



Haym @SalomonCrypto

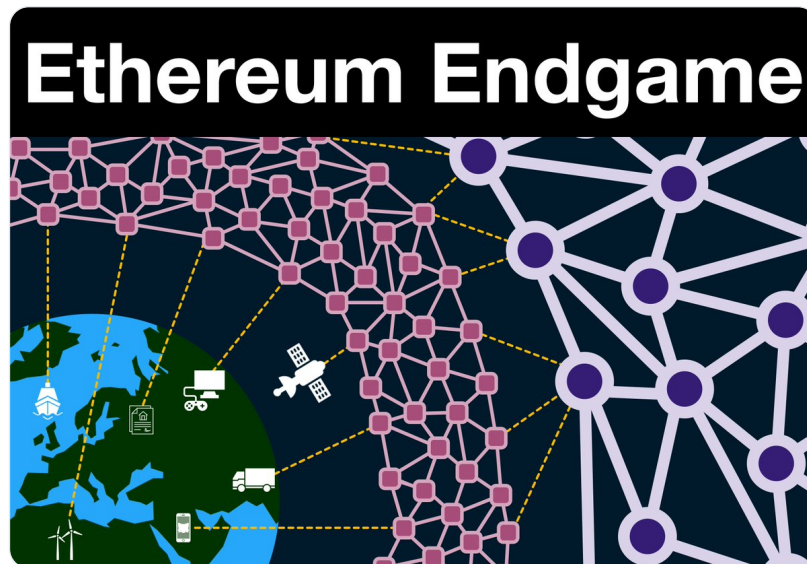
Nov 10 · 30 tweets · [SalomonCrypto/status/1590809909733441536](https://twitter.com/SalomonCrypto/status/1590809909733441536)

(1/29) [@ethereum](https://twitter.com/ethereum) Endgame

The World Computer was born ~7 years ago; like the Internet before it, its development is chaotic and the growing pains are substantial...

...but WOW was the payoff worth it!

The future is closer than you think, you want to see what \$ETH looks like?



(2/29) [@ethereum](#) is a distributed computing platform. A network of 1,000s of computers (nodes) coordinating using Proof of Stake (PoS) to keep the Ethereum Virtual Machine (EVM) in sync.

The EVM is the shared computing platform, the blockchain its history and \$ETH its lifeblood.

**Haym**
@SalomonCrypto · [Follow](#)



(1/21) [@ethereum](#): The Big Picture

From 1492 to 2022, the context, technology and vision of the World Computer. The complete, top-to-bottom case for [\\$ETH](#).

An (unprecedented) mega-thread.



3:00 PM · Sep 3, 2022 

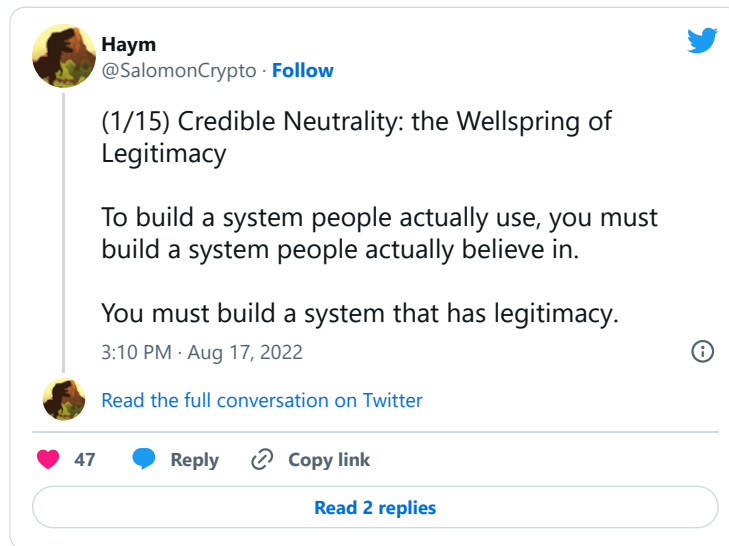
 [Read the full conversation on Twitter](#)

 993  Reply  Copy link

[Read 47 replies](#)

(3/29) This "network of nodes" is the foundation upon which [@ethereum](#) ultimately derives its value. The more decentralized, the more value.

