EPISODE 530

[INTRODUCTION]

[0:00:00.4] JM: There are two factors that limit the rate at which transactions are accepted into the Bitcoin blockchain; block time and block size. Block time defines how often a new block is appended on to the blockchain. Block size defines how many transactions fit into a new block. As of March 2018, the current block time and block size allow for about seven transactions per second to be accepted into the Bitcoin blockchain. In today's episode, we discuss the technical limitations of the Bitcoin blockchain and some potential solutions to scalability; SegWit and Lightning Network.

Today's guest is Peter Ullrich, the host of Explain Blockchain. Explain Blockchain is a podcast I have found tremendously useful as I've started to learn about blockchains. He provides thorough technical explanations of complicated topics, and I recommend subscribing to his show and listening to the episodes multiple times because there's a lot of content condensed into a short amount of time.

Over the next month we’re going to be exploring a variety of blockchain-based technologies and some of the interviews will be high-level conversations. Some of them will be deeply technical and assume a strong understanding of Bitcoin and Ethereum, and some episodes like today's will be aimed at the developer who is just getting started but going down the rabbit hole and really trying to take this area seriously. So it gets pretty technical in this episode. Don't worry, not every episode that we do about the blockchain will be like this, but I have a feeling that some people who are really going down the rabbit hole will appreciate the level of technical depth in this episode.

If you're looking for an internship, you can apply to the Software Engineering Daily internship at softwaredaily.com/jobs. If you're looking to recruit engineers, you can post jobs for your company there as well. It's completely free to post jobs and to apply, and we’re hoping to find interns to contribute at softwaredaily.com/jobs. The project that these interns will be working on is the Software Daily Open-Source project which you can find at github.com/

softwareengineeringdaily, or you can check out our apps in the iOS or android app stores. They have all 650 or more of our episodes with recommendations and discussions and much more.

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

[SPONSOR MESSAGE]

[0:02:27.6] JM: LiveRamp is one of the fastest growing companies in data connectivity in the Bay area, and they're looking for senior level talent to join their team. LiveRamp helps the world's largest brands activate their data to improve customer interactions on any channel or device. The infrastructure is at a tremendous scale, a 500 billion node identity graph generated from over a thousand data sources running a 85 petabyte Hadoop cluster and application servers that process over 20 billion HTTP requests per day.

The LliveRamp team thrives on mind-bending technical challenges. LiveRamp members value entrepreneurship, humility and constant personal growth. If this sounds like a fit for you, check out softwareengineeringdaily.com/liveramp. That softwareengineeringdaily.com/liveramp.

Thanks to LiveRamp for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[0:03:34.7] JM: Peter Ullrich is the host of Explain Blockchain, a podcast about blockchain technologies. Peter, welcome to Software Engineering Daily.

[0:03:43.5] PU: Thanks, Jeff, for having me.

[0:03:44.7] JM: So you're the host of Explain Blockchain, and that's a podcast I've listened to every episode from — I think there's just four or five episodes, and I know it's time consuming to make those episodes because your format is a lot more work intensive than mine. You do monologues where it sounds like you write them beforehand and they're long form, very well-written episodes explaining specific aspects of blockchain technology. Why did you start that podcast?

[0:04:14.3] PU: Well, Jeff, first of all, thank you for the compliment. It is certainly a lot of time, but when I started researching this area, so I'm personally a computer scientist and I'm just more interested in the technology, I had the problem that I couldn't find very easily accessible resources to specially go into the depth of the technology behind it. So I was researching more and more, then I just figured I need an outlet to summarize all this knowledge that I gain here, and I started writing obstacles, but I'm a really terrible writer, because I'm a perfectionist on that and it takes me two days to write one article. So I thought about doing something more audio-based, and I listened to your podcast, I must say, and also to other very great podcasts and I thought, “Well, why not give it a shot?”

So I started Explain Blockchain a couple of months ago and I was basically just summarizing what I was learning along the way, and I hope that I put it in simple terms so that also people who are not necessarily computer scientists can also learn from that and gain something from it, because I must say that especially in this ecosystem where nobody really wants to trust anybody. I mean, that’s ground of truth of it, that people are sometimes also just throwing wrong terms to make a point, and if you don't understand what they mean, if you just don't understand what the background of this is, then you can also not make an educated decision, I would say. So making this podcast was a way for me to also give back to the ecosystem and to educate the people who are joining it.

[0:05:43.4] JM: What's been so useful for me listening to the podcast is — So like most topics that I cover, I don't have a whole lot of trouble getting to a level 6 maybe, or a level 5 understanding, out of 10, or level 4 understanding, basically whatever I need to talk about it cogently with a guest. I don't have much trouble with that. What I've learned is that you can't do that with blockchain stuff, because it's like a totally new stack and it's actually worth learning and it's worth going rather deep. I kind of feel like people who take themselves seriously as computer scientists or software engineers who want to keep up with the curve, and I mean they tell you that in university even, like they will tell you, “This field moves really fast, and if you want to stay on top of it, you just have to constantly be learning and constantly be doing research,” and blockchain technology is like this whole new area of it. For me, it's just I have to do significant additional labor relative to other topics, and there's no choice, and that's why it's useful to have an outlet like a podcast that I can listen to. I listen to yours. I listen to the

Epicenter Bitcoin Podcast. I listen to Unchained, but I really can't get enough audio contents. It's really complementary to really good written content that's out there, like textbooks and mastering Bitcoin and whatnot.

I think it's really important to have it in multiple different formats, but would you agree with me that this is something that basically if you're software engineer or you're a computer scientist, you really should take this stuff seriously and you should look into it or do you think it's a niche? Like is this just something that only blockchain aficionados should learn about?

[0:07:25.5] PU: So you mean blockchain technology or —

[0:07:28.0] JM: Yes. I'm trying to say like is this something that every software engineer should learn?

