? If I have a nested set of JSON blocks, it's a graph, but it's not composable in a secure way.

It's all or nothing. ACDC says, "No, that's a bad idea. We need secure composition." ACDCs are designed to solve problems where you need Perpetually Verifiable Issuances. Can you use an ACDC with a bare signature with a short crypto .. [inaudible] Absolutely. You don't have to anchor your ACDCs in a Transaction Event Log or a KEL to make them perpetually verifiable.

But we have to solve the perpetual verifiability problem. Let's solve the hard problem first. I use this sometimes that people are familiar with it. If I have a bunch of rocks, a bunch of gravel, and a bunch of sand, and water, and I want to get all of that into a jar, and get the most that I can get in there, what do I put in first?

The water, right? I put the water in, right? No, I put the rocks in first, then I put the gravel, then I put the sand, and then the water fills in the blanks. Ephemeral applications are the easy thing. Everybody knows how to do ephemeral application. Nobody solved the perpetual verifiable problem before KERI and ACDC in a decentralized way. Obviously, if you have a database and a central authority to the database, you can solve any problem you want. It's easy, right? But that's not the hard problem of decentralized data. The hard problem of decentralized data is how do I solve perpetual issuances without a centralized authority?

Right? Each fragment is separably authenticatable. That means the graph doesn't have a single source. Every node, every fragment can come from a different source. They can all come from one source or different ones. That allows me to do delegation, all sorts of things.

Right? I have Delegable, Attenuable, Aggrable, Revocable, Transferable Authority. – Can we break that down? – Okay. I haven't talked about aggregating, but notice that if I have multiple ones going in one thing, I've aggregated. If I have a chain, that's attenuable delegable. And transferable means that I can I can change who's in control.

And actually I'll talk about that in my plenary tomorrow. All right. What this allows us to do for verifiable credentials, which are all about authority, is that we can have high fidelity modeling of real world authority structures.

Not authority structures that are the best we can do because we have a broken model, which is current security, current OAuth OIDC. It's all broken models because it says, "This is what we can do with the technology we have. We have to use the secure sessions. We have to use encryption. We have to use things like Diffie-Hellman, TLS key exchanges, and we have to do ephemeral stuff because we don't know how to solve the perpetual verifiability problem. And then we'll make the rest of the world change their authority structures to match that."

And what ACDCs say from the beginning is, "No, no, that's not the way real world authority structures work. Real world authority structures have these properties." And it doesn't matter whether it's organizational identity or personal identity, right? If I'm a parent and I have a child and I want to do and I want to have control over what my child is allowed to do, then I've got a delegable attenuable authority structure.

Right? It looks like organizational identity. Those are the problems we're solving. This was supposed to be animated. I messed up. Okay. I'm not going to talk about this stuff. I put this on the slide because I'm going to talk about blind bulk issuance.

But a lot of people say because I spend a lot of time talking about about the other things that ACDCs do, that we actually don't support all of the least disclosure mechanisms that all the other credentials do. We just don't support a particular type of proof stuff because we feel that it makes an unfortunate security compromise. We follow the principle of least disclosure. We have something called gradual disclosure that allows us to progressive least disclosures so that we can negotiate how much information is disclosed in a particular context. Now, control over confidential context. That means I don't disclose any more information than I need to, to facilitate a transaction. And that transaction might mean negotiating over what I'm going to disclose.

I can have multiple interactions and transactions to negotiate the full set of disclosures and how I do those disclosures. We have something called compact disclosure, metadata disclosure, partial disclosure, nested partial disclosure, full disclosure, and selective disclosure. Those are all disclosure mechanisms that don't exist anywhere else except in ACDCs, except for selective disclosure. They have one mechanism for selective disclosure and nothing else. It means you can't model real-world authority structures when it comes to disclosing information. You need more capability. And then, you can do combinations of the above. And then, we have bulk issued instance disclosure, which I'm going to talk about today.

I'm only going to talk about one of the things on this list. And then, we have contractually protected disclosure, which you've heard about, right? There's several people that have mentioned that. You've seen demos of contractually protected disclosure. That is essential if we're going to have a comprehensive approach to privacy.

Comprehensive approach says you invert the economics. One of the ways you invert the economics is impose liability on the recipient of the disclosure. They don't get a safe harbor. They don't get to do things like de-identify the data and then they can do whatever they want with it. You attach strings to the data and those strings are enforceable which changes their their behavior. We're changing behavior of the recipient of the disclosure, not trying to minimize the disclosure and then hoping they can't use it against us. We actually say you can't use it against us and if you do, this is what we're going to do to you.

Right? We need to have a contractually protected disclosure and we need to have one that the recipient will agree to. And that's why we need graduated disclosure so that we can gradually move into a situation where we're mutually agree. And then we have blindable ACDC state registries. Those are a little bit a part of bulk issuance so I'll talk about those a little bit as well. That's the full set of mechanisms and there's a lot. And that's why often people look at ACDCs they go, "Oh, there's a lot there." Right?

But if we want to really do it right, we have to do all of those things. All right. Let's talk about infrastructure. Just to remember, we have controllers, they have witnesses and watchers. That's for KERI. That's key to it. ACDCs are built on top of KERI. When we talk about ACDCs, it's another protocol, it's another set of infrastructure, it's a bunch of new stuff that you have to learn about.

We have registrar and observer and this is what they look like. And then we have We can do bulk issuance with the registrar and observer if we do it right. And we use sparse Merkle trees to be able to do that. Sparse Merkle trees are technically a type of zero-knowledge proofs. We do use zero-knowledge proofs. A digital signature, a conventional digital signature, is technically a type of zero-knowledge proof.

Just nobody calls them that, but if you look at the math, a digital signature is disclosing that somebody who knew a private key and a nonce created the signature without disclosing either the private key or the nonce. It's a zero-knowledge proof. A Merkle tree, you can do an inclusion proof or an exclusion proof of entry in the tree without disclosing all of the tree. That's a zero-knowledge proof, by the book. Right? We take advantage of those. We just don't take advantage of the ones that have weak security because the crypto periods for perpetual verifiable issuances. But, if the application fits an ephemeral application, I have no problem with using zero-knowledge proof.

I use them all the time. I just don't call them that because most people don't call them that either. We also have something called a user presentation registry, which is also a new thing that nobody else does as far as I've never seen anybody do it, which allows presenters to detect compromise of their proofs that are in their credentials that they're presenting. They themselves will protect it from fraud, not just the issuer. What does an issuance registrar registry look like? Here's the block diagram. You have Let's see, can you see my cursor?

No. No. All right.

We have an issuer, an issuee, and we use that terminology, it's a little bit odd because we don't want to get confused about a data subject or a holder. We're really talking about the ACDC is targeted. We have untargeted and targeted ACDCs, and if it's targeted, then we're issuing it to some entity, and that's the issuee.

The issuee makes a presentation. The presentation can either be bare signed, which means it's ephemeral, or it can be anchored, which makes it so that they can detect if they've been compromised because a verifier will have to check to see if the presentation is anchored. This is an unanchored presentation.

The issuer creates a registry of the state of the ACDC, it hosted by a service called registrar, and then an observer, which is controlled by the verifier, the registrar is controlled by the issuer, notice that this models the witness watcher governance structure. We learned a lesson. We shouldn't have shared governance.

We have witnesses that are controlled by controllers, watchers that are controlled by verifiers. We split the verification between two places. We do the same thing with ACDCs. We have registrars controlled by the issuers, and observers controlled by the verifiers, and the ACDC state goes to the observer, the observer gets it from the registrar, the verifier checks the state against the presentation says, "that credential hasn't been revoked." For example, if I have dynamic state ACDCs. That's not a new concept.

Here's what it looks like from a a cryptographic point of view, the data point of view. What I have is I have the Key Event Log of the issuer. And I have events in the Key Event Log and I just show different types of events, incept, interact, rotate. But each event in the Key Event Log can have it sealed.

We call these anchoring seals. They basically are hashes. They're actually technically SAIDs of things I want to anchor. They are privacy preserving because I don't know what the hash is. It's a cryptographic hash. I can't infer what the hash is of unless I know what it is to begin with. But I can cryptically verify once it's disclosed to me what it is, that it's there, right? I have a cryptographic anchoring of a registry. A registry is essentially another type of log. We call them a Transaction Event Log. It is a hash- chained data structure.

But it's managing not key state but ACDC state. If my ACDC is dynamically revocable, I have issued and revoked, are the two states. And I have this new thing called No Change and that's because this is a blinded state TEL.

I can issue events to the registry that are blinded. I don't know what the state is. A third-party observer can't correlate the state just by looking at the registry. Only second parties that I disclose to, can figure out what the state is.

A given issuer can have a registry per ACDC that they issue to manage the state. Blinded state registry. These are the fields in an ACDC. The top-level fields. And the S, A, E and R, we call them sections because they typically are expanding the block of other things.

But we have attributes, edges, rules. I talked about edges. We have schema. We have a registry field that points to the registry that governs this ACDC. ACDCs don't have to have registries. If they're issued and they expire with time, they're not dynamically revocable, they don't need a registry. The registry field's missing.

Empty. I then have in my registry, I have a Registry Inception Event, which defines governance for the registry. What is governance? Well, it's the issuer AID. It says this registry is governed by this AID. And then it has a SAID like everything, has a serial sequence number, has a date time.

And then I can issue blinded updates. The blinded updates have a prior connection so that change age sequence number. And then has this BLID, which is this blinded ID. I don't know what the state is if I look at the registry. The registry's blind to me. Third party looks at it and says, "Okay, you got a registry got events in it. I don't know what ACDC it manages. I don't know what it's doing because all of that important information is blinded over here."

My blinded attribute block is where I get that information. I have this the BLID, I have a UUID, which means the BLID is universally unique. I call it a salty nonce. I have the SAID of the ACDC and then I have the state.

The only way I can get this is that the presenter, the issuee, unblinds it for me and gives it to me. And they only unblind it after for example, if I'm a verifier, I agree to the terms if it's contractually protected disclosure or after we negotiate what it is or what parts that that I'm going to disclose. And now they can at least verify that the ACDC is validly issued, but maybe the ACDC used selective disclosure to only disclose one field in the attribute block. But now I can at least look it up and say, "Yeah, it was professionally verifiable because I can trace it." This is what it looks like.

I've got the Key Event Log. I've got the events in the registry. I've got an incept. Incept is vaculous. It just creates the registry. Why is this useful? I don't want the creation of the registry to be correlatable. If I'm the state of Utah and I'm issuing birth certificates, and babies are born, I create a registry on their birthday, then I've correlated their birth date to the registry creation. That's a really bad thing to do. I want to be able to do is just create registries bulk that are just placeholders.

On day one of SEDI, the state of Utah will create enough registries for all the citizens of the population of the state of Utah for the next however many years they want to do that. And now they're only correlated to the date that's went into place.

Anybody that's born before that date, their registry is created on that date. Anybody born after that date, their registry is created on that date. A lot of mechanisms that people build for privacy don't actually consider the statistical correlatability of the meta data that they use when they build their systems. Every time I look at something I go is there metadata there?

Yes, there is. That metadata is inherently correlatable. It will defeat any mechanisms I use that are cryptographic because I just have to pay attention and I can correlate it. I have to build into the system anti-metadata correlation mechanism.

Bulk issuance, this is bulk issuance. That is a… Sorry. Oh, no. I got to go redo that. It's helpful to focus. All right. I got to go slower. 1 2 3 4. There we go. All right. What else is on here? I update… See all these updates?

They just have these BLIDs in them. The BLIDs are blinded. I don't know what the update is. Look over here. This one is an update but it doesn't have update my registry with nothing. If I wanted to make sure that like let's say that as the state once a day I create updates whenever I issue new birth certificates.

Now the update is correlatable to the birth certificate. I want to do is I want to have a bunch of registries that I just update randomly. That there's no correlatable information but the fact that I had an update. A lot of mechanisms for verifiable credential that have revocation registries, you look at the metadata when the transaction was posted to the to the registrar or the verifiable data registry, it means you can correlate out the information you're trying to protect. You don't actually have to have anything. You don't have to break anything. You just have to pay attention. You just have to be a good data scientist.

Since I started my life as an AI data scientist, I always think about those things first. I'm late comer to cryptography. I go, "Well, cryptographers don't think about data science. They don't know how to… They don't really think about statistics.

They don't know about much of the statistical stuff that they use. They just know that there's certain things that they can get away with." And I'm sorry for criticizing cryptographers. Anyway, so I wanted to do that and I keep hitting the wrong one. All right, one more time. Sorry about this.

I guess I could just exit out of the animation. All right. Here we go. It's on here. All right, so… No, this is the wrong… This is No, that won't work. That's a different one. All right, here we go. Go slow. It's sensitive. All right, so what I have here is the one that's actually issued. It's some update.

And then I can update it with the same state, but somebody looking at it doesn't know what the update means. This example actually came up with when I was talking to some people a few years ago for Swiss ID, and they looked at the existing KERI stuff and they said, "one of the problems they have with at the point in time pretty much all of the verifiable credential systems on the planet is that if you know what the state is at any point in time, and there's only two states, and there's an update, then you don't have to actually be disclosed the update. You can infer, cuz there's only two states, that the update changed the state and what the current state is.

