EPISODE 435

[INTRODUCTION]

[0:00:00.7] JM: Visa processes 1,600 transactions per second. PayPal processes 193,000 per second. Bitcoin processes only 3 to 4 transactions per second. In order to fulfill the dreams of financial programmers, in order to get decentralized peer-to-peer micro-payments, Bitcoin needs a much higher transaction throughput. Bitcoin's scalability issues have led to debates within the community and changes in the software.

In this episode, Jordan Clifford gives an overview of some of the scaling limitations of Bitcoin and discusses SegWit; a change to the Bitcoin protocol that improves scalability. Jordan was previously on the show to discuss the basics of Ethereum and Bitcion, and this show covers some more advanced topics of Bitcoin. If you're out of your comfort zone, don't worry, you aren't alone. I was confused for much of this episode. This stuff goes into the weeds of Bitcoin and scalability and we will certainly do more shows around this topic to get you acquainted as well as me.

We've covered the basics of cryptocurrencies in detail and lots of other episodes including one previous one with Jordan. We've tackled more complex aspects of them in other past episodes. If you want to find all of these episodes and many more, you can download the Software Engineering Daily app for iOS and android and find all of our old episodes. They're organized by category, and as you listen, the SE Daily app gets smarter and recommends you content based on the episodes that you're hearing. If you don't like this episode, you can easily find something more interesting by using the recommendation system.

The mobile apps are open-sourced at github.com/softwareengineeringdaily, and if you're looking for an open-source project to hack on, we would love to get your help. We are building a new way to consume software engineering content. We have the android app, the iOS app, a recommendation system and a web frontend and more projects are coming soon. If you have ideas for how software engineering media content should be consumed or if you're interested in contributing code, check out github.com/softwareengineeringdaily. You could join our Slack channel. There's a link on our website, and you can send me an email at any time, jeff@softwareengineeringdaily.com. I would love to hear from you.

With that, let's get on with this episode.

[SPONSOR MESSAGE]

[0:02:30.6] JM: Indeed Prime flips the typical model of job search and makes it easy to apply to multiple jobs and get multiple offers. Indeed Prime simplifies your job search and helps you land that ideal software engineering position from companies like Facebook, or Uber or Dropbox. Candidates get immediate exposure to top companies with just one simple application to Indeed Prime, and the companies no Prime's exclusive platform message the candidates with salary and equity upfront.

Indeed Prime is a 100% free for candidates. There are no strings attached. Sig up now and help support Software Engineering Daily by going to indeed.com/sedaily. That's indeed.com/sedaily if you're looking for a job and want a simpler job search experience. You can also put money in your pocket by referring your friends and colleagues. Refer software engineering to the platform and get $200 when they get contacted by a company and $2,000 when they accept a job through Prime. You can learn more about this at indeed.com/prime/referral. That's indeed.com/prime/referral for the Indeed referral program.

Thanks to Indeed Prime for being a new sponsor of Software Engineering Daily. If I ever leave the podcasting world and need to find a job once again, Indeed Prime will be my first stop.

[INTERVIEW]

[0:04:10.6] JM: Jordan Clifford is a Bitcoin enthusiast. Jordan, welcome to Software Engineering Daily.

[0:04:16.1] JC: Great. Thanks for having me.

[0:04:17.5] JM: We had you on a while ago to discuss some of the basics of Bitcoin and Ethereum, and in this episode I want to discuss some of the more advanced topics particularly the scalability issues that triggered a debate over the last couple of months and a forking of Bitcoin. I think the place to start to motivate this discussion, I think we should assume that everyone listening has a pretty decent understanding of Bitcoin. If they don't, there's plenty of older episodes they can go to to find out the basics.

Bitcoin has some scalability issues or it had some scalability issues that were under the spotlight a couple of months ago. Describe those scalability issues that have been debated over the last few years that really came to a head a couple of months ago.

[0:05:07.0] JC: Sure. I think this kind of debate really kicked off maybe three or four years ago when Gavin recognized that transactions over the network were growing at an exponential rate, yet the capacity for transaction over the network has basically remained flat. The Bitcoin network has a blocksize limit on each block that's produced, and blocks are produced every 10 minutes. Currently, the blocksize limit is 1mb, although SegWit effectively raises it to approximately 2mb on average, but that 1mb blocksize limit in the native block restricts the transactions per second to about 3 to 7 depending on the exact format of the transactions within any given plot.

3 to 7 transactions per second is miniscule. Visa, on a peak transaction velocity, has about 40,000. If we're on the Bitcoin network and we can handle 3 to 7, that's really just a tiny, tiny fraction of the active commerce.

What's been happening is we've been having this clash of visions about how to scale Bitcoin. Satoshi Nakamoto was a bit blocker. I don't think there's anybody that can refute that claim. Satoshi believed that the blocksize limit which should be allowed to go larger as a hardware improves and as software improves. He actually foresaw the specialization of the network.

