

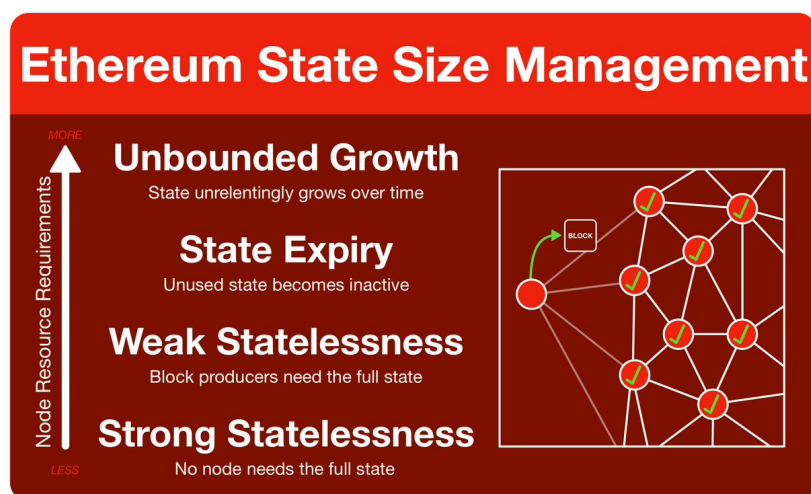


Haym @SalomonCrypto

Nov 3 · 23 tweets · [SalomonCrypto/status/1587983584471633921](https://twitter.com/SalomonCrypto/status/1587983584471633921)

(1/22) As the World Computer gets more use, the size of the EVM's state grows. Unceasing, unrelenting, unending growth. Eventually, the individual computers that make up the network wont be able to keep up.

[@VitalikButerin](#)'s theory of [@ethereum](#) State Size Management.



(2/22) [@ethereum](#) is the World Computer, a single, globally shared computing platform that exists in the space between a network of 1,000s of computers (nodes).

The nodes provide the hardware, the EVM provides the virtual computer and the blockchain records Ethereum's history.



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](#)

 987  Reply  Copy link

[Read 49 replies](#)

(3/22) The EVM sits at the center of [@ethereum](#), providing a decentralized computing platform to the world.

Everything else is all designed to construct this virtual machine and expose it to the world, so that anyone who is interested can permissionlessly interact with it.



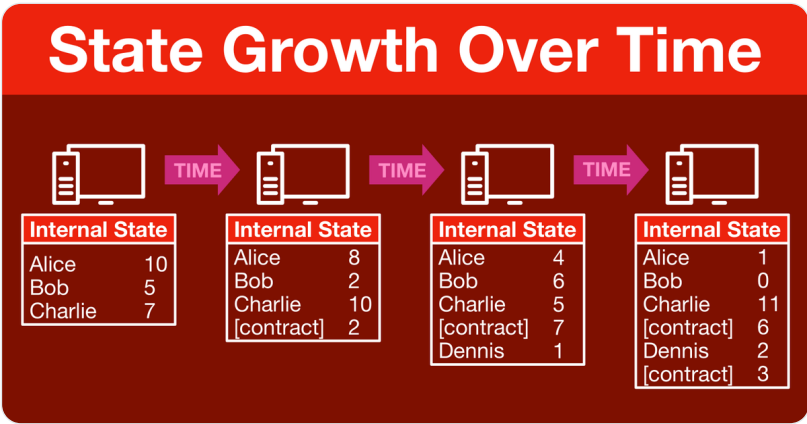
The image is a screenshot of a Twitter post. At the top left is the user's profile picture and name 'Haym' with the handle '@SalomonCrypto' and a 'Follow' button. To the right is the Twitter logo. The tweet text reads: '(1/23) @ethereum Virtual Machine (EVM)'. Below this is a paragraph: 'Ethereum is the World Computer, the future's internet-native global settlement layer. The EVM is the core of Ethereum; it provides the world in which settlement and decentralized computation happens.' This is followed by another line: 'Read on to learn about core \$ETH tech!'. The main content is a dark blue square with the text 'Ethereum Virtual Machine (EVM)' in white, bold font. Below the text is a white geometric diagram of a network with a blue Ethereum logo in the center. At the bottom of the tweet, it says '4:33 AM · Sep 27, 2022' and an information icon. Below the tweet is a link to 'Read the full conversation on Twitter'. At the bottom of the tweet area are icons for likes (381), replies, and a copy link button. A rounded rectangle at the very bottom contains the text 'Read 21 replies'.

