EPISODE 599

[INTRODUCTION]

[0:00:00.3] JM: Cryptocurrency infrastructure is a new form of software. Thousands of developers are submitting transactions to Bitcoin and Ethereum, and this transaction volume tests the scalability of current blockchain implementations. The bottlenecks and scalability lead to slow transaction times and high fees. Over the last 20 years, engineers have learned how to scale databases. We learn how to scale internet applications like e-commerce stores and online games. It's easy to forget, but there was a time when e-commerce systems and online games didn't perform so well either.

Scaling a blockchain is different than scaling a relational database or a microservices infrastructure. Blockchains are peer-to-peer databases with an append only ledger shared by thousands of nodes. With different scalability solutions, there are trade-offs between decentralization, scalability and security. As an example, in Bitcoin, the core developers are working towards deployment and adoption of lightning network. Some would argue that this approach, lightning network, favors scalability over decentralization because there is potential for centralization into some of the sidechains.

Today's show is about scaling Ethereum. Raul Jordan and Preston Van Loon are developers who are part of Prysmatic labs, which is a team building a sharding implementation for the Go Ethereum client. In this episode, we discuss Ethereum's approaches to scaling, including sharding and plasma. I also want to announce that we're looking for writers for Software Engineering Daily. We want to bring in some new voices. We're focused on high quality content about software that will stand the test of time. You're listening to content about software engineering right now, you probably read content about software engineering. If you want to write, go to softwareengineeringdaily.com/write to find out more.

We're looking for part-time and full-time software journalists. We want to explain technical concepts and tell the untold stories of the software world. We're looking for writers who only want to produce a single piece of content and also people who want to produce a series or want to do investigative long-form journalism. For example, we'd love to do a series on lightning

networks. I haven't read an in-depth discussion of the bottlenecks of lightning networks, for example. I'm sure there's one out there, but I haven't read one. We'd love to do written content about scaling Ethereum as another example.

We just launched a new design for softwareengineeringdaily.com, and if you'd like to work with us, go to softwareengineeringdaily.com/write. You can also send me an email, jeff@softwareengineeringdaily.com.

[SPONSOR MESSAGE]

[0:02:52.6] JM: Software Engineering Daily is brought to you by Consensys. Do you think blockchain technology is only used for cryptocurrency? Think again. Consensys develops tools and infrastructure to enable a decentralized future built on Ethereum; the most advanced blockchain development platform.

Consensys has hundreds of Web3 developers that are building decentralized applications focusing on world-changing ideas, like creating a system for self-sovereign identity, managing supply chains, developing a more efficient electricity provider, and much more. So, listeners, why continue to build the internet of today when you can build the internet of the future on the blockchain?

Consensys is actively hiring talented software developers to help build the decentralized web. Learn more about Consensys projects and open-source jobs at consensys.net/sedaily. That's C-O-N-S-E-N-S-Y-S.net/sedaily, consensys.net/sedaily.

Thanks again, Consensys.

[INTERVIEW]

[0:04:08.9] JM: Raul Jordan and Preston Van Loon are developers with Prysmatic Labs. Guys, welcome to Software Engineering Daily.

[0:04:15.1] RJ: Thank you. It's a pleasure to be here.

[0:04:16.6] PVL: Yeah. Thanks.

[0:04:17.9] JM: So today we're talking about Ethereum scalability, and in order to get towards that discussion, I want to start with Bitcoin scalability. Why doesn't Bitcoin scale?

[0:04:31.2] RJ: Yeah. Traditional blockchains face the limitation that if you try to create a decentralized ledger, where you require every single node in the network to validate and verify transactions, you end up with an incredibly slow network with the gain of security. So the problem is that whenever a transaction is broadcast to the Bitcoin network, we have nodes all around the world that have to download these pending transactions, processes them, and actually do a proof work algorithm until we end up the block that can be propagated back to the network with a solution. So you end up having around anywhere from 5 to 10 transactions per second, usually on the lower end because of this global validation that needs to occur on these blockchains.

[0:05:11.1] JM: And we've done a number of shows about Bitcoin scalability. The main solution that we seem to be on the road map towards is lightning network. Lightning network is the idea that you have these networks that are off chain. They're off the main chain and you have a lot of transaction throughput that goes through these networks and occasionally reconcile with the main chain in order to take a high volume of that transaction volume off of that main chain and lower that bandwidth that the network validation system has to deal with. What do you think of lightning network and why is lightning network taking so long for Bitcoin to implement?

[0:05:53.5] RJ: Sure, there are two different approaches to scaling these platforms. One are layer one scaling solutions where you actually modify the core protocols and the others related to solutions such as a lighting networking and Raiden, which is the equivalent in Ethereum. These deal with the issue of transaction throughput and being able to broadcast things faster, be able to settle things and do some sort of micro payment system.

However, layer one solutions are the ones that are concerned with actually modifying the core blockchain to accommodate for a scalable solution from the core. So the idea here is that with these approaches you gain higher throughput, but this does not necessarily solve the problem

of having high computational requirements for many nodes, the ever-increasing storage of the blockchain that is a pressing concern given that it's only been a few years. I personally run an Ethereum full node and we're almost up at 1.1 TB and just pure state data plus blockchain data.