[0:07:32.4] PU: I would say at least the basics of it, because it became such a buzzword in the last two or three years especially last year, 2017, that if you're a software engineer, I'm pretty sure your relatives or your boss will have asked you, "Okay. Do you know anything about blockchain? What can we do with it?" And if you ask these kind of questions, then I find it really valuable if you have at least some kind of a basic understanding, basic knowledge not necessarily of the really underlying technology, but maybe also about the use cases for which you could use blockchain, for example.

So I would like to quote the CEO of Chain, the company. He gave a really good interview at [inaudible 0:08:11.3] a couple of weeks ago where he said that you can use blockchain for two main use cases. One of them is if you want to keep track of updates, if you want to audit yourself. So if you, for example, have a database and you really want to record every update you make to that database. So this is the first use case, because it makes it really easy for you later on to just some up all the data and put in a spreadsheet and then given to, for example, auditing companies or to compliance companies, and it takes away a lot of extra overhead of also storing every update somewhere in a different database. So that's the first use case.

The second use case is if you want to have communication between companies or entities in general without any intermediary. You also have to be careful because sometimes I must say an intermediary, although you have to trust that intermediary, it’s still sometimes more applicable to a use case, so that you really have to look into, “Can we actually trust a central party?” or if the trust is broken, that's always a good question, like if that central party is hacked, then will we lose a lot of very sensitive data, for example? If that is the case, then I would say, “Okay. Maybe rather use something like the blockchain technology to just cut out the intermediary and then have direct peer-to-peer communication,” and those are, I think, the two use cases in which you could very well use the blockchain. And to come back to question, yes, I think just pick up maybe the book, Mastering Bitcoin by Andreas Antonopoulos. I really recommend this book over and over again. Also, my podcast, because it is a great read, a great introduction to the Bitcoin blockchain particularly, but if you learn that one, the original blockchain, then you really understand also the future development of blockchain.

[0:09:57.1] JM: Definitely. So the two use cases you listed, essentially the ability to checkpoint a database or some kind of data structure that you have in an organization to checkpoint it with a place of public record. Then use case number two, the ability to do transactions without a trusted intermediary.

It's funny, because neither of those are a currency. I mean, those are things that actually require you to have a currency involved, because you have a currency in order to pay the miners to power the system, but I think that’s something that actually confuses people, is that a lot of the value of this technology is not in the currency. Would you say that’s a common point of confusion?

[0:10:44.4] PU: Absolutely, but I also think that it comes from the fact that it started with a currency. So the first blockchain was used for the application of cryptocurrency and, also, now people just talk about cryptocurrencies much more than they talk about blockchain technologies. Yes, in general, I think that the underlying blockchain technology will be much more valuable and will also reach many more areas of the industry than cryptocurrencies.

[0:11:11.1] JM: Definitely. So we could talk about basics for a long time, but I want to get into discussing scalability with you, because that's what we prepared for, and you have a great episode about scalability. That's what inspired me to reach out to you. So ancillary information on this episode can be found in your podcasts, in the Explain Blockchain Podcast, the episode about scalability. So I think we'll assume that people have a basic understanding a Bitcoin for this episode, and if they don't, they can go back into my back catalog or your back catalog regarding scalability.

We are talking about the Bitcoin blockchain's scalability. The Bitcoin blockchain is communicated about across the internet. We don't assume that the internet has a scalability bottleneck. I mean, it does have a scalability bottleneck in some sense. You could say that that's what net neutrality is arguing about, but for the most part, we can send around movies and audio files and high bit rate things, whereas Bitcoin has problems even sending around these small blocks of financial transactions. Why does Bitcoin have a scalability problem?

[0:12:20.1] PU: Yeah. So it is not a problem of internet throughput of the data that is transferred, although it's also part of the architecture, but I will come back to this later. The big problems is actually the computational bottleneck and also the storage bottleneck. So if you go to Bitcoin core or Bitcoin in general, the underlying architectural decision of Satoshi Nakamoto in the very beginning when he or she or the group created Bitcoin was that it should be decentralized network eventually. And if you want to have a truly decentralized network, you need to make it accessible to almost everybody. Ideally, really everybody, but not everybody can afford a cloud server or a whole data center. But people, like you and I, we can afford like a home computer or maybe a Raspberry Pi for a couple of 40 bucks, something like this, and we're able to use that hardware that is at our disposal to run a Bitcoin node on it, then you will also eventually have many more nodes in that network and they will also make it more decentralized.

So the scalability issue of Bitcoin stems from this architectural decision that, first of all, the blockchain shouldn't rise too rapidly. So the size of the blockchain shouldn't rise rapidly, which is why we have the 1 megabyte block size cap, or we segued it around 4 megabyte now, and that is basically so that you and I can also in 10 years from now on still store the whole Bitcoin blockchain on our Raspberry PIs on our home computers. It wouldn't be a problem. Like if you take YouTube, right? They have like 400 gigabytes, I think, a minute that they have to store somewhere and they are equipped to do that and they also have the financial resources to do that.

But let's say that you want to have a second YouTube. So you want to separate the whole database of YouTube and save it like a second time. Obviously it's linear increase, so you have double the cost and everything. So these companies, they can have such a high throughput because they centralize everything, they highly optimize their own system and so on. But as I said, it's not applicable to the average Joe, and we should be able to run the Bitcoin network. So that is the first part, that you have the block size cap.

The second factor why Bitcoin has a scalability issue at the moment is the block time, and those are the 10 minutes you hear about every now and then. Satoshi Nakamoto again, in the beginning, thought that we should only add a block to the Bitcoin blockchain every 10 minutes, and the rationale behind this was that, let's say that somebody in South Africa makes a transaction using Bitcoin, and the full node in South Africa will pick up this transaction very quickly and then also can add it to their own blockchain or can mind it, can try to put it into a block, because they have this transaction. But in order for Bitcoin to become a truly global network, you need to give the full nodes, for example, in South America and in Asia enough time to also pick up this transaction. So that means that the transaction needs enough time to travel around the globe so to say.

