Ethereum in 30 minutes

(0:16 - 0:51)

Great, so Ethereum in 30 minutes. So this is a presentation that I've given at I think basically every single DEF CON since launch, though I think what's interesting is how as Ethereum the ecosystem changes and as Ethereum the protocol changes, as the times change, the contents also end up changing quite a bit, right? So if you go back to the equivalent of this from 2015, you'll hear a lot about uncleblock. Of course, uncleblock are a feature of Proof of Work that we have Proof of Stake, and so we don't have uncleblock anymore.

(0:52 - 1:50)

Now Ethereum also, of course, now has Layer 2s, and back in 2016 this would not contain Layer 2s at all, and now, of course, as we know, Layer 2s are half the story, right? So Ethereum is above all an evolving ecosystem, and the contents of this are going to keep changing as the technology changes and also as the emphasis of the ecosystem changes, from building basic infrastructure to now also really putting a huge amount of focus on the applications. So what is Ethereum? So first, Ethereum is the world computer. So who here remembers Ethereum being the world computer? This is the place where I'm supposed to put that meme where there's the guy with the gun in space saying, always has been, but I don't know.

(1:50 - 2:29)

I never understood why giving someone such an amazing and beautiful revelation is supposed to come at the same time as shooting them in space 20,000 kilometers away from home, and so I'm not going to do that today. So Ethereum is the world computer, it always has been, always will be. So in addition to this, Ethereum is an incredibly large and diverse on-chain economy, and fortunately Josh, right before, one skipped before me, made an incredibly good presentation talking about all of the different aspects of the economy.

(2:29 - 3:03)

And also, Ethereum is an incredibly large and diverse global community, and Aya and Josh and the automated intro thing before that, I think, did a really good job of showing that too. So Ethereum is many things, right? I think they're also probably supposed to put up a flashing sign saying, ETH is money as well. But Ethereum is a lot of things, right? And Ethereum is all of these different things at the same time.

(3:05 - 3:56)

So how does the world computer work, right? So the way that I think about this is, layer one is the trust machine, and layer two is the GPU, right? So the layer one of Ethereum, that is the core Ethereum blockchain, this is the anchor that keeps the rest of the ecosystem safe and holds the rest of the ecosystem together. Layer one is the reason why layer twos are able to trustlessly talk to each other, and at least once everything upgrades to stage two, you will be able to take an asset, issue it on optimism, and then move it and hold it inside of a smart contract on Arbitrum and do so with zero counterparty risk. Layer one is not there to be ultra-fast.

(3:56 - 4:10)

It is not there to do a million transactions per second. Layer one is first and foremost there to be decentralized, to be robust, and to be something that is dependable. Layer two is the GPU.

(4:11 - 4:31)

So layer two is all of you in the room who are part of any single layer two. Who here is part of a layer two? Yay! Who here has used a layer two? Who here has used layer one? Good. We're all users here.

(4:31 - 5:24)

So actually, there's lots of these different GPUs, right? And Ethereum is a world computer that's able to hold together because each and every one of these GPUs is connected to the trust machine through optimistic proof systems, zero-knowledge proofs, Snarks, Starks, Binias, Jolts, Plunks, Plunkamadunk, you know, whatever the buzzwords are. And all of these systems actually ensure that layer one is able to trustlessly verify what is going on inside of the layer twos, and layer twos also are able to trustlessly read what is happening on layer one. So there's an important interplay between these two components, and together they make up the Ethereum that is here today.

(5:25 - 5:30)

So what is the L1? It's a chain. It runs proof of stake. It doesn't afraid of anything.

(5:33 - 6:17)

So what does this mean, right? So Ethereum layer one is a chain that has existed altogether for more than nine years now, and it's existed in its current proof of stake form for a little bit over two years. And I think one of the really important things for a base layer to have is you want to have clear evidence that you're building on a base layer that is decentralized, that is open, that is robust, and that is likely to keep those traits going forward into the future. And one part of that, of course, is something that is slow to change.

(6:18 - 6:53)

One part of that is being something that is not suddenly going to wake up on a Tuesday, have new management, and then suddenly decide that it's just totally going to start deleting a whole bunch of different applications, change its entire model, and push the fees up by a factor of 10 and do other things without warning. Another part of decentralization and resilience is actually recovery. So whenever a problem actually starts to arise, actually be able to recover from it and have some way of actually improving the qualities of the layer one over time.

