EPISODE 770

[INTRODUCTION]

[0:00:00.3] JM: Ethereum allows developers to run decentralized applications, but the tooling for building and managing these decentralized applications is immature. Experienced software engineers have difficulty getting started with writing Ethereum applications, because the stack of tools is so unfamiliar and different than traditional software tools.

Whether or not Ethereum itself succeeds, developers in the future we'll probably be building some decentralized applications. We will be treating money as a first-class citizen on the internet and architecting software that transfers financial value as easily as we transmit JavaScript today.

Web 3 will be a world in which many more software applications will be possible. As we move towards Web 3, many new tools will be built. Web 2 was the result of Ruby on Rails, Amazon Web Services, the iPhone. There were so many other software tools that made it easier to deploy web servers and consume internet services. In the world of Web 2, we saw innovations like Airbnb and Uber and Netflix. It's hard to pick one specific technology that enabled these kinds of companies.

In the world of Web 3, we will see new types of gig economy applications, sharing economy platforms, content companies, social networks. These new applications will arrive gradually as the tooling improves and they'll make it easier for developers to hack together businesses and side projects and build on cryptocurrencies.

Brian Soule is the founder of ETHSimple, a company that makes tools for Ethereum developers and other Web 3 application developers. Brian joins the show to talk about the state of cryptocurrencies, the tooling that developers have access to and his company ETHSimple.

We cover high-level ideas such as Bitcoin maximalism. We also talked about more technical areas of the Ethereum ecosystem, such as the Ethereum name service. Brian was also the first person that I ever worked for. We have a bit of a personal relationship and it was really fun

talking to him. I have a lot of respect for him, as you will you will hear in the show, because I have a hard time working for people. Brian was somebody I was able to work for. I hope you enjoy this episode with Brian Soule.

[SPONSOR MESSAGE]

[0:02:37.1] JM: DigitalOcean is a reliable, easy-to-use cloud provider. I've used DigitalOcean for years, whenever I want to get an application off the ground quickly. I've always loved the focus on user experience, the great documentation and the simple user interface. More and more people are finding out about DigitalOcean and realizing that DigitalOcean is perfect for their application workloads.

This year, DigitalOcean is making that even easier with new node types. A $15 flexible droplet that can mix and match different configurations of CPU and RAM to get the perfect amount of resources for your application. There are also CPU-optimized droplets, perfect for highly active frontend servers, or CICD workloads.

Running on the cloud can get expensive, which is why DigitalOcean makes it easy to choose the right size instance. The prices on standard instances have gone down too. You can check out all their new deals by going to do.co/sedaily. As a bonus to our listeners, you will get a $100 in credit to use over 60 days. That's a lot of money to experiment with.

You can make a $100 go pretty far on DigitalOcean. You can use the credit for hosting, or infrastructure and that includes load balancers, object storage, DigitalOcean spaces is a great new product that provides object storage, and of course computation. Get your free $100 credit at do.co/sedaily. Thanks to DigitalOcean for being a sponsor.

The co-founder of DigitalOcean Moisey Uretsky was one of the first people I interviewed and his interview was really inspirational for me, so I've always thought of DigitalOcean as a pretty inspirational company. Thank you, DigitalOcean.

[INTERVIEW]

[0:04:44.3] JM: Brian Soule. You're the Founder of ETHSimple. You are also the first person that I ever worked for. Welcome to Software Engineering Daily.
[0:04:50.6] BS: Happy to be here.

[0:04:52.4] JM: I want to start off by talking about internet money and what you're working on with ETHSimple. Eventually, we'll talk a little bit about the past because I think you and I were working together from an early age. In terms of talking about the present, internet money; in the early days of the internet there were ideas around being able to pay people through the browser. This is the same idea that has gradually come to fruition with cryptocurrencies and I've report on this stuff as much as possible. Why didn't that happen back in the 90s? Why didn't we have internet money back in the 90s?

[0:05:30.6] BS: I think that's a great question. In my opinion, I believe it's because we didn't have a truly trustless notion of a distributed ledger. Some of your listeners are probably familiar with distributed ledger technology. It really comes from a cynical perspective of game theory and an evaluation of what the worst actors can and will do, because that incentive will always incent people to take advantage of the system. The distributed ledger solves that problem and that's why I think Satoshi created stage one of this open sourcing of money in the application layer that we're seeing take place over the next five to 10 years.

[0:06:12.6] JM: With Ethereum, there was the second community that developed significant network effects and significant adoption. More recently, there's been some skepticism of Ethereum. There's been some skepticism of the multi-cryptocurrency vision. Are you a maximalist in any regard, or do you have any affiliation with Bitcoin maximalism, or Ethereum maximalism?

[0:06:40.3] BS: I'm not a maximalist. I mean, I signed up for a Coinbase account in 2011. I saw the thing on Hacker News when someone bought a pizza for 20,000 Bitcoin or whatever it was. I always have been interested in Bitcoin, but the thing that really piqued my interest was when Ethereum came out and you could actually program on top of it. That opened up the possibility for this distributed application stack, where you can have software that can run in a transparent way. You can see all of the logic that goes into it and it can be – you can trust it a 100%, because there's nothing left the choice, which is in contrast to the existing Web 2 conventions

where you have a rail server, or some private server that creates APIs, you most likely don't know what's going on behind the scenes.

You just have to trust that they're doing nice things with your data, which of course a lot of people aren't these days. I think there's this Cambrian explosion of all kinds of experiments in different blockchains. Some Bitcoin maximalist are sympathetic to the view that all technology will eventually come back to be incorporated in Bitcoin itself.

I think for certain reasons, there are trade-offs that just can't be made by a Bitcoin. Bitcoin is great. I love Bitcoin. I love that they have the store of value first mover thing, but I think in terms of executing the computing layer, the community probably won't rally around creating efficient Turing completeness. I do believe there's room at long-term for all kinds of interesting blockchains to come out. That makes it more entertaining for folks like us.

[0:08:25.1] JM: What you said about Bitcoin not being a good Turing complete system, couldn't you say that about assembly language? Assembly language is not a great Turing complete system to work with, but it's comprehensive enough to be used. I mean, couldn't you just build second layer solutions on top of Bitcoin that would just look at Bitcoin like assembly language?

[0:08:51.0] JM: I mean, that's a valid perspective. Some say that it's a requirement for things like exchanges that – you implement something like sharding for Ethereum that will let you persist state in a distributed but fast way. State channels would be great for things like games and anything that doesn't need a universal set of state. Bitcoin just doesn't seem to have the ability to adopt rapid things like sharding and stuff like that. I'd actually think Bitcoin's – I think it's great the way it is. They have so much security based on all the mining that's going into it.

It would be extremely difficult to 51% attack Bitcoin right now. That's a valid perspective and I think there's going to be so much creativity going into layer 2 applications. Potentially, Bitcoin could be this universal Merkel route of security that – Yeah.

We'll see. There's a lot of creative chaos going into it right now and there's a lot of developer and financial economics that are going into it. Let's stay tuned. When sharding is coming, we hear that in Ethereum. When sharding does land this year as they're saying, it'll be really

interesting to see how developers, like existing applications benefit from that speed, because as a developer right now, it's an interesting UX problem to sit there and think about if a user that has never used a theory before comes in and they query a contract and they have to wait for a central blockchain to clear the transaction, how do you convey that to the user? Is that too much of a hurdle?

Part of what we're doing with ETHSimple is enhancing the user experience by creating a domain that points to their wallets. You don't have to share a hexadecimal address, right? We'll observe and I think we may be in a little bit of the trough and in the financial markets, but I think creatively, things are just getting started. It's going to be a very interesting time.

