



Haym @SalomonCrypto

Nov 5 · 25 tweets · [SalomonCrypto/status/1588752629118554112](#)

(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.

(2/24) [@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](#)

993 Reply Copy link

[Read 47 replies](#)

(3/24) 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 tweet from a user named Haym (@SalomonCrypto). The tweet is the first in a thread of 23 tweets. The text of the tweet explains that Ethereum is the 'World Computer' and the 'future's internet-native global settlement layer', and that the EVM is its core. It also includes a link to learn more about core \$ETH tech. The tweet features a large graphic with the text 'Ethereum Virtual Machine (EVM)' and a stylized Ethereum logo. The tweet has 382 likes and 22 replies.

Haym
@SalomonCrypto · [Follow](#)

(1/23) [@ethereum](#) Virtual Machine (EVM)

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.

Read on to learn about core [\\$ETH](#) tech!



Ethereum Virtual Machine (EVM)

4:33 AM · Sep 27, 2022


[Read the full conversation on Twitter](#)

382 [Reply](#) [Copy link](#)

[Read 22 replies](#)

(4/24) The internal state of the EVM is stored in a data structure known as a Merkle tree. Merkle trees are very powerful, but they are not perfect.

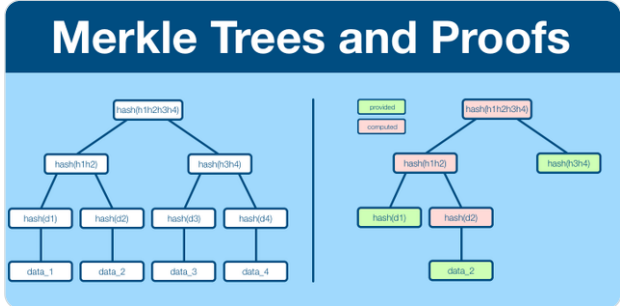
Specifically, they don't scale efficiently enough for the World Computer.

 **Haym**
@SalomonCrypto · Follow


(1/13) Computer Science 201: Merkle Trees and Merkle Proofs



If you want to understand @Bitcoin, @ethereum and blockchain technology, you need to learn:

- How a Merkle trees expresses a large dataset
- How a Merkle proof works
- Why a Markle tree is so efficient



10:17 PM · Sep 7, 2022

 [Read the full conversation on Twitter](#)

♥ 473  Reply  Copy link

[Read 17 replies](#)

(5/24) Today, the World Computer stores the entire state of the EVM; every account, every entry, all the way back to genesis. For now, that amount of data is acceptable, but without intervention the state's size will grow to infinity.

Forget decentralization, goodbye @ethereum.

(6/24) Fortunately, the gigabrain of [@ethereum](#) have been thinking about this problem for a while now. Before we talk solutions, it's helpful to think about the theory.

Tl;dr statelessness is a spectrum.

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 won't be able to keep up.

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

Ethereum State Size Management

Node Resource Requirements ↑

- 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

1:42 AM · Nov 3, 2022

[Read the full conversation on Twitter](#)

262 Likes · Reply · Copy link

[Read 13 replies](#)

(7/24) Strong statelessness is the idea that no entity, including block builders, need to store the state. While this idea is attractive for idealistic reasons, in practice going all the way to full statelessness isn't worth the effort.

Instead, we focus on the weaker version.

(8/24) The idea behind weak statelessness is that most nodes do not need to store the EVM state, as they are able to verify blocks statelessly (via proofs).

However, in order to generate these proofs, we still require block builders to hold the EVM's state.

(9/24) Today, verifying nodes ARE block producers, so weak statelessness doesn't give us much.

Fortunately, we have Proposer-Builder Separation (PBS) in the pipeline. Block builders become a specialized entity, who provide prebuilt blocks to proposing nodes/validators.