Now, when Satoshi gave a control to Gavin, Gavin was following in the same footsteps of that vision, but, later, Gavin decided that he wanted to step back from the active maintenance of the GitHub repo and play more of a researcher type of role. He'd became chief scientist for the Bitcoin Foundation and stepped away from the daily maintenance duties of the Bitcoin software, and when that happened — and Wladimir van der La an took over, we began to see kind of a shift in the vision and the core developers became much more of a decentralized maximalist. They rejected the specialization of the network. The core supporters and the core developers today really would rather have Bitcoin be able to run on commodity hardware from years ago. They do not want to see any sort of a situation where Bitcion requires specialized hardware or requires maybe connectivity to a data center. They would like to see Bitcoin to be able to run on some modern PC or a laptop with a modest internet connection.

The core developers have really latched on to this idea that we can do all of the transaction scaling off of the blockchain itself, and instead they'd like to use the blockchain as a settlement layer for these systems. You'll hear about things like Schnorr signature aggregation, which will increase the transaction capacity by combining signatures across many transactions into one signature. You'll hear things about a lightning network where they'll actually do lighting — Let me just explain a lighting as briefly. Lighting is this idea that not every transaction needs to go on to the blockchain. We can actually aggregate transactions and batch them into the blockchain. Without getting too far into the details right away, everybody connects to the lightning network through one or more just kind of connections to the network, and then the lightning protocol has methods for routing payments between each other across these hops, so you can have like three hops to Sally. We go through Bob and Charlie to get to Sally, and you can pay Sally without actually having a direct connection with her and without actually doing a transaction on the Bitcoin blockchain. You can actually just update the balances kind of off chain, and that only when you actually need to cash out of Bitcoin completely, that you do an off chain transaction to get out of the network.

We really have just a clash of visions happening right now, and it's gotten so political. To be honest, the science is really kind of rudimentary and still not there. We've gotten a lot of just kind of ad hominem attacks and it's really been quite nasty.

[0:09:21.1] JM: To give people a picture for the teams here on either side of this debate, the lighting network side of things, if I understand this situation correctly, might be characterized as people who are more pro-centralization, because a lightning network is an abstraction that's on top of the core Bitcoin layer and you have smaller chains of trust that go in that lightning network so you can process a higher volume transactions with that circle of trust and then you use, like you said, the underlying Bitcoin layer is a settlement layer, more of the core data layer, and so you have almost these caching — You could describe it as a caching layer of transactions. It's not a great way of appropriating it.

I really want to help motivate the ideological difference here, because my understanding is Bitcoin started, everybody is like, "Okay. This is a cool, crazy thing. Let's start messing around with it," and then it started to get some traction, and then as it started to gain traction, there were different players in the space. There were people who were transacting a lot. There are people who are mining a lot, and they built up capital assets that were based on their view of where Bitcoin was going, and because they built up those capital assets, for example, if I'm a miner, I've purchased all these mining hardware, and that's going to bias my thinking towards where Bitcoin should go, because I want Bitcoin to go in a direction where my capital investment, that hardware, is going to continue to generate a good amount of capital to warrant the investment that I put into it. There are other people who maybe have not put as much money into hardware, and they have a different opinion for where Bitcoin should go.

I think what's interesting is a lot of times, well, at least the way I see it, either side frames their argument in terms of, "Oh! What is good for the community? What is good for the ideology?" but it's curious that their ideologies and their beliefs happen to coincide with the capital investments that they have in the system. Am I portraying things correctly?

[0:11:37.5] JC: I think so. The miners, they presumably have the most skin in the game. To mine today you need to have a specialized hardware and you need to have very fast internet and you probably need to have cheap electricity, but in any case, it's a significant upfront investment. The miners have a lot of skin in the game and they really want to see Bitcoin succeed. The problem is that the miners can be seen to be biased towards on chain fees. The miners largely want to see an appreciation of the Bitcoin price, but they also want to see an appreciation in the total Bitcoin fees that they get to charge per block, because that's their revenue.

Miners have been seen as being against kind of improvements to the Bitcoin, because they don't want to see fees go off chain. This is kind of one of the propaganda things that you'll hear from decentralists, or the core developers. They say that the miners are really not playing nice, because they are only looking after their own selfish interest.

I think you're pointing out correctly that everybody is looking out for their own selfish interests. Again, I think you're pointing out correctly that everybody is looking out for their own selfish interests and it just happens to be that the miners want to see an increase in Bitcoin price and they want to see more transaction fees per block. That's largely why you'll see miners like Jihan Wu be advocating for a larger block size. He's one of the strongest advocates for a larger block size and he's the largest miner.

Whereas the core developers, they really see this as consensus software and they talk up as if they're engineering a rocket ship. In some ways that's maybe true. This is consensus critical software. You really want to take a very careful methodical approach, but the degree of conservatism maybe is a bit extreme. Core developers have not budged an inch for years now on the block size debate. Now we're seeing businesses and miners talk to each other outside of the purview of the core developers, and they came up with the New York agreement back in May. In May, Barry Silbert of the Digital Currency Group got together as portfolio companies. He got together the largest miners representing over 80% of the hash rate, and they all signed this letter that says, "Okay. Enough is enough. Let's lay down our arms and let's come up with a — Compromise it." It doesn't make both sides necessarily happy, but it doesn't piss off both sides either.

What we're going to do is we're going to do SegWit, which core wants. The core developers have all of these improvements that are kind of waiting on SegWit, but we're also going to do a 2 megabyte hard fork. What's happened is they created that in May, and in August 1st, so the UASF people, which I hadn't got to yet, but the UASF people really — This is kind of the extremist contingent among the core supporters. I don't know if you're seen it, but they're in these hats on social media, the UASF. The UASF if a User Activated Soft Fork.

