

EPISODE 899**[INTRODUCTION]**

[00:00:00] JM: Cryptocurrencies are decentralized monetary systems built on open source software. The open source software movement has evolved from the world of Linux, MySQL and Apache to a thriving ecosystem of commercial enterprises built around open source software. This ecosystem includes projects such as Kubernetes, MongoDB and React.js. It includes large organizations such as Amazon Web Services, Elastic and Facebook. In a parallel software universe, the crypto ecosystem has built revolutionary new financial tools creating billions of dollars of value, but not very many massive commercial companies.

In the world of cryptocurrencies, many of the same rules applied to the classic open source world, but other rules do not. How do these two worlds differ from each other? How are they the same? And how might they end up colliding?

Haseed Qureshi, Joseph Jackson Jacks and Alok Vasudev join the show for a spirited discussion of cryptocurrencies and open source. Haseeb is a cryptocurrency investor. JJ is the founder of OSS capital, and Alok is an engineer and the founder of Crypto Venture Capital Firm Standard Crypto. This was one of the first episodes I've done in my new studio setup and, it was really fun. This is one of my favorite episodes so far, and I hope there are more like it. The in-person setup is pretty sweet, and I'd love your feedback on it. Thanks for listening.

[SPONSOR MESSAGE]

[00:01:46] JM: GitLab Commit is GitLab's inaugural community event. GitLab is changing how people think about tools and engineering best practices, and GitLab commit in Brooklyn is a place for people to learn about the newest practices in dev ops and how tools and processes come together to improve the software development lifecycle.

GitLab Commit is the official conference for GitLab. It's coming to Brooklyn, New York, September 17th, 2019. If you can make it to Brooklyn on September 17th, mark your calendar

for GitLab Commit and go to softwareengineeringdaily.com/commit. You can sign up with code COMMITSED that's, C-O-M-M-I-T-S-E-D and save 30% on conference passes.

If you're working in dev ops and you can make it to New York, it's a great opportunity to take a day away from the office. Your company will probably pay for it, and you get 30% off if you sign up with code COMMITSED.

There are great speakers from Delta Air Lines, Goldman Sachs, Northwestern Mutual, T-Mobile and more. Check it out at softwareengineeringdaily.com/commit and use code COMMITSED.

Thank you to GitLab for being a sponsor.

[INTERVIEW]

[00:03:17] JM: Haseeb, Joseph, Alok, welcome to Software Engineering Daily. The first in-person recording of Software Engineering Daily.

[00:03:25] AV: Wow!

[00:03:26] JJ: : Awesome!

[00:03:26] HQ: This is exciting.

[00:03:27] JJ: Thanks for having us.

[00:03:28] JM: Alok, this is my first time meeting you. I've spent some time talking to Haseeb about his beliefs on crypto. Haseeb is pretty grounded. Haseeb believes in the idea of decentralized money, but when I talked to him about DAPS, and decentralized file storage, and building like a shared world computer, he's more skeptical. What are the long-term goals of the cryptocurrency ecosystem? Are we just building decentralized money? Are we also building the whole DAP world computer? What's your perspective?

[00:04:00] AV: It depends on how long-term your thinking. I think that right, now things like money are showing the most signs of working. They're also the things that are technically more easy to implement than some of the more advanced DAP functions that people are kind of projecting out. But I think in the long run, I think crypto networks can replace a lot of internet networks. I think the benefits of the tech, I think, I'm very optimistic on. So question of timing, where not going to have Uber on chain anytime soon but, but maybe someday.

[00:04:34] JM: Give me your description of the developer ecosystem of cryptocurrency protocols in 2019.

[00:04:43] AV: Communities are really important. In fact, kind of quality of community is one of the most important facets for kind of quality of a crypto networker or a project. The developer community is kind of a very important constituent there. I think right now we're in a situation where you have a small number of really competent developers flanked by a much larger number of earlier in the curve, less great developers. But I think that good projects can attract good engineers, but it's still far from the quality of the open source community that we have and kind of infrastructure broadly.

[00:05:17] JM: Haseeb, anything to add to that?

[00:05:19] HQ: I mean, I think one of the key storylines of the last couple years – So, Ethereum has attracted by far the largest developer community and people outside of the Ethereum Foundation itself actually building infrastructure and contributing to the ecosystem there. Most other projects don't really have that same wellspring of developer activity, and the ratio of outside developers to internal or internally-funded developers is very lopsided.

That said, looking at some of these ecosystems, we're going to have projects coming to light pretty soon that are going to have like hundreds of millions of dollars in capital purely to incentivize developers to build on their platforms. One of the real open questions that hasn't really been answered yet is is it possible to pay your way into a developer ecosystem? We're going to see the first trappings of that as some of these mega lunches actually start happening over the next year or two.

So far, I think the early signals have been kind of dim. It seems that there is some degree of – I mean, the way that I kind of think about this is like concentric circles of developers, where in order to get somebody to actually come build stuff for your blockchain, you have to decide, “Okay. Who am I actually targeting?”

Now, there's this core inner group of like the most hardcore crypto people who just love crypto. Even if they weren't getting paid for it, they'd be doing it. Those people already have allegiances and so they're very difficult to buy. The people who are working on Bitcoin, the people who are working on Ethereum, if your project number X, Y, Z and you just have tons of money and nobody really knows who you are, it's very hard for you to buy those people to come and have them build on your platform just by incentivizing them with bounties or with grants.

So you have to start appealing to these wider and wider concentric circles outside of them of people who are less ideological, who are less intrinsically aligned with the storyline of cryptocurrencies. It's not obvious that you can actually buy those people, because what they get out of open source development is usually not money. It's prestige, it's interest, it's satisfaction. So it'll be a tough time seeing if people can crack that nut of having these sort of paid incentivized open source networks, which is really not something we've seen yet.

We've talked very briefly about the problems in open source incentives that like there isn't potentially enough public funding for all of the open source infrastructure that tons and tons of companies depend on. Crypto sort of has the inverse problem. They have too much money and not enough open source development. I don't know. I am curious what your thoughts are about that.

[00:07:42] JJ: Haseeb is pointing to me. This is JJ. Not to presume we all sound the same.

[00:07:45] HQ: Oh, yeah. Sorry. I'm pointing to JJ.

[00:07:47] JJ: We all have reasonably unique voices. So, one thing I think a lot about, which is kind of fascinatingly obvious, but not talked about publicly a lot in these two kind of universes like cryptocurrency blockchain world and on the open source world as two kind of big separate ecosystems, but that are very interrelated, is that, exactly as you said, Haseeb, like the funding

subsidization effect in open source is basically the thing that underpins the sustainability of most open source projects, most open source communities.