The example is I apply for a job someplace, and I'm using a verifiable credential that's privacy-preserving, and I disclose that I'm an employee of a company, and then they don't hire me, but they put me on the list of saying, We want to keep track cuz we really like this guy, but he just won't… Timing wasn't quite right, right? But, since now they know the registry can then watch it. And if there's a new state in there, and they know that there's only two states issued and revoked, guess what? They know when I get fired.

Might be 5 years from now, and then they go, "Oh, he got fired. Let's see if he wants a job with us." I need to have my state updates be blinded so that there's no statistical correlatability. Has nothing to do with cryptographic correlatability or cryptographic linkability. It is I don't have any metadata that is statistically correlatable. That means I have to be able to update my state with blinded updates in a way that it… and the user can request this. The user can go back to the issuer and say, "Hey, you know what?

I did three presentations of my state. I want you to issue a new one so that decorrelates everything that I just did." Because now they can't go and look and see the latest state cuz the latest state is after the one they had, so they don't know what the state is.

They don't know if it went to revoked or it went to some other state. They just know they don't know it. And then when you revoke it, you can just keep updating it every once in a while, and they go, "Oh, well, it may must still be employed because his registry is still updating, right? He didn't abandon the registry."

It gives you perpetual protection from metadata correlation because the registry can live forever and protect you. All right, so… – Is it derived from something? – Yes, it's a blo… it's an HD process. What we're trying to do is minimize the key management. The issuer and the issuee exchange a salt.

And then you use the salt to derive the blinding factors. You don't have to have any interactions. What happens is the issuee can go look at the registry, see that there's a new event, and they can figure out what it is cuz they know the salt.

They can actually try all the combinations of state to figure out which one it is, without them having to have any ongoing interaction. I hate interactions. If you ever pay attention, I think interactive protocols are bad idea. I want non-interactive protocols in general. That's a non-interactive way to allow them to do it is to use a salt. All right.

Shared Blinded State Registry. Maybe you say registries are expensive. I don't want to have all these registries. Because the state's blinded and the blind includes the actual ACDC that the registry's for, there's no reason you can't share one registry across any number of ACDCs.

As an issuer I could use one registry for all of the credentials I issue. Now, that means that the registry might get heavyweight over time and I start to get what looks like blockchain bloat, so I may not want to do that if I'm issuing a lot, but I can certainly can decide what the granularity I want to manage that, right? That's the advantage of a blinded state registry.

Yes. – The salt? Exchange something like a symmetric key exchange? – Yes. But it's not for security, it's for privacy. KERI says it's okay to use shared secrets for confidentiality because every encrypted thing that you send somebody is has to use a shared secret. – Does that create a class of algorithm which has [inaudible] salt and symmetric salt exchange for which the cryptography community doesn't know what that is.

– I'm not sure I understand the question. – I'm just drawing a parallel between symmetric key exchange protocols and salt exchange protocols. And are they different? – No, it's just a secret. – How are they secure? Have they been looked at by cryptographers?

– Anything that you would exchange with Diffie-Hellman is just a secret. The security of the exchange is whatever Diffie-Hellman gives you. Diffie-Hellman doesn't care what what secret you're exchanging. – Hellman is anonymous by design, right? – Yeah.

It doesn't care. Basically you can use… your salt can be the shared key that the Diffie-Hellman gives you. That's the easy way to do it. – Okay, I think there's a man-in-the-middle attack, but I don't know. – Yeah. – Isn't that why you designed SPAC, though?

– No, but yes. SPAC solves a lot of other problems. You can certainly do an out-of-band salt exchange. And the reason you can do that is that when you do the identity assurance, if you're doing IAL level three identity assurance, it has to be an in-person thing, and then you can exchange the salt then, and it's out-of-band. And then you don't have to worry about Diffie-Hellman. But that's it.

It scales for SEDI because you can't get a SEDI credential unless you do the IAL at the start. And that means you can exchange the salt. – That's physical registration stuff? – Huh? – Your physical registration? – Yeah. And I'll talk about that. Yes, I have a slide on that.

Right, so let's talk about this infrastructure. I have registrars with registries. I have observers. All our observer is doing is it's keeping track of the registry state. It's asking the registrar, what are you registries you have? What state do you have? And now it has a local cache copy, and then when a verifier wants to verify something, he goes to his observer and says, "Tell me what the state of this ACDC is."

For this to work, you want to use bulk updates. That means you don't update instantaneously, you update like say once every 24 hours. Otherwise, if you update it instantly, then the observer could correlate a change to some registries and not others. But if all the registries, 100% of them update at the same clock time, because that's when they broadcast the bulk update, then there's [synchronization] in the time of update. Every time you do anything, if you want to make it so that it's not statistically correlatable, you have to do it in a bulk or herd privacy protected way. You can't do things one-on-one because the act of doing any kind of interaction one-on-one leaks correlatable metadata and it can't be defeated. You will correlate around whatever else you're doing because you leaked it, right? Even Tor, you can correlate around Tor because guess what metadata Tor leaks? Anybody know what metadata Tor leaks?

Tor uses variable packet size. Right? You get enough of that, you can figure out Tor. SPAC, TSP protocol, we have uniform size packets. There's no… Packet size can't be used for correlation and we have no packets that are just like vaculous blinded updates, so you can whiten your time of arrival, time of departure. And that's built into the protocol because otherwise, you do all of this work to cryptographically encrypt stuff and to try to protect your metadata from being correlated and then your protocol just leaks it all away. The basic idea is I want registries that bulk update the observers.

The observers can't infer anything because it's bulk updated, so you have herd privacy. What is the weakness here? The weakness here is that if an observer is malicious and the issuer is malicious, they can collude and the observer can say, "Did you issue this? Did you change the state of this particular point?" Right? You depend on repressing observers and issuers from correlating, but there's no phone home here.

The issuer can't force an observer because there's no phone home. Because all it needs is the information in the registry to do the verification. There's no built-in mechanism for the observer to trace anything in the system. It requires criminals to get together and form a criminal thing, and as long as you make it sufficiently legal, it might happen sometimes.

But then you are in the boat of, "You've got fraud in the system." And guess what? You can catch them because because they're colluding against the law and and you have repression. But yes, it doesn't make it impossible. Right? There's no technical thing that makes it impossible in this, but what it does is it means that there's no easy way for them to do it.

The first step in SEDI is an identity assurance process. What is identity assurance? Identity assurance solves one problem. It's called the cheap pseudonymity problem. Anybody can create a pseudonym, a cryptographic pseudonym, and prove control of creating an AID, right?

And as soon as they use the AID for something and they did it for bad and they say I'll just abandon and create a new one. Right? Nobody can trust an AID that anybody can create. Decentralized identity has a big problem. The problem is anybody can create as many AIDs as they want and then abandon them whenever they want to. Nobody can trust them, right? They're only good ephemerally. If I want them good for long-term, they need to have a reputation that says there's a reason why you can trust this.

What GLEIF does is provide with

TALK 40

The Future of KERI

Samuel Smith

KERI Conference April 2026 · talks · 1:03:31

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Samuel Smith looks ahead for the KERI suite, emphasizing ACDCs as perpetually verifiable graph fragments that model real-world authority, bridge trust domains, and resist recursive privilege escalation. He proposes authority reversion when delegations lapse and foresees broader applications for credentials, authorization, and fixing the broken internet.

I guess I talked too much yesterday, so I guess I'm gonna be swallowing the mic today. Anyway, like to thank everybody for showing up. Today we're going to talk about a little bit about the future. I did the past yesterday, the present. Today's a little bit about the future, a little bit about the past.

And I have a little teaser there: Saving the internet. The reason I started doing identity was I wanted to save the internet. I wanted to fix the broken internet. And I didn't realize I had to work on identity to do that. And I know other people probably knew that better than I did, but I didn't realize that. One of the things that the future of the KERI suite is about more and more applications that use ACDCs, right? And I gave my talk last night, talked a little bit about what an ACDC is. I just want to review it a little bit. An ACDC is this generic graph structure for representing data.

And when we represent data in graphs, we get all kinds of useful properties. And so we've got a vertex and edges. And we can use those to do all sorts of fun things. But the most important thing that ACDCs provide is perpetually verifiable issuances. That's the hard problem it solved, that wasn't solved before. And now that we have perpetually verifiable issuances, we can do all sorts of things we couldn't have done before.

Each fragment is separately authenticatable. That provides something that a normal graph data structure doesn't do. Lots of people have graph data structures, but the whole data structure source from the same source. Where with ACDCs, every every node in the graph can have a different source. You can use them for a lot of different things that you couldn't use a graph data structure for.

And these are the things that you can do with it. Delegable, attenuable, aggrable, revocable, transferable, authority. We have some talks today. Phil's going to be talking both Phils: Phil [Feairheller] and Phil Windley about authorizations. We're going to talk about some of the things that now that we have solved identity, [plus] we solved basic issuances, now we can start to do more complicated things with ACDCs and so I see the future people are going to find more and more things to use ACDCs for. The idea is: with ACDCs we can have high-fidelity modeling of real world authority structures. Can you guys hear me? I can't tell. Okay.

I can't hear myself because normally I talk really loud and my whole head resonates with my own sound. It sounds like nothing's happening. One of the things that we can do with ACDCs is we can bridge. KERI and ACDCs solve the cross trust domain transfer problem in an elegant way and that's a hard problem to solve.

Recursive Privilege Escalation Attacks (RPEA) are a Negative consequence of using identity systems that don't solve the cross trust domain transfer problem. Instead, what they do is just increase the size of the one trust domain because since you can't bridge trust domains, make your trust domain bigger. And that means now that you can do recursive privilege escalation attacks. If you keep your trust

domains small and bridge them properly, then you don't have recursive privilege escalation attacks. With KERI, we can solve that problem. The idea is that we have two trust domains and we use ACDCs to move data trust authority between those domains. (That was a cool effect, right?)

Perpetual identity leads to perpetual issuances means, and I'm going to steal Daniel Hardman's phrase, lossless trust transfer. We don't lose trust when we use carrying ACDCs to move trust across trust domains. If I have ephemeral identity, then I only get ephemeral in issuances. That means every time I do trust transfer, it's lossy. And that's what we're living with right now in the internet.

Now that I can have a lossless conveyor of trust, I can also use it to bridge ephemeral trust domains and the bridge doesn't lose anything. Normally when you take cables and connect them together, every connection is lossy. The longer the connections, the less signal you get. But if I have really high fidelity connections, then I lose it in other places. This is this allows us to take any application that's currently using the current stuff, and we can bridge it better. A lot of the discussions people have is well I want to use ACDC with whatever it is you know the some existing system. Well, it's okay for us to use it to bridge those existing systems as long as we understand that we don't corrupt the ACDCs with lossy stuff, right?

I already mentioned that the whole idea is we can remove this is like the bane [curse] of current security. This thing recursive privilege escalation attack vulnerabilities. It's it's horrible and we can mitigate that. When we talk about end-to-end in the the world we talk about edges but we also talk about ends and in KERI they're pretty much the same thing because each AID is its own identity provider we're using keys at the edge.

The edge is where authentication happens. There are no third party IdPs, no relying parties, no trusted intermediaries. We remove a whole layer of infrastructure. It just goes away. Now if somebody says you got to have it, you got to have a relying party. You got to have an OAuth server. Fine, Leave it in there. But don't use it to expand its trust domain. Go. You want to bridge these two trust domains. Don't bridge them with more lossy stuff.

Bridge them with KERI and ACDCs. In KERI an end-to-end and an edge to edge, they mean the same thing, right? The middle doesn't matter, the middle is just a store and forward network. That leads us to the next thing that's going to happen in KERI more and more and this is authorization.

What do I mean by that? Well I got this word down here. I love making anacronyms. I know somebody said that's problem for people to understand but you guys are the tech people. Edge Verifiable Access Control (EVAC). What does that mean?

Well, I'm defining authorization to access a resource is granted by the resource controller. I'm going to say there's some authority that has authority over a resource and if some other entity wants to get access to that resource that authority controller grants that access and that access comes through an authorization.

Each authority is itself an edge in KERI with respect to other authorities resources because it's all edge-to-edge end-to-end. Each authority can authorize access by any other edge to its resources via an ACDC. That's the whole point of delegable aggrable authority, is that you grant authority to somebody and now they can access your resources because you're the authority who can grant that access.

Each issuer in ACDC graph is potentially both a resource authority and an edge. Mutually authenticated edge accessor and resource authority prevent impersonation attacks. A lot of the impersonation attacks come because there's a a misconfiguration of how those two talk with respect to each other. You look at

how VPNs .., A common VPN problem is that I have a VPN and I give an edge access through the VPN, but the VPN isn't mutually authenticating in a strong way.

The edge thinks that it's talking to the VPN, but it's not. It's talking to an impersonation of the VPN. When it grants access, it thinks, "Oh, okay, I can now share my other credentials to what's ever on the other side of the VPN." But I'm not on the VPN I thought I was. Because I don't have strong mutual authentication. I've got a shared secret and that means now that I'm going to disclose to that other end my credentials to get further into the system and I rinse and repeat, right? But if we have strong mutual authentication at every edge, then there is no recursive privilege escalation attack.

We basically do this with ACDCs. I'm going to show you how we can use an ACDC. This is a proposal. I haven't written a white paper on it yet. I started writing a white paper on it, but I haven't finished published it yet of how we can just repurpose ACDCs as generic authorizations.

