EPISODE 781

[INTRODUCTION]

[00:00:00] JM: Computational integrity is a property that is required for financial transactions on the internet. Computational integrity means that the output of a certain computation is correct. If I deposit money into my bank, my bank sends me a number that represents the account balance. I assume that the number they have sent me is correct, but the bank could be lying to me. Maybe this bank is not actually trustworthy, but I use a bank with a good reputation. If the bank stole money from its users, it would quickly go out of business. Therefore, I feel safe by trusting a bank with my money, because the bank needs to maintain its reputation.

The problem with a reputation-based system is that it's opaque. It's not easy for us to audit the bank and prove the bank actually has the money that it claims to have. Most of the time, the reputation-based systems that we use work fine, but occasionally we have catastrophic events. Think of the 2008 financial crisis or the Bernie Madoff financial scandal. The circumstances would've been avoided if the financial institutions could've been continuously audited for their solvency.

With blockchains and cryptocurrencies, we now have tools that allow us to maintain computational integrity without the opaque systems of reputation. We no longer have to trust a central authority. We can verify computational integrity with math.

Ellie Ben-Sasson is a cofounder and chief scientist at StarkWare Industries, a company that is bringing Zero Trust technology to market. Implementations of Zero Trust technology include ZK-STARKs, ZK-SNARKs and Bulletproofs. StarkWare is focused on the applications of ZK-STARKs, which can be used to improve scalability and privacy.

Ellie joins the show to discuss the topic of computational integrity and how Starks can be used to provide scalable, secure infrastructure to blockchain applications.

[SPONSOR MESSAGE]

[00:02:14] JM: HPE OneView is a foundation for building a software-defined data center. HPE OneView integrates compute, storage and networking resources across your data center and leverages a unified API to enable IT to manage infrastructure as code. Deploy infrastructure faster. Simplify lifecycle maintenance for your servers. Give IT the ability to deliver infrastructure to developers as a service, like the public cloud.

Go to softwareengineeringdaily.com/HPE to learn about how HPE OneView can improve your infrastructure operations. HPE OneView has easy integrations with Terraform, Kubernetes, Docker and more than 30 other infrastructure management tools. HPE OneView was recently named as CRN's Enterprise Software Product of the Year. To learn more about how HPE OneView can help you simplify your hybrid operations, go to softwareengineering daily.com/HPE to learn more and support Software Engineering Daily.

Thanks to HPE for being a sponsor of Software Engineering Daily. We appreciate the support.

[INTERVIEW]

[00:03:37] JM: Ellie Ben-Sasson, you are a cofounder and chief scientist of StarkWare industries. Welcome to Software Engineering Daily.

[00:03:43] EBB: Thank you for having me here, Jeffrey.

[00:03:45] JM: I want to start by talking about the topic of computational integrity. What is computation integrity?

[00:03:52] EBB: It is a desirable property that means that the computation being executed on your behalf is executed correctly. So if you have a program and it receives a certain input, you would like to be assured that the output was computed correctly. We think of this as a trivial matter, but it actually is extremely important in financial worlds or forensic applications and we sort of take it for granted, but it's really not.

[00:04:20] JM: One way that we can get computation integrity is from an authoritative source, like if we interact with a bank, we have some degree of compositional integrity. How does a bank provide computational integrity?

[00:04:32] EBB: It really is a trust assumption if you think about it. There is nothing on the technological level that prevents a bank from, say, seizing all of my funds and all of the funds on my accounts and transferring them someplace else. What backs computational integrity in a bank, first of all, are things like its reputations. So it has some incentive to operate with computational integrity. There are also auditors and regulators that look at it and all various bankers are signing with their name on statements that they are operating with integrity.

So there is a pretty big manual apparatus that is set in place to check and enforce computational integrity, but there is nothing at the system level that inherently prevents compromising integrity if those institutions choose so.

[00:05:26] JM: What's difference between having a system-level enforcement of a computational integrity versus a system where there are auditors and there is reputation?. This seems like its own sort of system for providing computational integrity.

[00:05:41] EBB: Well, people are prone to corruption. They could be bribed. They could look the other way, and these things have happened in the past. There is also a pretty large cost for maintaining all of this manual apparatus, the big buildings, the fancy suits, the people who are doing all of these checks. So you could get things to be a little bit more efficient if you didn't have this manual aspect to it. I guess those are two of the main problems with the way the system is right now.

[00:06:12] JM: We can potentially get computational integrity from a blockchain. How would a blockchain provide computational integrity?