(4/22) The World Computer is still in its infancy; it was only born ~7 years ago. And yet, even in that short time it's already attracted a huge amount of use...

...a worrying amount of use. For you see, [@ethereum](#) must keep track of every new user, smart contract, NFT, etc.

(5/22) Every new entry increases the size of its internal state (and therefore the space needed to store it).

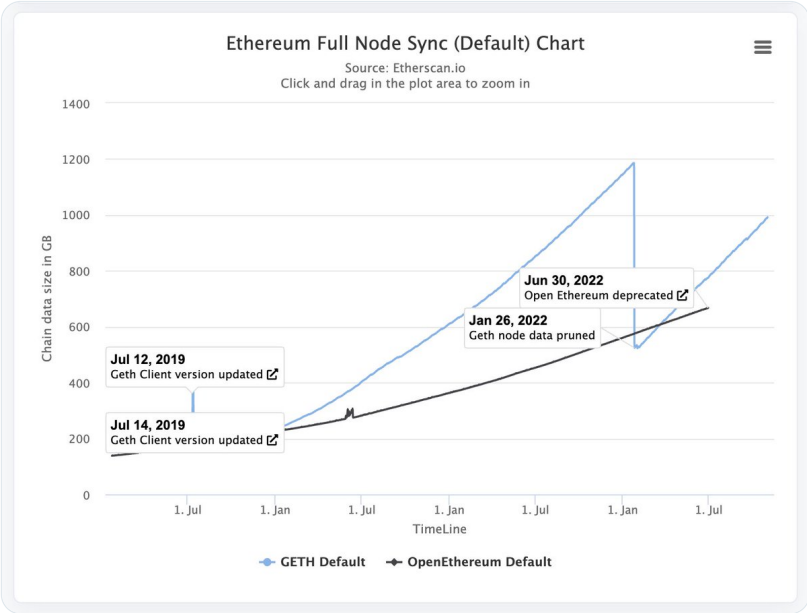
Over time, the state grows and grows, placing heavier and heavier burdens on the real computers (nodes) running the EVM.



(6/22) Again, [@ethereum](#) (which I believe will be the dominant global settlement platform for generations) was only born 7 years ago; it has already generated so much data that each node requires ~1 TB just to get caught up.

Not sustainable.

<https://etherscan.io/chartsync/chaindefault>



(7/22) This problem is a bigger issue than the storage/bandwidth requirements, it creates a vulnerability in the crypto-economics of [@ethereum](#).

A user only has to pay once to increase the state size, but the increased size will create a perpetual burden (cost) on the network.

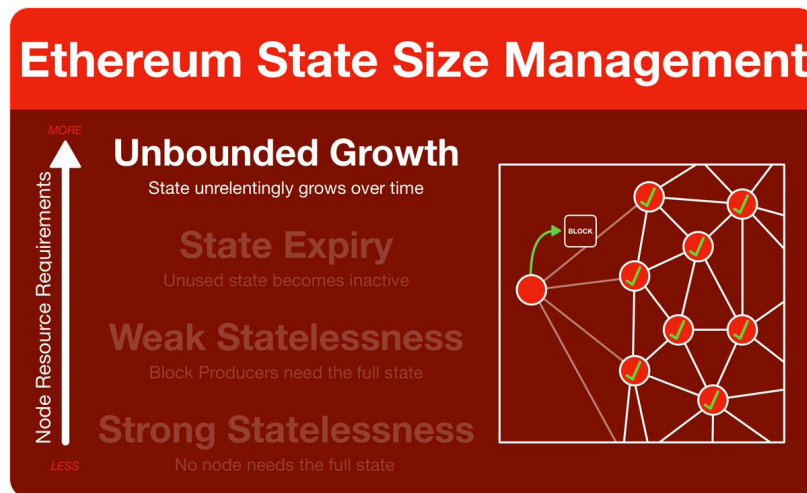
(8/22) And so, unless we find a solution, the World Computer is doomed before it even starts.

Which brings us, dear reader, to the subject of this thread: [@VitalikButerin](#)'s Theory of [@ethereum](#) state size management.



