00:00 It's gonna go to the cloud. All right. All right. We are at the official first 2023 distributed systems reading group discussion group.
00:17 And yeah I am, I am now officially the kind of main host of this giving Brooke more time to view Brooke things.
00:28 And and we have other fellow Visioners around here with James and Brian and Phillips. So I falter in any case, you know, I have, have good friends around and so yeah, this is our first paper.
00:45 We'll get into this in a second. I think for people who are maybe new to the group. Obviously we ran this, I think for the first time last year which I've linked on multiple things, but then the Luma and elsewhere all the previous recordings and chat transcripts and everything that are available.
01:04 So you can go back through those. A lot of last year was an interesting mix of covering kind of foundational distributed system papers among some other more interesting ones around <inaudible> data log and some other ideas.
01:19 This year you know, I, I've only scheduled, we're gonna have one every month, but I've scheduled just the first three.
01:25 So let's see when people start getting more thoughts on some interesting papers and ideas. We already have ones for February and March.
01:36 So, you know, those are ready to, to read personally for these kind of group things, I have to read them lit like the day before, or I forget if I try to read early.
01:46 Unless it was really, really good or I needed it for something. So yeah that's kind of the intro. Hopefully we could do some interesting stuff and people have more ideas or themes or things that they wanna kind of go into.
02:00 And I hope to try to like, have a good combination of, again, more recent research. And then older stuff. And already in the first three months we have, you know, this paper's new.
02:09 The next paper is actually from 1990 about the She Reset, and then we have some stuff moving toward multi here computing, which is something I did a lot of looking at.
02:20 And I was realizing, oh, there's a bunch of new papers that I saw from a some university reading groups. That's back to some new stuff.
02:27 So we're, we're kinda going new and old and new and old. So everybody doing well. Everyone, everyone having a good time?
02:33 23 of these, so, so far so good. Yeah, we just announced our company this week two days ago at the Triplet Summit, which is all about distributed system stuff.
02:44 The paper that you should all be all be interested in is there's a brand new paper published by basil Alka, who is in Samir Al Kwans group at the University of Waterloo.
02:56 It's called Partial Network Partitioning. It will probably probably be important paper that's published this year, and I'm hoping that we're going to cover that paper in a future session of, of this distributed systems reading group.

03:09 Cool. I will partial network. Okay. Cool. That, that sounds actually, sounds really up alley. Cool. Thank you, Paul for that.
03:19 Yeah, and, and we have a you know, for people who wanna add papers, we have the, the Discord Channel. We have a discourse form for that, so please, please add stuff and we'll go through it.
03:29 All right. So for this paper you know, we always, I think we'll continue this tradition from last year. Does somebody wanna give kind of a a kind of intro, the paper that we, that we read on reflections on trusting distributed trust?
03:44 It has a very linking title to a very famous paper, but we can talk about that. So anybody want kind of give it a go?
04:06 No, to me. Is that what you're saying? <laugh>? Sure. Nobody wanted to give it like a quick intro. A quick how many people read this paper?
04:16 You do that? Oh, wait, I got a hand. Listen. You wanna give it a go? Is that what that means?
04:22 Yeah, sure. I can. All right, I like that. I like that. I, I, I, I have a lot more to say on this paper.
04:28 Cool. Go for it. Alright. So sort of gist of it is to have a distributor trust according to the design, you, you you depend on various trust domains and you basically bet on a number of them all giving you same, all, all executing same code and giving you the same you know, values.
05:00 And what they propose here is that one you have to use trusted execution movement, which is always the case. But to have developers, application developers or to make it simple for application developers you use one trust domain, which is like the application developer itself maybe running a local machine, running on a local machine, and then using various other domains and all of them running you know, a secure hardware.
05:46 And the way they sort of design it is in that all these secure hardwares, they have upend only, upend only log, which would keep track of what programs the secure hardware runs, and also can give a hash of the current program that's running on it.
06:10 So whenever a client wants to know what you know, the, the what program is running, or, or once you verify what program is running on, on the server, it's, it's readily available for them.
06:26 And they also propose a way to say do the deployment where they introduce a sort of framework in which the application runs.
06:36 So that is to enable updates, as in when the application developer wants to deploy it. The program in itself will be deployed into a execution framework in the secure hardware.
06:52 And the program is then sealed within, and the hash is provided for anyone for verification. And also the, the code as well as the framework are, are supposed to be open source to increase you know, trust in them.
07:13 And yeah further they sort of put in some suggestions as in how to do updates. So that's again, like only to have the application available to updates use a public key where, you know, for each update you have to provide a private key and verify against the public key.

