EPISODE 546

[INTRODUCTION]

[0:00:00.3] JM: The DAO was a system of smart contracts on the Ethereum blockchain that investors put millions of dollars into. Back in May 2016, it was the largest crowd-funding event in history. We discussed it in detail in a previous episode with Matt Leising. The DAO was hacked due to a security vulnerability and this event led to a hard fork of Ethereum.

The DAO was organized by a company called Slock.it. Slock.it's original goal was to allow people to connect devices to the Ethereum blockchain. If you could connect smart locks and cars and electricity system to the blockchain, it could create decentralized systems for sharing these devices.

To raise money, Slock.it created the DAO. Although the initial scope of the DAO was to raise money for Slock.it, over time it expanded in scope to become a decentralized venture capital system. When the DAO was hacked, the events that followed shook the Ethereum community. The hard fork lowered the financial damage inflicted on the investors, but there were still outrage within the community of Ethereum. How was it possible for an open source, crowd-funding project to launch with a security vulnerability like this?

As the Ethereum world looked for someone to blame, they turned to Slock.it. Thus, began a very difficult period in the life of Christoph Jentzsch. Christoph is the CEO of Slock.it and he has been involved in the Ethereum community since the early days.

When people think of Slock.it, they might imagine a group of people that move fast and break things. In fact, Christoph's early work on Ethereum was around rigorous unit-testing of different Ethereum clients. Christoph was obsessed with testing and consistency between the different Ethereum interfaces.

In today's episode, Christoph and I talk about his early experiences with Ethereum, his reflections on the events of the DAO and the direction that Slock.it is going today. Since the

events of the DAO, the company has refocused its efforts on the original mission to connect devices to the Ethereum blockchain.

Meetups for Software Engineering Daily are being planned, and you can go to softwareengineeringdaily.com/meetup if you want to register. We've got upcoming ones in New York, Boston and LA. All the information is on softwareengineeringdaily.com/meetup. If you're looking for our old episodes of Software Engineering Daily, all 700 episodes are in our apps on the iOS and Android app store.

You can find categories of different episodes, like blockchains and business and distributed systems. We've got lots of other topics and a great search engine over all of our episodes. If you want to become a paid subscriber to Software Engineering Daily, you can hear all of our episodes without advertisements.

You could subscribe at softwaredaily.com and all of the code for our apps is open source. If you're looking for an open source community to be a part of, come check it out at github.com/softwareengineeringdaily. Thanks for being a listener and let's get on with this episode.

[SPONSOR MESSAGE]

[0:03:18.2] JM: This episode of Software Engineering Daily is sponsored by Datadog. With automated monitoring, distributed tracing and logging Datadog provides deep end-to-end visibility into the health and performance of modern applications. Build rich dashboards, set alerts to identify anomalies and collaborate with your team to troubleshoot and fix issues fast.

Try it yourself by starting a free 14-day trial today. Listeners of this podcast will also receive a free Datadog t-shirt at softwareengineeringdaily.com/datadog. That's softwareengineeringdaily.com/datadog.

[INTERVIEW]

[0:04:04.7] JM: Christoph Christoph Jentzsch is the CEO of Slock.it. Christoph, welcome to Software Engineering Daily.

[0:04:09.6] CJ: Thank you very much.

[0:04:10.5] JM: Today we're going to talk about your background in Ethereum and Slock.it itself. Let's start with that background. When did you start working on Ethereum?

[0:04:21.8] CJ: I started in summer 2014. That's when I – actually right after the crowd sale.

[0:04:28.8] JM: Okay. What were you doing in those early days? Were you just an investor, or did you get involved as a developer?

[0:04:34.0] CJ: No. I was a 100% developer. My background is I'm a theoretical physicist. That's what I started. I was doing my PhD at the university doing software freelancing to get some extra money. I was following Bitcoin since I think 2013, summer 2013 and in January 2014 I found out about Ethereum, read everything about it and then Gavin Wood gave a took where he certainly ended up looking for C++ developers in Berlin.

This is not too far for where I live. I applied. He said yes. Since then I was working more or less fulltime for Ethereum. I was focusing on the testing. You could say I was trying to prevent as many hard forks as possible when it comes to consensus between the different clients. My first task was to study the yellow paper, especially the Ethereum virtual machine, and then write – this was literally thousands of tests in the JSON format for every client to check if they are true to the protocol.

Every single thing you can do within the Ethereum virtual machine, I was writing EVM bytecode and then the Go client, the Python client and the C++ client and the Java client. Those were the clients which existed back then. They would run my test to see if they are in consensus. Sometimes one of them would fail a test and then I would have to explain to developers what I think they did wrong, because they got a different result than the other clients did get with same transaction.

I was just trying to find differences between those clients and also trying to help to make this specification, the yellow paper more clear, then I found something to be undefined, it needs to

define it. Does for simple things as dividing by serial, or just doing a division, site integers. You get a different result in Python than you get in C++ if you do site integer divisions.

Those are just such things you just cover in those test and then you just specify it in yellow paper. Once it's specified, you have clear test and it's fine. This was my main job and I work really fulltime developing and writing tests and finding box in the C++ line and also writing part of the C++line.

The last thing I did was writing the specification for [inaudible 0:06:50.9], the mining algorithm of C++. I didn't come up with the algorithm. This was not my job, but after it was done, specifying it in yellow paper and writing this last part, this one of my last jobs before I left in this locket.