From decentralization comes credible neutrality. Without credible neutrality, we might as well be using FB-dollarroos in Farmville-DeFi.



(4/29) Today, [@ethereum](#) is decently decentralized and super not scalable... by design. Raising the minimum requirements of a node increases scalability at the cost of decentralization.

Put another way, if we want to increase decentralization, we need to sacrifice execution*.

(5/29) Because here's the thing: running a node, the one and only way to access [@ethereum](#), is kind of a big deal. It requires dedicated (high end) hardware, unlimited bandwidth and constant maintenance.

The reality is that it is too big a task for 99% of the population.

**Haym**
@SalomonCrypto · [Follow](#)



(1/28) [@Ethereum](#) Basics: Accessing the World Computer

In order to access Ethereum, you need a node. But running a node is a big deal, it requires serious hardware, bandwidth and active maintenance.

Today we have centralized solutions, but the future of Ethereum is trustless.

Accessing Ethereum



Node
Nodes are computers (physical or virtual machines) that run the software that powers Ethereum. A node requires AT LEAST 1 TB of SSD space and >500 GB of bandwidth/month.

Light Client
In the future, Ethereum will support lightweight nodes that can trustlessly access the EVM from hardware with limited resources or directly from an application

Node as a Service
Centralized service offering node access via an API. Same storage/bandwidth requirements as any node, plus those needed to process API requests

Users
Users access Ethereum by communicating with the service over the internet. The node processes the requests using its node on behalf of users.

ETH Network

Internet API

12:21 AM · Nov 9, 2022

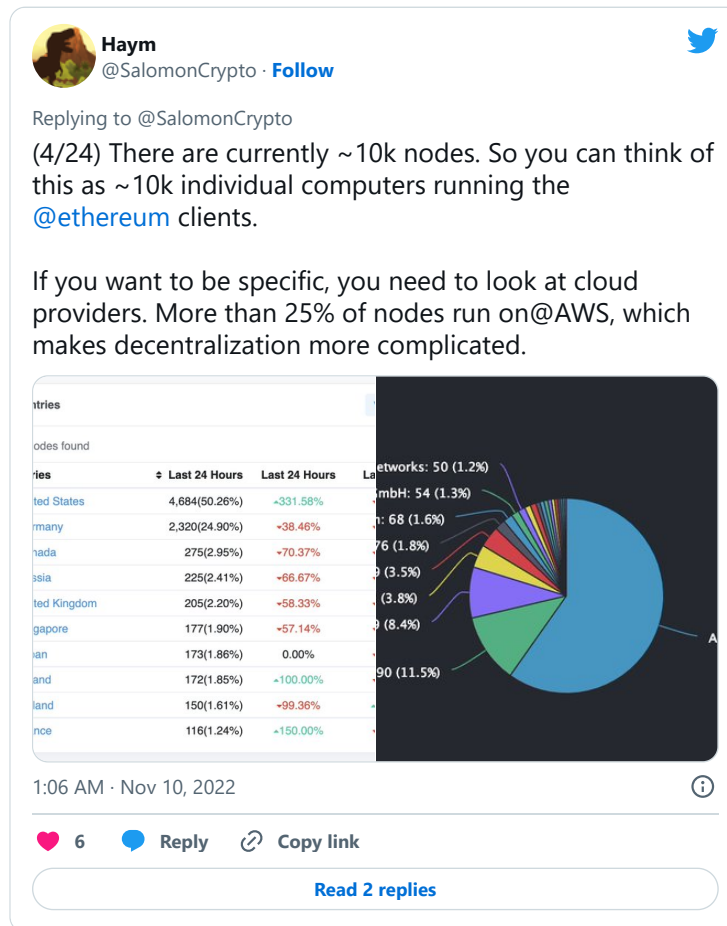
[Read the full conversation on Twitter](#)

 295  Reply  Copy link

[Read 14 replies](#)

(6/29) There are currently about ~10k nodes, meaning there are ~10k access points into [@ethereum](#).

