EPISODE 544

[INTRODUCTION]

[0:00:00.3] JM: To use a web application, you probably open a web browser or a mobile app. To access an Ethereum application, many people use an Ethereum browser. In previous episodes we explored MetaMask and Mist, which are Ethereum browsers for the desktop. In today's episode we explore Status; a mobile Ethereum browser. Status founders Jarrad Hope and Oskar Thoren join the show to talk about the engineering of Status. How Status connects to the Ethereum blockchain, what people want from Ethereum applications, and the engineering of the Status app itself.

Status is built using React Native, which his working out quite well for them. We also talk about some of the mechanics of an ICO. Status has raised $100 million in their ICO for the Status Network Token. An ICO differs from racing equity in several ways. Rather than representing a direct stake in the business, a token represents a stake in the ecosystem that is being built. Through their ICO, Status raised much more than a startup at a similar stage in a company's development would have raised and the vesting schedule for the founders is two years. After two years, their stake will be liquid. This illustrates another way that an ICO can contrast with a traditional startup equity offering.

In a traditional startup, there is not a liquid open market for equity prior to the company going public. This can be good as it forces the founders to maintain their skin in the game until they have proven the business. But forcing owners to have the equity be liquid can also be bad if the founders are in a position where their equity is illiquid, then they may have too much pressure. They should be able to take money off the table arguably if the business model is doing okay, but maybe the economics have incompletely worked out so the company can't go public. So there are pros and cons to the classic equity model versus the ICO model, the ICO model being highly liquid, the classic equity model being less liquid.

In the interview, Jarrad explained that he anticipates the open source community around Status to be contributing more to the Status app development overtime, because the community has a

stake in the app by purchasing the Status token. I hope this is the case, because it would be very cool to see more consumer-facing open source applications.

Status is a consumer-facing app, and this conversation made me think that it is kind of strange that there's so much open source software for building applications, if you think about application components like React Native, Kubernetes, Kafka, but there are much fewer consumer-facing open source apps. There's not really an open source Uber, or an open source Facebook, of an open source Google, and it's worth wondering, "Why is that?"

Maybe that's because we are still in the days where somebody has to be paying for the backend compute layer. There needs to be today a centralized actor that is paying for the hosting of Google, for example. The open source code itself is cheap to host today, which his great, but if you want to run the actual application infrastructure, it still requires that owner to pay for it. Kind of makes sense that consumer applications are still developed and maintained by central actors.

With Ethereum, that might change. We might see more consumer-facing open source decentralized applications, so it would be quite cool and that is certainly the world that Status.im is hoping for. It's what they're working towards. It's what we discuss in today's episode.

Speaking of consumer-facing open source applications, you can check out the Software Engineering Daily app on iOS or android. We've got all 700 episodes of the show that are in the app. We've got tons of episodes on blockchains, business, distributed systems, tons of other topics. If you want to become a paid subscriber to Software Engineering Daily, you can hear all of our episodes without ads. So you can subscribe at softwaredaily.com. Again, all of that code is open source. So if you're looking an open source community to be a part of, come check out github.com/softwareengineeringdaily.

Let's get on with the episode.

[SPONSOR MESSAGE]

[0:04:42.5] 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. So, 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 Consensys 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]

[0:05:59.1] JM: Jarrad Hope and Oskar Thoren are developers of Status.im. Guys, welcome to Software Engineering Daily.

[0:06:04.8] JH: Thank you so much for having us.

[0:06:06.0] JM: Today we're talking about Status.im, which is an Ethereum client for mobile phones that you both work on. Before we talk about mobile clients specifically, what is an Ethereum client?

[0:06:19.8] JH: Yeah, good question. I guess generally speaking an Ethereum client is anything that is able to interface with the Ethereum network and it probably has some degreeing of user interface as well. However, it's very broad and the manner of which they connect to the Ethereum network and the manner of which they present themselves to humans can be greatly varied.

[0:06:39.7] JM: We have mobile apps, we have desktop apps that do not interface with the Ethereum blockchain, like Slack, for example. Slack on the desktop is pretty much the same as Slack on mobile, but it's a smaller interface. Are we interacting with different applications from the mobile client versus the desktop client, or is it pretty much the same?

[0:07:05.8] JH: When we're talking about a client, I think it depends on what context we're talking about it in. So, for example, we have the sort of full implementations of Ethereum such as like Go Ethereum, or Parity, or the C++ version, or even the Java implementation, and at this level, really, they're kind of like a library that just connects to a peer-to-peer network.

On the other sort of spectrum, you have much more sort of user-facing interfaces such as desktop applications, like Mist or browser plugins like MetaMask, or even mobile applications, like Status itself. But then some of these actually connect via sort of HTTP gateways, which then connect to, say, full implementations. We decided to make this in very early design decisions to basically do the best of both worlds in both creating a very easy to use user interface, but at the same time running a full implementation of Ethereum directly on your mobile device.

[0:08:02.4] JM: Go into that in a little more detail. What does it mean to run a full implementation on your mobile device?

[0:08:06.8] JH: Sure. So one of the large problems that sort of blockchains and decentralized technologies are trying to solve is this concept of like a disintermediation. Currently, everything that you do on the internet tends to go through like a server or some kind of gatekeeping service. When this happens, there's some problems that are involved and it mostly revolves around trust. However, with the advent of public blockchains, we can actually create essentially trustless systems.