[0:07:05.6] JM: You were doing QA testing on Ethereum, and I think that's amazing because people look at Ethereum and they're like, "Oh, it's this amazing world computer project." I think people probably don't think about as much of the fact that there are developers plugging away at it writing QA tests to verify that the Go client and the Python client are coming to the same conclusion.

[0:07:35.0] CJ: This was a lot of work. I mean, in the beginning they did not syncing at all. We have to figure out why they are not coming to the same conclusions. Always of a different hash of the state. It took a while until again syncing the same test chain, and then took a month until all passed all my tests.

Then I need to add of course new test, which we are covering some educators. I thought that some clients may be messing it up, because it was not really specified yellow paper, so it's very implementation-specific.

Then brought us also lot of fun, but I introduced fast testing to Ethereum, which is still due today and actually was – that was still finding stuff that we had about 100 course, the AWS note. I was more or less randomly creating certain types of transactions and throwing them at the Go client, C++ and Python client and also Java client and see if they always get the same result.

We found a lot of issues through fast testing. This is I'm saying which they have now refined and are doing even better, because in the beginning I was just – if you have completely random fast testing, you get almost no results. You need to specify to make the distance action, which makes some kind of sense. Then you start finding things and that's something that's a rated, but how to still find consensus issue today the test group of Ethereum by doing a lot of fast testing.
[0:08:57.7] JM: I know you mentioned the divide by zero, or the division error that result – you had a test where there was some division that resulted in different results from the Go client versus the Python client, I think you said, or C++ client.

What were some of the other reasons why you would have different results between – just to clarify for people, when we're talking about clients, what we're talking about is when people are using Ethereum nodes that are written in different languages, like these are – whether you're talking about a wallet, or you're talking about a mining, these are different types of nodes, but different nodes that are running software that interfaces with the Ethereum blockchain.

You want them to recognize transactions. When they process the transactions, the ordered set of transactions across the world's computer, which they all are processing the same set of transaction, so it's very important that they all come to the same conclusion. That's what you're talking about here.

[0:10:02.6] CJ: Exactly. It's different from Bitcoin. Bitcoin didn't have a written specification. They just had the implementation, which was the C++ client, but just a reference client. You have the specifications written in code, so to say. For Ethereum, one of the main accomplishments and you have to give a lot of credit to Gavin Wood here, who wrote a technical specification, which was yellow paper, which was extremely important to specify what Ethereum actually is.

Some of the things you can find just very simple; you think about it, but you don't think about it when you write code. It's one that they call stack depth. Meaning, if you call one function into another function into another function into function and so on, each machine has a different [inaudible 0:10:45.5] set as a maximum call stack depth.

You could write a smart contract, which calls another contract, which calls another contract and depending on your machine, depending on the compiler and different things, it would fail at a different level. Of course, this cannot be because they all need to fade at the same level. Because of this, we have to introduce the 1024 call stack depth limit.

This was one thing which we introduced after we found this issue. That an action just fail after calling into another function within 1024 times. This was one issue. The site integer division was one issue. Then we had one issue which actually led to a hard fork, then very early days of Ethereum. Meaning, summer 2015 was two weeks after the blockchain was launched, and Go client, C++ client and the Python client were alive.

Then we found the German time, it was in the evening about 8 or 9:00 there was a hard fork. Like the Go client and C++ client were not going on the same chain anymore. They're basic calling me, let's look at what's happening in an imagine group, in the Skype group. Then it was for me to find out, this is the transaction. Okay, now I look at the EVM code, I look at the trace, like what's happening in the EVM, this each action, what happens in the Go clients, what happens to C++ client?

Then we found the difference was there was part of memory set to a certain non-zero value. Then it was making a call which was supposed to write into this memory. One client it was writing – nothing was coming back, but one client was interpreting it as writing zeros. The other one is leaving it as is.

No data came back. C++ I think said, "Don't change it." Go said, "Change it to zero then." This was a difference. It took about two or three hours to find it. Then after we found it, I got a test for it. Then with the test, the government implementers could see, "Okay, I know what we are doing wrong. According to the protocol Go was wrong, C++ was right." Then they had to change it.

Although they haven't had the maturity of the chain, you could say well 90% of the network is running – well, actually 95% is running to Go client. Only 5% is C++ then. They could have just said, "Let's just fix the minority clients, because it really doesn't matter." It just specified in the specification a yellow paper and then fix the C++ client. They are very strict about this. We have

a specification. It doesn't matter if it's a maturity client or not, the one who makes a error or bug, he has to fix it.

We fixed the Go client. Back then we could just call the miner saying, "Well, there's a bug." They just updated in the couple like six or seven hours. Everything was fine again and people didn't even notice. This was like the first hard fork and people didn't pay any attention to it. Anybody in the news, nobody looked at it because it was fixed within hours, and you could still – the community was so small that you could communicate very quickly. This was one of those experiences where I was there to basically see what is going wrong, why do we have different results in different clients.

[0:13:51.0] JM: This community within Ethereum has changed a lot since 2014. Its grown. How have you seen the community change? What have been the notable changes since you joined the community?

[0:14:05.0] CJ: It changed a lot, I have to say. I mean in the beginning, we have visionaries who just want to build, develop 3.0. I was really attracted by Ethereum for the whole rap 3.0 mission at decentralized internet, especially the whole smart contract thing. You had those people who they did make a lot of money, but they didn't came for the money. They came for something – they take a huge risk, because nobody did really know if this would work out or not.

