EPISODE 1468

[INTRODUCTION]

[00:00:01] JM: Blockchains were the first systems to allow guaranteed permanent storage of public data. As cryptocurrency technology has advanced, a rich ecosystem of permanent storage and compute has developed as well. One of these is Arweave, a system for permanent information storage.

Sam Williams is a founder of Arweave and joins the show to talk through how Arweave works and his motivation for starting it.

[INTERVIEW]

[00:00:22] JM: Sam, welcome to the show.

[00:00:24] SW: Thanks for having me on.

[00:00:25] JM: We've done many shows about various cryptocurrency-related technologies. And one thing that is obvious about cryptocurrency is the permanence of data. Typically, it's financial data. But there's a more general use case for compute related data, storage data.

And the coverage of that that we've done has typically been of IPFS, which is a object storage system. And you run Arweave, which is a newer type of storage that has permanence associated with it. Can you explain what the data permanence problem that you were trying to solve with Arweave is?

[00:01:14] SW: Yeah. The thing about IPFS is it's in the name. It's a file system, right? In the same way that NTFS, or Ext3, or Ext4 on your machine is an addressing system for finding a piece of data. It doesn't store the piece of data for you. This is a bit of a quite common miscomprehension with IPFS.

And so, Arweave is taking this different approach. Arweave wants to be the physical hard drive, if you will, for permanent data storage. It essentially works by solving two problems. First, we scale blockchains so that you can fit arbitrary amounts of data inside them directly. And the second, we've created an endowment structure, which we can get into later I'm sure, which essentially pays for the storage of that data perpetually in a sustainable fashion. That's really what Arweave is focused on. It's focused on true data permanence.

[00:02:07] JM: The lowest level of storage, is it just laid out in hard drives? Or is it on replicated object storage? Can you give me – Let's work up from the lowest level.

[00:02:24] SW: Yeah. It's on hard drives physically distributed around the world. And you can essentially imagine it like Bitcoin. But you replace the proof of work mechanism with useful proof of storage. And so, essentially, what we're doing is instead of guessing random numbers and checking whether hashed – They have this mathematical property. What we do instead is ping the data. Pull the data set.

And so the more of the data set that you store, the more likely you are to have this random challenge byte, which is requested when you take part in the mining process. And so, we're constantly stochastically checking this data set and incentivizing miners to host as much of the data set as they can, in the same way the Bitcoin works by decentralizing this network around the world. And the upshot of what you get is a decentralized storage network with thousands of replicas of the data set, yeah, physically distributed across the world with no single centralized point of failure.

[00:03:20] JM: Is the access pattern similar to any other storage system you can describe? Or is the – Maybe you could talk about the API for reading and writing to Arweave and describe the path to a read and write.

[00:03:40] SW: Yeah. There's two approaches that people typically take here. The first is that they use the network sort of first-class. They access it directly. And that feels a little bit like you're using BitTorrent. You become part of the swarm, and you swap data with other participants in the network.

But the typical way that people access data in Arweave is via a gateway. And that makes the information available via HTTP, or HTTPS obviously. And that's really interesting because it enables what we call the permaweb, which is all of this data, like 150 million documents, stored inside the system, available to anyone's web browser all across the world indefinitely. That's not just like documents of the **[inaudible 00:04:25]** text, or videos, or images. But also, full web applications themselves. And then some of these gateways index the data set, creating a sort of queryable database that you can access by GraphQL.

And when you add all of this stuff together, what you end up with is a total stack for permanent decentralized web applications to be built, which is really, really powerful, because it allows you to create applications that essentially have no controller. And they're sort of like Ethereum's code is law for smart contracts. But we just apply that whole mechanism to web applications, which enables many, many different really powerful applications to be built.

[00:05:07] JM: If I want to deploy a database that is backed by Arweave, what do I have to do?

[00:05:14] SW: Something that people do is they just run a Postgres database and then they'll make a dump to the network. And then they'll have other nodes that can read out those dumps over time and sort of use that as essentially a synchronization layer, decentralized synchronization layer, across nodes operating your database. That's one approach.

Another approach is to use Arweave's native, or at least the gateway's native GraphQL interface, where you write data to the network. It's like you're creating a transaction. The user just signs a piece of data. It has certain tags associated with it. And those get inserted into this massive database the gateways are maintaining. You can just query those back and say, for example, 'Hey, give me every tweet from – Or post on a Twitter-like system which has an app named this between this and this time from this user," or something like that. And so, that way you build up typical-looking web applications. But they're completely decentralized in the backend.