So, at the moment, these blockchains, we need a layer two scaling solution such as a lightning network that is able to handle things off chain and allow for faster throughput. However, what we believe is really important is to fix the problem from the core, which is fundamentally shift the design of the blockchain to accommodate for a more sustainable scaling strategy that preserves security and also preserve decentralization.

[0:07:08.0] JM: We'll talk about that in the context of Ethereum in a moment, but are you suggesting that that kind of solution is required for Bitcoin as well?

[0:07:17.0] RJ: So there are certain approaches to try to do kind of like a sharded UTXO blockchain, which is what Bitcoin. So sharding is a fundamentally different design to a blockchain protocol, in which you have a system in which only a few set of nodes have to validate a certain portion of transactions and other nodes validate other portions of transactions. So you kind of split the blockchain into a bunch of different kind of shard chains that allows you to parallelize the whole throughput of the system.

So this approach works for – It's a lot easier to do, and like proof of stake based systems, and we'll get into that a bit later. But, yes, it could work in theory and systems such as Bitcoin's UTXO model, which works with proof of work.

[0:07:56.9] JM: Okay. I do have a bunch of questions about a Ethereum prepared, but just to stay on this Bitcoin stuff for a little bit longer, is this debate around Bitcoin scalability that gets acrimonious in some places, is it driven by rational debates or is it more about tribalism?

[0:08:18.0] RJ: That's a very controversial point. I think Preston can also speak towards this, but one of the key talking points of the Bitcoin scaling debate is increasing the block size, right? So you have this idea that, "Okay. We can have the short-term," where we increase the block size. Thereby we can accommodate more transactions, and thereby increase the throughput of the blockchain. However, this doesn't really – The issue here is that doesn't really solve the core

issue, and that's kind of what one side of the argument goes for, is that we're going to have bigger blocks. We're going to have bigger blockchain over the long term. We're not going to be he reducing computational resource requirements on nodes. However, this does see a very tangible benefit in the short term. So that's kind of where the two camps have split, and that was a big part of the debate kind of towards the latter part of last year. So there are a bunch of different arguments on those sides and a lot of it comes down to very polarized opinions.

Preston, do you want to maybe comment a little bit about the Bitcoin scaling debate?

[0:09:09.6] PVL: My only comment is I haven't been paying too much attention to Bitcoin. The community there is not as exciting or friendly as I found the Ethereum community to be. When the whole block size debate happened, is it was rather hostile and it seems the community was very divided and people had their own interests rather than building a better ecosystem for everyone. They were more incentivized. I would profit them the most. I've kind of tuned that out just trying to focus on Ethereum scalability. So I can speak too much to Bitcoin.

[0:09:46.3] JM: Although from an architectural standpoint, do you see this lightning network as encouraging centralization? Also, another question is, this premise that there needs to be at change to the core Bitcoin blockchain infrastructure. I don't quite understand why that is because it seems to me like you could do everything in layer two solutions. You could have a free market of layer two solutions, and as long as they have an agreed-upon way of how they're essentially compressing transactions and writing them to the main chain, I don't understand why the core Bitcoin infrastructure would need to change. It seems like you could just have decreasingly decentralized layers in the layer two solutions. You could do whatever you want in the layer two solutions. Again, as long as there's a standardized way of writing that data to the immutable core Bitcoin blockchain, I don't quite understand why it's problematic or why it even is going to lead to necessarily bigger block sizes.

[0:10:50.7] RJ: So I was referring to kind of the debate that was going on around actually modifying the block size of the core protocol level that was happening towards the latter part of last year. Yes, I agree that the layer two approaches is kind of what the community is tending towards, and what I think we'll see a lot more innovation moving into 2018, especially as lightning network gets more usage and such. So yes, that's right.

I think like lightening network and these layer two solutions are pretty good approaches for Blockchains that are just kind of payment mechanism, source of value, such as Bitcoin that have a UTXO model where you don't really have a Turing complete programming language. You have smart contracts running on it. You have applications running on it. So it's a lot more of a constrained design space, and I think for us in Ethereum, we're concerned with execution and the efficiency and kind of the security decentralization and censorship resistance around execution of transactions. So because they carry data, because they're able to execute code on a global computer, so at that point things such as sharding become a lot more important. Yeah, I do think that the community is going to tend to stick to just layer two solutions, as you mentioned, Jeff, at least in Bitcoin. But for us in particular, like layer one is very, very critical.

[0:12:02.9] PVL: I think you know layer one solutions solve more use cases. Something with like lightning network, it might be useful for people who transact commonly between each other or common channels that will have a lot of transactions flowing through them, but if I want to send you Bitcoin and I'm not running on lightning network, it doesn't help me. If I only send two transactions a year, it's not going to help me.

[0:12:28.3] JM: Okay. Well, let's move on to Ethereum. What are the scalability issues that Ethereum has?

[0:12:33.6] RJ: So the scalability issues that Ethereum has come down to a lot of the similar constraints, such as the ever-increasing size of the blockchain, the fact that certain applications end up taking up a huge portion of the network throughput given that there's so much volume and traffic going through them right. When you have applications that actually have constant user interaction, have people kind of doing things that are settled on chain and you aim to go for mainstream usability, these things can clog up the entire blockchain in its current design.

So at the end of the day, what we care about at Prysmatic labs and for those unfamiliar, is that we're building the first sharding implementation for the protocol, that is we're working on partitioning the Ethereum blockchain into different shards chains that allows us to not only gain transaction, higher transaction throughput, but also maintain decentralization and allows for smaller storage requirements and computational requirements for nodes.