Back in 2008, Satoshi Nakamoto then said, “Well, let’s just use 10 minutes.” I'm pretty sure there was some kind of rationale behind it, but it's just now 10 minutes. This also then poses the problem that only every 10 minutes you can add a block which has a certain hard cap on the size. So every 10 minutes you can only add so many transactions to the Bitcoin blockchain, and this leads to a theoretical throughput of the Bitcoin blockchain of three transactions per second, and now we can go in the next step, if we compare them, the Bitcoin blockchain to, for example, the visa network, which has a theoretical throughput of 50,000 transactions per second, and that's only the visa network. We have to then make Bitcoin somehow more scalable, or to solve the scalability issue in order to compete, for example, within visa network.

[0:16:18.7] JM: So step back for a moment. People are making transactions across the Bitcoin network all the time. They're taking place in South Africa, in Argentina, in North America and as these transactions are being created, they’re going into the mempool, which is where the transactions that are waiting to be processed sit, and then the miners select from the mempool the transactions that will be included in a block. Could you explain that process and explain how miners decide which transactions to add to a block?

[0:16:56.8] PU: Absolutely. So when you make a transaction, you will send this transaction to the full node which are connected, maybe your own full node, and then also to all the other full nodes around you. So that means that the miner nodes will eventually get this transaction into their, as you said, mempool, a memory pool, and then they select the transactions with the highest transaction fees. Now, I assume that you know a little bit about how a Bitcoin transaction looks like, but in general they are inputs and outputs values. Inputs are previous transactions that you now use to spend Bitcoin further, and the output values the addresses to which you want to send Bitcoin.

Now, if you sum up the input values that you put in, so you take like two or three different transactions, each of them let's say has like one, two and five Bitcoin. So the overall sum is 8 Bitcoin that you have as an input. But then in the output you only send, let's say, 7 Bitcoin and to a certain address and the one Bitcoin you leave unspent. So you don't send it back to yourself or you don' send it to anybody else. This difference between the input and the output is called the transaction fee, and then be collected by the miner ones they take your transaction and put into a block.

So the thing here, it’s a free market. So if you now want to have your transaction be on top of the list of transactions that go into the next block, then you also have to pay relative to the other transactions that are in the mempool a relatively high transaction fee. Well, the highest transaction fee. This led to the problem that in the end of 2017 where you had a very lot of transactions going to the mempool, that you had to pay very highest fees, like around almost $40 on average so that your transaction would be part of the next or the over next block. But luckily in the last month or so, actually this fee drop tremendously down to a couple of cents now, I think. So this is not a problem anymore, luckily.

[0:18:57.9] JM: Why did it drop so precipitously?

[0:18:59.8] PU: That's a very good question. One reason for this or a believed reason is that while in December you had a very strong, a very fast adoption of regular users of Bitcoin. So it was all over the press. A lot of people learn about it, and then they also started to make Bitcoin transactions, and these Bitcoin transaction, for example, mostly came from exchanges where people were buying Bitcoin with a normal currency, with a fiat currency, and then they were sending that Bitcoin to their own wallets where they can control their Bitcoin. All these transactions from and to exchanges and also between people and so on just filled up the mempool. So there were around 160,000 transactions in the mempool, but only a thousand or so can always go into a block, and then people still needed to make transactions or one or two have quick transactions and then they just started paying more. It's a supply and demand problem there.

So just to sum it up, in the last month so we saw that, first of all, there were fewer transactions made just by — Because the adoption drop, because the price dropped. We also saw a change with Coinbase. Coinbase is the biggest exchange for Bitcoin, and what they did is they, first of all, used implemented a batching system so that they didn't send out transactions for every single purchase of a customer, but they batch these purchases together into one transaction. So instead of, let's say, like a 10 or 100 different transactions, they only collected those into one single transaction and it also led to a decrease of the transaction fees eventually.

[SPONSOR MESSAGE]

[0:20:43.0] JM: This episode of Software Engineering Daily is sponsored by Datadog. Datadog integrates seamlessly with container technologies like Docker and Kubernetes so you can monitor your entire container cluster in real-time. See across all of your servers, containers, apps and services in one place with powerful visualizations, sophisticated alerting, distributed tracing and APM. And now, Datadog has application performance monitoring for Java.

Start monitoring your microservices today with a free trial, and as a bonus, Datadog will send you a free t-shirt. You can get both of those things by going to softwareengineering.com/datadog. That's softwareengineeringdaily.com/datadog.

Thank you, Datadog.

[INTERVIEW CONTINUED]

[0:22:09.2] JM: So in that high traffic period where there's thousands of transactions waiting in the mempool to be processed and only seven transactions per second, or something like that are being processed. If I don't pay a high transaction fee and I submit a transaction, does that transaction ever get processed or could it just sit backlogged and never make it into a block?

[0:22:33.9] PU: Absolutely. So if you pay a very low fee, then it could happen that you just don't get into the blocks and the next time, and I think there's a limit to how long such a transaction can stay in the mempool until it's just disregarded, which is around two weeks if I'm not mistaking. After two weeks, if your transaction wasn’t processed, then it would become invalid and then you have to make a new transaction.

[0:22:57.8] JM: But there is a way for somebody who would have such a transaction to know that, right? Like they could check blockchain.info and see their transaction sitting in the mempool or eventually evicted from the mempool, right?

[0:23:09.6] PU: Exactly. So the thing is mostly you make a transaction using a wallet that handles all of these equation of a transaction and so on for you, and they are quite some wallets that had very bad algorithms to determine what the optimal transaction fee is. I think Coinbase was one of them that was called out, and I only know about Coinbase that was called out, because they algorithm to estimate the transaction fee was queued. So they always paid much higher transaction fees.