So you have a big bank or an insurance company or a media company that employs lots of software engineers and, invariably, they're building some business logic application that's targeted at solving for a trading application at a bank or a web service in a media company or some underwriting application in an insurance context, but those software engineers that are part of the tens of millions of software engineers on earth are paid a salary by those companies, and that salary is the subsidized funding indirectly effectively for the open source technology that those software engineers use. Vastly open source. So open source languages, compilers, operating systems, and many cases increasingly open source applications, like Mattermost and other things.

So we have basically trillions in investment from like the world at large paying for the salaries of those tens of millions of software engineers over time and compounding too, because the number of software engineers is growing and increasing. That's kind of what I would classify as the open source funding ecosystem, and most people sort of like will assert that the open source funding world is nonexistent, which I find very oxymoronic and nonsensical.

Then I'm relatively, to Haseeb and Alok, very ignorant and a novice to understanding the deep kind of profound nuances of the cryptocurrency blockchain world. But as you just said, it's kind of inverted. There's basically this incentive monetary intrinsic mechanism built into the process of as a developer contributing to any of these blockchains, or tokens, or ecosystems, because you sort of have to buy into some kind of economic architecture as you're doing that.

So I guess maybe a question for both of you guys, Alok and Haseeb, which you probably have like a ton of insight on. At what point or what are the sort of things that would need to come together in order to sort of cause the cryptocurrency blockchain ecosystem to sort of see similar indirect subsidization effects that are independent of convincing any given developer constituency that a given incentive structure is the right one?

[00:10:26] HQ: I mean, that's a tricky question, because what a lot of people in crypto would claim is that if you look at something like Bitcoin. Bitcoin, many would claim, solves the open

source incentivization problem precisely because if you work on Bitcoin, Bitcoin becomes more valuable and you own some Bitcoin. You have skin in the game, and therefore you're increase the value of your own portfolio by working on Bitcoin. That's the sort of naïve argument. I think it's a very bad one, because it's sort of like is the classic example of a game theoretical if you're writing a problem, where if I'm a developer and I own a little Bitcoin, the incremental value that I will increase my own Bitcoin is trivial compared to the amount of work that I'll be putting into it.

[00:11:06] JJ: There's also the thing that Bitcoin is Bitcoin and it's not a web server. It's not an operating system. It's not a language. It's not all the other wide array of tools that developers actually need that don't have a built-in mechanism for economic incentive.

[00:11:20] HQ: Right. I tend to believe that at the end of the day, like the open source ecosystem between crypto and the just standard open source at the end of the day has to be similar and that it's basically self-directed based on a prestige economy and more about like a social network of people working at a passion than it is at a being purely paid incentives, because it seems that we've not seen any good examples of a developer ecosystem that has worked purely on the basis of paid incentives.

We've seen EOS and TRON try to get at this a little bit, and they just are not attracting high-quality developers.

[00:11:56] JJ: Unpack that a little bit more. So you were all like deep into the mazes here. So we might be kind of understanding, but the listeners might be lost. So when you say that in the crypto world, we haven't seen something that has worked in terms of getting developers to contribute based on some paid incentive. What do you mean by that? Unpack that a little bit more.

[00:12:14] AV: Yeah. I think there are elements to it. One is getting people to work on the core protocol itself and the others are to get people to build on top of it. Two different types of developers, two kind of different incentives as well, and neither have really worked. In kind of a pure pay for plays answer, trying to directly monetarily incentivize.

Often it comes down to if you're looking for people to patch and upgrade your core protocol, you are either putting bounties up for very specific features, very specific conditions, and that doesn't work, because you'll kind of do the minimum that passes your unit tests and there's not a sense for a longer commitment to the project.

[00:12:54] JJ: You would just want to hack to get to the reward and move on.

[00:12:56] AV: Yeah, exactly. Similarly, hack to get the reward I think is one way to think about why the incentives are rough. But to Haseeb's earlier point, one thing that's interesting about crypto is that you can play the game on different timescales. So could something like a token, there are short-term incentives, but there's also a long-term incentive at play. With cash, for example, where if you were just paying, say, your traditional open source project, paying people per commit. One, there's a measurement problem.

I think the problem with paying cash for code contribution is that is kind of a one-off and is very transactional and it's hard to have kind of an iterated game or kind of a long-term approach when you do that. But there's a thought that something like a token where there is kind of a long-term optimism for its value capture that could help provide a longer term incentive.

[SPONSOR MESSAGE]

[00:13:53] JM: When I was in college, I was always looking for people to start side projects with. I couldn't find anybody. So, I ended up working on projects by myself. Then when I started working in the software industry, I started to look for people who I could start a business with. Once again, I couldn't find anyone. So, I started a business myself, and that's the podcast you're listening to. But since then, I've found people to work with, on my hobbies, and in my business, and working with other people is much more rewarding than working alone. That's why I started FindCollabs.

FindCollabs is a place to find collaborators and build projects. On findcollabs.com, you can create new projects or join projects that are already going. There are topic chat rooms where you can find people who are working in areas that you're curious about, like cryptocurrencies, or React, or Kubernetes, or Vue.js, or whatever software topic you're curious about.

We now have GitHub integration. So it's easier than before to create a FindCollabs projects for your existing GitHub projects. If you've always wanted to work on side projects or you want to find collaborators for your side projects, check out FindCollabs. I'm on there every day and I'd love to see what you're building. I'd also love if you check out what I'm building. Maybe you'd be interested in working on it with me.

Thanks for listening, and I hope you check out FindCollabs.

[INTERVIEW CONTINUED]

[00:15:44] JM: Haseeb, you said that there was not an example, a past example of a developer ecosystem centralizing around an open source project where they are directly financially remunerated. But that's kind of commercial open source. I mean, commercial open source is basically you are building MongoDB and it's in the open, but most of the people that build on it are financially incentivized to build it, because they have stock options.

[00:16:14] HQ: That's an interesting point. What you're talking about is like, okay, within the structure of a firm itself, like if I create a company that is building a database and then I pay my people to work on it and the database happens to be open source, I totally agree with you to the same extent that the Ethereum Foundation or the Eos Foundation can like hire developers in-house and work on the project.

100%, they can pay people to do that. If they give stock options, and benefits, and so on. Yeah, totally, that works. The question is not can you get a particular company with a particular business model to hire developers to work on stuff. 100%, you can. The question is can you create a broad ecosystem of multiple actors in multiple jurisdictions who are all working toward the same end.

In regular open source, I mean, obviously it's not a cakewalk for anybody, but there is a broad set of constituents that are willing to work on infrastructure just purely for the kudos or just for the love of the game. What I think is probably the pivotal issue with trying to do a pay for play and trying to just literally say, "Hey, if I'm Linux," the Linux kernel, "and I just start paying people

out for every single commit that they add to the repo. I would assume that the biggest problem is that –” This a classic problem in psychology. When you start paying people for stuff that they're doing out of the love of their heart, it changes the way they approach that thing they're doing.