[00:06:19] EBB: It does so pretty much by design. I mean, if you take Bitcoin as the most famous example, but other permissionless blockchains are the same. There really is no easy way for me to seize someone else's funds inside Bitcoin even if I want to, and there is no manual trust. There is no trust in humans to ascertain this or enforce this. It really is cryptographically hard. Stealing someone's funds is as hard as either breaking a hash function or a cryptographic digital signature, or the simpler thing that actually does happen sometimes is stealing their keys by other means. But the system is pretty robust by definition without any human intervention. That's pretty impressive.

[00:07:06] JM: So in the system of banks, I might have to get a bank audited in order to know that the bank actually has enough money on the books to be able to pay me back. There are similar kinds of relationships over a blockchain environment, where we have supposed guarantees about some other account having a certain amount of money in it. How does the world Blockchain computational integrity contrast with our normal banking world?

[00:07:35] EBB: That's a terrific question. So the conventional world delegates the process of accountability. We delegated to other humans, to experts, to auditors, the various accounting firms, and they come and they inspect stuff. They may get access to the computers or they see some reports and they check them. So it is a delegated form of accountability.

Bitcoin and the other permissionless blockchains have a very extreme different solution to this problem that I'd like to call inclusive accountability. It’s a little bit like direct democracy. In inclusive accountability, everyone is invited to participate in the process of checking and verifying computational integrity. So for instance if you're going up download a Bitcoin or Ethereum or Zcash client and run it is a full node, one of the very first things it will do will be to download the full blockchain from the Genesis block and verify each and every transaction, and thereby you will be serving as one more auditor in this big process of inclusive, inclusive accountability.

[00:08:50] JM: Let's reinforce why a blockchain is new. Why it provides something newer that we didn’t have before blockchain and cryptocurrency technology? What is so novel about how blockchain provides computational integrity?

[00:09:05] EBB: What's novel here is that there is a very ingenious game of incentives that invites everyone to sort of participate and check the integrity of a system, and it's a very open and transparent and inclusive framework where the trust basically lies on every and each of us participating in this process. That was very different from – If I want to go and audit my bank and see everything about it and check that everything is okay, I can’t, right? No one would allow this. There are actually laws probably that prevent this, and certainly I wouldn't be given access to this.

So it's a very different system in which sort of all of the human race is invited to participate and boost trust in the system by just running their computers to verify all the transactions that are going on there. It's pretty different.

[00:10:04] JM: Now, of course, with the auditor solution, getting our banks audited. There are plenty of problems. But if we want to have computational integrity insured on a blockchain, we need to replay all of the complications that have occurred across history. What are the problems with a blockchain’s solution to computational integrity?

[00:10:25] EBB: So to achieve this magnificent principle or ideal of inclusive accountability, the system has to pay in other ways. One of the ways in which it is paying is in limiting severely the rate of transactions, the throughput of the system, and this is set in place in order to allow everyone to participate in this inclusive accountability. But this leads to severe restrictions on the scalability of such systems.

Now in a sort of more delegated model, you could just increase the throughput by requesting the parties that are involved, let’s say the banks or the financial system to buy a faster computer or a bigger disk or have a higher bandwidth. But if you want to maintain inclusive accountability, you can't have that. You need to limit the throughput so that everyone can, on their laptops, verify integrity of everything that's happening.

The reason is that sort of everyone needs to check – I mean, the way the system is right now, everyone is checking each and every transaction a little bit similar to the way that we used to inspect restaurant bills and the days before computers, and when we got these slips of papers that owners sort of computed the total sum. Similar thing is going on in Bitcoin where our computers are checking each and every transaction in order to maintain this computational integrity. That's a pretty big price.

[00:11:56] JM: There are prices that we’re paying in terms of scalability, but also in terms of privacy, and I’d like to go through each of these. Can you start with the scalability issues of a blockchain?

[00:12:08] EBB: Yeah. So famously, Bitcoin I think was – It’s throughput was around 10 transactions per second, whereas let’s say Visa deals on average with 2,000 transactions per second and peaks at 25 or 50,000 transactions per second. So it’s like three orders of magnitude higher than that of Bitcoin.

Now, there was a huge battle in Bitcoin over this scalability issue. How do we increase the throughput? Whether you could double or multiply by eight the number of transactions, but it was clear that you can’t just multiply it by or increase by three orders of magnitude, not because there aren't strong enough computers that could check transactions at that rate, but because it would completely exclude a lot of the people that are currently helping to check all these transactions. They would sort of need to go and buy some very strong server in order to track what's going on in the system.

So scalability is a huge problem in permissionless blockchains because of the principle of inclusive accountability. If you want to allow everyone to check everything that's happening on their meager standard computer devices, you can't have enormous scale of transactions as you have in a dedicated financial payment system. So that's the problem of scalability on a permissionless blockchain with inclusive accountability.