[0:08:38.3] OT: So in Ethereum network, you have multiple nodes, right? Then they can have varying capabilities, but they're all equal nodes and sometimes you have full nodes, you have nodes that do mining, you have light nodes and so on. But it's just a matter of sort of how much you need to verify. So with the light nodes, for example, you don't have the entire blockchain history, but you still have sort of — It can verify the headers, so you still know you have a

consistent state. We're running sort of a light node on a mobile device, which means that there's no difference in the trust model. Maybe this is a lack of some information, like historical transactions, for example. So we're running one of those on the device.

[0:09:14.0] JM: Right. You run a light client in contrast to a system that might use a remote client, and I think one example of that — I think MetaMask uses kind of a remote client. Is that right?

[0:09:29.5] OT: Yes, that's correct.

[0:09:30.7] JM: Okay. Instead of just keeping a light node on the user's computer, the user's device, MetaMask uses this remote node, and what's the difference? Why doesn't MetaMask use a light client?

[0:09:44.9] OT: So MetaMask, it's different, because you're still using these HTTP interfaces where you're trusting some central server to provide you or some kind of state. Whereas if you're running an actual node, it's a separate process and you're sort of verifying the [inaudible 0:09:57.7] directly. So it's a slight difference in terms of trust model. Jarrad, do you have anything to add to that?

[0:10:03.8] JH: Yeah. I mean, definitely different in terms of this trust model. I imagine it has some historical roots as well. Back when we started doing these things, light clients weren't even possible. I imagine that MetaMask shows the most sort of easiest route to build these technologies, because building them completely decentralized from the get go is an immensely difficult undertaking.

[0:10:25.7] JM: If I use a web browser today, there are plenty of different web applications that I interact with. There's LinkedIn, Facebook, Twitter, my banking applications. What are the Ethereum-based applications that people are interacting with on a regular basis today?

[0:10:43.9] JH: Yeah, it's a good question. I think it's pretty early days for a lot of different apps. I guess prediction markets are one, such as [inaudible 0:10:51.0] and you've got Marker.

Actually, I don't know how to answer that question. It's such early days and everything is still being built. What's actually production?

[0:10:59.1] OT: It depends on what level you look at it, because if you look at blockchains more generally, you could say that's sort of store and transactional value. That's sort of one general application, and then you have the founding application, which has happened with Ethereum and ICO's and so on. Those are the two sort of most real-world usage kind of things. Then you have more frivolous things, like CrytoKitties and so on. But as Jarrad said, it's so early days that we still don't know sort of what the most fruitful directions will be, like sort of what apps will be successful so to speak, and that's something I think we'll see more of these next few years, sort of excite me what form that will take.

[0:11:35.4] JH: Yeah, keep in mind at this stage, like a lot of the blockchain technologies itself and the surrounding peer-to-peer protocols are still being worked on. Things are very experimental. For example, things like certain crowd sales and even CrytoKitties tend to slow down these networks, because they are not designed to scale. So the largest thing that we're all talking about and then trying to work on at the moment is how we can actually get blockchain scalability.

[0:12:01.3] JM: So the main purpose today for a mobile Ethereum client, would you say it's the wallet functionality, having a mobile wallet?

[0:12:11.3] JH: Yeah, absolutely. I think in terms of cryptocurrencies, transactions or value are one of the core sort of promises. But a client self, there should be an expression of what its blockchain is capable of. For example, in Bitcoin, wallets make a lot of sense, because it is really about sending and receiving some kind of value. Whereas an Ethereum, transactions tend to be a lot more like state changes in a database. What that looks like and how you manage that can be entirely different.

[0:12:42.7] JM: So when you imagine 5 or 10 years down the line, what do you envision people will be using their mobile Ethereum clients for?

[0:12:52.4] JH: Very good question. How I envision things is a lot of IoT devices will be running some kind of trustless forms of decentralized networks. A lot of our sort of infrastructure in our society will be running on this as well. I imagine whatever our personal computing devices look like at that time, whether that is a mobile device in the smartphone form factor or not, or be used to interface with, say, smart cities.

[0:13:18.4] OT: Another component of that is also the sort of messaging part of the application, which is done with [inaudible 0:13:22.9], which is also part of the Ethereum sort of sub-protocols. An interesting part there is that if you look at traditional social networks and instant messages and so on, often they have some kind of some model where the user is the product and it's being sold to advertisers in some form or it's sort of — It's not clear exactly how the economic incentives work, whereas with something like Ethereum [inaudible 0:13:43.7] we can actually create the decentralized sort of messaging system and social network where individuals, sort of users and stakeholders pay for specific services and so on, and I think that's an interesting change where instead of the user being the product, sort of they can take on the system and the incentives are sort of more aligned and you can have micropayments and things like this.

There are lots of interesting directions in terms of decentralized social networks and instant messaging and so on, but they're kind of more around like charities in a way, which is maybe not the most sustainable solution to actually getting something that's user friendly and works great for end users.

[0:14:20.4] JM: Yeah. I mean, we saw that the LO model of a kind of a open, none for profit social network didn't quite work. Whisper is a decentralized way of doing messaging. Explain how Whisper differs from something like Telegram or WhatsApp for example.