If you so there those of you who aren't familiar with an ACDC, these are the top level fields: version, type, its SAID, a UUID, the issuer, the ACDC, the registry for the ACDC state, the schema section, attribute, edge, and rule section. There's two registries here: There's an Issuance Registry and we can now add a Usage Registry and the way we manage that is in the attribute section. The attribute section itself has a SAID, a UUID the issue AID and it also can have a registry identifier and then a resource map. The resource map over here looks like object capabilities. You basically have a route and then a set of capabilities.

It's a list of things like read, write, execute, whatever it is. Standard standard access control. All we do is we just augment ACDCs to have a resource list and we can implement any sort of authorization system. But what we get are some properties that are difficult to get in other ways, without ACDCs.

This stuff is actually in the spec but I'm pretty sure almost nobody's read it. Right here: the issuee ID people know about, but in the spec we actually have a registry identifier defined; but this registry identifier is different from that registry identifier. This is the issuer registry. This is the issuee registry.

Why does the issuee want a registry? [question gets inaudible response] That's right. If the issue wantes to know if they .., here's the use case: somebody gave me an authorization let's say this is my authorization I want to know if my authorization's been compromised now in ACDC if I want to present it requires at the very least signing it so compromise looks like an a compromise of my signing infrastructure now if it's a token, compromise looks like somebody just steals the token. It's an easy compromise. It's much harder to steal my private keys. But let's say someone has stolen my private keys.

They can then present my authorization someplace and I don't know that they've stole my private keys unless I require that a verifier check that the presentation is anchored in my registry. Every time I do a presentation I anchor in the registry.

The verifier checks that the issuer issued it by checking the issuer registry but checks that the presenter actually says "I want ultra security when I present. Don't verify the presentation unless I anchor the presentation." Makes the presentation less ephemeral. But it also means now that if I have really strong security over presentations and if somebody steals my private keys as a as a presenter, I know that they stole them because they the only way that they can gain access is to anchor it in my registry that I control. That allows me to detect compromise of my credentials.

Here's showing what I just said. Here's the issuance registry, with a presentation. The issuee just signs the presentation. The verifier checks, it checks its observer, checks the registrar, controlled by the issuer says "okay it's good. The ACDC is fine."

This is what it looks like over here in terms of the Key Event Log of the issuer and then the registry. But then we have the registrars and the observers. But if I'm doing a usage registry, what does it look like? The presentation itself gets anchored in a in a registry that's controlled by the issuee. This one is controlled .., this registrar is the issuee's registrar and now the verifier checks here. The verifier would have to check two registries; has to check the issuers's registry and the presenter's registry.

But that makes it so that both the issuer and the user can detect if either one has been compromised and the verifier knows it. That's what it looks like here. Basically any presentation gets anchored in the issuee's registry and you can see the state of the presentation and you can share one registry for all your presentations because it's one controller of the registry.

This is a another difficult problem in authorizations and that's when you have delegation and we didn't solve this problem from the vLEI in this explicit way, but this is in the spec It solves the problem of chained authorizations in a delegation chain where the delegate issues an authorization and then the delegate's authority to issue the authorization gets revoked. The question is: What happens when a delegate's authority or a delegatee's authority to issue authorization is revoked. What is the status as any authorizations or entitlements that it issued before its authority was revoked? Are they revoked?

That would be really bad, in some cases, Let's say I'm a company and I issue a purchase order. My purchases agent while he works for me issues a purchase order. Purchase order has a 90-day expiration on it. It's valid for 90 days, right?

And then I fire the guy because he wasn't showing up for work, right? Do I have to go through every purchase order that he signed off on and reissue it? If it's on paper, everybody assumes that it's still valid, right? But if it's electronic, if it's digital of an ACDC and it's got in a revocation registry, I could revoke all of them and then reissue them. But maybe I'm okay with what he issued. I The reason he got fired wasn't because he was misuing purchase orders. He got fired because he violated HR or something, right? Everything that he did officially is still good. But he doesn't work for me anymore. But if I revoke his authority, I break the chain of authority. And if I don't have any other mechanism, then I am forced to reissue everything that he did.

That's a hard problem of delegation. And we don't solve these these problems in general in decentralized identity. If I have a database, I can solve any problem. If I have a centralized database, all of these problems are easy. It's just software. But if it's a distributed decentralized [database], I have to solve it cryptographically. I have to figure out how I'm going to do it in a decentralized way. How do we do that?

Here's a way that we can do that. (I just I said that: When the chain is broken.) This is how we do it. With something called a Bound Blinded ACDC Register. What did I put there? "Reco"? That should have been Reversion.

I misspelled that. This is how it works. We have this situation here. We have an issuee AID. The issuee is the issuer of another ACDC. We have a chained set of ACDCs. There's an edge pointing back from the delegated one back to the delegating one. Okay.

That's the setup. Now we have the issuer uses this bound blinded state registry and the thing that it adds is the sequence number and the SAID of the key state of the issuee who is also the issuer. Right? The guy in the middle is both an issuer and an issue. He's an issuee of his delegator and he gets to issue stuff himself.

Right. It's a delegated chain but what we're doing is we're binding .., we're binding the key state of the issuee at the time we give him authority. Right here, we issue to him authority and we bind his key state. And then when we revoke his authority we also bind his key state. What that says: between two and five He has the authority to issue things and here's his registry. He can issue things in his registry. In this list right here, he can do things to this other thing and they all work. But number six here, he doesn't have authority anymore.

Even though he's got an update event, that update event, the bottom one's not not verifiable because he doesn't have authority to change that. We have a problem. The only way that this works is if the issuances over here are all time expired.

[Sam sorts out the tools for presenting] What happens is that the type of ACDC that he can issue here have to have some sort of timeout because they themselves can't be revocable because we have an ambiguity about what we do about it when we revoke his authority. But as long as it times out or something then it'll just self-revoke.

How do we solve that problem? That's the harder problem. This is not in the spec. This is actually a proposal for the spec. We add another event. The way we can solve it .., it's not written up in the spec but this is an easy modification of the spec, is that because there's an edge right here that points back to the issuing authority we can do a reversion of authority.

Let me give the example. Let's say that I'm a lieutenant of a company and I take one of the members of the company and I give him an order and I say you're going to go over there, right? And he goes over there, right? He has authority to do it. I told him to do something. Let's say that I I resend that order. He can't do that anymore.

What do I do at that point? The authority to do that still rests in me, the lieutenant. I can go over there and do it or I can authorize whatever it is myself. The authority always reverts to me as the higher authority. What we do is that when a subordinate or a delegate loses authority because we've revoked it, any outstanding issuances that were validly issued that we want to dynamically revoke the revocation now becomes the responsibility of the delegator. It just reverts back to the delegator. And because we have an edge, we know who the delegator is. All we have to do is make a policy change that says in this case where I revoked .., his authority right here.

You see this one over here is issued, I can't revoke because he's been revoked over here in number five. But guess who can revoke it? See the line that comes back here? He can revoke it. And a verifier just needs to know that's the policy.

It just looks it up and says, "Oh, okay. The approximate delegate doesn't have authority because it's his authority has been revoked. The chain's broken." But any subsequent actions on that ACDC are still the responsibility of the delegator. The delegator can revoke it. Now we can have dynamically revocable credentials and break the chain and still dynamically revoke them.

Since there's always edges, you can the reversion of authority can go all the way up to the top of the chain. You can break the leak as many times you want and you just keep you just rinse and repeat all you just ascend the chain. Now you have an elegant solution for dynamically revocable delegated

credentials all the way down. You can break the chain any number of times and still control those credentials dynamically.

We have solved a really hard problem that in ephemeral authorizations isn't a problem because no authorizations last long enough that you care. But if I'm doing perpetually issued authorizations that last indefinitely unless they're revoked and I'm doing a chain, I need to solve that problem.

That's the solution. You revert the authority to the delegator when the delegates authority is revoked and that allows you to continue to control those issuances. Now we can do it even better. This requires something not in the spec.

This is a proposal to add in the spec and that's just adding one additional one additional update. It's called a transfer event. If we look over here .., Oh, number three. Number three is transfer event. We have an inception event and a transfer event. Those are analogous to an inception event and a rotation event in a Key Event Log. In a transaction event log, we now have meta control events. We always had the inception event which establishes the who's the authority over the registry. Now we have a transfer event that says 'we can transfer authority to a different AID.' That allows us to do an authority transfer. It's not just reversion. We have a reversion of control. The reversion allows us to issue a transfer event. Now we have a new delegatee, that now is a control of all those credentials that are controlled by that registry. We could do a seamless transfer all the way down at every level. And all we have to do is add one event. And the main thing in this event is just that it's got It's got the issuer. It's got the new issuer AID, not the issuer back here. The issuer at the end.

Yeah. – [inaudible question] The cryptographic context is that every event in all of these registries is anchored to somebody's Key Event Log. That's perpetually verifiable because all of the security mechanisms that protect the key event log protect everything anchored to the Key Event Log.

– Right? Post transfer you want to minimize the number of nodes you have to recurse through in order to get your verifications to work. You're not minimizing them, you're not changing the number because you have a chain. You always have to verify the whole chain.

You didn't increase the length of the chain, you just switched a branch in the chain. – Right. There's a dead branch in the chain because the employee got fired. Right. – And I don't want to maintain his logs forever. Well, his K logs, right?

No, you don't have to maintain his key event logs. You do have to maintain his Key Event Logs for as long as those issuances are valid because that's the only way you can verify the original issuance. That was my question.

Yeah. You wouldn't do it for you, you wouldn't do it for something that was .., This would allow something that's longer lived .., Let's say an employee issues a purchase order that you want to be dynamic revocable, but you're not going to have it sit out there for more than a year, right? You could transfer to a new employee. It still lasts a year, but you could still revoke it if you found out there was something wrong with it. If you're the state of Are we out of time? Oh, you're just asking questions. Okay. Let me let me go ahead. I guess – I was just curious because you nailed it with key rotation can already transfer control. Is this analogous to a sort of [key rotation] The difference here is that key rotation is heavy weight transfer because you're talking about control of the AIDs.

– Yep. Right. In the case of an employee leaving, you would at very least have to have a multi-sig so that multiple employees controlled the AID and one of them rotated out. This allows you to avoid any of those issues – because you're doing it on the registry level.

You're doing on the registry level instead of the KEL level. It's a much lighter weight, easier to manage situation for doing things at scale. – What I'm wondering though is that it could this be a point of failure if someone were to key compromise and then Transfer a registry toward a malicious controller out.

Anytime you compromise keys, you can do something malicious, right? But you can't do it without being detected, right? Which means you recover from it, – Right. That's the point. – Okay. I'm just trying to understand if it opens up potentially any like vulnerabilities.

Not any new ones. – Okay, good. This now means that we can maintain control over issuances, out at the edge of our graph, authorizations, and they can survive for long periods of time and we have dynamic control over them and we can change control over them dynamically.

Not all issuances benefit from that. There's some you don't want to do that for. But in the cases where you have really long-lived things, like guardianship. This is an example of guardianship. You have a child who has parents that are guardians of the [child, red.] and they get divorced. Who's the new guardian?

Do you want to reissue the birth certificate to the child or do you want to transfer control of the birth certificate to the new guardian? You want to transfer control. You don't want to reissue the birth certificate and lots of authority structures in corporations that have to do with things like 'who has control over the domain name registrar for the company?'

Do you want to reissue all your domain names every time that person moves? Or do you want to have some mechanism that you can transfer control? And do you want that mechanism to be a multisig group, where you have a group of IT people that all have a threshold control? You could do it that way or you could do it this way. In some cases, this is going to be more scalable because it's not requiring tight coordination for key management.

This would be in version 1.1 of the ACDC spec. My proposal would be to add another event [transfer event]. Basically we're just adding one event so we can do control transfer on a registry. And I didn't think of it a year ago when I wrote up the registry, but as I was thinking about this problem some more, I said, "No, no, we probably should add that one." And I actually changed it three or four times because I kept trying to decide what I needed. And I realized that the only thing it needs to have is the issuers' AID. Everything else is the same as the inception event.

It just declares a new issuer. It's the event type is the only thing that changes. Any other questions before I move on? Yeah. – Does vLEI authority fit into this somehow? Right now vLEI don't solve these problems. With the vLEI, what you have right now is a grace period. When you're issuing the QVI loses its authority, all of the vLEIs they issue are still valid for a grace period. Karla [mcKenna] wants to answer.

Go ahead, Karla. Oh, get the mic.. – Yes. And I knew that this was coming Because Kevin told me that you were working on it. We have been looking at a way in order to be able not to have to go back to the legal entities.

[Case:] Your QVI is leaving the system, you're going to have to hire a new QVI and recreate everything that you have. This way we can transfer control to GLEIF and then as the legal entities hire their new QVIs we can transfer control again to the new.

That's what happens: GLEIF now becomes the authority, not the QVI, and then you don't have to make the transfer control be forever. It can just now be a dynamic time period to give you a grace period to find a new QVI and reissue. But now the grace period now is dynamic instead of fixed, right? Right now

it's fixed to what? 90 days.

– Okay. Sam, I'm gonna give you a practical example: I work for a civil engineering firm. I have my engineer stamp. I've stamped the bridge as being ready to build or meeting the spec. I leave the company. What is the company going to do? I'm trying to see that how that hierarchy would work, especially considering that in the traditional world, we just file those blueprints and that engineer stamp is always on there and we consider it valid.