So the idea here is that in the future, we want Ethereum to be able to run on consumer devices even on phones, even in areas where there might not be as fast of an internet connection, taking into account real-world latency and focusing on what we can do to turn this technology into something that is usable and that we can build actually useful applications on top of it.

[0:13:43.0] JM: Raul, you are saying that there is a difference between the Ethereum blockchain and the Bitcoin blockchain, and that Ethereum has a wider design space. So Bitcoin design space, you're just writing information about financial UTXO's. Ethereum design space, you're writing these opcodes. You're writing the programs, the programmatic execution to the blockchain. Why is that significantly different enough that we cannot just replicate the same scalability solutions from Bitcoin to the Ethereum blockchain?

[0:14:21.6] RJ: So you can replicate certain things. You can replicate layer two solutions, right? Which is why things such as state channels, which are extremely powerful and useful in Ethereum have been gaining a lot of traction. So there's a group called L4 ventures that's working on a generalized state channel implementation that allows you to do very interesting things such as like create games on the blockchain and be able to track state through smart contracts and do that in really efficient ways. So we are able to do those kind of transaction throughput optimizing methods.

I think what we really care about with layer one scaling is think about the long-term sustainability of the system itself and realizing that we have a unique opportunity to make it more decentralized or have a unique opportunity to make it more secure and require less capacity from the nodes that are running the chain.

So what we care is tending more towards a world where the network is decentralized and we're able to maintain these constraints at a reasonable capacity.

[0:15:18.4] JM: And why is it important to do that at layer one? Why can't you – Raul, it sounds like there is a flaw to the layer two approach or at least a trade-off that you make in the layer two approach. Why is the layer one approach important?

[0:15:32.5] RJ: Sure. So at the end of the day, like layer two approaches are useful for certain things where you don't demand the complete false security of the chain given that you will be doing certain things off chain, right? You will be trusting certain nodes, certain layers to do these things. However, we care about also being able to guarantee of full security of the main chain and have the main chain be usable. So that is have the core protocol be backed by the full security, allow people to transact another full security of the chain. So, particularly, with state channels or kind of micro payments, if you're like buying coffee at a shop with some Bitcoin, you're using a currency for this reason. You don't demand a full security of the chain, but we have like really important applications that demand censorship resistance. The man having full confirmations from nodes around the world, I'd rather have these things – I think the community would rather have these things happen and be committed to the full security of the chains.

[0:16:25.7] JM: Now, the scalability solutions for blockchains, if we're talking more broadly now that we've explored a little bit around both Bitcoin and Ethereum. The scalability solution's trade-off between decentralization, scalability and security. Explain why there is a trilemma between those three things.

[0:16:44.1] RJ: Right. Sure. So if you partitioning the blockchain, so you have a network of – You have a blockchain and you try to partition it into a hundred smaller chains. What happens is if in a traditional proof of work chain, now people – A malicious attacker only needs 1% of the hash power to take control of an entire shard chain as an example. So in this case, doing so naïvely can actually harm you where it's really easy to take full control of one of these partitions and then that partition can do a lot of interesting malicious things such as issue – Say, that double spent transactions are valid. Send malicious transactions to other sort of shard chains and network.

So this approach kind of you gained kind of a scalability, however, you compromise security by a certain factor. The other thing is if try to go the full route of decentralization, where every single node in the whole network has to validate and verify every transaction, then you obviously sacrifice scalability, right?

At the other end of the spectrum, if you have scalability and you have security, then that means probably have a centralized database. So that's kind like the trilemma that's been explored throughout these different approaches, and sharding has been touted as being able to be right in the middle of this trilemma and kind of tackle all these three pieces in a single solution and achieve it in a way that has a reasonable balance between the three of them.

[0:18:05.4] JM: Explain what sharding is.

[0:18:07.1] RJ: So sharding is the ability to partition the blockchain into K-different shards, where you're able to guarantee that these shards offer the security of the main chain and are also able to – It allows you to increase the throughput of the entire blockchain by having nodes process transactions in parallel. That is, if I have five transactions coming into the network and I have five different shards, I can have one transaction be processed on each of these five shards at the same time.

It also allows you to preserve security – And decentralization. Sorry. By still maintaining a decentralized network of nodes where people kind of report to or kind of commit these changes on these shards chains to the main chain. We'll get into some of the more technicals, but the basic idea here is that, yeah, you're splitting up the chain and coming up with some interesting and fancy schemes to be able to preserve security and decentralization.

[SPONSOR MESSAGE]

[0:19:09.4] JM: Azure Container Service simplifies the deployment, management and operations of Kubernetes. Eliminate the complicated planning and deployment of fully orchestrated containerized applications with Kubernetes. You can quickly provision clusters to be up and running in no time while simplifying your monitoring and cluster management through auto upgrades and a built-in operations console. Avoid being locked into any one vendor or resource. You can continue to work with the tools that you already know, such as Helm and move applications to any Kubernetes deployment.

Integrate with your choice of container registry, including Azure container registry. Also, quickly and efficiently scale to maximize your resource utilization without having to take your

applications offline. Isolate your application from infrastructure failures and transparently scale the underlying infrastructure to meet growing demands, all while increasing the security, reliability and availability of critical business workloads with Azure.