After it was launched, of course you have more speculators coming into the game and early adopters, people looking into this new technology. Back then and still today, but back then much, much more. We had this strict you could ideology of being 100% decentralized. There's no governance at all.

This is also rare the DAO was born out of this idea. If you look at the early Ethereum projects, like ogre. Ogre was one of the first ones. They also did a crowd sale, but their intent was always to build something where they could walk away afterwards with functions forever, like Bitcoin, like Ethereum. They intended the same for ogre.

This was the same soft behind the DAO. Maybe also they heavily evolved in. Then after this failed, the DAO project have had this long story, which is maybe something for another episode.

I could speak for hours about the whole DAO incident. Then people moved into ICOs which are much more centralized and out for security reasons that people said.

Well, even though purely decentralized governance doesn't really work. It's too dangerous. Let's just do an ICO and we just get all the money and then people hope for the best and we do something with it.

We lost a lot of this purely decentralized ideology of building something, which is there forever like a protocol. Something you build once and then you move away just is there. Just exists. There's something I missed a little bit, but I also see now in the last months or years coming back a lot of the governance questions and people try to build like what Vitalik called dye course, or color neo, Aragon. They are doing a lot of this direction of having governance on the blockchain and define governance.

You still have those people would dream of this new form of in society and decentralized governance. Just of course also have all those speculators which has try to make a quick buck. Also you now have, since – maybe it started a lot this Microsoft at DEFCON 1 I think it was, where they joined the Ethereum movement if you want.

That big industrial players and corporates taken interest in Ethereum. This of course meant a lot for Ethereum, because now much more money was flowing in, much more developers from large corporates that are interested in Ethereum. Just see today for example the Ethereum enterprise, alliance and others were focusing on use cases to be a decentralized stage and is maybe not the key issue, but a key point, but maybe more automation of business contracts and other things.

It was started more as a movement, ideology, an idea, and move more into a mutual technology, which can be used for different things, which is fine too. I still think the core Ethereum community exists and still very active. Just that the expectations have increased by a huge amount, and now other problems such as scalability issues and of course governance is still an issue to some extent.

Yes, we have seen some changes, but I wouldn't say that this is completely bad. It just have grown so big that now a lot of different groups and communities belong to the bigger Ethereum group that the corporates, the anarchists, the banks, the military. Just remember, one of first fund relation, it was in California. I think it was 2015 to a blockchain conference about certain people there.

You had those people there. Those [inaudible 0:18:02.2] with a laptop destroy the banks, the anarchy. Then you had people from the banks there and you have the investors there. It's some people from the military, because there's one thing about how can we use blockchain for military staff. Such a diverse group there and I was thinking about how great this technology is to create together this kind of diverse group, and how they all see a use case in it and think all those exist would just have expanded and growing by a lot.

[SPONSOR MESSAGE]

[0:18:34.6] JM: Software Engineering Daily is brought to you by ConsenSys. Do you think blockchain technology is only used for cryptocurrency? Think again. ConsenSys develops tools and infrastructure to enable a decentralized future built on Ethereum, the most advanced blockchain development platform.

ConsenSys has hundreds of web3 developers that are building decentralized applications, focusing on world-changing ideas like creating a system for self-sovereign identity, managing supply chains, developing a more efficient electricity provider and much more.

Listeners, why continue to build the internet of today when you can build the internet of the future on the blockchain? ConsenSys is actively hiring talented software developers to help build the decentralized web.

Learn more about consensus projects and open source jobs at consensys.net/sedaily. That's C-O-N-S-E-N-S-Y-S.net/sedaily. Consensys.net/sedaily. Thanks again, ConsenSys.

[INTERVIEW CONTINUED]

[0:19:50.6] JM: It sure is interesting that some of the people look at Ethereum and they see basically a way to have a shared database. They're excited about like, we can just have a shared database like you said. That's all it really is to them, but that notion is powerful enough that they want to use it. Then of course, there are the decentralized anarchist-type people who are still in the community who have much bigger ambitions, who want to have a decentralized autonomous Uber and a decentralized autonomous Airbnb, or a decentralized autonomous Amazon.

I think the technology for having a shared database that if somebody wants to verify their soybean supply chain, we pretty much have the tech for that today. I think the implementation can be carried out by various consulting groups and so on. I think that the more exciting technological developments to discuss are what we need to get to have these decentralized apps like the decentralized Uber, or the decentralized Airbnb, or decentralized Google.

What do we need in order to get to that world? Is it sharding? Is it something like plasma? What are the basic fundamental technological breakthroughs that Ethereum needs to make in order to have high throughput, mass adoption, everyday people using smart contracts and apps behind the scenes?

[0:21:30.4] CJ: Many things. Of course, you mentioned scalability. This is of course why DAO, with the bank problem in focus. I think you always focus other people about this and yes, plasma is a nice solution. I really love state channels and what [inaudible 0:21:44.6] is doing, also rating doing payment channels.

This is an interesting off-chain solutions, that I think off this is very, very important absolute key. Before we see any mass adoption, we need a huge progress on the side of scalability. Of course, sharding is also one solution. I think they will go hand in hand with each other depending on the application.