(6:54 - 7:39)

So over here, this is on the left a chart of Bitcoin proof-of-work mining pools, and on the right, Ethereum proof-of-stake mining pools. Actually, I'm deliberately being generous here, putting Bitcoin proof-of-work mining pools, because before the merge, actually, Ethereum proof-of-work was even more concentrated than this. And so what we see here on the right is you see actually a pretty diverse different set of staking pools, and if you zoom into either of the top items on the chart, on the top left, so Lido, and it has somewhere a little bit under 30%, but then actually, Lido is like, it's not one actor, right? It's a DAO.

(7:39 - 8:11)

Technically, the deposits are split between 40 different node operators approximately, and there's some more complicated structure that keeps updating, but so I think it's reasonable to think of it as being like somewhere between one actor and 40 actors. Exactly where this is like a problem for you social science people to figure out, right? How do you actually quantify the decentralization thing? There's even grant opportunities for this stuff. So that's number one, right? And then number two, unidentified.

(8:11 - 8:44)

Now, unidentified is not an actor, just like the any key on your keyboard is not actually a key. So unidentified, like we actually don't know what it is. There's probably lots of solo stakers, lots of small business stakers, lots of various tiny staking pools, also not one actor, right? So Ethereum ML1 today actually has a surprisingly high degree of decentralization in its proof-of-stake design, and this is a property that I think has only gotten better over time.

(8:45 - 9:24)

Here is one property that actually has gotten a lot better. So Ethereum about five years ago was basically just geth, right? And when you have an ecosystem that is dominated by one client, that itself becomes a central point of failure, right? So who here remembers the 2016 DOS attacks? Yay. So remember, wake up 5.17 a.m., get a military-style wake-up call, say, hey, you have to go down to the thing that we suddenly turned into a war room now.

(9:25 - 9:41)

Go down, and then, oh, someone discovered a bug in the clients. Now the entire chain has stopped, and then we fix it. And then two to three days later, some guy somewhere finds another bug, and then three days later, another bug, and then that keeps going on for a whole month until eventually we have to hard fork.

(9:41 - 9:56)

But during that period, actually, the ability for Ethereum to have two different clients at the time they were geth and parity basically saved the chain. There were times when there was a bug that would hit geth but not parity. There were times when there was a bug that would hit parity but not geth.

(9:56 - 10:23)

And so Ethereum actually gained a lot from having multiple software implementations, and this is something that, as of 2024, is probably at the best that it's ever been in Ethereum's history. So on the left, you have consensus clients. Consensus clients handle the proof-of-stake part of Ethereum.

(10:24 - 10:35)

On the right, you have execution clients. Execution clients handle the EVM part of Ethereum. So as we can see, the single client with the highest market share is geth, which is somewhere around 50%.

(10:36 - 11:06)

And actually, this is from a source that's basically assuming that most unidentified clients are geth, and so it could easily be below 50%, right? So let's say it's about 50%. So what happens if there is a bug in any Ethereum client literally today, right now? OK, anyone here has a bug? OK, Josh, can you check? Is Ethereum still running? I'm going to guess it is. Bet like 99.99% plus on Polymarket.

(11:06 - 11:21)

OK, so case one, what happens if the client is geth? That's the worst case. If the client is geth and there is a bug, what realistically happens? Basically, chain splits in half. One half follows geth.

(11:21 - 11:27)

One half follows the other clients. But on both sides, the chain stops finalizing. To finalize, you need two thirds.

(11:27 - 11:44)

If you have less than two thirds, the chain keeps going. Blocks keep getting created, but the chain stops finalizing. And so if you are a user, an application, a business, and you're waiting for confirmation on some transaction, you're probably actually going to be waiting for a finality.

(11:44 - 12:03)

And so if the chain splits in half because of a client bug, well, actually, neither chain is going to finalize. And so you're going to detect that, and you're going to be on standby. What's realistically going to happen? Core devs get on high alert, figure out which client actually has the bug, and the bug gets fixed.

(12:03 - 12:14)

The one time this happened in Ethereum, actually, fun fact, I actually set the transaction and did it myself. This was back in 2016. Basically, everything was fixed within 12 hours.

(12:15 - 12:31)

So that's the worst case. Every other case, basically, if Prism and Whitehouse have a bug, all that happens is that Ethereum stops finalizing for about a day at most, and then it just goes back to normal. Any other client, you're not going to notice anything at all.