07:39 And also to always run the new updates in a sort of sandbox environment to make sure it's not sending out a false update or false hash to its clients.
07:53 Yeah, that's, that's sort of what I got from it, mostly the gist of it. Yeah. Cool. Thank you. That's actually really good.
08:02 Yeah, you, you hit on a lot of the right points. And I think, so maybe we can cover this in a couple ways.
08:07 I wanna, and, and for people who maybe even skimmed it would be good is I think this section obviously yeah, we'll talk about audit, audit auditability versus like verification.
08:16 I think that's obviously a big choice that they made here. There's this, actually, this section on two, I think it's a really great kind of view on all the different kinds of applications where distributed trust is, is really key.
08:31 You know, I worked previously at a, at a place that was obviously around blockchain and wallets and things like this.
08:37 And financial custody in this digital space is a major issue, <laugh>. And so that, that comes into play along with some of the other stuff around privacy.
08:45 And that's a big part of, you know, we think about fission too. But I wanna get a sense from like, in the motivations you know, one of the notes I have here is that they talk about them most applications today go with the assumption of non-coding servers.
09:01 Which, you know, having worked in a lot of places, you know there's a lot of people I've worked with who are like, yeah, we run in Amazon, we just trust the network in Amazon.
09:09 Don't worry about it. Which, you know, you never know, right? I've seen bad things happen. So from a, maybe people can talk a little bit about from the motivation of the, of the, of the space.
09:22 And this is a workshop paper, right? So it's really trying to motivate you know, the key thing here is trying to, you know, motivate.
09:29 It's, you know, and I, I think the two big motivations are obviously around the audit log around easy deployment and around you know, really leveraging secure hardware in a hetero of union way.
09:41 Does anybody wanna talk about, like, do the motivations of this paper, are they important? Or is this kind of like hitting from hitting from a field where that's too obvious?
09:51 Might be something, a really good question. So I leave it up to people to kind of talk about, like, from a, from a motivation and goals of this paper.
09:58 What, what do people think? Is there tumor trust and secure hardware? For example? Here, There has been a lot of concern about how much can we trust secure hardware?
10:19 How much can we trust hardware not to inject flaws or viruses? How much do we trust like do we trust the, the foundries given international spying issues?
10:34 I remember Knuth asking about trusting the C computer not to inject the C compiler, not to inject something. It's the same problem, right?

10:42 I, I'll be honest, I also haven't read the paper. But what I'm reminded, what, what I was reminded during the description was, oh yeah, this is also the problem.
10:51 Holo chain tries to be solving with check sums on the executables and guaranteeing that he, it's a perfectly valid concern or obvious do political reasons in a global supply chain.
11:10 So, just to give you a quick heads up here, I was at Apple in their infrastructure team and had a great deal of insight into how supply chain management was an issue for security.
11:21 I can't tell you anything about it cuz I was under N D A, but this is obviously an issue these days and I think this paper touches on exactly the right kind of problems.
11:32 Just so you know, the last three days in San Jose, there's conference on triplet we call a Chit summit. Almost half of the papers and the discussions were about how do you manage the supply chain and how do you deal with the security issues in there?
11:51 Do you use, for example, use puffs physically unclonable functions? That's something you should be in, in your vocabulary now. Yeah.
12:02 That's actually a great point, Paul. And, and Mark bringing up, yeah, I mean, I think, I think this is, this is the like oncoming thing.
12:08 I mean, I remember even you know having done a lot of stuff in, in some of the networking space and there was a lot of workarounds obviously all the SGX stuff out of Intel.
12:19 And I remember we had, for papers we love in New York when I was still there, I think one of the, one of the pa you know we had, I think Jesse Frak was gonna present one of the SGX papers and like the day of her presentation, there was like a pa either a, a post or another paper that exploited all the, and some of the issues that they found, right?
12:38 This is like an ongoing changing environment. I mean, they talk about some of the privacy dns examples there, and it's how hard it's been to actually get stuff, get stuff off the ground which is obviously a big part of this paper moving toward through this kind of audit trail and user and, and client user auditing.
12:56 So I think to that point, yeah, I mean, secure hardware, we're going through a lot of that. And obviously I did a lot of stuff at Comcast in networking space and we looked a lot of this kind of stuff as well.
13:07 And obviously a lot of the pain points that are there. So yeah, it's a, it's, it's, it's pretty interesting, I think where things are going on this front and how much trust we're giving to to, you know, secure enclaves or trusted execution.
13:23 In general. Has anyone else kind of worked in this domain of hardware security model modules this kind of thing at all?
13:34 Other than Paul, I guess <laugh>, but or Paul can talk more about it, I would love to hear, cause I, you know, there was a, you know I you with the, I guess the SDX stuff more how we can leverage hardware, but also, you know, how we control what software runs where.

