EPISODE 605

[INTRODUCTION]

[0:00:00.3] JM: Big blocks or small blocks? This is the fundamental question of Bitcoin scalability. The argument for big blocks is also known as on-chain scalability. Under this strategy, each block in this append-only chain of Bitcoin transaction blocks would grow in size to be able to support lower transaction fees and higher on-chain throughput. A set of Bitcoin users who supported this idea forked Bitcoin to create Bitcoin Cash, a version of Bitcoin that has a larger block size.

The argument for small blocks asserts that scaling Bitcoin does not require a larger block size. Under the model of the small blocks, the scaling demands of the Bitcoin blockchain will be handled by sidechains. A sidechain is a network of person-to-person payment channels that only reconcile with the Bitcoin blockchain to checkpoint batches of transactions. These sidechains can be connected together to form the Lightning Network.

Lightning Network is hard to implement. To implement a Lightning Network requires solving real-world distributed systems problems that are unprecedented. It's much more complicated than deploying a blockchain with a larger block size. In addition, opponents of Lightning Network suggest this will lead to a centralized banking system of being constructed on top of Bitcoin. Opponents of Lightning Network fear that instead of a decentralized payments network, the world of a Lightning Network would be a lower-cost version of the present-day financial system; a world in which JP Morgan and Blockstream would partner up to battle Coinbase in a decentralized, but actually centralized war for control of the unbanked.

These big blockers, the people who are arguing against the Lightning Network, they argued that the new banks on the Lightning Network will be just like the old banks. They're going to be censorious of transactions and they're going to be held in the domineering palms of the global financial kleptocracy. At least, that's the position of the big blockers. Why bother with the Lightning Network approach? Why are we building this inelegant, kludgy system of off-chain, potentially centralized banking to dato-complexity? Why not just increase the block size

indefinitely and keep things simple? Even if we increase the block size today, couldn't we still deploy Lightning Network in the future while appeasing the transaction volume of today?
Well, one major reason is that growing the block size does have a cost. The bigger the block size, the more demands it places on any node that wants to maintain a record of those blocks. If you grow the block size today, you forego the experiment of seeing whether a small block size, plus Lightning Network could in itself handle the transaction volume of a global decentralized financial system. Of course, this framing of big blockers versus small blockers is a conveniently polarized reduction of a much more granular reality.

To believe that there is no subtlety between the two sides of this big block versus small block debate is to underestimate the number of dimensions involved in this argument. This is an unfortunate side effect of rigidly programmed Twitter bots that are on one or another side of the big block versus small block debate, and also a general political atmosphere of 2018 in which your lines in the sand are demarcated by which Subreddit, you choose to affiliate with.

That said, my impression is that the more experienced engineers in the cryptocurrency ecosystem are overwhelmingly on the side of small blocks, plus Lightning Network as the most promising approach to scaling Bitcoin. Take whatever side of the debate that you want. A single line of Bitcoin core code that is actually committed speaks much louder than an avalanche of tweets.

In today's episode, Jameson Lopp joins the show to explain why Lightning Network is an appealing engineering construct. We play the devil's advocate and contrast Lightning Network with a big block approach, as well as a big-block plus Lightning Network approach. Jameson also describes his experience working within the Ethereum ecosystem and gives a sober explanation of some of the issues that Ethereum scalers may themselves encounter.

As political as this issue is, what I like about the big block versus small block debate is that people get really impassioned around it. When they get impassioned, it's more entertaining to watch them debate. When people are more entertaining, it makes a dry subject more accessible. If you're having trouble understanding cryptocurrency scalability, I recommend checking out the past interview we did with Roger Ver.

There's also an interview out there on a podcast that I will link to in the show notes, where Jameson and Roger – Jameson is today's guest, Roger is a previous guest on the show, but there's another podcast that I listen to that again, I'll link to in the show notes, where they had a debate over this subject over big block versus small block Lightning Network versus no Lightning Network, and it was really impassioned and super useful. If you're having trouble getting into this topic, I recommend checking out that episode that again, I'll link to in the show notes.

If you're looking for all of our episodes about cryptocurrencies, you can check out the Software Engineering Daily app for iOS, or Android. We have all of the episodes in there. We've got related links, we've got conversations that people can have with each other about the episodes, and they're all searchable, you can save them offline. We made the Software Engineering Daily apps for our Software Engineering Daily audience. If you're curious about those back catalogue episodes, then you can check out those apps.

As always, you can send me an e-mail or send me any suggestions for the show, or whatever else is on your mind, jeff@softwareengineeringdaily.com. I'd love to hear from you.

[SPONSOR MESSAGE]

[0:06:35.5] JM: Every team has its own software and every team has specific questions about that internal software. Stack Overflow for Teams is a private secure home for your teams' questions and answers. No more digging through stale wiki's and lost e-mails. Give your team back the time it needs to build better products.

Your engineering team already knows and loves Stack Overflow. They don't need another tool that they won't use. Get everything that 50 million people already love about Stack Overflow in a private secure environment with Stack Overflow for Teams. Try it today with your first 14 days free. Go to s.tk/daily.

Stack Overflow for Teams gives your team the answers they need to be productive, with the same interface that Stack Overflow users are familiar with. Go to s.tk/daily to try it today with your first 14 days free. Thank You Stack Overflow for Teams.

[INTERVIEW]

[0:07:48.4] JM: Jameson Lopp, you are a engineer at Casa and a noted Bitcoin writer. Thanks for coming on Software Engineering Daily.

