

**00:10**

This meeting is being recorded.

**01:36**

Hello. Hey, bro.

**02:15**

Good morning. Happy Monday.

**02:23**

Good afternoon. Happy Monday.

**02:40**

It's taking me a moment to digest that you're all virtual, not in person in that room anymore.

**02:52**

All right, well, we can get started. So I've got like three major sections for the meeting today. One will do a little show and tell. I know that Bryce has some stuff to show and then we're going to obviously talk about the hackathon and then go through all of the different wins and updates from Sprint Three. Bryce, do you want to kick it off?

**03:27**

Sure, I can do that. I will need to just talk and delay for a little bit because I have a build that's finishing right now, but I can do that.

**03:37**

How about we do a recap of the hackathon and give you a few minutes to get set up? Sure, sounds good. Cover some of the big items. Okay. So, yeah, it was a really I felt like fruitful, productive, meaningful get together to those of you that were able to make it and were participating IRL. Great hanging out. I got a lot out of it. I'm obviously more of a facilitator and less of a contributor, but just being able to sit in and be a part of the conversations really helped me sort of continue to build my mental map of everything that's going on. And there were a lot of, I guess, dots that were connected during the couple of days there. So I put together a little bit of a retro document here for people to go through and if you link to that, it will kind of talk through what the major unlocks or wins were for each day, what the targets of the day were, and then also goes to highlight some photographs.

**05:06**

We tried to take photos of the whiteboards just to capture notes and thoughts that were added there. And then I also wanted to make sure that we added all of the early, I guess, the deliverables that we identified on the first day. Here we can go through and take stock of where we ended up. And then also a link to some of the presentations that were given during the hackathon. So going back to sorry about that. Okay. Going back to the TLDR, I guess the last session of the last day, were all sort of, I guess, confronting some of the headaches and challenges with shipping SBTC mini. And I feel like we had a collective epiphany and it was, hey, why are we putting so much effort into shipping mini? We're not going to have a ton of rollover code that goes from mini to production.

**06:33**

We're going to try to build an airplane as it's flying with this roadmap of trying to activate SPTC production and Nakamoto at the same. Then as we're going through the testing plan for Nakamoto, I think it was dawning on us all that there's a lot of opportunities for people to contribute now, and there's a lot of opportunities to build portions of SBTC production as we're building Nakamoto. And so we landed at the end of this very fruitful hackathon with this idea of what we're calling the Nakamoto flip. And so that is essentially just the broad strokes. There's like three bullet points there pausing or halting on SBTC Mini in favor of shipping a more robust SBTC Alpha, or as I jokingly call it, SBTC Alpha Romeo. And the goal is to ship a concise and stable SBTC testnet no later than October 1, and then that Alpha will transition from Alpha to main net with SBTC and then start by prioritizing Nakamoto right away through this more sort of test driven development approach.

**08:06**

With all of the great testing plan that Jude worked on, were able to go through and identify how and where folks can start to contribute to that in a pretty immediate manner. So then the goal would be to ship testnet for Nakamoto near November before the end of the year. And then there would be, obviously, the audit process during the holiday season and the main net launch and hard fork that would go around that maybe around February. All. In the meantime, we're able to have Alpha and continue to build off of that and improve upon it until that becomes the final SBDC production launch, which may require second hard fork more in April. So there are a lot of people that I guess, were contributing to this. I'd love to just kind of, like, pause on this particular item. I probably butchered some of it, but we could go around the horn and hear from some of the people that were in the you know you did a great job with summarizing this all in the hours after that meeting and following up with a call with maneve and making sure that we kind of get all aligned there.

**09:41**

Could I hand it off to you and ask you to say a couple of words? Yeah, sure.

**09:46**

I think you covered the main bits fundamentally. We learned a lot during the hackathon, and a lot of the various leads from the product side and the tech side were talking toward the end. And it was just clear that Mini became sort of this thing to try to please, really just like a couple of app builders that we're very excited to make happy. But were really putting the overall system at risk because of it and making it a lot more complicated. We were creating a lot of code that couldn't be recycled. One of the key insights was that for Mini to be really valuable to the app builders that we had in mind, it really would need three to \$5 million in the system. And one, the engineers and myself were not really comfortable putting that kind of money on the line with the security promises without Nakamoto.