But the VAST majority of all requests flow through just a few specific nodes. Platforms like [@AlchemyPlatform](#) allow users to interact with their nodes over an internet API.





(7/29) Left unchecked, this centralization dynamic could threaten [@ethereum](#). Why bother growing the network if the only way anyone can realistically access it is via 2 centralized, private entities?

Node-as-a-Service is a bridge solution; we need to scale direct access.

(8/29) In order to achieve the [@ethereum](#) Endgame, Ethereum must support light clients - software that can directly, independently access (and cryptographically verify) Ethereum.

In order to get there we need to develop two big concepts: statelessness and the Portal Network.

**Haym**
@SalomonCrypto · [Follow](#)



(1/24) [@ethereum](#) Light Clients and the Portal Network

In order for Ethereum to achieve its potential as the World Computer, it needs to support decentralized, trustless access. Today, that is just not possible... but the path is clear.

We are going to need the Portal Network.


Ethereum Portal Network




Ethereum exists between a network of computers (nodes), running the software that operates and secures the EVM

Today, to gather information about the state (account balance, Merkle proofs, etc) of Ethereum, users must send a computationally expensive request to a (full) node

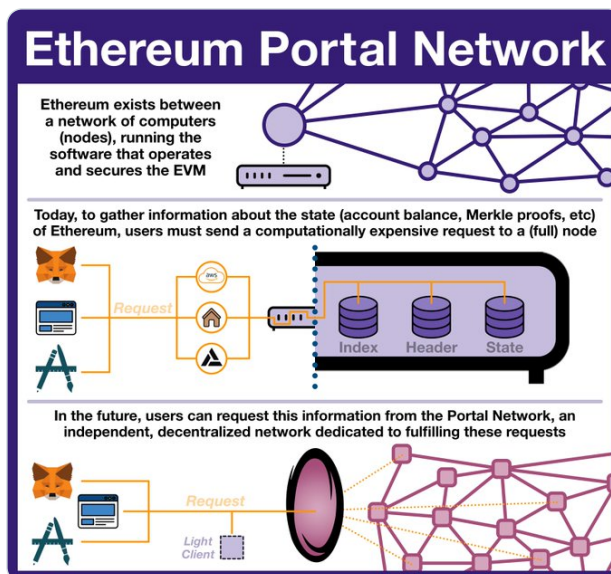
In the future, users can request this information from the Portal Network, an independent, decentralized network dedicated to fulfilling these requests

1:06 AM · Nov 10, 2022

[Read the full conversation on Twitter](#)

 175  Reply  Copy link

[Read 5 replies](#)



(9/29) Today, the internal state of [@ethereum](#) requires ~500 GB to store locally.

A stateless Ethereum will allow anyone to access its state without actually storing it, instead allowing users to directly and independently prove they are securely interacting with it.

**Haym**
@SalomonCrypto · [Follow](#)



(1/24) The Road to a Scalable, Stateless [@ethereum](#)

The World Computer is booting up. As it gets more use we are all watching the data required to manage its internal state inflate to unsustainable levels.

But Ethereum is not yet complete; let's talk about what's coming.

4:38 AM · Nov 5, 2022 


 [Read the full conversation on Twitter](#)

 200  Reply  Copy link

[Read 27 replies](#)

(10/29) Light clients need statelessness, but statelessness isn't enough.

In order to directly interact with [@ethereum](#), users need data to verify the cryptographic proofs. Instead of asking the nodes to process these (increasing node's work), we will turn to the Portal Network.