Right now the engineering stamp isn't dynamically revocable. There's no mechanism to do that. This only applies to engineering stamps that you already need to dynamically revoke. Because it's very difficult to do dynamic revocation almost no real world issuances have dynamic revocation. But because now we're doing this stuff cryptographically, all of a sudden all sorts of use cases where we want them to be dynamic revocable but we never thought it possible so we didn't and we just lived with it now become good applications.

We're talking about using ACDCs in new ways. You can refactor multiple licensing regimes to have dynamically revokable licenses and it's practically workable. There should be some business opportunities there. – The basic idea of this transfer registry that you create with that you're delegating or changing transferring control over a set of identifiers to a new identifier. It creates a registry.

No, you're changing control of the registry. One identifier, the registry's identifier. That's what you're changing control of. I misspoke there. That's right. Okay. Just the registry that you might have already issued many credentials from and so is essentially .., Or one credential that has state changes and you don't want to reissue that credential. You want the credential to continue to have viability. You just want to change who has the authority to change the state of the credential.

That's all you're doing. – Got it. And you're doing that .., I want to point out something out here. See this event right here, the transfer event. See it has two arrows going to it. That because a validator first has to verify that the reverting authority approved it, and has to verify that the new authority approved it.

And then once that happens, then they they don't care because it's just the new authority. All right. Any other questions? All right we'll go to the next one. – Do you need [to have] reversion in place in order to have a properly functional transfer registry Of this type? (– yes) There's another type of transfer registry that we haven't defined yet that I have an outstanding issue on KERIpy [for], that's different from this. – Okay.

This is reversion for a hierarchy of authorizations and delegation, not a replacement for asset transfer. Ned [Smith] is going to give a talk on secure asset transfer later today. – Can I ask one clarifying question? Yes.

Previously earlier in this talk, I thought there was another methodology where it was like 2 through 4 was valid. That's a different style, where it's a different style of dealing with the same problem? No, its the same thing, It's the same thing. It's just this says once you're invalid, what do you do with the ACDC of the controlling authority?

If nobody's in charge of it? You can revert back to the delegator and now the delegator is in control. But that now imposes a burden on the delegator to have to make a controller from that point on. – Right. I just thought I understood previously like to answer his question about that use case where you could potentially just keep a certain subset of the issuances of the credentials previously.

That works as long as the delegated credentials expire on their own. If they're dynamically revocable and you break the chain, you can't revoke them after you broken the chain. It works for conventional issuances that time out. If I have a food handlers permit that times out in a year, I can fire the employee that issued it and it's still valid, but it'll expire in a year. But if I have something that I want to be able to revoke any point in time, then I can't break the chain and still I have to define what that means.

I could say "it means X, " right? But we want to make it so that we can actually verify and enforce it cryptographically. (I sound like somebody that doesn't have a voice. All right.) I'm going to move on because we're going to run out of time. I have a couple more slides. The other thing we can do is that EVAC can also mean edge verifiable agentic control.

Now, here's the thing. An agent is an authorized delegate of a natural person, a legal entity, right? An agent can be another natural personal legal entity. It can be a piece of software. It can be an AI. The term agent means .., nothing in anything, that I showed you before, is any different if it's a person or an AI that's in the delegation chain. It all works the same. Now if you're worried about whether your agentic AI has permission to do X, you can cryptographically verify that it had permission to do X and you can revoke its permission to do X and any verifier can verify that. It doesn't matter.

You just give the agent an AID, a delegated AID, right? That's the original meaning of the word agent. When see people say agentic AI, they're just software agents that are a little bit smarter than software agents used to be, right? The ACDC mechanisms are exactly the same.

They don't change. If I want to do EVAC, I do evac. You may want to have regulations require attributes in your chain that says 'this agent is a person versus an AI,' you might want to disclose that. But as far as verifying their authority and what they do with it, ACDCs accommodated without any changes. – One of the other problems with delegation is controlling the recursion of delegation.

There's a bunch of weird stuff in X.509 for how to do that. I'm just wondering what mechanisms exist here to manage the the degree of delegation that a delegate .., What do you mean by recursion? Like how many levels .., – A delegate can delegate to another delegate.

In KERI When you have ACDCs they're edge operators and so you can have an edge operator that says 'do not delegate' or you can have a configuration for a delegated AID that says 'do not delegate.' You can say how many layers down you can have delegation. Right? But I'm familiar with that.

Do you have a whole policy engine for how far you delegate? Actually you can have a whole policy engine with ACDCs because we have edge operators. You can make it as complicated as you want or as simple as you want. ACDCs don't care, it's up to the user application.

The cryptographic verifiability is: "I have an edge. The edge has properties." The properties can define whether that edge allows further delegation or not. Does that make sense? Did you guys understand? Yeah. Okay. All right.

Safety. I'm going to change topics. I have a couple of questions. All right. Back here. – I guess it's the same with natural people and organizations to natural people. How would you enforce if possible attenuation such that subsequent delegates or delegatees do not acquire more power than the .., There's two governance mechanisms that we use in ACDCs for that. One is called an ecosystem governance framework okay so for the vLEI there are 200 pages of documents that say 'this is what you can do with these ACDCs.'

They define the delegation, what you can delegate, how much you can delegate, how many layers of delegation, what the ACDCs are, who does that.., right? That doesn't have to be encoded into the ACDCs themselves because the schema for the ACDC which is an immutable schema is defined to be the schema and then you look up the ecosystem governance framework for that schema and you go "Oh, this is how I use these guys In real world you're probably going to have quite fixed well-rigidly defined ecosystem governance frameworks because that's how the real world works. We don't go around and say, "Hey I want to be a real estate agent." 'Agent,' right? Hear the word 'agent.' What are the rules for real estate agents? How can they operate? What things can - and can't they do? There's a whole body of regulation about it. We don't have to .., we just have to say "Well, we're going to make it so they can verify that they're an agent. We can reference all the regulations that apply to them as an agent." Now, we've modeled the real world authority that already exists, but we just made it so that they can do the verification quickly and securely without impersonation or fraud. We don't have to build all of that rulemaking stuff if we don't have to.

There's no reason to do that. But if we want to add some of that [rulemaking, red.] because it's practical, then we can encode it in the edge logic with edge operators. But we want to minimize that. What tends to happen is people try to model all of the stuff in the world, real world in data, and then they get data that is ambiguous. They get ontologies that are broken.

Yes. – We've been toying with with this delegation structure a bit more from an application perspective, right? An agentic AI is not one thing. It's a spectrum, right? In other words, there are rule based on the lower end and you have loose guidelines and we want to see if the agent that is taking control and delegating to further agents did it right in the context of those guidelines. Right.

It is a loosely defined structure? Yeah. (Microphone replacement)

Trying to build other systems, you're welcome to do that. I'm just saying that we can't solve the problem that ACDC solves right now. Whatever you're doing isn't solving that problem because if you're not using ACDCs to solve it and you're not doing this, then I know what you're not able to do. But you can certainly do other things. Does that make sense? I'm not prescribing for you how you want to solve Agentic AI.

I'm just saying that if you want to solve the act of modelling the authority chain in a cryptographically verifiable way and the elements of that chain happen to be agents, it doesn't change it doesn't change the mechanism. The mechanism works just as well either way, right? (A question; waiting for the mic to arrive) – The guardrail is that you want least privilege built into the system so that there isn't an opportunity for a delegate to introduce more privilege than what was delegated to him originally.

And is the enforcement mechanism on the shoulders of the verifier (Yes!) or there's something intrinsic in the structure that prevents that? No, because the only person that enforces it is the verifier. Otherwise, what you're going to have to have is a policy engine that the verifier asks to enforce it for them.

Right. You've just moved the enforcement someplace else, but it's but it's still .. [the verifier's responsibility, red.] – That's the part that I think a lot of the industry will have difficulty accepting. I'm sure they will. And Phil [Windley, red.], I think you're going to talk about some of that, right? Or he's writing a book on it. I'm talking to you, Phil Windley.

I don't think he heard me. Hey, Phil. Phil. Oh, he asked a question and I said, "You're writing a book on Authorization Policy for Agentic .." Okay. All right. Later. It's all right. I – Sorry. What was the question? I apologize.

I just explained how we could use ACDCs to do delegate authority, and Ned [Smith] asked, "Least privilege for the delegatee to be able to do stuff. How do you enforce that? And I said, right now in a decentralized system, the verifier can enforce it or you could have a policy engine that the verifier refers to, that enforces it, but that's [your option, red.]

– Yeah so basically I think there are two big avenues of how you would do this with delegation. One is delegation as data. The other one is delegation as policy where you essentially write new policies for the delegation instead of relying on a broad set of policies. In delegation as data ACDCs especially with what you've described today are really nice because you can put all of the delegation in the ACDC the credential and then, as you've pointed out, you can keep that from expanding, you can revoke it, you can revoke the whole chain when necessary, I think that's the broad approach, right, as you write policies that interpret the data in the delegation credential.

Yeah. And to Ned's point, that's not how most people do it now. They may not like it, which is a valid point. I'm going to change topics for the last five minutes. The internet is a dangerous place. What we want, what I would like, I hope some of you want this, is a digital refuge An online shelter, an internet sanctuary, a surveillance asylum.

We're protected from all of these things when we go online. When we use the internet, we want to be able to use the internet and be safe If we think about those .., "Safe" is to require accountability of the participants.

If they're not accountable, it won't be safe. If you can act in any way you want and there's no repercussions, there's no consequences, it's going to incentivize bad behavior. And that's what the internet is. That's what the internet is, it isn't a safe place because there's no accountability on the internet. It never came with it in the first place. And everybody assumed that well people will just act the way they normally do. I have a really good example of this. My daughter who you all saw sing, right?

Yeah. She has a pretty substantial internet presence, lots of followers, and there's a lot of trolls that will go onto her Instagram or TikTok [and do] all kinds of really horrible things. You know what my wife likes to do, who you all met, is when they're do really horrible things and they say stuff like that occasionally, she will go research them and they're always using some alias. They think they have anonymity. She'll go figure out who they are and then she'll go ask them, "Why did you say this about my daughter?"

And they will SCRAMble for the hills. It's just like, "Wait a minute. Who I am? That I said this vulgar thing and sometimes they're just average people in your neighborhood who act like they don't do this. Why do they do it on the internet? Because they can get away with it. And what she always likes to say to them is, "If you met my daughter in person, would you say that to her in person?"

When they go, "Of course not." Well, then why do you say it online? And the difference is there's no accountability online. If you want safety, you have to have accountability. Here's the other thing that people don't believe me when I say this. Privacy requires accountability. People think privacy requires being able to hide. You can't have privacy if without accountability.

Here's why. Second parties to repeated interactions can correlate the first parties. If I interact with you, even though we're both anonymous, the interaction themselves is enough to correlate. Guess what? I can't interact and have privacy.

I can't. Any interaction system that does not enable first parties to hold second parties accountable is fundamentally incompatible with privacy. You can't get there unless you start with accountability. And you don't get privacy by hiding, you get privacy by interacting with digital refuges Surveillance asylums

where the participants hold each other accountable and you can trust that they'll act in the way you want and everybody else is excluded.

That's the whole point. If I want to interact with him or her, right, and I can hold him accountable, then I have a trusted interaction. If I can't hold them accountable, I can't trust it. But what I want to do is I want to have that interaction and not have anybody else surveil it.

And that's what we really want. We want contextualized confidentiality with accountability. And how do we do that? There's a protocol called the Trust Spanning Protocol that, combined with the KERI Suite, allows us to create and control safe relationships where interactions protected from surveillance by third parties and second parties are held accountable for any information shared with or by them.

My breakout talk today will be talking about the TSP protocol and how we can do that. Keep going. We only got a minute. Okay. You can have some of my time. All right. Okay. When I do this analysis, I always use this model.

They say "All models are wrong, but some models are useful." I find this a very useful model. Any interaction on the internet involves the first party and second party. I can encrypt the communications between the first party and the second party so that the content isn't observable by any other party.

But if I'm using the internet, what I can't protect is the metadata of the fact that I sent data over the internet. It has to be routable. Routing tables are public. Routing information is subpoenable. All of that metadata is not protected.

The Supreme Court even said so. Right? Reasonable expectation of privacy is limited to content, not non-content metadata. Non-content metadata is how you get from first party to second party. If I can correlate out from the metadata who the first party and second party are, I've already defeated privacy in general. There's no privacy on the internet ever. Never has been, never will be. If you use that definition, right? But what I can do is I have privacy where I can make it so that I that they don't know who the first party and the second party are because the first party and second party they see aren't the first party and second party that matter.

The basic idea is: I don't use clandestine means to get privacy. I use covert means and covert means you have a cover identity that basically provides a layer of protection to some activity that you're doing that isn't observable. Clandestine just means you're not observable. But any information that's broadcast, is eventually observable. Clandestine can only be used for ephemeral applications. And I spent 20 years and 20 millions of dollars of federal money building clandestine surveillance mechanisms and at some point we said, "Oh, we got to make them covert because we need Persistent Lliteral Surveillance;" that was the name of the program. And when we went from ephemeral to persistent surveillance, we had to change our concept of operations.

