

00:03 Okay, this happened. All right. So I'm Zha Ma. I am an applied researcher here at Fission. And obviously as I was mentioning kind of pre-recording this is my, my first tuneup in kind of leading the distributor systems reading group.

00:20 I'll be starting more officially next year. Brooke was supposed to be on a plane. I think she might have been snowed, snowed in or something like this.

00:31 You know, traveling at this time of year is not fun. So but she gave me the reins to, to kind of start today.

00:39 And so I'll try to save a few minutes at the end to kind of talk about, you know, next year and get some people's thoughts and stuff we can open up before for that.

00:49 So we'll go there. But yeah I also have a background of people. I think this was mentioned before. I co-founded Papers We Love which is all about kind of reading papers, thinking about papers, obviously not just attributed systems, but everything in between.

01:06 And we run a conference and do all these things. And so, you know, obviously reading papers are cool. It would be cool you know, as I think about this paper actually, and thinking about toward next year you know not only is this about like kind of talking and discussing papers, but if this, if things, it's okay if things, you know, one thing I always tell people when we do, when I do reading group stuff internally is, you know, if you can't maybe relate to where the paper, what the paper is, particularly like from an academic perspective, we can talk about what that, what it means academically.

01:43 But we can also start talking about, Hey, what does this mean for, like, what is it akin to something that I do or work in practically or, or can relate to.

01:52 That's a, that's a major part I think in, in reading group discussions. It's something we've used a lot in, in our, in our papers.

01:58 We love kind of model. So without for Ado, today's paper was a, on the shorter end compared to some others, we had merge what You Can, can Work What You Can't, managing Data Integrity, local First Software.

02:13 And so just cuz we're on a recording, you never know, just say the author's names Nicholas Cipher, Jeffrey Lit, who has been doing a lot of great work with In and Switch, which are, which you're friends with are as well.

02:24 Well, he's been, he's MIT and of course Daniel Jackson, who was just here at C M U a couple weeks ago, but I had to miss it cause that was in Portland.

02:32 And Daniel Jackson's really well known for a lot of work and some famous books on software engineering and, and, and various principles.

02:41 So this is coming outta his, his lab at m i t. So as typical who wants to give the first kind of like quick synopsis intro of this paper?

02:56 I could go for it. So this is the, the, the very high levels synopsis is they're kind of contrasting merge at Bright versus merge at read, like the, the, a lot of CDTs and concurrent systems try to, the moment you write something, okay, how can we make this into a consistent view that then everybody can read.

03:24 And here what they're trying to do is, well, it's not always appropriate to the, the unicity of view is less important than people having exactly what they

want.

03:38 And that means each doing their things, sharing sometimes and, and getting the result of other people's actions sometimes, but really not losing what they've done and not not getting anything unexpected from others.

03:54 So there's subtle case of merging that don't work well and subvert expectations. So the idea is to design a way to think about event causality.

04:06 How can we do partial ordering of events? And the partial ordering of events should yield, like, again, if it can be merged, if the events are independent, then you can just assume there in one causal chain, in whichever order.

04:25 Though of course you want it to be deterministic but it, the order doesn't matter. That's the whole point, and that's the merge what you can.

04:33 But if there is conflict, then creates something like a branching model so that I'm know, I'm looking, I'm continuing on the consequences of my actions, and if I want to integrate with the consequences of someone else's action, and there's been divergence, let's make that an explicit merge operation.

04:52 Now the merge is a bit more complicated because it's after the fact, but at least you don't have surprises from changes from other people.

05:00 So that's the general goal, how it's accomplished. Should I continue in <laugh> the next level of the pyramid? Or should I say Yeah, Actually I think though, I think at least saying how they, how the ba their, their kind of base form works would be really good.

05:15 Yeah. Yeah. So the idea is each event, okay, so there's the, the proposal is really the section four. So there's, there's a lot of very interesting stuff before that about the consequences of various writing models and figure, figure one really goes into that, and it may be worth getting back to that, but the core of the paper is this notion section four, and you have each each event has okay, first there's a version.

05:51 Version is a single consistent ordering of events that gives a consistent view of data. So it's not just a sequence number, but each event has a sequence version.

06:03 So this event has this number, and again, sequence versions are partially ordered. It's not a total order, there's no it's not well time clock.

06:17 And then there's a notion of conflict set. So what did you read? What were you reading? What state were you looking at when you made that change?

06:28 And what data are you changing? And there modeled the world is a key value set. So like, if you have multiple tables, then j j j just means a longer key, but the keys, which the keys make up the substance of the conflict set.

06:48 And what you're saying is, I viewed at this key at this point, and it's last modification time, not necessarily the time at which I look, but the last modification time is this.

07:01 And now my event starts at this sequence number and I kind of committed at this later sequence number that has to be later than all the last red and the event sequence number.

07:15 And then you can see that two events are conflicting if they both have like the, the informal version just before 4.3.

07:27 The conflict exists when another event wrote to a key that an event read before it wrote to it. So you're reading, you're writing, and somebody else wrote here.

07:37 And, and, and basically the point is to avoid that situation and to do that you order the event in any which way that pretty much obeys this constraint.

07:48 The, the, this what they call, that's what they call history, serial serializability, no conflicts. And the causal order that is your event comes after everything it reads for, from so to speak.

08:03 This is a lot of freedom and they point out it's a lot of freedom. And you, there's other reasons where you want this to be deterministic, but those are the only constraints they put on.

08:12 Let's order the events into a tree. And from there there's a component that transforms it into a tree, and there's a component that says, okay, well given all this, let's materialize, here are the key value of your version on that branch.