[00:13:39] JM: There are also privacy issues, because we’re publishing all of our transactions on a blockchain, maybe they’re pseudonymous, but there is a degree to which we are exposed, and that’s problematic if we want to have privacy across our transactions. Explain the privacy issues of a blockchain in more detail.

[00:13:59] EBB: Yeah, that’s a great point, Jeffrey. I mean, if all transactions need to be verified by everyone, then it's like seeing all payments that are going on in the payment system. Everyone gets to see them. It's important that everyone sees them. So now if you don't take special precautions or build special systems like zero-knowledge proofs, then financial privacy will be lost. You can't really imagine a world where big business, let's say Boeing, can do all of its financial transactions on such a blockchain, because then its competitor, Airbus, would learn a whole lot about its financial dealings. This is something we can't allow, not because of criminal activity, but because of legitimate businesses that need to be able to compete and sort of keep their financial transactions private.

[00:14:51] JM: There are these entities that can be built on top of blockchain technology, for example, cryptocurrency exchanges. Maybe I want to start a cryptocurrency exchange, and I'm an honest guy, I want to have my customers feel confident that their funds are intact. But I don't want to make everybody's balance public. I could do that if I made everybody's balance public and I gave everybody public proof that everybody's balances were intact. That would ensure that I do have the funds intact and people would trust me. But then people wouldn't have privacy. So how can we solve this problem of the trade-off between the privacy and the security?

[00:15:31] EBB: So one solution is trusting the operator of the exchange, but this is sort of back to the conventional model that we're sort of trying to improve on. The other solution is to use modern cryptography, in particular things like zero-knowledge proofs that prove in a cryptographic way the computational integrity of the system, and the zero-knowledge aspect actually proves this without revealing any information about the state of the system. It just proves that it is being maintained with integrity. So you could use that as a different solution for this problem, or paradox of privacy along with integrity on a public blockchain.

[00:16:16] JM: A zero-knowledge proof is a type of interactive proof, and we've been using interactive proofs in computer science for a long time. These are useful for allowing multiple parties to come to an agreement on some shared truth. Can you go a few simple examples of how we use interactive proofs in software engineering?

[00:16:34] EBB: Well, one would say that a lot of the password protocols that we're all doing where your computer gets a certain challenge and responds, these are special cases of interactive proofs, various handshakes being done when you set up a new secure connection. Those use interactive protocols that involves some aspects of interactive proofs in some very simple aspects of them. But I guess those are pretty ubiquitous examples of the use of interaction and cryptography in secure network systems.

[00:17:11] JM: A zero-knowledge proof is this type of interactive proof where we are proving something without exposing more information than we need to. Can you describe a zero-knowledge proof in more detail and describe what a zero-knowledge proof enables?

[00:17:27] EBB: Yeah. So if you think of a proof, it's something – Some string of characters that's written or is in the interactive setting. It's sort of this transcript of questions and answers that all of them revolve around proving that a certain claim is true. So think a little bit of like an examination in the courts of law where claim is being made. Someone is saying some claim, like for instance, someone who is being prosecuted for theft might say as an alibi, "On the day of this the crime, of this theft, I was in a different country," and then there will be this process of examination with questions and answers, "Can you prove to us that you boarded a plane? What was the name of the city you visited on the day of that thing?" and their answer is being given.

So a zero-knowledge proof, which is in Stark contrast to the kind of examination that would go in a court of law, a zero-knowledge proof is one in which if you look at the transcript at the end and you ask yourself, "What have I learned?" The answer would be, "I gained zero-knowledge beyond knowing that the claim was correct," and it's a very magical and counterintuitive kind of notion, because in a court of law examination you would say, "Oh, if I listened to what's going on, I'm going to learn a whole lot more." For instance, I'm going to learn what was the name of the city this person visited, which hotel he stayed at?" Well, with a zero-knowledge proof, you would learn nothing about that. You would only learn that the person was not in the country or the city at the time of the purported crime.

[SPONSOR MESSAGE]

With Wix codes, built-in database and IDE, you've got one click deployment that instantly updates all the content on your site and everything is SEO friendly. What about security and hosting and maintenance? Wix has you covered, so you can spend more time focusing on yourself and your clients.

If you're not a developer, it's not a problem. There's plenty that you can do without writing a lot of code, although of course if you are a developer, then you can do much more. You can explore all the resources on the Wix Code's site to learn more about web development wherever you are in your developer career. You can discover video tutorials, articles, code snippets, API references and a lively forum where you can get advanced tips from Wix Code experts.

Check it out for yourself at wicks.com/sed. That's wix.com/sed. You can get 10% off your premium plan while developing a website quickly for the web. To get that 10% off the premium plan and support Software Engineering Daily, go to wix.com/sed and see what you can do with Wix Code today.