13:53 But you know, there's obviously a lot, you know, obviously one thing when you, when you, when you think about moving things towards these enclaves, you're paying a bigger price or you're trust, you're moving back to trusting some of the bigger companies like Apple and others who can actually invest in the space.
14:08 How does that work for developers who can't do that or can't afford that and have to do more in software, right?
14:15 That's great. I think part of their full idea though was that was to try to approach it in a trustless way, right?
14:25 And so I wonder what folks think about you know, like their, their sort of approach really requires that the, the code that's running in the enclaves is open source or, or at least public, you know?
14:40 So do folks think that's realistic? I, I there's a very interesting problem here, and that is I don't wanna disturb the flow of the, the meeting, but, but I'm deep in this stuff right now and I have been for at least five years.
14:59 And one of the things that we do is we talk about Byzantine fault tolerance. And the reason for that is because you can have evil things happen to you even though someone didn't intend to.
15:14 So <laugh> it's really important to recognize that, you know, at deliberate, you know, intentional attacks are just one hazard that you have to deal with, and sometimes they're indistinguishable from unintentional hazards that people inject in the system.
15:29 So that's why it's important to have teams like this. And the, the, the z the zhan is is, is leading here.
15:37 Actually explore all the space and, and discuss these topics. Yeah. A hundred, I mean a hundred percent, right? And I think I was in the big one here, again, having looked in the space personally you know, that this 4.2 section right?
15:54 Deployment tomorrow it, this is like a hard from a, you know, you know, so the paper this has a lineage entitled to, right?
16:03 I think that may, referring to this one, mark Anto was the, it's a Ken Thompson paper, right? Reflect reflections on trusting trust, which is one of my favorites going back for forever.
16:12 If anybody's ever read that, somebody can link that. I'll make sure I do it afterwards as well. You know, which was about trusting the compilation and like what if the, what if the thing you compiled and when you read it on the compiler and like issues occurred, they're like, there's, this is essentially this like super inception like world that we can live in, right?
16:30 And, you know, having a lot of friends who look at things in programming language land and compiler, land like comp, where like, oh yeah, sure.
16:38 You, you might have your, your binary that you compiled is safe and secure and doesn't, you know, treat, treat host memory really horribly.
16:47 But when, but the, how do we know the thing you compiled it with, which you've trusted and maybe it's been around for 30 years is correct.

16:53 Right? And obviously I think as we move into distributed, you know, distributed trust, I mean, this is more and more you know, this is much more of like a, a really open space and like it's cool, like AWS is offering nitro and cell bx, which other providers are also giving.
17:12 So people have been thinking about it and paying for it. But it's funny cause when I think of, I mean, this is probably the closest paper to think about, like the open view on this, right?
17:22 Like for, for anybody, to Brian's point where when I've always thought about this space, it's like, yeah, you know, apple <laugh>, it was like the only, these only people who run these things really, really care about this, right?
17:35 Like, how do we, how do more people care about this, right? So I think that's that's really key. Cause they talk about bootstrapping heavily here, right?
17:44 And and that's really tough. And that's, you know, obviously I think to be the biggest like idea really coming in.
17:50 Like, can anybody kinda do this? Even for small small kinda trusted close for computer or computation, whatever the case might be.
18:02 Anyone else kind of pick up on, on some of these things in terms like, what do we think about, is this enough in terms of like having this audit log having these building blocks.
18:15 I wanna talk about that before we maybe look at their, like prototype evaluation. So any thoughts there from any, I was gonna point half on the previous page.
18:33 There are some non goals, which essentially are things that this doesn't do, but are definitely are important, like verifying that the code that you're actually pushing is what you expect in, for example, when it's compiled that you're not getting changes to it.
18:49 So I just wanted to point that out for people that they do mention. There are aspects of this where it doesn't cover everything.
18:56 So we would definitely need to approach those as well when we're doing this, Right. Of course, you know, I think the one thing they hit on with the back door is like the application developer who upon the deployment, who would kind of take themselves out of or just be one participant of a larger system.
19:16 Of course when they wrote that code, they could have put in a back door, right? Which would cause everybody running to, to have to, to make an, which I think is, yeah, these are interesting.
19:26 Those are kinda interesting things, right? Yeah, so the nongo are really good course. Like, I think it's kind of in this paper, and I don't know if this is something to kind of bring up for the larger group, and obviously I'm somebody who works between, you know doing more academically minded work around, you know, less around security.
19:47 But like, you know, I do like obviously this divide of like type systems and people who work in like dependent types versus the language that we think about today and all these things.
19:55 Obviously from this case, it's you know, where do we, where do we look at verification versus something which I think this paper puts forth for something much more practical minus some of the open source