Luke Junior decided to basically run this political campaign to get core develop supporters to run UASF software, which basically means they're going to reject any block that doesn't have SegWit after August 1st. This was kind of a ticking time bomb that would fork the network, unless the network adopted SegWit.

The New York agreement basically already had anticipated adopting SegWit, and the hash rate adopted SegWit ahead of the UASF in order to avoid a split. Now, we did see a split on August 1st, but that was because of Bitcoin Cash, which explicitly was trying to create an OPT coin. It's kind of a little bit unclear whether or not that violated the spirit of the New York agreement. My contention would be that it does not and that Bitcoin Cash was explicitly created as an OPT coin. If you're doing something outside of Bitcoin, I don't think that affects whether or not you've violated a Bitcoin agreement.

In any case, we got SegWit adopted on August 1st, and now we have this kind of three months until November when we'll see the two megabyte hard fork potentially. 90% of the hash rate approximately and many of the largest companies in Bitcion are all signaling that they're ready for two megabyte blocks. However, the core developers are digging in their heels and rejecting this premise. They think that this is way too premature, way too rushed. Also, it just doesn't fit with their ideology or the vision that they have, which is to make Bitcoin easy to run anyway, even in third-world countries, even with poor internet connectivity, even if the only connections to the blockchain you have is through Blockstream's new satellite program. They want you to be able to run a node. We still have this clash of visions, and as you point out correctly, I think, the clash of incentives.

[SPONSOR MESSAGE]

[INTERVIEW CONTINUED]

[0:17:39.1] JM: How do changes to Bitcoin functionality gets proposed and passed and to what degree is that governance process formalized?

[0:17:49.4] JC: Great. Changes to Bitcoin happen through what's known as the Bitcoin improvement process. This is a process that was created by Amir Taaki for the Bitcoin Core software project. What happens is a change will get proposed to the mailing list and hopefully the change is specked out and has kind of a description of what the problem it's trying to solve, and then it gets assigned a Bit number. Then once it has the bit number, it can go through iterations of the proposals and then eventually get coded up and then eventually it's adopted into Bitcoin software.

This is kind of a very loose structure. It's just a bunch of files on the internet that get tracked among the Bitcoin developers. How difficult it is to get it changed can vary extremely. If it's a consensus little change, that means it has to be adopted by the entire network, and if it's a hard fork, it has to be adopted by the entire network at once versus the software, which can be adopted by different contingence overtime at different rates. This is one of the reasons that core developers really prefer soft forks, because it doesn't require the entire network to synchronize up with the timeline for adoption.

That's kind of the overall process. Now, how difficult it is to get things through depends on the nature of a change. If it's not a consensus little change, you really are free to just write it up and use it however you want. There's no requirement for coordination across the entire network. If it is a consensus little change, we've seen that these things can drag out basically forever and many people have the fear that if we don't do the block size increase now or soon, there's a very good chance that we never will be able to simply because of the political factors and the characters that are involved.

[0:19:44.5] JM: Okay. There's a hard fork where the new blocks that are created are not compatible with the older software, and then there's a soft fork where the new blocks that are created are recognized as valid by old software. Is that right?

[0:20:00.1] JC: That's exactly right. A hard fork relaxes rules and it makes previously invalid blocks now valid. For example, a two megabyte hard fork, that's a hard fork because now we can have bigger blocks. The older software would flat out reject the bigger blocks.

A soft fork changes things in such a way that the older software will accept it as valid even if it doesn't totally understand it. This is SegWit. SegWit is a soft fork, because it removes the signatures from the transactions and puts them in a parallel data structure outside of the purview of the existing block size limit. This is kind of a neat accounting trick that can be used to increase the block size limit without actually increasing it.

It doesn't sell in a way that old notes, they see these new transactions as "anyone can spend". They actually are not able to validate signatures of the new transactions so that they cannot validate them, but they still see them as valid. If that makes any sense.

[0:21:01.6] JM: When a hard fork occurs, or when a hard fork is proposed, I guess, you signal with your node whether you are going to update to that software. Talk a little bit more about it, because I think a hard — It's like you don't have to update. You have a choice. You vote with your node, I think. Maybe you could just talk a little bit more about how that works.

[0:21:28.9] JC: Yeah, this is really interesting, and we're kind of going into new territory here. We've never really had a major hard fork. They only kind of hard forks we've had are kind of quick fixes, or there was an accidental hard fork I believe a number of years ago, but this community was much smaller and coordination was much easier back then. Bitcion has grown to be just a fairly massive size. Over $70 billion in market cap with thousands, if not millions of players involved.

Coordination is getting harder and harder and harder. The hard fork, as it stands, we do have an upcoming hard fork attempt, which is the New York agreement two megabyte hard fork, and what we've seen so far is signaling via miners. Miners are signaling, in the Coinbase strength, their intention to hard fork the network in November.