The problem with this is, again, that the mempool, or the memory pool, it’s not a global state. It’s only what a full node has in their memory pools. So every full node has an own memory pool. Then if some exchanges or some, let's say, exchanges have their full node where they have a lot of transactions that have very high transaction fees, then they would also recommend a higher transaction fee to the customers, but then you also have individuals who have their own full node and they only see that not that many transactions were very high transaction fees, so they have a different estimation of that transaction fee that they should pay. So it's very difficult to come up with this magic number of how much you should pay.

[0:24:17.9] JM: Yeah, because — So as you said, each of these full nodes has their own mempool and they’re solving their own blockchain puzzle based off of the transactions in that mempool, and if your transaction is not in their mempool, they're not even going to be trying to solve your puzzle. I mean, do the full nodes aggressively share the different transactions that are in their mempools or is that more of a lazy kind of thing, or do they only share them when a block gets discovered?

[0:24:48.5] PU: Yes. So the full nodes use two systems, one of them is the flooding algorithm. So whenever they get a transaction, they just send it to every full node they're connected to, and if they already received that transaction, then they just disregard it. The second algorithm is the gossip algorithm. So just every now and then they connect to the other full nodes and just share the transactions that they received.

In general, I just wanted to point out that full nodes don't necessarily mine the blocks. So you said that a full node in the mempool, they will take the transactions and try to make a block out of that. That is not necessarily the case. You can — I mean, miners also our full nodes. So they have their own mempool and from which they take their transactions, but you don't necessarily need to mine if you have a full node.

[0:25:32.4] JM: I mean, how often are the miners — Are they consistently trying to process the same blocks as each other? Like are they consistently sharing the same set of transactions so that they are chasing after the similar set of blocks, or how often is it that their chasing after disjoint sets of blocks? I’m just trying to get a feel for how different the mempools are.

[0:25:56.3] PU: I am not sure whether they would purposely not share transactions with other miners of full nodes.

[0:26:03.2] JM: Because you could imagine, they could just wait around and, “Oh! We've got a bunch of like large transaction fees that we can get from these transactions. Maybe we should just hold them.” But I guess those transaction fees are pretty small relative to the actual award that you get.

[0:26:16.1] PU: At the moment, yes, but in general it’s also — If these miners that just hold back transactions would create a block that includes transactions that no other full node has, that block would not be accepted by the other full node at first. They would hold back until they also receive the transactions, because then only they can validate the transaction they hold in the mempool against the transaction that the miner put into the block. A miner is also through that incentivized to share the transactions that they receive. Otherwise, other nodes will not accept his block.

But what you said is very true. So let's say you have a miner in China and a miner in America. Obviously these two will have quite different memory pools, because in China you have more maybe transactions from China than America, vice versa for America, but sharing these transactions between the full nodes doesn't take too much time. It should take a couple of seconds only. Sure, you have differences, but I don't believe that they're very significant.

[0:27:13.1] JM: Okay. So to go back to the scalability question, we've got this question of block time and block size. So these are two different variables that we could potentially change, or I guess we did end up changing with SegWit. So what are the different options? Like if we were to change block time or block size, how does that affect the scalability of a blockchain?

[0:27:36.2] PU: Sure. First of all, the block — Well, will basically every 10 minutes you can add more transactions to a block, and that is fine at first, then also use more transactions and put them on to the blockchain, but your blockchain will also rise quicker. This is not a problem at the moment. I mean, if you think about the fact that the Bitcoin blockain until now is 140 gigabytes, you can still store it on an external hard drive. But the problem here is also scalability in the long term. So let's say that the only solution you have is increasing the block size, until you meet demands or the use that the Bitcoin network has.

So let’s say you want to scale the Bitcoin network to San Francisco, for example, then you already need gigabyte of blocks, which can hold all the transactions that are made in the metropolis, the area of San Francisco. Then you already have a problem, because you have to download gigabyte blocks within the 10 minutes to also verify them and also add to your own blockchain. If you want to scale this further, if you want to scale this to the level of the visa network, then already get, I think, like 8 or 9 gigabyte blocks, and that also leads to multiple terabytes per year. So that is then, again, not accessible to the average Joe anymore. So you also get fewer people running full nodes, because they can't afford having or adding terabytes of hard drives price every year, and that also leads to a more centralized network eventually, because only fewer people can run a full node, only those that can afford it, and then also only fewer people will hold an verify and mine the transactions on the Bitcoin network.

[0:29:12.0] JM: Here’s a naïve question. So if it's all transactions, like we just got a bunch of transactions in the mempool and we're turning those transactions into blocks, why does the size of the block matter? We’re just handling N-transactions. Are we ultimately just like all these full nodes are just holding transactions and why would the block size actually matter? That’s just where you’re slicing these series of transactions into.

[0:29:39.8] PU: Yes. The block size model is for two reasons. One of them is only the problem that you have to store it, because of full node has to store the whole blockchain from 2008 on until 2018 at the moment. If you increase this size of the blockchain rapidly, so if you increase the block sizes, then, well, in the years’ time or two you will not have 140 gigabyte, but you will have multiple terabytes, and that's not something that somebody like an average Joe can just store like that.

The second problem is that if the block size becomes very large, then also not an average household or an average person can download that block within the 10 minutes until they receive the next block. So let's say that you have to download every 10 minutes 8 or 9 gigabytes of data. If you increase the block size rapidly, then at one point, just normal people just can’t download the block anymore within the 10 minutes until they get the next block already. That also means that they can't verify and validate that all the transactions are valid and then also that the next block is also in a valid way added to the old block. So it becomes a problem of bandwidth to download the block and it also becomes a problem of computational power to verify the block. So that's why there's a threshold for the average person to hold the blockchain.

[0:31:01.8] JM: So is that to say that there is always this large mempool of transactions that are waiting to be received and the block size determines how rapidly the mempool is depleted?