[0:11:02.9] JM: Yeah. Okay. I completely agree with the whole creative destruction thing. We'll see on the second layer solutions. On the Ethereum front, sharding – so I've done some shows about sharding and sharding clients and Casper. I found these things hard to understand. There was that – there was a post by a guy named Tour Demister a couple weeks ago, that a lot of people were paying attention to, because he's a crypto-influencer people that has I think a fund. I've heard other podcast with him and he's a very smart sensible guy in most of the things I've heard him say.

He had a post that was fairly critical of the Ethereum community, not in a insulting way. Just listings some facts and some anecdotes about the development pace, and perhaps the over complexity, perhaps the design strategies of things like Casper and sharding. You as somebody who is building a business that is related to Ethereum, do you feel like you need to have a close understanding of the pace at which something like sharding or Casper is moving? Because you said, it's going to happen later this year. Do you actually believe that? Do you even have a sense for how fast it needs to move, or where in the development cycle it is? Because for me, I have literally no idea what the progress is.

[0:12:37.0] BS: Yeah. I think, one of the pieces of value I bring is like a product director for my company is evaluating where is the most developer excitement in the ecosystem? I think developer interest is a leading indicator for where we should invest in creating tooling for them to build distributed applications. I think Ethereum still right now is the leader for that.

I co-office with the Lightning folks and I think there's a lot of excitement around Lightning and I think that's very cool. Part of my focus is on user experience. What I really bring to the table is evaluating how do you look at an existing – a normal Web 2 user, how do you think about – how do you empathize with what they're thinking when they encounter crypto for the first time? How do we make it easy for them? Is there some way we can abstract away, opening a Web 3 injection tool, getting a browser extension like meta mask or something like that?

Is there a way we can bootstrap that to get people on quicker? Is there a way we can get them to their aha moment sooner to where they're just playing around with crypto and just having fun? I think that's one of our main goals.

In terms of the scaling features, there's enough of a forcing function that this will come along. Something I'm a little bit optimistic about is humans traveling to Mars. We've got this forcing function with interplanetary travel going through cosmic radiation might provide a forcing function for humanity to cure entropy in DNA. If you're being bombarded with radiation, if people are constantly traveling back between planets and we've got to say, how do we retain the original state of your genome? How do we capture that data set and rejuvenate humans to their younger self to reduce aging? I think we've got the exact same thing in crypto.

All these people have a need to send money across borders to create trustless, free and open applications. It's just a matter of time until you can get Visa scale and you can go beyond that. Bitcoin could be the core root node, like Ethereum could be the big number two. We'll have to wait and see. Regardless of what happens, we're going to be there building the absolute best tooling for name services and making it super easy for normal folks to come in and have a great time, as well as enhance their lives playing around with crypto and having a sovereign store value and all that good stuff.

[0:15:14.2] JM: Okay. With you on the long game. I think, doubling down on your product expertise and your product development leadership as opposed to getting down in the weeds and trying to figure out where Casper is, or where sharding is, or where debates between core Bitcoin infrastructure people and core Ethereum infrastructure people, who scored points on Twitter today. That's probably less important for your professional success than actually thinking

about the product and your own durable skills and the durable skills that your company can develop.

We'll get into ETHSimple and what your business actually does. I want to start with just – in order to get there, I want to talk a little bit more about the Ethereum ecosystem and give some context. First of all, sending ether. We've done a lot of shows about Bitcoin, we've done some shows about Ethereum. Why do people need to send ether to each other?

[0:16:20.2] BS: Well, I mean, take one example. Last week we had an investor. He's like, "Looks great. Let's do it. Where should I send the money?" I said just send it to ethsimple.eth and 10 minutes later I get a transaction confirmation. "All right, your money is sent." That was it. I didn't have to share a hexadecimal address. I didn't have to share a QR code or anything.

Investing, like investments, buying coffee, paying for fairies, potentially we'll see how productization comes along for things like mortgages and advance things like that. The question is why wouldn't people want to send crypto to each other? Your followers may be familiar with Maker, which created a stablecoin called Dai, which is really interesting. They basically created a distributed organization that has an incentive to peg the price of the dollar to Dai, or vice versa.

If the price isn't pegged to the dollar, then the DAO participants get penalized, so they have this strong incentive. It's worked perfectly so far. It's gone through a lot of stress tests, obviously going through the bear market. There's been a lot of drawbacks. This completely distributed trustless organization of random people has performed perfectly. That's just one example, but as engineers and as product people, you think about what can I make to make people's lives better?

One of the more interesting things about the open source web 3 convention is how can I do the same thing, but involve other people? How can I make a protocol based on this? It's like when you were kid on the playground and you were – I'm inventing a game like, all right, no one can touch the ground. The ground is lava. Everyone's like, "Oh, that's a fun game. All right, we're going to walk around on tables and monkey bars and stuff."

If you can go out and create a fun game for people to play, they'll play with you. You might start something interesting like a distributed domain system, or a distributed hedge fund or something cool. If your idea is not so great, then people won't want to play. They'll be like, "Oh, I'm not playing." Yeah, how do you –

[0:18:38.1] JM: Because your rules are terrible.
[0:18:39.3] BS: Yeah. Make up some interesting games. We've got all this experience building user interfaces and building cool products. How do you add a little bit of humanity to that? How do you add a little bit of game theory? Make something in this new open source, trustless world that we're all working in.

[0:18:57.4] JM: Exchanging value is core to that. Exchanging value is core to most games that are worth playing and having primitives for doing that. That's why we need ether. That's why we need to be able to transfer ether from one person to another.

[0:19:12.7] BS: Yeah, exactly. I think the crux of the value transfer is you can't have games without stake. You can't protect against civil attacks if you don't have stake. If you just have a game where anyone can query an activity unlimited number of times, they're going to do that and they're going to kill you – they're going to DDoS your game.

Value stake and value transfer is core to creating anything on these systems. If ether and Bitcoin having value, we wouldn't have this. Maybe that's one of the more less well thought causes of all this Web 3 activity. I don't know. What do you think?

[0:19:49.2] JM: Sorry, what's the question?

[0:19:50.6] BS: The fact that people have skin in the game, one of the unappreciated causes of the crypto revolution.

[0:19:57.9] JM: I mean, I think it's definitely core to it. Well, there's skin in the game on a bunch of different levels. It's on the level of people buying your token and people who have started something and then issued themselves tokens that are only worth money if the protocol does well. Although most of those token issuances, they've set up schemes where they can make

money even if the token doesn't do well, or at least those were the schemes that were coming out last year.

Let's talk more about the nuts and bolts. If I want to send ether to somebody, I need to know that person's address. What are the problems with the addressing system? Compare it to PayPal, for example, where all I need to know is somebody's e-mail. That's pretty easy. I know a lot of people's e-mails. Most the people that I would want to send money to, I know their e-mail, so it's pretty easy for me to send money over e-mail. Other people I can just send money over SMS. I think I can do that with Venmo maybe. What about with Ethereum? What's the payment transfer system like in Ethereum?

[0:21:05.3] BS: The initial way was someone sends you – I don't know. I forget how many digits it is. 32 digits. You send them a hexadecimal string, so it's a lengthy string to basically say that there's more strings than there are atoms in the universe or something like that. It's a long hex string that is basically impossible to memorize. Yeah, you basically have to share that. Some people package them into QR codes to make them a little bit more digestible.

Yeah, basically you send that to somebody over a text or an e-mail. They'll send a test transfer, maybe a dollar or something and make sure it works. Then you send it again. It was a little bit delicate. Part of the initial vision of Ethereum was will create a distributed domain system, which I think we'll get a little bit more into later. Yeah, that's how it happens now. It's the same way with Bitcoin.

[0:22:03.3] JM: Is this analogous to DNS records, where you have these numerical things that point to the actual server where something is hosted and you have this network of DNS. Then over the top of it you have actual website addresses that people enter into and those website addresses get aliased to DNS addresses, so that we have a better user experience? You can go to facebook.com, instead of 151.256.34.715? You can actually enter human readable address. The Ethereum had a vision for having the human readable addresses for addressing wallets, or addressing other areas of value that would overlay this lower level wallet addressing system.