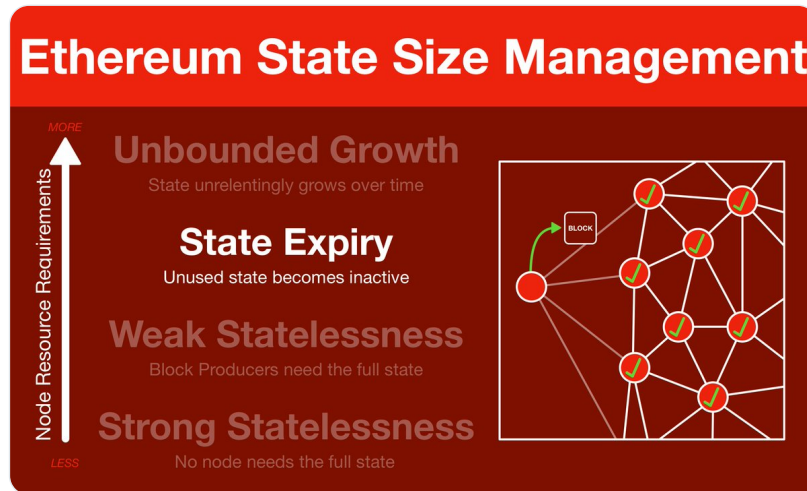
(9/22) Today, the EVM state operates in a mode of unbounded growth. Not only is this the most resource intensive way of managing the EVM state, it continues to grow over time.

This type of state management races against Moore's law, and works against decentralization.



(10/22) Instead of allowing the state of the EVM to grow into infinity, we could look at schemes that move inactive parts of the state into retirement.

State Expiry schemes do not look to delete data out of the EVM state; any inactive nodes can be cryptographically resurrected.



(11/22) The first question when considering a state expiry scheme is how to renew state. While there are many ideas, the community is moving towards a model based on "touching" the data.

In general, the idea is all data is set to expire by default and must be explicitly renewed.

(12/22) Data that expires can be revived. A resurrection would require a proof that shows the data is currently part of the inactive state.

Proof generation would require being able to access the inactive state, locally or otherwise (think [@etherscan](#) or [@AlchemyPlatform](#)).

(13/22) From here, state expiry gets much more configurable (and complicated). Some of the more important toggles:

- Expiry at the account or storage-slot level
- Creating an inactive state tree or just marking nodes as inactive
- Managing resurrection conflicts

(14/22) From here, we cross the line into statelessness.

The thread below will give you a good understanding into how it's possible, but the idea is pretty easy:

Instead of holding the entire state, nodes can trustlessly operate the EVM using only proofs.

Haym
@SalomonCrypto · Follow

(1/25) @ethereum Fundamentals: Sync Committees and Light Clients

Just about one year ago, the Altair upgrade gave validators a new duty: sync committee. Learn how this core consensus feature enables the end game:

A truly decentralized World Computer

Ethereum Light Clients

Full Node
The node builds the aggregate public key based on the full validator set and provides it to the validator

Light Client
The light client builds the aggregate public key by extracting the sync committee from the block

Public key verification ensures the block has not been tampered with

Public key verification ensures the block has not been tampered with

Validator Sync Committee

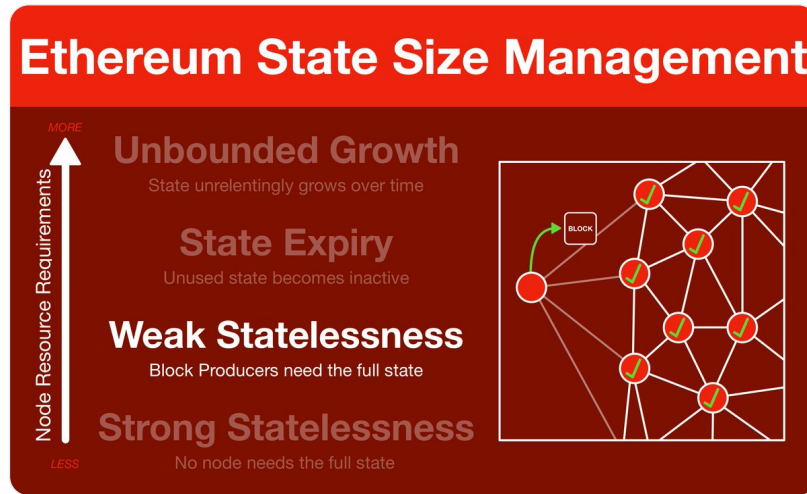
1:30 AM · Oct 10, 2022

[Read the full conversation on Twitter](#)

261 Reply Copy link

[Read 22 replies](#)

(15/22) Weak Statelessness allows all verifying nodes to be completely stateless. Nodes (or other entities) that build @ethereum blocks still need to hold the full state, specifically to generate the proof components.



(16/22) Today, verifying nodes ARE block producers, so we aren't gaining much... for now.

We already have mev-boost, a prototype of Proposer-Builder Separation (PBS). The idea: block builders become a specialized entity, who provide prebuilt blocks to proposing nodes/validators.