things.
20:08 You mentioned Brian, of course. But I think, I think there's a lot here of like, you know a lot of people have been trying to say like, Hey, before I run something, can I, can I make, can I verify?
20:20 Can I guarantee these things? Versus this kind of ongoing again, audit, audit, auditability, auditability. Where do people see, I mean, like, are we, are we, should we focus much more on verification or are these, are these kinds of, or are these kind of world views this paper showcasing really the direction we should go more practically?
20:40 Or where do people think the divide lies there? I Can give Some call Insight on this. So the biggest hazard that I've seen in large infrastructures and, and think this is true also at at Amazon and at at Apple, at at at Microsoft, is that you can have a perfect system that's, you know, properly audited.
21:06 It is got a tool chain that's verified and it's putting the right code in the system, but then the attackers come in through the network and they basically use a zero day attack to to get from the application into the kernel.
21:20 This happens a lot. And then what happens is once it's in the kernel, it bears itself hides its its logs or, or raises.
21:28 Its logs, the logs that found, saw it there, and then it can sit there for hours, days, weeks, months, sometimes years.
21:36 And when it, some event happens, maybe it's a timestamp, maybe it's you know, an arriving packet or whatever else, and it wakes up on, when it wakes up, it takes over the network inverse controller.
21:49 It basically puts it in promiscuous mode, it listens to all the traffic coming through, and then it can hear anything, any authentications that's happening that's in the clear.
21:59 And that happens a lot, unfortunately, behind a firewall. People assume that they're safe and they're just not. And then what happens?
22:05 It just sends a few, you know art poison packets to the the switch which basically exhausts the the tables in the switch and then all of a sudden the switch starts the same and then it can hear everyone's packets.
22:19 And then it's basically got access to everything in that local network. So the problem is, is not the, you know, is the tool chain getting this executable into a system correctly?
22:31 And are you able to audit it? The problem is that the existing system, particularly the way that networks work, the any to any addressing system in both ethernet as well as ip it causes this problem of, of, you know, other things, being able to hear the packets and these vulnerabilities that are intrinsic in the way that networks work.
22:53 So that's the, the bigger issue in my mind. Yeah, that's actually a great point. And yeah, and, and, and obviously there's like a whole space, I mean, we haven't really covered this in this group.
23:08 You know, I actually highlighted a person here, Panda, we is now at nyu and was a big influence on work.
23:15 I, I had done previously at Comcast. We were doing work around network functions and verified network functions and stuff like that

we run.
23:23 But when I was doing a lot of this networking stuff, to Paul's point, I mean the, the network, you know, there is a whole set of research on trying to verify things.
23:32 I mean, between bag configurations, bad hardware you know you know, distance apologies that we have issues with bgp. I mean, there's only things that come up in the space just within the network.
23:47 And so the whole line of research trying to do that better. And that's, it's very slow, right? Actually coming into practice.
23:54 So at least working at A I S P like Comcast in the, in the United States, like, things were, those new ideas were just not coming that fast to core networking.
24:02 And I can imagine that's pretty much the case in a lot of other places. Yeah. Yeah, that's a good point, Eleanor as well.
24:15 So, does anyone else wanna kind of talk about that? Like where do they see, kind of see these larger problems lie?
24:21 I mean, you know, a again, where I think this is like really fits in the, in the workshop paper cases, I think to Paul's point, like there are just so many, I mean, there, you know, there's no way to kind of fix this.
24:32 I mean, we still haven't fixed Ken Thompson's original kind of view on reflections on the comp compilation. How are we gonna do this with the network and distribute and distribute distributed computing?
24:43 So anybody wanna talk about that a little further before we look? There's always some interesting issues that come to play with these trusted execution environments running things dynamically and performance, right?
24:53 Which I think is a big reason why this stuff also takes the backseat a lot. But did anyone else wanna talk before we talk about evaluation?
25:02 Well, since that original paper about the compiler security issue there are, we've realized that there are many, many more issues with the compilers.
25:14 Even with c there are so many things in, in, in Lipsy and other aspects of the way that the language compiles that this is why we've had to invent languages like Rust because we need to base basically prevent comp programmers from doing things that they don't know are bad.
25:35 And another very, you know, big discussion going on right now was, is with another language which which which is even more strict and but easier to understand perhaps is Zeke, I don't know if anyone's familiar with Zeig.
25:49 It's worth taking a look at. There's some good, yeah, For sure. Some good, there's some good tutorials on, on YouTube.
25:56 But this is really a, a really an on an ongoing discussion about the, the whole compiler and what it's doing to you.
26:01 It's not just malicious things, it's things that you don't know you're doing to yourself sometimes. Yeah, that's right, <laugh>, yeah.
26:12 Things that you, that you don't definitely don't know that you're, that that's happening. Yeah. And you know, obviously there's a

really good space to your point on exploration of languages like Zig and, and rust, obviously, even things like rust to allow for unsafe functionality which is obviously you know and a lot of environments that work with ffi, you still end up doing a lot of unsafe things.
26:34 You hope that you, that the developers write abstractions around that unsafe stuff so that you don't have to work, you know, you would hope that things are correct.
26:42 I mean, but you know, this is an interesting thing cause there's people like Derek Dreyer, this is much more in the programming language world, but they're looking at Russ his group and Zurich I Zurich what school is it?
26:53 I school in Germany? It's but Dreyer's Group is looking at how can they reason about unsafe Russ how can they reason about unsafe properties of rust still semantically being well, like be being sound with the safe stuff, because obviously people are not gonna not stop writing unsafe code and, and rust and things.
27:18 Obviously in this paper there is the use of of here, right? Which obviously comes with some really good properties about sandboxing and stuff, but also brings about interesting things because you, you know, arbitrary really running WAM modules as a prototype shows.
27:35 And then maybe we can, we can talk about it. Obviously we get to performance here, you know, we talk, we, we see here what, what's the increase with adding the sandboxing and, and, and, and press environment execution?
27:49 What, you know, the, the addition here that obviously they're doing they're creating this, this kind of environment to run things dynamically as they get arbitrary code.
27:59 And they talk about this issue. Obviously they're running it with no js where they get this, this kind of huge kind of performance drop.
28:05 Having worked in a lot of places where, you know, good or bad product or engineering or not, where like, everything seems to be performance matters.
28:14 How much are we, where is the trade off with performance and security? I mean, it's really, I think, a little bit of what this paper kind of hints at.
28:24 And when, you know, obviously we want, we won't fully home home encryption. We want all the things to be secure and fast, but how, how do we, how do we, how do we draw the line?
28:34 Or when do we draw the line? Any thought there? It, it's a, i I don't think we can draw the line.
28:45 I think it's kind of like a, a good alien problem that it's going to be constantly unfolding and, and therefore it, you know, there's gonna be this warfare between the people who are trying to get into the systems and the people who are trying to protect.
28:58 There are some big things that we can do in terms of shifting our intellectual vantage point on how to deal with this.
29:07 One of the issues is that networks are managed by apples. You know, and you go into a, a, a router and you, you're trying to figure