[0:22:53.2] BS: Yes. That's exactly right. I was very young at the time, but I assumed people would share IP addresses to visit each other's blogs, or send each other e-mail addresses. They

would send each other e-mails over ARPANET, whatever it was at the time. It's exactly the same.

Domain systems have come along so much that you don't even see a notion of IP addresses in most development tools anymore. You don't actually have to include an additional library to be – resolve this domain to an IP address and then wrap that in a function and then it consumes the IP address itself. Domain systems are our first-class in everything. There's no reason that this won't happen to crypto.

Yeah. We've got a little bit of a rough edge in crypto right now, because we're in year two of Ethereum. Bitcoin has been around for a while, but it hasn't iterated quite as quickly. I believe in the long-term, we'll see the same forcing functions apply and we'll think it was amusing the day that we actually shared hexadecimal addresses.

[0:24:01.4] JM: Here you say this, I'm actually realizing how big of a problem this is, because so I'm a pretty technical user. I have not tinkered with the crypto ecosystem that much, but I did go through one exercise which was I did at gitcoin issue. I was like, I'm going to go through the steps to do a gitcoin issue, which is you post bounties that are associated with git issues.

I had a git issue for an open source repository that I maintained. I went on gitcoin and I had to install meta mask, which is a browser extension that allows you to integrate crypto into whatever you're doing on the internet. Setting up meta mask was annoying and weird and did not feel like a Web 2.0 experience. It felt something else. I didn't know what the heck this thing was, meta mask. It involved me doing something with an Ethereum wallet.

I think I had the Ethereum wallet setup before and I didn't have to set it up when I was doing meta mask. Meta mask was a great onboarding experience, I think relative to some other crypto tools. It was still pretty good experience. I made it through the whole gitcoin thing. I eventually had my bounty awarded to somebody. Somebody actually did claim it. The whole UX of the thing, I was – It was arduous. I think about somebody else with much lower tolerance for internet pain, or for a technical pain, they would have given up at step one in the funnel and I made it through step 35 in the gitcoin, meta mask, Ethereum funnel. This is a good experience

on using Ethereum, at least that was my perception of it a year and a half ago when I did this gitcoin issue.

Thinking about that, like thinking through okay, what actually are – I'm sure you've thought more about this than I have, but the actual hurdles of UX that need to be gone through to accommodate the average user. One of the earlier hurdles is this wallet thing, because I'm sure there are plenty of users who have just turned away when they're like, “Oh, my God. What is this wallet address thing? I've never seen anything like that. It looks like a virus. I'm not going to even highlight it, because it's a virus.”

[0:26:20.4] BS: I mean, yeah. User education, I believe is we shouldn't downplay the tasks before us. I believe that we did have probably a good amount of user education when the web first came out. I remember CompuServe when I was a kid. It had the internet and link. It was a strange place. I believe at some point, people will develop their product and will develop the products and tools to make it really easy for people to create their mnemonic.

Some of you are probably familiar with the mnemonic; it's a string of words that's essentially just entropy that is impossible to guess. That's basically your key. That's your sovereignty. That's what makes you – that's what makes your data different from Facebook. Facebook, they've got all the keys to all of your data.

There is some pain with basically creating a 24-word string, right? The benefits are huge. I believe that we can create products that will make it easier for people to not only create the mnemonic and store it, but – I've got a vision of and if anyone is listening, please do this, but a vision of some product where you go on the internet and you click buy with Apple Pay and they send you a label printer, that gets to your house and you push a couple buttons to create some entropy and it just prints out a label of 24 words, which is your mnemonic string. Then it has a nice box that you can put it in and you put it in your closet and then there's your sovereign identity right there. You can put it in a bank vault if you want.

It's a big shiny box and it – is commensurate with the value that it provides to you. You want to keep it safe. You don't want to share it and yeah, you definitely don't want it to leak. When you have this germ of entropy, I believe money is the initial thing. If you follow along with the broader

implications of this sovereign identity, you can use this key to encode anything. You could encode – you can use this to unlock your accounts on distributed services. You can use it to – you could theoretically use it for a distributed social network or something like that.

When you have this key, like the application developer can never see your – they can never unlock your data. You're the only person that can. They just create the product. When will we get to a world where none of the application developers know what their users' data is? I think that's the end play here.
The start is obvious, it's money. It's the most compact data structure, so it was required to prevent simple attacks. It was the first thing. I believe distributed name services are next and then just potentially the entire application layer can become – be a target for becoming open source.

When we have products that are – the UX is well-known and it doesn't take a lot of fast-moving product experimentation, I believe there will be a long-term forcing function to create open source versions of these products. An open source Twitter or something like that, they basically have a distributed micro-blogging platform. Since something is open and belongs to everyone, you won't have the bandwidth to experiment a lot, because there's a lot of difficulty in running that pass a community and being like, "I proposed changing this button, or adding a feature, or something like that." If you have to get all of the participants to vote or something like that to accept a change, you'll be a little bit more slow-moving.

We can draw parallels to the open source operating system world early on. I think about Windows is being first. They were the fast mover. They got to market. Their motto was a computer on every desk. They did that amazingly. They got Windows to every desktop. They created this awesome platform for developing applications. I'm not sure how much of it is appreciated that they did that, but you could create a visual basic app, boom it could be on all these desktops. They were first.

In my opinion, the second wave is yes, but we want it to be open and we want it to be better. We want it to be trustless. I believe for a consumer that was Macintosh, they were largely built on UNIX. All your Linux commands work in a bash and stuff like that, but there were some

problems with like, we have to produce hardware in China and stuff like that. I think it still needed to be sponsored by a corporate entity.

Yeah. I do believe there's a long-term forcing function to apply the trust and the warm fuzzies to stuff that we use everyday in the application layer. Facebook and Twitter and banking and obviously, name services, money, I don't know. Who knows? Someday there might be an open source, Uber or something like that. Who knows?

Once the protocols get quicker and the UX hurdles are solved, just imagine what will happen. I think it's incumbent on us as engineers to think of the next cool thing that we can build on Web 3 and experiment and make it happen.

[SPONSOR MESSAGE]

[0:31:58.4] JM: Kubernetes can be difficult. Container networking, storage, disaster recovery, these are issues that you would rather not have to figure out alone. Mesosphere's Kubernetes-as-a-Service provides single-click Kubernetes deployment with simple management, security features and high availability, to make your Kubernetes deployments easy.

You can find out more about Mesosphere's Kubernetes-as-a-Service by going to softwareengineeringdaily.com/mesosphere. Mesosphere's Kubernetes-as-a-Service heals itself when it detects a problem with the state of the cluster, so you don't have to worry about your cluster going down. They make it easy to install monitoring and logging and other tooling alongside your Kubernetes cluster.

With one-click install, there's additional tooling like Prometheus, Linkerd, Jenkins and any of the services in the service catalog. Mesosphere is built to make multi-cloud, hybrid cloud and edge computing easier. To find out how Mesosphere's Kubernetes-as-a-Service can help you easily deploy Kubernetes, you can check out softwareengineeringdaily.com/mesosphere, and it would support Software Engineering Daily as well.

One reason I am a big fan of Mesosphere is that one of the founders, Ben Hindman, is one of the first people I interviewed about software engineering back when I was a host on Software

Engineering Radio. He was so good and so generous with his explanations of various distributed systems concepts. This was back four, or five years ago when some of the applied distributed systems material was a little more scant in the marketplace. It was harder to find information about distributed systems in production, and he was one of the people that was evangelizing and talking about it and obviously building it in Apache Mesos.