Now, that signaling is really more of a social signaling than a software signaling. The BTC One project led by Jeff Garzik, it does code up new, what they call version bits for the hard fork. I think it's bit one or bit four. I've got them all mixed up at this point. There is a signaling via bit four for a hard fork to two megs that is in the Bitcoin 1 software, but the Bitcoin 1 software is still a very small percentage of the network. Bitcoin Core is still the dominant software on the network.

We have these coordination mechanisms via signaling, via miners, but the problem is that not everybody sees that as valid. If you're rending a core software client, your client doesn't understand that new signaling mechanism for the hard fork at all. It's really not a very clean process, and it's very, very contested at this point. I wish I had a better clarity into what exactly will happen, but right now it looks like we are kind of in for a messy November.

[0:23:29.5] JM: When you say that when a hard fork can occur and I can signal, and you say it's a social signal, what does that mean? What is it mean to signal? Because I actually see this on — There's some website that I went to one time where — Because I was trying to understand SegWit and this stuff and it was like — It showed the percentage of nodes that were signaling for SegWit or signaling for some other kind of update. I don't remember exactly what it was, but tell me what it means to signal with your node.

[0:24:09.7] JC: Sure. Signaling means that my node is ready for a feature upgrade. If I'm signaling for the New York agreement, I'm basically saying I'm willing to accept two megabyte blocks. If I'm signaling for SegWit it means I'm ready to understand and process SegWit transactions.

Now, the thing about signaling is you can have half the network signaling for something, but that's not enough to really update in a confident way. If half the network is signaling for something and it's a controversial update, like a two megabyte hard fork, basically Bitcoin kind of revers to the status quo by default.

This is a dynamic that it's going to be studied for years on end I'm sure, but signaling just means my node is ready to do this new thing. Now, when is there kind of a critical mass of consensus? This is a totally political question and it's really the one that's being hotly debated right now.

Some people will argue that the economic majority of the major companies within Bitcoin plus the miners, that's overwhelming consensus, whereas others say they don't have any vote in the consensus process at all. The consensus happens among the community of users and developers.

Those are kind of the four contingence of Bitcoin in the community. We have the miners, we have the companies, we have the developers and we have the users. To do a successful hard fork, you really want to have all four of those groups totally bought in. When I say totally bought in, I mean you really want nearly 100%. That would be ideal. Whether we actually get there, it's really unclear.

We're kind of behind this debate, debate, debate, attack, attack, attack, toxic community, toxic community, but this is going to end at some point and we're watching it unfold right now, and it's really quite interesting to see decentralized consensus happen or not. Consensus failure may be what happens.

[0:26:15.8] JM: It's certainly interesting to see this happen on a human level, because this distributed system's emphasis on consensus I first learned about how important of a concept this was in computer science just when you're dealing, for example, a distributed database where random stuff can happen. This is like on a human level, and it's like the distributed general's problem. It's like at the level of humans now.

[0:26:46.6] JC: It is. It's totally at the level of humans. It's political. There're groups that all want different things. Everybody thinks they have what's best for Bitcoin in mind, whether it's Roger Ver who's advocating for bigger blocks, because he thinks it's just totally crazy that we wouldn't make more space for more users on chain, or whether it's Gregory Maxwell who's debating for smaller blocks and things that are totally crazy to view on an actually strip kind of a home user's ability to run at full Bitcoin node software implementation.

We have different people with different perspectives that all think they have what's right for Bitcoin and they call think that the other side is totally trying to destroy Bitcoin. When you take a step back, it's almost comical. Last year, I myself was very impassioned arguing for larger blocks, and in that arguing I've kind of learned some of the better arguments for smaller blocks and I kind of been able to emotionally distance myself from the outcome just because I was getting so upset that the community wasn't really agreeing with me, at least not completely. One contingent of the community agreed with me proliferously, or passionately, but the other side just said, "Hey, you're an idiot. You want to destroy Bitcoin," and I was like, "No. I love Bitcoin. I've been in Bitcoin since 2013. I hold a huge chunk of my net worth in Bitcoin. I definitely don't hate Bitcoin and want to destroy it."

I kind of had to peel back the onion of what are these people talking about when they say I hate Bitcoin or I want to destroy Bitcoin. I just learned that they had a completely different set of principles they were going off of. That's true on both sides and I've kind of gotten to a place of relative Zen where I can't care too much about the outcome, because it just takes too much of a toll on my mental and emotional health. I've been able to kind of synthesize both arguments from both sides and realize that there's very strong arguments on each side, and whatever happens is just going to have to happen. It's really too much to try to care about deeply anymore.

[0:28:49.6] JM: I would love to hear a little bit more about the conclusion that you've come to. It sounds like you've hinted at various arguments of either side, but maybe you could condense those arguments into what your current editorial stance is on the different arguments for and against larger block size.

[0:29:10.9] JC: Sure. Instinctually, I love larger blocks. I think Bitcoin is social software and it shows this fundamentally that money is language and language is what we use to create transactions and to get those transaction confirmed. Bitcoin to me is a value transfer network, and in that value transfer network emerges these useful tokens that could contain the value that can be used on the network. I instinctually really would love to see Bitcoin just become huge. I want to see everybody doing their transactions on Bitcoin, and in that process some people will lose the ability to run nodes, but that's okay in my view because there are still ways to confirm that the moneys coming in is valid, and more space to send transactions means more economic activity. It means more utility and value to the world. That's kind of the basic big blocker position, and that's the one that makes sense to me intuitively speaking primarily from an economist point of view, whereas I want this value transfer network to just become huge and widely used. In that process, we're going to generate a lot of wealth and a lot of utility. That's kind of where I was coming from last year.