[00:06:11] JM: And would I need to stand up some kind of caching system for it to have fast access if I have a database that's backed by Arweave?

[00:06:22] SW: It depends. Right now, the simple answer is no. You could use a gateway that is open like arweave.net. And the idea is that these gateways, over-time, create monetary relationships with their users. Such that, essentially, the cost is accruing there. That can be offset by basic things, like showing an ad one in every 50 times you access a web page or something like that. Or a basic subscription model, or a pay-as-you-go model. Or the developer could make a relationship with the gateway where they pay for their users' usage on behalf of those users. There's really like a ton of different ways that you can monetize that relationship in order to deal with the cost of compute, essentially.

[00:07:02] JM: Tell me more about the consensus mechanism for Arweave.

[00:07:05] SW: Yeah. The consensus mechanism is something we succinct proofs of random access, which is a bit of a mouthful. But what it basically means is a small way of proving access to a very large data set polled at random. And what happens there is a challenge is generated for a block. You try and make a candidate blocking the system. You're a miner. You're storing all of this data. You take all of that, or the hash of the last block, and you generate a new sort of, say, derivative hash from the current block that you're trying to mine. You mix that with a random number and nodes. And that will give you together a randomly-generated – A pseudo-random, of course, generated place in the data set. A single byte.

And the game is just go fetch that byte for me please. Then you bring it back. And then you get all of that. You hash that together. And the output of that has to satisfy essentially a proof-of-work puzzle in the same way that Bitcoin does. You're looking for something where the output is over or under a certain value. That would be the difficulty.

But of course, what's happening here is that the more of the data you have, the more likely that randomly chosen byte is within the data set you hold. That's one component. And the second component is the faster you can get access to that data, the faster you can read it from the disk, the higher your – We call it a SPoRA rate. Not really a hash rate. But if you're comparing it to Bitcoin, it'll be a hash rate.

[00:08:36] JM: And how does the mining, the proof system, compare to proof of work or proof of stake?

[00:08:46] SW: You can think of it like proof of useful work, essentially, where that useful work is maintaining the data set. Yeah, I think that's the easiest way of thinking about it. It has the same useful properties of proof of work that you can validate that time or at least effort has elapsed when you look back at the chain, which is the big complaint people have about proof of stake, of course. Which is that if you come to the chain after 10 years and you're presented with it, well, you don't really know that it wasn't forged in the past. There's a sort of permanent honest majority requirement for proof of stake. And this isn't to slander proof of stake or to promote proof of work. I am sort of agnostic on the question.

But for our case, it was just useful to have a proof-of-work system that did something helpful for the network, which the network only really has one job, which is maintaining the data set. You're able to use that work usefully to achieve that end.

[00:09:41] JM: Why did you need a new proof system? Why not just reuse one of the proof systems that has already been used for, I don't know, IPFS or Bitcoin?

[00:09:54] SW: Well, IPFS doesn't have a proof system, nor does it have blocks. We call it distributed hash table. It's a way of rooting. If you go to a single IPFS node and say, "Hey, I'm looking for this hash." It will sort of direct you to the next node that will be likely to tell you where you can get it until you get to the data. It's based on something called Kademlia from the 1990s, more or less.

IPFS doesn't have a proof system at all. Bitcoin – Well, it's really just wasting energy. Fundamentally, it's a game of who can waste the most energy for the amount of Bitcoin that's going to be produced? Well, I suppose the answer is twofold. The first is we didn't want to do that because that's just fundamentally – I used to be friends with Joe Armstrong, the creator of Erlang. And he told me once, "That's just not what we should be doing in the world." And he's right. We shouldn't just be wasting energy like that. There's got to be better things we can do with it. And it turns out, one of those better things is storing data. That's part of it.

And the other part is, if we used the basic proof of work or even proof of stake, we would still be left with this question of, "Well, how do we prove that there are replications of the data set out there?" And so we kind of tie those two, if you will, incentives, imperatives really, for us together and come up with this one solution, which is a useful proof of work.

[00:11:13] JM: And so with the succinct proof of random access, can you describe how the proof is distributed through the network of miners? If I have one random access that needs to get proven and distributed to all the other miners, what's the like time to finality? And just how does that distribution work?