(12:32 - 13:12)

So in terms of practical decentralization, having multiple clients is extremely helpful. And also having multiple clients is extremely helpful because it diversifies the power and the control over the Ethereum ecosystem, especially in any kind of contentious situation. If some kind of Dow fork-type incident happened again, and one particular development team made a choice that is unpopular, then even if that one development team just insists on pushing it through, users are going to be very easily able to just switch to the other clients and just completely route around them.

(13:12 - 13:27)

This is not something that is anywhere close to that practical in a single client ecosystem. Having a diverse multi-client ecosystem is actually very difficult to achieve. I think aside from Ethereum, no other chain has really gotten to achieve anything close to this.

(13:28 - 13:50)

And even outside of Ethereum, like even web browsers. Web browsers are supposed to be an open standard. Reality is something like 80% of the plus of the ecosystem, probably even more, runs on some fork of WebKit, and to the rest, it runs on Firefox, which is valiantly trying to hold its own as an alternative and is doing pretty well.

(13:50 - 14:02)

It's one of the browsers that I use. But like you can see, there are natural pressures toward homogenization and toward one implementation winning. And Ethereum actually has managed to buck this trend.

(14:04 - 14:36)

In Ethereum, two years ago, this chart was worse. And so Ethereum's decentralization is not only able to not get worse, but it's also able to actively respond to problems that exist and over time push to make them better. If you're building an application that you needed to exist 5 years from now, 10 years from now, and that you want to be robust and continue working 5 or 10 years from now, what properties do you want the chain you're building on to have? I think properties like these are exactly what you would be looking for.

(14:38 - 14:59)

With 32 ETH, or with less than 32 ETH if you join a pool, there's more and more really interesting pool options that are appearing. There's even things like Obol squad staking. There's options around basically creating smaller pools with your friends.

(14:59 - 15:11)

There's different pools that follow all kinds of different rules. And so there's lots of different ways for you to become a staker. And if you're a staker, then you become part of this network of nodes that is securing the Ethereum blockchain.

(15:13 - 15:20)

Earn rewards while securing Ethereum. So I guess this is how the foundation propaganda describes it. It has a cute elephant.

(15:21 - 15:35)

So good cheers for the elephant. So you too can join the network and you too can help secure it. Also, even if you're not a staker, you can also run an Ethereum node and you can verify the chain.

(15:36 - 15:58)

So this is a computer. I mean, I guess these days nobody even has desktops that look like this, right? It's like you either have laptops or you have some crazy GPU construction that basically looks almost like a server, right? But I don't know. I just shoved a desktop computer into a stable diffusion 3.5 and it put this out.

(15:58 - 16:25)

So that's the computer you get. But you too can run an Ethereum node on your computer and you can verify the Ethereum chain. So this is really important because if you have users that are verifying the chain, then even a majority of the stakers acting together, even two-thirds of the stakers acting together, are not able to change the rules on people without everything breaking.

(16:26 - 16:38)

There's the Ethereum rules. They can only change through a hard fork that is agreed upon through wide community consensus. They cannot be pushed through by a majority or supermajority of the stakers.

(16:39 - 17:11)

And this is something that is really key and really important and makes blockchains very different from almost any other kind of protocol. And I think this is one of those things that's really valuable and really important to preserve. I think even among blockchain ecosystems, it's basically Bitcoin and Ethereum that really have a strong culture, at least among the large ones, of actually trying to make it possible and keep improving people's ability to verify the chain.

(17:12 - 17:34)

And there's a lot of protocol upgrades that are coming that precisely have the goal of making it even easier to verify the chain. So today, you have this. Tomorrow, you will be able to, one, run a node without requiring more than a small amount of storage using stateless clients.

(17:35 - 17:45)

In my big series of six posts, this is going to be the one on the verge. Stateless clients is a very important section. Light clients.

(17:45 - 17:54)

There is a project called Helios, which is doing a form of light verification of the chain. So light verification is not perfect. It still follows the consensus.

(17:55 - 18:22)

And so if 51% of nodes change the rules, you are going to follow them. But it does mean that if you're using a light client, that you do not have to trust an RPC node to tell you any kind of information about the chain. And so Helios is building a light client for Ethereum that does verification of the proof of stake part of Ethereum and also starting to build light clients for L2s as well.

(18:22 - 18:42)

The longer term future, of course, is we want to snark the whole chain. And once we snark the whole chain, you will be able to verify the Ethereum rules on extremely large or extremely tiny hardware. So node running requirements and pushing those down is a key part of the Ethereum roadmap going forward.

(18:43 - 18:54)

Also, staking with less ETH. This is one of those very active research requirements, right? So 32 ETH, it's still high. And it's even higher than it was two weeks ago.