So there's this famous example of a – There's like this Israeli school that – I think it was like daycare, and they had some social science researchers trying to figure out why did parents end up coming so late to pick up their kids? So the school administrators would like sit around after it was time for the kids to go home and parents would be reliably late. So they decided to impose a fee, where if your kid was late, then you'd have to pay this fee. What they found was that actually increased the rate at which parents came late, and the reason why they posit that it increased the rate at which parents came late was because it now made this thing transactional.

If you're a parent and you're running late to pick up your kid, now you think, “Well, I can pay \$10 to be late. It's fine. I'm paying them for their time.”

[00:18:12] JM: But that's not right. It's transactional either way. They're just shifting the opacity of the transaction.

[00:18:20] JJ: Yes. Legibility, right?

[00:18:21] JM: Yeah. Because before it's like, “Okay, you show up on time, because otherwise your reputation suffers in this small community and your reputation with the daycare owner, and you might get kicked out of the daycare, and that sucks.” But if you make a financial thing out of it, then the daycare is implicitly saying, “Look! You have the option to pay \$10 extra and show up 20 minutes later.”

[00:18:44] HQ: Right, but ultimately it commoditizes that lateness. It commoditizes time. By treating time as a commodity as supposed to a social imposition or a threat to your own reputation, that changes your calculus in deciding how you're going to deal with it.

[00:18:58] JM: But not necessarily in a negative way. It just turns into a more market-based opaque.

[00:19:04] HQ: I agree. I agree. But the interesting thing is that –

[00:19:06] JM: Or it's like transparent.

[00:19:06] HQ: It doesn't do what you'd expect. You would think, "Well, they're not losing any money."

[00:19:11] JM: But that's a separate problem.

[00:19:12] HQ: What do you mean it's a separate problem? Here's the analogy I'm making, okay? The analogy I'm making is that if you started paying people per commit in the Linux kernel, people would start thinking, "Okay. Well, my consulting rate is like 100 bucks an hour or 150 bucks an hour and now I'm getting 30 bucks for the five hours I put into this commit. Is it really worth my time?"

[00:19:28] JJ: Maybe an improvement. You may see an improved ecosystem. I mean, you cannot say today that that doesn't work because payment system sucks still. The payments integration problem is still so bad, right?

[00:19:42] HQ: I don't disagree with that, but I would be very surprised if that were just – Let's assume that people are not stupid and that if this was a markedly better system for open source, people would've tried it and –

[00:19:52] JM: Have you tried to use Bitcoin? Have you tried to use – Doesn't GitHub have something like primitive system for incentivizing people now and it's like still not great because they just rolled it out?

[00:20:02] HQ: Yeah. I think they added something very recently.

[00:20:04] JM: This is just not have been tried.

[00:20:04] JJ: You have sponsors, and the very talented and brilliant, Devon Zuegel, is running this program at GitHub. I think we're glossing over something a little bit more fundamental that kind of touches on what Jeff and Haseeb were kind of bouncing back and forth on, which is intrinsic motivations and incentives and extrinsic motivations and incentives. Intrinsic ones are sort of you do it for the love of the game, as Haseeb was saying. You do it for your own, internally kind of altruistic.

[00:20:31] AV: Is the developer an artists or are they a capitalist?

[00:20:33] JJ: Yeah, moral reasons. Extrinsic motivation is where you introduce some type of reward loss function essentially to the whole thing and you have this like opportunity cost. People associate money with the measurement barometer system of like opportunity cost self-worth value ego. So when you sort of say, "Okay. Now, there's some kind of like money-based scorekeeping mechanism. The extrinsic motivations all of a sudden introduced into the system."

I proposed with the Kubernetes community a number of years ago to use Balaji Srinivasan's early implementation of a payable inbox, earn, com for creating a list of Kubernetes maintainers that were sort of expert people and using that as a way to sort of call and dramatically reduce the thousands of open issues on the Kubernetes repo and GitHub that were vastly troubleshooting related. So people were like, "Hey, how do I basically –"

[00:21:30] JM: So paid stack overflow.

[00:21:32] JJ: "Troubleshoot the port mappings of my service endpoint to my Docker container port and my web application port that are misaligned," and this is like one of the most common things that would result in like this networking error and IP tables, so.

[00:21:47] AV: This is one of the primary functions of a company.

[00:21:49] JJ: So people have this question. They go to the question list. They ask the question of the list. They set a budget and a certain number of experts will respond. Balaji's idea with the payable inbox is pretty profound. It basically introduces an extrinsic incentive mechanism into sending people messages. So if someone responds to the message, you have to pay them

typically a small amount, like \$1, \$2, \$10 \$50. I think Marc Andreessen on like 100 bucks. To get a response from Marc, you have to pay 100 bucks, and it's all crypto. So you basically load up some money into a wallet.

The Kubernetes community vehemently pushed back on this, like all the people that were at the time kind of on the steering committee. I won't name names, but pretty obvious. The list responses is preserved on Google group's history. Tim Hawkin, Brian Grand, Eric – I forgot his last name, from Google. Basically some of the sort of –

[00:22:35] JM: Googlers.

[00:22:35] JJ: A lot of the Googlers kind of said this would sort of introduce like a perverse incentive into the mix of intrinsic. This thread evolved into a pretty philosophical back-and-forth on intrinsic and extrinsic motivations sort of independent of the scope and the proposal and like earn.com the whole thing, which I found very interesting. It's up there. Maybe I can send you a link and you can put it on the show notes. It's actually a pretty interesting thread. It wasn't anything like antagonistic, but there's some very –

[00:23:03] JM: Google wanted to maintain their control. I mean, altruism.

[00:23:05] JJ: Essentially, Google won the argument at the end of the day, and that wasn't something that was really pushed forward. It ended up getting a bunch of people on the list, but never really helped. I think the point about Linux kernel contributions and like, basically – Well, I think Jeff's actually right. I sort of agree on both sides. There are two schools of thought on both sides. Right now we have a technology problem, a user interface problem. It's really hard to actually get something like that to be adopted, because it's just so crappy. The implementations and the options people have are really bad.

Things like HackerOne are working really well. 20 years ago, we had White Hat Hackers that would do it for intrinsic reasons. They would find vulnerabilities, cross-side scripting, SQL injection attacks. They would tell the web owners, "Hey, if you don't close these vulnerabilities, you're going to have like Russian hackers like basically still clear text SSN, credit card information, and sell it on the black market."

Now we have HackerOne that basically pays people bug bounties and access as an arbiter to basically build a network of companies that want to get hacked and will voluntarily opt in to getting hacked by people all around the world, ideally, White Hat Hackers. We have like a sort of centralized incentive mechanism for that particular type of developer activity. We don't have it for generalized abstractions for like everything.