**10:49**

And then on top of that, the only other way to get there maybe safely was to do a closed signer set, which opened us up to pretty complex, maybe legal concerns, but also just trying to find the right partner who has a money transmitter license, who then could also work with us quickly if we had an issue. And when real money is on the line on main net and it starts to get really complicated really fast, and then when you also realize you can't reuse a fair bit of that. I think all of us in the room just felt like one. Let's get Nakamoto live, let's deliver people value. They'll be able to use SBTC on testnet. In the meantime, we'll be able to launch and make sure we have a secure system and then layer on the additional complexity of SBTC. And so from the resident side, from my side, definitely it's a bit of a change, but we are getting value in people's hands.

**11:47**

And so given that we were going to wait to launch them at the same time into Q One at some point next year, already, this actually brings them some pretty significant value sooner. And yeah, I think that's it. I think fundamentally it was you can't make everyone happy, but this makes the most people happy and also gives us the best chance to build a really secure, safe system, the sort of most responsible way. And what I'm hearing from engineering is that as soon as we get our plan kind of built out around this direction, it should be more predictable as well, because it's not less complexity, but dealing with the complexity in the right order. Yeah, there's a whole doc we can share around that describes sort of all the advantages of the plan. And I know Sarala has been working on a more detailed plan, like downstream of the shift.

**12:48**

Anything that you'd like to add?

**12:51**

Yes, I think both of you cover really well on the context before I share more details. Word of caution, these are super aggressive and very optimistic timelines. So these are not yet shared publicly and externally. Mitchell and team will do that. So let's keep those timelines that are more towards the main net and forking just within this group. What we can strive and work towards are the testnet timelines, and I want to build confidence within this group to figure out if those timelines are viable. And then Mitchell will and the other comms teams will figure out what the main timelines based on other activities would look like. Yeah, so looks like our singular focus is and narrow focus is all things Nakamoto to go with right now. That means shift in priorities for few people, as you can see, especially for the Steiner team, for Jess and Sayek, this shouldn't come as a surprise to you.

**14:00**

And we are derisking Mini, SPTC, Nakamoto and main SBTC with this plan, which I'm feeling very particularly confident about, because then we are really bolstering our confidence on the Nakamoto timelines and then building on top of those faster blocks, better blocks for the main SBTC. Does anyone have any questions? Most of you have already been filled in, I suppose, but if you all have questions, I think now is the time.

**14:32**

So I have one question, if I may. So one bullet from above said that the design of Alpha should anticipate a seamless transition from Alpha testnet to mainnet test BDC. What does that exactly mean? Is it just a matter of having the same UX for the users? Or is it something else.

**14:58**

I may have? Misspoken Martin, is that something that you could speak to in terms of how Sptttc Alpha could mature into SPTC production with minimal interruption?

**15:18**

Was that directed at me? Sorry, my tab just crashed, so I missed.

**15:22**

Yeah, okay.

**15:26**

I think I heard a I mean, the intention of SPDC Alpha is to focus very much on UX user flow. So as we're developing, we have a pretty good idea of the UX of SPDC that we're targeting and want to implement full parity with that in SPDC Alpha. So, as opposed to SPDC Mini, where we did a lot of UX compromises to accommodate for the fact that we're building a decentralized system and making sure designers worked and had decent economic incentives, alpha is going to be a custodial system. And the only purpose is that an application that works with Alpha, an application that can actually deposit SPDC in Alpha, should just have to point to Mainnet instead of testnet and point to another smart contract address, and then it should work with Mainnet SPDC. Does that sort of answer the question?

**16:20**

Yeah, that makes sense. Thank you.