[0:31:18.5] PU: Yes. Well, again a supply and a demand problem. So if you have the same amount of transactions that you have at the moment, so the demand to be put in the block and then you increase the supplies, so you increase the block size and the capacity of adding blocks, then yes, you would have a decrease of the mempool in the long term. But then also if the demand suddenly rises with the block size, so the demand rises with the supply, then you have the same situation as we have now, where you still, well, a month ago had to pay $40 to get into the next block.

[0:31:52.2] JM: I see. So if that block size increase happened, then people would eventually realize, “Oh! It's actually getting cheaper to transfer money on the blockchain, because things are happening faster. Transactions are getting accepted faster,” which is making it cheaper, and it ultimately could end up being attacks on the miners because they are the ones who would have to store, I guess, all full nodes, but mainly the miners because they're the ones who are, I guess, I predominantly going to be storing this entire chain. I guess that gets us to why this was — There was so much debate around this Is that correct?

[0:32:33.7] PU: Yes, that's very true. So there are some people in the ecosystem that think, “Well, it should be of free market and whoever can pay for the computational power and the bandwidth, well, is just not able to pay.” So those who don't pay our can’t pay, well, it's bad for them, but I don't care. That's more like the free market approach to changing the parameters of the Bitcoin blockchain.

On the other side you also have the more — Not necessarily altruistic, but more incorporating approach by, for example, the Bitcoin core team. Let’s say, “Well, we as network are only as strong as the lowest, so to say, network full node.” So if the lowest full node with the least computational power and bandwidth can still hold a full copy of the Bitcoin blockchain, then we will also have more nodes. So we’re only as strong as the lowest full node.

[0:33:21.4] JM: So if we have the block time, that would be essentially the same as doubling the block size, right?

[0:33:27.3] PU: Yes, very true. So then you also add more blocks faster to the blockchain and that also means it increase in size faster. But the problem is that not every transaction then can reach all of the notes in the network and they will also be at this balance of power among the miners, for example, where some miners hold more transactions and can also pick and choose the transaction fees and some miners don't get as many transactions and just can't keep up with the other miners anymore.

[0:33:54.9] JM: Okay. So I think we've outlined the scalability problem as reasonably as we can do in audio format. Let's start to talk about some solutions, some solutions to scalability. First, there's SegWit, so Segregated Witness. It has two parts. It's got a new structure for transactions and a new way to calculate the block size. In order to get there, maybe we could just refresh people on what is in a block, like just give an outline for the different fields that are in a block.

[0:34:29.6] PU: Absolutely. So you, first of all, have the input section where you use your unspent transaction outputs from previous transactions to make future — Or to make new transactions. So you use a past transaction that you haven't spent yet and use it as some kind of proof that you are still holding or owning a certain amount of Bitcoin. So you put this in the input field and then he also put the unlocking script for that particular transaction.

So the unlocking script is a — Or at least on the Bitcoin network, is a combination of your public address from which you took that unspent transaction and a signature of that transaction. So that means that you as the owner of the public-private key pair, you sign or you encrypt the data of a transaction and basically add it to the input. What the script that — It's a little technical here, but just follow me. So what the script in the input field in the transaction then can do is take your public address, decrypt the signature with it and then see in the transaction data you encrypted is the same as the transaction data that you put into the transaction. So it's a verification that you own the public-private key pair to which these Bitcoins were sent. It's a very low level, but that's in general the input section.

Then you have the output section, which is easy. It's just the address to which you want to send the Bitcoins and the value of how many Bitcoins you want to send. That is the output section. Then you also have a third 3 section, which is the block header, and on the block header you have a couple of meter information about the blocks, so the block height, for example, what number this block is in the long chain of the blockchain. Also the Merkel tree root, which is a verification or a signature of the data that is in the block, and also some other things, like a timestamp and the version of the node you are running and these kind of things.

[0:36:31.9] JM: And one of those parts is known as the witness. What is the witness refer to?

[0:36:37.5] PU: So the witness is what I just explained, the input or the unlocking script that unlocks your transaction so that you can use it for making other transactions with it.

[0:36:47.7] JM: Wait. Is it input or is it the output field?

[0:36:50.1] PU: It's the input field.

[0:36:51.4] JM: The input field. Okay, got it. Yeah, the input field, it's an ID of the UTXO, which is an unspent transaction, because you don't actually just hand around Bitcoins. You hand around transactions. So the input field has the unspent transaction, and then it has an unlocking script which defines how that UTXO can be unlocked, and then the output field, which is not the witness. The output field has the address of the recipient and then your own address for the remainder coins. So the input fields is what we're talking about here. The ID of the UTXO that you're dressing to somebody, and in the unlocking script which defines how your UTXO is going to be spent, and SegWit proposed moving the unlocking script. So let's just talk a little bit, what is an unlocking script? What's the purpose of this unlocking script? Why is that a field in a Bitcoin transaction?

[0:37:49.5] PU: Absolutely. It becomes a little bit technical here, but I will try to explain it in simple terms. First of all, a transaction can be made to any public address. The public address is the public key of a private public key pair. Now, if you want to prove that you are the owner of this private public key pair, what you do is you take the transaction, the UTXO, and you encrypt it with your private key. So you just take all the data that's in there, encrypt it and get a signature out of it.

Then you use this signature in the combination with your public key and use it as an input parameter to the locking script of the transaction. So the locking script, it's actually a very short script. It's a program that you can execute and it is written in a language called Script. It's a very low level, a very basic but therefore also very secure programming language. You can add any input parameters to this script. You run the script and the script decrypts, again, the data you put in, so the transaction data. It checks whether that data you put in is equal to the data of the transaction, and if this is all the case, then it will just return true, and if it returns true, this is the proof that you are the owner of the private public key pair. Is that better?