[0:14:39.9] OT: Whisper is, as I said, completely decentralized and it's built on sort of the peer-to-peer layer of the Ethereum protocol. You have Ethereum and then you have sort of the main blockchain protocol, and then you have Whisper and then Swarm which is both file storage, but they all use same — Connecting via the same nodes and so on.

Whisper itself, it's kind of a bit like a distributed hash table, like a broadcast protocol and it's novel in that it's completely identity based, which is sort of useful for maintaining, I guess, privacy and sort of darkness I would say in routing. The way it works is essentially you have a topic, which is just some bits of information with some intent of roughly who should receive it. You can also choose to have like almost no topic or the same topic, and then you have a key, so that could be a metric key or public key. The way it works is that all its messages are forwarded to all the nodes, and then only the person or the people who are able to decrypt the message actually reads it. This way it's identity-based in a sense that you need to have the actual key to read the message. No one can tell if it's addressed to you specifically, because every node gets it. So it's very different I would say from other applications which rely on central services and so on. It's an entirely different thing, I would say.

[0:15:57.1] JM: Yeah. Does Whisper operate by interfacing with a smart contract or does it create transactions in some other way on the Ethereum blockchain?

[0:16:08.1] JH: No. It's a separate sub-protocol. So we all know the Ethereum public blockchain, and that's a specific sub-protocol of a peer-to-peer transport called dev p-to-p. Ethereum is more than just a blockchain. It also provides a messaging communication layer for any kind of communication the app needs to make. It has to be faster than creating blocks and obviously cheaper, as well as decentralized file storage, which is called Swarm. This is something like — You could imagine like Bit Torrent, except it has an incentivization layer. So seeders have a reason to keep files around.

Imagine this being like a Web 3.0 stack where you can deploy your quotation mark server side logic as smart contracts. You can have real-time communication — Or not near real-time communication over Whisper, and you can have your actual HTML or JavaScript or whatever your DAP interface is is deployed into a decentralized file storage.

[0:17:10.4] JM: Again, we're talking about Whisper, which is a protocol for doing chatting, but the main topic of this conversation is Status, which is the Ethereum client that you guys are working on. Just to describe it, it's a really nice UI of a mobile client and you can do things like chat within it. You can find decentralized apps to interface with, and once there are

decentralized apps that people want to interact with from a mobile phone, that will probably be useful.

[0:17:43.3] JH: The biggest transformation or shift that's happening right now is currently we have this model of, okay, a service is free, but at the end of the day the service providers have to pay the bills. They have to pay the server costs and keep the entire service running. This model naturally creates a bit of a problem in such that they have to either monetize the user-base.

In many ways, I like to think of it as sort of digital feudalism, where you have a lord and the villages and they end up having to tax them in some form. How we see this at the moment is either to advertising networks, which then gives these service providers — Another problem is how do they make better customers out of their existing user-base and how do they get them to buy more or at least click more ads. Then we start looking into sort of ways in manipulating user's data and even manipulating the information that users can consume through the platform to make them better customers.

What's happening now is we can actually recreate a lot of these systems, but we can do it in essentially a trusted way, where there is no — where the user itself has control over their own data and they have a general understanding of the deterministic ways that this particular smart contract of DAP can actually behave and they understand what kind of information is published to that and used by that DAP, which gets really interesting, because this allows us to create instead of a messenger that has centralized service and you are dependent on some company. We can do this completely peer-to-peer, completely decentralized and you don't have to trust us. Then there's this other concept called distributed autonomous organizations where we could actually put the entire organizational infrastructure into smart contracts so users themselves can have direct control over how the software that they use on a daily bases gets build.

[SPONSOR MESSAGE]

[0:19:47.5] JM: Today's sponsor is Datadog; a cloud scale, monitoring and analytics platform. Datadog integrates with more than 200 technologies so you can gain deep visibility into every layer of your stack and any other data that you're interested in tracking as well. For example,

you can use Datadog's Restful API to collect custom metrics from your favorite crypto data sources and analyze trends in Ethereum prices over time.

Start a 14-day free trial, and as a bonus, Datadog will send you a free t-shirt. You can go to softwareengineering daily.com/datadog to get that free t-shirt, and thank you to Datadog for being a continued sponsor. Get that free t-shirt at softwareengineeringdaily.com/datadog.

[INTERVIEW CONTINUED]

[0:20:45.1] JM: Now, what if — Since we're very early days, we don't actually know what will become, for example, the decentralized app platform for consumer applications. I mean, we know that Ethereum is great for fundraising, for example, like fundraising is arguably the killer app from Ethereum. You can have an ICO via Ethereum, which is super useful, and obviously it's got flexible smart contract development. But it still remains to be seen. Is this going to be a platform where people build consumer-facing applications, or will some other platform arise? Do you think that if there were some other platform that came — Because, I mean, when I look at Status, it looks like a consumer application. It looks like if there was a decentralized Instagram that you could interact with, you may want to interact with it via Status. Like maybe all of the primitives, the file storage and whatnot is just handled by the Ethereum blockchain. I don't know, there are some way of deploying the Instagram decentralized frontend to Status and you interact with it through Status. But what about a world in which that Instagram that's decentralized happens to crop up on a different blockchain? Do you feel like you're tightly coupled to Ethereum?