I think, currently, in the open source world, the incentive abstraction lives at an indirect level. So it lives at the level of the company. So you have commercial open source companies as we call them at OSS Capital and other places, like HashiCorp, Confluent, MongoDB, Red Hat, many others. They have thought different business models, but they are the centralized entities typically that are acting as stewards and significant kind of center of gravity and credibility points of presence for the open source projects that they maintain and they commercialize and that they build products around. Those companies are very often venture funded. Not all them are, but many of them are, and they have very well-understood and increasingly, highly scalable business models that enable them to become 5, 10+ billion dollar companies.

So the sort of incentive mechanism lives at the layer of abstraction at those companies. It lives very far away. In fact, I think optimally so from the open source projects. With the crypto sort of token ecosystem world, the exact reverse is sort of materializing where there's this very significant pent-up interest and a desire to shift that incentive mechanism as close to the open source technology and open source project as possible.

[00:25:31] HQ: The difference is there's no business buyer. I think for most commercial open source, there is a business buyer.

[00:25:37] JJ: Enterprises.

[00:25:38] AV: Yeah, enterprises are buying and there're large contract value up for grab. But even within like crypto aside, there are certain kinds of open source projects that venture capitalists are more excited about, the business buyers more excited about, and those I think can get funded well. They can hire developers directly and pay them to work on it. Then there's also a class of open source projects, things like frontend frameworks, things like programming

languages that are much harder to build business models around that don't get the same kind of

–

So, interesting question, and I don't know the answer to this, is the quality of development, quality of the community. What's the difference between kind of, let's say, like something like Kafka compared to something like React?

[00:26:19] JJ: Wait a minute. Finish your question, Alok.

[00:26:21] AV: I'm saying, is there a noticeable difference between the shapes of the communities and ultimately the quality of development and leadership for open source projects that have obvious business use cases that can seek funding more easily versus those that are harder to monetize that almost by necessity need to have a different shape of contribution?

[00:26:41] JM: Okay. So JJ said, he said that the way that the companies do it right now, the way the enterprise open source works, you said it was – I believe you said it was optimal relative to pushing the financial incentives closer to the developers, which is what the crypto people want to do. Am I correct?

[00:26:57] JJ: So you're embellishing a little bit what I was saying. I think currently we found of various scalable equilibrium and an abstraction in terms of where the incentive lives in order to create a significant sort of growth market around commercial open source. That was sort of 1/20th what it currently is five or six years ago. I don't know how useful that data point is, because you could say the same thing of crypto five, six years ago, and crypto is like 1/1000th what it is today five, six years ago.

[00:27:29] JM: So here's the point I wanted to make, which gets to what Alok was raising. I think what we have right now is a crutch. I think that the Google model of launching Kubernetes. I think that the MongoDB model of doing MongoDB is a crutch, and I think you see the fractures in that crutch in things like Istio and things like the licensing debates, where we see the fracturing that comes from this weird incentive structure where you have this open source thing that has this air of altruism that comes from the roots of the Linux foundation.

You don't see this with something like React, where React says – They come right out and say, “Look, the stuff mostly comes out of Facebook,” and they're very transparent about it. They're like, “Look, Facebook is basically on the bleeding edge of frontend development. We're solving the hardest problems here internally at Facebook, and we're going to go out and talk to tons of people and we're going to make sure that our problems are consistent with the external problems. But we're going to be very honest and transparent.” They're like, “Look. We've got a dedicated team that's working on this stuff. So, yeah, we control it. Why wouldn't we?”

Whereas with the Kubernetes and Istio thing, you see like this really weird, like molding of the community, and I just think – I'll summarize real quick. I think that the pushing of the incentives closer to the developers will ultimately work. I think it'll work way better than this weird indirection of the commercial open source stuff.

[00:29:10] JJ: Yeah. So to be clear, to qualify what I was saying earlier, I'm not saying it won't work. I'm not saying I'm against it. I'm saying, currently, where value is being captured sustainably and scalably based on current models is indirectly through companies that commercialize open source as cost businesses as we call them, commercial open source software companies, or any company that employs any software engineer literally doing anything, paying them a salary, and then that software engineer has autonomy to choose what open source technology they use.

I want to mention that you've conflated two hugely profoundly different things in what you mentioned before, which is like worth pointing out. You basically articulated a contrast between the motivations and source of open source projects that came out of either Facebook, Google, and then you also mentioned foundation-led or foundation-driven open source projects.

The motivation and creative drivers behind different open source projects is entirely orthogonal to how sustainable they become as commercial ecosystems, entirely orthogonal. We've done lots of research on this at OSS Capital and just historically myself about like what are the sources for the projects. Where did they come from? Why were they created? In particular, projects that became the basis for huge companies. So Elasticsearch, Vagrant, Git, Kubernetes, what have you, applications, compilers up and down the stack.

There is a wide range of source creation, motivation, variance in why these projects were created as contrasted with how they ultimately came to be commercialized at scale. In fact, there is no correlation between where and why the projects are created and whether or not they were highly commercializable or not at the end of the day. So Elasticsearch was a personal project built by Shay Banon. Eventually became widely adopted in the industry and it's now the basis of a \$10 billion company.

Vagrant, also, personal project built by Mitchell. Sort of became the basis for HashiCorp and lots of other projects. Something like Cassandra came out of Facebook. It's the basis for data stacks, which is a successful commercial open source business. Something like Spark came out of AMPLab in UC Berkeley Academia. Became the basis for data bricks, multibillion-dollar company. Something like Kafka came out of LinkedIn. Big internet hyper scaler, became the basis for a multibillion-dollar company, Confluent.

There's a very wide variety and there's probably 50 other examples that are similarly distributed and there isn't any particular concentration. I just wanted to mention that, and what you are saying around the sort of crutch. I don't think you really elaborated on what you meant by crutch. I think I understand, for the listeners, it's probably useful for you to elaborate a little bit.

[00:31:52] JM: I just mean we're on the way towards developer networks that are paid.

[00:31:57] JJ: I think there's going to be very – So at OSS Capital, we've invested in a company figuring out and sort of iterating and exploring models to shift that incentive closer to the developer. So we're very for business model innovation, incentive structure innovation, licensing innovation, innovation all around. That's just going to continue.

I guess the broader thing that I'm kind of in particular convinced of is that the future will be very heterogeneous. I don't think there's going to be any pendulum swinging in the complete either end of the sort of spectrum or direction. We're going to have tokenized open source ecosystems and we're going to have commercial open source ecosystems.

The latter, go out and take money and build products that are sold to the \$4 trillion enterprise market. The former will somehow create a very usable, something that has a low-level of friction

to get massive adoption. Experience to tokenized open source and build some incentive structures close to the open source projects as possible, or maybe something like GitHub sponsors will work, where there's a big centralized network where people host their code and collaborate around it and you can actually introduce some kind of sustainable – Ideally, it's not a digital tip jar. It's something that will literally create a two-sided marketplace to make things compelling.