**16:29**

Any other questions? Obviously, big change. Change can create anxiety. Someone like Aaron Jacinta, it's almost like we had this one situation, like, 95% of the time that you're at the hackathon, and then like five minutes after you left for the airport or something, things changed dramatically. In particular, I'm sure that's probably discombobulating, if I remember the timeline correctly. So, Jacinta, how are you handling this? I know that there's obviously a lot of work and effort and set Zeus into the SBTC mini strategy, and I don't want anyone to feel rug pulled. Essentially.

## **17:34**

To be honest, every single week I had rug pulling happening. So it's not particularly different. And honestly, I think it's the right way forward. It's just too bad we didn't come to the conclusion six months ago, but, I mean, hindsight is 2020. I think actually it is a little bit too bad that some of the work that I did isn't going to get used. But most of the work that I had done was like, understanding the final system and how mini fit into it, and that was always very confusing. So I still learned a lot about the final product, which is the most important part anyway, so I don't feel like it was a total waste, so don't worry about me. I'm very happy with this direction. There's been Whiplash my entire time here, so it know, adds to it a.

## **18:26**

Great and Jude, love to hear from you. I know that a lot of what was sort of prompting us to move this direction came back to some of the conversations that were having around the testing strategy and I can kind of go back to that was really like in day two. We were starting to go through this particular document and trying to identify, hey, what's the dependency graph look like, who can start contributing now? How long is this going to take? Trying to do this in parallel to SVTC Mini was just all feeling quite overwhelming. Anything that you wanted to double click on here?

## **19:18**

Speaking of whiplash, let's add onto the pile here. That testing plan is probably going to be scrapped.

## **19:23**

Okay.



**19:27**

Aaron has taken the initiative in coming up with a, in my opinion, much better testing plan that kind of does this but in reverse and we're still working on it in a Google Doc right now.

**19:38**

Oh, great. Well, we look forward to that. Is that something that you think we can set aside sometime during the Sprint planning call tomorrow to honestly Will, in.

**19:56**

The interest of Minimizing Whiplash, let's sleep on this for a little bit until you feel more okay.

**20:04**

Okay.

**20:04**

Yeah. Jude and I are working on getting it into a form that I think we're both happy with relatively soon. We do understand that it's blocking a lot of folks, but I think it's better to block sort of right now and then unblock everything so that.

**20:27**

Maybe.

**20:27**

Worth giving everyone on this call the context on why we're changing that errand a little bit.

**20:33**

Yeah.

**20:34**

So the high level idea is that the goal of it is to get to something like a testable super alpha early, earlier than November, just so that there is kind of like an integration point where everybody can start building out things. So it's more of like a high level first. So like, starting with an updated node and then try to fill in all the components as we go, adding new features. The idea would be for the first milestone of the Nakamoto Workstream to be something maybe lovingly titled that's kind of the context around it.

**21:29**

Excellent. Great. Well, looking forward to that, Bryce. So I think in the wake of all of this happening, things are still moving forward essentially as planned with clarity wasam. I guess I do want to go back and make sure that we 100% celebrate some of the wins irregardless from the hackathon. So I'm just going to go through some of these real quick. So Tyler on board. He's not on this call, but really great getting to meet him. Amazing addition to the team. Just rolled up his sleeves was in there from day one. Managed to digest a whole lot of information and lead on a number of fronts. Really. The decision that we arrived at with Nakamoto Flip comes back to, hey, we're all in a room, we're talking through the legal implications of the commit reveal of money transmission, like the gymnastics that we're going to go through.

**22:51**

We're feeling like we need sort of legal opinions in order to write code in this vein and that probably doesn't sound like it's that great of a position to be in. And then, as Mitchell was mentioning, like trying to get liquidity into the system and the lead time involved in that and trying to identify Jude as you is who has the reputation to handle this in a trusted manner that we can trust. So a lot of conversations going into that decision took getting people in a room and talking through this in a number of different directions. Andre, there's a link to his presentation on the go to market. Jude shipped Stacker DB, which was a really seminal moment of day one. At the end of day one, day two, elena came in and hosted a really great workshop on SBTC signer user stories. Presentation is also linked here.