And that's what I'm saying is we have to change that. This is the model. And then and to get from first party, second party, I have to have intermediaries. The intermediaries can surveil stuff that a third party can't. Your ISP is an intermediary.

Given that model, these are the two principles that we use. One is end-to-end verifiability which tells me that I know who the parties are. The parties know each other. If I have KERI then I get ambient verifiability any day anywhere anytime by anybody of who it came from. I can hold the source accountable right; non-repudiation.

Now the dual of that, is what I call 'Only At End Viewability'. I need to be able to send information to somebody and have reasonable confidence that they're the only ones that got it. Because the only way I can hold the second party accountable is if I know they're the leaker of data.

Because if I send encrypted data to a second party that they can decrypt, they have the data. I no longer have any ability to restrict where that data goes. There's no technological means that allows me to protect that data once the second party has it. I have to trust the second party. I have no privacy of any information I share with the second party if the second party can share it with somebody else. There's no longer technology here. It's all legal, regulatory, and economic.

If any information that I interact with or share in the internet once it gets to the second party the second party wants to collude with anybody they want to then I have no privacy. The only privacy only comes if I can trust the second party to respect the terms and conditions, which is called confidentiality, of the information I give them. What I need is a mechanism for sending information from here to there that is asymmetric. I need to use hybrid public key encryption so that I have asymmetric encryption and I send it to one end. I don't use group encryption. I don't do shared secret encryption. I use hybrid public key which is technically shared secret encryption but the key on my end is ephemeral. I have a reasonable confidence that only the targeted one got it and I can in include information in it that makes it unique to that end which they can get rid of if they're smart but it makes it really hard for them. I said reasonable confidence. What that means is that I have this property of End Only Viewability. The end that I intend to is the only one that can view it.

There's "one data, one where, one time, by one body, "and that means that it's undeniable. Now they can deny it. Plausible deniability. I can't prove, may not be able to prove it, but I know. And guess what I do when I know? I break the interaction. I exclude them from my interaction profile. I stop talking to them. I stop sharing data with them because I can't trust them. The whole point is: I need to be able to control who I talk to and who talks to me. And I need to do it in a mechanism that allows me to decide that and to control the boundaries of that. And that's the way I do it. The TSP protocol lets you decide with whom they interact, what they share and why. You have control over your internet presence and your context when you do this. And we use relationships graphs. And this is actually into my talk for this afternoon.

I'll stop there. Great. Thank you.

TALK 41

Provenanting Patient's Medical Records

Scott Stuewe Mark Scrimshire Jared Jeffery

KERI Conference April 2026 · talks · 48:46

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Scott Stuewe, Mark Scrimshire, and Jared Jeffery discuss using KERI and vLEI to provenance patient medical records and operationalize secure healthcare APIs. They stress interoperability at payer scale, governance alongside technology, and healthKERI's push to replace paper-bound record exchange with stakeholder-aligned digital trust.

Mark, do you want to start with your intro and then Scott and then all… – We can do that. We're going to be talking about KERI and vLEIs and so forth. Let's see if we can bring that up. I'll kick off. I apparently you guys were talking yesterday, so people may know you. Yes.

I'm new. First time here. It's been a while since I've been in Utah. My name is Mark Scrimshire. I'm chief interoperability officer at ONYX. And we focus really on interoperability largely in the payer market for I suppose a bit of background about me.

I was actually invited to go into CMS back in 2015 To build Blue Button 2.0, which is an API for 53 now 60 million Medicare beneficiaries to be able to share their data with the apps of their choice. That led to me now being retained by HL7 where I've written some of the other implementation guides that CMS points to in their regulations. I'm very much focused around how do we take secure APIs and operationalize them.

That's really the focus and that is how it's drawn me to KERI and the way that we can potentially solve that. Gentlemen. Let me go next. My name is Scott Stuewe. We did talk a little bit yesterday. I'm the president and CEO of DirectTrust and we're a nonprofit trade association that supports trust in the healthcare ecosystem broadly and actually increasingly we're describing ourselves as somebody who cares about trust focusing first on healthcare.

With the understanding that healthcare is adjacent to virtually every other sector on the planet. And in particular in this country healthcare and financial services are, deeply related. My background is mostly in the healthcare IT space. I grew up working in the healthcare IT, the electronic health record business. Watched that business kind of grow and but I've also been involved in interoperability and identity kind of from the very beginning of my career.

Worked as what they used to call an interface guy. You remember what that's like. Doing business-to-business interfaces before that there really was a thing called interoperability. I've been doing this… … started that life in '93. And then in the late part of my life in the EHR business, I actually worked on trying to get healthcare information exchange to work as well. We have that commonality. I'm more of the marketect. I think arguably he's more of the architect. Yep. – And I'm just the marketer, right? That's the… My name is Jared Jeffery. I'm CEO and founder of healthKERI. As the name suggests, yes, we are all in on MDL.

That's what we do pretty much every day. No, we… We have seen the light and have decided to roll this innovation that Sam kicked off into healthcare. This really started for me about 15 years ago. I had a job at a law office. My role was to try and get paper records, medical records from point A to point B. And felt like I was going insane because I had a supercomputer in my pocket, but we couldn't get data moving around inside of healthcare.

15 years ago I really started to say, "Okay, somebody's got to be able to solve this problem." And that's led me up to here and to starting this company. Excellent. We're going to cover a couple of things as we go through in really talking about how we're really taking things like vLEI and really applying them into really increasingly established protocols and technologies that are being used in healthcare because I think if we go back to some of the things that Sam was mentioning, we've really got to try and make it easy for for organizations to take that first step and start to get comfortable with the technology.

We're doing a… we'll talk about it later of a particular… we're trying to get a pilot off the ground that is very very focused. But once you see that, you can then see how it could be applied in many different situations across healthcare.

We'll go through all of that. We'll talk about that for FAST security explain a little bit about that because that's driving how the healthcare industry in the US is going to be basically connecting to each other. Let's go and let's talk a little bit about some of the problems you – Sure. The challenge we have is that the healthcare ecosystem is immense and diffuse and also has a need for connectivity which is extreme.

A lot of this is already connected in a way. A lot of this through intermediaries. When you look at the numbers though it's just this kind of staggering story about the different kinds of organizations we're talking about. They all have protected health information that is fundamentally what is the sensitive information that needs to be moved back and forth. It needs to be encrypted all the time. Needs to be protected.

Getting this stuff to actually work is first of all a problem of scale. It's a huge problem. Now interestingly when you look at the providers… providers don't really have very good identifiers. Payers don't have identifiers at all. Like so literally are unenumerated. When you hear about… there's a… public database federal database called NPPES, which was originally in its conception designed to enumerate both payers and providers, but in the end only ended up successfully enumerating individual providers.

They don't do organizations very well, and they don't do payers at all. Next slide. – Can I just put a little bit of note on that as well, which is that the the onboarding process for the NPPES system is really light. Ryan House, who was here yesterday, didn't mention this in the conference there. He has an NPI number. He is not a clinician. He does not provide care for anybody. But on a lark he decided to go see if he could get himself an NPI. I asked him what code he used. I think it was probably therapy. If you need a shoulder to cry on, Ryan House [laughter] is your man.

– I think the next slide, we actually kind of get into the NPI question, I think on the next. We can't answer this basic question. The onboarding process for all these organizations and the onboarding process for our CMS is weak at this point.

They're looking to strengthen it. I think they're talking about strengthening it and including IAL2 identity assurance as a part of the process. That's something that they've said they're going to do. Unfortunately, that's only the very smallest part of the problem.

The manual onboarding that these organizations do with networks, for example, is a tremendously onerous process on the one hand, but also not standardized. It's standardized in policy in theory. In other words, there are a couple of big national networks, one called Carequality, one called TEFCA. They're are very similar. They are almost have the same participants. One came first and the other one kind of came second as an effort of the government.

The 11 different organizations that operate the TEFCA ecosystem, those 11 different, they call them QHINs, each have to onboard the organizations to those various places. And the way the topology of this stuff works is that the people who are connected by one QHIN then are passed from the QHIN to another QHIN and to their participants and sub-participants. And this weakness, these extra hops, basically means that the people who are receiving messages have no idea who sent them the message cuz there's no actual understanding of the organization across that boundary.

There's also no universal organization ID. Particularly, the only thing we really have that does organizational vetting is the issuance of a digital certificate, which is something DirectTrust does at scale today for a whole bunch of organizations, but it's not really all that useful to tell you about the organization. It just tells you that this person has control of this domain, which is not all that useful, particularly if patients are trying to find where their doctor is, which is one of the problems we're trying to solve.

Is if I've got a directory that has a, listing of all the places my I might look, how can I tell which one is where my doctor's stuff is? It's a big problem. There's also a big deadline. – I can maybe talk a couple of things here.

You see digital identity, individual identity there. That doesn't solve the problem. My sister's a doctor in the UK, and on Monday she would be working for one organization, on Tuesday for another. It's all about context.

You really need to be able to tie that organizational identity and the individual so that you understand what context they're actually acting in. Now, the other thing that we were about to get to is CMS-0057. It Sorry, it rolls off the tongue. We add the two zeros, so it sounds a bit like James Bond. But it isn't. But this is really mobilizing the industry because CMS a couple of years ago put out regulation. And remember, CMS spends what, about 1 and 1/2 trillion dollars a year, which they give to basically payers around the industry.

They hold the purse strings. They put out regulations that basically say, "You need to do this." Right? And 0057 really does a couple of things. One, it's expanding patient access. If you have a plan with a Medicare Advantage organization, a payer, you're allowed to go and get your health data that they hold. Now, what they're doing is they're adding in prior authorizations to that.

The other big thing that everybody's focusing on this rule is providers are going to be able to submit in a standard way a prior authorization request and get an answer from the payer. That means you've got a secure API, and you literally going to have tens of thousands of provider organizations having to talk to a thousand different payers to get an answer.

Today it happens through provider portals at the payer, through faxes, through phone calls, all of these things, no standard way of doing it. What CMS has done is saying, "This is an API standard to do this." Everybody's busy automating that, and the deadline is the 1st of January, 2027. We are about 7 months away.

– And I'll say one other thing here. If if you're interested in FHIR, my session later today talks through I think in depth what FHIR is, how it works at a very atypical or atechnical level just to kind of put some definitions in there.

– Yes, so a lot of the CMS series of 57 is really driving the adoption of FHIR (fast healthcare interoperability resource) specifications. that drives a standard way of expressing exactly what you want to do. Here's a patient record, here's a coverage record, here's an encounter record, etc.

The industry, particularly the payers, are busy implementing this and hopefully also the EMRs will as well. I think we can go on. – This is sort of the nature… Interestingly you know United Health Group and actually most payers are publicly traded companies. As a consequence, they actually mostly have LEIs, which is actually a useful thing.

But also just to understand the challenge of the complexities here, disambiguating an organizational unit over here, the names of these health insurance organizations. If there are 1,100 or so, some I don't know, 10 or 12 of them are actually under United Health Group. For for you to find a health plan that's associated with United Health Group, you need to understand what the LEI can provide for you, which is the who's your daddy problem, right? All of these children health insurance companies roll up through United Health Group. Now not shown on this slide is the complexity underneath United Healthcare Services.

UHS actually operates an incredibly complex network of variable kinds of organizations, some of which are actually provider organizations. One of their subunits is called Optum. They're actually the largest owner of providers in the country.

The largest employer of providers in the country. And they roll up through a company that also has a bunch of health insurance companies. It becomes very important for us to figure out I'm not just talking to UHG. That would help me in no way at all if I want to figure out what the bonafides of this organization are and what frankly they ought to be able to do on the network. I need to know exactly what organization I'm working with, but also I probably need to know more about that.

I actually need to know if they are bonafide for a given purpose of use. Who says that they can… And there's actually a huge controversy right now on the care side relative to whether or not you are bonafide for treatment. The question of whether or not a provider, just a provider who's making a query can do so on the basis of treatment. In other words, that they're actually treating a patient at this point.

They might work for a law firm at this particular moment. They actually are practicing provider and they might work for all sorts of other companies potentially. But they they shouldn't be able to operate in that context if they are working in that particular setting.

Understanding where they are really literally, but also what is the purpose of use that they are… that they are bonafide for. – And Scott, I think… – I'll just really quickly sort of what our role is and I think we're going to move into a very much more technical mode here after this, but one other little thing I can say is that we play this role today. When I talk about our community, we do not ourselves issue anything.

We're a non-profit trade association. Our members are the operators of a very large healthcare network that actually operates a PKI framework for that purpose. And if you actually pull back the covers on what DirectTrust does, it looks oh so very much like what the vLEI ecosystem looks like. It's it's actually a very good map in terms of how it would look cuz we're fundamentally a a governance engine. We're not a we're not… some people argue we're not even a network, but we do operate a network. This idea of our accreditation process is what gives us the ability to make sure everybody's doing what's right associated with the governance not just of our governance expectations, but also for expectations of external sources of truth like 853 or 863 as we were talking about yesterday. If it's identity assurance or if it's taking care of your cybersecurity bonafides, all of that stuff is what we do today.

– And I think it's important to note Sam said this morning that the technology doesn't solve the problem alone. And I think increasingly with with Daniel's presentation earlier with what we're talking about here, you do have to go get the stakeholders from the governance side involved because you can have the best technology in the world, but if you're not solving the policy concerns you're not going to get to actual results. – Yeah.