[00:33:09] HQ: One reason why I am skeptical of this sort developer for hire model of open source, like pushing the incentives does to the developers, is that I think while economically it sounds pristine. It sounds lovely to like have this idea that like, “Okay. Well –” The reason why I don't think that economic model applies well to open source is the few reasons. One is that developers are very non-uniform. Like there're some developers who are just absolutely amazing killer beasts and they are the people who actually maintain and build these things in the first place. Those people are extremely valuable relative to the marginal developer who is just somebody onboarding on to React for the first time and wants to grab their first poll request. The way you incentivize those two developers is very different. You care a lot more about how you incentivize the first developer than the second if you want a thriving and stable open source ecosystem.

I kind of think in a way it's a lot more analogous. If you think about what is driving people to do open-source and what even is like that? The best analogy I can think of is professors agreeing to edit journals, like peer review, exactly, where it's something that you do kind of just for – Because it advances your career, makes more prestigious. It's the kind of thing that like you can put on your lapel and look compelling to your peers. But you're not –

[00:34:20] JM: Which ultimately gets you paid.

[00:34:21] HQ: Ultimately gets paid, but much later on down the road. It gets paid indirectly through better career advancement and having more friends and more Twitter follower or whatever. The second reason why I think the economic model doesn't work that well is that there is very low liquidity when it comes to developers, by which I mean if I take the maintainer of Redis and I go port him over to React, he's not a fungible developer. I can't just say like,

“Well, he's a great developer. Therefore if I push him over here, he'll be great.” These projects are very sticky. ‘

Because of that, at the end of the day –

[00:34:53] JM: Not if you pay them enough.

[00:34:54] HQ: I'm sure. Yes, if you pay them some absurd amount of money.

[00:34:57] JM: Salvatore is a very talented programmer.

[00:34:59] HQ: Yeah, but you have to push over that stickiness threshold, which basically means that – I mean, open source is the classic public goods problem. That's why it's so weird that it works at all.

[00:35:08] JJ: It's an infinite resource though. So one thing that I –

[00:35:11] HQ: What is an infinite resource?

[00:35:12] JJ: So, Adam Jacob, who is a frenemy intellectually, let's say. We actually disagree profoundly. So he's the creator of Chef, which is this configuration management language.

[00:35:22] JM: Oh! He had a great show on Changelog recently.

[00:35:24] JJ: He actually did the recent OSCON keynote and he said something really awesome that I totally agree with, which is that open source makes software infinite, an infinite resource. Something that you can actually like expand out without boundary in terms of like the sort of elastic nature of like the resource itself. So it doesn't have any kind of like fundamental constraint around.

So perishable dynamics, like monetizing some perishable good is one way that like software is used to create value propositions for automation. We have perishable space in our cars.

Software creates the ability to place people in those cars efficiently and solve a bin packing problem. We work basically software abstraction for our office space.

[00:36:09] JM: Haseeb is talking about writing software, not distributing software.

[00:36:13] JJ: But I guess those things are two overlapping. We're kind bouncing around a lot. I've wanted to do this for a while and I've been thinking pretty deeply about this, and you too, Alok and Haseeb are very deep on the crypto blockchain space, and I want to run this by you. Something that I think applies to both the commercial open source world and the token open source world if you will, the crypto open source world.

There's two types of fundamental capitalistic orientations and the way you sort of look at like if you were to build an Uber in the way that Uber exists today or you were to build a commercial open source Uber or tokenized or blockchain Uber. The first approach, which is what we see today, centralized companies raise billions of dollars it's public, it has a board and so on, is what you could sort of in some ways classify as zero-sum capitalism. So it's a highly competitive market. There's lots of ridesharing competitors. There's Lyft. There's all the M&A. There's antitrust issues and so on, centralized, zero-sum monopolistic capitalism effectively.

If you look at the commercial open source Uber that doesn't really exist and hasn't succeeded yet, although there is something like LibreTaxi on GitHub and things like that are at the fringes. But let's just say that that doesn't exist yet, or tokenized Uber. Capitalism dynamics would be fundamentally different.

Something that I have been thinking about was when we described it as a meritocratic capitalism, where you decouple the core technology that creates the value from the company or the entity and anyone can capture value around it and anyone can commercialize it. So that's how commercial open source companies work, is you have an open source core. It's permissionsless and has had high levels of permissive rights to the users, contributors in terms of running the software and seeing it and so on.

[00:37:47] AV: Domains are an analogy here, where you have open protocols but then you have registrars.

[00:37:52] JJ: What do you think about that? There's monopolistic capitalism and then there's meritocratic capitalism. I think the idea of meritocratic capitalism is more scalable and it works more in this kind of sovereign individual economy that we're starting to see unfold. I think it applies evenly to both worlds, both the crypto blockchain world and the commercial open source world. Obviously, they're two different implementation approaches, but what do you guys think of that?

[00:38:15] HQ: I tend to disbelieve any claims of fundamental phase shifts in the way that economies or capitalism works as a result of crypto. It sort of like I'll believe it when I see it, but so far, there the sense in crypto that sort of the laws of economics are suspended somehow, because you can launch – I guess because you can launch a project and it's worth \$50 million and no other project went down or somehow a network forks and the collective value of those forks is more than the original chain. There are lot of really weird phenomena that make you kind of – That are almost convincing that like, "Well, this is just like a new kind of –

[00:38:48] JM: Throw away all the old ones.

[00:38:49] AV: I mean, we are creating money out of thin air.

[00:38:51] HQ: Right, exactly.

[00:38:52] JM: This is a pretty contrarian response.

[00:38:54] HQ: Yes.

[00:38:55] JM: It's eventual consistency.

[00:38:57] HQ: Yeah, eventual consistency is great way to put it. I think the world is eventually consistent. Ultimately, if crypto works in the long run, it works because it is a better version of things we already want. I don't think we will want new things because of crypto. I think we basically want what we want, and crypto, if it succeeds in giving us a better way to get what we

want, it ultimately gets cashed out in the real economic wants and desires that we have and what you'd otherwise call the capitalist economy.

[00:39:23] AV: But don't discount innovations in capital formation as allowing us to build things we've never built before. Accomplish things we've never accomplished before. This is like the joint stock corporation allowed us to build railroads. So I do think that the means of funding and organizing these things could absolutely lead to us building different things. But I mean, humans are human, so that's not going to change. Yeah, but I think kind of reducing frictions in small places could actually lead to very big impacts.

[00:39:51] JJ: This is Jevons paradox, guys. Like you make something easier to consume and you're going to have exponential increase in the usage of the thing. That's very well understood and it may not change the fundamental economic properties and laws that we know of, like Haseeb is saying, which I agree. It's much more pragmatic kind of view. I would probably subscribe to something similar. But the Jevons paradox thing, I think it feels like Jevons paradox still kind of applies, because you're making the incentive structure infinite as supposed to centralized one entity and it's somewhat disaggregated, like a meritocratic CAP table or something. Instead of establishing balances of power upfront, they're constructed evolutionarily based on contribution.