[00:11:37] SW: Yeah. Once you've found a candidate block, it works very much like Bitcoin. You get that proof. And they're succinct in the sense that, no matter how large the date you're proving access to, the chunk that you have to move around is at maximum 256 kilobytes. That gets packaged up with the count of the block that you created and then gets gossiped around the network, just like in Bitcoin, to other miners that can take it. And they say, "Okay. Does this validate?" And if it does, everyone accepts it. They move to the next block. And the mining game continues.

[00:12:07] JM: If I'm a miner, how does my incentive system – Like, compared to something like Bitcoin, in terms of earnings, how does it compare?

[00:12:18] SW: Well, that's the interesting thing about cryptocurrency mining, is that the answer simply is it fluctuates, right? What crypto mining does at a basic level is say, "Okay, my budget for –" in the case of Bitcoin, simply security of, say, finality of transactions. But in the case of our Arweave, finality of transactions, and also the security of the data set is determined by this number of tokens that I'm willing to emit during this time period.

And what will happen is more miners will join the network until that is essentially saturated such that it's down to your opportunity cost. And after that, no more miners will join the network. And perhaps the cost fluctuates. And after time, some miners will leave, then other miners will join, and so on.

We're just sort of doing there – And we actually use this in the system, it's quite interesting, as an approximation of the price of Arweave tokens. There's this big question, right? Imagine you've got this storage system. Okay, it looks like it makes sense. But you've got this problem that, well, if the token price increases – And of course, it's completely variable. So it will increase or decrease. How do you keep the price of storage stable, right? That's a fundamental issue. And the answer cannot be, "Well, we have a –" I'm not sure if you've spoken about this much on the podcast before. But the answer can't be that we rely on some centralized Oracle. So we don't get the data from somewhere in the external world, if you will, outside the network and just kind of trust it. Because as we saw during the 2018-2019 rise of DeFi, it was Oracle-based attacks. Attacks on those estimations of what the prices of tokens were that led to most of the hacks in DeFi. We didn't want to do that.

What we figured we could do instead is say, "Okay, well, at time step zero, we can benchmark and say we've got the present price of the token. The amount of work that is going into producing a block. And the number of tokens that are being emitted from that block." We can take a few of these timestamps if you want to set the benchmark rate.

And then, at times of N, we can say, "Okay. The network is out in the wild now. We don't know the present price of Arweave tokens. And we don't want to trust the outside world to tell us. What we can do instead is say, "Okay. Well, we know the amount of work going into producing blocks. And we know the amount of tokens that have been emitted from blocks. And from these two factors, we can infer essentially the third factor, or at least a heuristic for the third factor, which is how much of the token is costing?" And in fiat money, or at least in, you could say, a value-stabilized version relative to the value of fiat at the time that the benchmarks were taken.

And then once you've got this, you can use it to normalize the price of storage. And so we do this quite effectively. And has been in production for three years now. So we don't have to have this external oracle, which can be hacked. The network can just guess its own token price, essentially.

[00:15:17] JM: If I'm a miner, is there any particular constraint on what I need in terms of hardware? Like, how can I contribute productively to the network? How does that compare to – Is it the same kind of like typically like GPU-related hardware that miners are using?

[00:15:39] SW: Yeah, not so much. The interesting thing is that it's – First off, we worked with the Monero team a couple of years ago to build this new hashing algorithm. It's actually pretty interesting. It's called random eggs. And, basically, the question was, "How can we make it so that the hashing algorithm is CPU-bound?" Because, of course, the problem is CPUs are general purpose, right? If you have a specific type of compute, like a SHA256 hash, or an Eth hash, then presumably you can create a CPU which is a more effective physical device for computing just that particular workload than a general-purpose device.

And so this led to the creation of these ASICs, specialized mining hardwares, essentially, in the Bitcoin mining community way back in 2013. We were trying to make it so that mining in Arweave – And, sure enough, the Monero team were thinking the same way. Mining the Monero was, how do you say, as accessible as possible. People with normal hardware should be able to commit that hardware and be rewarded somehow for that commitment.