**24:13**

There are a bunch of conversations, architectural and otherwise, about the Commit reveal signer coordinator, different workshops there. Again, went through the Nakamoto testing plan, started workshopping that and I feel like wheels are starting to turn. And also talked about integration testing. Bryce had a demo, which we're going to go over again now on day three. Continued a lot of those workshops. Jude, you shipped this SBTC mini skeleton and then went through the Nakamoto flip. So that all kind of came together in the afternoon. There's a document here that is what Mitchell put together and presented to. Again, lots of great wins. Definitely don't want anyone to feel like anything was for not Bryce.

## **25:19**

Well, real quick, just on the Nakamoto kind of shift, I want to be sort of clear about the communication plan. So right now we're in the phase where we're getting all the engineers on the same page in various teams and affording folks at Trust Machines especially, time to chat with their apps and let them know there are other paths for those apps. So it's not like they're going to have to stop development altogether in the meantime. And then for others, mini wasn't really an unlock for them anyway, so most should be fine. And thankfully, we've been strategic about not sharing a lot of specific timelines externally. And so actually against what we have shared publicly, we're still on track. And so I don't want to go around to the community and saying things like delay or big change or big shift. It is internally in the way that we're deciding to roll this out in the order that they'll receive it, but it actually is fairly well aligned to things that we've said before because we were intentionally vague, because we wanted to give ourselves the space.

## **26:32**

And so I would say for the next few days, if you get asked about it from a community member or just like the general public, feel free to redirect them to myself or whatever. We'll have some public comms up. Andre and I are already working on that. The small roadmap that we do have up, we'll bring that up to speed. Again, it's still accurate, but we can just add more details to it now that we have it. But just be careful about how you present it because actually the important thing here is that people are getting value faster. And this is quite a big unlock for probably even a wider audience than SPTC because Nakamoto brings security and speed. So if anything, the message should be reduced complexity to ship better to ship faster versus like, oh, there's this full stop, big change. We're actually getting rid of something we didn't need to build and reducing complexity.

**27:29**

And that's sort of what I want to be the main message. If you're not comfortable delivering that's totally fine. Like I said, we have some resources coming. But again, the takeaway point is we've been very careful about what we said just for situations like this. So we haven't really set the expectations externally differently. And so there's no reason for us to be like, hey guys, so actually we have this big change now. We weren't even detailed enough for them to really realize. So keep that in mind and if you get questions about it, I'm happy to be the point for now until everyone is up to speed.

**28:05**

Excellent. Zeus, I did want to hear from you just on the clarity front. I know that SBCC Mini obviously very clarity heavy. Could you just speak a bit about how maybe this changes or helps or sets back any efforts that are going into that?

**28:33**

As most everyone knows now, mainly had a come to Jesus moment in a negative. Yeah. On that note, again, feelings aside, it seems like it's the right move for the ecosystem based on requirements from all the entities. So I'm happy and on board to move with what's going on mine. I'm still trying to understand what this means for our group in terms of immediate actionables, it seems like. I'm not sure what we're going to have to do in terms of the new alpha. So first I'm trying to wrap my head around whether there's any changes on the clarity contracts that need to be done with alpha because I want to make sure we stay available for that. And then after that, it's going to be a lot of planning and designing for Nakamoto, which on my end, I'm not sure what are going to be the final contracts for that.

**29:28**

I have been talking to Jude about which contracts are going to persist which aren't and which ones are going to need. But yeah, definitely a big reset on my end. So I'd say the next week or two is going to be a lot of designing, planning, and researching.

**29:42**

Jacinta yeah, just a quick comment on the clarity only just from the very brief meeting that I had with Morten and Stefan had done a large meeting. I kind of joined at the end, but I think we could probably reuse some of the logic for minting and burning dependent on a BTC transaction. So I know that the alpha clarity contracts, for example, you just called the mint and burn and it just did it. If you made the call, it would just do it. There was no validation that a BTC transaction had been confirmed, so that definitely could get pulled in. But I'm sure there's other stuff that.