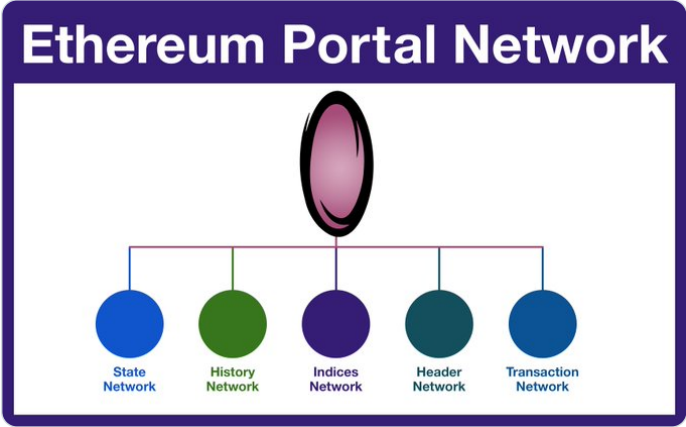
 **Haym**
@SalomonCrypto · Follow

Replying to @SalomonCrypto

(16/24) The Portal Network is a group of networks that are dedicated servicing [@ethereum](#) light clients.

Nodes in the Portal Network opt in to the specific networks they want to support with whatever resources they have available.

youtube.com/watch?v=0stc9j...



1:07 AM · Nov 10, 2022

8 ❤️ Reply ↻ Copy link

[Read 1 reply](#)

(11/29) So, statelessness + Portal Network = light clients. In this world, a light client would power on, be able to sync with [@ethereum](#) instantly (currently > a day) and immediately begin transacting.

All possible with minimal hardware and a mediocre internet connection.

(12/29) Light clients are really, really big.

Ever thought about why De-Fi/NFTs are like the only things being built? It's because accessing the EVM is so hard that once you're within you need to stay there.

A world where trustless access is trivial has infinite possibility.



(13/29) And really that's what this is all about: trust, or more specifically, the ability to interact and coordinate without relying on trust at all.

[@ethereum](#) provides the way for the internet to trustlessly express ownership, light clients will allow trustless interaction.

(14/29) Trustless is a strange concept; it's the idea that if you can independently verify everything about a system, there is no need for trust in the first place.

The entire project is about creating this trustless trust.

(15/29) Trusting in [@ethereum](#) ultimately means trusting the Ethereum node operators... and node operators are real people with human motivations.

So Ethereum issues \$ETH and coordinates around Proof of Stake. No need to trust intentions, you just need to trust the economics.



(16/29) Proof of Stake is crypto's method of transforming a trust assumption into an economic one.

You don't have to trust a node operator, you just need to know that acting in bad faith is INCREDIBLY expensive... more expensive than it could ever be worth.

(17/29) You've heard of \$ETH as ultra sound money? That's only part of the story.

In the [@ethereum](#) endgame, \$ETH isn't just the most decentralized monetary asset, it has a uniquely Ethereum property:

\$ETH is the currency of trustless trust.

(18/29) The path forward is already clear, the Middleware Gold Rush is inevitable.

Just like the California Gold Rush of 1849 began when James Marshall found gold at Sutter's Mill, the [@ethereum](#) Middleware Gold Rush will begin when [@sreeramkannan](#) launches [@eigenlayer](#) on mainnet.



(19/29) Tl;dr [@eigenlayer](#) allows a node to re-stake its staked \$ETH in order to use its excess computational capacity to provide trustless services.

Instead of a new PoS network for every trust assumption, \$ETH will become the asset of trustless trust.

(20/29) Need some examples? Let's say we want to monitor the rapidly changing oceans.

We can deploy 1000s of sensors with embedded light clients. These sensors can trustlessly post data directly to [@ethereum](#), which will be backed by the economic guarantees of re-staked \$ETH.



(21/29) Let's look at a transportation; imagine embedding light clients into a car.

The car will be able to trustlessly verify everything around it. Which cars can be trusted as a driverless car? Which garages can be opened? Who and when can a company charge for a service?



(22/29) Last one, think about container ships.

Today, each one is simply a box of metal with a barcode. Logistics is super intense and sensitive data must be shared with non-sensitive parties.

In a light client, re-staked world, trustless secure logistics becomes trivial.



(23/29) This is the endgame for [@ethereum](#). Everything we are doing is to transform \$ETH into the most pristine asset a human could ever ask for: the currency of trustless trust.

Need some more convincing? Let's go back to the old way we used to talk about Ethereum scaling.