08:34 And then this is the core of the idea. And then they get into what would a user do to resolve things?

08:45 And it may mean sometimes subtle operations of undoing in action and reapplying the action. So this is there, this is more constraints on the users, but it's a promise of no surprise for the users.

09:03 So of course there's a small cost and then the section five goes into, okay, can we do a classic C R D T with this?

09:12 Can we do a distributed version control system with this? And this gets into the details of how to reproduce something like C R D T or c V s and specific C R D T, like the last writer wins and things like that, or, or app and set.

09:32 So again, there's another layer, but this is a good stopping point. Yeah, that's great. That's great, mark, actually, yeah, I, I think, yeah, that there's that line here.

09:40 If I, if you could see my screen, sorry, but you put it less precisely, A conflict exists when another event wrote to a key, an event read before it wrote to it.

09:47 Right. I think it's pretty, pretty well put. And yeah, get their, their Treat DB proposal. I think that's a super interesting kinda, you know, it, it's a, there's an interesting part later in, in section five that talks about simulating other CRDs on top of the tree db proposal, which is also really interesting.

10:06 We can talk about that. I was, that's almost to me like another thread. But yeah, I mean, the big thing in this paper, and I wanna to give people's thoughts, you know, it'd be really good to kinda get, I, I think the pros and cons on this is like, you know, if you think of it just from this photo sharing app which they, you know, they talked about this essentially with this idea of of the gardens, right?

10:29 This kind of work different than databases, it's different than things we might be used to, we've seen SIR apply, or it's even different than textual based kind of collaboration doc stuff that we've seen a lot of the CRTs applied to.

10:44 But they're talking about this idea of these, of these gardens and, and like photo sharing apps. It, there has been a lot of work in the past.

10:51 You know, I've worked on stuff where we gave we gave users the opportunity to determine how they wanna resolve conflicts in database times.

11:01 And that did not go well, <laugh>, nobody wanted to do this. Maybe, maybe with these kinds of gardens that they're talking about here, maybe that's a different paradigm with local first software.

11:13 So I wanna get some of people thoughts there that they want to see what, where people think these are really the kind of pros in terms of how to work with stuff.

11:18 Quinn, please. Yeah, so first off, I love this paper. I think I was the one who suggested it months ago.

11:25 And great summary, mark Anon. Just to add a little bit of context to what you're saying, something that this paper does that I really love about it is that it sort of breaks with the tradition around sir DT research from the way that I see things, I guess, where a lot of earlier series research is focused exclusively on correctness and about whether or not the system converges.

11:47 But then what they do here is really try to add a lot of slack into the system with these great conflict conflict sets and right conflict sets that you were saying, add a lot of flexibility that allow you to move away from just this like, binary notion of either convergent or non convergent.

12:02 So you can focus on user intention. And I think that that's really the big insight and benefit over the approach that they're bringing over a lot of these possibly much more simple binary cr RDTs, like last rider winds, registers and so on.

12:19 Sorry, mark Avan, I see your hand up. No. At, at, no, no, let's finish first. I'm just next. Okay. So I, going back to Z Sean's question, that's really where I see this paper shining when you start having these CRTs that maybe are not going to be managed and operated entirely by automated systems, where you wanna start introducing humans into the feedback loop to determine exactly what is happening when conflicts occur.

12:52 And obviously the digital gardens that they bring up are a really clear example of that sort of thing. But I think possibly a more concrete use case that might be more at hand for a lot of people here is a lot of the stuff that we see nowadays in the Fed averse, and especially with systems like Master, where and I've talked about this a lot in the past too, but where we have all these different communities where different norms around things like content warnings that maybe disagree about how they should be applied, or to what extent they should be a part of the culture, but where it possibly makes sense to have systems that allow you to layer on all of these different, like intents on top of the underlying data.

13:32 So that's kind of ly, but that's kind my high level perspective on the paper. I, I, I, I totally support this Quinn, I know you've been talking a lot about post-modernism in database design.

13:47 And for me, this is really I've, I've been taking it from an another angle, like the notion that facts have a truth value.

13:55 It's they don't have a single truth value, they have a social truth value, meaning they have many truth value for many people.

14:01 And this is very much going in that direction if we are dealing not with the unified truth, value for a specific sub-community use case that needs immediate convergence, and that's a use case, but sometimes you want to have a picture of where a more diverse community lies, and then you want social, what I call social truth.

14:26 And this is very much in that direction. I think it's extremely important. Mm-hmm. <affirmative>, I like that notion of social truth.

14:33 You bring up, when I think about systems like this, I, I had a really good conversation with song recently about basically exactly this, where when you have events in a distributed system, really the only thing that you can say with certainty is that those events were emitted.

14:50 But on its own, that doesn't really tell you anything about what the intent behind those events is. And like in the photo sharing app, for example, we know that Alice has uploaded a picture.

15:01 We know that Bob has uploaded a picture, and we know that they have performed different operations against those pictures, but the events themselves don't actually tell us why they performed at those events or what they were hoping to pull out of the resulting images.

15:13 And it's, that's sort of extra flexibility. And with it, the complexity of something like 3D that I think allows you to like, reconcile and make it explicit, that sort of intention, I thought that point in particular was really interesting talking about user intent.

15:34 One of the things that it talked about also. So from that photo example I think there were like two it was like a product two of like two different, like, pieces of data that it held onto oh Yeah.

15:48 The cont is what did that mean again? And saturation and 2.1, but effectively, yeah, there were two dimensions that you could attribute to a photograph.