**30:22**

Could be used as well.

**30:24**

Let me write that down exactly.

**30:26**

We don't need full validation, but we need to be able to make sure that a mint doesn't materialize on a fork where the deposit didn't happen, for example.

**30:35**

And did I hear right that let's get this sure, because I asked you this as well. Do we want the balance of SBTC to also be migrated seamlessly from alpha to Nakamoto mainnet, or do we not care about that?

**30:52**

Let me reply. I see Ashton has his hand raised. But just a quick reply on mean, we're only aiming at deploying this on testnet.

**31:00**

So.

**31:03**

That being said, if we deliver a system that anyone could deploy, someone might deploy it on mainnet. And if there's a good idea, I think we have some ideas on how to do a migration pop that could be a nice add on. I see it as a nice to have, and I think we should have a broader dialogue on the priority of that. But yeah. Ashton yeah.

**31:24**

I'm actually going to bring us kind of back to a much earlier point. So if anyone has anything that they want to say on this topic.

**31:32**

I'm.

### **31:33**

Happy to lower my hand and then come back up when we're done discussing it. So actually, I want to go quickly back to something that Jude and Aaron said about rewriting the testing plan. I'll say, at least from my perspective in the meetings that I was in, that concerns me a little bit, especially since I'm looking at this list here. Those numbers come from the original testing plan that we had. So if we're rewriting it, basically what we're also rewriting is the deliverable timeline. I think in the meeting that we had on Thursday, we all kind of sat down and agreed on that deliverable timeline. So I'm wondering if we're taking this opaque approach to designing this testing plan, does this translate to a list of deliverables? And then how are we validating that this testing plan matches what we think as a group, we could actually deliver on time?

### **32:23**

Because that's just sort of the warning bell that's going off in my head. And anyone else, feel free to jump.

### **32:29**

In to answer your first question. Yes, this will turn into illicit deliverables pretty quickly. To answer the second concern, this actually, I think, removes uncertainty from the current system. The key difference between the testing plan we talked about last week and the one that we're coming up with now is that the former takes a bottom up approach where we build smaller subsystems, and then later on, we defer the time required to combine them together into a node to a later date. We tried that before with Sax 2.0, and it led to product delays because the active deferring building the customer deliverable also meant deferring all of the unbound learning that had to happen to figure out what pieces were actually necessary and how they really ought to fit together. The new approach is more top down. We're starting with a node, and then we're mocking the subsystems internally and building them out over the next six months.



**33:27**

The advantage of this approach is that we know from the get go and can figure out much more quickly what the system has to behave like. It also cares that it front loads all the learning work that has to happen to figure out what has to be integrated, where and what the real modules of the system are.

**33:44**

So then when we are talking about deadlines, ultimately this document that you and Aaron are putting together is a list of deliverables, but it isn't going to be referencing timelines. And then maybe as a group, we can discuss how long we think each thing is going to take.

**33:59**

I strongly suspect there'll be overlap between the two. Yes. To answer your question, just because at a high level, we know roughly what goes into a blockchain, what this will hopefully let us do is by starting with the testable artifact sooner, we can identify pain points that are unknown. Unknowns right now, earlier.

**34:17**

Yeah, this makes sense to me. It seems more like a test driven approach. Thank you for clarifying that, Jude.

**34:25**

So with the work that's happening right now, jude and Aaron, obviously it could throw a wrench into Sprint planning tomorrow. Are there topics that we feel we should use that time to have a more protracted group conversation about, or do people feel like there's enough items that they can button up with regard to this SBTC Alpha work and or testing and integration testing frameworks that can be worked on absent of clarity on the Nakamoto front?

**35:18**

So if I may chime in while Erin's gathering his thoughts.

**35:22**

You're muted, too.

**35:25**

I'm muted. Can you all hear me?

**35:27**

No, we can hear you.