out how to manage, you know, sometimes hundreds of thousands of ales.
29:19 This is what happens in a modern data center. I think that's the wrong perspective entirely. What what attackers do is they, they understand graphs.
29:30 So basically they go in and understand what's going on through their telemetry to the outside. They go in and understand what the graph is and then they basically attack the graph.
29:41 So defending that with AKs is like, is pointless. It's, it's, it's the wrong mindset in my mind. Yeah, it's actually a great point.
29:52 And obviously working at fi when you think about the decentralization and where a, you know, looking more toward capabilities over, over ACS is obviously a big thing.
30:01 Anyone else kind of on that at the point's? A really good one, Paul? Anyone else kinda thoughts there? Yeah, I mean, you know, if anyone wants to say more on that.
30:20 Yeah, I mean, well, I mean mark, yeah, I mean, AOLs kind of come from a, a, you know, or acl, they kind of come from a concept of centralized access lists, like how you work on Yeah.
30:35 Yeah. That's where I'm kinda going. Though I do think, I mean, to Paul's point on this too, I think a lot of the things that we've been, that we've kind of been okay with on computing, which I think are things like, like acl, I think of things like how we've, how we've viewed configuration over time, how we, you know, kind of treated, you know, treated, you know, things in networking that we're used to.
31:01 I think a lot of the paradigms that we've been used to are the, are the ones that have now actually made it very, you know, made it more easy to attack, made it really hard for you as, as big store in this paper, in the space of like, you know, again you know, private DNS and, and not ability to show someone's IP addressing query.
31:19 At the same time. These kind of things that I think are really interesting ways we want to go are, are exact, are exactly, are exactly things we've been so used to that make a distributed trust environment really, really hard.
31:34 Right. It's, it's actually the foundational things that we've been just accustomed to that have been problematic. Does anyone else, kind of thoughts on here about, I mean you know, where, I dunno if people work in some of these spaces that were talked about in this paper, but other than Paul, of course, and maybe a few others, where do you, where do we, you know, where do you see yourselves kind of coming into play where, like this, you know, the idea of bootstrapping and this, these issues with deployment and you know, hardware as we talked about earlier, where do you see some of that, like actually meshing with what you do day to day?
32:11 Or maybe not, maybe we're all, we're all cool with the way things are. I don't know. Yep. Go for it.
32:28 Yeah, Like Pascal yeah, no, we're definitely not cool with it. And so the speaking from the perspective of someone who spent a lot of time in traditional enterprise computing and then spent time in cloud computing, and now I guess maybe I'm kind of coming back, you know, the it's a package defense in depth is a package, and it's a different

way of thinking.
33:04 And I think that the, anyone who understands that should immediately, you know, at the beginning of any conversation, determine whether or not the other party understands defense in depth.
33:17 Because if they don't, if they're just thinking about perimeter, then the conversation's gonna be difficult going forward. So I actually, I'm personally, for what it's worth, I'm kind of done with that conversation.
33:33 I, I don't, I have not been successful assisting perimeter based organizations into making a transition. And yeah, I have hypotheses as to why that's the case, but so the good news is that it's now possible to make defense in depth available for free to anybody who needs it, because we've got spiffy for authenticating services and applications, we've got small step and I'm just mentioning these names as basically proxies for whatever other open source projects might be adjacent to them.
34:10 But, you know, we can authenticate services, we can authenticate dns, we can authenticate we, we can, we can, we have high level abstractions for authorizations with like OPA for example.
34:23 And so we can talk about authorization in a way that's independent of the specific implementation. We've got, you know, forward secrecy for a whole bunch of different protocols.
34:35 We've got small step, thank God I think I have a little candle on my altar for those people that are, you know, that have helped us automate certificate management.
34:47 And all that's ex accessible, you know, you don't even have to be, you don't have, you know, I do that with no money.
34:55 I mean, I've been unemployed for months, I'm literally like on public assistance and I've got a home lab with more security tools than any of the banks that I deal with.
35:10 That's All that's a, yeah, and that's a really interesting point. I mean, you know, having seen the enterprise world as well, I mean, you know, this paper talks a lot about not requiring a lot, requiring a lot of cross organization coordination and human coordination or like, you know, the enterprise, you know, I mean, obviously not related to always, always trust, but like, when there was a, you know, I remember a certain places I'd been that, you know it, it would be like when it was like on call, it's like, we gotta get, we gotta get the triage together.
35:42 There's like 40 people on a call and you're like, you know, <laugh>, and that still exists. It still happens, right?
35:47 Like that world's still there. It's sad. And you know, you know, a lot of this, you know, when, when it comes to like again, auditability you know, trying to find a chain of trust to just, with human coordination, it's just not ever gonna work, right?
36:08 Like you know, especially as you get into larger and large institutions is like worse and worse and worse, right? So I think that's a, that's a really good point that things have gotten more, you know, component based as well, which doing really good.
36:22 Yeah Paul mentioned a good thing as well. So that's actually a