[0:07:56.8] JL: Thanks for having me.

[0:07:57.9] JM: We've had some previous shows where we talked about the basics of Lightning Network, some pros and cons of various approaches to scaling a blockchain. Today, I'd like to start with the focus on why Lightning Network is the preferred scaling mechanism of the Bitcoin core developers. Maybe we can talk some about the actual state of the engineering of Lightning Network. I know you're not a total expert in the field, but I think you're enough of an expert to give us some overview, and to give some context in relation to other episodes of Software Engineering Daily.

I had Roger Ver on the show a while ago. He gave his argument for a larger block size, why that's in his view better to scale the Lightning – than Lightning Network. Because I had Roger Ver on the show, I actually got flooded by anonymous crypto-Twitter people who were either harassing me, or supporting the episode depending on what side of the bot line they fell on. Can you give a brief overview to these two alternative ideas, the larger block size versus Lightning Network?

[0:09:06.3] JL: Sure. It really comes down to trade-offs, right? That's why it can get pretty heated, especially when people start saying, “Oh, that will never work,” it's because there's so many variables that play that really what we're talking about is what trade-offs we’re willing to make in order to get it to “work.” Then of course, a lot of times you have to define what work really means in the first place.

The trade-offs that I find that Bitcoin scaling debate really comes down to is prioritizing low transaction cost for the user at the potential expense of a high validation cost for something who is running their own node, to make sure that nobody is breaking the rules on the network,

versus prioritizing a low validation cost of the entire system at the possible expense of high transaction cost.

That is really what this has all come down to of course, this is not only technical, but also economic arguments, a lot of philosophical and ideological arguments. There's no objective correct way to scale Bitcoin. It really comes down to what trade-offs you're willing to make.

[0:10:19.3] JM: Even the trade-offs themselves as you laid them out, those are hypothetical trade-offs. We don't actually know what would happen in practice if the Bitcoin blockchain scaled up its block size and the community was run wild in that world, versus having Lightning Network implemented and having the world run wild similarly. What gives us such confidence that we know what the outcomes would be in these two hypothetical scenarios?

[0:10:49.3] JL: Well, I see it more from a security standpoint. My standpoint has changed over the years, where several years ago, I was actually more on the scale everything on the blockchain camp, and that's because my perspective at the time was from a capacity planning perspective. I had worked with big data, Hadoop clusters, HBase clusters and whatnot for a number of years, doing large-scale analysis for marketing companies. I looked at Bitcoin blockchain from a capacity planning perspective of saying, “We need to have a lot of headroom available. Otherwise, the user experience degrades and that's bad for Bitcoin.”

On the other hand, now I'm actually looking at it more as a security perspective of Bitcoin as a whole ecosystem is setting this block size cap, not because necessarily it would kick a bunch of people off the network because they couldn't run a 2-megabyte block size on their nodes, but more because it's very hard to control these networks.

Once you do make a change to increase the capacity, it's going to be very hard to reverse position and go back if you make a mistake and you pushed it too far. I’m these days leaning more towards being very conservative about the changes that we make to the protocol, simply because I think that it needs to be more of a like, aerospace engineering perspective taken to the caretaking of the protocol and the network, rather than a move fast break things perspective.

It's interesting if you look at the crypto ecosystem more broadly, you can definitely see that there are some projects out there that are more on the move fast break things side. They can be interesting to watch, but I just don't think that that is the right way to engineer what is supposed to be a sound monetary system.

[0:12:50.8] JM: Your position is more about one of humility. We know that the current Bitcoin block size is working decently. We can see, we can imagine a world where off-chain scaling works. Even if we don't have it just yet, it's not worth putting the cart before the horse and saying that because we can't – we won't have Lightning Network for another two years, or five years, or 10 years, we shouldn't threaten the experiment by increasing the block size.

[0:13:25.4] JL: Right. It's making these trade-offs of we're protecting Bitcoin, the ecosystem and the network at the expense of having a poor user experience at times when there's a lot of demand, a lot of congestion on the network. You can definitely go the other way with it and try to make the protocol more malleable, more flexible. That's why it's actually interesting to see really this side by side, like AB experiment. Though now, it's not just AB. It's like 40 different forks of Bitcoin that are all out there and trying to compete.

It's not all that dissimilar to what has been happening for the past seven years, where a lot of alt coins have come out and just tweaked a few parameters and said, "Okay, now we're going to be able to be a lot better than Bitcoin at doing this one thing," whether that's throughput, or privacy, or what have you.

At the end of the day, Bitcoin still manages to retain on top position, and you could speculate as to why that is, but I think it's mainly because it has proven itself to be very robust and is not going to be changing for the whims of a few people.

[0:14:38.1] JM: Probably people are making an implicit bet on the same conservatism, the same humility that we're talking about here. It's as much about a bet on the technology as it is the ethos of the people who are making commits to the technology.

[0:14:57.2] JL: Yeah, I mean, and it's a diverse ecosystem. I definitely hate to ascribe power to the developers, and this happens very often, especially when debates get heated is you start seeing people looking for scapegoats, or trying to find centralization of power, so that they can

blame somebody. For all of the blame that is being put out there against Bitcoin developers, and I don't even like saying Bitcoin core developers because there's like seven or eight different implementations of Bitcoin, and I hold no allegiance to Bitcoin core as an implementation over any other implementation. If they decided to start doing something with which I disagreed, I would be the first one to jump up and say, “You know, what? I'm going to switch to using some other implementation, because I don't agree with what you’re doing.”