[INTERVIEW CONTINUED]

[00:21:14] JM: Let's take a simple example. Let's say that you, me and several other people, let's say three other people have each deposited money in a bank. So we all put in $100, and at some point in the future we all want to audit the bank, but we don't want to hire an auditor. We want to do this in a way that maintains computational integrity, but does not require an auditor. How can the bank prove to each of us that the bank has enough money to pay all of us back?

[00:21:41] EBB: So if the bank publishes, let's say, on a daily basis a hash or a commitment, a cryptographic commitment, to the set of all of its assets and it does so and let's say posts on some blockchain. So all you see is this hash that you know that came from the bank on a daily basis. Then later on – But this hash is supposed to be – If it's maintained with computational integrity, it's supposed to reflect the true state of the banks accounts on any given day.

So if later on, a month later, some people come to the bank and say, "Oh! We want the bank to prove to us that our funds were included a month ago in the accounting process." The bank could produce a zero-knowledge proof that says, "Person such and such, your funds," that had let’s say a thousand dollars,” were included in the hash from a month ago,” and when we computed all of our assets and liabilities and deposits and everything, we were in the black and we included your assets in the process.

Zero-knowledge proof would tell that person that her funds are included in the bank's reserves, but would leak no further information about the state of the banks accounts. That's the magic property of zero-knowledge proofs.

[00:22:58] JM: How do we know that the bank was telling the truth all along? Why couldn’t the bank be defrauding us by publishing false hashes on the blockchain?

[00:23:08] EBB: So in the crypto currency universe this would be much harder because the bank can't just claim to have funds that it doesn't control. Part of the proof would be it knows the secret keys that control the funds, but in the conventional system, yeah, the bank could have been cheating all along, but it does raise the bar a little bit, because the bank in advance, like on any given day, needs to already make up the sort of inconsistent statements that later on and commit to them on some blockchain or some public site, and later on be able to answer in zero-knowledge that that person’s balances are being accounted for as part of the general process. If everyone knows that this is the thing that's happening, we walk around with our little apps that are constantly monitoring the computational integrity of the whole big bank by, again, the process of inclusive accountability, but one that also maintains privacy.

So it's not like I would need on a daily basis to go and pick up the phone and ask the bank. Rather, my app would alert to me if didn't receive on an any given day a valid proof. So you could set it up to be some automatic process that constantly check and validates the computational integrity of the bank with privacy.

[00:24:34] JM: Now at this point, it should be intuitive to the listeners that zero-knowledge proofs are useful or privacy applications. Clearly, we've illustrated that we’re able to declare a solvency of the bank without revealing the balances of all of the users, but what about scalability? How does the fact that we have zero-knowledge technology improve scalability of blockchains?

[00:24:57] EBB: So certain zero-knowledge proofs, not all of them, have this magical scalability aspect, which means that if the computation for checking a certain statement of computational integrity, if the naïve way to check it by running the computation and replaying it takes T-steps, verifying a zero-knowledge proof could be exponentially faster.

So for instance, if running the naïve computation would take 1 billion time steps, verifying is your knowledge proof of the correctness of the statement could be exponentially smaller than that, and the logarithm of 1 billion is 30. So in something that is more like 30 times steps, you could verify the computational integrity of a statement or a correct execution of a program that naïvely would take a billion steps to finish.

[00:25:53] JM: How would we actually apply zero-knowledge proofs to improving the blockchain infrastructure? Do we have to update the core cryptocurrency protocols or how can we introduce this into our ecosystem?

[00:26:07] EBB: So it all depends if it's a layer one layer two solution. So, for instance, in Ethereum, you could envision a world, and that's actually something that StarkWare, that our company is going to be deploying soon. You could envision a smart contract that runs a zero-knowledge verifier for scalability, and that verifier could be checking that a huge batch of computations or transactions executed correctly. For that, you don't need any change to the underlying blockchain.

If you wanted to be what's called a layer one solution, something like plasma or some solution where at the very basic level of the blockchain it operates correctly, then you would need a change or a fork to the system.

[00:26:55] JM: So describe the state of zero-knowledge proof technology as it's deployed today. Where do we have zero-knowledge proofs in production?

[00:27:05] EBB: Well, I think the obvious and most famous place that has it in production is the Zcash cryptocurrency, where ZK-SNARKs are used to the shield transaction. So I'm also a co-founder – Sorry, a founding scientist of the Zcash company. I was one of the co-authors of the academic research behind it. I think that's probably the first real product in the wild of a full-scale universal kind of zero-knowledge systems out there. That’s Zcash.