I'm really happy to have Mesosphere as a sponsor. If you want to check out Mesosphere and support Software Engineering Daily, go to softwareengineeringdaily.com/mesosphere.

[INTERVIEW CONTINUED]

[0:34:17.8] JM: Isn't it amazing how many engineers there are that still “don't believe” in crypto, where they'll just say, “I'm skeptical. I'm a crypto skeptic.” You understand that this does allow this trusted system, it allows this self-sovereign identity. It's a set of computer science primitives that were put together to invent something. They'd be like, “Yeah, I'm skeptical.” No, no, no, but do you get it? This is a building block. This is like somebody said, “Here's the internet browser.” I get the internet browser is a set of things that were put together that existed previously, but the internet browser is wholly new, as a new artifact. That's what we've done with crypto; this is a building block. People are going to do crazy stuff with it.

It's just amazing that there's still universals, like universal skepticism. Why do you think that is? Do you think that's willful skepticism? Are people just – are there some people who feel they missed out, and so their way of coping with missing out is to just basically reject the idea that crypto is actually a real, important building block?

[0:35:28.9] BS: It's possible. I think I think there is value in the criticizers.

[0:35:35.8] JM: What's the 2019 bear case for crypto?

[0:35:39.3] BS: I think the UX is the biggest hurdle. Yeah. The bear case is a Web 2 – PayPal remains supreme, because they're attached to the e-mail address. Yeah. I mean, it's perfectly valid to respects the advantage that PayPal has in user experience right now. I do believe that

we should take those – I think, Elon talks about really learning a lot from his critics. I think in crypto, we can really learn a lot from our critics as well, even if they're perennial haters.

There's certainly a fair amount of that within the community. I've never really understood it, because I always came from the developer community, the entrepreneurial community. We make cool stuff and we improve people's – we improve things, we have proved experiences, we're not playing a zero-sum game. That's one of my few criticisms of the show Silicon Valley. I understand you need a plot, but it seems to have this zero-sum, like sometimes they hate another company.

[0:36:47.9] JM: That's why I stopped watching it.
[0:36:49.4] BS: They hate on the pizza company in the show, because it's a silly idea. It's an app that delivers pizza. Fine. As someone who's lived in Silicon Valley for six years and has always built products, I've never really gotten that vibe in the community, like people resent others. There is more than enough room for everyone to create something awesome and improve the world in a perpetual way.

I'd say, be respectful of the critics. Engage with them. They're actually helpful. I think in a long enough timeline probably, they might be using your stuff in the future. Can you imagine someone going around and telling people like computers are going to be the big thing, operating systems and spreadsheets? Spreadsheets are I guess V1 of distributed – or V1 of applications. How can we use spreadsheet – They probably seemed completely crazy, because people had been organizing their business around IBM, paper-based workflows, the Xerox machine was this great innovation in how business workflows go.

I actually have some respect for people for the way people did business back in the day based on paper. We're so used to having our worlds be run on computers and servers and computers and phones run on applications and on top of that. If you didn't have access to any of those things, how would you run a business?

[0:38:26.0] JM: The mailroom. You've got the mailroom. You don't have routers, you have the mailroom.

[0:38:31.4] BS: I don't understand how that works at all. Yes, there's some intelligence in how all this stuff ran, but it was a – I don't know. Maybe I can sit down with someone and hear about the good old days of how did they run General Motors based on – how did you validate that people actually did things? How did you have credit and how did you do financial transfers and stuff? You've probably seen Catch Me If You Can. Apparently, a guy was just run around the country making blank – writing fake checks.

[0:39:00.3] JM: Right. Yeah.

[0:39:02.7] BS: I'm sure there were skeptics back then as well. They're part of the fun. I love the critics.

[0:39:09.7] JM: There is something called the Ethereum name service. This was described in the early Ethereum discussions and then it has come to fruition. What is ENS? How does it compare to DNS?

[0:39:25.1] BS: ENS. Like we talked about, it creates an address that lets you tell people about and they can send money to. People can use our service ETHSimple, to basically do the perform the actions required to buy an address. It will do it automatically. You just basically create it once and you can get a domain.

You can also do things like speculate on .ethdomains if you were – if you draw comparisons to the early days of the dot-com system. I've heard so many people say like, "I should have bought apartments.com in 1996." Well, this might be your chance to buy some interesting domains and learn about crypto at the same time.

Part of the original vision was to create ENS. ENS is basically a smart contract that just has a key value store. It's like, these addresses belonged – these domains are belong – are owned by these addresses and you can resolve other things to it. ENS is completely on-chain, so there's basically no technology gap between your server, or your address in the domain system. It is a smart contract itself. It exists entirely in Ethereum.

There are some interesting extrapolations to that. DNS is a little bit separate from the servers that they – that applications run on. Your server could be at a data farm, it could be in your house and DNS runs separately. There's a lot of infrastructure in DNS that's been built up over the years and e-mail. DNS is a very pure, trustless, crypto domain system. Your address can't be revoked. Potentially, you can purchase the address and your perpetuity in the future, which I think is very cool.

One thing I think is really interesting about all of this crypto development is the securitization of DevOps. What is a world like, where you can you can prepay for hosting for a 100 years or something like that? What are the creative consequences of that? If I want to create a memorial website for a loved one that's passed away, potentially I could do that. Before, I host my blog on Amazon Web Services in S3 and I created an account to host my blog and I think it charges 53 cents a month or something like that. Every two years, my credit card will expire and I'll have to go and I'll get three warnings from Amazon like, “You need to update your credit card, because you're 53 cents wasn't billing.”

I’m like, fine. I could’ve go – remember what that account was and log in and add some credit card. There's no way to prepay. I can't put 10 bucks in for it to just run, right? That’s where you got amazing performance and everything, but what's it like in a world where I can say, “I want to put my asset on – I want to put it on IPFS and pay with it for Filecoin, near perpetuity.” I’m going to point my Ethereum domain to it, so people can – they can view my domain and resolve the asset completely on-chain, so there's no servers required. There's no self-managed servers required.

How cool is that when you can make a product, publish it and just go completely hands-off and not have to think about DevOps. I think that's fascinating. We're just building the building blocks right now. What is that going to look like when people actually start building on then? Are there going to be hedge funds that trade on hosting and DevOps?

I believe there are some hedge funds that do interesting things, like they'll buy an oil tanker full of petroleum and park it and wait a couple months for the prices to go up or something like that. There's interesting arbitrages. I don't know. It’s a crazy chaotic world right now. I'm really curious

to see how it all plays up. Yeah. The securitization I think is going to be really interesting. Stay tuned. Stay tuned.

[0:43:28.6] JM: Totally agree on that bright future if these – some of these, where I guess we should say when these primitives get solved, when the UX issues get ironed out. DNS, I mean, I feel like there's a lot of – I have a lot of annoyances with DNS. When I buy a domain, the experience that I get from provider to provider is highly variable. I buy my domains from Google domains now. Google domains is pretty good. I like Google domains a lot. I actually don't have a lot of complaints about that.

The previous providers I've used, well we'll not name, because they're just so bad and so egregious with what they do. I don't even understand, why they charge me so much? I paid, what is it? $12 a month, or $12 a year or something, depending on how aggressively they're charging you? What are they charging me? When I when I go to a domain provider and I'm buying a domain. I want to buy jeffsawesomepodcast.com. Why does the experience and the price vary so much from two provider to provider? What are they actually giving me?

[0:44:36.2] BS: Yeah. There's a lot of web to just incentives at play there. You can go with a registrar that has a huge marketing budget. It's like Wells Fargo and I'll explain in a second. If they have to buy a bunch of ads to get you to convert to become a user, then they're going to have to make – they're going to have to make that margin somehow and they end up charging you for it.