Just touching on the scale of the problem, what we've seen if I go back to the work I did with Blue Button, you would have individual consumer apps coming and trying to connect so that they could then have a member use that and give them permission to go and get the data. We built, at ONYX, we built an app Get my data, and we wanted to connect to health plans across the country so that you could go to this one place, you could find the health plan, authorize sharing, pull the data.

We actually got 350 health plans connected, but it took us about 6 months. Because everyone you had to go individually to the health plan. And this is where you've got like an Aetna and they've got 47 plans. But you rapidly get down that curve where you're talking you run a plan, you've got an API, I need to connect to it.

And the process is a manual back and forth. You answer the same 50 questions 150 different ways at every different plan. And they'll finally decide that they're going to give you access. Under the regs, there's very little they can do to refuse.

It's a heavily manual process. As we were looking at this requirement to do things like providers have access to payers to pull data about the members they're treating or to be able to submit a prior authorization or for one payer to connect to another payer because a member's moved and they want their data, you get this math problem of the number of connections you need to make. And we need to… we are not going to be able to operationalize the APIs if we cannot find a way of basically automating this.

This again comes back to this core issue that particularly in health care space, we don't have a standard identifier. We haven't got an easy way to identify. And in fact, the plans tend to push away from that. Talking about having a national plan ID it's taken years and it's gotten nowhere.

But we do know as you said Scott, there's about 90% of the health plans already have an LEI. Right? They're already on the slope. We've got to just prod them down there. And I imagine Karla can tell you it is a very laborious process to get your LEI. I think it's about $50 and what a day, 2 days if you've got all the documentation.

– 15 minutes of actual real work. Yeah. Complete completely untenable for the industry to adopt, right? – Can't be done. – Yeah. Let's go into what we're really trying to propose here. The first thing is that let's get people from an LEI to actually having a digital credential for the organization. Okay?

That brings a whole bunch of benefits and Jared, feel free to chime in on this. But really what we're trying to do I suppose I should take a little bit of a diversion here and talk a little bit about TEFCA from the payers' perspective. Scott, you mentioned, you There's a lot of the providers have gone into TEFCA. They were in CommonWell. They're probably still in both.

The challenge is that to get into TEFCA, you're then agreeing to do Basically return information about treatment in multiple different formats. The payers are looking at this and saying I'm going to have to respond to a whole boatload of queries. I can't really do the same in the opposite direction.

It's very imbalanced and a lot of this stuff I don't want to be dealing with. I'm worried about January 1st, 2027 and getting my FHIR APIs operational. I don't need to worry about having to do IHE formats and so forth. They've been resistant. How do we get them moving forward? And this is really the first step. – one other reason why they're resistant is they already have the data that they are required to ask for in this new model. They're like, they already have this data. They've done individual data sharing relationships with all the electronic health record companies. They don't see the value proposition. It's like you can't really get them interested in TEFCA. And so, they want to try and do something that is extra TEFCA. I like to call it extraterritorial.

– And really the purpose there is that then they can start to set the boundaries for the connections. Right. CMS has said this is what you got to get stood up. It's got to be FHIR APIs. But, in terms of the value for the payers, it really comes down to and Phil did this demo yesterday and I'm sure he'll talk a little bit more about it today.

It's that idea of can I take a member's request and bring it through from one payer to another payer to pull the data back out through. And that's what they're trying to get solved here that doesn't get discussed as deeply in TEFCA, but will be incredibly relevant for the CMS-0057 connections that are required by January.

– The key thing as we're looking at using vLEI is really being able to identify who is the organization, what's the purpose of use, and who's actually been delegated the authority to actually engage into those transactions, right? And I'm basically coming this at this from a very simple perspective.

And I've been in calls where we start going down the digital identity, the individual identity route. And I'm saying, "No, no, no, no. What we're actually trying to do here is you've got this first step of actually getting secure APIs connected. And how do you have confidence that the organization that is representing is actually who they say they are and the person that's trying to do that setup is permitted to do it.

And there's multiple types of transactions we're talking here, PriorAuth payer to payer, provider access. How can we really start to bundle that together so that you can answer that question and ultimately be able to automate it?

This is where Jared and I are going to try and make Phil at the back there cringe as we try and do more of a technical explanations. – Let me just run through this at a very high level and again point you to some of the stuff that Phil's going to be presenting later today.

The way that we see this is that there's with some minor augmentation at the ECR level, what GLEIF has done with the vLEI is actually pretty much perfect for this use case. It's purpose built for what we're trying to do here, which is if I don't understand the other organization on the other side, then the GLEIF ecosystem solves this problem. And I would… Just raise of hands, is anybody in here not familiar with the vLEI's architecture, how they lay these things out?

Okay, then I think we can go very quickly through this just to say that all the way down at the bottom, the ECR credential piece, that's where there's going to be a lot of work done I think with DirectTrust and others in the industry to determine what is purpose of use. And we're starting with payer-to- payer, but the amount of ECRs that you can issue within this ecosystem that is specific to healthKERI is incredibly valuable. Everything that we're doing, we really want to keep it as stock standard vLEI as we can to the point that if there are vLEI issuers that are working on issuance outside of the healthKERI industry, we've really pushed to say the vLEIs that operate there should be able to operate here. Now, one of the

questions that we had was, okay, well let's say you've got Bank of America and they've got a valid vLEI. Just because they have a valid vLEI, can they come into the healthKERI ecosystem to one of these payers? Can they go to CVS Aetna and ask for your data because they're a validated organization? The answer is no because in that ECR credential space, we'll have specific purposes of use that we talked about already. – Yeah.

And I think that's key and obviously by being more precise about the purpose of use. Whereas TEFCA is basically you join the network and anybody within the network we're seeing now some of the legal challenges saying are you really who you say you are? And are you able to really ask for that data?

By being very precise about the purpose of use, it actually potentially simplifies the policy framework that's got to be around that. To the point of one of the things I'm proposing is in FHIR, we actually create implementation guides that define effectively a use case for the standard, right? Each of those IGs has a canonical basically URL that you can reference. And that could be a mechanism for actually defining the purpose of use because what you're saying is within an implementation guide, there is a API for doing, for example, provider access.

There's actually a canonical I know I wrote it. you could define that as something unique that everybody then understands this is the transaction type that I'm engaging in. We're really then leveraging particularly down the ECR credentials to define those purposes of use.

It really goes this way. We get an organization gets their organization credential. We're looking at the fact that there are going to be people within the organization that will actually do the transaction. In effect somebody, for example, in an IT department or in a compliance department is going to be responsible for making sure that these connections get created. Should they have been delegated the authority to do that?

Another thing, with just the complexity of healthcare, we by the end of this year, we will actually represent the interoperability capability for more than half of the Medicaid agencies in the country. Right? We are operating on their behalf.

They would effectively, if they brought into this, would have a vLEI for the Medicaid agency, and they would have a delegation of authority to ONYX to be able to go and set up these connections with whoever they wanted to set up. – And similar to the Provenant use case here, that is incredibly essential, right? Because while we will want ONYX and others to be able to act on behalf of those Medicaid organizations, we don't want them to be confused in the pathway, right? You still need to know whose throat you can choke, essentially. – Who's it going up to? Exactly.

We're building this effectively this chain. One of the things we're suggesting is also when you make a connection, what typically happens today, we can get into the technology of smart on FHIR, so it's the issuing of a credential to allow you to now effectively open up the API.

You normally… that's just a transaction between the two parties. What we're saying is, well, what happens if the party that issues you those credentials, so payer A goes and says, "I want to connect to payer B for payer-to-payer."

When they issue the relevant smart on FHIR credentials so that the APIs work, what if they give back a credential that says, "Payer B has approved payer A." And now you put that into the package. And now you go to the next payer, payer C.

And you can actually present to say, "I'm already connected somewhere else." You get to a point where a payer is going to go, "What the hell am I doing by trying to do a manual verification of you when you're already connected to another bunch of folks in the in the network?"

That then starts to lead to you can automate the process of actually registering. It goes from being as being conserved at $2,000 at a time to hundreds of dollars a time. Right? Just because you can now automate. And as long as it we can use the capability in the vLEI to say, "If there's a bad actor, they can be revoked."

You can check that. And therefore there's confidence in the network. – I really want to hit this concept here really hard, which is that Mark has coined it as connectedness is trust. And the way that I put it is trust equity. The reality is that for every organization, every single connectivity point that they have to operate today is both a bottleneck and a cost center. You get nothing from it other than the connectivity. It is a bother, and once you're done with it, you go to the next, and you repeat that entire process. And the vLEI allows you to start to do this connectedness is trust. You start to build a trust equity within your ecosystem with those back filed credentials that say, "Hey, we're connected. We've done the check. We trust who you are." And at a certain point you get to the opportunity to actually look at one of the biggest monoliths in healthcare, which is the clearinghouses, and say, "We don't actually need to rely on you to do our network of connectivity anymore."

Because one of the problems that healthcare has right now is you've got a monopoly of a few very small parties that are making rent-seeking to the tune of billions of dollars on facilitating all of this exchange. And that sounds fine, I guess, if it was just that problem. But one of those clearing houses was Change Healthcare that got blown up and it's been 18 months now. And you go ask some of the mom-and-pop clinics that had to fold up shop because Change got completely destroyed in an attack.

What we're talking about here is an alternative. If you go talk to an Availity, if you go talk to one of these large clearing houses, their answer is, "This is a solved problem. Just be our customer." And the contention that we have for the entire industry is that that's a really, really bad idea because we don't want to centralize power into a few different private organizations as they move forward. Now, I don't think the clearing houses are going anywhere.

But what we do want to present is an alternative where you can build your own trust equity, you can build a network of connections, and you can do it automatically and at scale. – And I wasn't here yesterday, but Ryan was.

When we first presented this concept to to Ryan, and he's been very much promoting this, the first words out of his mouth were, "Is this open source?" Because if you're going to get this adopted across the industry, you don't want those barriers to implementation.

That's been one of the real benefits of HL7 and the FHIR protocol because that was made open source. Anybody can go and look at the specs, they can take them, use them. There's no barrier, whereas we're looking at X12, and I have to go and buy a license every year just to look at what the spec says, right? We don't want that barrier. That's really important. One of the features that KERI brings is this open source basis.

Anyway, let's… Let's talk about this What we're actually trying to do here. There's a couple of initiatives going on. TEFCA, as Scott mentioned is really this national network. It's largely providers. Payers haven't leapt into it yet.

There is an accelerator FAST (FHIR at scale task force). It started as part of the ONC initiative, but now is an official accelerator within HL7. And they're really looking at how do we build the infrastructure to enable all of this interoperability.

It really is an underpinning of of FHIR. Now, they have an implementation guide. They have multiple. FAST Security or FAST Security for Scalable Registration, Authentication, and Authorization. Terrible. But it's one of the longest.

That is actually in regulation saying TEFCA is going to use that. And that defines the mechanism for basically acquiring those smart on FHIR credentials to do a connection between two organizations. That is in there. Now, that FAST Security protocol implementation guide is extensible.

And that's what we've really tried to leverage here in saying, "Let's use the vLEI and all of the associated credentials to make it easy for an organization to move into this and have more confidence in who they're connecting with." I'm just saying, "One thing about this that in its first iteration, this was expected to be an X.509 certificate that anchored these connections for organizational security.

This idea of being able to have clients dynamically register with servers and get so that they can eliminate that access topic was initially imagined to be a strictly an X.509 exercise. – Yes, so I suppose as an aside, I actually trace the dynamic client registration protocol in a OAuth 2.0 is really there in what we call SMART on FHIR. Okay? It's the mechanism of getting connected. You've got this ability to get credentials. Dynamic client registration has been there in the OAuth spec for years.

And I can trace it back to in 2015, I wrote a blog for the Department of Health and Human Services about how could we actually really leverage dynamic client registration? Nobody used it because they didn't trust who was coming in and saying, "Just register my app."

I went to, I think, organizations like DirectTrust and a couple of others and says, "Is there a way that you could tell me whether the person that's actually submitting can be trusted?" That really emerged into what is often referred to as UDAP, but if you're talking to the FAST folks, if I say UDAP, someone's going to come out and punch me. It's FAST security.

What that's done is basically put an X.509 certificate on top of dynamic client registration so that you can say, "I know who the organization is. Here's all the information you would normally have plugged into a developer portal. Go register." Okay?

What we're proposing is using the extensibility in fast security to say, let's add an additional cryptography check, which is for the vLEI. So, that you can put the vLEI in there and I know Phil's going to be cringing at this because I'm not as expert as he is.

You can basically reference the vLEI, I'll call it the vLEI package. All of those bundles of credentials and validate that they are legitimate. Right? What we're also… in there is effectively setting a confidence factor.

Because if you are able… What's going to happen is we have this crazy scenario. You go and talk to health plans and they basically say, we're trustworthy. It's all those others out there that you cannot trust them, at all.

There's a problem with looking in the mirror and really saying am I trustworthy? We've got to try and break that. What happens is they are going to want to manually validate anyone that even submits this. Right? What we're saying is let's allow them to configure a confidence factor. If you pass that threshold, you could be automatically registered, go into, get DCR runs, you get your credentials, and away you

go. – Yeah. And I want to call out a couple of things on this slide in particular that I think a staunch KERI advocate would bristle at. One is the word OAuth 2.0 and the other's token grants. And I know that there are people in the room that look at that and go, well, why?