To learn more about Azure Container Service and other Azure services as well as receive a free e-book by Brendan Burns, go to aka.ms/sedaily. Brendan Burns is the creator of Kubernetes and his e-book is about some of the distributed systems design lessons that he has learned building Kubernetes. That e-book is available at aka.ms/sedaily.

[INTERVIEW CONTINUED]

[0:20:45.1] JM: So when the blockchain gets sharded, what's going on with the different shards? Are the different shards all writing to the main chain in parallel or are they batching their own sort of – I guess, they must not be batching, because it's not layer two. Maybe you could just give a little bit of an overview for how those – What does it mean for transactions to be parallelized?

[0:21:11.5] RJ: The key question here is reconciliation, right? How do these shards all come together and agree on what's canonical or not? So the idea is that in an Ethereum, we have a smart contract actually called a sharding manager contract that's going to be living on the Ethereum main chain. So we're still going to be having a main chain for Ethereum, and the idea is that you're going to have smaller shard chains that are then going to be kind of submitting – You have agents that are going to be submitting what happened on these shards chains to the smart contract, and then that's going to be mined into a block on the main chain.

As we can imagine it this way, you can imagine that you have like 10 different shards and then you have in shard one like five transactions happen with these addresses. Shard two you have like three transactions happen. Then what's going to happen, what's going to go on the actual main chain is it's going to be sort of like a summary of what occurred in these shards at a certain time. So what's going to go on the main chain is going to be, "Oh, shard one had these actors committing transactions and finalizing blocks on them. Shard had this thing happening. Shard three had this thing happening." So you're able to kind of, at a high level, aggregate metadata of what occurred across these shard chains in a secure fashion and kind of store that in a smart

contract that lives on the main chain. You have the security of nodes actually mining that block on the main chain. So you're storing a summary of what happened across these decentralized network of shards, right?

[0:22:27.7] JM: How is that different from a lightning network?

[0:22:31.1] RJ: Sure. The lightning network is you're able to like lockup funds and then be able to do these micro transactions off chain, and then finally do settlement when parties decide to cache out. In this system, we have a main chain and we have a smart contract on the main chain that kind of keeps a summary of what happened on the shards. However, these shards are, in the end, are going to be what we call tightly coupled, that is that they're going to be pegged directly to the main chain. So they're part of the protocol. They're not some sort of off chain construction that people come up with. If people want to interact with Ethereum blockchain, they will send transactions that will go through these shards and go into them and their summary will be committed into the main chain. It's part of the protocol. It's part of the Ethereum network. It's not something off chain that people are coordinating.

[0:23:13.9] JM: So there's tighter constraints around how frequently the shards would need to checkpoint with the main chain.

[0:23:21.7] RJ: That is right. That's something that we call a period, and Preston can elaborate a little more on this, but the idea is that how do we maintain security? So we have people that are voting on what's valid across these shards and that's going to go on the main chain, and these people are selected at a certain period. A period is a set of blocks. So we can say that like five blocks is one period, and you have people that are being selected at each period kind of to vote on what occurred in these shards and make sure that we have a bunch of different mechanisms to tell that they're telling the truth, they're not lying, they're not trying to cheat the system. Then for each period, you have people voting on what happened at a certain shard and what was canonical or not, and that kind of allows us to maintain a degree of security in sort of this sharded universe.

[0:24:06.1] JM: Okay, Preston, maybe you could give your perspective on this conversation we're having.

[0:24:09.9] PVL: Yes. So sharding basically is splitting the network into smaller pieces. Each shard would have its own chain. So if you think about like before we have a one Lane highway and now we're going to have a hundred lane highway. So we can, in theory, put a hundred times more traffic through there.

Initially, this is going to be closely tied with the main chain. So every period, which is every five blocks or so, there will be actors who will submit like a summary of the state of the shard chain and then we'll have other actors that will sort of verify that those exists, that the data is available and that the things that happened in that snapshot are true and valid. We have a lot of opportunities here, since this is such a non-backwards compatible change. We have opportunity to sort of change how everything is working. There's a lot of opportunity there.

[0:25:12.0] JM: Non-backwards compatible, so meaning everybody is going to have to upgrade to this system or be compliant with it?

[0:25:18.3] PVL: Yeah. Once we go to the sharding, you can't really go back. There will be a lot of one-way conversions. Once you join a shard, your account is – One theory, say, your account would be assigned to a shard based on the prefix of your public address or your wallet address, and once you've moved from the main chain into a shard chain, there's no going back. So this – It's not backwards compatible. As such, we have opportunity to implement a lot of things that would otherwise be rejected by the community. For example, charging rent for storage. Right now, the state is getting big and getting out of control. It's not a long-term solution to just allow people to pay once for storage and then live forever on the block chain. So one idea is to charge rent for that or rent for access to that data.

[0:26:11.2] JM: So you're saying, today, anybody that can write a transaction to the blockchain, they're essentially getting free storage after that upfront cost. I guess it sounds like you're saying that it was underestimated that the per unit cost of that storage and the ramifications of that.

[0:26:30.8] PVL: Yes. So I can upload a picture of my dog onto the blockchain and then every full node forever from now on will have to download that picture and store it on their machine and validate it for all of time. I paid once and now everyone has to bear the burden of that cost.