[00:27:38] JM: There are several zero-knowledge data structures and algorithms, there's STARKs and SNARKs and Bulletproofs, and I don't want to break these down in detail on the podcast. I think of this podcast more of as just giving people kind of a hint at what is interesting about the technology here and some of the applications. If they are interested in the gratuitous details, they could certainly find plenty of online resources, including ones you’ve produced. But I would like to describe some of the tradeoffs at a high-level that these different zero-knowledge tools are making.

[00:28:09] EBB: Yeah. So at a very high-level, I think the tradeoff as go from SNARKs, to Bulletproofs, to STARKS, is as you go in that direction, you are increasing the communication complexity or the size of the zero-knowledge proof. But in return, you're removing some trust assumptions and some cryptographic assumptions.

So, for instance, I mean SNARKs have the shortest proofs, the ones that are in Zcash. They are roughly 200 bytes long, but it requires a trusted setup of toxic waste thing. So that SNARKs. Bulletproofs, their size are roughly in fact one order of magnitude longer. Let's say around 2 kilobytes instead of 200 bytes, but there is no trusted setup. Now, both of these systems are still prone to attacks by quantum computers and rely on number theoretic assumptions and their approving time is quite heavy. STARKs are plausibly post-quantum secure. Have the fastest provers and no trusted setup, but their proofs are in order of magnitude larger than Bulletproofs. So as you go from SNARK to Blueproofs to STARKs, you’re increasing communication complexity or proof size, but reducing the crypto assumptions and also making things a little bit faster.

[00:29:38] JM: What is that term trusted setup mean?

[00:29:41] EBB: So trusted setup means that there is a set of parameters. Think of them as some global set of keys that every user of the system must use, and these keys, actually, there is associated with them some sort of trapdoor and it requires great care to make sure that this trapdoor is not revealed to anyone as the keys are being generated. Zcash company orchestrated an amazing secure multiparty computation process for ensuring that this is the case, but it's a pretty high bar to stand by and everyone agrees that you'd be much better off if the systems you're using just simply do not require a trusted setup, as is the case with STARKs.

[00:30:33] JM: One characteristic of the cryptocurrency space that is so interesting to me is the fact that much of this technology in terms of its implementation is so unprecedented. So with Zcash, you had this unprecedented technology being brought to market. As you are observing the Zcash team bringing this – Maybe you'd call it product currency to market. What were your observations of that process? What was hard about it and what lessons about engineering cryptocurrencies did you learn?

[00:31:05] EBB: Oh! I mean, I have such a deep level of admiration not just for Zooko, the CEO; and Nathan, the CTO, but like all of the amazing engineers of the Zcash company, Daira and Sean and others there that did a marvelous job of releasing cutting-edge cryptographic technology under tight schedules in the most secure and yet transparent manner that can be done.

I think it's a lot about the ability to focus to deliver a very high-level of technical power and mathematical depth and ingenuity, and it's really amazing to see the level of expertise that is delivered by the Zcash company. It's really inspiring.

[00:31:57] JM: And you are a founder of StarkWare, and I want to talk about that in some detail. But did you learn anything from the commercialization strategy, the business strategy of Zcash and the Zcash company? Did that at all inform how you thought about company creation?

[00:32:14] EBB: Yeah. I mean, just getting to watch Zooko and the great team operate, I'm sure that it's sort of by diffusion I learned a lot of stuff, because after all, I'm a professor and a theoretician. So there was so much to learn that would be a little bit hard to put into words the exact take away messages that I learned from it.

A lot about defining what are the most important things for the company, and probably the most important thing is the people that you partner with and the way you partner with them. I mean, again, what I saw in Zcash and continue to see is really very inspiring the level of trust and leadership there. So I think that's what I learned from that experience.

[00:33:03] JM: What caused you to start a company?

[00:33:04] EBB: Well, in the case of STARKs, it was this sort of obvious next step. So the research behind STARKs is something that I've been passionate about since the 2001 when I started my postdoc at Harvard and MIT, and there's only so much you can do with stuff when you're inside the academic world in the amount of resources and focus that you can bring to sort of really bring it into life. So it's sort of was obvious that in order to see these things come to fruition, commercial efforts is really needed at some point.

[00:33:44] JM: Now, with StarkWare, we'll talk about what the company is building, but in terms of the problems that you're solving, the first problem that you're focused on is scalability. You want to move computations and storage off-chain. Explain what this means.

[00:34:00] EBB: So what it means is – So, remember, we talked a little bit about this property of inclusive accountability. So we want to allow everyone to keep tracking the computational integrity of the system without increasing our buying some big server. But at the same time we would like to scale up the throughput of the system hopefully exponentially.