This is the one side of thing, getting the basic infrastructure ready. Now the other creation's a bit bigger. If you think about you want to have autonomous object and it's also now we're getting a little bit into Slock.it and to how can we move into this world. One main thing is Ethereum is also

a picky eye, so a public key infrastructure system. We need people and objects to manage their keys.

This is hard, because they need to store them securely, they need to feel responsible for them, they need to understand what it means to hold their private keys and risk those private keys. They interact with every system. If you get this, there will never ever be a lock-in into a system like Google, or Amazon, or whatever, servers you use.

You just have you key and you have multiple keys for different things you use. Those key can on a blockchain, especially public Ethereum chain interact with another thing on this level. You can interact with a machine, because the machine can also hold the key permissions to use something, can be stored in a smart contract. Because if you really think about just had the shared database, this is true, but blockchain is more than a shared database and I have really thought a lot about this in the last year when I thought applications and having tons of workshops about people saying what can we use blockchain for?

One thing what it really is, it manages permissions, especially smart contracts. They say who is allowed to do what? It can be financial transactions, like in Bitcoin it just who is allowed to spend how much Bitcoins. In Ethereum, it's also who is allowed to spend how much ether, under which conditions which are defined in a smart contract. We have pro credit money.

Even more important for us in Slock.it is you can say devices can now have something that ditch the cash, and they can say I manage my commissions on a smart contract and only the smart contract is telling me who is allowed to use me. This can be based on simple payment. It can be based on some keys having authority to say who is allowed to use me or use me directly. This is my highest authority of who can use me.

This could be a card, this could be an apartment with a smart door lock or other things. Then on the other side, the user side it just signed message which can come over any protocol, like Bluetooth, Wi-Fi, whatever. Then they just say, "I'm going to use you. Please give me access." They didn't look at the blockchain and see if he's allowed to do so or not.

In this world many up-checks as well as people or cupboards need to manage their own keys. I think public key infrastructure was a very good private key management. This would be key to adoption. Once we get there that everyone has his couple of keys in their phone knows about them, knows how to manage them and feels responsible for them, then they really this infrastructure.

The third thing is of course, you need to have a connectivity to the blockchain. Right now it just means wanting at least a light client and this is already – you can do this in a smart phone, but it really brings it to its limit when it comes to how much CPU power you're using, but especially bent with.

If you're running a light client constantly, you get so much bent with, it becomes unfeasible for IoT objects, or for cars or other things in this direction. We also need a solution how to securely connect devices to the blockchain. Not people just using my crypto, or my ether wallet or something just to have a browser, which is all really nice, but it would still rely on some node presented to it. It's not a direct connection to the blockchain, but people do it because it's such a burden to run their own client.

We need to improve light clients, or parity said thin clients and Slock.it is also working on a solution in this direction how to connect, how to securely connect IoT devices without any single point of failures. Also I think there's three main issues. To summarize, scalability for mass adoption, its managing of private keys and its connection to the blockchain and a secure and decent less way without needing a lot of interest and storage and computation to power.

[0:26:04.0] JM: You talked about some use cases there. The registering of internet of things objects, maybe connected bicycles or the lock to my home. Who has permission to unlock, like let's say we had a Biometric lock where you just put your thumbprint on my lock and you can open my door. Who has permission to enter my home?

Maybe if I am trying to run a decentralized autonomous Airbnb, it's very useful to have these biometric locks with a decentralized permission system. For most organizations, the systems that they would want to use manage permissioning and public key infrastructure, it seems like those things could be centralized. Why wouldn't they just use AWS or Google Cloud. Do you

have like a set of rules for what kinds of applications would make sense to port to decentralized infrastructure?

If we're really being serious here, what are the apps that people need decentralized infrastructure for, versus ones where yeah, it's exciting, but probably it just makes more sense to put things in AWS.

[0:27:18.8] CJ: Yeah, that's a very good point. The answer of course, first everything want to remove any single point of failure and also sensorship resistance if you need this. For many cases, the simplest stuff you just don't need sensorship resistance, or you also don't need to avoid a single point of failure. If you have those, then decentralization is basically a must.

On the other hand, sometimes if you'll ask me why just don't use a database by using a blockchain? I sometimes feel it – the certain application fits even easier to build on top of a blockchain, than to just use AWS. Because the blockchain automatically gives you a picky eye system, it gives you a securely managed backbone, which has zero downtime, which is always up, which doesn't require any logins and those kind of things.

Sometimes even faster to just use this without building your service system which you need to secure to a firewall and whatever means you need to protect the data. You say, "I don't need to protect the data because I don't have the data." Because people hold their own data. If you want to move into this world, then you have no logins, people hold their own data, then also a decentralized way of managing this is maybe important.

I also have there in my mind, I have moved also a lot from having everything decentralized, because I believe there are a lot of things that just doesn't make sense. In the IoT world, they have this problem of things should be there for 20, 30, or 40 years. If you buy a car, it should work without touching any of the inner workings for 20 years or so. Meaning, even if the manufacturer doesn't exist anymore.

If you buy a lightbulb and LEDs, technically there's no problem on lasting for 20 years or so. The company having produced it that's going to have a server infrastructure, which your lightbulb can always connect to and manage it for dozens of years. When you think about long-term

stability, you also would have something that you have no server infrastructure, or no single point of failure.