I think GoDaddy, or not GoDaddy. I think Google has other incentives, so they want to – maybe they just want you to be incorporated, maybe they're just tired themselves of paying for domains. I mean, they've got Google Fiber. Obviously, they've got some incentive to just make computing easier. I think an interesting thing about these distributed domain services is they're all at cost. It's you don't pay for Linux like it's an open standard that belongs to the world. It belongs to everyone, right? Potentially, this is going to be something that you don't have to.

I'm not the person of record for this, but there's an interesting backstory as to how the DNS system developed. I can in Verisign and Verisign and I'm 100% not the final authority on this, but I believe that they got a contract from the government to operate the core registry for the

dot-com system and the .net and .org and a couple others, I believe. They basically maintain this database of who owns what domains and what the name server is.

They have this no bid contract, I believe, something like that. I believe it's 750 per domain per year for the privilege of maintaining this database, with there being no competition and for bidding to do this. There's all kinds of crazy – I can get to a little bit, but they're not quite as much as Verisign. Verisign doesn't manage all the domains, so you get some cheaper ones like .info. Yeah, there's some interesting history behind all these. I do believe it's one of the forcing functions for crypto domain services to come to the fore is there's not going to be nearly as much rent, just because the infrastructure is already built.

We know the data structures, if you want to create a name service that emulates the existing domain system. There's not a lot of novel research that you have to do. We'll see. I think when you build an entire business based around a single string, that's in a name service. Imagine you're a company, you're facebook.com, think if they revoked facebook.com or something, you'd be having a very bad time. Don't you want –

[0:47:20.8] JM: Because even Facebook is beholden to Verisign, or whoever actually controls that database.

[0:47:26.9] BS: Yeah. Why wouldn't developers want an open source, trustless thing to manage their reputation and their identity? Yeah. I think a lot of DevOps people and developers have voted with their feet that they end up using Ubuntu for their servers and they end up using Linux and Linux commands on Mac to do development. We'll see how it plays out, but I believe that there's a big incentive for people to vote with their feet for these distributed domain services, some handshake.

[0:48:00.8] JM: DNS is – we don't have to go into it in detail, but it's something like I go to a DNS hosting provider and entering the domain that I want to buy and they said, “Okay, the domain is available.” It’s $12 and then they register it for me and they have some back-end stuff and I guess they're talking to ICANN or Verisign or whatever else happens on the backend when you register a domain that has not been taken by anybody. That's great. We don't need to go into that.

When somebody has taken the domain before and they own it, you can ask them – you send an e-mail. You can look up the who is, of who the – of who owns that domain, who owns softwareengineeringdaily.com. You can send them an e-mail like, “Hey, can I buy this from you,” and you can figure out a transaction for that.

Then there are some systems where you can actually set up auctions for these kinds of domains. I think from what you've said, it sounds like in Ethereum, there's an auction system built in at a lower level for these domains. Can you contrast? If we're talking about highly competitive or even just marginally competitive domain names, how does the price for a competitive domain name and how does the person who actually buys that domain name on Ethereum, how does that vary between the ENS and the DNS system?

[0:49:20.3] BS: Yeah. DNS is a first-come first-served thing. There's no auctions. The reason ENS has auctions is it was basically created to manage the land rush, that happens when a domain system gets released. When it initially comes out, people are going to rush to buy all the valuable domains. I think there's 250,000 domains or something like that registered in ENS right now.

They created the auction system to make those domains be priced fairly. Now part of the development process of ENS is there's an initial version of ENS, which I guess it has an 18th month, a two-year something like that initial period, where they let people buy the seven characters or more domains, so the longer domains. They wanted to put it out there, wait for feedback, make sure they have all the intel they want before they go to the permanent version.

Now the permanent version should come out this year, where there will be no more auctions. It'll be instant buys. The five-day auction thing will go away and you can buy the domain instantly. In terms of the second layer markets, when you go to GoDaddy and you type in a name and it's like, this is for sale, there is a notion of that in ENS, I think the tooling for to get pervasive will come along, but there's been tooling created to attach domains to ERC 721 tokens, which are non-fungible tokens, like CryptoKitties. They’re unique items.

People can buy domains, stick it into 721 token and let it be exposed by all the 721 non-fungible markets. Potentially, you could find a domain and you could find it in every resale market, every crypto domain resale market without having asked for permission, which I think is pretty cool. There's something to be said for developer economics for all these standards. ERC 20 tokens, ERC 721. I'm sure there's stuff for other protocols. I think that's one of the big features of crypto in my opinion is being able to rally everyone around a single standard, so you can have interoperability and you can basically have permission list development.

There's this interesting project called crypto dragons. Have you heard of CryptoKitties? I believe it's a Chinese project. It's called CryptoDragons and you grow your dragons by cryptographically feeding them CryptoKitties. The dragons the dragons have to actually eat these CryptoKitties to grow. You have to provably destroy the CryptoKitties and this is built on top of CryptoKitties without permission from anyone.

[0:51:59.8] JM: That's brilliant. I love it. ETHSimple. Let's talk about the company that you're building. Explain what ETHSimple does for people.

[0:52:08.9] BS: ETHSimple is our foray into making crypto user experience really – our goal is to make it better than Web 2 UX. Yeah, basically what we do is you log into our site. Right now we use meta mask in the Web 3 conventions. You query our endpoint and we manage all the subsequent auction and configuration events to create your domain for you. Normally, you'd have to do a few queries over a five-day period. Then when you win, you've got a point to a resolver contract and set up a few other things.

We do all those queries for you. Basically, you just have to query once. Then at the end of the time period, you have your domain. You can potentially hop on this simple – think of I don't know, a dozen domains or something and just query them all and you'll have a nice little portfolio going at the end of the period. Yeah, just making crypto fun.

[0:53:07.7] JM: Okay. How does that purchasing experience, if I want to buy Ethereum domains, what is that experience look like, relative to the experience off of ETHSimple, or on another Ethereum registrar?

[0:53:23.7] BS: Yeah.

[0:53:25.9] JM: Maybe you could just define what is an ENS registrar, or Ethereum registrar.

[0:53:31.6] BS: It's a little bit funny, because they call the actual registrar contract a registrar. Then we also call ourselves a registrar. Is GoDaddy a registrar and core DNS a registrar if you're drawing those parallels? Pre-ETHSimple, a lot of Ethereum people went through this problem. ENS comes out, you want to buy some domains, you have to hop on and you have to query a registrar – you have to query the core registrar contract, you have to start an auction, submit a bid, you have to follow up three days later and reveal your bid. You have to save a JSON blob of your passion, your entropy in the string point in your domain and all this stuff.

Then two days after that, you have to finalize your bid and you have to create a resolver and point all these things to it. It was a big pain. You have to be a borderline developer to do it. Even if you are a developer, you don't have time to be going around querying smart contracts over a five-day period. It's just impractical. I think 50% - it was 40% or 50% of auctions were just never revealed after the fact. That might indicate that people were creating bids and just forgetting about them, just because they forgot or they're on vacation or something.
I wasn't really buying domains at the time that DNS was new, but I assume that there were some user experience problems with buying DNS domains as well. Yeah, we need to make it super easy and fun. We've got some really cool stuff in the pipeline coming out. We want to make it really easy to start publishing content and just have fun and play with crypto. We want to make it easier to make crypto – easier to make crypto websites on Web 3 than Web 2.

It's actually still a pain in this day and age to make a website and on DNS. You got to go to a registrar or a hosting service by FTP. You have to go to your DNS manager and point it to an A record. It was actually a learning curve. It's not intuitive as to what an A record, or a C name is. Potentially, crypto could be even easier than the DNS system. It’s part of our goal, just to make it fun for people. Yeah, publish them websites on IPFS, or swarm or one of the distributed asset-serving protocols that are coming out and just yeah, have fun with it.

I'm thinking, maybe someone might want to make a memorial service that lets people publish websites for a 100 years or something like that. Send me an e-mail, we can talk, brian@ethsimple.com. I think it'd be really cool.