[0:26:46.7] JM: So if I'm writing a transaction to the Ethereum blockchain today and it's all going to the same chain, tomorrow I'm going to switch to sharding. How do I determine which shard I'm writing my transactions to?

[0:27:01.0] RJ: Yeah. So that's an interesting point, in which we have to consider where we are putting the burden on. Are we putting it on the user or are we putting it on the protocol itself? We tend to put more burden – There've been discussions that putting burden on the user makes it easier for us, and probably makes it easier for us to guarantee certain security parameters. However, that's not very feasible from the UX standpoint.

So the idea is that sharding will be an in protocol construct. Like you as a user won't have to know anything about kind of how shards are working underneath the hood. You're able to issue a transaction to the network. It's picked up by the peer-to-peer discovery mechanism and it's able to rebroadcast to the different shards, in which that transaction will live based on the account that you're sending it to.

So the ideas is that at least in terms of accessing a state, it should be an in protocol construct, and we shouldn't have you, the user, have to provide us with like what state the transaction will require or what sort of Merkle paths in the Merkle tree we'll have to modify, and this is something that we call access lists. So I think there are some opinions about access lists, and the idea is that, in an ideal world, the users should not have to know about what's happening in the sharded system.

[0:28:06.9] JM: Okay. So are there race conditions that can come up between these different shards, because if I – You talked about database. If you're talking about one big database, we know that different database replicas can have race conditions if they're not properly handled. So how do you avoid race conditions among shards?

[0:28:27.3] RJ: Sure. So that comes up in the context of cross shard transactions. So there's a canonical problem that's been posed, which is called the training hotel problem, which is that if you book a train and you book a hotel, you want both to go through or none at all. You don't want to end up stuck with the hotel going through, but not your train or the train going through and not you're hotel.

So what this means in the context of Ethereum is that if you have a transaction that depends on another transaction finalizing and these kind of live on different shards that have different latencies, have different parameters, different types of nodes going on them, you can run to the condition where the whole system would break if one of the things goes through and maybe the other one doesn't beforehand, and someone's accountant ends up having way more Ether than they actually do, and that just creates a whole issue throughout the whole chain.

So the way we do this is it's currently heavily under exploration, but there's something called a receipts method, where you have a transaction on one chain kind of issue a debit receipt. So, say, I'm sending money to Preston whose account is on a different shard, that will be debited from my account and stored in something called a receipt. Then on Preston's shard chain, the shard would be able to check this receipt's tree and see that there was a debit from my account on this other shard and there's going to be a credit given to Preston on his shard. So this way we kind of keep a global receipt's tree that is able to just be used for this purpose itself. So this is still heavily under exploration. The core developer [inaudible 0:29:56.4] is also working on an approach where we have something called merge blocks, and which you have blocks on that live on these different shards that are kind of intrinsically linked to each other. So they have certain aspects in these blocks that are linked to other blocks and other shards.

That allows you to fix a whole issue of what we call atomicity or lack thereof, which is we don't want transactions to have these risk conditions happening. We want these cross shard transactions that happen in order to be atomic. So that's one other approach. Then, again, this is something that will come later down the line. At the moment, we're more focused on kind of getting a basic sharded system to work where you split up and partition the chain and have certain security considerations that are met and allows you to see how the throughput will go up.

[0:30:40.3] PVL: Yes. So what we're envisioning is that a lot of transactions will be contained within the same shard. So you can think, for example, you have a smart contract that also cause five other smart contracts. You can deploy these in the same shard such that you don't have to do as many cross shard transactions.

[0:31:01.2] JM: What are the potential issues that can arise as we're going from this world with no sharding to a world with sharding?

[0:31:11.5] RJ: Sure. So I guess we can talk a little bit about the security considerations that happen here, and probably also the standard concerns about wealth distribution and kind of inequality that could occur based on the proof of stakes system. So, as I mentioned before, why sharding is a lot better in kind of a proof stake system is because I mentioned that we have these people that are going to be voting on what's happening across these shard chains. So we refer to these people as notaries.

So these people are going to be selected by the smart contract. They're going to have to stick ether upfront to below to participate in this kind of game and are going to be the ones in charge of actually checking that, that the data is available, the things happen on certain shards and kind of vote on making things canonical and finalizing blocks. So the idea here is that you have the standard concerns is that you're making the rich richer. You're going to have these notaries have a lot of control, and it's how can you prevent something called the validator's dilemma by which certain notaries try to be lazy and try to piggyback the work of other notaries. So whenever you have like these sort of in protocol finalize voting systems, there is a lot of risk that is involved, especially in the way you randomly select these people.

So we wouldn't want these people voting on shard chains to know exactly what shards are going to be participating in or what time they're going to be voting on, because at that point you're able to do some interesting things to game the system. Like if I knew that I'm going to be a notary selected on shard five in like 20 blocks, for example, you can run into some very risky cartel-like scenarios. So it's about figuring out what's a really good source of randomness. How can we design this? What are the constraints that we have the keep in mind the system? How can we take into consideration wealth and equality kind of that comes through with the proof of

stake system, and these are all things that have been addressed really well in the latest Casper presentations.