This is why I think for IoT applications that some that makes a lot of sense to use this – more like a protocol. I often say Ethereum is just a protocol, like e-mail, you have SMTP and you have other protocol. They just exist and they don't stop existing. As long as it's working, your device is working.

It becomes a part of the device itself to be able to do certain things. I sometimes compare to if you have a smart log and we will connect it to the blockchain, then you give the lock itself ability to be controlled, or used by the blockchain, or by a payment for example. Before when you buy a lock, you don't care if the manufacturer exists or not. No other key I can use, it was a key. It's a feature of the lock itself.

By connecting it to the blockchain and saying it's a feature of the lock itself that it can open it via a payment. If I do this with centralized cloud, there is not a feature of the lock itself, it's a feature which is given the lock by a third party, as long as a third party exists. It's a different value proposition.

If you want to have something where the feature is in the device, or is in the service, then you need to connect to something which more or less exist forever, or at least for a very long time.

[0:30:34.1] JM: You started Slock.it with a couple other co-founders. What was the original vision of Slock.it?

[0:30:41.9] CJ: To connect all smart devices to the blockchain and control it via a smart contract, in order to enable a decentralized sharing economy. This was now a long sentence. Meaning, if you want to rent out something, all you need to do is lock this thing as a smart device, could be a smart power lock, a small padlock, a small bike lock, a small door lock or whatever else, you can control in the smart way. Smart meaning just connected to that somehow. You just lock it and then you're done.

People could find it within the app, because it's registered on a public Ethereum chain. They can pay for access and when they are in front of it, they can also use the same app to access it. For example, you have a bike and you want to – you'll wait for six months for a trip. You want to make some money with it, all you would have to do is buy a Slock.it powered bike lock, lock it and then you're done.

People can use it for a while and then you come back, you look at your app and GPS, find your bike, you hope it's not broken and can take with you and you did make some money on the way. Allowing people to share almost everything, which you can somehow connect to the end of that.

This was what still is, that which an idea and we are working hard on making this possible. We have a mini label product out, maybe have not make – we did not make any buzz about it. You haven't enough – no marketing or anything like this, but if you go on the website mvp.slock.it you can register to become a test user and use the app already. It works right now for smart door locks and smart power blocks. Everything you can connect to it, you can register there. People can find it, book it, pay it and access it in the same app.

[0:32:19.3] JM: I want to talk about the internet of things developments that you're working on at Slock.it. I did want to talk a little bit about the DAO first, so before your current efforts at Slock.it, you were working on the DAO project. We did a whole show on the DAO, but I would love to get some reflections and conversations about that. Can you summarize what happened with the DAO?

[0:32:47.8] CJ: Summarize is good. It failed. After we started Slock.it in November 2015, we thought about how to fund it. Well, initial idea was doing some token sale or ICO, retrospect they're not as popular as it is today. We would be like the second or third company doing something like that.

When I programmed the ICO contract, I said, "Well, why not give you token holders more power." I said, "Let's have a game and vote how we use the money." Then let's say, "Let's give them even more power." They can keep the money. We asked for it. If they vote for it to get it. Then I said, "Well, let's open it for everyone so everybody can ask for money and get it based on the token holders saying it." This was the origin of the DAO.

I wrote a smart contract and then we start basically released version 1.0. Now everybody could deploy it to the blockchain or use it. This is of course now a long story made very short. After this, you know the story about 150 million dollar went into the DAO smart contract, which I didn't feel very comfortable about, because it was not this – I could say, it was not designed to hold so much value, but we had back then about 5 or 6,000 Slack users.

If each of them would have given a $1,000, we would have ended up as 5 or 6 million or something this range. We certainly did not expect anything like this. I mean, we did the best we could when it comes to security. I mean, [inaudible 0:34:08.4] didn't even have compile of warnings. There was no depact and there was nothing like this.

It was very, very early and you had – maybe could advise us. All the main Ethereum founders, or inventors I showed the code, they had to look at it. We gave it to a security audit company. We felt good with it. Of course, there was a bug called the re-entrancy bug, which allowed a hacker to restore about 50 million dollar into a so-called split down, so into another account, you could say where the money was locked for more than five weeks.

This five weeks now was time to do something, but then it was a really decentralized, autonomous organization, nobody had any control over it. You could just look and then we said, "Well, let's try for a soft fork. Let the miners at least buy us some time by sensoring all the transactions which would go to the DAO." Meaning stop it from operating, so nobody can restore any money and we can think about what to do with it.

It turned out this would open up for DOS exploit. The software was canceled and then there was only one solution left, which was a hard fork. Here there was a vote going on in the blockchain, but even there I think about 5% of those who had ether voted, so participation could be much higher, but wasn't.

They voted, like 80% or 90% in favor of a hard fork. The mining pools had to turn a vote and also there you have about 80% to 90% of voting in favor of a hard fork. Every sign we could have from the community, even on social media, every votage was done, was shown into a

favor of the hard fork, that's why the developers of those clients have implemented it, even they gave it as an option for users to choose in favor or against the DAO hard fork.

The real road was people installing or updating their clients, because that's the only way you could do a hard fork. People who were doing nothing, but be against the hard fork. Only those who actively updated their clients or installed it said some parameters on the command line, they would run the client, which would run on the hard fork Ethereum chain.