16:05 It's kind of, so the mental model that have been forming in my head a little bit is that each one of these CDTs is an indexed.

16:14 And so it's got a bunch of dimensions. I, I'll use dimension and space and type interchangeably. There's this really cool proof that comes from a lot of the homo topi type theories community that talks about the fact that spaces and types are literally the same thing.

16:37 So the mental model that's forming in my head is that you've got a bunch of these indexed types, indexed meaning that it's got an attached key set that keys into that space, into that type.

16:50 And it talks about editing so you can edit independent, you know, that you can edit independently if you have unique kind of spatial dimensions and you know that you don't have key collisions within that space.

17:05 And then also the vector plot arithmetic, knowing like when you can have causal ordering and kind of the human intention going back to human intention, the messages that we send to a distributed system about what we wanna do on the space that we're interacting with, right?

17:23 Let's say that it's a photo space, photo space. I've got like a red channel, a green channel, a blue channel, alpha channel.

17:29 And I'll say that simply so I can edit each one of those independently. When I say something like Brighton an image, like that's a message which needs to be interpreted by my distributed system or my, whatever, my programming environment, and then translate it down into the, like, we have to do this dimensionality reject reduction to take that message and translate it into some representation layer of how it manifests in whatever space that we're playing with.

17:55 I thought that was kind of interesting in terms of like, the fact that there are a ton of different ways, right?

18:00 Because it's a lossy operation. There are tons of different ways to interpret that. And then it, it, it gives like a kind of an interesting way to think about it in terms of like message compaction by message compaction, I mean like how do you take a bunch of like input intent messages and summarize them all into like one structure where you can guarantee things like of operations.

18:25 That was kind of interesting to think about. Cool. Yeah, sorry, I know that was a bit unstructured. No, it's good.

18:35 I think they're good examples here. I mean, I think photo sharing it is actually maybe more and more has become the one of the kind of canonical distributed you know, kind of distributed problem problems that people have used.

18:48 So I think, I think it actually is a really good example thinking about the kind of lossiness, as you mentioned, encoding and stuff, who yeah.

18:55 And maybe anyone else? I don't see any raised hands, but anyone else kind of, Oh, Go for it. Yeah, so I really, so one, I really like that model of like changing the, the like state space you're operating in from like user intention down to like like pixel representation on screen or whatever.

19:15 That's really neat. And throughout reading this whole example, I was constantly thinking of like, wait, why don't you just store the operation separately from the picture data?

19:22 But that's <laugh>, that's just like, they just needed a, an example so they could talk about it. But one thing specifically to z Sean's like opening question this paper presents a really good, like low medium level model for like, thinking about how to model this stuff and extremely punts on the, like, how do we service this to users in a way that does not just make, make them extremely frustrated with their computer?

19:52 And it feels like we probably need like whole entire new metaphors in order to be designing applications that work like this that we haven't really come across.

20:04 I mean, maybe they exist out there and I just haven't stumbled across them, but yeah, and like thinking about, this is not too hard for me, but I have been like elbow deep in computers since I was a child.

20:16 And so I've got a severe case of computer brain and like, I'm, I'm used to thinking about stuff in this way, but I suspect many people are not.

20:24 And like, just being like, Hey, you're in a different universe now. Try and resolve it. Maybe if you want to.

20:31 It seems like a a hell of a thing to throw out a user I have, but Emma, you have your hand up, so if you want Oh yeah, no, like that kind of like followed on with that.

20:44 And also just kinda like some of the human computer interaction stuff. Like this paper gave me anxiety, <laugh>. Oh. So it was like, it was kind of like a roller coaster, cuz I was kind of like reading it more from just like the perspective of like, okay, like what, what is this?

21:03 Like what would it look like to be a user in this world? Like, you know, like what, what, what would, what would be exposed to me?

21:13 And like my initial reaction was like, oh my God. Like I, I don't, like, I have, I have gift branches that are just like hanging around and then they like disappear and I, like, I just forget about them.

21:27 And like, I don't want, like, I don't want this to be like, in more aspects of my life. But then it was like kind of after the whole like digital gardens metaphor, like, I actually kind of like, was like, okay, wait a second.

21:40 You know, like, this actually might be okay to like, not just have a single canonical like resolution there. And I was like, that was, that was kind of a weird, like freeing revelation.

21:57 I was like, okay, you know, like, like may, maybe this doesn't matter if these ever converge again. Like that's, we just, we just, we just go <laugh>.

22:11 But yeah, and then like the anxiety like went back up again where I was like, okay, s**t, they're really punting on this like, like whole like, language for how it's actually going to work.

22:23 Like, oh my gosh. Like what, like what is a UI for, like presenting these differences between these sorts of things and like, how, how do you express that?

22:36 Like how yeah, like there's so much that it was just like, I don't know. So it was, it was a roller coaster of anxiety <laugh>.

22:44 Yeah. So you know, I think I had the same anxiety I said of yeah, I have a lot of friends in HCI here at C M U and I think this is the first HCI program ever, right?

22:55 And they would, they, they would've crazy anxiety reading this paper. And that's why, you know, one of the things is they come from the academic perspective, but this is a workshop paper at a distributed systems workshop.

23:05 This is not like a top, you know, whatever higher tier peer reviewed paper. And that's one reason it puns on a huge problem.

23:13 Probably the other paper this could be is actually mostly about treat db in

the simulation of s sc RDTs, that could be more of like a systems paper that I think could be its own accord.