really good point. And obviously, you know, one, one question that I want get to at this paper and maybe and maybe, you know, Paul's kind of hinted this already, but maybe people have thoughts is they, they, they offer here some building blocks for kind of bootstrapping these systems, but they also leave this question of you know, will there be a whole different set of build building blocks?
36:48 Like maybe, maybe, maybe this approach is maybe, maybe trusting, you know you know hardware, so your hardware is not the way, maybe there's something else, a whole nother world we can go that also won't just be like, you know, 40, 40 humans trying to like, make it all work, right?
37:05 So do people have any thoughts there? Like if they, if if just given the term like distributed, you know, trust how would you go about building your, how, how would you think that?
37:17 Nothing related to this paper, even from that point, I have a, a comment kind of towards what Blaise was saying.
37:24 So I'm, I'm in this group just like, as a disclaimer here to learn about distributed systems. So I feel like I can't say too much, but, but I do feel just like, like systems, you know, like, like, like you're talking about a, a call with 40 people, that, that just seems like really inefficient to me.
37:42 And, and it, it's just like, you know, I mean, like, you have just, sometimes the best designs are done by either very small group of people or one person.
37:52 I mean, take, take get, for example, you know version control. Just, just like, and, and, and, and Blaze obviously, you know, how's a lot of experience in this, but, but yeah, it's just like, how, how do you get a whole bunch of people and pick the right person or a small group of people to design something like this that's gonna work?
38:09 And obviously there's like a whole lot of political, non-technical hurdles to deal with there. Yeah, that's a great, that's a great point.
38:26 Yeah, and, and Eleanor makes is a great point, right? The extra is killer as well, and the paper provides the way to independently audit, right?
38:35 And so maybe I think, you know, maybe the one thing is whether we dis evaluation of these building blocks are the exact ones, this idea of like independent auditing right?
38:45 Might be a thing that we, you know, should be more, we should be more accustomed to maybe in distributed to, yeah.
38:52 But Mark Antos is like, yeah. You know, whatever <laugh>, what that means, right? Probably and so, so, you know, you know, that's kind of a key that I think on this.
39:04 So though, to, to Brian's point earlier, right? You know, will, you know, maybe this, this paper shows it as open source, maybe it's like some sort of you know, open code within a within a within a, a company or a set of companies or, you know, we're like, we're this thing kind of, you know, will work for, for auto, all the, all the possibilities that third parties.
39:29 I do think you know, people will be companies and stuff who

wanna move more in this direction. We'll be kinda afraid though, to have that kinda openness and transparency.
39:38 That's also always a problem that people are worried about, right? Even if you can't, even if you know parts of binary or black box or something like that.
39:46 So I can tell you actually working at Comcast in the networking space, there's been a lot of money made by like, people not knowing about what's happening in the network <laugh>.
39:57 So, you know transparency is always an interesting thing there. Anyone else kind of on these, like what are other building blocks or what are other, you know, other means you might think about for, for, for building, for building out who straps this?
40:13 Like this, is this completely wrong? You know? Yeah. Paul, please. Is anyone familiar with the work that's going on at oxide computer?
40:24 A little bit. A little bit. They're doing really nice stuff. It's really worth looking at what they're doing. One of the challenges with buildings systems out is basically bio computers from a white box supplier.
40:36 And and all of the issues start in the by and the motherboard. And, and there's a massive amount of, of old, you know, legacy garbage that's, that's in the by that just is, is incredibly vulnerable.
40:56 And so they've taken a fresh look at that and said, how would we rebuild, build a computer from the, the ground up, including the, the bios and the firm war and everything else?
41:05 And I really recommend to everyone take a look at what oxide is doing. The, the CEO o of of, of oxide gave a great talk at Stanford on the E three 80 series about three years ago, I think it was now, or two years ago, and yeah, maybe two years ago, it's worth looking at.
41:21 So oxide computer, Yeah, Brian Kentrell, Brian Kentrell, obviously sun joint, long, long history on things, and they're doing amazing work, big fan, big fan of their podcast as well.
41:33 You really wanna get into the, the weeds on especially on the metal when they did that in the weeds of a lot of these, these things and systems and ware really, really good.
41:42 Blades, you have another point. So so just to, to to one thing that has worked, and this is described it's alluded to in that book called Accelerate That by theora people, that the, it's to change culture.
42:00 It's sometimes useful to move towards blameless culture. It's sometimes useful to describe phenomena in terms of the metrics and not the organizations or even necessarily a status.
42:12 So instead of saying it's something is up or down, you could say that like, the latency went from this to, to that.
42:21 And so maybe like in terms of when you're doing your, your security audits, you can say, well, we had, you know, there was this much average entropy among passwords, and now there's this much, or there were these many factors in use before, and now there's, you know, these many more.
42:36 Do you see what I'm saying? So describe the, do the comparisons