[00:40:27] HQ: I mean, we're seeing these experiments in real-time as different kinds of teams and different kinds of organizations are being attempted of like DAOs, decentralize autonomous organizations, which are basically these completely loosely coordinated set of actors who are just economically tied together to complete distributed teams, to teams where a very small number of people are working on the core protocol and most everybody else is building on top of it.

[00:40:49] JJ: Can I ask a question that I think a lot of people gloss over? Like DAOs, I think of as you describe them, like I feel like every blockchain crypto project markets itself as a DAO. They don't draw the line of differentiation between like, "Well, no, no. At certain layers, there's more decentralization than not."

[00:41:08] AV: A DAO is an org structure, right? A DAO is about business logic that's encoded on chain that allows people to kind of coordinate under a well-specified set of rules. So a DAO is very specific thing in the crypto world.

[00:41:22] JM: Because like Ethereum is not really autonomous, right? You basically got Vitalik.

[00:41:27] JJ: What are the two big examples of like DAO and not a DAO?

[00:41:30] AV: So the Dao was where the name came from, which was built on top of Ethereum, and what it was was it was basically a decentralized venture capital firm, essentially. So the idea was that the DAO was built on Ethereum. There would be DAO tokens, and you would put money into the DAO. Receive tokens in return, and those tokens would give you some exposure to the upside of the funds that the DAO accumulated and would also give you some governance rights on being able to kind of influence what the DAO funded.

The idea, all VC fund, if you think about it, is there is kind of a capital formation event, and then there's a decision-making process to invest in things. Then you have holdings, and if those holdings appreciate, then people that own stakes in the fund will have their stakes appreciate.

[00:42:11] JJ: It sounds awfully similar to how REI works. What is REI?

[00:42:15] AV: Oh, co-ops.

[00:42:16] JJ: It's a co-ops. Yeah, it's kind of a co-op model, right?

[00:42:18] AV: Yeah. I mean, the difference here, it was you who invested in the DAO as an LP and you were looking for direct upside.

[00:42:25] JJ: Thousands of mini LPs basically.

[00:42:26] HQ: Yeah. The interesting thing about the – I mean, the DAO obviously was hacked very famously and fell hard.

[00:42:33] AV: Yeah, it was hacked and basically there's – What? 120 mill worth of –

[00:42:37] JJ: This is not a new idea. The point is it's not a new idea. It's just at a different – It's a different implementation of a co-op.

[00:42:42] HQ: Well, I think the DAO itself was a particularly recognizable sort of twist on a venture capital firm. What I think of like the best example of like a DAO that is just totally unrecognizable to most people in the normal economy is Randao. So Randao is as a decentralized random number generator. The way that it works is – We don't have to go over the gritty details, but essentially you have like a bunch of people who join this DAO and what they do is they contribute randomness. That randomness gets XORed together and then that produces this random number.

Now why would people do that? Why would people do that? The reason why they do that is they get paid by the consumers of that randomness. So this Randao contract just keeps running and it essentially pays people to come work for it. When they're done working, they can leave Randao and other people can join.

So you can see how this is completely unlike any other system where the entire economic activity is getting coordinated by a smart contract and by pure incentives. Whereas the DAO, in actuality, the DAO was like it had these sort of managers [inaudible 00:43:40] and there were a lot of humans in the loop.

[00:43:43] JJ: When does quality of work gets solved? Because like the biggest – I think a driver for dictatorial, highly-hierarchical, founder-led companies is like, okay, the founder sets the bar. Sets the culture, sets the motivational driver, sets the pace, sets the keens, sets like how hardcore you're going to be. Famously, people go work for Tesla and they could if they can keep up with Elon's pace and his standards and his work ethic, and so on. At what point – For good or bad, and that goes on the other end of the spectrum too. At what point does the quality of work effectively problem gets solved when you don't have a centralized entity driving a certain standard?

[00:44:19] HQ: Well, for things like Randao and even things like proof of work contributions in crypto, you have the benefit of it's very easy to check whether what's being submitted is accurate. It's programmatic, right? In Bitcoin, you can check if the hash was generated correctly. In Randao, you can check if the entropy or your contribution it was generated correctly and it's very easy to verify.

[SPONSOR MESSAGE]

[00:44:49] JM: You probably do not enjoy searching for a job. Engineers don't like sacrificing their time to do phone screens, and we don't like doing whiteboard problems and working on tedious take home projects. Everyone knows the software hiring process is not perfect. But what's the alternative? Triplebyte is the alternative.

Triplebyte is a platform for finding a great software job faster. Triplebyte works with 400+ tech companies, including Dropbox, Adobe, Coursera and Cruise Automation. Triplebyte improves the hiring process by saving you time and fast-tracking you to final interviews. At triplebyte.com/sedaily, you can start your process by taking a quiz, and after the quiz you get interviewed by Triplebyte if you pass that quiz. If you pass that interview, you make it straight to multiple onsite interviews. If you take a job, you get an additional \$1,000 signing bonus from Triplebyte because you use the link triplebyte.com/sedaily.

That \$1,000 is nice, but you might be making much more since those multiple onsite interviews would put you in a great position to potentially get multiple offers, and then you could figure out what your salary actually should be. Triplebyte does not look at candidate's backgrounds, like resumes and where they've worked and where they went to school. Triplebyte only cares about whether someone can code. So I'm a huge fan of that aspect of their model. This means that they work with lots of people from nontraditional and unusual backgrounds.

To get started, just go to triplebyte.com/sedaily and take a quiz to get started. There's very little risk and you might find yourself in a great position getting multiple onsite interviews from just one quiz and a Triplebyte interview. Go to triplebyte.com/sedaily to try it out.

Thank you to Triplebyte.

[INTERVIEW CONTINUED]

[00:47:09] JM: Are these objective things? Because like you hear about these things in like a Mechanical Turk. If you have something that subjective or that's like even slightly, slightly not objective, you submit it to three Turks instead of one, and you just take the two out of three. Does that ever happen in crypto where you need to have a two out of three kind of like subjective voting sort of thing?

[00:47:30] AV: Yeah. Well, voting schemes are really common in crypto. In fact, some of the newer layer one protocols are actually – They use a voting scheme for the core consensus mechanism. So, for example, a lot of the proof of stake systems as supposed to proof of work, there's a voting quorum that basically decides whether a transaction is valid or not.

[00:47:48] JM: Does that lead to any perversions or does it actually work out well?

[00:47:51] AV: Well, it's gamed theoretically supposed to get people to act honorably, because if you're acting dishonorably, then if you lose the vote, then you lose money that you've put at stake, in principle. Again, that's kind of consensus. That's pretty core.