And so, we realized that the – Or actually the Monero team realized, that the way to do this is to make the compute involved with the hashing algorithm itself general purpose computation. They built this super cool system, which is kind of like a small virtual machine layer on top of x86 hardware where they generate these random programs for it. And it just stretches all of the parts of the CPU, such that if you were to try and build a specialized piece of hardware that is faster at computing that component, you would actually just end up building a faster general-purpose CPU. So that was a really, really neat innovation they made. And we worked with them on that. And we deployed it for the hashing part of our system.

Other than that, what you need is just a hard drive or some kind. And the faster, and the bigger, the better. It's kind of trade-off between how much you're willing to pay for speed versus how much you get capacity-wise. But that essentially means that anyone with good SSDs and a good CPU can contribute to the system.

[00:17:44] JM: That general purpose algorithm, I guess, what you're saying is it just levels the playing field for all kinds of CPUs to participate.

[00:17:55] SW: Yeah, yeah, yeah. And geographically levels the playing field, too, because we saw this mass migration of Bitcoin miners middle of last year, right? But before that, 87% of the hardware that was capable of doing SHA256 hashes at speed was located physically in China. Which for a network like ours, which is trying to be a neutral permanent ledger of history, that is obviously a problem on the geographic distribution. Of course, general purpose x86 CPUs are fairly well geographically distributed. Not so much in Africa, fortunately. But at least in most of the rest of the world, there is strong spread of this technology.

When you talk about the application layer on top of Arweave, you call it the permaweb. And that's documents and applications that are permanent. I guess I'd like to get a better understanding of of how that permanence works. If you think about S3 – You have a document on S3, it's replicated to at least three places. And if one of the servers gets knocked out, the document gets replicated to another server.

Maybe you can talk about the resilience of the underlying Arweave storage layer. And what happens if – Or I guess just outline a little bit more about how that storage system works. Is it sharded? Or just talk about the distribution. And what happens if nodes fail?

[00:19:31] SW: Yeah. With Arweave, I mean, you don't replicate the data to three places. You're replicated right now somewhere in the high hundreds to low thousands of places. It's very, very, very well-distributed. Probably, I don't know if it's the most well-distributed data set in the world. There's an argument that's like constitution or the **[inaudible 00:19:49]** is much more well-distributed. But certainly, for a data storage system that the end user can access today, it is the most well-replicated data set that you can add your data freely. That's part of the power of the system.

I would say that relative to amazon S3 – And this is why we started it as a network. Not as a – Another way to do this might have been as a foundation, right? A non-profit foundation or a company even. The reason that we did it as a network is because you can trust the math isn't going to change its business model. Math is immutable. That's really powerful. Because you can store your data with S3 today. But the problem is you have no idea what S3's business model is going to be five years from now. Let alone like 20 years.

And what Arweave is really focused on is building a record of history. For example, there's people in our community that have contributed 13, maybe 14 million documents from the Ukraine crisis and the war, unfortunately, now, to the network, those documents is really powerful if they're around for hundreds of years. Not just tens of years. And there's absolutely no way, if you put those on Amazon s3, that you can be confident that Amazon isn't going to change its business model. It almost certainly will. In the same way that you can be confident that Bitcoin won't change its business model. It's always going to have 21 million tokens. And that's the end of the game, right?

That you can say the same with Arweave. Arweave has this storage layer, which is going to pay miners in order to store the data and no one can ever change that. You don't have to trust a company. You don't have to trust a board of trustees in a foundation. You don't have to trust anyone. You just trust math.

And to speak to the math, it's like, "Okay. Well, this is all pretty interesting." But how much do you pay for like permanent data storage, right? As an ever continual cost for a single fee, how can that possibly work? And the answer is essentially to create an endowment structure. In the same way that like a university might put aside a very large amount of money and then gain interest on that money and use that interest to pay for university tuition for students, we essentially do the same thing, but with data storage.

When you put a piece of data into Arweave, you pay for 200 years' worth of storage of that data at present prices upfront. Sounds like a lot. Actually, storage is so cheap nowadays that that's like a cent per megabyte, something like that. Maybe it's expensive if you're storing 8K video or something like that. But for almost all kinds of documents that people really care about, it's actually fairly cheap. It's affordable. That's the base.

And then over-time, as the cost of storage declines, that storage purchasing power you've got, let's say, 200 years, at the end of year one, you've used one year. You think it's 199 years. But actually the cost of storage has declined. And it typically declines at a rate of about 30%, or at least has done for the last 50 years. But we expect a rate of 0.5%. If it stays above 0.5%, at the end of any given year, you end up with more storage purchasing power than when you started. Essentially, that's your interest in the form of storage purchasing power.