I went and I argued with a lot of the Bitcoin Core supporters and developers themselves, and I told them, "Hey, why are you guys not allowing more people to use Bitcoin? If you really believe that the only way to use Bitcoin is to control your private keys and if you really believe that Bitcoin is going to be the thing that allows us all to escape government money systems and transact freely of our freewill, why are you not making more space for more users? You can only onboard so many users a day with the current block size. To onboard a user on to the system, you have to give them their own coins in the form of an unspent transaction output. We need more room for these. If we want to have more users, we need more room. Let's increase the block size. This is crazy that it's still so small. One megabyte doesn't feel like a lot of data in 2017. It feels like an extremely tiny amount of data."

Now, their point is, well, constraining the block size limit allows the network to be a lot more nimble. Constraining the block size limit means that if the government agencies come in and like shut down a number of nodes, more nodes can popup elsewhere relatively quickly. This nimbleness is something you lose if you have a large block size. If you'd let the block size limit go to 32 megs or even more extremely, gigabytes, every 10 minutes, well now you do have to have a data center with full of hard drives just to keep up. When you have that kind of a dynamic where you need huge amounts of hardware, very fast internet just to stay on the network, that really does limit the number of people who can run these nodes and that really does kind of create a bit more of a target for regulators, state actors, attackers in general.

We are winning something by keeping the block size low. The network can remain resilient and nimble, and there are improvements that we can get to scale without increasing the block size. I've kind of come to a point where I see arguments on both sides. They both make sense to me, so I become more of, I guess, centrist, if that's even such a thing. This has become such a bitterly debated, hotly contested, mudslinging affair that it doesn't really seem like it's clear that there's room for a centrist, but I kind of feel like I am a centrist at this point, whereas I want to see modest block size growth overtime just because I do think that it's so important to give the network room breathe, to allow more transactions to happen and to keep these from getting cost-prohibitive for using on a day-to-day basis, or even a month-to-month basis. Fees, I think at this point, are over $5, which that seems kind of crazy. I think they should be a bit lower than that, but they are what they are.

[SPONSOR MESSAGE]

[INTERVIEW CONTINUED]

[0:34:58.0] JM: Let me see if I understand your position correctly. The argument in favor or larger block size that favors computers that are more powerful, because they can process larger blocks. The extreme example of this is you need a quantum computer in order to transact on chain, on the fundamental lowest level of Bitcoin. The argument in favor of smaller blocks would — And the extreme example, you can just use your smartphone to transact with Bitcoin directly on chain. You can process these smaller transactions. Is that a reasonable explanation?

[0:35:46.5] JC: Yeah. Large blockers, they're okay with some specialization of the network. They're okay if it takes modern hardware to keep up-to-date. I don't think quantum computers would be necessary, but certainly you would need a wide array of storage and a decent internet connection to keep up with a larger block size.

Now, the small blockers would object and they would say, “Hey, if you require that, you’re really putting small miners and small node operators at a huge disadvantage and you’re maybe eliminating them completely.” They really want to see the barrier to entry for mining and for running a node be as low as possible and they say this because they want to see maximal decentralization.

Now, maximal decentralization I don’t think is really the name of the game, but I can understand people who do see that as a way that Bitcoin network remain sensorship resistant and the way it really remains outside the purview of anyone’s control. If you start to centralize the network, you are consolidating control and power and that is a real risk. I’ve basically come around to understanding the small blocker’s fears and sympathetic to them.

[0:37:00.9] JM: Where we stand today, there was the Bitcoin Cash fork. Explain to me what the Bitcoin Cash instantiation represents.

[0:37:13.0] JC: Sure. The Bitcoin Cash has really an interesting birth story. Jihan Wu, as a contingency plan, in response to the user activated soft work splitting the chain. Jihan Wu proposed a user activated hard fork, which is kind of a joke since it’s really not a user activated hard fork. It’s a minor hard fork, but that goes for both ways.

In any case, Jihan Wu proposed this as a contingency plan and then Haipo Yang from ViaBTC latched on to this idea. He said, “This is great. Let’s do Bitcoin Cash. This whole Bitcion block size debate has just gone on too long. It’s too bitter. It’s too toxic. We’re going to actually just fork the chain and explicitly create an alt.”

They actually created Bitcion cash as an alt, and in doing so they didn’t just forked Bitcoin, because that may not have survived. They forked Bitcoin, but they added an emergency difficulty adjustment clause, which basically says if blocks haven’t been found quickly enough over the course of 12 hours, we’re going to reduce the difficulty at a fairly rapid rate, 20% I believe. This can happen in succession.

This provision within Bitcoin Cash means Bitcoin Cash can survive even as an extremely minority fork, even if they don't have very much of the hash rate at all. This kind of represents a departure of people who really want to pursue the big block vision. People who maybe don't have sympathy for the small block side and they really just want to see cheaper fees for on chain transactions. They want to see more space for users. Kind of the natural position that I was talking about earlier with the big block sentiment. They really just want to pursue that full force without compromise with the small blockers.