It's really about the ecosystem as a whole, because these developers can't just make changes. They have to propose changes, get consensus within their own developer group, and then put them out there for the rest of the network to decide whether or not to adopt. It's interesting to see what happens with these network splits where you have 15%, 20% of the users on a network disagree with a certain change, or certain lack of change and then they split off and, and go create their own network.

[0:16:16.5] JM: When you say there are seven or eight versions of Bitcoin, are you talking about the Bitcoin cash, Bitcoin gold flavors of Bitcoin? Are you talking about things that are compliant with Bitcoin itself?

[0:16:29.8] JL: Right. I'm talking about implementations that are compatible with the Bitcoin protocols. Of course, just talking about this gets really difficult to discuss, because like you said, there's also dozens of different actual networks that call themselves Bitcoin something. It’s one of the interesting aspects going back to I guess, the governance or lack of governance of the systems. It really is this new crypto-anarchy that's emerging on the internet, and we're still trying to figure out how to deal with some of the weirder aspects that happen when nobody controls even the usage of a name of a thing.

[0:17:10.4] JM: There are full nodes, there are miners, there are simplified payment verification nodes, there a variety of node types on the blockchain. What is the weakest type of hardware that should be able to run a miner, or that should be able to run a full node? Should you be able to run a miner on a Raspberry Pi?

[0:17:35.3] JL: Yes. This is one of the fundamental ideological questions. I think that you don't find people that are really setting specific monetary goals. It's usually one extreme or the other.

It's either we want as many people as possible to run a full node, you know, possibly even on a mobile phone with a very crappy data connection versus the other extreme of we don't really care who can afford to run a node, because anyone who is doing enough economic activity will figure out a way to afford to run a full node.

If it costs tens of thousands of dollars a month to run a full node, then that's still okay because then the businesses that are making the most money can afford to do that and it'll be worthwhile to them. This goes back to what we were saying earlier is there's no real objective, correct answer on that. It's just a matter of your opinion.

[0:18:32.3] JM: With the larger mining pools, we already see a notion of sharding within the mining process. You have miners that are teaming up together to look for a valid block. I mean, I say sharding, that's probably an improper use of sharding. You could imagine a more sharded version of mining, where you could have a trustless system, where me and you and 500 other people in it, in an IRC channel are all running some open-source software that is running some – is maintaining some shard of the Bitcoin blockchain across our phones, and therefore, we have a sharded full node essentially, and that would give, maybe it's on IPFS or something if we want to get super fancy. Do you think that's a plausible model to getting everybody running a full node?

[0:19:32.0] JL: I haven't really seen much research on anyone trying to shard a full node. The most node sharding research I've seen seems to be on the Ethereum side. Intuitively, it seems like that is going to change the security model, but it's going to really matter on the details. I think a way that you could consider the nodes or the network to be sharded in a trustless fashion is actually via sidechains, where you can then have these pegs between the main Bitcoin blockchain and some other blockchain.

If you're not using that sidechain, you don't care, you don't have to validate it. If you are using it, then you might care and you might want to validate it. That way, you can have a specialization between different blockchains that are trying to do certain things better, whether it's privacy, or throughput, or what have you, and then only the people who actually care about that particular functionality would then have to worry about validating it.

[0:20:37.5] JM: When I propose something like sharding in the Bitcoin ecosystem today, just saying it to you, it betrays a lack of understanding of the engineering complexities of implementing something like sharding against the Bitcoin blockchain. I think there's a lot of lack of understanding of the finer engineering details of how hard it is to maintain. If you think of some magical solution to the current Bitcoin scale and projected scaling difficulties, or present-day scaling difficulties depending on how you look at it, it's very easy to conjure up solutions that sound like they work, that sound quite good in a white paper, but probably would be less good if you ran the experiment, tried to roll it out, tried to get enough engineers behind you. How do you develop an intuition between the scientific laboratory, things that sound good in a white paper, versus things that are actually realistic?

[0:21:43.9] JL: Yeah. That's tough. Actually, I would say that there's probably a lot of people in academia who have been displeased with how their work in this space has been received or not received as it were. Mainly because there's a lot more to deploying changes on these networks than just coming up with a great idea. You actually have to convince the people on the networks to adopt it, and that can be a much more challenging thing than just writing a paper and doing an experiment and getting peer review on it to show that it is theoretically a good idea.

You actually have to be willing to engage with the technical community and basically go through a gauntlet that is not guaranteed to end well for you, and sometimes that results in people then going off and creating their own network, because they got rejected from getting those changes into Bitcoin, or whatever other network that they were trying to change.

This is like once again a sort of, lack of governance aspect of it. For the average person, it can be pretty baffling to try to keep track of proposed changes, and there can definitely be a lot of lingo and game theory and other understanding that's going on under the hood, that's not obvious if you're just reading these papers, or discussions between developers. That's one, I think people start gravitating around the other technical folks on the ecosystem who are better at dumbing things down.

[SPONSOR MESSAGE]

[0:23:40.1] JM: We are running an experiment to find out if Software Engineering Daily listeners are above average engineers. At triplebyte.com/sedaily you can take a quiz to help us gather data. I took the quiz and it covered a wide range of topics; general programming ability, a little security, a little system design. It was a nice short test to measure how my practical engineering skills have changed since I started this podcast.