[0:56:22.4] JM: The internet tombstone.

[0:56:23.7] BS: Yeah, exactly. Yeah. Because potentially, a static website isn't going to be a huge amount of storage, so why not?

[SPONSOR MESSAGE]

[0:56:40.3] JM: Triplebyte fast-tracks your path to a great new career. Take the Triplebyte quiz and interview and then skip straight to final interview opportunities with over 450 top tech companies, such as Dropbox, Asana and Reddit.

After you're in the Triplebyte system, you stay there saving you tons of time and energy. We ran an experiment earlier this year and Software Engineering Daily listeners who have taken the test are three times more likely to be in their top bracket of quiz scores. Take the quiz yourself any time, even just for fun at triplebyte.com/sedaily. It's free for engineers. As you make it through the process, Triplebyte will even cover the cost of your flights and hotels for final interviews at the hiring companies. That's pretty sweet.
Triplebyte helps engineers identify high-growth opportunities, get a foot in the door and negotiate multiple offers. I recommend checking out triplebyte.com/sedaily, because going through the hiring process is really painful and really time-consuming. Triplebyte saves you a lot of time. I'm a big fan of what they're doing over there and they're also doing a lot of research. You can check out the Triplebyte blog. You can check out some of the episodes we've done with Triplebyte founders.

It's just a fascinating company and I think they're doing something that's really useful to engineers. Check out Triplebyte, that's T-R-I-P-L-E-B-Y-T-E.com/sedaily. Triplebyte, byte as in 8 bytes. Thanks to Triplebyte and check it out.

[INTERVIEW CONTINUED]

[0:58:30.2] JM: What you're describing about the internet still being too hard to use, just like setting up a website being still too hard to use. I mean, you see this in the growth and growth of the Squarespace of the world, the Wix's of the world. Wix.com sponsors this show, full disclosure. When they first started sponsoring us, I was like "Why would you market Wix to software engineers? Software engineers can build their own websites."

I think what I realized is first of all, probably there are some software engineers who just don't want to tell anybody, but they don't know how to set up a website. There's a lot of steps in setting up a website. You could be a really good developer, you could know Spring backwards and forwards, or maybe you're really good at COBOL, you have no idea how to set up a website.

Even if you follow all the how-to tips and Stack Overflow or wikiHows or whatever, it's still going to be hard for you. You do see simplified developer tools too, things like Firebase, or other things that are a little – at least lower level than a Wix or a wordpress.com, or a Squarespace. Yeah. I mean, I can imagine looking at that space from your point of view and saying, "Okay, it took the regular internet," or I shouldn't say the regular internet, but Web 2.0 this long to get to the point where we have Squarespace and Wix and it's – those are giant markets and they're growing markets. What would the Squarespace, or Wix, or WordPress experience be for an – why do we need that for Ethereum and why would it be – why would it be any different? Why wouldn't the Squarespace of Web 3.0 just be Squarespace?

[1:00:12.8] BS: Yeah. I mean, I think the goal is to let people publish their own content and just have it be outright owned by them. I remember a day when blogs were big. I do have a romantic longing for this time, because there's a little – small point in time where basically only nerds could make websites. It was great for people like us. You'd have your RSS reader and you have all these blogs that were set up. Facebook and these Web 2 companies came along and let you create dynamic websites without knowing how to code or do DevOps. I think that was the big inflection point.

We basically want to let people create their own content with their own private keys and make it easier than using a Facebook site or something like that. If we have anything to do with it, what happened to the internet after Twitter and Facebook came out won't happen to crypto. We want

we want everyone to have total control and not to rely on third party services. If you got a blog publishing tool and they're still a moat to publish crypto sites, I would say that's not really – that's not really achieving the vision of self-sovereignty and letting people own their own content.

The way we design our site, there's no analytics, there's no tracking. It's a static site. We want to know as little about people as possible. It follows the convention of Web 3 injection convention. It's interesting how – I can maybe explain this a little bit if you're new to crypto development. How it works right now is there's something called meta mask and there's a similar thing for Lightning. It's a browser extension. It injects Web 3 in every website. If you develop crypto site –

[1:02:02.2] JM: Web3.js.

[1:02:03.6] BS: Yeah, Web3.js. Yeah. Injects that in the site. If you develop adapt, you basically say is Web3.js present in this tab. If it is, then they've got meta mask, or they've got the lightning tool or whatever and you can interact with that person's identity, but that's provided by them. Basically, you create a tool, you add some conventions around like logging in and out and stuff like that, like some UI and you let people interact with your product based on their identity that's brought by Web 3.

You can see things like they're a list of accounts and you can query to submit a transaction. You basically have all the dynamism of a client-side JavaScript app with their Ethereum identity baked in. You see a lot of React development in the Ethereum community for that reason. If you've got a fully trustless DAP, it's just JavaScript and client-side code only. You can see a 100% what's in the code. Anyone can take it and they can put it on their own CDN, or they can run it locally and it'll have the exact same behavior. You're just trusting the JavaScript code in front of you that you can 100% read and audit and the blockchain on the backend that you can a 100% read and see what's going on. There's no shadowy, private servers on the backend that you have to trust. That's pretty cool.

[1:03:31.9] JM: You're suggesting a lot of different things that actually are significantly different than Web 2, that suggests that the Squarespace of crypto might not be the Squarespace of Web 2.0. Things identity as a – If you can have expectations around how to interact with a user's identity and it's not their Google identity, or their Facebook identity, it's them, it's their

identity that they are comfortable with. Because all of us, I love Google, but I'm not comfortable with Google managing my identity. I mean, I have to do it, but I'm not super comfortable with it.

I can see a path to crypto actually having an identity system that I'm comfortable with. You have the identity system, you have payments being closely integrated and all the abstractions that you could build on top of the expectation of payments and the expectation of payments and identity. Then you have just this lower-level infrastructure stuff, like ENS and just all the random crypto lower-level stuff that has tons of churn right now, but is going to develop in its own way.

You take all that together; you've got higher level components and lower level components that are different than Web 2.0, in addition to all of the stuff in Web 2.0. That's a totally different stack. You're looking at that stack and you're saying, “That's where I want to play. That's where the puck is headed.” Today you're focusing on domain registration, but the vision is to simplify that whole stack and be a trusted, simple party to be able to deal with for developers, or other people potentially who want to build or invest in that stack.

[1:05:15.8] BS: Fun.

[1:05:16.4] JM: Fun. Right. Okay. Ross Ulbricht, the creator of the Silk Road went to our high school. You and I went to high school together. You're one of the first people I ever worked for. Ross, I think about Austin sometimes. Austin as a tech city. You and I both grew up in Austin and we ended up moving out here to Silicon Valley. Austin is a great tech city, but there is a difference in entrepreneurship between Austin and Silicon Valley, where if you were to try to name off the best entrepreneurs in Silicon Valley, it would take you 13 hours to name all the fantastic entrepreneurs that have come out of here.

If you're talking about Austin, there are plenty of great entrepreneurs, but they don't have the same scope. They don't have the same scale to their ambitions, except maybe somebody like Michael Dell. That's fine. That's not a value judgment on them, but when you just look at the size of the business and the impact of businesses that people from Austin have built, it's just different than Silicon Valley.

Ross who went to our high school, the vision for the Silk Road I think was pretty big and pretty ambitious. Is Ross Ulbricht the most successful entrepreneur to come out of Austin in the last 10 years?

[1:06:33.6] BS: Well, if you value the current stake of his Bitcoins confiscated by the government, I don't even know, what are you – how many billions of dollars is that?

[1:06:43.3] JM: I have no idea. Is it billions?

[1:06:46.0] BS: I don't know. It has to be. I don't know. It's a lot. I don't even know – they might have auctioned them off at this point and they might have been better off just – they might have been better off just holding them Bitcoins for the next 20 years or something. Yeah, geez Ross.