in terms that don't necessarily reflect on any individual person or group.
42:46 That's like a political thing. The other thing I would say, and those of us in networking take this for granted, but to the point about oxide is in software defined networking, there's a concept of control plane and data plane.
43:01 And I would say anybody going into any of anything with distributed computing, that's the very first step is to always understand that there's like the world of con, there's the dimension of content, and then there's the dimension of, of control and all that.
43:15 So in a sense, what oxide is done is they've built the buys is now a data plane thing, so they can load and unload stuff, but it doesn't give you visibility into the content of the traffic.
43:29 It's something the router people came up with. I think That's it. Yeah. Yeah. And, and there's been a lot of work in this kind of control plane, data plane world a software defined functions you know, network functions research has done, have done really have done really well with a lot of stuff.
43:47 You have a stuff like obviously like, you know between either kernel bypass or things that working only in the kernel with with ppf.
43:56 I mean, all this stuff is companies like Cilium as well. We're trying to do the routing, the network there to work with Kubernetes.
44:03 And obviously, you know there was a talk even this kind of supply lines and auditing, supply lines and packaging. I think Kelsey High Tower gave a talk at Strange Loop this past year.
44:14 They say, you know, all the big companies, the cloud companies, they're looking in this direction. They, they want to say, and they want to attempt to provide these kind of things as being a cloud provider.
44:26 I think that's, like, in this paper, they talk about where cloud providers need to go further, right? And I do think that's where you're seeing some of the, in some kinda slow innovation for these kinds of things.
44:40 You know there always comes with that risk, right? If we're gonna put, if we're gonna just go all the way that only these cloud providers can provide these means of of security and, and maybe even some forms of auditing that I use, if I deploy to the, to these providers, what does that mean for people who don't wanna run compute on, you know, do, are we gonna live in just a cloud world?
45:03 Yeah, we're moving closer and closer to this in a lot of ways. Either you're big enough to run your own data centers, you run on Amazon, essentially, or maybe if you're doing this kinda stuff.
45:14 So that's always a, a, a pro, you know, it's a, it's a, it's an interesting step even made by this paper towards Anne's conclusion, but it's also, yeah, it's kinda a scary step its own way, right?
45:25 Like you know so it's a, it's a good point. And again, the oxide stuff, would that be really, really good?
45:34 And again, but a certain niche, right? A ghost head about your hand. Yeah. Thanks. So I haven't been I haven't, don't have very much

experience with the topics paper discussed, but I learned a lot from this discussion and also just from reading it.
45:51 So one question that I had, or that was a little bit yeah, I would like to know more about is if you think that there will be more distributed trust deployments in the future, or that there are other technologies that can address the same problems, or my thinking is zero knowledge, cryptography, homomorphic, encryption, that will kind of take over maybe of addressing these problems.
46:24 Well, the, the, the standard criticism of homomorphic encryption is that it's incredibly slow. And that yeah, in theory you can actually do things.
46:34 But the, the reality is it's really slow to actually do any kind of computations. I think there are some other theoretical angles that you can come into security.
46:45 Most of these are by the category theorists, if you're familiar with that kind of mathematics really and command you to kinda like, think that way.
46:52 And, and that's because you can actually start reasoning about the the proofs and the and the architecture of things in ways without getting bogged down in the details.
47:03 So there's a lot of work being done here. I think the theorists are actually helping us a lot. And I would imagine that things like homomorphic encryption will, someone will figure out how to make it practical.
47:14 Maybe not this month, but maybe, you know, next year or the year after. Yeah. Or maybe in the next 10 <laugh>.
47:21 But, We dunno, that's research, right? <laugh>. Yeah. And to your point, I mean, that's that line again, we talked about earlier, the trade off of perform even with these, some of these hardware secure things and secure enclaves of these kinds of work, that line of where, how much performance, you know, is necessary, right?
47:37 If, if I'm not doing anything in real time, I'm cool to wait longer and longer, but if I care about real time, what does that mean?
47:43 Right? And so, or, or closer to real time or soft. I mean, these are the, this is like that, that line that we keep going back and forth and where all that kind of research is going.
47:53 So there is a lot of stuff happening to Paul's point I mean, there was even a dream. This is more me coming from the PL side.
47:58 I mean, a dream when I was first looking at, you know, a lot of the dependent site type stuff and, and, and, and, and proof languages like agta and C**k, but also really like I was really into F Star at Microsoft and they had this dream and they were the first to have like, I mean, the other languages compile out code the F Star stuff was like really cool with actually the OCaml that it could, or the f I guess that I could compile out to was like, pretty good.
48:24 Like these little demos they showed and what was, you know, the whole dream of like, I can write the proof of my code, my, I can write the proof of my code and also get my code at the same time.
48:33 And then now I know it's like that at least is proven and verified and it's like awesome. You know, I think only Microsoft is