Basically, sharding runs in parallel with the Casper researcher that's going on, and the idea is that by the time we have something on a public test net on the main net, we should be able to have the Casper contracts live and be able to work with that, because the great thing about sharding is – And with proof of stake, is that, with Casper, we already have a global validator set aid that we can select these notaries from and we can do a lot of interesting things from here and take advantage of the flashing conditions and other benefits that come with Casper.

[0:33:28.4] JM: Explain Casper in a little bit of detail. We have a whole show on Casper, but maybe you could just give an overview for what Casper is.

[0:33:35.9] RJ: Sure. So Casper is one of the approaches to implementing proof of stake, and proof of stake is an alternative consensus mechanism for blockchains. With proof of work, you're spending physical electrical output, putting that into securing the chain, which as we know has huge ecological footprint, creates a lot of issues, is very costly to different countries and electricity costs are even mounting more and more, as I think Bitcoin has surpassed the electricity requirements of Denmark and countries as large as that.

With proof of stake, you're able to create in which you no longer require this sort of electrical input to power and secure a network. Instead of relying on a system of rewards such as proof of work, you're allowing a system of punishments by where you have people that are actually staking their ethers. So that means they lock up their ether inside of a contract and use this kind of these funds to vote on kind of blocks and finalize them.

Basically, the idea is that if these actors are malicious and they try to cheat the system, they're going to be losing their stake. So why is this important? This is important because in the case of a 51% attack on the chain, in a proof of work system you can do something called a spawn camp attack, where – Sure, you will require a lot of hash power to do a 51% attack, but if you require that hash power, you can just keep doing it over and over and over again until you succeed or you're able to just completely overwhelm the chain and the network.

With proof of stake, to do a 51% attack, that means you would basically be have to be burning your funds every time you want to do an attack. So this is equivalent to basically your ASIC farms just burning down every time you try to mount an attack on the network. So proof of stake gives us a lot of very strong security guarantees and the system based on punishments like these is heavily – Basically, disincentives people that try to cheat the system in a lot stronger ways.

[SPONSOR MESSAGE]

[0:35:27.8] JM: Software workflows are different at every company. Product development, design and engineering teams each see things differently. These different teams need to collaborate with each other, but they also need to be able to be creative and productive on their own terms. Airtable allows software teams to design their own unique workflows. Airtable enables the creativity and engineering at companies like Tesla, Slack, Airbnb and Medium.

Airtable is hiring creative engineers who believe in the importance of open-ended platforms that empower human creativity. The mission of Airtable is to give everyone the power to create their own software workflows, from magazine editors building up their own content planning systems, to product managers building feature roadmaps, to managers managing livestock and inventory.

Teams at companies like Condé Nast, Airbnb and WeWork can build their own custom database applications with the ease of using a spreadsheet. If you haven't used Airtable before, try it out. If you have used it, you will understand why it is so popular. I'm sure you have a workflow that will be easier to manage if it were on Airtable. It's easy to get started with Airtable, but as you get more experienced with it, you will see how flexible and powerful it is.

Check out jobs at Airtable by going to airtable.com/sedaily. Airtable is a uniquely challenging product to build and they are looking for creative front-end and backend engineers to design systems on first principles, like a real-time sync layer, collaborative undo model, formulas engine, visual revision history and more. On the outside, you will build user interfaces that are elegant and highly customizable that encourage exploration and that earn the trust of users through intuitive, thoughtful interactions.

Learn more about Airtable opportunities at airtable.com/sedaily. Thanks to Airtable for being a new sponsor of Software Engineering Daily and for building an innovative new product that enables all kinds of industries to be more creative.

[INTERVIEW CONTINUED]

[0:37:47.7] JM: So that episode with Karl Floersch in the past was the one where we explored Casper and proof of stake in a little more detail. I think if I'm understanding this correctly, so there's proof of stake and there's sharding, and proof of stake is a method by which any transaction on the Ethereum main chain will be validated in the future once Casper is fully rolled out and it'll be those transactions will be validated by people who hold – Who have put up a sake, and if they validate falsely, then they will lose that steak that they have put up, and this is a way for processing any transaction across the Ethereum blockchain.

The sharding side of things is a smart contract that these different shards are writing to so that you can have the scalability across the different shards. The shards are reconciling through that smart contract over a periodic basis. So there must be some economy of scale across if you're gaining some sort of scalability advantage from that sharding process. There must be some economy of scale by which that smart contract that is negotiating the sharding is able to process those sharded digests or whatever it is, however you want to call it, faster than the total cardinality of the transactions across all those different shards. So help me understand, what would be the difference between the proof of stake Ethereum blockchain without sharding processing all of those transactions individually versus the world that you're describing where you have this contract that is negotiating the sharding. What is that contract that's negotiating the sharding doing in order to process those transactions in a more economical fashion?

[0:39:43.5] RJ: Sure. So what makes it economical is that, basically, you're able to fit in a lot more information of what occurred throughout the entire network into a single main chain block. So with Casper – Basically, the purpose of Casper is to just have a more economically efficient, secure consensus algorithm. It doesn't do anything to change the fact of how much information is going to be stored in each block.

With sharding, for example, in each block I'm able to store a lot of metadata of what happen across every single chard, and each of these single shards are going to have their own blocks that have a lot of information packed into them as well. So you're able to kind of compress this entire set of occurrences across shards into a single block in the main chain, and that's kind of the difference here. So, yeah, Casper is just basically purely a consensus algorithm. Whereas we're actually trying to improve the whole – How much information can be stored on a single block. How much can be processed at a single time.