They created Bitcoin Cash. They have this guy Amari create the Bitcoin ABC Client, which is the reference client, kind of the equivalence of Bitcoin Core. Bitcoin ABC is the core of cash. Now, we have basically just two versions of Bitcoin. The Bitcoin Cash supporters believe that if Bitcoin Cash gets the most of hash rate and the biggest market cap, it will become Bitcoin.

I have a slight hesitancy to say that. I don't like names changing out from underneath me. Kind of one of the hardest things about computer science is naming, and you don't want names to be shifting between different things. That just creates a lot of confusion. In any case, Bitcoin Cash believers really believe that their Bitcoin is the true Bitcoin, and now we get to kind of see both visions play out in parallel. It's really kind of a neat experiment.

[0:40:00.4] JM: Why doesn't that solve this debate, because it sounds like there are still a lot of acrimony and toxicity? When I look at Bitcoin Cash, I'm like, "Here, the miners got their currency." Why doesn't that solve things?

[0:40:18.0] JC: Bitcoin Cash, in my mind at least, they explicitly are not Bitcoin. I'm a Bitcoiner. I'm not a Bitcoin casher. I like Bitcoin Cash. I'm still holding on to my Bitcoin Cash, but all of my attention, my energy, my focus is still on Bitcoin, and Bitcoin still needs to grow, in my opinion at least. Maybe some other people are fine, just kind of taking their ball and going over the Bitcoin Cash court, but I still think that there's so much network effect. There's so much inertia within Bitcoin.

One of the things I'm personally most excited about are these new investment vehicle, rappers and derivatives that are being built within legacy capital markets and they're all being built on top of Bitcoin, not Bitcoin Cash.

Bitcoin still has the biggest brand, the biggest market cap, the most integrations with the legacy financial markets, kind of the most awareness among the public. It's still very important to get Bitcoin's future correct, at least in my opinion.

[0:41:22.8] JM: Okay. There's a term that I want to discuss a little bit, and that is segregated witness. I think we've hinted at it, but I want people to have a bit of a better vocabulary for understanding SegWit, because it's a term that's come up a lot. What is SegWit?

[0:41:43.5] JC: Sure. SegWit, I think you may have already said, is segregated witness. Now, what does that mean? That sounds like mumbo jumbo to most people, right? Segregated witness is very simple in reality. First; witness. A witness is just a signature. A transaction within Bitcoin can be thought of as kind of two distinct parts. One is it's an update to the database in terms of where the Bitcoins are. This is kind a state transition of Bitcoin. You go from one unspent transaction output to another, and kind of that movement from one to another is actually the movement of money.

The movement is money is one part of the transaction, and that's kind of the fundamental semantics of a transaction. Now, you can actually allow money to move unless the sender has authorized the movement of the money. The authorization of the movement of funds is what's known as the witness, or the signature.

Segregated witness, it actually is relatively simple. It means we take the transaction format, which includes the state transition and the signature and we simply take the signature out of the transaction itself. We remove all of the signatures from the transactions. We remove those signatures and we aggregate them into a parallel kind of Merkle tree structure, and that parallel structure lives side by side with the existing block. We have a new data structure and it's a Merkle tree, so we take the Merkle route of the new data structure, which is all the signature data and we take that Merkle route and we put it in the Coinbase transaction.

What we've actually done is we've taken the signature data out of the transactions, move it alongside into a new data structure, and we've taken that Merkle route of that new data structure and we put it into a Coinbase transactions. We've actually just changed kind of the block format in such a way that signature data is no longer part of transactions.

Now, this has a very neat side effect. When we remove the signatures from the transactions, we now can rely on the transaction ID to be stable and consistent. If we ever have movement of funds from one address to another, from one unspent transactions output to another, as long as that movement of funds is the same, the transaction ID is the same.

No longer can we be a victim of what's known as third-party transaction malleability. That means no longer can somebody else see our transaction, change the signature data slightly, which they actually don't even need the private keys to do, which it's kind of a really bad bug. We actually fix this problem. No longer can somebody change the transaction IDs out from underneath us.

Now, this has the very nice property of allowing lightning network to be a lot more simple, and also just allowing [inaudible 0:44:49.4] software to be a lot more reliable. If transaction IDs no longer can change, a lot of things become a lot simpler, especially as it relates to transaction chaining. You can actually have a chain of transactions that goes many layers deep. If the transaction IDs are malleable or can be changed out from underneath you, you have to wait for those transactions to confirm before you can trust kind of the end of that transaction chain, not so once you've fixed the malleability problem.

[0:45:21.7] JM: It's important to note, SegWit is what allows for these lightning networks. Tell me if I'm wrong, but lightning networks are type of side chain. Side chain is what some — I guess the small blockers would call a form of centralization, because you get these transactions that get centralized into these lightning networks.

[0:45:48.3] JC: I don't think that's quite accurate.

[0:45:50.3] JM: Oh, okay.

[0:45:50.7] JC: I don't think a lightning network would be considered as a side chain. Lightning network is kind of an overlay on top of the Bitcoin network.

[0:45:56.9] JM: Oh, okay.