Now Starks are a special kind of zero-knowledge technology in which you can do this, because if you have a single prover that processes a lot of transactions, that prover can prove to everyone else in the system that updates then to the system are correct, are done with computational integrity, and everyone else can verify that exponentially faster than the amount of throughput.

So what StarkWare is going to do is offer solutions in which off-chain, some big prover machine, generates proofs that process an exponential amount of transaction, and everyone else can check that the system is still operating with integrity by just verifying exponentially small proofs.

[00:35:08] JM: The first target application for StarkWare is decentralized exchanges. Can you explain what a decentralized exchange is?

[00:35:15] EBB: Yeah. So a decentralized exchange – I mean, that's the common term for these things, but more accurately they could be described as noncustodial exchanges. So the big exchanges, like NASDAQ, and whatnot, they don't really hold the custody of the various stocks. That's a different entity, a clearinghouse or things like that that holds it. The exchange only operates – Only as this marketplace where people trade and then settle elsewhere.

Now, today we have this anomaly in the cryptocurrency world where a lot of the biggest exchanges also need to hold custody of the funds that their customers are trading in. So that actually each trade, the counter party to it is the centralized exchange, and this causes – This is very costly to those businesses. It's also very dangerous. We hear from time to time about exchanges being hacked. I mean, there are these huge honeypots, and it's really an anomaly. In other markets you don't see this, but it's because the way you want to do things, which is the decentralized way or noncustodial way, is just impossible today because of scalability.

So what is the decentralized way? If I own an asset and you own a different asset and each one of us puts our order inside some order book, we would like the trade to occur without needing to transfer custody to the exchange, and that's what a decentralized exchange allows. It allows people to trade without transferring custody to the exchange, and the problem with these systems right now is just that they have very limited scalability because of the limited scalability of blockchains, and that's what we're going to solve. That's what StarkWare is going to solve.

[00:37:08] JM: Just to clarify what kind of exchange we're talking about here, we can kind of contrast it with something like Mt. Gox. So Mt. Gox was an exchange that lost customer funds, or they're unknown. Mt. Gox was not a decentralized custodial service. Could you contrast Mt. Gox with the decentralized exchange?

[00:37:28] EBB: Yeah. So Mt. Gox was holding custody or maintaining custody of the funds of its customers. So if you had an account at Mt. Got, let's say for 100 Bitcoin, so you didn't really hold those hundred Bitcoins. The keys to those hundred Bigcoins were held with Mt. Gox, and that's why when Mt. Gox got hacked, you would have lost your funds.

Now in a noncustodial or decentralized exchange, that's not the case. The exchange never maintains custody of your funds. You come there, you put an order on the order book, you sign it. If that order is filled, then sort of there is an atomic swap between the asset you’re trading and the other asset you want to get exchanged. But the decentralized exchange itself never holds custody of those funds. That's a very important property, because now you remove this security threat of someone hacking it and doing Mt. Gox on it.

[00:38:27] JM: So if we want to audit these decentralized cryptocurrency exchanges, what do we need to do?

[00:38:33] EBB: So first, you could use a zero-knowledge technology to audit those exchanges, but actually since those exchanges don't really hold custody of the funds, auditing them is not that important, because the customers are the ones holding the funds. So it's not as important to audit those exchanges as it is to audit a custodial exchange, because with Mt. Gox, you have to worry, do they really still hold my coins? But with a noncustodial exchange, I know that by definition it never holds my coins and it’s can't steal them. So there is less of a threat.

[00:39:10] JM: Indeed. There are scalability issues, though, to the decentralized exchange model. What are the scalability issues?

[00:39:18] EBB: Yeah, definitely. So the huge advantage of a centralized exchange is that most of the settlement doesn't happen on-chain. It all happens only in the books of the exchange, right? Let’s say Mt. Gox. But with a decentralized exchange, all trades must be settled immediately on the blockchain, and the blockchain has limited throughput. So because of that, you can't really reach large-scale, and then there is limited liquidity, and the exchange is not as interesting as one of those custodial exchanges. But we’re going to change that with Stark-Decks and allow much greater scalability on noncustodial exchanges.

[00:40:01] JM: Describe the application of Starks to this particular use case.

[00:40:07] EBB: So we’re building StarkDex Decks, which is a settlement engine at large-scale for Dexes. So how it’s going to work is – I mean, the way it's going to work is like this, a decentralized exchange will have an order book and people will put their orders and sign them, saying, “I am willing to trade today for this cost this certain asset,” and I sign it, but I'm not handing custody. I’m just putting an order. I’m just placing an order. It's a conditional statement.