Other examples are there is a project called Augur that's about prediction markets, and basically what Augur does, it allows the outcome of a real-world event or the truths to be crowd sourced. So, for example, if –

[00:48:18] JJ: Their prediction of it.

[00:48:19] AV: Yeah. If you're betting on the outcome of an election –

[00:48:20] JJ: So, like Bernie is going to win the election. If he doesn't, certain people who took that lose money. People who took the other side of it make money yet.

[00:48:28] AV: Yeah. Well, so there's the market for the bet. Did Bernie win or not? Did you bet that he would or not?

[00:48:33] JJ: Then Augur does the bookmaking on the people who have actually –

[00:48:35] AV: So part of Augur's game theory is about is the truth that's being reported accurate?

[00:48:42] JJ: Because these are certain things that aren't categorically true until you get certain data inputs on them.

[00:48:46] AV: Yeah. For the prediction market to resolve, you need an input signal on kind of which side won or not, and Augur generates that input signal in a decentralized way by crowd sourcing it.

[00:48:55] JJ: That's a great application for this, Augur. I've heard of Augur before. I think I met Joey once actually, the Augur founder.

[00:48:59] AV: Yeah. There's a bunch of game theory on what happens.

[00:49:03] JM: Does it work? Does the Oracle process work?

[00:49:05] AV: So, reporting, yes. The rep tokens, or there's a sophisticated mechanism that's probably too much to get into here, but it works right now. Augur's live and you can see these markets are resolving correctly. If you report an incorrect outcome, then you have money at stake, that you lose if you do so. The protocol has a way of also adapting and there's an arbitration process that happens.

[00:49:27] JJ: How fundamental are self-governing blockchains, Haseeb and Alok, like the OCaml one by Tezos?

[00:49:35] HQ: Yeah. How fundamental are they?

[00:49:37] JJ: How interesting or how interesting are they these days?

[00:49:39] HQ: I think they're very interesting. I'm personally very skeptical of what's called onchain governance. Onchain governance, judicially the way that –

[00:49:46] JJ: “Onchain”.

[00:49:47] HQ: Yeah, “onchain”. So the way onchain governance is kind of defined is like an opposition to the way that Bitcoin or other normal open source projects are governed, which is like people on a Google group run a news list who basically are like –

[00:49:58] JM: There's like the foundation maybe and benevolent dictator.

[00:49:59] HQ: Sort of trusted emissaries in the community. Exactly. Benevolent dictator kind of model or benevolent oligarchy, whatever.

[00:50:06] JM: Benevolent apostles.

[00:50:07] HQ: Benevolent apostles, exactly. So the onchain governance model says, “Okay. We're in crypto land. What are we doing with all these gross meat space stuff? Let's instead enshrine in the protocol some kind of vote. Of course, we can do vote entirely onchain. So in the protocol itself, and then enforce the outcome of that vote also onchain.”

For example, in Tezos, one of the things that you can do, and that in fact I believe they might have already done or they're about to do, is they're going to do a protocol upgrade. Normally, to upgrade your software, you have to like click, “Yes. I'm willing to upgrade my –” Accept the thing. What Tezos does is it runs a vote, and that vote, everybody's voting on the hash of this new software binary. If everybody agrees, all of their clients, in order to follow the protocol, they automatically download and apply the upgrade themselves, and this avoids a lot of the problems and coordination that have happened with Bitcoin or other cryptocurrencies in general, where some upgrade is recommended, but like not everybody upgrade.

[00:51:01] JJ: So it disaggregates the like governance logic evolution. The governance logic evolution shifts to some automated protocol embedded in the block chain as supposed to living at the level of humans that need to codify that and like roll it out to voting.

[00:51:19] HQ: Yes. I mean, ultimately everything lives the level of human, because humans are voting. Humans are agreeing to install the software in the first place, but the record of it is there onchain.

[00:51:24] JJ: But unpack the nuance of how onchain governance works, again. Again, this is like we're super deep in the vortex here. Just try and simplify it for people who – People understand democratic voting. People don't understand onchain governance for crypto. What are the differences? There's a reason why I'm asking this question. It dovetails immediately to source available licenses, which is a big thing with open source.

[00:51:44] AV: What are the things that people can vote on? For example, one things you can vote on is an inflation schedule. The rate at which new coins are emitted by the protocol. So there'd be a motion to change that and it would be a couple of lines to edit.

[00:51:57] JJ: How many things do you mint.

[00:51:59] AV: Yeah. How many tokens get emitted per second? What's the shape of the curve if it's something that asymptotes –

[00:52:03] JJ: Allocations maybe for the –

[00:52:05] AV: Yeah. So the idea is the implementation would be a couple lines of code that you would change and then there'd be a vote onchain and there'd be a record of who voted for what. Then if it's done in the way that the protocol specifies and the change happens automatically, and then however many blocks later you would start getting the new issuance schedule.

[00:52:25] HQ: I think a good way to think about this is like imagined that we had an organization, and basically like I am the benevolent dictator and you guys were – We're going to discuss things, and then when we agree, I'm going to go implement them and then we'll push everything out, right? That is basically how most cryptocurrencies are governed.

You can imagine a decent analogy for what Tezos and other onchain governance systems are doing is they are like, “Okay, let's adapt these very systematic bylaws that basically enforce everything that can be done and exactly when it is executed.

[00:52:53] JJ: Rights management is effectively what it is. You have the right to do certain things, and that's codified in the governance model logic of the chain.

[00:52:58] HQ: Right. At a certain point, we can't press the brakes anymore. The onchain governance just takes over, and I tend to think that if you think your governance model is already great, codify it. Make it make it run forever. Awesome. I tend to believe that we are so early in the development of all these systems, that onchain governance ends up leading to more problems and more inflexibility that it gives in benefits –

[00:53:20] JJ: Oh! So you're saying we haven't yet reached an understanding of what the right rules should be basically, and we don't know that. So here's the reason why I asked that question. That was insanely informative. Incredibly informative, and it didn't get more insane. This is very valuable response.

The reason I asked that question is actually referring back to why I used the word synesthesia or synthesis in texting Jeff about potentially doing this podcast.

[00:53:45] JM: Those are two totally different things, by the way.

[00:53:47] JJ: I know they are, but I'm – I know. Synesthesia is like colors and sounds and stuff.

[00:53:53] HQ: Synthesizer.

[00:53:55] JJ: Yeah. Sorry, guys. Home-schooled here. Please, don't be so hard on me. So open source licenses basically codify rights, codify what you can do with the software along primarily for dimensions that basically go back to Richard Stallman free software. What you can see, so the source that you can see, source that you can mutate and augment and modify.

It's typically across all the source code that you can see. Where you can run it and how you can run it, where you can execute the code after you've compiled it and built it and so on. Then how and where you can I distribute the same code or modify versions of the code. So those are four verbs or actions that you can take that involve very complex multivariate rights, codification and legalese, which is what we have 8,000 open source licenses. But some industry standardization on things like Apache, MIT, BSD and so on after 20 years in. Nowhere near the level of clarity that we have on the governance logic and rules in the crypto ecosystem. That's where things start to overlap.