**35:29**

Okay, so I think the Clarity Working Group, the testing and hardening the Clarity Working Group can continue. That changes nothing for the plans to not change. So that train is already off the station. We can use tomorrow's call to figure out the plan for that the only caveat to what Aaron and Jude are working on. June mentioned flipping how we approach deployment. Some of the moving parts around the interim milestones, probably more or less stay same or similar. The moving parts are around. How do we test and deploy that? Erin has an alternative proposal for so that I think I don't know if you'll have time ready by tomorrow?

**36:22**

Probably not.

## **36:26**

Yeah. So I think that the work streams likely impacted are basically like, if you look at this, it likely impacts the block producer and stack signer the most. But ultimately, I think our goal is hopefully to get to at least near agreement before Sprint planning tomorrow. We'll see. But yeah, the people probably most impacted are, like me and Jeff and Jude.

## **37:12**

Yeah.

## **37:13**

And the Testing and Hardening group also has some other quality of life changes that they can continue with and can set milestones there. So I think it's definitely worth having the planning session tomorrow, even if it's for a shorter duration, to kind of agree on that. I know. Andre, your hand must be hurting quite a bit. You've had it raised for a while. Did you forget to lower? Do you have something to say?

## **37:37**

No worries. This was a good conversation. The one thing that I wanted to add quickly was just for me, this is kind of an opportunity to think about how we do release planning in general, and potentially how we can improve those processes moving forward. And so when we think about what is the process for how we go from discovery of a new proposal to validation of the idea and the amount of scope that goes into it, ultimately to acceptance, I think is something that might be helpful for us to kind of maybe think about how we can create a good process around that moving forward. Because I think one of the takeaways that I had throughout the Nakamoto flip is really mini was, I think, the preferred approach because it allowed us to get as close as possible to SPDC in production without the Nakamoto rules.

### **38:31**

And I think that one thing that we've learned, really, is that really, we actually just wanted to have more of a minimum viable SPDC product on testnet sooner and be able to essentially get the nakamoto rules in production so that we can actually have a better sort of testing ground for SPDC. Since the system actually requires nakamoto to be successful. And so, yeah, just thinking about how we can improve that release planning process, I think is one thing I'd like to kind of take away from this conversation and think about how we can.

### **39:08**

Move forward on that.

### **39:40**

Yeah. Andre, I think that's maybe something that we should try to come back to and give some time after people have a little bit of a chance to think about it, perhaps in the call tomorrow. On Thursday. Yeah, sounds good. So Bryce, what are you going to show us here?

### **40:09**

Yeah, let's do it. So first I'm just going to show the benchmark the same as the one I showed last week, although this time it runs both the current interpreter and the VM we're working on with WASM. So I'm going to go ahead and start that. It takes about a minute to run, so I'm going to switch to my other terminal here. But this is basically doing, if you didn't see the previous one, I have this long list. There's like 81, 92 elements in this list and we're just doing a fold over it. We're multiplying and then adding and summing it up. So this is called the fold add square benchmark and it's going to run with both the current interpreter and the new WebAssembly VM. But then while that's running so I'm going to show the other thing that we got working. Now this is not like actually merged or deployed or anything like that, but inside of Clarinet.

**41:00**

So I just ran Clarinet console here and this is basically going to run. It can execute contract calls in both of the VMs. So if I run fold ad square here, so this is the normal result and then I'm also seeing the WASM result. And so this is kind of useful for interactive testing. I can make sure things are working really easily in this console. Eventually the plan would be to keep both in there and run both of them. So we did this for a while with the new parser where we run the old parser and the new parser, check the results and if they don't match, then we put a big error message up and says, please report this issue. But basically it allows us to get a lot more testing of random things that users are going to throw at it and we can quickly compare the two.