I will admit that, though I've gotten better at talking about software engineering, I have definitely gotten worse at actually writing code and doing software engineering myself. If you want to take that quiz yourself, you can help us gather data and take that quiz at triplybyte.com/sedaily.

We have been running this experiment for a few weeks and I'm happy to report that Software Engineering Daily listeners are absolutely crushing it so far. Triplebyte has told me that everyone who has taken the test on average is three times more likely to be in their top bracket of quiz scores.

If you're looking for a job, Triplebyte is a great place to start your search, it fast-tracks you at hundreds of top tech companies. Triplebyte takes engineers seriously and does not waste their time, which is what I try to do with Software Engineering Daily myself. I recommend checking out triplebyte.com/sedaily. That's T-R-I-P-L-E-B-Y-T-E.com/sedaily. Triplebyte, byte as in 8-bytes.

Thanks to Triplebyte for being a sponsor of Software Engineering Daily. We appreciate it.

[INTERVIEW CONTINUED]

[0:25:38.1] JM: Okay, I want to talk in some detail about Lightning Network. By the way, if anybody out there is listening who works at Lightning Networks, or perhaps Blockstream, I did a show about Lighting Networks with somebody from Blockstream, with Rusty Russell from Blockstream, I think it was like two and a half years ago before I even understood how Bitcoin worked. That show was not great, but it was decent. Anybody out there that really understands this stuff, or was working on this stuff in a lot of detail I would love to do a show, because again, I know you're not working on this day-to-day, but it's like I was really looking for somebody and I

just found you. I heard you do a podcast or two about this. Let's talk about Lightning Networks for a bit.

Lightning Networks as I understand are a way of networking together payment channels. You and me set up a payment channel with each other against the Bitcoin ecosystem, and Todd and Sue also set up a payment channel between each other, and Jameson also sets up a payment channel with Sue. Therefore, I can make a transfer of money to Jameson, Jameson can make a transfer of money to Sue. Therefore, there is an implicit payment channel between Jeff and Sue, and all of this can take place off-chain. Explain – go tell me if that's correct, like give an explanation for how Lightning Network works.

[0:27:05.8] JL: Yeah, reasonably accurate, I mean, in order to create these channels though, you are making on-chain transactions. You're basically anchoring into the Bitcoin blockchain, or Litecoin blockchain, or really any network that has support for a few primitive functions like time-locking and multi-sig, because we need to be able to construct these hash time lock transactions, which are used to update the state of a payment channel.

When you're creating a channel between yourself and Alice, then basically the two of you are entering into a multi-signature arrangement that you can then update between the two of you privately and without broadcasting to the rest of the network, and you can update the values of basically how the value is balanced between yourself and your counter-party in that channel.

Then there's a lot of game theory involved around making it a very bad idea to try to cheat somebody and post a stale channel state that would give you back more money than you're supposed to have. That's where a lot of the time locking comes in. There's also various recovery transactions that basically allow you to steal the money away from someone if they try to steal it from you incorrectly. This works pretty well between two people, but it's not that interesting. This technology has existed since I think 2012, or so.

What gets interesting is then, when you start connecting all these payment channels together and you start routing money through them in a way that is more, I guess opaque to the end user, they can just say, “I want to make a payment to this address and I don't really care how it gets there, let's just let the software under the hood figure out how to route it.” Now since we're

doing this all off the blockchain and only transmitting the data between the few parties that are interested in it, its orders of magnitude more scalable. You basically become limited to network latency and the actual processing requirements are very low.

I think that some folks, like Christian Decker have said that you could update a payment channel hundreds of times per second. I think that's pretty much going to come down to whatever the network latency between the two parties in that channel is coming out to. From a more macro perspective, this is much more appealing for scaling a distributed decentralized network than just throwing more data on to the whole network itself.

The comparison that I often use is actually with the Internet itself of this is a pretty tried-and-true method of scaling distributed networks, where if you're looking at the way the internet is constructed with layer 0, the Ethernet layer, that is a global broadcast to all protocol where you put data on to the Ethernet and it goes to everyone else on that Ethernet network. Of course, that would never work. If we tried to scale it up to the entire world, we wouldn't be able to do things like we're doing right now with streaming audio,, because we would have to simultaneously send and receive everybody else's audio streams at the same time and obviously, our Ethernet cards would get maxed out and the fiber would get maxed out.

Instead, what they did to scale the internet was they added layers on top of it and specifically TCP-IP, the routing layers, make the internet much more scalable. What you're doing there is you're just finding the fewest number of entities on the internet that need to know about given piece of data in order to get it to the desired destination.

When you're sending data to someone, it doesn't have to go to everybody else on the internet, it just gets routed through a minimum number of parties. Of course, if any of those parties stop working or start trying to screw with you, then you just route around them to find some other ones, and that is really what gives the internet an extra level of robustness. It's really the same type of thing with Lightning, where instead of routing just data, though it is just data, it's now a special type of data that is representing value.

Now we have this financial network that is much like the internet itself and distributed, permissionless and pretty robust, though that's what a lot of the arguing that's going on right

now is about how robust will it really be. There's definitely still plenty of issues that need to be overcome, but I've seen a lot of different people that are devoting their own resources in order to improve this protocol. I think that we're going to keep working on it until it becomes good enough for actual production money use.