[0:22:12.7] JH: So basically, we're definitely working with Ethereum, and the reason being is the most public and most decentralized program of blockchain. A lot of the best research is currently being implemented on top of Ethereum, but who knows what the future looks like? We may be dealing with a world or an internet of blockchains, and maybe blockchains themselves aren't the best data structure for trustless decentralized systems.

What the future looks like is still unknown, of course, but we will make sure that we interface with as many as we could possibly can.

[0:22:45.6] OT: I guess one thing I want to add as well is that we also — Like a lot of effort is spent on sort of the usability for developers and sort of making a great experience for DAP developers and arranging hackathons and making like a great API and all these things. So I think it's a pretty great opportunity in that sense, also because with more users, it would sort of be a more attractive target to use Status as a platform to sort of target DAPs for, and mobile is also great sort of platform in general just because you have access to all these, say, like camera and geo-location and all these things. I think there's lots of interesting things you can do, and we aim to sort of be the best place where you can do these kinds of things.

[0:23:26.4] JM: What kinds of stuff have come out of those hackathons?

[0:23:28.5] OT: One interesting one, which I had no idea about. I might get the details slightly wrong, but there was hackathon in South Africa a few months ago and it was something about how they use cattle as collateral. This is like a common thing that they used. Some mode sort of a DAP where you could tie that to a smart contract so each sort of cattle had some kind of — Was like a mini token or something like that. I probably got the details a bit wrong, but it's an interesting sort of novel thing which no one — Like it's not something we would have come up with, right? It's just having a platform that enables these kinds of local markets to do what they're already doing in real-life, but maybe with sort of more legacy, institutions like traditional banks or pen or paper or — I don't know, whatever it might be, and sort of just using technology to sort of slightly improve on it.

[0:24:14.9] JM: What did they do? They put cattle on the blockchain?

[0:24:19.0] OT: It's a matter of having it as sort of collateral. Let's say you want to invest in some business, you want to build a farm or something and maybe lots of people don't have traditional assets, so they would use cattle as collateral instead and it would be a way of proving that you sort of have the ownership of the cattle in some way.

[0:24:35.0] JM: Oh, that's kind of cool. I could imagine like if you want to use cattle as collateral, there's not a good way of doing that via a mobile app and why not use Status for that?

[0:24:46.6] OT: Right. I mean, it's like a very maybe fringe thing, but it's just an example. Kind of novel things that people are doing that are completely unexpected and the only way you get that type of unexpected interesting results, they may or may not be useful to people in various markets is by having a great platform where people can do these kinds of experiments.

[0:25:04.3] JM: Absolutely. I mean, I think about then and I'm like, "Well, how else would you implement cattle as collateral from a mobile app? You pretty much need an —" I mean, I guess if you really want it to be decentralized, you have to use basically a mobile Ethereum client for that, for that use case today at least, unless somebody spends up a completely new blockchain.

Let's talk about the technology a little bit. One thing I found interesting was that you chose React Native, and I found that to be a bold choice. How is React Native served you so far? How is the performance? Are there any areas of the app where React Native is like not as performant? Because I've heard it can be sometimes hard to develop with.

[0:25:50.9] OT: Sure. I mean, I think it's a tradeoff like a lot of things. The main thing it gives us is sort of productivity and largely because we can share the same codebase. We're actually writing closure scripts which is compiled to JavaScript, which is React Native is. So we have both android. We have iOS. We also can leverage some web tools for debugging and so on. We can also run tests in Node.js locally and things like that.

There's a lot of benefits in terms of having a shared code base, and also there's a lot of productivity gains we find in terms of using language closure, or close script just in terms of those [inaudible 0:26:27.2] you can use and so on. I guess in terms of performance, I think it varies. I think it's maybe you get it slightly more for free if you go the native route. I know definitely it's some sort of parts where maybe it's not fast by default. It's an unsolvable problem I would say. It's more a matter of you try something and then you can always move things to native as it sort of turns out to be [inaudible 0:26:50.5].

I think, for us, the bigger problem in terms of performance is that fact that we are running a peer-to-peer application on a resource-constrained device that's intermittently connected. So actually running the node, I think that's possibly a larger problem than sort of direct native side.

[0:27:07.6] JM: So like you have to figure out how to do networking and retries and stuff.

[0:27:11.7] OT: Exactly, yeah. We have an application protocol on top of Whisper, which implements some sort of basic semantics that we use. Like for example, acknowledging of message and online statuses and discovery requests and things like that. I guess in terms of the React Native performance things, like a lot of it is the fact that you have a JavaScript main UI thread and then you go back to native and you don't want to cross that bridge too many times. That's one problem. When we get lots of incoming messages on the node, then sort of that has to be communicated, and there's some subtleties in terms of how you communicate it and making sure it doesn't block the UI rendering and things like that. It's something that we're working on, but it's not an unsolvable problem I would say.

[0:27:53.1] JM: Totally. So there is no backend. It's just a mobile client that interfaces directly with the blockchain. So I know there's also this idea of the status nodes. You're thinking about or this has been implemented where you have some peer-to-peer nodes where there could be basically a decentralized network that is relegated specifically to doing Status specific things. Can you explain what Status nodes are?