Now, if there is a match for my order, someone else wants to sell what I want to buy, then the exchange will sort of match together the buys and sells, and then send a batch of these trades to the StarkDex settlement engine, and the StarkDex settlement engine will take a large batch, let's say of – I don’t know, 500 trades, and will generate a proof, a Stark proof, that all of these trades settled correctly. Meaning that the orders were matched and the various parameters that each buyer and seller wanted were okay. So the StarkDex proof will assert the computational integrity of the settlement process, but crucially you won't be sending those settlements on to the main chain. All of that will be done offchain. The only thing you'll be sending to the chain will be a proof of computational integrity.

[SPONSOR MESSAGE]

Get your free $100 credit at do.co/sedaily, and thanks to DigitalOcean for being a sponsor. The cofounder of DigitalOcean, Moisey Uretsky, was one of the first people I interviewed, and his interview was really inspirational for me. So I've always thought of DigitalOcean as a pretty inspirational company. So thank you, DigitalOcean.

[INTERVIEW CONTINUED]

[00:43:50] JM: Could you describe why this cannot be done on the main chain? Why you need to do this at the second layer?

[00:43:57] EBB: Yeah. So take Ethereum for instance. Ethereum has a guest cost limits of 8 million guests per block. Now settling a single trade is roughly 200,000 guests, and this means that you can at most settle 40 trades on a block, right? 8,000,000 divided 200,000. That should be 40 trades. You can't do anything more. There is linear cost. The number of trades times 200,000 guests, that must be less than 8 million.

Now with StarkDex, we already showed at the Stanford Blockchain Conference about the two weeks ago, we already showed in our demo that we can settle something that corresponds to a batch of 500 trades well below the guest cost, and this is because of the exponential speedup of Starks. So we're already one order of magnitude better than what you can settle directly on the chain. I think we'll even improve things further.

[00:45:00] JM: What kind of software do you have to build in order to implement this, and what's the deployment process for that software?

[00:45:07] EBB: Yeah, that's a great question. So we have all proof systems, including ours, has two parts. There is the prover and the verifier. The verifier is going to be sitting on the Ethereum blockchain. So it is going to be a smart contract written in solidity, and a little bit of Ethereum assembly. So it's going to be some Ethereum smart contract, and we're developing that, and a lot of effort is going into reducing the guest gas cost of this specific contract and also integrating it with the other parts of the decentralized exchange world, and we're working on this closely with the 0x team, Will Warren's team and closely with MCO, one of their core developers there in order to integrate StarkDex the 0x platform for decentralized exchanges.

Now the other part is the prover node. The prover node doesn't sit on-chain. So it's not being written in solidity. It's a very heavy piece of code. It’s a pretty heavy computation, and that part is being written mostly in C++, a little bit in assembly, but it doesn't sit on-chain. So it doesn't have to be written in solidity.

[00:46:25] JM: Does the prover node sit on Dex’s infrastructure, or does it sit on client devices? Where does that stuff sit exactly?

[00:46:34] EBB: The nice thing about using a Stark proof system is that from the point of view of the main chain and its computational integrity, it doesn't really matter where the prover node sits. I mean, it's a heavy computation, but as long as the verifier is happy with the proofs, it means that the prover operated correctly. So it could sit on the big Amazon machine. It could sit on some strong server. It could be distributed in a number of ways. I mean, it's definitely a heavy computation. So typically you won't run it for scaling. You won't run it on a smartphone, but could be running on a laptop, or a strong server, or on the cloud.

[00:47:14] JM: What's the state of deployment process? How much of this software have you built and tested in production?

[00:47:20] EBB: So none of it is production yet. So our plan is to put the smart contract and the prover alpha out there by the end of Q1, which is roughly in a little over one month. But we will put it first on Ethereum’s test net so it won't yet be a production. We’re in the final phases of finalizing the code, and then we want to leave a few weeks for testing and auditing and that sort of stuff before we put it on the test net. Then the we hope that within half a year we’ll be ready for moving it into the production if we don't see any bugs or issues. So that's roughly where we’re standing with the development of the code.

[00:48:06] JM: To drive home the value of this particular innovation, Starks applied to decentralized exchanges. What are some of the downstream impacts this could have? Whether it's to liquidity of the markets or how that liquidity in the markets would affect the broader blockchain ecosystem. What are your thoughts in the downstream impact?

[00:48:26] EBB: Yeah. So I think that we may be – I mean, I hope that's the case. But I think we may be someday in a situation where there isn't a whole lot of business being done in the custodial kind of exchanges, because those are extremely risky to the exchanges themselves and to the customers. So I think with our technology we'll help move the industry a little bit towards the more conventional model where clearing and settlement are not, and certainly maintaining custody are not part of what the exchange offers. This makes things much easier, first and foremost, for the businesses, but also for the customers. It makes things safer and better. So I think that if all goes well, I think that many and most of the large exchanges out there will be doing most of their business through decentralized and noncustodial exchanges, and we're eager to help them reach that state.