The hard fork was basically trusted rule that we created a new smart contract called restored DAO contract. All the money which had something to do with the DAO was going into and people could exchange the DAO tokens for the ether. It was basically giving people the money back. There was still some edge cases where a group that look at explicitly, which was a bit more difficult, so thanks also for the wide head hackers who was very active and generous back in this time, or still are.

This was in short the story of the DAO. It as a failed experiment in some sense technically, because it failed because of the bug in the smart contract. It should've been a really interesting experience in decentralized governance, because it did hold about 12 million ether. Don't go into think about how much value of this is today, but with this – the thinking of it was being something like a decentralized company.
Where it would say, it would order Slock.it to build those smart locks, but every time they're going to be used, that would be a profit for the DAO. Then it would ask a marketing company to do a marketing photo Slocks. It would ask lawyers to look at the legal registration of some bed and even some [inaudible 0:37:18.2] loyal, so you're going to have a representation in Switzerland for a DAO.

That was people professional investors which look at each proposal and make a report for it. It was basically a group of people would decide who to give money to for certain things. When I started it, I thought of it as a decentralized sharing economy company. After there so much money went into it, people thought of it as a decentralized investment fund.

This was just how they looked at it. I myself didn't comment to it at all, so I kept myself quiet because I said I do not want to define what this thing is. It's why the name the DAO, nobody

gave the thing a name. I was asked for name and I said, “Well, the DAO has to vote for its own name. I cannot say how this company should be named, because I just wrote a smart contract and I want to be –” I’m more or less the lawyer who prepared the contract which other people signed.

You had about 50 to 20,000 founders creating a new company and then operating. If they give the money to charity, then it’s a charity DAO. If they give the money to companies, it looks like a fund. If they use it for buying products and services which they would market and sell it to someone else, then it’s more like a company.

Technically it was just like a big joint bank account where many, many people had joined access and could only spend it together. The rest is just history how people perceived it. Of course, it was a very challenging situation for the Ethereum community when it comes to governance. For the first time, we had a technical hard fork, but a political hard fork you could say.

It was good that you had this challenge early on and dealt with it. We’ve had the consequences, we learned how a political hard fork turns out, that you cannot kill the old chain. It will always exist, which people didn’t really realize after it really happened. Of course, you could have thought about this before and just people said it will happen.

Many didn’t believe it would happen, so it was a big, big learning experience when it comes to smart contract security and it was a personal learning experience about how not to do things and how to do things. It was still a great experience after all. Although, a experience which it hurt a lot, but sometimes pain is the best way to grow.

[0:39:27.3] JM: I’ll agree with that. It must be interesting, you started working on Ethereum doing this rigorous QA testing. I’m sure it’s in your DNA to be very sure that something functions before you put a lot of momentum and money behind it. It doesn’t surprise me that you had some uncertainty when you were starting to roll out the DAO perhaps before it had been thoroughly tested.

Do you have any ideas for how you would test the smart contracts, or how you would build a smart contract testing pipeline, if you could go back and you had to actually do the DAO in a way that would make sense?

[0:40:13.3] CJ: It's a very hard question. The thing is, you cannot really note the unknown unknowns. Meaning, it's a bit of an irony that it happened to me as the one who being the main leader of testing in Ethereum. On the other hand, I really can say if we'd have test it two or three times longer than we did, it wouldn't have changed anything. We wouldn't have found it, simply because we are not aware of it.

I have looked at this conflict, which is about 600 lines of code. For soft developer this is really nothing. I have looked at it times and times and times again. Other people have looked at it and we had experts. As I said, we had create devices that the ones who created solidity, they are the ones who looked at our smart contract.

I'm not playing them here. I mean, they have not paid for it. It just looked at – they did the favor and looking at the contact for me. You paid one company who looked professionally at it. I was with them and said, "If you find something, just tell me." I didn't really find anything serious.

Looking back, although it was this big failure, I really didn't – don't know what I should've changed, except of putting in a cap or saying start an experiment slowly and saying, "Well, let's just cap it with 1 million or let's say a 100,000 to see if people can hack it." On the other side, you should have seen me doing this type – I was not comfortable with this whole iteration and it felt like the community was right behind me and say code test them, release –

You had this community of about 6,000 people who just wanted the thing to start. We have basically postponed it and postponed it and postponed it, and only because I said I'm not ready, I'm not ready, I'm not ready. Everybody else was saying the thing is ready since a month. You didn't change anything. Just started.

It was a difficult iteration. Looking back at it, of course testing it was a cap. This would have been I think the really only solution, but on the other hand people said if you put in the cap that there is just one guy buying 1 million into it and feel not be decentralized.

Today, the arguments are a bit different. Back then, people feared mostly for decentralization. If he said for example, we want to keep 5% and do something like security audit, checks and stuff, they screamed online at us. You keep something. No, that is not autonomous again. That's not decentralized.

A cap was this cost in the Slack channel, but everybody was against it, or most of the people were against it simply because people are fearing of, "No, this is not decentralized. It's complicated to get everything. No." I wanted to keep it as pure as possible, and to do a pure DAO this was the only way to go, and the risk was high, that's why I didn't feel comfortable with it, but I did let people influence me and convince of releasing.

This was maybe my mistake to not be stronger in saying no. After I released it, I just went on vacation and I was physically, mentally so stressed. I just couldn't look at Slack online, Reddit, social media, anything. I was just quiet.