What's interesting here is that Casper also gives you something called finality, like deterministic finality. So that means that like in Casper, once the validators actually vote and reach consensus on what a canonical block is, that's basically finalized. That is like it's kind of irreversible. Whereas in Bitcoin, for example, finality is probabilistic. That is that proof of work is an NPR problem that you have to iterate over a bunch of nonsense to reach a certain solution, and there's not a certain amount of time by which you will get there, but an average should be around 10 minutes. So you have this probabilistic consensus versus a deterministic finality that you gain. So that's the difference.

Whereas we're leveraging the finality to kind of enforce constraints on a system so it doesn't go out of control, given that you said that, "Oh, the speed of sharding could be really, really fast," to the point where how can the smart contract keep things under control? We're also packing more information into what happens in a certain block.

[0:41:30.9] JM: One thing that's interesting about this, it sounds like there could be multiple implementations of sharding, like just depending on who deploys what smart contract. Is that right? Because you're just developing one specific implementation of sharding.

[0:41:45.2] RJ: So that's a good point to talk about. So, yeah. I think, yes, there can be different implementations, but not really of the contract. The ideas that we're all going to agree upon a certain contract. So there are a bunch of different teams working on sharding implementations. What they're working on is the actual client. So the actual code that runs these sort of shard chains, kind of connects to each other, connects other clients via P2P. Is able to submit stuff to the smart contract, but the idea is that when we deploy in a public test net, we're all going to be using the same contract just as people are going to be using the Casper contract. So this is

something that's kind of going to be agreed upon by the community and also different implementers.

So the Ethereum Foundation Prysmatic Labs, which is us, we have drops of diamond, which is a team working on a Rest implementation and a few other teams also working on sharding. So we're all going to be agreeing on the single smart contract that we're going to be using, but our client implementations are going to be different. Just like you have Git, you have parity, you have other teams in the current Ethereum ecosystem.

[0:42:40.4] JM: Okay. So what are you working on at Prysmatic exactly?

[0:42:44.1] RJ: Sure. Preston, do you want to jump in?

[0:42:45.9] PVL: Yes. So we're extending the Go Ethereum project to support sharding. Basically, we're implementing all the actors that participate into a sharded system and we're also implementing the interface by which a common user can interact with a sharded system. For you and I, just in transactions, we can do that.

[0:43:11.2] JM: Okay. So this is sharding within the context of the Go Ethereum client.

[0:43:16.8] PVL: That's right.

[0:43:17.5] JM: Can you talk a little bit more about the spec of what you're building? What does that job require?

[0:43:23.5] RJ: Basically, we all communicate through something called Ether Research, which is the official forms for the Ethereum community kind of, the Ethereum Foundation, and we through together with the other different teams to come up with a minimal spec that is acceptable for a sharded system.

So Vitalik put out something called a minimal sharding protocol, just to think around a month ago, and that's kind of what all the teams have been following to implement. The idea here is that we have to implement these people called notaries that are going to be voting on kind of on

this contract. We have to implement like these actors that are processing sharded transactions, storing them on their local machine and broadcasting them via P2P. We're also working on modifying things that we have a unique opportunity to do, such as picking a different storage backend for Ethereum. Given that this is not a backwards compatible change, we're able to this really cool interesting changes.

So that's kind of what it entails. We're working with the Eth research team and the other people working on sharding to come up with a reasonable spec that we all agree to. Constantly communicating with people that are following us and just leading a very open kind of transparent development initiative to get this done.

[0:44:31.0] PVL: And we're also proposing ideas and contributing research wherever we find new knowledge.

[0:44:36.5] JM: Okay. So I want to understand how a transaction being written to the Ethereum blockchain today will contrast with a transaction being written in a post Casper, post sharding world. So maybe we're talking about the creation of a CryptoKitty or the purchase of a CryptoKitty. Whatever transaction you think would make for a good, tangible example for people to understand. Can you contrast what happens today with what will happen in the near future?

[0:45:11.5] RJ: Sure. So for the example of CryptoKitties, for example, any other Dap, you could imagine a imagine a future in which you have a kind of shards that are specific to certain types of Daps. You have transactions that are more like games. They can live on a certain shard that is a little bit more optimized and kind of as actors that are heavily skewed towards these sort of Daps, and the ideas that it shouldn't change as much on the user side. Perhaps one of them things that will concretely change the idea of rent, so paying for storage rent. But the idea is that, yeah, you'll broadcast transaction to the network. Based on your address and where your address lives in the shard space, certain nodes are operating in the certain shard will pick up the pending transaction. They're going to be processing into a shard block, which we refer to as a collation, and then you have people that are going to be submitting the escalations header, which is kind of like a description of its metadata to the smart contract that lives on the main chain.

So then you have all these people submitting these transactions to the main chain and then you have the people called notaries that are voting on what happened in these shards. So what they're going to be doing is they're going to be downloading you transaction and they're going to be checking like, "Okay. Did Jeff actually send this CryptoKitty? Is the data available? Am I able to download it? Okay, cool. Let's finalize it through a proof of stake system." The block on the main chain that contains metadata, what happen in the shards is in mind or basically finalized in the context of Casper, and then there you go. You have a record immutably stored on the blockchain of a full transaction that occurred through a sharded system.