[0:45:57.7] JC: As I mentioned before, a caching layer. I think that's exactly the right way to think about it. Whether lightning results in more centralization or not is really not understandable until we see the network typology. If a lightning network can be implemented with a bit pay or a Coinbase at the center, and if that were the case, I think we would all agree that's fairly centralized.

That's not the end vision, even if it starts out that way. The lightning network that the core developers are really interested in is more of a peer-to-peer kind of — Just a peer-to-peer algorithm that really — It looks almost like a full tree, but not really. They want to see multi hub. Each person connects to five random other individuals, not necessarily huge hubs. You can kind of use — What is it? The six degrees of bacon kind of approach whereas if everybody connects to five people and they connect to five each and they connect to five each, pretty soon I'll have a connection to almost anybody. That's the ideal. That's kind of what they're going after.

[0:47:01.7] JM: Is it more like compression? It's like compressing transactions, and then you get to process that compressed series of transactions.

[0:47:10.0] JC: Exactly. It's a lot more like batching, compressing and batching. Basically, we can all aggregate transactions and without writing into the chain and it can update balances many times per second, even billions of times per second. Only once we're ready to get out of the system do we have to actually settle up and close our lightning channels on chain.

[0:47:35.6] JM: Okay. Lightning network sound unambiguously good. Is this something that people are debating?

[0:47:41.9] JC: They are an unambiguously good, but they also don't exist yet. The core complain with them is that they've been promised for years now, and we still don't have them. You mentioned that SegWit enables lightning network. That's almost true. Lightning networks are possible without SegWit, but SegWit with the malleability fix really makes coding them up a lot simpler. It's kind of like a soft prerequisite. It's technically possible to do lightning without a malleability fix, but a lot of the monitoring of the lightning channels just becomes a lot more difficult and it's just not been done and it's kind of — Yeah, been waiting for SegWit.

Another feature of SegWit is what's known as script versioning. Right now, we have the Bitcoin script, which started off as relatively expressive and flexible and overtime we've kind of deprecated and removed different OPT codes for security reasons. Now, adding OPT codes basically requires coordination amongst almost the entire network.

With script versioning, this becomes a bit easier where we can actually use a version number for a new set of script. This allows us to add more kinds of features more easily and more in parallel. The script versioning is really a neat feature that SegWit brings us.

[0:49:08.4] JM: Yeah. Side chains though unrelated to lightning networks? Side chains, where only a group of people are verifying a set of transactions and then, I guess, the rest of the people on the chain verify their verification, or something like that.

[0:49:30.6] JC: Yes. Side chains also are still kind of in the works, but the basic idea of a side chain is it's kind of creating a sister network to Bitcoin that uses Bitcoin tokens. What you do is you lockup your Bitcoin on the Bitcoin network, and when you lock that up you actually can release new tokens on the side chain network, and the side chain network can have any number of different features. For example, MimbleWimble could be implemented as a side chain and you would be able to lock up your Bitcoin, get MimbleWimble coin, use MimbleWimble coin and then come back to Bitcoin later on.

This is basically contingent on what's known as a two-way peg. This two-way peg doesn't yet exist, but there's high hopes that side chains will allow for more experimentation and allow Bitcoin to really try out a bunch of new features without risking the main Bitcoin kind of blockchain.

[0:50:31.8] JM: Again, the fear here is that if you get really high performance out of these side chains, then more and more traffic will go towards them and then you have the problematic situation where you've created a centralized system where it could be taken over by a government or some other powerful actor.

[0:50:56.4] JC: Yup, that is precisely one of the fears. I think the main reason we don't have side chains yet is kind of the incentives problem. You don't want to create a side chain where all the transactions are happening and all the fees are happening, but in order to participate in that side chain you need, again, a big data center, a big internet connection.

The core developers are very, very concerned giving advantages to larger miners at the expense of smaller miners. Until we have a side chain model where everybody is able to participate with low barrier to entry, it seems unlikely that side chains will become a reality.

[0:51:36.1] JM: I think there are some different aspects of scalability to Bitcoin that I'd like to just dive into as we draw to a close. I know we're nearing the end of our time. We haven't really gone at a granular level at some of the different kinds of scalability. There's throughput, there's latency, and there's finality time, which is how many transactions per second can go through the system, the speed with which a transaction could be mined into a block, and then the time before which a block can be considered safely committed to the blockchain.

I want to thank Haseeb, my friend who gave me this question to ask. I should give him a shout out. What are the different kinds of scalability that the different blockchain engineers of the world are focused on, and where do we stand with these different problems?

[0:52:31.3] JC: Okay, in my mind, scalability is really about throughput. Now, the other things you mentioned which were settlement time —

[0:52:39.8] JM: Latency and finality time.

[0:52:41.7] JC: Latency and finality. Yeah, I'm not sure if those are really scalability questions. I guess in a sense they could be considered that. Latency is, I guess, the amount of time before you learn of a transaction. Is that what you're kind of referring to? If somebody send you money, how long does it take for you to basically have that transaction appear in your wallet?

[0:53:02.8] JM: Well, the speed with which it can be mined into a block.