The image is a screenshot of a tweet from user Haym (@SalomonCrypto). The tweet is dated (1/26) and discusses the Ethereum Roadmap, specifically Proposer-Builder Separation. The text of the tweet reads: "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". Below the text is a diagram titled "Ethereum Roadmap Proposer-Builder Separation (PBS)". The diagram is divided into two main sections: "Ethereum Nodes" and "Block Proposers". Under "Ethereum Nodes", there are two sub-sections: "Consensus Layer" and "Execution Layer". Under "Block Proposers", there are two sub-sections: "Block Builders" and "Block Proposers". The diagram shows a flow of data and blocks between these components. To the right of the diagram is a purple box with the Ethereum logo and the text "Ethereum Roadmap Proposer-Builder Separation (PBS)". The tweet is timestamped "11:38 PM · Sep 15, 2022" and has 531 likes and 11 replies.

(10/24) 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 giving up decentralization, by constraining the requirement to (possibly centralized) block builders.

(11/24) The combination of weak statelessness + PBS gives us the advantages of statelessness without sacrificing decentralization, and so it is good enough.



And besides, if you look into implementing strong statelessness, you begin to realize it's all just trade-offs.

(12/24) Which brings us to the fundamental trade-off of any form of statelessness. State-free verification frees up a lots of node resources, but this is offset but the amount of proofs that need to flow through the system.

More statelessness = more bandwidth requirements.

(13/24) But here's the thing... both versions of statelessness just aren't possible with Merkle trees.

Back in tweet 4 I mentioned that Merkle trees weren't scalable enough for the World Computer. Here's one reason: the proof sizes are way too big to implement statelessness.

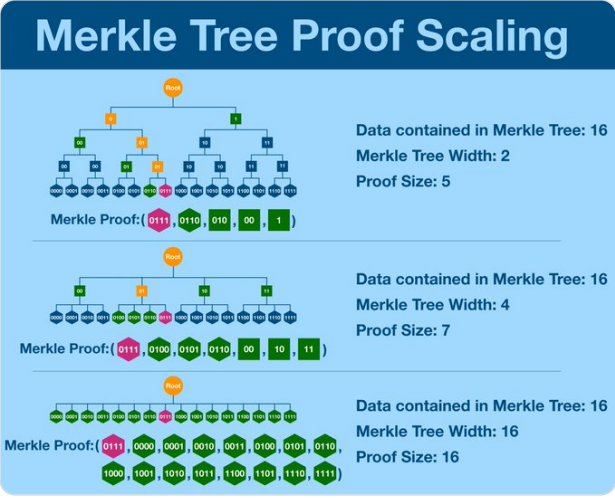
 **Haym**
@SalomonCrypto · [Follow](#) 

(1/18) The Problem with Merkle Trees

At the heart of [@Bitcoin](#), [@ethereum](#) and many blockchain computers is the Merkle Tree. While this data structure has served us well, it is not perfect... and if you look ahead, you can see impending problems.


Let's talk Merkle proof scaling.


Merkle Tree Proof Scaling






The diagram illustrates three Merkle trees of different widths, each containing 16 data points (leaf nodes). The root node is labeled 'Proof'. The trees are shown as follows:

- Tree 1:** Width 2. Proof Size: 5. Merkle Proof: (0111, 0110, 010, 00, 1)
- Tree 2:** Width 4. Proof Size: 7. Merkle Proof: (0111, 0100, 0101, 0110, 00, 10, 11)
- Tree 3:** Width 16. Proof Size: 16. Merkle Proof: (0111, 0000, 0001, 0010, 0011, 0100, 0101, 0110, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111)

4:14 PM · Oct 29, 2022 

 [Read the full conversation on Twitter](#)

 917  Reply  Copy link

[Read 57 replies](#)

(14/24) Now fortunately, we already have a replacement for Merkle trees - Verkle Trees.

Looking at a picture, a Verkle tree is much wider and shorter than a Merkle tree. Using advanced cryptography, it exploits this structure to create proofs of constant size.