[0:28:23.4] JH: Sure. Keep in mind, Whisper is a dark-routed pub/sub communication tool. Messages are somewhat have a fixed time to leave. So they only bounce around the network and there's no guarantee they'll actually reach the end destination. So, for example, we're trying to build a user interface and a messaging sort of application that has a lot of the same user experience qualities that you would expect from a normal application. But in dealing with peer-to-peer sort of systems, particularly with Whisper, you don't have things like push notifications or offline inboxing. An offline inboxing is like storing the messages when your client is not online and ability to retrieve them at a later date. This is something a centralized service can do quite well.

Doing Status nodes is basically how can we retain a peer-to-peer network, at the same time trying to give the same user experience on that side. So people running Status desktop will be running kind of offline inbox in the network and they can actually serve as holding messages for their friends when they come back online.

[0:29:37.9] JM: Interesting. Have you guys looked at Scuttlebutt at all?

[0:29:40.9] JH: Yes.

[0:29:41.2] JM: What do you think of Scuttlebutt? Have you taken any inspiration from it? By the way, for people who don't know, it's like a peer-to-peer social network, basically, kind of like a peer-to-peer Slack.

[0:29:50.6] JH: Yeah. I think Scuttlebutt as a technology is pretty interesting. It doesn't have some of the qualities that we're looking for. What I really like about Scuttlebutt is its ability to sort of the append only and have some kind of direct history, but that's not something that we want in our design.

[0:30:07.4] JM: Okay. There's also the hardware wallet feature which you announced at Dev Con. Explain what the Status hardware wallet does.

[0:30:18.5] JH: Right. So basically operating under the assumption that your devices are not the most secure, and they could be hacked at any moment. We know some there are some people out there who are actually holding large amounts of value and have been on like public trains with that in a very leaky client, and it would be very unfortunate for those kinds of people to have their phones hacked or taken away from them, and that would be the end of it.

So that's pretty much what hardware wallets in general are trying to do. They're trying to create some kind of gap between your untrusted device to a form factor that has a much smaller surface area for penetration. Really, the hardware wallet we're trying to make is something that should be credit card size so you can still take it with you on the go, but still have all the functionality of a full-fledged hardware wallet.

[0:31:12.1] JM: So why would you want a Status hardware wallet as supposed to just like a off-the-shelf, like a Trezor kind of thing.

[0:31:20.8] OT: Well, really, this comes down to form factor.

[0:31:23.1] JM: Okay. But does interface with the Status app in any way?

[0:31:28.6] JH: Yeah. We already implemented it for [inaudible 0:31:30.2] Ethereum and we're looking at creating a form factor that has a Bluetooth communication so you can actually use it on IaaS as well.

[0:31:38.4] JM: Okay. If I wanted to use a hardware wallet in companion with my Status client on the mobile app, what would I do?

[0:31:47.6] JH: You would tap it on to the back of your device. You'd press a set up button, you'd enter in a pin and then anytime you wanted to make a state change or sign a transaction, you would put the card as close to the device as possible and then press the button. There's also a signing phrase, lets the user know that it's a trusted screen and it's a trusted device.

[0:32:12.0] JM: Right. Okay. That's great. That basically adds a significant degree of security to any kind of transactions that you want to make with your mobile wallet.

[0:32:23.7] OT: As well as convenience, I would say in terms of the form factor being maybe more familiar. I mean, I use a hardware wallet, but it's kind of you plug it into your computer and it's a bit of a set up and it's like a new kind of thing and it's a bit geeky still, where as a credit card is something — Like in London or whatever, you have these EZ cards or whatever it's called, oyster cards. It's a more familiar user friendly interface I would say as well.

[0:32:46.9] JM: You guys are working towards the one production version of Status. What stands between you today and getting to production? What are the things that you're focused on the most?

[0:32:59.3] OT: I would say it's — A lot of it is performance in terms of — so there's getting the traffic uses down. There's also — we want to have a better onboard experience as well. The main thing I would say is we're doing a security audit. Actually, right now we're running on Testnet by default, but we want to be running on Mainnet by default obviously. Mainnet, this one is actually value transfer. Actual money being transferred, and I guess that's a bit of a process

because we need to make sure that all sort of critical paths when it comes to transactions are secure. I would say that's the main blocker in terms of getting it to production.

[0:33:37.5] JM: When you're messing around with it on a day-to-day basis, do you just walk around and do you use Status to just — When you're dog-fooding it to send messages to each other or to make payments, and do you try to dog food it on a regular basis?

[0:33:52.8] JH: Yeah, we actually have an a initiative, which is a catalyst for finding things to work on inside Status, and that's called Status Everyday. We'll have this sort of buddy system that keeps us accountable if we're chatting with each other and making sure we can use it. Some days are better than others, or some nights are better than others I should say.

[0:34:11.9] JM: Okay. I'm fascinated by the development process of software in this space, because from top to bottom, everything has a twist on — If you compare it to traditional software development, like the incentives are different, the ways of developing backend and frontend interfaces are different. Can you just maybe outline some of those differences? Why is Status, for example, different than developing some SaaS company, like a traditional sales SaaS company or developing a product at Google or Facebook? How does development work at a product like Status?

[0:34:54.7] JH: A lot of it is like alchemy. It's definitely an art and a science and it's partly because a lot of research is happening right now, and a lot of the protocols are highly experimental in and of themselves. So there's that sort of side of things. The other side of things is the sort of cryto-economic or game theoretic considerations you have to create, which doesn't really crop up too much in Status as an application itself. But it does happen when we start talking about how you design systems that rely on smart contracts. Keep in mind you also have to deal with "immutability". Once you deploy that smart contract, it needs to be formally verified ideally and you need to have some very different upgradability path, let's say.