(18:55 - 19:12)

But my take, I would like for people to be stakers with one ETH. And so I think there's different ways to do this. There is Orbit.

(19:12 - 19:31)

There is just making a large number of aggregation improvements. But there's a lot of protocol improvements that are coming specifically in order to make staking easier and more accessible and to make running a node that verifies the chain easier and more accessible. So what runs on the Ethereum L1? So some high-value application use cases.

(19:32 - 19:52)

So a lot of high-value DeFi runs on L1. ENS is currently on L1, though it's increasingly doing more and more with Layer 2s. Then a lot of various applications, just all kinds of things in every category.

(19:53 - 20:08)

People just hold assets on Layer 1. So lots of high-value application use cases on Layer 1. Also, managing block routes and state routes and proof systems for Layer 2s. Layer 1 is the thing that secures Layer 2s. Cross-Layer 2 operations.

(20:09 - 20:50)

And Layer 1 needs to still be powerful enough to handle a lot of things happening, especially in the case where a Layer 2 fails. The difference between a Layer 2 and an independent chain is that even if your Layer 2 gets 51% attacked or 91% attacked or the team shuts down, Layer 1 still stands there to protect the users. Users are able to prove their assets and their ownership and their state inside of the Layer 2 and migrate it back down to Layer 1. In order for this to actually be possible, Layer 1 needs to actually be powerful enough to handle the load if a Layer 2 actually does fail.

(20:51 - 21:04)

Recently, there was a live experiment of this. So DYDXv3, I believe, shut down recently. And the L2B people actually wrote their own implementation of escape hatch software.

(21:04 - 21:39)

And so without any involvement from the DYDX team, users are able to take any assets that they have inside of DYDXv3 and bring it back down to the Ethereum L1. Layer 2s are not just multisigs. The ability to actually move your assets out of the Layer 2 and back to Layer 1 if the Layer 2 fails without the Layer 2 team's involvement is not just theory, it is reality.

(21:45 - 22:04)

L1 runs some applications, and L1 protects the L2s. What do the L2s do? L2s provide speed and scale. So on the left here, you have a chart of basically the same as the chart that Josh showed earlier, basically Layer 2 fees, and this is just what they've done this year.

(22:05 - 22:23)

So this year, Layer 2 fees have gone down from about 50 cents to less than 1 cent. So this is a massive change. This basically means that for an incredibly wide class of applications, Ethereum has overnight gone from being basically unaffordable to being completely affordable.

(22:24 - 23:00)

So that's fees. Now, transaction inclusion times. Who here remembers the experience of sending a transaction and then having to wait some arbitrary number of minutes, could be 10, could be 40, could be 90, for the transaction to get included? Who here remembers having this experience in Ethereum? Who here remembers having this experience in Ethereum in the past six months? So Ethereum has improved massively.

(23:01 - 23:21)

Now, this is actually a pretty linear progression that we've had all the way since the beginning. Right? So Bitcoin, 10-minute block times, and Bitcoin has this thing called a Poisson distribution that, of course, Ethereum also had during the proof-of-work era. But then we decided we don't like fish, and so we switched to proof-of-work.

(23:23 - 23:32)

So in Bitcoin, the average block time is 10 minutes. Often you have to wait up to one hour. Now, in Ethereum, the average block time is 12 seconds.

(23:32 - 24:16)

But then there's still this annoying phenomenon where if you send a transaction and you get unlucky with the gas price that you set, maybe you would have to wait 10 minutes or 20 minutes accidentally. EIP-1559, which came in 2021, basically solved that and created a situation where you're able to reliably send a transaction and expect to see it included usually within one block. And then the merge, because of this interesting quirk of mathematics, even though the average time between blocks reduced only from 13 seconds to 12 seconds, the average time between you sending a transaction and your transaction getting included in a block reduced from a little bit more than 13 seconds to a little bit more than six seconds.

(24:17 - 24:25)

So math problem, figure out why this is true. It's actually a really fun fact. So post-merge basically goes down to about 6 to 30 seconds.

(24:26 - 24:40)

And then finally, with Layer 2s, you have pre-confirmations and you have Layer 2s that are fast enough to just confirm your transaction within a couple hundred milliseconds. So Layer 2s provide speed. Layer 2s provide scale.

(24:40 - 24:59)

And the way that you as a user see scale is you get to be part of an application where lots and lots of things are happening. And at the same time, your transaction fees get to continue to be cheap. So history of Ethereum, right? So November 2013, the white paper.