23:24 So that's why this is again, fits in this kind of workshop that's presenting ideas. And I think, you know, the merge, the right seman, the, the kind of right view on this as you talked about Mark, is, is exactly what this paper's through line really is.

23:35 Cause the ACI part, yeah, it's completely thrown out. You know and obviously with treaty B being based on a lot of the kind of get graph semantics, the issues there, that's a programming model.

23:49 That's something that programmers are used to, right? Which you know, I can tell you a lot of people in ACI where, where this would really falter is a lot of the ACI stuff for good or bad is gonna focus a lot on no code or low code.

24:04 There is a lot around tooling and developer tooling and stuff, but, but one of the things is that, yeah, I just, being based on a GI model already throws out a whole set of users who wonder reason about photo sharing apps.

24:17 So yeah. I think next was Mark. And then we'll go through the list there. I, I'm, on the one hand, I agree like most users, I mean, so many users can't understand trees, forget graphs or branches, and that's fine.

24:35 On the other hand, most of us have theory of mind that is we know that somebody else understands something else and may know differently about something we know.

24:46 And so I think making it very user-centric, saying, okay, this is Bob's view, this is Alice's view when here's you, let's see it as Alice in this notion of delegation.

24:57 Like I remember many CMSs or saying, okay, now I want to use so-and-so's permissions or, or, or views to act as a delegate.

25:07 I think this is something that is not totally outside the pale of user's mental model. Now the merging is more demanding, but saying, here's your view, here's Alice's view.

25:22 Do you want them to remain separate or do you want them to, or do you want to create something that integrates the best of both and propose it back to Elvis?

25:30 And, but I think there is a dialogue that can be done there. But yes, exposing, I agree on the one hand, exposing people to the tree metaphor is a non starter for a lot of users.

25:41 I'm like, what's wrong with git? I use GI every day. <laugh>. But yeah, no <laugh>. But but there are ways to the, the, the, the subjunctive universe.

25:53 Like what if universes is a very basic human metaphor that we all have in our brain over Think great. I think next.

26:12 Oh, okay, Philip. Okay, cool. I think you were next. Yeah. Oh, sorry. Yeah, kind of going back a little bit to the Edelweiss paper a little bit.

26:22 So in terms of communicating intent so one of the innovations of the Edelweiss paper was saying that, so you've got your message a d t, right?

26:30 Just a tree of possible like, like a grammar or here are all the valid

messages. And one thing that edis said is, all right, let's take the ability to snapshot the representation layer, right?

26:41 So you have your state space of all the different value dimensions so key value dimensions that you wanna be able to edit.

26:49 And then adding a message which says, Hey, take a snapshot of the representation layer and broadcast that out to all of the people in the distributed system.

26:58 So there are messages which say, okay, here are operations that we want to take, which need to get flattened into this representation layer.

27:06 And then you've got a snapshot thing. I think that's a really interesting UX point to make users consciously choose to snapshot something, right?

27:14 It kind of reminds me if you guys have ever worked with like video editing software where you kind of recompose or ize and then you lose the ability to have a lot of the more precise like, like you take a vector image, which says the description of how you want things to be and then you ize it out into like snapshot and then you're able to make further edits, but you kind of lose a lot of the causal history.

27:38 So what that's doing is it's forcing compact, a lot of distributed systems try to like do the compaction of like messages into like a rasterized quote unquote like representation of the space smartly.

27:55 But it's kind of interesting to think about those as two separate things and imagining what it's like to have the users have to consciously choose when they want to kind of recompose ize take a snapshot.

28:15 I, I find the, all of the kinda the, all the points that were raised here and all of the discussion points are interesting.

28:22 I feel bad about like bring up something else again. Yeah, and I don't know exactly, I, I forgot how this relates to something that someone else said before, but I was thinking about the the, the title and how it said merge what you can and fork what you Can't.

28:43 And I was like, but can you think of the forking as a way of merging? Because in a way you could say that the view of a bunch of forks is the same as like, let's say a map from every participant's view to their state, and you just merge all of the, the, the forks together, right?

29:07 And so the data type as a whole is like, is like a set of branches in in in the good sense.

29:19 Yeah, basically that's, that's one thing I wanted to raise and wanted to know what what you think about that. But that's basically it.

29:26 I don't know, That's actually a point that ties in really well to what I was going to bring up, which connects both back to the HCI stuff and a little bit what you were saying here something that I think isn't super clear in this paper is that it's called merge what you can Fork What You Can't.

29:49 And so there's something that that's a little bit lost in the analogies to get, because in get, when you make branches in, in a repository, you eventually wanna merge those back together usually.

30:00 And I think that's something that this paper is saying isn't necessarily the case, especially in these cases where you have these like tension conflicting operations.

30:12 So I don't think it's a pert analogy and that a lot of the things that make it challenging to use even for us as engineers aren't necessarily going to be exposed in a system like this to the end user.

30:27 And just as an example of what I mean, for example when it comes to surfacing a lot of these a lot of these cases to the user, we don't actually care about the hash of the branch that we're dealing with.

30:42 For example, we just hear about somehow visually distinguishing the different branches that we've created from, for example, the files that we're operating against.

30:53 And again, if you think back to like the Edelweiss paper where you have, where you have these, this like explicit causality tracking, that's essentially what we have here with third read and the right conflict sets.

31:05 And kind of the way that I've been thinking about these things, especially with regard to the database project I've been working on in vision is by considering like the directed acyclic graph of the content, hashes of all these events in the system we're dealing with.