Yes. And so, through this system, the network can pay for storage for hundreds, if not thousands of years from this point far beyond the lifetime of the network where it will eventually be replaced by some other sort of protocol achieving a similar thing would be my expectation. And then the hope is that the cost of storage is so cheap at that point that someone just copies across the Arweave, because it's an interesting data set, into whatever new storage system. And the system goes on. That's a broad, I suppose, outline of how the system works. Why it's a network. And how the endowment works.

[00:23:31] JM: Can you tell me a little bit more about the APIs that sit on top of that high-volume of storage? And if I want to request a document or an image from that storage layer, what happens?

[00:23:48] SW: Yeah. I mean, typically you'd use a gateway for that. You just go to arweave.net, or gateway.rdrive.io, or wherever the gateway is located, stroke, and then your TX ID, and then you just get the data back. It just feels like HTTP. If you want to write data to the network, you probably use the library. I mean, there's a protocol in the same way that HTTP is a protocol. So you don't want to interact directly. Typically, you want to have a library on top.

And those libraries, they say create transaction. And every piece of data now is a transaction. It can have as many tags as you want, every kind of metadata, if you will. And then you give it a body, which is the document. It's kind of like a document store in that sense. And then you just sign it. And then you just press send, basically. Or you'd rather call send. And then it goes to the network. And if you have tokens to pay for it or some other forms of payment, then it just works. And we're currently subsidizing all transmissions to the network under 300 kilobytes. You don't even have to have any tokens in your wallet if you want to store something under that amount, at least for the moment. And there's many ways that you can pay otherwise.

[00:24:58] JM: Right. And I think it's worth talking a little bit at this point about why this is a useful exercise to make a permanent web. It seems like censorship at the storage layer doesn't really happen that often. I mean, I guess people lose things in the web. If I don't pay for my for my storage, then I lose it over time. This, in a sense, amortizes storage across the entire network such that I don't ever lose my photos. You don't have the –

[00:25:34] SW: Yeah. You you paid for the endowment. So you don't need to worry. Like, there's no further payments required from your site. You're just generating interest from your contributions to the endowment that that's paying for storage. But I think that actually stuff does get censored from the web all the time. Might not be so much in the west, or it might not be so noticeable in the west. But certainly, for example, in China, all the time.

And so, a lot of the uses for the network are sort of aimed at places where they have authoritarian regimes as the local power structure. For example, it was used in Hong Kong to back up Apple Daily, one sort of pro-democracy newspaper that was very prominent there. And then the authorities came along and just started stealing the data, fundamentally. But, fortunately, there were a bunch of activists who are able to capture some of that and start uploading it to the network. And now there's this archive of 13,000 or 14,000 pages from Apple Daily inside the permaweb. And of course, that's distributed all around the world. And so, it's very difficult for the Chinese government, if they wanted to, to remove access to that data. They would have to – It's kind of like whack-a-mole. You would have to block every address. And the addresses are ever changing. Pretty annoying. Relative to – Well, just in the case of Apple Daily, it just seized a server, and that was the end of it. That's just one example.

But I do think that censorship does happen reasonably frequently at the level of services around the world. Another example being Weibo. In China, back in the beginning of the coronavirus pandemic, in about February, there was this outcry for freedom of speech on Weibo from users that wanted to – Well, they wanted to express their displeasure with the Chinese communist party, fundamentally. But of course, they were getting censored left and right by powers that be there. And so, they didn't have a platform where they could speak freely, frankly.

And again, some people in our community started archiving data from Weibo inside the Arweave network. And then they built a permaweb app on top. So a full web application. And it ran in the way that I was describing previously where you could ask it, "Hey, can you find me everything that's been censored on Weibo?" which was really fascinating. So they found a way of flagging old data that had been censored. And yeah, they called it WeiBlocked. And it was just an index to the censored of things. And of course, that was available from many different user interfaces, or from many different locations on the web, and many different addresses. And so very, very hard to block access to that. I could go on. But I think these are some good examples.

Well, I mean, to the more fundamental point, why build something like this in the first place? The simple answer is that archives haven't improved technologically for around a thousand years. And technology has improved dramatically in that time. So why haven't they improved? And I think that we can now build a library of Alexandria that is impervious to fire, flood and government censorship in a really pretty profound way that just wasn't possible before.