Then when the DAO hack happened, I just had my hardest time of my life, I would say, but I'm very thankful. Even now, maybe of course Ethereum community is listening. Thank you again Ethereum community for basically saving the DAO by giving those people their money back, which was something I could not have done alone, which was their decision.

It was this experience I learned from, we at Slock.it learned from, the community learned, everybody learned from. Now nobody really has done anything like this again. There are people calling and save DAOs, but when I say it's a DAO it means – if there's anything I can remove and it doesn't work anymore, then it's not a DAO. This is something I have not seen until now.

[SPONSOR MESSAGE]

[0:43:56.2] JM: If you love Software Engineering Daily, I think you'll also love the Google Cloud Platform Podcast. It's a podcast about Google Cloud Products, how they're built and how you can use them. Really it's all about the changes that are going on in software engineering, as told from the point of view of Google engineers.

You can find the Google Cloud Platform podcast at gcppodcast.com. You'll hear from Googlers like Vint Cerf and all about Tensorflow and Firebase and BigQuery, from the high-level use cases to the low-level implementation. The GCP Podcast has all of that covered.

Find the Google Cloud Platform Podcast at gcppodcast.com and subscribe to it wherever you get your podcasts.

[INTERVIEW CONTINUED]

[0:44:47.5] JM: This show is mostly about software engineering, but I've talked to a lot of different people who have started businesses that have had really bad problems at certain points and they have really controversial problems that come up. Oftentimes, it results in the founder where the creator of the project going through a time of incredible pain.

Obviously, you grow because of that, but did you have any lessons or reflections from going through that super difficult time? Because I'm sure there is somebody out there that's listening right now that is going through something similar, probably not to the same financial scale, but it will probably seem to them personally like they're going through something that is of that magnitude.

[0:45:36.9] CJ: Now it's not technique, now it's personal. Very personal here. I myself – this is of course not true for everybody, but I had two people in my life who I can always trust and which helped me through this time. One is God, I have a strong faith in him and he helped me through this time and another one is my wife.
They helped me mentally and physically to go through this time. Then of course, it comes now to again – to development under the company level. We have found a team. We have three people helping each other and that's of course important that you trust each other. If you have a clear vision, it just continue this persistency, hard work and just move on.

Not just moving on without taking lessons with you, but the good thing is every bad time has an end. Even the DAO now, as hard as it was, during the time of things like, no this is written in the blockchain forever. This would be with me forever and it could destroy Ethereum. This is so big

and this is change everything. Looking now back at it, it's now more than two years, you see that as well it happened, but not almost two years.

Time heals wounds. You move on and you take lessons with you such as taking care of security and having – doing things step-by-step. If ever I do something like this again, I would say moving from centralization to decentralization step by step is a wise idea. Don't forget why you are doing it. You need a very good for the why or to move on. It cannot just be money, because this will not hold long enough. There are easier ways to make money.

[0:47:09.8] JM: I hear you. After those events, you eventually pivoted the business to the Slock.it discussion that we had a little bit earlier about connected devices and IoT. Can you describe how the business runs today, like what your focus is on, because I know you're building this IoT ecosystem. You're also doing some partnerships with various corporations. What are the steps that you're taking to accomplish those goals of making that IoT ecosystem?

[0:47:44.0] CJ: Yeah. Basically we've been back to our roots, like Slock.it started as a company connecting IoT devices blockchain. Was a bit distracted by the DAO, and now we are back to our roots. You have about whole of the company doing projects with large coverage.

We have connect their devices or things to the blockchain to enable the use case, or business case. We're working with seamen, we're working with some companies in the automotive industry. This is one part of a business and it pays to build you could say, but it also it's a lot of fun to do in those proved concepts and having those big coverage being very innovative and saying how do we want to use the blockchain.

The other part of the company is saying, we want to be – enable a shared network. If you connect those things to the blockchain, this enables them to rent themselves out. They can ask for payment to be used. We have our app. We want to enable to sharing that form, but we also have to say we are not – we haven't started doing any marketing yet, so I would say as of today, we are focusing more on this – could say IoT gateway. I don't know what is a good comparison.

Maybe it's between the stripe, a payment platform, a payment provided for platforms, which connects ratcheted users to other ratcheted users. We are connecting humans to machines,

machines to humans and in the end, machines to machines. We want to help companies and also private people of course to connect their device to the blockchain. This is the core mission and we are doing a lot of different things for it.

We are working the smart home gateways to get integration there. You're also trying to get – or we are getting cloud integrations for smart lock manufacturers to control their locks by the cloud, which is of course not decentralized. It still gets us closer to a vision of controlling devices via the blockchain, which can also still happen in the cloud.

Everything which increases the number of devices, which are controlled by the blockchain makes us more successful, having us more devices on our platform and being to service to corporates and private people.

[0:49:44.0] JM: Today, you can do that to some degree on Ethereum. The scalability bottlenecks that we talked about earlier, does that inhibit the ability to build IoT connectivity on the Ethereum blockchain?

[0:50:03.4] CJ: Yes, it does. Right now via on the coven test net and the main chat would just be too expensive for a use case. Despite in the beginning, we just build all of these big smart contract system which manages the permissions of devices and booking, renting, selling, everything.

Now we say, it cannot reason to be release this beast to the public chain, because it's just faster work because of scalability. Despite looking into state channels heavily, but also into the quick solutions of course doing a proof of authority chain is comparative reach to the main chain.