31:19 And that allows you to derive a new hash for the current state of a branch or for a sequence of operations and so on in terms of everything that they depend on, which means that from that hash, you can use, for example this hash guitar library that I'm a big fan of to make these really beautiful visualizations of the hash function that will be stable and consistent and something that you'd be able to display in the ui, for example, so that a user's able to recognize that these two files represent like two different branches of the same base file.

31:52 They don't care what goes into those hashes, they don't need to communicate those hashes to each other, but they're just something that offers an accordance in the ui, but then that a developer or someone would be able to drill into and see what exactly goes into that for like de bargaining or audit purposes.

32:13 So like, I'm not saying this is the solution or anything like that, but I think that there's a little bit of nuance and that yet analogy that changes things slightly.

32:24 Yeah, that's really good point. And they talk a little bit about the analogies to get, but not this one in particular I think is a good thing to bring up.

32:32 John, I'm trying to find my mute button, so I'm not as knowledgeable about Systems terminology and stuff, but I, I'm just thinking like a little bit just researching AI recently and just thinking like may, you know, maybe like AI could help with some of these like merging decisions like maybe that you do some machine learning on, on basic intent or really common use cases, or even customized.

33:01 I mean, I mean this is like, oh, just, oh, this is easy. Just throw ai <laugh>. Obviously it's like really hard and involved to do something like that.

33:07 But the other thing, like, just thinking about it critically, like the example they gave is like, is there, and I think Quinn maybe mentioned this, but like, is there intention to merge these back together and there's like a space issue.

33:20 Like you have to have like two copies of this cause it doesn't merge as well as like a textual with like get, you know, where it's like, oh, all these lines are the same except these, and so you can't really it very easily.

33:30 But yeah, but some interesting things to think about. So at least in the photo example, the end, the point there is not to merge them back together.

33:39 And in that case it's like more likely that from some base image you're trying to produce two different images that you might wanna put it into different photo libraries or assign different names.

33:51 But there's not really like a meaningful operation that involves combining like a low saturating image with a high contrast image other than deleting the other one.

34:04 Paul agreement. I thi I think it's, it's perfectly appropriate to, to when there's a, a merge point you've shown to the user, here's two versions, do you want to merge them?

34:16 And then the AI can help, here's a proposed merge and that's perfectly fine. But, but the, the whole point of the article is don't do this behind the user's back <laugh>.

34:25 Do make sure the user's involved in the merge operation. And, and Yeah, I was gonna say the title, the other part of the title, we haven't talked about managing data integrity.

34:35 That's what you're talking about <laugh>. And that point is so important. Like we, we spend so much time thinking about how to do the merge and making sure that the users, it's, it's a no surprise.

34:49 And that's a fundamental H c I principle, like make sure that nothing happens that surprises the user. I mean, if the user is aware that multiple people are working on this, it's perfectly fine to have a signal, Hey, this is happening and what do we do about it?

35:05 <laugh>, but detecting conflict is useful, but, but resolving conflict too early is a, breaks user's expectations. C c can I bring another point in or is, is it someone else's turn?

35:21 If you wanna start maybe if you wanna, did you wanna reply to Oh, yeah, yeah. I had a, I had a a like small quick reply.

35:29 So in the, like in the earlier half of the paper before they get into the treaty B stuff one of the things that they call out specifically when they're talking about like why forking histories are useful and good is that they give you like availability cuz you can edit when you're offline.

35:44 And then data integrity, like you mentioned, so that like the, the auto merge won't cl like your changes or merge them weirdly.

35:51 So now you've got a desaturated high contrast picture or whatever and the like conflict resolution can be deferred, like specifically that you can say like, nah, we'll deal with that later.

36:02 And that was like one of the, the things that seemed super important to me was the like, here's why. Here's here is the, like whole reason why our forking histories are a big deal and it's useful to think about things in this way.

36:14 And so it's like specifically the like, Hey, we can defer conflict resolution until this there's a good time to do it, or we can work out a way to do it that like, works for everyone or whatever seems like a big deal.

36:27 That's, that's all. Cool. Yeah, I think that's, those are all, yeah. Yeah. I mean, I, yeah, I think the, the not clobbering your data part is really important here.

36:41 And actually maybe the biggest through line, right? Minus the, the hand wavy how it, how you actually manage when you come back.

36:46 Maybe you never come back. I think it's something you said you just go off and you do your own thing, right?

36:51 Went from some, some, some sort of point. Emma, did you wanna reply to that? I know Mark, you had another point, but Oh, no, this was kind of a, another thread that I had just come across that, Yeah.

37:02 All all right, let's start with Mark and then We'll go down. Okay, I'll go very fast. One thing that did worry me a bit, and maybe it's my misunderstanding, but when I read the sequencing algorithm and that there's a lot of freedom and there's a lot of ways it could go, I had a bit of worry, is it possible for a event to come late and reorganize the whole free so that suddenly your, you get a very different view because of sequencing things.

37:33 Now, I don't know if that's possible. I, it requires hard thinking about the, what the constraints do entail, but I just wanted to say that I had a bit of this slight concern about can, can the sequencing freedom lead to some surprise also?

37:50 Yes, <laugh>, Yes. That's you thought about it more. I did, I spent so much time this year thinking about things like differential data flow and this is also why you end up with things like an aase, having those high and low water marks for when you kind of limit causal stability and stop allowing old events to be merged and so on.

38:13 Yeah, That's great. Absolutely. Yeah. Okay. Thanks for confirming. Definitely. Did I think it is a concern? Yeah, no, like, so one that I had also just kind of realized is especially with this being like this, this might be another just like c i topic, but like, what, what would like actually your notifications look like in this sort of situation?