And for blockchains, blockchains had the seed of that, right? You mentioned how they replicate data all around the world, and the ledger all around the world. And they don't have single centralized points of failure, but they didn't scale. We solved that problem with the mining system we described. And there was no incentive system for replicating all parts of the data. We solved that with the endowment system. That was essentially how the network came together.

[00:29:04] JM: Granted, yeah, there is definitely censorship. We shouldn't have given such a western-centric point there. Aren't they still going to censor at the application level? If they want to censor something, they still can censor at the application level. I guess the point is just that you're building unsensible storage and other people will build unsensible compute layers.

[00:29:28] SW: Well, I'm not quite sure how you're separating up the stack. Because we see that almost all compute related to like basic web applications is more or less database queries, right? It's like create, read, update, delete. Runs web 2. Some non-trivial proportion. And that is sort of inside the system. But compute, in that sense, runs inside the gateways as described. And there are many of those across the world.

You certainly can – As a developer, if you want to add a sort of censorship layer on top, that's totally fine. The point with the permaweb app is that after something is launched, the relationship between developer and user cannot change. Or rather, the developer might release a new version. But users aren't forced to upgrade to that new version. And so this sort of what we call platform integrity guarantees built-in. Once you have access to a version of an application, it's what you see is what you get forever. They can issue a new version. And you can upgrade if you want to. But no one can force you to. And that new version might have some form of censorship on top. And that might be appropriate for the application you're building. But everybody knows what the deal is that they're getting into when they start using an application.

Putting censorship aside for a moment, you can see why this is important if you think about something like email. In the case of Gmail, when I signed up for my Gmail account, like 2007, whenever it was, I had no idea what Google's privacy policy would be in 2022. Or whether they would be showing me adverts or selling my data to people. Who knows? It's totally up to them. But my identity got locked into that service. And it was forcibly upgraded without my request.

Whereas, if you use permamail on top of Arweave, it's got a user interface today. You can see it and say, "Okay, it's fine enough. I guess I could use this for a decade." And it's just never going to change. If it doesn't sell your data today, it's not going to sell your data tomorrow.

[inaudible 00:31:30], the guy from our community that built it, you could come along and say, "Okay, here's a new version with ads. Go use it, guys." But you don't have to. And so the deal that you're making between user and developer is set in stone at the time the thing is launched. And it can only be modified essentially by consent of the users if they decide to upgrade. Does that make sense?

[00:31:52] JM: Totally. I guess, when I was commenting on the censorship, I was just imagining, if I'm Apple Daily, or whatever the sensor-worthy site, the user makes a request to Apple Daily's servers. And if I'm the Chinese government, I can censor at Apple Daily's servers, right? I always have access to that centralized point. My point is like, unless you have the users directly routing to decentralized infrastructure, there's going to be a point where you can censor, right?

[00:32:33] SW: Well, appledaily.com is down. The archive is put the data on the network. You don't go to Apple Daily anymore. It doesn't go anywhere. It's useless, Instead, you would go to a permaweb address for either an item from the network or a user interface built on top that sort of aggregates the items, the distorted network.

[00:32:52] JM: Got it. Okay. Does the user need a new kind of browser to do that?

[00:32:58] SW: No, no, no. They would just go to an Arweave gateway, like arweave.net, or gateway.ardrive.io, or whichever one you want, and then the address that they want to get to. The TX ID, we call it. Transaction ID, essentially.

[00:33:11] JM: Got it. China would have to censor the entire Arweave network in order to censor anything on it.

[00:33:19] SW: Essentially. And that would mean censoring an ever-morphing network, if you will. Like, new servers come online. All servers go offline. All the time. It's part of the the natural, which is the life cycle of the network. It's an ever-changing beast, which is much, much harder to hit than of course something like appledaily.com where you go to where that server points to and you go seize the hard drives, and there you go.

[00:33:42] JW: Okay. And there's no like universal TLD where they could just cut it off at that point.

[00:33:50] SW: Yeah, the whole premise of the system is not to make a single centralized point of failure of any kind.

[00:33:56] JM: Very cool. So since we're nearing the end of our time, let's zoom out. When you look at the Web3 stack as a whole, it kind of seems like all the lower-level building blocks are there at this point. We've got compute. We've got storage. We've got decent user interfaces. I guess, maybe Metamask leaves something to be desired. What are the weak points in Web3 infrastructure? Like, where is the tipping point where Web3 maybe becomes the default for deployment of applications?