Haym
@SalomonCrypto · Follow

(1/26) @ethereum Roadmap: Proposer-Builder Separation

The Merge was successful, \$ETH is Proof of Stake! As the era of miners closes, we find ourselves entering a new meta: the age of MEV

Your guide to existential threat facing Ethereum... and the plan to vanquish it

The diagram illustrates the transition from a traditional Ethereum node architecture to a Proposer-Builder Separation (PBS) model. It is divided into two main sections: "Ethereum Nodes" and "Block Proposers".

- Ethereum Nodes**: Shows a node connected to a "Consensus Layer" and an "Execution Layer".
- Block Proposers**: Shows a node connected to a "Block Builders" layer, which in turn connects to the "Consensus Layer".

The diagram also includes a large purple box on the right with the Ethereum logo and the text "Ethereum Roadmap" and "Proposer-Builder Separation (PBS)".

11:38 PM · Sep 15, 2022

Read the full conversation on Twitter

529 ❤️ Reply Copy link

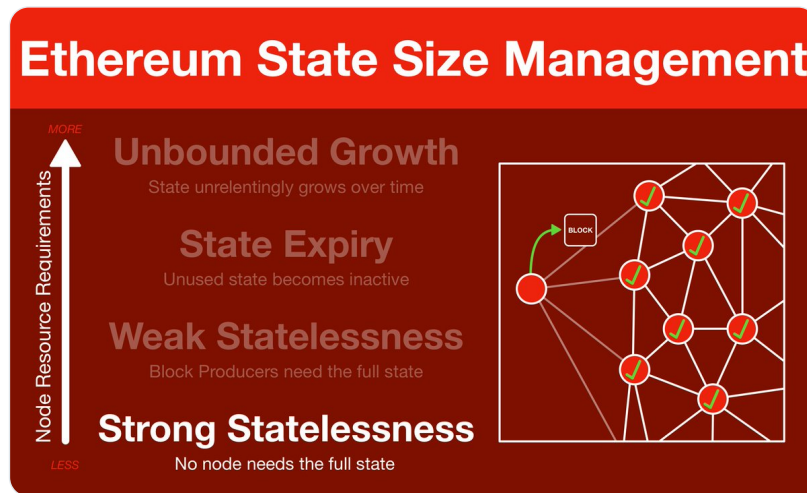
Read 12 replies

(17/22) PBS was born from MEV research, but the paradigm has become important for many of the upgrades to [@ethereum](#).

With PBS, we can implement weak statelessness without sacrificing decentralization by constraining the requirement to (possibly centralized) block builders.

(18/22) At the end of the spectrum is Strong Statelessness, the idea that no node (or block producer) needs to hold the state.

Every participant can be 100% stateless.



(19/22) Instead of using a local copy of the EVM state to validate txns, txn senders would provide proofs that guarantee the validity of their txns.

Txn senders (or, realistically, 3rd parties) would be responsible for storing enough of the state tree to generate proofs.

(20/22) This dynamic highlights a core property of state size management: there are many options creating complicated trade-offs.

For example, strong statelessness moving so much burden from the network to those using it.

(21/22) Taking a step back, any move towards statelessness increases the amount of data that moves through the network (bandwidth requirements).

Proofs are lightweight, but every transaction must be accompanied by 1+ proofs (components).

(22/22) And so, in conclusion, the solutions to [@ethereum](#) state size management are a spectrum. There is no "right" solution, there are only decisions to be made.

More of a long-form reader? Try this out:



Haym
@SalomonCrypto



Ethereum State Size Management

Ethereum State Size Management | Haym
(1/22) As the World Computer gets more use, the size of the EVM's state grows. Unceasing, unrelenting, unending growth. Eventually, the individual computers that make up the network wont be able to k...
<https://typefully.com/SalomonCrypto/4nAJXbO>

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.



Haym
@SalomonCrypto · [Follow](#)



(1/22) As the World Computer gets more use, the size of the EVM's state grows. Unceasing, unrelenting, unending growth. Eventually, the individual computers that make up the network wont be able to keep up.

[@VitalikButerin](#)'s theory of [@ethereum](#) State Size Management.



Ethereum State Size Management

Unbounded Growth
State unrelentingly grows over time

State Expiry
Unused state becomes inactive

Weak Statelessness
Block producers need the full state

Strong Statelessness
No node needs the full state

Node Resource Requirements ↑



1:42 AM · Nov 3, 2022

[Read the full conversation on Twitter](#)

262 Reply Copy link

[Read 13 replies](#)

...