Haym
@SalomonCrypto · Follow

(1/25) Cryptographic Innovation: Verkle Trees

@ethereum is the World Computer; born a rudimentary number cruncher, on a journey to (inevitably) becoming the dominant global settlement layer. And soon, Ethereum will outgrow the Merkle tree.

Tomorrow's solution: Verkle Trees

Verkle Trees

Verkle tries look and act like very wide Merkle trees

Root

Proofs remain a constant size, regardless of dataset size

Proof (vector , data)

4:49 AM · Nov 2, 2022

Read the full conversation on Twitter

672 ❤️ Reply Copy link

Read 39 replies

(15/24) As a Merkle tree grows, its proof size grows. Verkle proofs will always stay the same size, regardless of how large the tree gets.

Statelessness still increases bandwidth requirements, but Verkle trees more than offset this increase AND don't grow over time.

(16/24) We know Verkle trees are the future of @ethereum, but replacing Merkle trees turns out to be incredibly complicated.

Here's one proposal. Just click on it and scan through, you'll see it's a freaking nightmare.

eips.ethereum.org/EIPS/eip-2584

(17/24) While Verkle trees appear to be very difficult to swap in to [@ethereum](#) mainnet, it becomes MUCH simpler if we combine it with another upgrade.

Interestingly, this is another upgrade that seemed almost impossible on its own, but totally doable in parallel.

(18/24) The upgrade: state expiry! The idea that each individual piece of the EVM state naturally expires over time, and much be actively refreshed or else be deactivated.

Of course, deactivated parts of the state can always be revived later.

Haym
@SalomonCrypto · Follow

(1/20) [@ethereum](#) Roadmap: State Expiry

As the World Computer attracts more use, the amount of resources needed to keep itself running increases. Today, that burden will increase to infinity, eventually crushing the network.

A roadmap to a sustainable Ethereum.

Ethereum State Expiry

Today, the entire state (from inception through today) is stored by every node using a Merkle tree

Eventually, the tree will grow so large that it will be unusable. The tree itself requires too much space and Merkle proof size grows larger than the Ethereum network can handle

State expiry starts with a period (think ~1 year). When a new period begins, an empty Merkle tree is initialized; any state updates go in the new tree

Nodes are expected to maintain only the two most recent trees. User can access earlier trees but must provide proofs

10:48 PM · Nov 3, 2022

[Read the full conversation on Twitter](#)

414 Reply Copy link

[Read 18 replies](#)

(19/24) State expiry implicitly draws on many of the ideas (and issues) of statelessness. In order to revive inactive parts of the state, nodes must provide proofs.

As we've discussed, explicit reliance on Merkle proofs is going to create bandwidth bottlenecks (eventually).

(20/24) Verkle trees can solve this issue, what's interesting is how (most) state expiry schemes make Verkle tree implementation MUCH easier.

State expiry involves starting a new tree every period (~year), allowing Verkle trees to be phased in rather than swapped in.

(21/24) Here's how a roadmap might look:

Initialization - implement a hard fork that delineates between period 0 (before) and period 1 (after). After this fork, there will be two state trees: the old Merkle tree (no longer editable) and the new Verkle tree.

(22/24) Administration - a few things need to be done before the next period. The most important are:

- changes to the address scheme to include period information (along with other modifications and a security expansion)
- recalculation of the Merkle tree as a Verkle tree

(23/24) Finalization - implement a hard fork that begins period 2 and sets all future period lengths. Nodes are allowed to drop the Merkle tree (per the state expiry scheme), and the Merkle root is replaced with the Verkle root generated during the administration phase.

(24/24) At this point, our state expiry scheme is fully implemented and [@ethereum](#) is solely using Verkle trees and roots to store the EVM state.

And so not only has the EVM state size problem been solved, but the World Computer is ready for a stateless future.

More of a long-form reader? Try this:



Haym
@SalomonCrypto



Stateless @ethereum

Stateless @ethereum | Haym
(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 l...
<https://typefully.com/SalomonCrypto/4BYRADV>

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](#)



(1/24) 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](#)

...