[00:34:32] SW: Those are two separate, but very interesting questions. If you don't mind, I'll address them individually. What are the weak points? I think you're right in assessing that the base infrastructure is more or less here now. If you want to build a decentralized – I've got into the Ethereum ICO because I wanted to use Ethereum as it was stated on the website, as the world computer. I wanted to host my website on this thing. I thought that would be really cool. It didn't exist at the time. Ethereum never became that.

But now, like eight years later, that infrastructure does exist. If you would like to build a decentralized web application, it's like top to bottom decentralized, not just a smart contract, then you absolutely can. I would say the weak point is, to some extent, we're really starting to very aggressively address this now in the Arweave ecosystem. It's reaching Web2 developers with this message. And yeah, I’m bringing them into the fold. It's only in the last six months or so that the infrastructure has really matured to the point that people can just build on this easily. And there's simple ways to do all of the important things you would be doing when you're building a web application.

Now, we're kind of pivoting into that like hyper-focused and aggressive adoption strategy phase. And so, what we're doing there is we're running this program that basically says, "Okay, well, we want to fund the next thousand founders building on top of the Arweave network. And we want to take them through this simple funnel where they start off as like someone working at a job in a Web2 company and they end up running their own project in building on the permaweb, which is funded, and they have a team and so on.

And we start by like introducing people to how you build a Hello World. Take them through a tutorial for how they build these applications. Meet the rest of the community. And we progressively invest like a hundred thousand dollars in each person, in each team building, as the program goes on until they get out the other end where we help them get their first funding from the external market.

And we're really trying to do this in like – Which we say, we're not interested in reaching crypto-aware developers. They're already here. We're really interested in reaching Web2 developers. If you just go to like arweave.build, you can get on board with that. And it's made for people that are not crypto-aware. That's what I would say on the weaknesses front.

I mean, for sure, to be clear, there are like rough edges. It's early stage technology nonetheless. But I think you accurately summarized it. But essentially, the building blocks of the stack are now there. To your second question, I’m afraid I’ve forgotten. Sorry. Can you restate it please?

[00:37:01] JM: When does Web3 become like the default –

[00:37:04] SW: The default. That's a fascinating question. I would say the first thing that comes to mind is, well, will it ever be the default, fundamentally? Because Web3 is a way of building a certain type of thing. It's not clear that every application should be a firmware app. I think, like, many, many, many of them should. But possibly, you can see that – I don't know. If you have a like cupcake factory or something like this, and it's got a website, I don't know that that has to be a firmware web app necessarily.

I think there will still be stuff that people want to use the centralized web for like that. And I think there's this question around monetization strategy. Should every company be a DAO? Well, it doesn't seem obvious to me that there are advantages for absolutely every company to be a DAO. Well, I mean, there's a fundamental advantage that it's just an electronic company, if you will. Everything is just kind of bureaucratically faster and it's adjudicated by code, which is helpful. But it's not. Like, which to say, that in itself is not world-changing. That's an evolution of a revolution.

Yeah, I think that's a very interesting question. But if you want to build a censorship resistant web application, there is nowhere to go on the Web2 space that will actually do that. You see people trying to, and they go to like alternative web hosts and all this type of stuff. That's not going to work. You cannot build an application where you don't have to trust the person running that application in Web2, fundamentally. Because the protocols are set up to connect you to a place. And that place has to be run by a person.

Well, on permaweb, it doesn't work like that. The protocol says, "Okay, there is an identifier out here. And that identifier is many, many different places. There are many people that are incentivized to store it and replicate it for you. We can go find it for you.

It's a really different model. There's sort of revolutionary stuff, I would say, which I'm not sure absolutely everything will ever fall in that category. But the evolutionary stuff, like, wouldn't it just be better if companies were DAOs? I guess that's probably true. I mean, this conversation frankly could go on a long time. There's just a lot of intricacy and detail.

[00:39:07] JM: Gotcha. Well, thank you so much for coming on the show. And congratulations on a really successful Web3 project.

[00:39:13] SW: Thank you. I mean, I see it as just about getting started. But, yeah, certainly the engine is going now. It's exciting. Thanks for having me on.

[END]