[0:39:12.9] JM: Yeah, definitely. It's as good as we're going to get. By the way, like I want to just address the fact that this stuff is really hard to explain over audio, and I think you have probably learned that. I have certainly learned that in in past episodes I've done, and this is typically a learning experience for both of us, and I'm sure the stuff will get easier to understand over time as we move up the stack, but I'm glad we're doing this. I'm glad we're continuing to go through the stuff that's difficult to understand, because I know that people want to hear about this low-level stuff and I think we're just doing the best we can. We'll continue to refine our ability to explain it over audio.

With that said, the SegWit proposal moved the unlocking script out of the transaction. Explain the motivation for that and what does that actually mean, moving the unlocking script out of the transaction.

[0:40:01.4] PU: Absolutely. So first of all I would also, because you mentioned it already, it's very hard to explain this via audio, and I can only recommend again here the book; Mastering Bitcoin by Andreas Antonopoulos, which is also freely available on GitHub. If you really want to understand this in more depth, then I would really recommend looking up this book. It explains it perfectly.

Now, back to your questions. So the problem with this script or with having these unlocking parameters inside a transaction, that means —Well, basically the whole transaction is just a key value dictionary or map. I's like a data structure. That's it. And somewhere in this data structure, in this key value map, you have then the inputs parameters. But this poses a problem which is called transaction malleability, and that means that let's say you put in your two parameters, your public key and the signature. What you also then do to create a block eventually, whether what the miner does in order to create the block, is it creates a miracle root of all the transactions that are in the block. So I won’t explain what the Merkel root [inaudible 0:41:11.3], but it's basically a signature of the data that is inside the block.

Here’s comes a problem, because if you, let say, have a transaction, you put in the two input parameters, but then you add different parameters to it. It could just be a random data and then you have sort of drop command. Just everything you put in, you just drop, disregarded, and then you put in the true parameters. This would still unlock the transactions. So the transaction you make, it’s still valid and it will still be included in the block. But the eventual Merkel tree root [inaudible 0:41:44.9] will change and also the transaction ID of your transaction, which is also a signature of the data that's in there will also change. So that makes the unique ID of a transaction changeable by just adding some random dates to the unlocking script. This means that you can't trust or you can't reuse the transection ID of a transaction for making future transactions.

So let's say that you want to create a chain of transaction, like two or three different transactions, but all these transactions you create before you put it into a block somewhere. Now, if you now change the very first transaction, if you just add some random data to the input, you change that ID and then the second transaction will just point back to a transaction that might not exist yet. So this poses a problem if you want to chain together a couple of transactions before you let each transaction hits the blockchain or let be included in a block.

Just to sum up, this is the transaction malleability problem, and SegWit tried to or did solve this problem by just taking out the unlocking script and put it into a different section. Again, you have to key value map and then you just have a high level key, like a different section that’s just called witness, and this witness section then holds the changeable parts of your transaction, so the locking scripts and someone, and it will not be included if you create this unique signature of a transaction, which is the transection ID.

So if you create a transaction, you cannot and you create the ID of it, then you can be 100% sure that this transection ID will not or cannot be changed afterwards anymore. It’s unique for the data. And this enables you to make, again, transactions based on the original transaction before the original transaction hits the blockchain in general, and this is a very perfect lead up to the lightning network, which I hope we will talk about soon.

[0:43:43.8] JM: Definitely. Just to recap though, why did SegWit reduce the burden of the blockchain scalability problem on Bitcoin?

[0:43:54.6] PU: So because it also added a second feature to the transaction, and this is regarding the block size and also the transaction size. Before this of the block size was calculated by only using the role size or the role the storage that's a block takes up. So how much space it takes up on your hard drive? But with SegWit, this was changed to a weighted system, and instead of, well, just taking the space, it gave points so to say for every byte you have in the transaction area, so the input-output area, and it gave a different way to everything you had in the witness area.

The weight [inaudible 0:44:35.8] is full for every byte in the input-output area and one for every byte in the witness area. Then also the overall block size of the hard cap on the block size was also changed to a wider system where you could have 4 million of these weight points, so to say, in every block. This means that the way you calculated the block size changed to a wider system and this practically increase the block size of the block to something a little bit underneath the 4 megabyte. It effectively created the block size as well, and that again led to a decrease in the mempool because you could just use more transaction, put them into blocks. Also that it decreased your transaction fee, because before that you also kind of had to pay how big your transection is, because if you have a very large transection, then the miners couldn't put smaller transactions into a block. So by making very large transactions, very costly and incentivizing smaller transactions, you also decrease the transaction fees for the people who just make normal transactions.

[SPONSOR MESSAGE]

[0:45:52.8] JM: Software Engineering Daily is brought you by ConsenSys. Do you think blockchain technology is only used for cryptocurrency? Think again. ConsenSys develops tools and infrastructure to enable a decentralized future built on Ethereum, the most advanced blockchain development platform. ConsenSys has hundreds of web three developers that are building decentralized applications focusing on world-changing ideas, like creating a system for self-sovereign identity, managing supply chains, developing a more efficient electricity provider

[INTERVIEW CONTINUED]

[0:47:09.4] JM: Was that what led to, for example, Coinbase doing this kind of batching, because it sounds like they incentivized people to do that kind of batching? They incentivized people to, if they could, make their transactions for bigger amounts.

[0:47:23.4] PU: Exactly. Sorry. I think I made a little mistake there, because with having the four weights of the four points on the input-output data and only a one weight on the witness data, you actually incentivize having more input-outputs in the same transaction, because previously the witness data, so the unlocking scripts, actually took up around 70% to 75% of the overall size of a transaction, and now with SegWit, you could put in many more transactions into the input-output sections and then also add all of these extra data, these 70%, 75% of data to the witness section where it would be weighted lower, where it would be more cheap to add it to. So this actually led also to Coinbase to start creating bigger transactions, because they could save basically on that.

[0:48:15.1] JM: Okay. Well, I think we've discussed SegWit as much as we have time for, and we should get on to lighting networks. People, again, can listen back to your episode for more on SegWit. So lightning network, this is an example of an off-chain solution. Explain what an off-chain solution is.