(24/29) Previously, the most important way we talked about scaling [@ethereum](#) was by increasing the number of transactions we could execute through the EVM.

Rollups are a paradigm change; instead of using the EVM to execute, we execute elsewhere and simply settle on Ethereum.



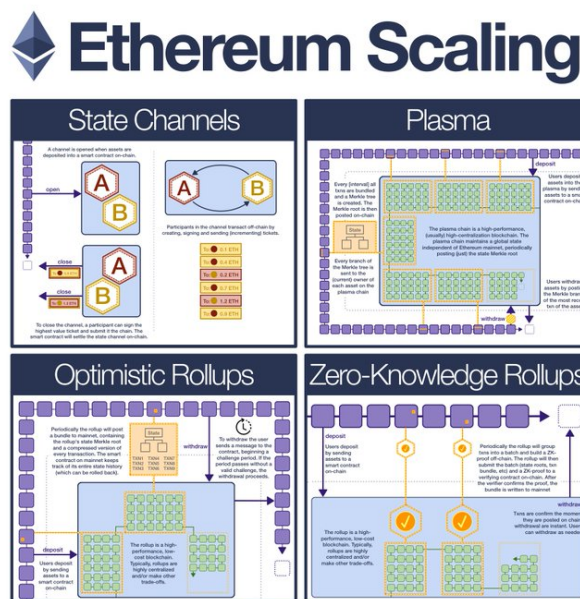
Haym
@SalomonCrypto · Follow



(1/15) [@ethereum](#) Scaling Technology

State Channels → Plasma → Optimistic Rollups → ZK-Rollups

Your guide to the technologies that will scale Ethereum from 12 to 100,000 txns/sec... at a lower cost than you pay today!



11:04 PM · Sep 12, 2022



[Read the full conversation on Twitter](#)



808



[See the latest COVID-19 information on Twitter](#)

[Read 18 replies](#)

(25/29) Settlement is the point where ownership is decided, and so the rollup paradigm works by posting the final result of any txn to mainnet.

Danksharding will introduce data blobs that DRASTICALLY increase the capacity for these receipts (and therefore Ethereum's throughput).



(26/29) Rollups are settling to @ethereum, posting their data to blobs. This data is inaccessible to the EVM, it's only important that it is posted to a shard and its data is made available to those outside the EVM.

Well... in the Ethereum Endgame, the EVM is stateless.

(27/29) Here's the question for you, dear reader:

What is the difference between a rollup and stateless EVM? Both simply post blobs of data to @ethereum shards that the beacon chain/PoS don't care about.

The answer: A stateless EVM is really just another shard.

(28/29) [@ethereum](#) is and always will be the World Computer. Today it is a slow, difficult to use computer; fortunately the path to 100k txns/second is clear.

But we did not build the World Computer for performance. We built it to achieve something humanity does not have.

(29/29) In a world with ever more complexity, coordination and automation, [@ethereum](#) is inevitable.

You just need to understand the Ethereum endgame:

Trustless Trust

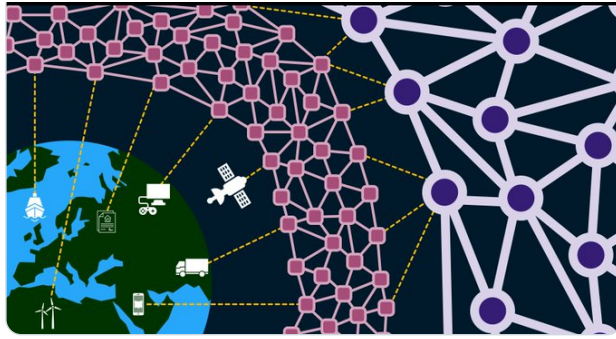
More of a long-form reader? Try this:



Like what you read? Help me spread the word by retweeting the thread (linked below).

Follow me for more explainers and as much alpha as I can possibly serve.





8:53 PM · Nov 10, 2022



[Read the full conversation on Twitter](#)



476



Reply



Copy link

[Read 8 replies](#)

...