[00:49:26] JM: Why did you choose the application of Dex's as your first target application?

[00:49:31] EBB: I mean, we have this process where we looked at several different things that we could do to bring Starks into the world. It seemed to us that Dex was the best thing for several reasons. First of all, it's a layer two solution. So we don't need to wait for some hard fork or some internal blockchain politics to mature, but rather we can write a smart contract and be done with it. That's a huge advantage of the solution.

The second thing we wanted is we wanted to be able to operate in a space where there are other players there. So I mentioned 0x. There are whole bunch of exchanges out there that are operating very successful businesses. So we wanted to be able to work with these partners as much as we can and speed things up, and we also wanted a pretty big challenge, and we think X is the right first step. It's not the last, but it's a pretty good first step.

[00:50:27] JM: If all goes according to plan, do you have any – Or can you say what your next application might be? What makes sense to go to next?

[00:50:36] EBB: Well, we're considering all kinds of options. I mean, there are various kinds of things one might do that are related to Dex, like offering some forms of solutions for KYC and AML. KYC is know your customer, and AML is an anti-money laundering technologies. There is a possibility for proving all kinds of self-auditing things, like proofs of reserve and proofs of solvency that we discussed. I mean, a special case of – Or simpler case of decentralized exchanges, is that of operating only with a single currency, and that would be some sort of a payment solution. Things that are like lightning and things like that. There's always the option of adding privacy and zero-knowledge to these things that the markets desire.

So there are really large number of options that we’re exploring right now, but we haven't made any decisions and we really need first of all to get out alpha there and our our MVP hopefully half a year later, and then we can see where the market takes us.

[00:51:43] JM: We’re in a time where there is – At least from what I've seen, a growing number of people who believe that everything you could potentially do on Ethereum, you will someday be able to do on top of Bitcoin as a second layer solution, and this is the Bitcoin maximalism argument. Is that at all compelling to you as somebody who is working in the Ethereum ecosystem?

[00:52:07] EBB: One of the advantages of not having our own coin and not doing an ICO is that we are, first of all, a little bit agnostic to the question of, “Which is the true and right – Whatever, cryptocurrency?” and we would be eager to deploy our technology on any all blockchain that allows us to do so. If Bitcoin will be that one day, we will of course be eager and happy to work with whoever it is to make it happen. I mean, I hear things that say that Bitcoin is going always stay too conservative and has governance issues, and I hear statements like you said that the only thing that's going to remain is Bitcoin and it will adapt all other technologies. Making predictions is very hard, especially concerning the future. So at StarkWare we just want to deploy our technology on any serious blockchain that will allow us to do so.

[00:53:06] JM: Why is web assembly useful for blockchain applications?

[00:53:09] EBB: I think it's mainly a standard that there are going to be a whole lot of tools developed for it and it's going to be a little bit faster. So it's like the next thing after JavaScript and things like that, and it's always good to use things that have good development stacks and so on in our cross-platform and so on. So that's why I think web assembly could be really useful. It also is this sort of encapsulated a way that makes a little bit more secure to use also in the environment of a Blockchain. So I think those are the reasons.

[00:53:43] JM: To close off, you are professor for starting StarkWare, how does your work in academia teaching existentially? How does that compare to life as an entrepreneur?

[00:53:55] EBB: Oh! That's a great question. So I love academic work. I think it's really so important for humanity. Academics are like the space explorers, but the space we are exploring is the space of knowledge out there at the boundary of knowledge and looking into the abyss and finding or trying to find our way in the dark. That's what the experience of doing research in academia is. You're aiming for the most far-fetched ideas that you're not even sure could actually work and you certainly don't worry about whether it's practical or is it profitable. You only care about, is it novel? Is it beautiful? Might it fly?

Then startups or industries, they're sort of a mirror image of this. You're trying to find the fastest way to reach something that is sustainable, profitable, that is actually usable. It's a different experience. I would also say that work in academic research is a little bit more of lonely. I mean, you have your students and a few peers, and startups are much more of a teamwork, and so it's a different experience. But I think I was fortunate to enjoy both worlds. I really like them.

[00:55:14] JM: Ellie, it's been a pleasure talking to you, and I'm looking forward to the further developments to StarkWare.

[00:55:19] EBB: Thank you, Jeffrey, so much. I really enjoyed this conversation.

[END OF INTERVIEW]

[00:55:26] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use, and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plug-ins. Use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

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

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

[END]