[0:32:11.9] JM: Basic internet infrastructure, things like e-mails and seeing web pages and loading videos. These problems did not get solved by liberal university professors and cypherpunks. They got solved by corporations, and so what we have today is you and I talking over a VoIP interface that was created by a company, talking on a browser that was probably, well in my case at least created by Google, running on pipes that are Comcast and other corporations. The worries of somebody like Roger Ver would be, okay if we take this delegated model of scaling where we move it off-chain away from the purest form of scalability which well, in perhaps Rogers view of purity, you threaten the decentralization of it.

You delegate this the scaling problem, such as how do payment channels discover and get connected to each other. That's a hard problem. We're delegating that to Blockstream, or Lightning Networks, or whoever else. I mean, or who knows? Maybe it's software people can run themselves, it's a peer-to-peer, that could be great. You got problems like people flicker on and offline, and I think there's some other trust issues. What's the argument that Lightning Network will lead to a degree of centralization that will threaten some of the core goals of Bitcoin?

[0:33:44.8] JL: Well, it's an interesting thing to hear in people talking about centralization of the Lightning Network. Actually, if you go back to my internet analogy, you can make a case that TCP/IP centralizes the internet, because now you're only routing through a small number of parties. I think the counterbalance to that is that if there are other parties available and you can route around the bad actors, then that's a important counterbalance to any “centralization.”

This centralization, and the fact that there are going to be economically larger nodes and economically smaller nodes and expect to see a power-law distribution that results in a scale-free network topology, it's mostly important because we're not talking about actual custodianship of the money. It’s not like you're – you might try to send your money through a node operated by

Coinbase and they're going to freeze it and steal your money. They could refuse to relay it along and then you have to find another route.

I don't think it's quite the same level of centralization problems that we would actually expect to see if we scale on-chain, because then scaling on-chain is going to drastically increase the cost of who can run the full nodes, which are actually determining the rules to the network. Now that seems to be a lot scarier, because now the individual no longer has control over what rules they're agreeing to. It seems like giving that “custodianship” of network rules to a small number of actors is a scarier proposition. Of course, this is once again just going back to ideology and what you think is the most important aspects to preserve.

[0:35:48.9] JM: I completely agree with you. I think, if I can reframe your argument back to you, the arguments that Lightning Network alludes to centralization is a portrayal of a world where banks stand up their own Lightning Network, maybe it's powered by Blockstream, or in partnership with Blockstream, or powered by lightning.networks, and we're all routing all of our payments through new banking infrastructure that's owned by the same people.

Maybe the main chain is still open to cypherpunks and whatnot, but if you wanted to transact only through the main chain, it's going to be a much worse experience. It's going to be much more expensive than if you were to transact through payment channels. That’s the centralization case. That's the scared bear case.

The bull case is that this is a – is that the world with no on-chain scaling with the small blocks, the small block world is that nodes remain easier to run. Not that they're easy to run today, but they're not going to get harder at least. In addition, it's not like – I mean, we don't necessarily know to what degree there will be centralization of the Lightning Network. For all we know, there will be Indy banks, like maybe me and you can go start a bank and be – or have a Lightning Network between us and we can charge whatever we want. Maybe we have to charge slightly more than Chase, if Chase has a Lightning Network-based banking system, but at least it's permissionless banking.

[0:37:28.2] JL: Right. That's actually my plan is that I intend to be operating nodes on the Lightning Network, and if other banks, if traditional financial institutions want to come in and

start routing money, then they're welcome to do that, but they can't stop me from participating, so I'm not worried about that.

[0:37:48.2] JM: How resource intensive do you think it'll be to run a Lightning Network?

[0:37:53.0] JL: I don't think it's going to be that intensive from a computational standpoint. I think the bigger question is going to come down to what level of economic risk are you willing to take, because running a routing Lightning node is going to be running a hot wallet. Hot wallets are a security issue you need to worry about. It's really, I think going to come down to the economics of how profitable can it be to route money through the Lightning Network versus the level of risk of operating this hot wallet.

That'll be another thing that people are going to learn over time. Of course, crypto security is a constantly moving target. We are getting better at operating more secure hot wallets, but there's still quite a ways to go.

[0:38:45.0] JM: I really want to take the devil's advocate argument too, as much of an extreme as we can, because I just – because I want to settle – I want to settle this as much as I can for the Twitter bots that are somehow transcribing this podcast and processing what I'm saying. If I'm a person in a developing country, or a poorly regulated financial system like Venezuela, why is a world with Lightning Network superior to one with big blocks?

[0:39:16.3] JL: It depends on the user experience and what they care about. Now, if you're living in Venezuela, or a country that has hyperinflation, your top priority is probably not being fully self-sovereign and in validating that no one is breaking the rules, because you're already living in a system where the authorities are breaking the rules so terribly that you have this huge incentive to put your value anywhere other than your nation-state currency.

For a lot of people that are in that situation, they may not care about the specific security model that they're being given. They just may care more about the store of value aspect. Of course, in any of these permissionless systems, they're pretty volatile, but they're at least, they're not guaranteed to go down in the way that a hyper inflating nation-state currency is.

[0:40:18.2] JM: Okay. The problems that you've outlined with Lightning Networks, there's a great article I read by you that I think was from like – it might have been from 2012, or maybe it was 2016. It was a little bit dated, but you outlined a lot of problems with Lightning Network. I don't know if they've been solved today, but there are things that we touched on earlier, like the fact that people can flicker on an offline, the fact that payment channels have this discovery process, this discovery problem, which is the problem of service discovery that a company has to face. What are the biggest unsolved practical engineering problems in Lightning Network? Or are you up to date enough to know some of those problems?