[0:53:06.8] JC: That seems to be a functionality of the mem-pool. Every miner and every node keeps basically a list of transactions that are waiting to go into the blockchain. This is the mem-pool. The speed with which transactions enter the mem-pool I think is relatively fast and they're not really a big concern, because that speed is much faster than the speed of blocks being found, which is 10 minutes on average. I'm not aware of any huge amount of effort being put into speeding up the time it takes to enter the mem-pool.

Now, the time it takes to have a transaction be considered final, Satoshi gave us this kind of rule of thumb of six blocks, six transactions are about an hour. That was just kind of a quick back of the envelope calculation as to how much time you should wait before it becomes relatively hard or infeasible to reverse that transaction. You would need quite a bit of hashing power to undo six block confirmations.

Now, the time it takes to get six block confirmations really depends on the transaction backlog, or that mem-pool. If the mem-pool is large relatively to the block size limit, like let's say 40 megabytes compared to the block size limit of one megabyte, if you've paid a low fee, you could be waiting a long time. I think it does come back to throughput overall.

The last thing I'll say about transaction finality is — Satoshi's rule of thumb is probably good for small amounts. For large amounts, the way I like to think about it is I won't consider a large transaction final until the amount I've received is less than the cumulative miner's reward. You don't want to be in a situation where you've just accepted a million Bitcoin transaction and you've trust it after six blocks, because the incentive of double spend is huge in that case. You really want to wait till the cumulative block reward is larger than the amount you've received. At that point, undoing those transaction, undoing those block confirmations will be more expensive than the reward of stealing the amount that was sent to you, or double spending.

In any case, the real thing is a throughput, which is throughput can happen in two ways. On chain, which block size limit is one way to get more on chain capacity, reducing the transaction size is another way to get more on chain capacity and kind of aggregating transactions is another way of reducing their size. There's lots of techniques for maximizing our use of block space. Then there're techniques for increasing block space. Those are kind of the things that are being debated in terms of getting more throughput.

Now, throughput can also happen off chain. If you're using Coinbase to send Bitcion, you still, in a sense, are sending Bitcoin even though you've never actually touched the Bitcoin network, and this is known as off chain scale. Using a centralized provider like Coinbase is one way to do it. You can also use lightning, which is another way to do it. That's kind of, I guess, the overall picture.

[0:56:22.6] JM: Okay, Jordan. I appreciate you taking the time to come on to Software Engineering Daily, and it's been great talking to you about Bitcoin and blockchains and scalability. Maybe we can talk again in the future.

[END OF INTERVIEW]

[0:56:37.6] JM: We are partnering with Indeed Prime to give you some tips on thriving in the modern workplace. You can go to indeed.com/sedaily if you're looking for a new job. Indeed Prime is a great resource to help you find that new job.

Today's tip is about assessing culture fit at different companies and finding development work on a product that your values align with. We're going to talk about how to assess culture fit for an open source project that you care about as well.

First of all, what is culture fit and why is it important? Culture is the things that people do at a company. It's also the things that people don't do at a company. That sounds bland, but that's just what it is. Culture often develops naturally, and if the founder or the person at the top of the company is not watching it closely, culture can go south. For companies that are successful, often they've done something right about the culture. Maybe the culture is not perfect for everybody, but it's distinct and it works for the people who are in that company. How do you find out about what the culture at a company is before you join that company?

The company is constantly telling you who they are and what kinds of employees they want through subtle queues. These queues will manifest in their external hiring materials where they talk about what they're looking for in employees. They're going to manifest when you're talking to recruiters on the phone, and certainly when you're being interviewed by engineers or other people, by founders or whoever else is considering hiring you. You will find subtle queues for their work environment.

There are pivotal questions, like what do they do for fun? How many hours a week do they work? Do they expect you to be in the office? What do they do when there's a conflict with an employee? There're lists of questions that you can find about how to assess cultural fit, but the most important thing is to be on the lookout for these queues and also know what you want. Who are you? What is important to you in a company?

Some people like a relaxing work environment. Other people like an intense and stressful work environment. Some people like to maintain software. Other people like to invent new software. There are cultures where these different practices are encouraged. There are also subcultures within giant companies where you could find the right fit. Maybe if you're applying to a giant organization and you feel like you're a technical fit, but the current section of the organization that you're applying to doesn't seem like a cultural fit. You can always look elsewhere.

By the way, there are websites like keyvalues.io for companies that have self-reported their values, because obviously a company doesn't want to misrepresent its values. They want people to know upfront this is an intense work environment or this is a relaxing work environment. The other thing you could do is you can ask around in Slack channels that you participate in or within your network. You could reach out to people on LinkedIn or ask people to coffee. You can ask them, "How do you like working where you work? What does it like? Tell me about your manager. Tell me about the interactions with your manager. Tell me about the projects that you work on in the work environment."

Just to conclude, I think that this also goes for open source projects in many circumstances. Open source projects are largely hacked on remotely, but there is still a culture. There is a culture of a certain kind of feedback, certain kinds of interactions and expectations. If you're looking for open source projects, you probably want to asses cultural fit as well as technical fit or any other kinds of fit.

Thanks to Indeed Prime for sponsoring this tip on Software Engineering Daily. You can go to indeed.com/sedaily to find out more about how Indeed can help you find your next job. If you want to work at a place like Facebook, or Squarespace, Indeed Prime can help you find that dream job.

Thanks again, Indeed.

[END]