38:45 Like, because I feel like that wasn't like talked about at any point in this, but like, that's kind of a fundamental thing of like, oh, hey, here's, here's other versions.

38:57 Like, do we, do we give like notifications for every like kind of like forking event? Do we, like, at what point does, do we just like not overwhelm a person with just like, Hey, somebody, there's this other version, here's this other version.

39:13 Like, hey, do you wanna merge this? Like, yeah, I don't know. That one just

popped into my mind and I was like, oh, s**t, <laugh>.

39:20 Yeah. Yeah, that's a good one. I mean, obviously I think if you take the GI model, it'd be like a thing, even if you're offline for a long time and you're like, okay, I'm gonna fetch now, I've been going on my own thread and now I'm gonna ask for things.

39:31 That's one way. But when you ask for things, which I've had, you know, if you go like a, a major Rebasing event or something like this where you've asked for something and now you're like, you know, your 400 commits behind because something, you know, thing blew up and you, you haven't been thinking about it. 39:48 And you could, you know, that's also a bad experience as well. So yeah, I mean from an, you know, yeah, I mean, yeah, it's an interesting point.

39:55 I mean, it's another thing where like, yeah, there's no, there's not really, I mean, so there is a lot of work and actually from MIT to different professors on and other ACI people, but like programming by example or example based programming, you can imagine some sort of like, Hey, I have a, I have the new version.

40:14 Here's an example of what might change and you can actually demonstrate this. And cause you can think of like, I guess like example by not programming, but example by merging or something like this.

40:27 But you know, these are obviously very difficult UI questions <laugh> to to, to dive into. That's another good point though.

40:37 Yeah. Anyone else on, on this kind of thread or, or a little bit to, to maybe Quinn's thread as well on these kind of differences for how we think about it?

40:48 Or, or I think Mark's thread is also really interesting that, you know things the, yeah, things can go awry <laugh>, it's a piece of this.

40:58 So I dunno if Quinn if you wanna talk a little bit more about that and stuff you've kind of seen kind of from a practical perspective, there're Oak Phillips, I think a little bit is also on the on the HCI perspective is I think they shot the, out in the paper, you basically can opt in into forking or not, right?

41:18 Depending on if you want to add reads to your conflict set or whatever it was exactly the, the, the keys that were captured in the events.

41:29 So technically you could have, well I'm creating this bulk action, but if anyone overrides this, it's fine. So Yeah, as part of the, the simulating other CRDs and then therefore you can just follow another crdt essentially, you know, deterministic pattern for how you want to do things.

41:45 Like with, with the ad Removes example was another one, right? With the or. Yeah, but like at the same time, so the system, the underlying system is very flexible.

41:56 How much of this flexibility do you really want to surface to the users? Another question, I guess <laugh>. Yeah. Is this just a intermediate place, right?

42:07 And then on behalf of the users, you say they choose some option and you can either simulate one of those kind of other C R D P types or you present some interface, right?

42:17 Or I mean, I think to, to John's point earlier, I actually do think the world of

AI essentially, or whatever the case might be with, with these, with modeling things this way is actually gonna gonna be that thing.

42:28 I mean, databases, you know, one of the bigger databases, and, and, and weirdly enough again, not to bring up cmu, but is a database I saw just talked about so much more lately is the stuff by Andy Pablo's group on auto, auto autotune, I think it's called Autotune db.

42:46 Which is essentially using machine learning modeling to actually have the database behave and work with certain heuristics on behalf of the user on behalf of how you tune databases.

42:59 This is really huge work. Pretty much, I mean, like everybody who work, I think autotune just got bought, like he's a professor and with students and they got bought into something.

43:08 So it, it's become like a major thing. Cause I've even talked to people at Lyft and other companies who are like, we're trying to apply these same ideas that we see from UNE to how to think about distributed data.

43:17 So in some ways it's like, you know, again, this, this idea of putting everything toward a user model is great.

43:24 And it, it, you know, for locals first por, we talked about it, but then I think a lot of people are just like, cool if the machine, if the machine knows what's better, <laugh>, I, I can do leak code and chat d p t, you know, maybe it's, maybe, maybe, maybe most users will be like, cool, that's great, unless I have to think about it, the better, right?

43:40 Which is I think the, always the hardship of even for developers and then going to the non-developers. That's a pendulum, right?

43:47 I said when we were at reoc, we gave people the ability and the da, you know, the da the great database of the mid two thousands.

43:55 We gave people the ability to determine how they wanna resolve their conflict for very basic key value stuff. And they did not want it.

44:03 This is why Mongo did so well, cuz Mongo said, yo last rider of wins. And if they just went away, it was cool.

44:09 And that was okay. Like, I mean, it, it was, you know, it made a decision on your behalf and you were just used to it.

44:14 And you would, the programming paradigm at the time was, oh, if something had changed, I was unexpected. I'm just gonna go grab it from memory or I'm gonna do something else and like reapply the update, right?

44:22 We're used to this kind of way to kind of reevaluate things when it's not for the expectation. That's like so much a programming.

44:29 It's called like, maybe that you call it like defense, you know, defense based programming or something like this. Yeah. And then SHOs came back again.

44:37 Yeah. And the SHO is great. I always go back to the SHOs still, it's still so useful, these reasons. So anyway that's another point I wanted to bring up to that.

44:46 But yeah anyone else kind of on this, on the through line? We'll try here

for five minutes before I, I talk about next year, but yeah, defensive programming.