Why do that at all? Like if KERI is what KERI claims to be and I think we all believe that it can live up to its promise, why go backwards? And the reality here is similar to I think what Provenant is doing with Telco, you've got to meet the market where they are.

You have to come to people and say, we're just adding a line for that kind of cryptographic proof on the ecosystem. And that's one of the most common questions we get as we start to go into the pilot on this is payers saying, "Are you trying to rip everything out and replace it?"

And the answer has to be right now, today, no. No, we're making an augmentation that is incredibly minor to gain a ton of value. Now, if you ask me, maybe not on stage, my answer might change and I say, "Yes, we want to rip all this out. We think we can do it better." But we're going to walk people into that path, right?

– Yeah. Put this in context. Back in July 2021, All of these health plans that are funded by CMS, so the Medicare Advantage, Medicaid, CHIP plans, so Children's Health Insurance Program, and Medicare, MCOs, all of these have to follow these CMS regulations.

There's about a thousand plans out there. They have invested in FHIR interoperability with smart on FHIR, so with OAuth 2.0. It's gotten through their security teams, which is a major thing. I took this concept through CMS. I had to actually explain what OAuth 2.0 was.

Now that's accepted. Really what we're doing is we're just adding an additional check on top of that. You're building on top of something they've already implemented, and we're trying to make it really easy for them to take this extra step and gain a lot in return.

– And the ecosystem also includes all the provider organizations, and they're all served by electronic health record companies and each of those electronic health record companies have also made that same investment. And so, as a consequence, what you've got is tens of thousands of organizations that need to make changes to their road map to include new capabilities. This is a whole lot harder than getting Google and Apple to do something, by the way.

This is really where we're at. It's not a replacement, it's one additional check is really what we're trying to drive home here. That we take the extensibility of FAST, we leverage vLEIs in that, with all of the subsidiary credentials and you've now got the ability to have more confidence in the organization that's connecting to you. You can also define specific purposes of use.

We're talking about payer-to-payer. It is required for 1127. A 1,000 health plans out there in the US are going to have to support payer-to-payer. When you decide to up and move to the next plan, as many of us do, you can say, "Go get my data."

They're going to have these connections. Rather than confuse things around the fact that I've got tens of thousands of providers and that's complication, say, "Let's get you comfortable with the technology between people that you sort of trust. They don't trust each other, but they sort of do, right? Just a little."

We're trying to drive this as a specific pilot, but if you talk to folks, I've talked to folks in the Medicaid agencies, I've talked to folks at the various plans, and their mind is immediately going, "I may have to deal with a hundred or a even up to a thousand health plans, but I've got tens of thousands of provider

organizations" And that's where their mind is already going there. We just don't want to push them there right now.

We'd rather them get comfortable and then say, "But we can extend that." Again, it's a small step further. But we bring them… – We're boiling the frog for sure on this one. –Yeah. Totally. The other thing that is happening is there's actually… we get… what's the time?

– You got about 2 minutes. We may we through. – What we're actually getting to is we're proposing this CMS aligned network. That's a sort of mini version of TEFCA focused on doing specifically payer-to-payer, so we can actually prove this technology in the wild between a handful of payers and then scale it potentially really quickly.

That's really where we're going with this. If we had… we were more or less at questions. I don't know if you've got time for questions. Yes. – I love what you're doing. It's really ambitious and it's obvious you have deep industry knowledge. What I was curious about was in particular the notion of the canonical URL.

To me it seems like if [inaudible] this city is associated with the URL wouldn't that be an anti-pattern? Wouldn't you rather have some sort of canonnicity that was tied to the content itself rather than a path to the content?

– What we're really talking about is what's a shorthand for for example payer-to-payer? Right. Just trying to take the easy path… is there's hundreds of organizations out there that are looking at implementing payer-to-payer. And they're going to have to connect.

They're all looking basically at the same implementation guide. And that… we have scopes in those implementation guides and one of them is you're going to give a permission to do, for example, a member match. And that… – And that permission is embedded in the URL? – That's… you can effectively look up the implementation guide by that URL, which says here's the use case that we're looking at. What we're saying is that could be a short hand to put into the purpose of use because it's very specific. It's about this particular use case that's been defined as a standard. – It's interaction. – Yeah.

Anybody else? – Okay. All right. – And after that question, we got a great announcement of Jared. – Back in the mid to late '70s, I was responsible for a database called Sparks, which has all of the admissions and discharges from every hospital experience in the state of New York.

And we traced the doctors and we traced the organizations without this then by their license number. And every state in the United States, there is a license number for doctors and clinics and hospitals issued by their local… their state health department.

Why aren't you using that? – Let me describe the challenge I think that that represents. I think… The payers, for example, frequently use EINs as a part of what it is they use to identify organizations across the ecosystem. The challenge here is that these organizations, first of all, operate across state boundaries.

The meanings of those things are going to change as you move from state to state. The other thing that is challenging about any of the enumeration models you have today is that they don't actually represent hierarchy. And this is actually a real problem for any of them. The biggest issue I see is with tType 2 NPI, which is what they do for the healthcare organizations. That is what CMS uses to identify those organizations, but it's not really the organization they're identifying. It's a billing group, which is not really

helpful. Not to the outside world. They don't understand that construct. Getting something that is universally understood as well as disambiguating is the challenge. And it doesn't work if it's a state-issued item because these organizations are multi-state entities with children who are in other states. It's a complexity that is difficult to manage.

– All right. Thank you. Then before we go to the last plenary speaker of today, I would want… so we've talked about this I think a few different times today. I guess the the announcement that I'd make for you is and this is probably deleterious to my own session. But I would encourage you as you look at the agenda for the rest of today and tomorrow to bias your sessions towards the KERI Foundation sessions and the sessions that are being hosted by Ari and Phil. The reason that I say that is because we have pushed a massive amount of code into the open source as a result of wanting to kickstart this ecosystem. We're talking about things like witnesses, watchers and the Locksmith wallet as well. If you are looking at the things that we're trying to do, if you think you've got an ecosystem that is very relevant here, I think we've seen really common threads through Telco and healthcare. If you have lines on other industries, you want to start doing these things we're equipping you with all of the tools you need to get started in that process.

Again as much as I'd love to have a full session later today, if there's a competing session that's a KERI Foundation or Ari or Phil I would bias you towards those to go learn more in depth about what exactly has been put into the open source and what the value is for you as an individual in the KERI community. – First, a very big thank you for donating all that code into open source. – it represents literal man years of time, – Fantastic.

TALK 42

KERI Infrastructure & the 5 W's

Thomas Mayfield

KERI Conference April 2026 · talks · 33:24

SUMMARY

AI generated summary, if a statement needs explication, see the video.

Thomas Mayfield of the Cardano Foundation explains KERI infrastructure through wallet, witness, watcher, web, and wizard. He contrasts tenant-like platform identity with key-controlled portability, then covers pre-rotation, delegation, multi-signature control, and ACDCs, showing how Veridian implements these roles as independently verifiable production infrastructure.

All right. Good afternoon almost. My name is Thomas Mayfield. I'm here from the Cardano Foundation. And I'm going to be speaking today about Veridian. And usually when I'm in front of this majority of this group of people, I am demonstrating what we've been building.

But today I have the honor of actually providing a bit more of a educational or academic. Is it not loud enough Jared? Testing better. Excellent. Okay. My name is Thomas Mayfield. I'm from the Cardano Foundation. And I'm here going to be speaking a bit about Veridian. And usually when I'm in front of this group of people, I'm demonstrating the tech that we've been building at the Cardano Foundation.

But today I have the honor of actually providing a bit more of an educational overview of KERI and more specifically speaking about the concept of the five Ws. I will intertwine elements of Veridian throughout the talk. But I think the majority of this audience, it's probably going to be Already content you're very familiar with, but hopefully for the viewers that are going to be watching this online and at home, it's a more of an informative content for them.

I'm going to put my timer on so that way Henk doesn't have to give me a reminder. All right. Today I'm going to walk you through the KERI infrastructure using a framework known as the five W's. Wallet, witness, watcher, web, and wizard. And by the end of this talk, you'll understand not just what each of these roles are, but how they work together as a system and also how Veridian is building all of this in the open.

Let's start with the problem. Digital identity today is borrowed. When you create an account, get a credential or receive a certificate. You don't actually own it. The issuer does, the registry does, or the platform does. And if that platform shuts down, that registry gets compromised or that issuer revokes access, your identity effectively disappears with it.

You are effectively a tenant, not an owner of your digital identity. But KERI changes this. With KERI, your identity is rooted in cryptographic keys that you control. No registry needed, no platform dependency. It's portable, verifiable, and it is yours.

What is KERI exactly? And I don't think anyone's actually said the acronym in the last few days. KERI stands for key event receipt infrastructure and there are three core concepts that make this work. The first is self-certifying identifiers and what this means is that your identifier is derived directly from your cryptographic keys. There is no registry or registrar that grants you your identity.

The second concept is that every critical action you take with that identifier, such as creating it or rotation those controlling cryptographic keys, all of these critical events are signed and chained into a tamper-evident log. And the third concept is everyone can verify your identity by replaying the same

tamper-evident log. There is no need for phone home to a central repository or authority. And if you'd like to understand a bit more about no phone home, I encourage you to visit to visit nophonehome.com and you can read a bit more about this.

This combination of concepts, this is what gives us something that no other identity system achieves and that is truly portable trust. Let's make this concrete. In KERI, your identifier is called an autonomic identifier or an AID. And this is a cryptographic identifier that you generate. Its authority comes from the keys that you made. And no one else or entity issues that to you. Every significant action that the AID conducts gets recorded in a Key Event Log and that is called a KEL.

Your inception event, this is when you create your AID. The rotation events that change the AID's controlling cryptographic keys or interaction events that anchor data that is critical to your AID. Each and every event is cryptographically signed.

Each and every event is cryptographically signed with the previous one. And this is what creates a tamper-evident chain. If anyone alters a single event in this chain, the signatures break and the tampering is immediately detectable.

If you're familiar with blockchain technology, this is the same hashchain data structure that makes blockchain secure or immutable. Now that you understand the foundations, AIDs and KELs, let's talk about the infrastructure that makes it all work at scale. And we've organized these into five roles. And conveniently, Sam has started them all with a letter W.

Let's move on to the wallet. And this is where your identity lives. It's the holder's home. The wallet generates and rotates the cryptographic keys. It stores the verifiable credentials. In KERI, these are known as ACDCs. We'll talk a bit about this later. The wallet signs, presentations, and interaction events. And it collects all of this and it combines all of this with device grade security. This includes biometrics, secure enclaves, and anything else that occurs on the hardware device itself. The key insights here are: the wallet is the only component that the holder directly controls. The vital functions of the wallet in KERI are key management and signing.

At Veridian, we've built an open-source mobile-first KERI wallet that is production ready and available today. You can download that from the Apple iOS stores as well as Android Play Store. and it is also available on the privacy-preserving operating system Graphine OS.

Witnesses are the second W in the 5W framework. They are designated services that receive, sign, and republish your key events. You can think of these as notaries for your KEL. They give your event log a highly available home so verifiers can access it when your wallet is offline. They produce receipts that prove the specific event has been seen at a specific time.

And through redundancy of these witnesses, we can resist censorship. Critically, you choose your witnesses. They are not selected or imposed by the system on you. And at Veridian, we operate witness pools that you can use directly or you can host yourself in your own infrastructure.

The third W is the watcher. Watchers are independent observers. They are critical to KERI. Their job is to detect duplicity. Meaning: they catch anyone who is trying to present two conflicting versions of the same Key Event Log.

This is KERI's answer to the double spend problem that you have in blockchain. But they do it for identity. Watchers give verifiers confidence that the state they are seeing is the latest legitimate state of an identifier.

They run separately from issuers and holders which means they are truly independent. This is what turns 'trust me!' Into 'check for yourself' or 'verify for yourself'. And at Veridian, we are deploying highly available watcher infrastructure services in the second half of this year. And I would be remiss if I did not pay attention and show gratitude to the HealthKERI for releasing the watcher and wallets to open source.

The fourth W is web. This is the discovery and transport layer of KERI. This is how parties find one another and how they exchange events and credentials and they do so through something known as an OOBI. This is an out-of-band introduction. You can think of these as sharable links that bootstrap trust between parties.

Effectively, these can also be assumed to be QR codes that establish an initial connection and from there mailboxes and endpoints handle the messaging moving forward. The design of KERI is transport agnostic. Today it runs over HTTP, but the protocol doesn't really care about the transport. It can run over Bluetooth, over NFC. In fact, it can run across anything.

And it's interoperable across different wallets and agents. Veridian provides hosted endpoints and sandboxes from our data center in Switzerland, including infrastructure deployment guides for those that are interested in deploying.

The fifth and final W, the wizard. This is how we achieve one-click KERI. For example, the majority of users are not going to understand how to set their witness threshold. Wizards abstract low-level details of KERI and they provide suitable defaults that ensure we make KERI accessible but safe for users.

For developers, we expose highlevel easy to understand APIs to build on. And in short, the wizard bridges the protocol to the product. Without KERI, sorry, without wizards... Yes, without Wizards, KERI is a powerful protocol.

With Wizards, KERI is a product that you can ship to market. And at Veridian, we offer one-click onboarding, witness selection, and automated schema discovery. Now, this is the whole picture. This is how they all fit together.