(25:00 - 25:17)

July 2015, the launch. Around 2018, that was the time when Ethereum settled on its approximate design for proof of stake and data availability sampling. So the original papers and documents for data availability sampling and erasure coding, I wrote those back in 2017.

(25:18 - 25:38)

And then we settled on Casper, the friendly finality gadget, also at the end of 2017. Actually, if you really want and you can dig, I think it's either GitHub Ethereum Research or GitHub Ethereum Simple Casper. You just search for the directory called Simple Casper, and you can find contracts written in Serpent.

(25:38 - 25:51)

Who here remembers Serpent? Okay. Who here uses Serpent? Good. I think Python is really beautiful, but I think if you want that, then you should code in Vypr.

(25:51 - 25:59)

Vypr is actually great. It's actually kept improving quite a bit for the last couple of years. So we had basically... Okay.

(25:59 - 26:02)

Yay. Clap for Vypr. Okay.

(26:05 - 26:31)

So in 2017, actually, we attempted to basically do this full abstraction thing and even write the proof of stake logic directly as a smart contract. And so there was this really fun demo that we launched at, I think it was actually like 2320 Bangkok time on December 31st, 2017. And we wanted to get something out before New Year's, but then the demo actually, yeah, it ended up breaking up pretty quickly.

(26:31 - 26:48)

It was early. But since then, of course, it's not early days anymore. And at the start of 2018, a really massive effort started to actually build out the Ethereum proof of stake system and to build out the Ethereum scaling system, which has since then turned into the blobs that we have today.

(26:49 - 26:56)

2022, the merge. Switching from proof of work to proof of stake. 2024, the surge part one.

(26:57 - 27:12)

Right? So if you look into the roadmap diagram and you look into the surge section, you see that there's two milestones. There's one that I call basic roll-up scaling and there's one that I call full roll-up scaling. And basic roll-up scaling basically says you need major layer twos to hit stage one and you need blobs to exist.

(27:13 - 27:29)

And so 2024, we actually hit that. The next step is having actual fully running data availability sampling and major layer twos hitting stage two. So I think that'll happen, and I think that'll happen sooner than people expect.

(27:29 - 27:38)

And it's up to the layer two teams to actually build this. Now, future of Ethereum. There are still a lot of problems that are left to solve.

(27:39 - 27:53)

So we want upgrades to decentralization. Who here wants Ethereum to be decentralized? Who here wants Ethereum to be centralized? Okay. Okay, so one person wants Ethereum to be centralized.

(27:54 - 28:02)

Okay. Censorship resistance. Who here wants to censor Ethereum? Okay, a couple more hands.

(28:03 - 28:11)

Quantum resistance. Who here wants Ethereum to break the first time a quantum computer comes out? Okay, one hand. That's good.

(28:11 - 28:35)

Sometimes you need collapse in order for renewal to happen, right? So further upgrades to make sure that decentralization, censorship resistance, and quantum resistance continue happening. Progressive upgrades to efficiency and scale. So layer two, I think, is going to scale extremely quickly, and its safety is going to improve quickly over the next few years.

(28:36 - 29:01)

And also, I expect progressive upgrades, cautious upgrades, but still progressive and definitely ongoing upgrades to layer one capacity in different forms as well over the next years. And I think one of the big reasons why we need to do this, one is to support the activity that continues to happen on layer one itself. And two, because ultimately if something breaks on layer two, layer one still needs to be there to be as a backstop.

(29:02 - 29:29)

And so the maximum theoretical safe capacity of L2s is proportional to the capacity of L1. We're going to have upgrades to data availability sampling to increase the number of blobs that Ethereum can support. As of about a week ago, Ethereum is actually hit price discovery mode, meaning in terms of blobs, meaning that the number of blobs that is actually being used on Ethereum exactly equals the protocol set long-term target.

(29:30 - 29:36)

Now, we need to scale this number. Upgrades are happening to actually do this. So upgrades to data availability sampling.

(29:37 - 29:52)

Now, we have also scaled enough that a wide variety of applications are possible. ENS, consumer payments, social. One category that I think is going to be extremely important over the next decade is mixed financial and non-financial applications.

(29:52 - 30:09)

Applications that make use of the power of finance but ultimately to serve ENS that go beyond financial goals. And I think there's lots of very powerful applications here. And I think we've spent a long time making the technology better.

(30:09 - 30:15)

And we will continue doing this. But it is at the level where now is the time to build. Thank you.