[0:35:38.8] OT: I guess another thing is also the fact that we are completely open source and we're also using — We have this initiative called open bounty, which tends to be about incentivized and open source. So we use it quite heavily where we have specific issues and we put up like a bounty in SNT or F where if you solve that sort of GitHub issue and you submit a

poll request, that gets merged. You get that reward, right? That's the way of sort of scaling our efforts and sort of creating a thriving open source community. I think that's also something that sets us apart.

[SPONSOR MESSAGE]

[0:36:15.7] JM: Users have come to expect real-time. They crave alerts that their payment is received. They crave little cars zooming around on the map. They crave locking their doors at home when they're not at home. There's no need to reinvent the wheel when it comes to making your app real-time. PubNub makes it simple, enabling you to build immersive and interactive experiences on the web, on mobile phones, embedded in the hardware and any other device connected to the internet.

With powerful APIs and a robust global infrastructure, you can stream geo-location data, you can send chat messages, you can turn on sprinklers, or you can rock your baby's crib when they start crying. PubNnub literally powers IoT cribs.

70 SDKs for web, mobile, IoT, and more means that you can start streaming data in real-time without a ton of compatibility headaches, and no need to build your own SDKs from scratch. Lastly, PubNub includes a ton of other real-time features beyond real-time messaging, like presence for online or offline detection, and Access manager to thwart trolls and hackers.

Go to pubnub.com/sedaily to get started. They offer a generous sandbox tier that's free forever until your app takes off, that is. Pubnub.com/sedaily. That's pubnub.com/sedaily. Thank you, PubNub for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:37:59.4] JM: So you guys raised $100 million in an ICO for the Status Network Token. What role does that token play in the network?

[0:38:10.1] JH: Incentivizing the storage of offline inboxing. As we mentioned before, it's probably going to be first use case. Another one, we'll be talking about how you do a username

registration, for example, is one way. You could imagine that picture — Like when you go to the supermarket and you want to get a shopping trolley, sometimes they have like a coin deposit sort of system. Are you familiar with this content?

[0:38:34.6] JM: Where you like put a quarter in and you get a shopping cart?

[0:38:39.0] JH: Exactly. Then when you're finished with the username or finished with the shopping cart, then you can get your coin back. So this is a way that we can handle username registrations, but we can also help incentivize a very small degree actually returning usernames when people are finished with them, because they would like to have this certain amount of value back.

This would be tied to an identity which will become like a whole other thing so you won't have to be logging in to webpages like you do now on every — Creating a new usernames on every single website. The other side of things there is how we're going to be approaching public chat moderation. For example, there's a large problem in the way that chats are created and then moderated in the future. You can see this on the astrophysics sub Reddit. I don't know if you've browsed there recently, but I think there's roughly 10,000 astrophysicists, but they have to deal with one moderator who's gone rogue. Basically, they put up a bunch of really bad CSS styling and anime. The side bar is filled with profanity and the same with the sticky threads.

This basically comes down to a problem which you call like one-time trust decisions, where this person was trusted at one point, but now everyone has to accept his law as gospel. How we're approaching public chats is quite different. Oskar mentioned in this where you can subscribe to a topic and you can imagine this being something like subscribing to like some kind of hash tag at Twitter clients and receiving any messages that has that particular hash tag. But this is just a large data stream information coming through and you want to be able to filter that out in certain ways.

What we'd like to do is separate the people who generate rules from the rules themselves and essentially create a constitution to public chats that anybody can submit rules to. In this case, these rules would allow certain people to be participants depending on a certain criteria, whether you forward their messages or not, this sort of thing.

Here, you want to incentivize people creating good rules and propagating rules, but how those rules are actually determined to be the best is basically by the dominant amounts of usage or subscriptions to those rules from other clients. Of course, this is also quite taxing for the end user if you're putting all of these responsibility on users. So it will allow you — It will allow the client to determine who your friends are who've already signed messages or who are friends of friends who signed certain messages, or certain rules I should say, that allow you to automatically subscribe to certain rules, which you can then opt out if you wish. SNT will be a large component in incentivizing those kinds of rule sets as well.

[0:41:30.3] JM: So Status raised money through an ICO. Could you talk about the process of preparing to do an ICO and talk through some of the numbers in terms of what you raised?

[0:41:43.3] JH: Sure. Our approach to the whole process is probably a little atypical. I mean, when we started, everything was really, really fringe. Basically, nobody even knew what Ethereum really was, and certainly not really anyone could give us any sort of legal advice. Basically — And of the various contributors and the firms that dig data, they didn't really have any legal infrastructure in place to be able to participate in something like this. Basically there's no norms. We did our best to pay diligence as much as we could, and basically we do everything we thought would put us on the sort of low end of the risk spectrum.

I mean, the process itself, I guess it kind of look like doing many things in parallel. We're speaking to many legal firms, we're doing a lot of idea generation, seeing what would pass to how we test or not. Looking at different jurisdictions as well around the world, which would be favorable. Talking with potential contributors. What they would like to see and what they wouldn't like to see. While at the same time trying to run the organization and build the application what we had at the time.