Wallets hold the keys and credentials. Witnesses attest to events. Watchers verify integrity. Wizards orchestrate operations and the web connects them all and provides scalability. Each of these five roles are independently operated, independently verified and uniquely can be independently replaced.

That's the friendly overview. We're going to delve a little bit deeper and I want to cover four concepts that take KERI from an interesting protocol to a production grade infrastructure. And those four concepts are pre-rotation, delegation, multi-signature control, and ACDCs.

We'll start with pre-rotation. And this is one of KERI's most elegant concepts. In fact, it was probably the main concept that enabled me to influence the board of the Cardano Foundation to allow me to build on KERI. Pre-rotation, as I mentioned, is one of the most elegant concepts of KERI. At inception of an AID, the moment you create it, you don't just commit to your current keys. You also include a hash of your next-keys set. Those next keys stay offline.

They remain unused only until you need to rotate your controlling key. If an attacker steals your current signing keys, they cannot forge a valid rotation event. And because they cannot do that and because they don't have your next keys, but because you have committed a hash of your next keys, that rotation is already locked in and you can verify someone is legitimate controller of an identifier without publishing the public key. This is extremely important.

This gives KERI something called post- compromise recovery. Even after a key compromise, you can regain control. And because of this commitment, using this hash, we have resistance to future quantum-computing attacks. There is no other digital identity system that offers this level of key management today.

Multi- signature control. This ensures no single point of compromise for high value identifiers. An AID can require M of N signers. For example, two or three board members must sign to rotate the organization's root key. The threshold can be weighted for governance scenarios. Rotation events are also respective of the threshold rules. This means you cannot bypass multi- signature control even during a rotation event.

This is critical for enterprise AIDs where a single compromise no longer means the entire organization is now compromised; and 'game over'. And Veridian supports multi-signature control coordination, out-of-the box, with our Veridian wallet and our cloud agents. And we handle the ceremony of gathering the signatures across multiple parties seamlessly for you.

ACDCs: authentic chained data containers. This is KERI's answer to verifiable credentials, but they go much further than the typical verifiable credential or VC that exists in the decentralized identity space. Each ACDC is signed and anchored into an issuer's KEL. Its provenance is always verifiable.

Schemas are composable and versionable, so you can build complex credential types from simpler ones. An ACDC supports graduated disclosure where you can share only the minimum information needed for a given interaction. And you'll notice on the slide that the credentials chain back through the schemas and issuers to a Root of Trust.

This is what authentic chained really means. Every link in the chain is verifiable. And Veridian wallet allows you to store and present ACDCs natively. It's also important to mention that although the wallet is storing a copy of your ACDC, the way that ACDCs work, the way that they are presented, if you lose your wallet, you are still able to recover those ACDCs because it is effectively the keys that allow you to prove ownership of those credentials. As long as you have a backup of that data somewhere, you can regain those credentials or you can request that they are reissued to you. This is extremely powerful.

Delegation of authority. This would probably be the second reason I was able to influence the Cardano Foundation to pursue this technology. Delegation is how KERI handles organizational identity. An AID can delegate authority through ACDCs down from a Root of Trust. GLEIF's vLEI system uses exactly this pattern for organizational identity within the financial sector. A qualified vLEI issuer, a QVI, receives a delegation from the GLEIF's root AID, and then has the authority to issue legal entity credentials to other organizations.

That same authority can also be revoked at any time. Delegated AIDs also enable top-down hierarchical recovery. For example, this is how GLEIF can recover a compromised QVI's AID. With KERI delegation, the trust chain is always auditable. You can always trace any delegated AID back to its root.

At Veridian, we are building all five of the W's. The open- source mobile wallet. We are hosting witness pools and self-hosting options for those who are interested. We are committed to releasing public and private watcher services this year. We have hosted endpoints, UB resolution, credential management services, secure KERI tunnels and identity access management integration modules.

We have enterprise onboarding, provisioning services, deployment and infrastructure automation as well as disaster recovery. All of this is open source. All of this is production grade and it is built in the open by the Cardano Foundation. I'd now like to share with you an overview of what it is Veridian is aiming to

achieve for the second half of this year and into the years to come.

Can you trust who's on the other side? Institutions cannot verify who they're children with strangers. AI agents execute without a story Today we cannot answer the most basic questions who's on the other side and who authorizes to act. Veridian has built the answer to both an indentifier you can verify a trusted .. to every transaction auditble, traceable, compliant, governments can verify identity patients' data owner controlled age verified without saying name birthdate or address Both sides of every transaction verify instantly ...

An AI agent can take over ... you can control over automated action Authorized on record before anything can happen. Veridian, the digital trust every industry needs. I leave you with these takeaways. If there are four things that I would like you to take from this talk, it is 1. KERI moves identity from platforms and registries to cryptography.

Self-certifying identifiers combined with signed event logs mean no authority is needed. 2. The five W's make KERI concrete. Wallet, witness, watcher, web, and wizard. These five roles in one coherent system. Three, pre-rotation, delegation, multi- signature control make it enterprise ready.

Post-compromise recovery, hierarchical trust, and threshold control, address real world needs for the digital age. And 4. Veridian ships all of this open source. You can start building today. If you just need to verify, you can just verify.

But it is all here available open source. And finally, I'd like us all please a round of applause and thanks to Samuel Smith for his life dedication to KERI. Thank you, if there are any questions? I can provide the slides if you'd like.

If you go to the website, reach out or just come speak with me directly. I can share them with you. Thomas, thank you for this great u presentation very just actually your voice just brings peace. But anyway, go going back to my question is given that Cardano is it's a proper blockchain I'd like to know more how you leverage it. How I assume witnesses or watchers are part of the infrastructure that the ledger itself provides.

The Cardano Foundation is a very different blockchain foundation than the others. We don't have castles and yachts. But we do have a vision for what is the right thing to do for technology and for people. And what we've been able to do is actually build Veridian completely off-chain. And the way that we actually leverage the chain is in a number of ways. One we are also going through the process of becoming the first Swiss-based QVI. Cardano is the only blockchain that is vLEI compatible at the moment as far as I'm aware. We also have the only vLEI compatible mobile wallet. We are working towards a verifiable smart contract format as well as a programmable token that leverages KERI as the identity layer for the token itself.

You can actually visit verifiablesmartcontract.com and see a report that was put together between GLEIF, Key State Capital and the Cardano Foundation if you'd like to understand more about that. We're also looking at how digital identity or decentralized identity and Sam touched on this a bit in day one.

Every blockchain uses decentralized identity as a mechanism to try and lock you into their chain and to sell their token. And although this initially seems to be a good approach they all fail. They never have interoperability and ultimately we are creating more and more fragmented identity with varying levels of security and recoverability.

What we've been able to do with KERI Is we've created verification across chains just using KERI as that identity layer. We have worked with partners in Canada that are using Corda and we've created verifiability of data between public and permissioned ecosystems.

We've integrated it with IoT modules for signing data that passes through sensors to have an irrefutable audit trail for emissions data on which actual Canadian dollars were then paid out to these participants. But we see this as being an identity that can work across any network, any chain. That agnostic element of KERI is extremely powerful in a blockchain space.

A very long-winded way is we are we are integrating it into the Cardano blockchain in ways that make sense but recognizing that the identity must live off-chain. We have one product as well called Reeve where we sign our financial records and upload them to the blockchain and we actually have had the auditor get their vLEI and they are also signing attestations and anchoring them on-chain.

This is the sort of direction we are moving in as it pertains to the Cardano blockchain itself. When Midnight came out I thought there was a lot of overlap with this. I don't know if you guys are involved with Midnight at all or if that's a sep separate thing. Can you talk to that?

I can. I have a meeting next month. We've had an initial conversation with them. Midnight was looking at DID methods in general. For those that don't know Cardano is broken down into a number of entities founding entities.

One of them is input output. This is a US-based entity out of Wyoming. Initially was founded in Asia. The Cardano Foundation is purposely put in place to ensure that regardless of what happens with the Cardano blockchain that Web3 and blockchain technology drives forward.

The mandate of the Cardano Foundation doesn't say that if the Cardano blockchain was to fail that the foundation closes. Our remit is to just drive decentralized technology in web 3. Obviously we are focused in on the Cardano blockchain. I know that they are very interested in the power of KERI and the value that vLEI brings to a financial privacy infrastructure that they are building on midnight. Midnight is a privacy-focused blockchain that is being deployed by our partners in the States. And we are having those conversations now.

What's your what has been the feedback when you talk to enterprises and companies, corporations about what you're doing, adoption. I'm just curious to see what you guys see out in the in the wild. Sure. Absolutely. Nothing is exciting us more than SEDI. But we are focusing.., what we are building is we're building tech that we can deploy across five verticals. We initially have started focusing in on Switzerland within the financial sector. We're having conversations with financial market infrastructure players in Switzerland who understand the power of KERI or more specifically the vLEI.

They understand there is an efficiency of reusable compliance that is initially attested by their internal compliance team. One of the challenges here is regulation. Specifically in Switzerland, there is a stable coin regulation that will come out in 2028.

There is also work that's happening with the Secure Asset Transfer Protocol (SATP, red) that you'll hear a bit about later today as well. One of the challenges here is you need participants that are willing to take a risk on a technology Until they're mandated to do so by regulation. So many people are open.

We are uniquely positioned to lower the curve of adoption. We are going in and speaking with these partners and being able to deploy pilots and demonstrate how the tech can be integrated into their systems. When we speak with other enterprises What we actually do is we position the value

proposition without delving too deep into KERI and ACDC initially because you can kind of lose them.

But effectively and I think it was but effectively, and I think it was mentioned previously, we need to angle this as a cost reduction or a security uplift. And then the final kind of caveat or carrot is increase market share. The real value or uptick and adoption that we see is when we can actually not only deal with the bank-to-bank compliance and verification but bring the power of KERI and vLEI into the end-client's wallet where you now have this end-to-end Root of Trust. When you start handling digital products financial products you have this very robust compliant Root of Trust.

It has never been more.., I have people reaching out now for vLEI and KERI before we were constantly pushing so it's never been more clear that there is a snowball that is building and building I think the successes around SEDI are only going to add to that momentum within Switzerland alone we are using a KERI-like method for the digital identity program verifiable history and my intentions are to start really driving that conversation to full KERI and using SEDI as the example of what's happening on this side of the world. Did I hear KERI light correctly? Yes, that's right.

Okay. Could you explain what KERI light is? I mean effectively what you have with a 'KERI Light' method is some of the elements of KERI which are easier to implement but not the full five W's effectively they are using what's known as DID Verifiable History (didvh) which is considered a KERI Light method that's sort of the best there's another way to explain that maybe that's a discussion that we could have offline because we don't believe that KERI Light exists in general because you will have to trade off security.

Neither do I. I got another question about this. You said that Veridian is the implementation of 5Ws and we have collected quite a few sponsors to finance the development of 5WS in the foundation as well. I wonder how is that synchronized In what regard? We are building code and you have been building code for the five W's and I wonder how that is synchronized.

We contribute to WebOfTrust. Fergal is a maintainer of KERIA. All right. We're also a sponsor of the KERI Conference. Yeah. Of course. That's the conference and we definitely we believe in contributing as much as possible to WebofTrust, KERI Foundation.

There's a way that we could maybe formal do some more stuff like one of the things that benefits wallets is to have multi-vendor suppliers of witnesses and watcher services. Standardizing on plugins that various vendors can contribute to wallets.

All wallets support multiple vendors of the infrastructure and then they can differentiate on other things as well. that I think that's attractive to enterprises that want to adopt to see that a given vendor isn't just supplying one set of stuff.

Absolutely. Go ahead, Phil. Can I add to that? I think We've heard several times over this course this week from the state of Utah signaling that they can't do this with just one vendor. They need multiple vendors and the way we synchronize with each other is the KERI protocol.

I think it's awesome that these guys are building this great platform that's going to help move KERI forward. That's what we all need. We need more of it. Yeah. The question is more that we that we don't do double work so we create the same code at several.

You almost have to because you need multiple vendors running the code so that people can. For example, we tell our customers, you can't just get your witnesses from us. We need other people with witnesses so that they can they have the security needed from the protocol. I there's some double work, but these guys have contributed a ton to open source.

I think it's working well actually. Thank you. Also, I'd like to give a shout out to Fergal. I have time, don't I? Yep. Excellent. Fergal has been instrumental in the success of Veridian and he's someone who maybe isn't at the front of the room all of the time, but because of what Fergal does dayto-day, I'm able to stand here and talk about the things that are being built and shipped and delivered. But Fergal is Fergal and I went to IIW about four years ago now and we attended a few sessions of Samuel Smith, we attended a bunch of sessions as well. And internally within the Cardano Foundation, you might find this funny, Sam, we refer to Fergal as our Samuel Smith within the foundation. But without his dedication and his willingness to always speak up and ensure that what we are doing from a development perspective is aligned with open source and with the ESOS of caring ACDC, We wouldn't be here today. I just also want to say thank you Fergal for all of your hard work, truly.

if I could just pile on to make him even more embarrassed. Also, his contributions to the KERI ecosystem in general as well as the open source community. He's one of the top guys that we have doing this stuff, moving us forward.

Thank you. Speech, speech, speech. All right, if no other questions, I will end it here. Thank you. Thank you very much.