still maybe <laugh> doing that stuff, but that stuff, right?
48:43 So it's, it, you know, it takes, it takes time and a lot of work and a lot of years and you hope to have like one, one good example that showcases like, maybe this can be better, right?
48:55 So I think computing moves seemingly much faster now than it, than it did. I guess maybe that's not true. It just seems really fast because I think people are having reading groups like us today and like we're talking about this stuff where maybe before it was just like industry and academic researchers.
49:08 But That reminds me of of what we, I was in the storage industry for a long time, and one of the things that we're trying to manage for example is backups and, and risk.
49:20 You know, if you, you know, should you be doing backups and which case how often and, and how often should you re restore them to see if actually did work?
49:29 Cuz most of the time our customers would actually find that when they tried to do a restore, it didn't actually work.
49:34 And of course, the last time, the last thing you want is when you have a real disaster and you've lost a lot of data to discover that your backups now don't work.
49:42 The same is true for security. And so I have a there's a, it is a quip, but it's it's an important one.
49:49 We used to say that there's only two kinds of, this drives those that have failed and those that already failed, and those that are gonna <laugh> and insecurity is very much like that too.
50:00 It's like you, how you know, this stuff's gonna hit you. It's gonna, it is gonna get you and it's gonna burn you.
50:05 The only question is you can't say when. And so that's why it's important as systems architects to, to really think ahead and, and try to anticipate not just individual hazards, but general themes of, of disasters.
50:20 Anyway, that's my, my history and Yeah when I knew Paul's, I used to work at Bash and Rio and like we were just when I was in at that, at that place and the group of engineers I worked with, we, everything was about where it could fail.
50:34 We just thought about failure all the time. And that was <laugh>, that was just the mindset, right? Like, there was never a time when things would, would not, would not work, you know, work.
50:43 I guess It was very impressed with the Baio and the Rioc folks. And I attended the conferences and in fact I tried to acquire Baio at Apple.
50:55 And it was I was there and I was, my idea and I actually decided that we probably shouldn't acquire them.
51:04 It was, I am, I won't go into the details but as you can imagine from the structure of their ownership that it, there were some challenges there, but I wanna, I've never seen such a concentrated group of expertise in distributed systems as a baio and, and I haven't seen anywhere else.
51:23 And of course most of the Baio people have, have gone off to the Four Winds or whatever now, and some of 'em went to conference.

51:29 Yeah. But it was one, were the, it was one of the best conferences and the best groups of understanding about distributed systems that I've ever seen.
51:38 Yeah. It, yeah, and it was at a, a time when, when, yeah, research and industry are really tight. We were, we were reading papers, implementing ideas, trying to make it, trying to make people use it like we did with CRD keys and stuff.
51:50 So I agree with you. It was really cool. And I think for people who haven't, the, the conference that Paul's talking about, I think the videos have come back.
51:57 I think they're like poor resolution, but their videos are back for recon, at least some of them. Wonderful. I didn't, I I was so sad when I heard that they'd been lost.
52:05 So the videos are Some, not all, but I think at least over like 50%, like are, are around maybe more actually.
52:14 Oh, I'm delighted. So that's great. Yeah. Yeah. So you can find them if you search Rica, I think they're on YouTube or maybe Vimeo or something like this.
52:21 And so the videos are around like, you know, it was a time that we, we obviously had people who use, you know, kind of more typical database company talks, but we ha we used to invite researchers all the time, right?
52:31 And so that was really a key concept because there was a lot of, you know, intertwining there between research and practice.
52:36 And I think things like distributed trust to kinda end it on, on the snow. When we see a paper like this out of Berkeley, these things are very tied to things, what people are doing in industry.
52:45 So this kind of back and forth between research and academic research and industry should, you know, obviously needs to be an ongoing, ongoing thing, right?
52:53 Like and not just with the big companies. Obviously Google, apple and all these companies are doing that a lot more.
52:59 A lot of you either go to NSD and these big system paper, you know, system conferences, AC and academia. You have papers that are now, you know, mixed between university people and and people from those companies.
53:11 But obviously we wanna see more, you know, more of that at the, at smaller scales I think would be really awesome.
53:16 So Cool. I'm gonna stop the recording. I wanna say thank you. Thank you for Paul for a lot of insight.
53:21 Thank you for everyone else. And you know, next, next time we have distributed reset an old but old but goodie old paper, but a goodie.
53:30 So thank you all and then we'll stop the recording now. So I just put the link in for the paper to the, the University of Waterloo people on partial network partitioning.
53:43 So if you wanna grab that link really worth reading deeply related to all of the things that you guys are talking about.