[1:07:02.6] JM: How well did you know him?

[1:07:04.2] BS: I had some mutual friends with him. I never met him in person. I was supposed to go to – I was supposed to meet up with him once in San Francisco. Incidentally, someone invited me to a thing he was at and I didn't end up going. He was apprehended soon after that.

[1:07:20.5] JM: Were you going to meet at a library?

[1:07:22.3] BS: No. I haven't had time to look into the Ross Ulbricht legal case. There are people with way more knowledge on it. I know what I don't know and I'm sure you'd have to sit there for a week and just review all the documents and knowledge that went around that case. I don't know. Like everyone says, he seems like a very nice person. I don't know. Isn't it his last appeal to get a presidential pardon or something like that?

[1:07:56.1] JM: I don't know. I'm not up to date on it.

[1:07:59.2] BS: Yeah. He did end up moving to San Francisco at some point. I do think it's interesting, crypto started out with this Silk Road thing. Now people don't even talk or think about that anymore. It's just this obvious global monetary system. It started out like it is – people were mining Bitcoins in their house on their towers and they would actually – they would mine

on their computer and actually end up making 30 grand or something like that when the price went up. I thought that was pretty interesting. Well, we've got two Super Bowl winning quarterbacks coming out of West Lake.

[1:08:39.3] JM: Oh, that's true. That's entrepreneurial.

[1:08:41.5] BS: Aren't they actually playing each other? Are they both playing in the Super Bowl?

[1:08:44.5] JM: I don't know. I have no idea.

[1:08:47.0] BS: We did football really at West Lake. I guess, we did dark net markets also well. I see bull signals for Austin. There's a giant building downtown with the Google logo on it now. Have you seen this?

[1:09:00.7] JM: Oh, yeah. I've seen that. Yeah.

[1:09:02.4] BS: Yeah. I think, I don't know. Maybe there's a tech diaspora out of the Bay Area. Who really knows? I don't have the macro data.

[1:09:10.9] JM: Isn't that adverse selection for epic entrepreneurship? Maybe not. I've had too much Kool-Aid. I think I probably had too much Kool-Aid. I also think –

[1:09:20.3] BS: It's just bigger here. It's a bigger ecosystem. It's a lot of averages. When you have a bigger pool of talent to choose from, you're going to have more crazy outliers. Maybe per capita tech entrepreneurs in other areas are aberrationally high. Who really knows? It could be lower. I don't have the data, but that's just the nature of the creative value of large numbers of people and a single focus, getting in the same place.

If you want to do fashion or something, you go to New York City, that's just where it is. You can try to do it in other places, but you're never going to quite – you're never going to quite get at that level. I think that's what we did. I think a lot of people go to Silicon Valley and do their "masters in tech" here. You go and you work for a medium-sized company, you do your daily

stands, you do your Kanban board, you do pull requests, you learn the best practices. It really does help, because you can catch up with two decades of best practices in a year or two. You can potentially use that and move outside of the Bay Area. You can make it a great product and never have to really – have a salary again. Lots of people have done that in Austin, like Noah Kagan. There's a lot of great examples of that. I love Austin.

[1:10:45.4] JM: I love Austin too.

[1:10:46.8] BS: I feel so much better when I go to Austin and come back here. It centers things for me. Why are we doing all this?

[1:10:54.2] BS: Well, I think it'll change. Me personally, I've been looking really hard, but I don't see anything here that is not replicable by Austin. I think Austin can replicate it. I think it'll get what we have here of value will get recreated in Austin. This topic has been litigated on why to come. So many people have talked about it. I don't need to talk about it in this episode, but there are a lot of good reasons to come here right now if you are an entrepreneur. I do think that it's going to change a lot. I could easily see myself moving back to Austin. I've thought about this a lot.

I think there was some cultural thing with Austin, where you – I think it's getting easier and easier to build a remote work for us. It's also getting easier and easier to build a great company with a smaller team. These pressures make it easier and easier to build a company in a place like Austin and have that company be really successful, really valuable.

[1:11:49.8] BS: Yeah. I have a theory about the macro forces that contributed to the Bay Area in the last decade, decade and a half. This is just a theory, but when we were kids Enron happened, right? There was actually – there are a lot of crazy tech companies in Austin when we were kids. There's AMD, there is Dell. I remember –

[1:12:10.1] JM: Motive, I think. Or Trilogy or something.

[1:12:13.2] BS: Motorola was big there. I remember a lot of the grown-ups talking about their tech stocks and it was really exciting and interesting and I think it was one of the germs for me.

Enron happened and congress created Sarbanes-Oxley, which made compliance for public companies really onerous, according to some CEO. If you've mentioned to someone that you had a good day at work and they go and buy stock and you could potentially be liable for insider trading is what I've heard.

It made it super onerous to go public. There's tons of compliance cost. People aren't going public anymore. What is Uber's evaluation? 70 billion or something like that. They're still private. I feel the access to finance, access to smart finance is one of the best things that Silicon has competitively. If crypto can actually liberate smaller companies from those prohibitive costs, that could potentially be a big bull case for cities like Austin, if you're like, "I'm going to create a security token that's a claim of ownership for a software project." Wveryone who owns a token will get a proportionate share the revenue. If I own 1% of tokens, I get 1% of the revenue and I can vote. It's basically a distributed ownership of a company.

If it ends up being – if all the legality is worked out and it ends up being straightforward in the United States, that could be huge for other ecosystems. Say you're a four-person company in Austin. If you can create a compelling case and create a security token around your project in Austin and raise two million bucks or something, you can work for a long time on that in Austin.

I remember, I had a one-bed for 750 right next to the river in Austin when I lived there. It was amazing. I don't want to say when I'm paying for rent in San Francisco, but if you can just sit there and experiment on your product for four years, you're going to hit something. You're going to get some traction. There's a lot of different cases for crypto. What were you saying before?

[1:14:20.2] JM: My thesis around Austin is more of a – there's a there's cultural issues at play that lead to – I don't want to talk about it too much, because it gets into – basically, it becomes political. When I was going to UT, there's something about just the Austin coffee shops. You go in an Austin coffee shop and just let's just chill and talk to each other. Here, it's like let's go to a coffee shop and talk business and talk about how we can help each other. Those are extremes, right? Obviously there's a gradient.

In Austin you go to a coffee shop and you do talk to people about business. In Silicon Valley, you might go to a coffee shop as a date, or a total casual conversation. In Austin, there was just

– I got the sense there's a lot of people just – Anyway, I'm not going to – it's a sense of complacency. I've talked about this on previous episodes. I think it's changing. I think people are – I think that the whole startup mentality, the startup saying Y Combinator has been really great for this, just showing people that there are some cool benefits to aligning yourself as a creative business person and saying, “I want to build something. I wanted to make money. I want to have impact.” Having this agency, instead of this apathetic malaise that sometimes is cheered in Austin, and I don't think it's necessarily good thing.

I don't think it's necessarily a bad thing. I think there are times and places when people can be malaise and just chill out and just reflect on the world, or be philosophical, or lounge around, or play guitar on the street or whatever. That's all great. The Y Combinator perspective that I just described was less well represented in Austin, at least when I left. I was really glad to get away from the lack of it.

You got to go in a while, so I want to wrap this up. Give me your condensed crypto thesis. In one or two minutes, tell me how the world of cryptocurrencies will evolve in the next couple of years.

[1:16:28.9] BS: Wow. Well, they say it like it's wise to know what you don't know. There's definitely a lot of unknowns in crypto. Like we talked about before, there's the huge forcing function for making things easy. That's what I focus on. There's going to be a lot of development in scaling. If we can get transactions, so five seconds or something like that, the UX will be really good.