[0:41:05.0] JL: Yeah. There's actually another type of flickering problem that actually came up recently and just got fixed, where there is this fee negotiation process that happens when you're routing money. There was an issue recently, actually that resulted in a number of premature channel closures happening, because nodes on the network got out of sync and we're disagreeing with what the appropriate level of fees were.

Trying to negotiate fees with another node who was then asking for order of magnitude higher fees would actually result in you closing the channel on them, because it was suspicious and spammy. That's just an example of a lot of these tiny little details that are going to need to be worked out over the long run. I also in that article, I think that was 2 or so years ago, I was just noting that it'll – you'll need to worry about different types of denial of service attacks, where there could be an attacker who is very well-funded and tries to flood the network and basically unbalance a lot of the payment channels and force them to close.

This is a layer on top of Bitcoin, and so you can't create a layer that is more secure than the foundation of course. There is going to be new game theory, new types of attacks that need to be tested and figured out how to mitigate. That's why it's not going to be this overnight process of, “Oh, okay everybody, the Lightning Network is deployed and you should just go ahead and use it,” but rather it's going to be a long process of well, the Lightning Network is already out there and there are thousands of people that are using it, but it's still in a beta testing phase and we wouldn't recommend putting a large amount of money into it.

[0:42:58.3] JM: We've done a lot of shows on testing and deploying internet infrastructure in the context of a specific company. If I'm a SaaS company, I probably have a continuous delivery

pipeline, I've got these different environments, I've got a test environment, or staging environment and I have a pipeline. If I'm deploying new software, I first deploy it to my test environment. I run some tests on it, okay it's looking good, I click – move along the pipeline to the staging environment and then I have automated tests that run in the staging environment, maybe I've got some manual people who are doing some tests in the staging environment as well. If it works, then I push it to production and then we have my SaaS company has been updated and it's in production and that's great.

With Lightning Network, we're talking about deploying software that has a wider reach, perhaps is quite difficult to roll back if you roll it out. How does the deployment process of something like Lightning Network compare to the deployment process of conventional software?

[0:44:08.4] JL: Well, so you do start out pretty similarly. You'll write your code, you'll write your unit test, you'll have peer review on the code itself, which is hopefully very rigorous. Then there are both local test networks and a reg test networks that you can set up, and then there's the actual test net that most people are using, and hopefully any issues get caught while it's still in the testing phase.

This is one of the reasons why life cycle for changes to these crypto-asset networks tends to be a lot longer, because you want it to be baking in adversarial test environments for as long as possible, because like you said, once you get to the production ready phase, even then it's not – it's not that you just deploy it because you don't have the ability to deploy, it's that you make the software available to deploy and then you're going to see probably like a bell curve distribution of actual upgrades that happen, where there will be a few early adopter techie types that upgrade very quickly, and then over the next month or two you'll probably get most of your upgrades, and then there's going a very long tail of laggards.

Really, as soon as you've made it publicly available and have the first few people that are out there and upgrading, it becomes almost too late to rollback in many cases, where a revert is going to be pretty damaging to anybody who gets affected by it.

[0:45:55.3] JM: To what degree can I transact on a Lightning Network today?

[0:45:59.6] JL: You can go set up a Lightning wallet on test net or main net. There are at least a few dozen providers on main net that you can actually buy goods and services through. I have links to all of those on my website, and also link to some other Lightning resources.

[0:46:16.6] JM: How does that experience compare to transacting through Coinbase, or some other non-Lightning Network payment provider?

[0:46:28.6] JL: It tends to be a lot faster in terms of just the responsiveness, because there's no need to wait for confirmations or anything like that before the provider gives you whatever it is that you're purchasing. Though, because it is still pretty new and there are some kinks to be worked out, there are the occasional payment failures. Though, I haven't experienced them myself, but I have seen some other people have basically routing failures. It's not guaranteed to work at this point for sure, but it seems like the vast majority of time people are having a good user experience.

[0:47:05.9] JM: There are these different implementations lightning.networks, Blockstream, I'm sure there are several others. What are the subjective design decisions? What are they making trade-offs on?

[0:47:19.7] JL: Well, they're all trying to be compliant with each other. I think, the main real differences that I'm aware of are the actual languages that they're being built in. The C Lightning implementation, which is being written by some of the Blockstream folks tends – it seems like it's more the Bitcoin core style, like C++ development. Whereas, LND Lightning is actually more like the BTCD style, where they're written in Go, tends to be a bit more readable developer-friendly whatnot, but possibly not as performant. Then there's Éclair, which I'm actually not familiar with what that's built in, but I think those are the three main implementations that have the most developers working on them.

[0:48:10.9] JM: Once Lightning Network is deployed. Let's say it works, it's secure, what kinds of platforms, what kinds of cool things could be built on top of it?

[0:48:24.8] JL: This is actually one of the reasons that I'm excited about Lightning, more than just for having cheap fast payments, is that I actually expect to see a proliferation of laps, as

some people have called them various Lightning-enabled apps, to interact with each other. Because now, if you're if building something on the second layer networks, it requires a lot less permission for you to play around with them. It's not like you need to try to get consensus changes, or even policy changes at the network level.

I've been seeing from my former coworker Alex Bosworth actually, he seems to be a big Lightning idea man. He's always talking about new ideas of stuff that he's working on. I'm most interested in seeing decentralized exchange happening, and that's for a number of different reasons, but mainly to try to get rid of some of the systemic risk of the current centralized exchanges. Also, just from a privacy standpoint.