[0:48:34.3] PU: So off-chain solutions any solutions that take transactions off the Bitcoin blockchain, and that means that if we were to make transactions, that we don't have to wait for every transaction to actually be included in the blocks and added to the blockchain, and lightning network, for example, is one of these solutions, because it enables just two parties to make direct payments between each other so not every payment needs to be added to the blockchain. But only, first of all, funding payments and that also later on settlements or settle transection.

So we could make a hundred or a thousand transactions back and forth between people, but then only two of these transactions, so the funding, the beginning and the settlement, the last transaction would be put on to the Bitcoin blockchain.

[0:49:21.2] JM: The lightning network is like a system of payment channels. So you have these payment channels that you set up with somebody else and you communicate to the blockchain itself that you’re opening a payment channel with somebody else and you create what's called a time lock contract when you do that. Then you can transact with somebody else off-chain for a while. Then, I guess, the time lock contract eventually, it runs out or you finish doing your payments with that other person and then eventually your off-chain interaction is reconciled with the main chain. Is that correct?

[0:50:01.6] PU: That's correct. Yes.

[0:50:02.0] JM: Okay. So what are the challenges around that? Because that sounds like a very useful way of doing scaling, because what we addressed earlier is that the main problem is that you have all these transactions that want to be processed by the main chain. So if we can just take some of those transactions off of the main chain, because the frequent problem is like you open a tab at the bar and want to just like buy a drink and then you want to buy another drink and then you want to buy a drink two days later and then maybe you go to the same bar, and it's a coffee shop during the day, and you get a cup of coffee. You don't want to have all of those transactions on the main chain. It’d be much better to just have one transaction that goes on to the main chain that covers all of your cups of coffee and drinks with that bar. So that sounds like a very useful way to scale the blockchain, is just move these transactions off. So what are the challenges to doing that at scale?

[0:51:00.1] PU: So the challenges actually only one big one, actually, because if you want to use it as scale, so want to use it with multiple or many different other parties, then you needs or you would have to need to create a payment channel with every single one of these entities. So whenever I want to trade with a new, let’s say, online web store, I first have to create a payment channel with this web store before I can make these lightning network payments. But this is eventually costly again, because then you have to make one funding transaction to open the payment channel. You send money back and forth and then, again, you close off the payment channel afterwards.

So the challenges here is actually to not have these direct payment channels, but rather to have a network, because I don't need to have a direct payment channel with a web store. I could, for example, have a payment channel with a friend of mine or with a different web store that then again has a payment channel with that particular web store. These people then become intermediaries. If I want to send any money to that particular web store, I have to, well, trust these intermediaries again, because I can’t just send money to my friend, for example, but that friend might not send or forward the money to the web store that I want to pay. So then you have the trust issue again.

He actually — What the lightning network does is it uses these hashed time locked contracts that you talked about earlier, and that means they are actually — You make a transaction to your friend that can be resolved or that can be spent again or used in two different ways. One of them is the time locked part of it. Actually, I make a payment to my friend and I just say, “Well, if this friend doesn't unlock this transaction, so doesn't use this transaction, then I would just be refunded within, let's say, 30 days.” So within 30 days I get my money back. I know that for sure.

Then the second part to unlock this transaction is the hashed part of it, and this means that the transaction I make can be unlocked with a certain random data that I put in, and let's just call this data R. This data R is actually created by the web store that I want to pay, and the web store only creates a hash of the data and gives it to me. I use this hash and I include it in the payment to my friend, and let's be an ideal world here, the friend then also pays the web store and also sends forward the hash of the data and the web stores knows, “Okay. This hash is equal to the hash of the data that I hold back,” so I know this payment comes from Peter, from me, and this web store then would also, in order to spend this transaction, have to broadcast the data R. So it will also be received by my friend who can then use this data R to unlock my payment that I made to my friend so he also gets his money, and I eventually also receive R and I can prove with this that the web store actually got the money.

[0:53:56.9] JM: Okay. Let’s go a little high-level than this. By the way, you should definitely come back on in the future, in the next episode. We’re going to do something more a little more high-level. Like I think we should, because I love talking to you, but this is like so low-level, that it's funny because I know that there's like 15% of the audience that just absolutely loves this stuff, because I actually — I was looking for like a very technical Bitcoin audio material when I first kind of started getting into this, and there just wasn't much, and it's like you can't listen to this kind of audio when you're maybe working out or something, but if you're maybe washing dishes, it’s so mindless that you can completely focus on what you're listening to. I think this is the perfect type of material to listen to if you're studying Bitcoin but you have to wash dishes.

For the 15% of the audience that's washing dishes, I hope you appreciate this material. The other like 85% is going to be like, “Well, maybe it'll be useful to them, maybe not. We’ll see.” This will be an interesting experiment to run.

I’ve done very technical shows in the past that people have really liked, like shows on database indexing or stuff like that where they’re like, “Oh my god! I totally love that show. It’s right up my alley.” Interesting.

So there's like kind of some debate around lighting networks. So you take somebody like Andreas, and he seems very bullish on lighting network. He seems like — From my point of view, he just seemed like, “Okay. the Bitcoin is the main chain, and right now we’re just using it for everything,” and of course it's going to have scalability issues. But we’ll build up the second layer solutions, which honestly sound a lot like what we've built on top of internet protocols or abstractions on top of abstractions on top of abstractions. Why wouldn't the same thing happen for Bitcoin? As far as I can tell, that’s kind of Andreas’ bullish case for lightning network. It’s like we’ll relieve the main chain of transactions and we’ll eventually just get really, really powerful lightning networks that we can transact with each other across and will have decentralized Venmo decentralized everything financial.

First of all, correct me if I'm wrong about that, but what are the bearish cases for lightning network? Why wouldn't that work as a scalability panacea for Bitcoin?