Yeah, like I said, the routing is going to be determined on where your address lives in the sharded space. For example, addresses that are previously 0X01 could live on shard one, 0X02 could live on shard two. Etc. Etc.

[0:46:55.6] JM: Yeah. To speak more towards that, as an end-user, if I'm sending a transaction to CryptoKitties and I happen to live on the same shard, it will feel exactly the same as it does today. But if it was a cross shard transaction., it might be a little different where you have different type of receipt and maybe it takes a little bit longer, but it should still feel really just like it does today.

[0:47:18.3] JM: Okay. I did a show a while ago about Plasma. So I know Plasma is a mechanism for Ethereum subchains. Does Plasma relate to this conversation? Is that the mechanism for sharding that we're talking about?

[0:47:34.2] PVL: From my understanding, Plasma I more of a side chain. It is really similar and sometimes we ask ourselves, "Are we describing Plasma right now?" But they are quite different.

[0:47:46.5] RJ: Yeah. So the idea with Plasma is that we have kind of this off chain kind of Plasma chain node where transactions are going to be rerouted to and then have all these different things that occur on this Plasma chain that is kind of – All these does happen off chain though. There is a reconciliation process by which things are committed to the main chain.

So once again, it's very similar to kind of like a lightning network approach, but instead having a bunch of different nodes, you have something called a Plasma chain, which is a single kind of entity. In the future it will be various different entities that are going to be reconciling information about what happened in there.

So it is an off chain scaling solution, and the idea is that Plasma should be kind of in production and happening way earlier than sharding. That's kind of like the short term solution, and they are indeed quite similar in a lot of the approaches, but it's something that we need to have that we're probably going to have earlier. Whereas sharding, it's going to entail a lot of different things including core protocol modifications to security, account abstraction and such that will come much later down the line.

[0:48:45.3] JM: Okay. So as we begin to wrap up, I have done a bunch of shows recently around cryptocurrencies, as Raul, at least you have heard, and one thing I'm curious about is giving your strongest bear case for cryptocurrencies, is there a strong negative case for why tokens and ICO's and cryptocurrencies in general might not make sense?

[0:49:09.3] RJ: I think one of strongest things to think about is user adoption, right? So thinking about how can we really package things up for people around the world to be able to use this seamlessly and how can we do so in a way that does not sacrifice decentralization or security or censorship resistance, right? There are a bunch of approaches to improving blockchain UX and a lot of them involve kind of having a single entity or a company controlling people's wallets, controlling their private keys, doing a lot of these things for them for the ease of user experience. Basically, at the end of the day, one could argue that that doesn't give you more than just having a centralized database would. So at the end of the day, it's how can we come up with something that people actually use and people from less privileged backgrounds and kind of poorer countries are able to use effectively. So that's one of the fears that I think a lot of my colleagues also share with respect to this. So we need more people also working on that side of crypto.

[0:50:01.8] JM: And what about tokens more fundamentally? So I think about IPFS/Filecoin as an example. Why does the IPFS/Filecoin network need its own token? wouldn't function equally well just using Ethereum or USD as its mode of value transfer?

[0:50:20.2] RJ: Right. That's the idea. So we see a lot of token that have maybe a few marginal improvements and have a few differences in kind of their own model to make things work. The problem is that when you create a whole new protocol, there are issues of fungibility with other tokens. So their approach is to do that, such as like bridges, sidechains, kind of you see a bunch of those happening and a lot of innovation happening around that space. So there's the idea of are we going to split up into having a world in which there's a token for everything and is that going to be really beneficial, or are we going to have a world in which we have a few tokens dominating in certain design spaces? And that's another bear case that people say like, "If we have an extremely, extremely diversified token economy, it's not necessarily a good thing for the world." The thing is that we don't know. I think that's something that we have to carefully think about as we design these projects.

[0:51:06.5] JM: All right, last question. Is there any other bottleneck besides scalability that you think it will take to knock down in order to get to a place where we have widespread deployment, widespread usage of smart contracts?

[0:51:21.6] RJ: Yes. So I think something interesting I heard the other day was that there are two types of people in blockchain, is people who what it is and they embrace it and believe it, and there are people who don't know what it is and they're skeptical and they don't understand it. So I think a big bottleneck for global adoption and mass adoption and getting people to participate in this ecosystem is some education. It's something that as researchers and developers were not really talking about, because it's probably not what we're best at, but it is something to think about like long-term, like getting the awareness out there, getting people to understand and to really embrace it, because a lot of people, they don't know how it works.

[0:52:07.0] JM: Indeed. I think it's a good place to close off. Preston and Raul, thank you both for coming on Software Engineering Daily. It's been great talking to you.

[0:52:14.0] RJ: Thanks for having us.

[END OF INTERVIEW]

[0:52:18.8] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use, and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plug-ins. Use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

With GoCD running on Kubernetes, you define your build workflow and let GoCD provision and scale your infrastructure on-the-fly. GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn about how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team who have talked about building the product in previous episodes of Software Engineering Daily, and it's great to see the continued progress on GoCD with the new Kubernetes integrations. You can check it out for yourself at gocd.org/sedaily.

Thank you so much to ThoughtWorks for being a longtime sponsor of Software Engineering Daily. We are proud to have ThoughtWorks and GoCD as sponsors of the show.

[END]