**41:55**

So it's both a tools for us as we're building to interactively test things and then also to get some more testing from random inputs that users are going to throw at it. And that would get deployed into Clarinet before this WASM stuff is deployed anywhere else. So it's a good early testing option. Okay, here we go. So this one finished running, so we see the WASM version finished in 218 microseconds and the interpreter version finished in about 20 milliseconds. So we're looking at like a 90 x speed up or something like that. And that's before we've done really much optimization. So it's still looking very promising. That's all I've got. And this is thanks again to I don't think any of them are on this call. Yeah, but this is like not just me, this is Anthony and Kyle from Bitcoin L, two labs, and Hugo from.

**42:55**

Hero working on this.

**42:59**

Fantastic.

**43:00**

That's awesome. What's that terrifying tab you have there called Oom clar?

**43:10**

We don't need to talk about that.

**43:15**

Bryce is planning to analyze service checks against the blockchain.

**43:19**

Yeah, I'm going to stop my sharing now. No, that was investigating a bug. But don't worry, it's fine.

**43:30**

Aaron's just hawkeye in disguise if you didn't know.

**43:45**

That's. Great. So I do want to go back through. Just make sure that we elevate and celebrate all of the accomplishments from Sprint Three. So I tried to pull some highlights here. I'm going to go down the list and then we can wrap up again. I'm kind of, like, lump people together. I know that this isn't always so perfectly delineated, but otherwise the list just gets to be, like, four times as long. So we'll call it nakamoto. Jude, Aaron, Jeff, shipsacker DB, worked on the producer architecture producer RPC endpoints, and then the block producer Binary. So, Aaron, Jude, Jeff, anything you want to say about those items, those I guess, efforts that you took on during Sprint Three?

**44:52**

Yeah, so super excited about the progress on Stacker TB. I think that with the RPC endpoints, we implemented some pretty quick and dirty RPC endpoints to do some of the things that a block producer might want done. But I think that big output of this work is just sort of in trying to inform how quickly we can get to first version of a working.

**45:30**

Jesse quality of life. I know there's several others that are working on this, but I know that you've been working towards pushing this very large CI related PR.

**45:47**

Yes, the PR is open. Aaron took a look at it. Ashton's been very helpful with comments, which I will be addressing soon. I tagged several other people in there, so I'd appreciate if everyone could take a look. Let me know what you think. And yeah, if we can get this merged, it will reduce testing time dramatically. I think it would really improve the quality of life around our.

**46:18**

Bryce, I think your list here kind of speaks for itself with the demo. Anything else on here that you want to highlight.

**46:29**

Now?

**46:30**

I'll just add that. So we're continuing to make good progress. I think we've got good planning going. We have an engineer joining Hero tomorrow that's going to be joining this team as well, and his background is a perfect fit. So I think he should be able to hit the ground running and we should be looking really good.

**46:51**

Awesome. Mark, I don't see Scott on here, but anything you want touch on here from the list on Stack security?

**47:02**

Nothing.

**47:02**

Not directly in Pexus program right now.

**47:05**

Okay. And testing. Ashton. Yeah.

**47:14**

Some of these I think one of those is very what's it called? Very trust machine specific when it comes to the SBTC Bridge. But I feel like a big success was being able to test a number of task runners and document the progress on them. I think that they have the potential to really ease up local development for a lot of our developers and make rapid testing of our local changes much easier. But yeah, I don't know. For testing and hardening. The big highlight for me has been looking at Jesse Wiley's CI improvements and being able to get that under 20 minutes, or I think what was it? Around 20 minutes?



**47:53**

Yes, just running the test, give or take about 25 minutes. For unit tests, specifically, when you combine building the binary, it adds another ten to 15 minutes.

**48:04**

So it's all beautiful from what I'm.

**48:07**

Caching so we can use the same binaries across every test.

**48:11**

Yeah, that's a big highlight.

**48:20**

Great. And then I tried to note, obviously these insights, like when I pulled this together come from the Daily so, you know, I'm not able to speak to everything because if you don't post it, I don't know about it in a lot of cases. So, Sergey, Carlos, Nicos, Jordy, could you all please start posting in the daily update channel? SPGC Martin, I don't see syac on here, but anything you want to speak to?

**48:52**

Yeah.