44:55 Yeah, exactly. I think ki yeah, I think one thing that's worth keeping mind, go ahead. Oh, <laugh>, sorry. I think one thing that's worth keeping in mind given that we're talking about, like what users can or can't or will or won't do, is that like these are entire people and they're like capable of, of learning stuff and figuring things out and being smart in their own ways.

45:20 And like, it's very likely that very frequently people don't want to deal with stuff because like the interfaces of juice or our metaphors are bad more than, like, it is just inherently awful to interact with or whatever.

45:35 And so that's worth keeping in mind, I think, And quickly to that, I think this way, what I love about the paper is it's a kind of meta system and people can figure out new ways to use it.

45:50 I mean, everything can be improved by a layer of interaction, but in this case, it's a layer of abstraction and I'm all for it.

45:58 Which also reminds me of like the, the, let's use the topple database and data log stuff underneath again, and it seems like I, I poked around the riff systems website a little bit and like the other the other article that they're writing about or they're writing they wrote they also mentioned and have a link to Martin's series from databases stuff.

46:27 So yeah, it's, it's all connected and I think what people want is like an interoperable way. Let's say let's have some something un some underlying data that is collaborative and then have different, have the power be empowered to choose how you're interacting with that data, right?

46:49 Yeah, for sure. And this is, I mean, this is the, the, the beauty, you know, we forget, again, this is a workshop paper.

46:54 This is why it's a workshop paper. It's meant to throw these ideas is kind of meta to your point mark.

46:59 Like this is the kind of meta, you know, piece of it, right? Like, if, if I wanted to put this system into sitcom, which is one of the big, you know, system papers like this, I would have to now build the system that works for the users and works for some sort of version of this concept, and it would have multiple layers.

47:17 And, and that that's the only way I would get into like sitcom from this work, right? Or, or, or maybe something like ATC or some one of these bigger system conferences.

47:30 So that's really the, the, that's why it's a workshop paper because it's really about kind of, Hey, how do we learn about how can we apply these ideas, right?

47:37 And I give it up to others, right? That you think can reference and make your system in your paper <laugh> your, your, your, you know, whatever large tier 25 page paper.

47:46 That's a slog to read with proofs. And and like evaluation Quinn, It's one last minor point we haven't discussed that I wanted to briefly touch on.

47:59 Yes, the system is about enabling collaboration with other people, but it could also be used to enable collaboration with yourself.

48:09 And that's something that was really functional on me when I read it. And it's something that I'm trying to think into the database that I've been building, because there's also like, just as in GI you have this idea of like working your app, your source code so that you can run experiments and so on.

48:30 There's no reason you shouldn't be able to do that for your application state too. And just as it's valuable to be able to have like my branch and like Emma's branch of a photo sharing app, why shouldn't I be able to make like a workspace that forks the current state of some application, do a bunch of experiments, then either throw them all the way entirely or decide that that's the universe that I now wanna live in.

48:51 And I don't have too much to say about that, except that I think it's like just a slated version of what the paper's trying to say that ends up being really powerful in practice.

49:01 And based on some of my own experiments, there actually are very feasible ways to implement that sort of thing and systems based on ideas like that.

49:09 So that's just something that I'm really excited to see more of. Yeah, and that does have an old line to like small talk stuff, Frank.

49:15 That was something they had essentially. And if you look at Alan k had this paper on Worlds it's essentially this idea with objects though that <inaudible> A totally agreed I, I'm, I said people have this mental metaphor of theory of mind and what is Alice seeing?

49:35 But I think you're right. The other thing that we all have at ID office, oh, what would the world be if the, what if subjective worlds?

49:43 I, I think you've read fuku on mental spaces, he examines a lot. The notion of how is this scene transported to this other subjective universes?

49:54 He riffs a lot on sentences as if you were a better painter. The the lady in the painting would have blue eyes and is it the same lady or different lady?

50:04 Is it <laugh>? And, and and, and what does it mean and, and how do you map between different subjective spaces?

50:10 And these, this is all thing we do unconsciously, subconsciously. And I think mapping these alternate versions of data to alternate subjunctive spaces, what if spaces is again, a strong underlying mental metaphor we can use?

50:28 Yeah, a hundred percent. There's like a, I said there's really a great line of work in various fields to this.

50:34 There's there's this language called black scheme by Ksi, which was this idea of everything being limited continuations, and you drop into these worlds that you can just work in and then pop, you know, essentially pop back in or out to, to different levels of the stack.

50:50 And I think this idea of having a scratch spaces, workspaces ways to work with emerge I mean, you know, in, in the case of this paper, that's a huge, I mean, that's a huge, it's also a form of like, how do you contextualize the, a good

UI with that concept, and then how do you make these things kind of work, right?

51:08 I think you know, obviously there is, there is work in this terrain too. That's really cool. Okay. Anyone else on this paper before we talk about 2023?

51:24 Cool. All right. So 2023 is upon us. I, I did create our fission discourse. I think there is a thread to put up papers and ideas.

51:35 I wanted to kind of get a sense for something, you know, you can think about as we move toward January are themes that you want to cover for next year.

51:43 You know, when I think of you know, I looked at, I looked at this past year, obviously I started later in the year fission.

51:52 You know, there's a lot of like kind of classic papers and stuff like this, which is more foundational thinking stuff, which is really great.

51:59 You know, and I kind of wanted to see what from a group perspective, or you, you can tell me offline again what people are kind of into what kind of themes, any specific papers or themes like, you know, when I think off the top of my head things that would, would be nice to kind of look at would be like you know, some series of historical papers where like some sort of theme where you have always a historical paper and like a, a kind of foundational paper and then like a modern paper that that kind of changes the idea around or kills the other paper tries to, or whatever the case might be.