[0:56:05.7] PU: Yes, I think it's a more ideological discussion here, because you always have the purists who will just want to use the Bitcoin core software as it is and just really want to stick to the ideology and the idea of Satoshi Nakamoto, and I can totally understand this. It's a very valid point, but especially because this whole project is a software engineering project, it is very hard to keep version 1.0 for more than 10 years. You always have to make updates. You always have to improve on things. You will run into new use cases and discover edge use cases that you have to solve and that you have to cover as well.

So from a software engineering point, it is just natural that Bitcoin and blockchain and cryptocurrencies in general develop over time, and that's good, because, yes, this way we can use Bitcoin in many more areas. We can expand it. It becomes cheaper. It enables more people to have a bank account and make payments across borders and these kind of things. So I think that's also what Andreas comes back to. So he is an idealist. He is a very — I don't like the word, but a thought leader. So he's looking ahead, that's what I mean. If he looks ahead he sees and that the current Bitcoin core software as it is, is not able to scale very well. So obviously we need to make some changes.

But on the other side, as I said, you always have the purist who say, "No. We don't want that. We want to stay as it is. It is not according to Satoshi Nakamoto's idea." we can maybe tweak the parameters a little bit, like Bitcoin Cash does with the block size, and this should be fine as well, right? Well, it depends on how you want to look at this project.

[0:57:47.2] JM: So here's what I don't understand, because I did a show about lighting networks like a year and a half ago with somebody from Blockstream, and this was long before I had any sort of ability to talk about Bitcoin at the level I can now, which is still really limited, but it was somebody from Blockstream, and I thought Blockstream was like the epitome of Bitcoin core conservatism. Maybe I'm mistaken, but, I, mean since they were working on lightning network, it seems like this is solution even for the conservatives.

[0:58:19.0] PU: Yes, I do know why they're working, but I think you always have to innovate in this field, and specially Blockstream that wants to stay at the cutting edge of technology, they also have to invest in this. Again, it just comes down to the ideology and what you believe in.

[0:58:33.5] JM: Are there a large number of Bitcoin core developers that are not fans of lightning network or is it like a really, really small but loud minority?

[0:58:42.3] PU: I actually don't know.

[0:58:44.7] JM: All right, interesting. You mentioned Bitcoin Cash. What did Bitcoin Cash do differently?

[0:58:49.9] PU: So they only increased the block size. So they increased, I think, to around 4 megabyte, or maybe 8 megabyte already, so they just tweaked the parameter of the original Bitcoin core software and they also didn't want to use SegWit, because they were also like, "Well, this SegWit changes the original format that was created by Satoshi Nakamoto of the transactions, and we don't want that."

One other reason they gave was also because the SegWit proposals, SegWit format is a soft fork of the Bitcoin network, and that means that older nodes that don't run the latest software, including the SegWit formant can still accept the new SegWit payments, and the problem there is only that the old nodes, they only see that the SegWit payment they receive can be spent by anybody, because the SegWit formant has a different format there. So for the old nodes, they can't, from that point on anymore verify whether somebody is allowed to spend a SegWit transaction, for example. So that's why the other idea then was that if you, as a full node owner, don't want to upgrade to the latest version, then you're just left behind, and maybe eventually they will also cut off that comparability, so your access to the Bitcoin network. So that is another argument that came from the Bitcoin Cash community.

[1:00:08.7] JM: Okay. Peter, it's been great talking to you. I guess, to close off, what are you working on these days? I think you're working on master's thesis. So what kind of stuff are you researching?

[1:00:16.8] PU: Absolutely. So at the moment I just started my master thesis at a company here in the Netherlands, and we are researching how you can use the blockchain to improve the processes in education. So we would like to see how you can put certificates on the blockchain, how you can also give a self-sovereign identity to students, for example, that the students are really in control of the documents and the data, and for this I researched different blockchain technologies.

[1:00:41.3] JM: Okay. Well, that's awesome. Everybody can check out Explain Blockchain, if you want to hear Peter's fantastic material. Thanks for coming on the show. I really appreciate you making the effort to explain these really complicated topics over audio, because that's my preferred format for long form material. Although I fully admit that with blockchain stuff, you kind of have no choice but to look at some diagrams sometimes.

[1:01:06.7] PU: Thank you very much, Jeff, for having me, and I can only recommend to check out also multiple medium. So, yes, I also like podcasts, but there are also really great visual mediums on YouTube, and also great books. Well, I personally, am more interested in the technology behind it. So I can also recommend the Bitcoin Wiki. They really explain all these topics, like SegWit and lightning in accessible terms, but if you really want to dive deeply into the technology, then you can also do that there, because they also reference to different resources.

[1:01:38.8] JM: Okay. Well, thanks for coming on the show, Peter.

[1:01:40.7] PU: Thank you very much.

[END OF INTERVIEW]

[1:01:44.1] JM: Your enterprise produces lots of data, but you aren’t capturing as much as you would like. You aren’t storing in the right place and you don’t have the proper tools to run complex queries against your data.

MapR is a converged data platform that runs across any cloud. MapR provides storage, analytics and machine learning engines. Use the MapR operational database and event streams to capture your data. Use the MapR analytics and machine learning engines to analyze your data in batch or interactively across any cloud, on-premise or at the edge.

MapR's technology is trusted by major industries like Audi, which uses MapR to accelerate deep learning in autonomous driving applications. MapR also powers Aadhaar, the world's largest biometric database, which adds 20 million biometrics per day.

To learn more about how MapR can solve problems for your enterprise, go to softwareengineeringdaily.com/mapr to find whitepapers, videos and ebooks. MapR can leverage the high volumes of data produced within your company, whether you're an oil company like Anadarko or a major fin-tech provider, like Cabbage, who uses MapR to automate loan risk and has done $3 billion of automated loans to date.

Go to softwareengineeringdaily.com/mapr to find out how MapR can help your business take full advantage of its data. Thanks to MapR for being a sponsor of Software Engineering Daily.

[END]