There are of course many reasons why I don't like the solution, because the inter-operability between the application is just not given. You create your own little mini ecosystem by having a private chain, and just to preach just gives you control of tokens, or ether from the main chain into your own proof of authority chain.

That doesn't give you any inter-operability between the different applications on the public Ethereum chain. It is a quick solutions to get going, but it's not very one to end. Really hopeful

of plasma and sharding to take off. That's my current hope. If it doesn't come, then this would maybe just build tons of private chains and try to connect into each other, what [inaudible 0:51:16.1] is also trying to do. By [inaudible 0:51:18.5], there would be a solution for connecting those chains, which is also okay.

Because of those scalability reasons, we have not put a lot of money into marketing because it wouldn't make sense to scale up the business when the infrastructure is not ready for it. That's why we are working more on getting technically ready for this, building – also we are working on having very, very thin clients on simple IoT devices, they cannot even run a light client on them.

If working on a protocol and the paper which we come out soon, so we are doing more theoretical and infrastructure work and enabling the business case theoretically, but have not taken off from the business side of things, except of our consulting projects where we just get paid for. That's where we stand right now. If you want to test it, you can go to mvp.slock.it and register as a tester, download the app and see what you can do with it already.

[0:52:07.2] JM: I think the – is the protocol that you're referring to the universal sharing network?

[0:52:13.3] CJ: That's how we framed this. Universal shared network means a network of devices that you can pay to access, using the blockchain as a backend. That's what this is. This is a lot of different things and that's why it's called universal. I would say our app is in user interface to this universal sharing network.

[0:52:33.4] JM: Got it. Another project that you've worked on with Slock.it is share and charge. Can you talk about your experience with that project?

[0:52:42.0] CJ: Yes, this was a company called energy coming to us asking for technical help. We help them to implement a use case for people to share their charging stations. If you buy electric car, usually you'll also buy a charging station, which is not used most of the day. We connected those charging station to the blockchain, and the credit it was in their user interface.

Their people could say, I see all those private charging station now and could go to them. Those people who own them can make some money with it. It was also a project that you connected those devices, in this case charging stations with the blockchain and then also connecting it to the user interface they're having.

It's also very interesting project and it was not our project. We did it for someone else, but was still very fun to work with this team and seeing objects like cars, using this public Ethereum chain in real life.

I think one of the first application by a large correlation was giving their assets, or offering their assets on the public Ethereum chain, had normal people using it without even knowing that there's a blockchain behind. This was very interesting and a good experience.

[0:53:52.0] JM: When you're putting cars or charging stations on the blockchain, do you have to create custom hardware for them to integrate with their car or their charging station?

[0:54:03.9] CJ: You don't have to, but it would like to. If you have built a [inaudible 0:54:07.2] solution, but you also have built an Arctic and Samsung Arctic is another IoT platform solution, that I believe you integrate this within the device now running your own light client. This of course is the best way of doing this.

This is just not feasible. That's by there using remote clients or a cloud API, which of course means the device itself doesn't get any added security. It just have a way of controlling a device over the blockchain, buta cloud reads the data and then the cloud sends the information to device to switch on/off.

To get going quickly is okay to maybe go with a remote client connecting to the cloud, but for real security we need a better solution. That's what we are working on actually right now is in the company to releasing a protocol. How to cheaply without synchronization time and a lot of bent with, how to connect an IoT device to the blockchain. I just can give you this teaser basically, but we will see it when it's online on a blogpost.

[0:55:04.7] JM: Okay. What are the other projects in the Ethereum space that you're excited about right now?

[0:55:10.0] CJ: I'm excited about all those governance projects such as Aragon and Colony. I think that's a very good use case for Ethereum doing governance on it. I'm very excited on those state channel which is done by Elfor and by the raiding team. This is exciting infrastructure, which is being built right now.

Less excited about ICOs, but that's clear. Those are the things that I think Ethereum will have a huge impact on governance and my field of course IoT payments. I do not see as a currency, we have normal average people that use it maybe, but I'm not – that's not the intend of Ethereum at least, but those are the applications at least which interest me the most right now.

[0:55:53.0] JM: Okay. Well, Christoph I want to thank you for coming on Software Engineering Daily. It's been great talking to you about Slock.it and the DAO and of course, your reflections on those events is much appreciated.

[0:56:03.4] CJ: Thank you very much for having me.

[END OF INTERVIEW]

[0:56:07.7] JM: We are running an experiment to find out if Software Engineering Daily listeners are above average engineers. At triplebyte.com/sedaily you can take a quiz to help us gather data. I took the quiz and it covered a wide range of topics, general programming ability, a little security, a little system design. It was a nice short test to measure how my practical engineering skills have changed since I started this podcast.

I will admit, although I've gotten better at talking about software engineering, I have definitely gotten worse at actually writing code and doing software engineering myself. If you want to check out that quiz yourself and help us gather data, you can take that quiz at triplybyte.com/sedaily. In a few weeks we're going to take a look at the results and we're going to find out if SE Daily listeners are above average.

If you're looking for a job, Triplebyte is a great place to start your search, fast-tracking you at hundreds of top tech companies. Triplebyte takes engineers seriously and does not waste their time. I recommend checking it out at triplebyte.com/sedaily. That's tripleB-Y-T-E.com/sedaily.

Thank you Triplebyte for being a sponsor.

[END]