However, what we're now with the growth and the materialization of commercial open source as a category is a lot of companies I believe, and I've said this publicly before, wrongly mutating and applying the standard cornerstone licenses at the layer of open source projects that create the most value. So if you apply one of these source available licenses that have discriminatory constraints against certain constituents, like cloud providers, you radically reduce the surface area potential of distribution and contribution and a variety of other things, I believe, to a given open source project.

I'm not saying that source available licenses and sort of like license innovation that isn't open source is a bad thing. If you apply it at a certain layer, is actually good, and that's a bit of a tangent and separate thought, but the reason I brought it up is what I've noticed is a sort of cross-pollination of kind of precedents and understanding the clarity open source world has around licenses is so eerily similar to basically like economic kind of constitutions that exist in the blockchain ecosystems, because they're all sort of like their own fragmented constitutions. These are the laws that exist to coordinate where the incentive structure lives. How it's supported and so on.

[00:56:09] JJ: Okay. We got to wrap it up, because we only got a couple more minutes. But let's give Alok and Haseeb each your closing thoughts on that ties between –

[00:56:17] JJ: I'll stop there. That's basically what fascinated me, and I think that's something that is worth thinking about.

[00:56:24] JM: So closing thoughts on the ties between open source and/or commercial open source and, specifically, the licensing issues, if you heard about them, and crypto networks, crypto incentives.

[00:56:36] AV: Okay, a few thoughts. One, this is probably a topic for a longer discussion, but I actually think commercial open source today is pretty different from open source, 15, 20 years ago in the sense that I think it has much more to do with businesses preferring open source for reasons related to things like vendor lock-in in forms of social proof and validation rather than kind of the actual shape of development are a kind of a grassroots community around it. So that one.

I actually think that crypto, and kind of originally, there's like the cathedral and the bazaar was the analogy, and I think that commercial open source looks more like cathedrals than bazaars today. But I think that crypto looks more like the bazaar in terms of like Ethereum right now, as bazaar as you can get. So I think crypto open source contribution today looks more like open source contribution did in the 80s and 90s. That's one.

The second thing is I think that what's valuable is a seat at the table and actually seeing how these projects evolve. So when I think about – I mean, right now, they even have steering committees for open source projects. How do you get on the steering committee? You build it. You sponsor it. Then you have informal steering committees, right? These are basically like largest customers of the commercial vendor. They have kind of proxy voice in what gets adapted in the project.

I think one thing that's different here is that you could have a more legible, transparent process for having a seat at the table in crypto for determining what happens. I think that things like tokens that have governance rights are an important part of that.

So, when I think about what are the incentives that we're changing? I think that having control over something is ultimately can be valuable, and that's something where crypto from an organizational structure piece can do something different. So that's kind of where I'm more optimistic about how do you provide kind of upside control to developers in a way that you don't have with our traditional open source paradigm.

[00:58:40] HQ: What I see is the trickiest sort of trying to thread the needle for crypto is that it's certainly the case especially when it comes to commercial open source, that one can be very explicit about the commercial ties and the companies that are being built on top of it and the different stakeholders that whose needs need to be addressed in developing the right kind of software. For crypto, it's actually very hard to do that, because crypto, on one side, trying to be a software project that is infrastructure for other software projects, but at the same time there is this competing interest of trying to be money and trying to be a decentralized cryptocurrency that is not beholdng to anybody.

[00:59:15] JM: [inaudible 00:59:15].

[00:59:17] HQ: Exactly. If a cryptocurrency basically shows its hand like, "Hey, we're actually a company, and then we're trying to make money." Because of all the regulatory constraints and the fact that like there's this certain de-legitimization that happens when you suddenly admit that like, "Okay, I'm not trying to be money for all human beings in the world. I'm trying to serve my actual end customers." That ends up harming the ability of a crypto currency to be viable in, right now, the marketplace of crypto.

So I think a lot of the incentives we see right now are kind of – They're particular to this time and place in the development of the crypto economy. As 5, 10 years from now, I imagine it will look quite different. I imagine we'll evolve more towards something that looks more similar to the open source community for things that are decidedly not money or decidedly not money-like. Right now, it's still early enough that everybody can kind of claim, "Hey, I'm I money too." Eventually, that will just sound less and less compelling.

[01:00:14] JJ: It seems like that could be something holding a lot of this back.

[01:00:17] HQ: I agree.

[01:00:17] JJ: Why be all things to all people when you can actually build something the incredibly valuable solving a specific pain point and use case? That's the thing with open source, is like open source projects are built for very specific reasons, for applications, for use

cases and then they grow and evolve from there. Whereas these blockchains is sort of like, “Hey, here's my new Merkel tree thing implementation,” and, “Oh! it's money,” and “Oh! You can run a DNS server on it,” and “Oh! You can also solve all of the world's economic problems.” That just seems like a slippery slope.

[01:00:49] HQ: Completely agreed. That's one reason why I think – Well, we can just leave it on this. But that's one reason why I think layer 2, which is sort of like synthetic software built on top of a layer 1 base blockchain is a very compelling architecture where you can just sort of separate out the needs of different layers of applications without having to have the base layer be all things to all men, which I don't think most base layers can be.

[01:01:10] JM: All right. To be continued. Guys, thank you so much.

[01:01:13] JJ: This is a lot of fun.

[01:01:14] JM: This is really fun.

[01:01:13] HQ: Yeah, this is great.

[01:01:14] AV: Thanks.

[01:01:14] JM: Awesome! Okay. Great.

[END OF INTERVIEW]

[01:01:19] JM: Podsheets is open source podcast hosting platform. We are building Podsheets with the learnings from Software Engineering Daily, and our goal is to be the best place to host and monetize your podcast.

If you've been thinking about starting a podcast, check out podsheets.com. We believe the best solution to podcasting will be open source, and we had a previous episode of Software Engineering Daily where we discussed the open source vision for Podsheets.

We're in the early days of podcasting, and there's never been a better time to start a podcast. We will help you through the hurdles of starting a podcast on Podsheets. We're already working on tools to help you with the complex process of finding advertisers for your podcast and working with the ads in your podcast. These are problems that we have encountered in Software Engineering Daily. We know them intimately, and we would love to help you get started with your podcast.

You can check out podsheets.com to get started as a podcaster today. Podcasting is as easy as blogging. If you've written a blog post, you can start a podcast. We'll help you through the process, and you can reach us at any time by emailing help@podsheets.com. We also have multiple other ways of getting in touch on Podsheets.

Podsheets is an open source podcast hosting platform, and I hope you start a podcast, because I am still running out of content to listen to. Start a podcast on podsheets.com.

[END]