As we got more and more clarity, at least to the best of our knowledge, all of these continued, and we also decided to do sort of a bit of a roadshow through Europe and throughout Asia as well. I think today like there's a few different approaches. Some of them don't look — I don't know the specific details and I probably couldn't go into anyway, but like the soft approach

doesn't seem to be super favorable. My impression, there is some consensus on how to go about things these days.

[0:43:09.3] JM: You mentioned something called the Howie test. What is the Howie test?

[0:43:12.0] JH: Yeah. It's a very broad way of looking at, say, securities, for example. So if you were to do something like a token — I mean, depending on its functionality and its overall design, the intent of the organization and its promotion. You can kind of use this as a sort of guiding stone so to speak. But it is incredibly vague. So it isn't immensely useful.

[0:43:40.2] JM: So what did the actual fundraising process look like? How much did you end up raising?

[0:43:45.6] JH: Sure. I hesitate to talk about [inaudible 0:43:48.0], but it's probably the easiest for your viewers. We rose roughly around like $100 million equivalent.

[0:43:54.4] JM: As we're talking about before the show, this would look like a lot of money if you compared it to series seed, or A, or B or C round, but the way that ICOs are typically done, you're raising all of the money that your company would need to raise in its lifetime in a single round. So how does that factor into the amount that you would want to raise as a company?

[0:44:22.1] JM: Yeah. I mean, the way that we've designed our token, there's no inflation built in to SNT. There's basically a fixed amount of 6.8 billion tokens. We basically sat down and tried to see what our runway would look like for development. Try to figure out exactly what we'd like to achieve and what we like to promise, and a lot of that is in our whitepaper and it's focused predominantly on the development of sort of new technologies including trust indicators and other sort of interesting smart contract things.

But the [inaudible 0:44:55.2] and myself actually come from a background in software distribution. Before we even embarked on this journey, we tested a few sort of messenger-like products. We got sort of cost of acquisition under control and we looked at our retention curves. Taking this back to our core mission, which is to try to reach mass adoption for Ethereum, be

believe we can build a sustainable user acquisition engine that's run every quarter hopefully on board, if not tens of millions of users, hundreds of millions of users.

[0:45:28.9] JM: Why do you have to issue all the tokens at once? Why can't you issue them in stages?

[0:45:34.8] JM: We certainly can. In fact — Basically, how it's structured is we had 51% went to the public at the contribution event. 20% was reserved for the team, and this is to be allocated to the core contributors to the software development. Then we have — Because it is a fixed amount, and the reason for doing so is because if you start introducing more complexity, then you have a larger tax surface, which is kind of interesting when you're talking about programmable tokens. The other side of that is a matter of sort of faith. Like, I mean, you don't really know exactly what you're designing.

The last portion is basically 29% sort of reserve, and this is sort of tokens are locked for 12 months and then we retain the rights to sell those to new contributors periodically should we need to raise additional capital for the specific content of further growing the network. I should also say that it's proven that we don't need to do this, all of that will be burnt.

[0:46:37.9] JM: Okay. I see. You could theoretically raise more in the future if you really wanted to.

[0:46:42.8] JM: Yes, but that was only because we specifically design that and it would be determined based on the market cap at the time. I mean, everyone's probably very familiar with the volatility in crypto as well. It needs to be very careful on how you actually structure that kind of sell. It would probably have to be done on a very slow, gradual basis, otherwise you would influence the overall market quite dramatically.

[0:47:04.8] JM: So the raise for the ICO, all that money goes to the company itself and then the way that that makes it way to the founders is through salaries potentially and then the founders also vest some number of tokens overtime. Is that the accurate depiction of the compensation?

[0:47:27.0] JH: Yes, something like that. So there's a couple of different things here. To fully kind of understand that, like we have to look at what we're trying to do here. A common topic that came up when this sort of question came up is the sort of idea of equity, right? Normally you think about a sort of traditional company and you allocate this equity to investors, you [inaudible 0:47:47.6] and you raise through that. Before we can even go into what we need to explain about sort of networks and tokens around networks is we have to look at why companies exists in the first place. This sort of basic idea is that you want to reduce transaction cost, and transaction cost I mean here is this sort of mental abstraction. Like time that you spent filling out forms, waiting in lines, your mental load when you're doing things, the cost of communication.

Basically, companies reduce coordination costs. But now we live in this sort of internet year and we have blockchains which gives us some kind of boarderless non-physical legal jurisdiction and we have instant communication with the internet. Now we have this opportunity to change how we organize socially.

[0:48:33.4] JM: You raise the money from the ICO and then you also have these tokens in reserves. I'm just wondering how that translates to a compensation structure for the people working in the company.

[0:48:42.1] JH: The tokens themselves go to the status DAO and the company itself just coordinate the contribution event. Now, the status DOW would be something like the equivalent of a company, except it's all done in smart contracts. For those who don't know, a DAO is a decentralized autonomous organization, and that will ultimately be paying out to the various contributors.

The company itself ends up being much more like a vestigial form of the DAO and basically access some kind of legal interface and to legacy infrastructure. The 20% that I mentioned is basically what gets reserved for the team, and that also includes the founders. That depends on how big we grow the core team, core contributors.

[0:49:31.1] JM: What's the vesting schedule for the tokens or the people who work at the company?