My contrarian perspective, or not contrarian, but less well accepted perspective is that there's going to be the open sourcing of the application layer on top of the protocol layer, which is the actual currencies themselves. I believe we're going to have this big force to create open applications in the name – there's going to be a notion of a decentralized application stack. If you can have a fully decentralized product that's totally auditable, everyone's going to gravitate towards it.

There's a feeling you get when you go to view a library's documentation or some open source product. Maybe there's a German word for this. It doesn't have the sexiest marketing page ever, but it's completely open and you can tell there's no profit incentive really in it. It's just some

nerds that makes – that maintain a library. There's just this feeling of credibility and trust that you get. You pick that over maybe a slightly slicker, closed-source thing. I feel that force can happen and will happen to the application layer, which hasn't really been possible before distributed ledgers, because there was just no – there is no DevOps method to maintain a distributed application.

Before crypto, how are you going to operate a rail server for a distributed app? Is there any way? Are you going to open an Amazon Web Services account and let people crowd fund it or something? Let random people do pull requests on it and direct the C name records? No. That just wasn't plausible. Now with the securitization of DevOps and stuff like IPFS, distributed asset stores that will store the front-end for your code, this is totally possible. Yeah, I think this is something that you should keep your eye on.

[1:18:45.8] JM: Non-fungible tokens, overhyped or underhyped?

[1:18:48.9] BS: Short-term overhyped, long-term underhypeed. Yeah, when we have conventions for things like I see owing apartment buildings and stuff like that, I think these tokens are going to be – they're going to be valuable. Yeah, if you can put your condo in a token, I will say that long-term the productization of the maintenance of your mnemonic will be pretty important. You don't want you don't want to lose the key for your condo and then you just don't own your condo anymore.

Making this super easy, using partial mutability and proxy conventions to have super backups of less common important stuff, but doing simple things with your address, making that easier, I think that will come along. I think there's another thing – there's other component which is how do you maintain off-chain ownership of on-chain assets? If you've got a condo or something like that and you manage ownership online, how do you connect to that with the legal system of managing property rights in a real world thing, like in a conventional legal system?
Some people are trying that with real estate. Similar to how protocols and TCP/IP come along to be these big conventions. We have the same thing in legal documents, so the safe is a big legal document for fundraising. I think there will be conventions for allocating ownership of off-chain things to on-chain contracts. Maybe there will be some document that says the owners of this token own this apartment building. If you own 20% of the tokens, then you get 20% of the rent

every month. Anyone from all over the world can buy a piece of this apartment building and potentially get rent from it.

You don't have to be an American citizen, or know the legal system to have a claim in a low-risk asset like an apartment building and generate returns for that. That's another piece of creative chaos. How does the fractal of crypto spread to everything, like the legal system is now? When you have programmable money, what are the consequences of that?

[1:21:07.6] JM: Okay, last question. I remember when we were working together at Soule Mobile, there was a time where we walked to coffee and this was the first time I remember – I can remember talking to anybody about crypto. We were having some conversation, you were telling me about the Bitcoin white paper and I was like, “Yeah, I don't care about that. I'm studying for computer science classes right now and we don't have anything like that in our computer science classes. I don't care about this.”

It took you a while to get into crypto also. You read about it, but you didn't really take it seriously, or just didn't get deeply involved with it. Are there any technologies today that outside of crypto, newer technologies where you're looking at them and maybe you're getting the same sensation that you got with crypto in the earlier days, where you're saying that thing out of the corner of my eye, it's weird but it seems it's going to be really impactful?

[1:22:03.7] BS: That's a germ of a thought. I mean, this might be a little cliché, but I think automated vehicles I think somehow have some extrapolating consequences. What is a world like where all cars can drive themselves? How does that affect real estate values? How does that affect where people live? If you can just pop in a car, you don't have to drive it, can you potentially live further out if there are tunnels underground that will let you go 150 miles an hour? Could I live in San Francisco and live halfway between here and Sacramento and have a giant 100-acre ranch or something? Okay, 15-acre ranch.

When cars drive themselves, could you potentially have autonomous pizza-size vehicles that are just driving around in a special-purpose way delivering you pizzas? I think the consequences of that are really interesting and probably the real estate people that stand and make money are thinking about this critically right now. What is a region of your city that's just

slightly too inconvenient to commute to, with low property values? Is that a potentially attractive place to begin development at this present moment? I don't know. Maybe I'm convincing myself to buy a rental home outside of town and rent it out for 10 years or something like that.

[1:23:22.8] JM: Or a house in Round Rock. North Austin underpriced, I think. Undervalued. I don't know anything about real estate prices. Thanks for coming on the show. This has been really fun. We should definitely do it again sometime. We have a lot of crypto stuff that we didn't discuss.

[1:23:37.1] BS: Yeah. I hope I can have bridged the gap between application development, regular engineering that we've all been doing for the last number of years and crypto and how these can be connected. I think, there really is a space for Web 2 folks that are really good at product, that make amazing user interfaces, amazing UX. There's a place in crypto for us now to make really cool products in at trustless way. I think, potentially it could be fun to dabble around and play with this. You could potentially make some cool stuff in a short period of time, so I encourage you to do that.

Keep your eye on the space when scalability picks up potentially. Yeah, you can be doing some really cool stuff with payments baked in directly. When has there been a time when you could just – every user had money available to query instantly? Typically in products, we're like, "Okay, it's time for the credit card conversion hurdle." You got to get them to enter their password in the stripe box, their credit card digits and their address. It's like, you will get 10% conversion or something like that.

We're actually in a world where you can do micro transactions where it doesn't cost 30 cents for a credit card fee. I don't know. I'm thinking about making just some fun pages where it's like, see what's in the mystery box and they're just – you query it for a cent and see what pops up.

[1:25:07.7] JM: Even while, you could just put it outside your house and just have people have to scan a QR code when they walk past. It opens and just whatever they're –
[1:25:17.0] BS: Like what is this important person doing right now? You can query it for 10 cents. The camera will turn on and they wave or something.

[1:25:25.9] JM: I think 21.co tried that, right? Isn't that what 21, or earned.com, but they were just early.

[1:25:31.5] BS: Oh, yeah. Wasn't it like you can ask them a question for money?

[1:25:34.5] JM: Yeah. The idea was microtasks, but they were starting with the question.

[1:25:38.9] BS: Before that, it was a Raspberry Pi full node, or something like that.

[1:25:42.7] JM: Yeah. Yeah. Well, I mean they had a really interesting vision around you have a crypto chip in your toaster and it reduces the cost of your toaster. A lot of stuff like that. That was very early. Really big vision, very early. Stuff will probably all happen and whatever. There's a great outcome for the guys there, since they're just guys and girls, since they just went to Coinbase.

[1:26:03.9] BS: Yeah. I mean, if you've got an existing gap, maybe you could throw in a crypto conversion or donation thing and you might get some interesting customers out of it.

[1:26:14.3] JM: I mean, that's another thing we didn't even talk about, is how the companies with existing moats are just going to be able to adopt this new technology and use it in interesting ways, like how is Apple going to use crypto? How is Facebook going to use crypto? We don't need to go down that rabbit hole. How is Fivver going to use crypto, I think that's another interesting one. Brian, this has been awesome. Let's do it again sometime.

[END OF INTERVIEW]

[1:26:40.8] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source, it's free to use and GoCD recently launched a test drive service that makes it easier than ever to try out GoCD.

You can go to gocd.org/test-drive-gocd. If you've been wondering about what continuous delivery tool you should use for your cloud native software, GoCD is worth checking out. Now

it's easier than ever to just try it out and see if this looks like something that you would want. Just go to gocd.org/test-drive-gocd and find out how GoCD fits your workflow.

GoCD has support for Kubernetes and it was built with the learnings of the ThoughtWorks engineering team. If you want to try it out, go to gocd.org/test-drive-gocd.

[END]