I think that anything that you're doing on the second layer networks is arguably going to have better privacy than doing out in the open, on-chain Bitcoin transactions. Now it's certainly debatable, but it does seem to be a network that's taking a much bigger focus on privacy.

[SPONSOR MESSAGE]

[0:50:07.9] JM: Citus Data can scale your PostgresSQL database horizontally. For many of you, your PostgresSQL database is the heart of your application. You chose PostgresSQL because you trust it. After all, PostgresSQL is battle-tested, trustworthy database software.

Are you spending more and more time dealing with scalability issues? Citus distributes your data and your queries across multiple nodes. Are your queries getting slow? Citus can parallelize your SQL queries across multiple nodes, dramatically speeding them up and giving you much lower latency. Are you worried about hitting the limits of single node PostgresSQL and not being able to grow your app, or having to spend your time on database infrastructure instead of creating new features for your application? Available as open source, as a database, as a service and as enterprise software, Citus makes it simple to shard PostgresSQL.

Go to citusdata.com/sedaily to learn more about how Citus transforms PostgresSQL into a distributed database. That's C-I-T-U-S-D-A-T-A.com/sedaily, citusdata.com/sedaily. Get back the time that you're spending on database operations. Companies like Algolia, Prosperworks and

Cisco are all using Citus, so they no longer have to worry about scaling their database. Try it yourself at citusdata.com/sedaily. That's citusdata.com/sedaily.

Thank you to Citus Data for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:51:53.3] JM: To improve your point, I think to strengthen your point, if I transact on the main chain entirely from a set of payment addresses that are associated with my wallet, you could maybe re-piece my transaction history and figure out what I'm up to. If you had a Lightning Network that acted as a dark pool, then it could pool together transactions and make larger transactions with the main chain to checkpoint to the main chain and everybody is shrouded in the additional privacy features of that Lightning Network.

[0:52:34.8] JL: Yeah. It's going to get interesting once we see more complex types of routing become more common, especially when it becomes easier to break up your payments and basically spread your payment out across a large swath of the Lightning Network, and then it gets pieced back together at the end.

[0:52:57.7] JM: Lightning Network, if we get fast payments implemented and they're working well and they're cheap, because they're on the democratized world of different Lightning Networks competing with one another, then we could see an explosion of programming languages on top of that, or we could have maybe an Ethereum-like language that is simply has payments as a first-class citizen within it just like Ethereum, except it's interoperating with the Bitcoin blockchain. How would that network, Lightning Network-enabled Bitcoin decentralized app platform, how would that compare to Ethereum itself as a platform?

[0:53:45.3] JL: I mean, there's a million different ways that this whole thing could go. There are some people that are already starting to try to think of what would layer three, or layer four type applications look like. There's also a fair amount of innovation going on with like scriptless scripts and better smart contracting languages, like simplicity for example, seems to be making some good headway there.

Also, better and more performant smart contracting on Bitcoin itself with merkelized abstract syntax trees and tap root and graph root; these things are all still probably a year plus away, but I'm actually fairly confident that over the long-term, we're going to see robust and yet, developer-friendly smart contracting languages become available for use in the Bitcoin ecosystem. It's obviously taken a lot longer for Bitcoin to get around to this as opposed to Ethereum, because the developer mindset is a lot more conservative, and I guess not as interested in the move fast break things type of development process.

It's definitely going to be interesting to see which of these, but ultimately is more popular, like the one that is very well-marketed and very developer friendly, or the one that is trying to be more secure and an edging towards developer friendliness. I don't think that we're going to have to worry about seeing that play out for probably at least another year.

[0:55:31.5] JM: At least another year. That's pretty fast as far as I'm concerned, right?

[0:55:36.4] JL: Yeah. I mean, a lot of this stuff really does come down to your personal time preference. I think that's also a big part of the scaling debate is that a lot of the people who want to increase the block size, they see it as a more urgent issue where if Bitcoin doesn't do this, then it's going to get overtaken by some other network that can scale better. Whereas, the more conservative folks are like, “We know Bitcoin is going to be fine regardless of how many people are unhappy with trying to transact through it.”

[0:56:08.7] JM: Yeah. We talked about the low-level development issues in Bitcoin. We're talking a little bit about Ethereum now. You had a great article about the challenges of building low-level Ethereum infrastructure. Tell me about those challenges. What were your learn – I think you just spend some time building on Ethereum, right? Because, I think you had to implement something for the company BitGo when you were working there. Tell me about the challenges of low-level Ethereum building.

[0:56:35.2] JL: Well, I mean, there were a number of surprises from the operational standpoint. It was a nightmare trying to run fully validating nodes. It turns out that most people in the Ethereum ecosystem are not fully validating the whole blockchain. They're actually using this warp sinking feature that is just getting the merkel route of the entire state of the system from a

recent block and starting to validate it from recent history, and assuming that most of the nodes are honest and the hash power is honest and whatnot.

That's a trick that Ethereum has gotten, put into most of the nodes just to make it usable. If you're trying to do full validation, there are certain types of hardware that it's literally impossible for it to ever catch up and get to the current tip of the blockchain, so you end up playing around with a lot of configuration parameters and different hardware types in order to get fast enough speed. It really mostly comes down to DiskIO, because of all the state change operations.