**48:55**

No, I think it sort of summarizes. This is a lot of low level details. There's been a lot of dialogues, as you know, that sort of culminated in the big Nakamoto flip. I think most of these things are explorations that led to that, so we don't need to go into these. Yeah, and Joey is also probably not in this call, but I think the interesting aspect from him was he had a full DKG running on top of Stackerdb at the end of the week, which was a big win. I think the only demobile artifact coming from our side. But seaplane can speak for the SDKs of things.

**49:43**

Yeah. So as mentioned there, we now have the prefixed contract names in the SBDC core. But what that basically means is that we correctly serialize and deserialize principal data. So it should be fully compatible with how clarity does it. I've also made a PR for commit reveal logic for the SPDC core. We now have the parsing of the operations, construction and some utility functions to make it easier to build those transactions out. And yeah, the next steps now should be to move the SPTC CLI so we are able to basically play with it and create deposit transactions, withdrawal requests and fulfillments. And something we discussed today is how to basically take what we have in the SPDC Alpha and push it to the next stage. And we also have a PR for that, which the issue actually, which I can link in the chat just now, I need to find it.

**50:57**

But yeah, that's about it.

**51:02**

Great. Mike, anything you could say to speak to what I call SBTC UX UI bridge signer dashboard?

**51:15**

Yeah, I've just only just today heard about what happened last week, so Justin and get my head around that and late Jesus are going to be figuring out what that means, but yeah, spent the last few days trying to get a reg test, local host kind of running against the mini contracts. So I think a lot of that will be reusable switching back to Alpha.

**51:42**

Great. Yeah. Sorry for not getting that communicated to you sooner.

**51:49**

That's okay.

**51:51**

The clarity side. Jesus. Fur, Jose.

**51:58**

Yeah, happy to take it on here. So a lot of the work that we are now doing has to do with the testing, specifically which way we're going to test. Friedger has definitely gotten the furthest in terms of what the integration testing environment looks like at the moment within our group. Outside of that, hackathon wise, I would say the biggest win was it was pairing up with Jacinta and the knowledge transfer that happened there. But substantial changes are going to happen and, yeah, I would say the big win is the fact that we got many close to 100 unit tests that were probably like 90, 80 ish something that were passing before all this happened. So I would say that was probably the biggest one that we had. And yeah, like I said, next weeks are just going to be this next Sprint is going to be all planning and research, I would say.

**53:01**

Great. Jacinta, anything you wanted to take a moment to talk about?

**53:08**

So yeah, I have no idea what the block producer signer actually entails. So this is going to like Sprint is going to be me reading up on the most recent Necromoto documentation because I've only really read the Sip 21 in relation to that. So it's mostly going to be the same for me. Planning, research. I'm sure I'm going to be bugging mostly Jude to get kind of an idea of how to best start so that I'm not stepping on toes. That's kind of the plan for this Sprint anyway.

**53:41**

Great. Andre, I pulled together a list of kind of management related things. Do you want to speak on behalf of this collection of people? Anything from the Sprint Three that jumps out as a real highlight?

**54:04**

Yeah, no, I think this is a good list. Top of mind for me is just kind of, I guess like everyone thinking through sort of how this changes sort of product related work streams. So there's been a lot of work done on documentation as well as for the bridge and the signer. So just kind of thinking through the changes that needs to happen for those work streams. I'm going to be doing a lot of planning for that over this next Sprint.

**54:39**

Okay, great. So for tomorrow, I'm already hearing obviously a lot of need for RMD to occur during Sprint Four. We also have the revised knock motor testing plan coming together. Anything that folks would like to talk with me on the side about planning for tomorrow? I think it'd be great to utilize that time to talk through anything that folks feel could benefit from the larger group. But feel free to reach out on the side. Otherwise I might be reaching out to you. All right. Thanks all you. Talk to you soon. Bye.

**55:30**

Thank you.

**55:31**

Bye.

**55:33**

Have a good week, everyone.

**55:40**

The recording.