52:31 Something like this. So, you know, you can imagine having a set for programming models for distributed systems. You can have a set where we can focus specifically on stuff like data log or data flow and these kind of things that have a tie-in to a lot of the stuff that Quinn's working on.

52:48 We can get we can talk more about storage. You know, we kind of distributed, it's obviously storage is, is part now so much with distributed systems cause everything is so distributed.

52:59 So, so you know, I have some ideas but if people have thoughts on what they want to cover and, and you know, kind of maybe like we can look at like four themes for the year and kind of pick again, like historical you know, historical kind of like current set that, or current, like kind of foundational what people view as the, as where, where things are and then where things are going kind of set for each theme was something I was thinking about.

53:26 But if people have thoughts, I'll, I'll open it up for a few minutes if, if anyone talk about that. I don't have any like specific, like this is a like topic that I'm interested in, but I am always, all of the stuff that, that I am most interested in is a lot about like, like, like this paper for example, is like largely about tools for thinking about how to model things, like models to use to think about the, the wacky world of distributed systems.

54:03 Cool. And like cool. Yeah. So I'm, I'm super interested in, in like, I don't know, fundamental ideas more than like, here's a neat implementation we built.

54:12 Cool, cool. Yeah, yeah, for sure. And I think that we, we can do that in the kind of thematic, like it's good to have implementation, but it's also good to have the, the kind of idea piece, right?

54:22 One thing as you mentioned that too that I was like, I was thinking about, you know, one thing that this paper kind of reminded me of that's focused on and, and kind of, you know, again this you know, fourth metaphor you know, good or not.

54:36 It's stuff like, there's actually a lot of great work from the early two thousands on like how do you do distributed debuggers and what is the interface for distributed debugging, which is really cool.

54:45 Microsoft a lot of the stuff around what was the database called? Now it's escaping me. I'll come back to that.

54:55 But they had a lot of stuff with when we were baked in the visual studio where you can get all the school stuff around distributed debugging and how that's a really, that's a really tough problem that's actually gained some interesting traction almost akin to this, right?

55:06 How do you deep, how do you stop something in the middle of all these processes and actually like intro spec right.

55:13 So that's that's something as well to think about. But that's a good point. Thank you for sure. Anyone else? I tend to agree with sodium.

55:22 I think in part because of the really diverse audience we have here. I really like the super math papers too, but historically I found that they don't always lead to as much like diverse and interesting discussion during the reading group because we have people who specialize in everything from like UX to dependent type theory.

55:43 And it's not always easy to find common ground when it's getting related to the weeds on things like that. That's not to say that I don't think that those have been valuable sessions, but I found that papers like the one we talked about today tend to get some, like really new voices into the room that I found really valuable.

56:01 That's a great point. And this is why I actually, I say workshop or you know, actually what are these conferences like, there are these, you know, idea conferences or things even like dog still people have seen those, but talk like, kind of like talk a lot about a certain field of work and where things are going, which are really good, but also like, yeah, there's a, I was saying for programming languages, they don't run this every year and I think they lost a year because of Covid was like Snap Snapple which you'll see like Sharon Ki and others produce like, it's like crazy ideas.

56:32 You know, one of my favorite papers ever is by ama about distributed link types, how to like, have compilers linked to other compilers in other languages.

56:42 Unbillable at the time. But really cool theory, really cool future future thinking ideas. So that's good to know. So I think, yeah, I agree with you that it does, it does create a lot better discussion.

56:53 And typically when I say when we did papers, we love the reason we do, when we did cover math papers and we have the conference clock, we have somebody explain them <laugh>, it's more like a presentation of like, how do you, let's go through a proof, you know, <laugh> like in more detail.

57:06 So for sure I totally get that from a reading your perspective. Anyone else? Cool. Well again you know, you can find me on masks on Twitter, on the fission stuff, on the chat.

57:22 Where all there Quinn as well. You can reach out to us if you have thoughts, papers, things you want to throw out.

57:27 I'll, I'll probably create some sort of thematic thing. We can try and start probably with January being just like one paper and then, you know, open it up for some time being before we start filling stuff in.

57:39 And then we can kinda, we can kind of go about there. But yeah, feel free to, you know, find, you know, me Quinn Phillip offline or essentially and and or online but not here.

57:51 <laugh> about things you're interested in or topics or, you know, you know, even something that, that people that worked people have asked me is like, there's this paper I read that I really enjoyed.

58:01 What's something like it? And that's, you know, maybe that would be a good foundation for us to kind of go, go off that as well.

58:09 I'd be curious, not necessarily live on the call right now, but to hear what papers we've discussed have led to the most valuable discussions for other people.

58:16 Like for me, one of my sessions, the one on yeah, has been the discussion on meta staple failures. And I think that was in part because there was something in there for everybody to relate to, but then we also could get into at least a little bit like some of the like system theory.

58:34 Yeah. Was any one paper from this year that really, that maybe it was, you know, maybe S'S point was this one, <laugh> for this one, but any, any, any that that really popped out for people this year?

58:52 Well, I got, I know we have people probably who were probably on the previous calls who were not here. I if we could do like, maybe just do a survey or something.

58:58 I dunno, maybe I'll, maybe we'll throw something out there. I'll think, I'll try to do that. <laugh>, maybe a survey.

59:05 It's not a horrible idea. We could do that through or something. That would be pretty cool. Cool. All right I'm gonna stop the recording's recording.

59:20 I think it was. Yeah. Here we go. Stop record.