From a even lower-level perspective, just dealing with 256-bit integers and 256-bit math can be a real pain, if you're looking to do that like database operations of atomic addition, subtraction, whatever operations with 256-bit numbers, you're probably going to have a bad time. I'm not aware of any that support that natively. We ended up having to basically do the arithmetic ourselves in in-memory and single-thread those operations so that we weren't accidentally corrupting the data that was currently in the database.

Then at a higher level of just the smart contracts themselves, smart contract security is still very new, and it's a real nightmare if you're doing what we were doing, which is writing our own multi-signature implementation. Interestingly enough, anyone who's familiar with the parody multi-sig contract and the different issues that it had, the first issue that it actually had was a bug that an audit found in the BitGo smart contract. That actually was found and fixed over a year ago, almost a year before a BitGo even went live. Then we had been live for about six months and that parody hack happened, and we looked at it and we're like, “Oh, crap. This is such an easy thing to overlook.”

It was the way that our whole smart contract development process went, where we got an audit, we fix things and then we would go to a different firm, get a different audit and they would find different things. It was just a process of this programming language and the theory and virtual machine, it's all so new that almost nobody really understands how it works. The problem that I have with solidity for example is that, it is incredibly user-friendly, it's developer-friendly, easy to get up and running, but it's also deceptively friendly, where simply leaving out a keyword can result in all the money in your smart contract to getting stolen, or frozen, or what have you.

That's why there are a number of different folks who are working on better standardization and better tools, provable smart contract security, stuff like that. It's just, there's a lot left to be done before I think that this type of development is really suited for storing millions and billions of dollars. It's just really hard to be completely sure about the security aspects of it right now.

[1:00:34.0] JM: Many people would decry the tribalism within the cryptocurrency ecosystem. You wrote about the advantages of tribalism. What are the advantages of tribalism within cryptocurrencies?

[1:00:46.6] JL: Well, I think that it mostly comes down to network effect, is that you need some sort of culture that is keeping people interested, and keeping people on the same page as to what you want this project to value and to thus, evolve in a certain direction. That's one of the more fascinating sociological aspects of this whole ecosystem is if you don't have authoritarian type of governance, how do you attempt to govern the network, or at least attempt to corral people together to try to get them to collaborate, rather than just dispersing and fracturing into a million different networks that have probably far less value than if you collaborated and work together?

We see some very interesting results from that. Some of them are very toxic, and there's a lot of fallacious arguments going on. Then there's also lighter side of things with the memes and in people coming up with inside jokes to keep folks entertained. I try to participate in the more optimistic side of that without attacking too many people. Though sometimes, you see some really terrible decisions that are made by some of the developers on various networks, and it's hard to believe that they're able to get away with these things and still have networks that are valued at billions of dollars.

There's a lot of nonsensical things that are going on too that I think, maybe you could attribute to this system being overhyped in certain ways. It's certainly fascinating. One way or another, it keeps me intrigued and basically plugged in constantly.

[1:02:50.1] JM: That makes two of us. I wish we had more time today. As I said before the show, you're welcome to come back on the future. I really enjoy your sentiment in terms of talking and writing about cryptocurrencies. I do want to give you a couple of minutes to talk

about your company, Casa, that you're building. I'd love to know what you're doing and what the long-term vision of Casa is.

[1:03:12.9] JL: Sure. We're starting out with a Bitcoin of vault product and anticipate eventually expanding that to support a lot of different crypto-assets. Really what we're trying to do is bring the promise of being your own bank back into the crypto-asset world, where I think that it's always been possible to be your own bank, but the learning curve is so high and the amount of time you have to put into it, and the technical requirements. They're just so onerous that the vast majority of people don't do that.

Even people at my technical level that I've spoken to, a lot of them do not go through all of that rigmarole, because either they just don't want to put the time into it, or because they don't trust themselves. I think this results in a lot of people just keeping their money with third-party custodians, and that is antithetical to, I believe what the premise of this whole movement is.

What we're trying to do is bring the usability and ease of use of a mobile phone app and combine it with the security properties that you get with a hardware signing device. When you create a Casa wallet, or Casa vault, it's a three out of five multi-signature wallet, and you're going to have three different hardware devices that you geographically disperse. You'll also have your phone with the key and the secure element, and then Casa will have one key as a emergency recovery mechanism.

Just having these keys mostly on dedicated hardware devices that are not in the same area is going to protect you against tax. It's going to protect you against many different forms of loss, acts of nature, or even physical attackers that may come into your home, or business, or what have. We're really just trying to provide a comprehensive solution, where we think through as many of the possible loss vectors and build them into our security model where we are taking the best practices and the experience that we've built up over the years and putting them into the software to help guide the users to do the right thing.

That's what I think is going to be the very interesting experiment over the next year, so as to see how well that works, see how much of the best practices we can put into the software so that we can lower the level of touch that is required, because we're starting out as a very boutique

solution about a $10,000 a year price point. If we can lower that level of touch and make it more automated, then we're hoping we can bring the price point down and make it more widely available.

[1:06:03.1] JM: Well, as soon as you become the Wealthfront of cryptocurrencies, you can count me in.

[1:06:08.4] JL: All right.

[1:06:10.7] JM: Jameson, thanks for coming on the show. It's been really great talking to you.

[1:06:13.3] JL: My pleasure.

[END OF INTERVIEW]

[1:06:17.3] 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 plugins. Use the value stream map to visualize your end-to-end workflow. 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 long-time sponsor of Software Engineering Daily. We're proud to have ThoughtWorks and GoCD as sponsors of the show.

[END]