[0:49:35.3] JH: Yes. That's basically 24 months vesting period, a six month cliff.

[0:49:41.1] JM: Do you worry that — Two month or two year vesting cycle where you have a token that is instantly liquidable. Do you worry about that leading to a risk of the liquidity being reached for the developers before the product or before the Ethereum ecosystem is even ready for mass adoption of the product?

[0:50:10.4] JH: Yeah, I guess it's a concern. That's definitely the case. Having said that, I think that the way that these tokens work is they're fundamentally driven by self-interests. If you do get tokens, it is your interest to see the market cap grow, which is exactly the reason why we started Status at all is because Carl and myself were so interested in Ethereum in the beginning and we wanted to see the ecosystem grow as much as possible.

As for blockchain scalability, that's kind of why we're working on the Nimbus project and we're working with the foundation to build out our own Ethereum 2.0 implementation. I think at the end of March there'll be a panel in Taipei about sharding in general and there's timeline that basically extends over the next 1-1/2 years more or less to build that out. With us having a direct hand in that, at least it's more under our control in terms of our ability to delivery.

[0:51:10.2] JM: I see. After that two year vesting schedule, the tokens are liquidable, right? You could if you want it to liquidate your entire stake.

[0:51:20.9] JH: Absolutely.

[0:51:22.5] JM: Okay.

[0:51:22.5] JH: That's not necessarily a bad thing either. I mean, the idea is the DAO itself — I probably wouldn't, because my heart and soul is in this organization and in Ethereum has been since the beginning, and the reasons we got into it is purely because of the technology in the first place. Having that said, what we're trying to do is we're trying to create software that the users themselves end up owning. There isn't supposed to be a necessarily strong dictator or a

strong sort of owning entity over this amorphous-like network. It's supposed to be software for the users by the users. It's really kind of like open source 2.0.

[0:52:01.7] JM: So you see your role more of as a bootstrapping. Your bootstrapping the DAO and in your ideal future in two years, the DAO would be taken care of by the community who has a stake in the DAO through those tokens.

[0:52:20.1] JH: Exactly.

[0:52:21.6] JM: I see. Just a few more questions. The SEC seems to be looking at ICOs with a little more seriousness recently. How is the SEC treating ICOs today? Are they giving any serious guidance as to the legality of these things?

[0:52:39.6] JH: Yeah. It's definitely a topic to talk about. What can I say is we have worked with a few regulatory bodies and helping them understand the landscape. I imagine that's what the SEC is also conducting right now. There seems to be much of a sort of — There seems to be an exploratory process and where they try to understand and sort of talk with various organizations that are in this. From there, they try to make some assessments on that.

Our entity is based out of Switzerland and we did exclude the U.S. participants from the contribution events. It seems like the balanced regulatory bodies, like they really have an opportunity to support consumers while at the same time leaving room for innovation. I do think that is kind of necessary and it's great that it is happening, because it is a sign that everything is becoming more mature.

[0:53:29.1] JM: I completely agree with that. I think I heard a podcast with Andreas Antonopoulos recently where he was talking about this, the fact that what it looks like is going to happen is the United States is going to have an outdated perception of these things, and places like Switzerland are going to be ahead of the curve and they're going to get additional business in their country because of their sophistication in terms of the securities. Obviously, there are some risks in opening those things up to public investors, but also there is a risk in being super conservative or just overly draconian, like the U.S. appears to be learning towards. I guess we'll find out.

[0:54:17.7] JH: Yeah. This is sort of an addendum to that. What I find really interesting is that with these kinds of things happening and the ability to be able to move jurisdictions, it kind of forces governments to start getting back. I mean, not that they aren't serving their people, but it forces them to be more competitive and serving their people as well. It'd be very interesting to see how this plays out of the coming decades.

[0:54:40.2] JM: Okay, Jarrad. Thank you for coming on Software Engineering Daily. It's been great talking to you and I look forward to seeing Status develop.

[0:54:47.0] JH: Thank you so much, Jeffrey. I really appreciate it.

[END OF INTERVIEW]

[0:54:51.1] JM: QCon.ai is a software conference for full-stack developers looking to uncover the real- world patterns, practices, and use cases for applying artificial intelligence and machine learning in engineering. Come to QCon.ai in San Francisco from April 9th to 11th, 2018 and see talks from companies like Instacart, Uber, Coinbase and Stripe.

These companies have built and deployed state-of-the-art machine learning models and they've come take QCon to share their developments. The keynote of QCon AI is Matt Ranney, a senior staff engineer at Uber ATG, which is the Autonomous Driving Unit at Uber, and he's an amazing speaker. He was on SE Daily in the past. If you want a preview for what he is like, then you can check out that episode that I did in conversation with him.

I've been to QCon three times myself and it's a fantastic conference. What I love about QCon is the high bar for quality, quality in terms of speakers, content and peer sharing, as well as the food and the general atmosphere.

QCon is one of my favorite conferences, and if you haven't been to a QCon before, make QCon.ai your first. Register at qcon.ai and use promo code SEDAILY for $100 off your ticket. That's qcon.ai and you can use promo code SEDAILY for a hundred dollars off.

Thanks to QCon for being a sponsor of SE Daily, and check out qcon.ai to see a fantastic cutting-edge conference.

[END]