



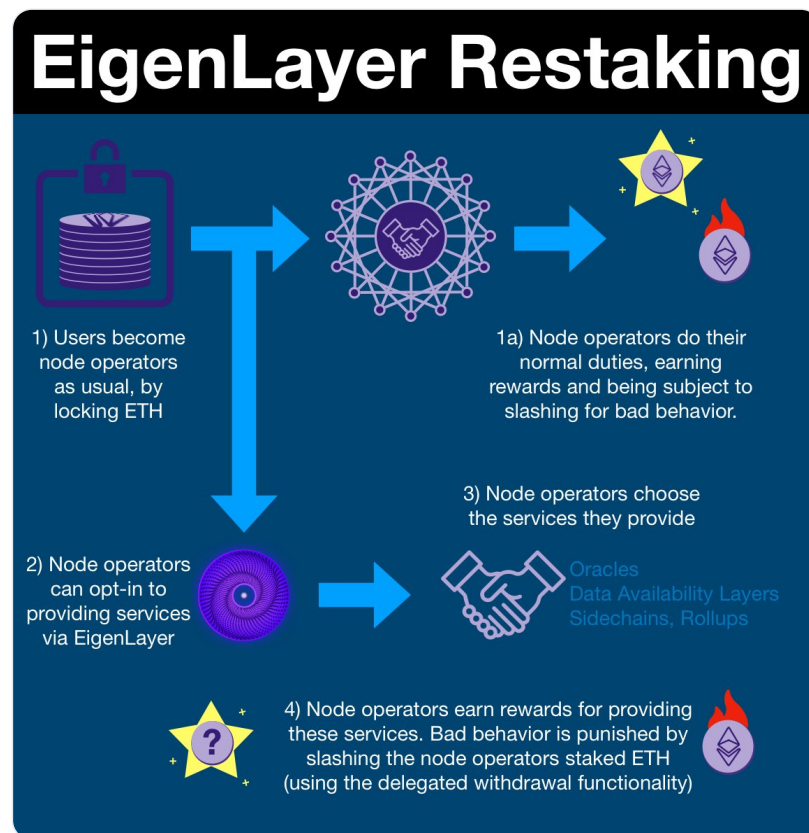
Haym @SalomonCrypto

Sep 20 · 25 tweets · [SalomonCrypto/status/1572094840619532288](https://twitter.com/SalomonCrypto/status/1572094840619532288)

(1/24) Are you liking my [@ethereum](#) roadmap posts? Do you want to further into the future, beyond the roadmap?

Bitcoin was our 0 to 1 for trustless applications. Ethereum for trustless computing.

And soon, [@eigenlayer](#) will extend \$ETH to provide generic, extendable trust.



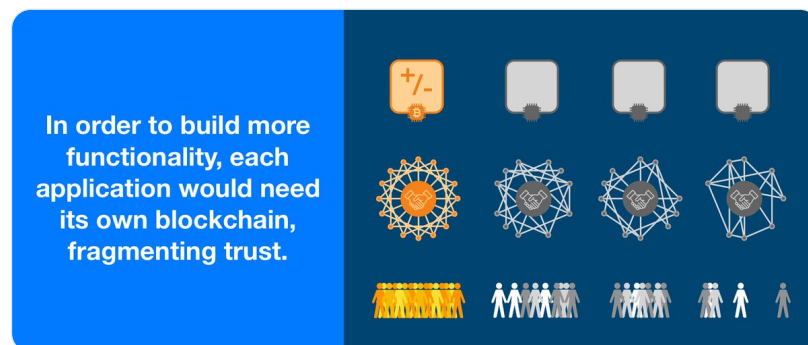
(2/24) [@Bitcoin](#) invented decentralized computation: the coordination of untrusted computers to achieve a unified computing environment.

However, Bitcoin was implemented as an application-specific blockchain computer. It's only functionality is to transfer \$BTC.



(3/24) In order to create any other application/functionality, you would have to deploy a new network with a new basis of decentralized trust.

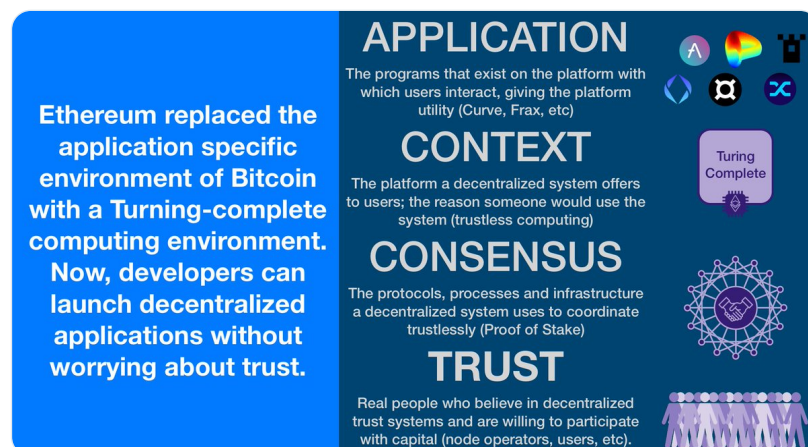
Each application would fracture the available trust further and further.



(4/24) In 2015, [@VitalikButerin](#) delivered on Satoshi's 2008 vision: generalized trustless computing.

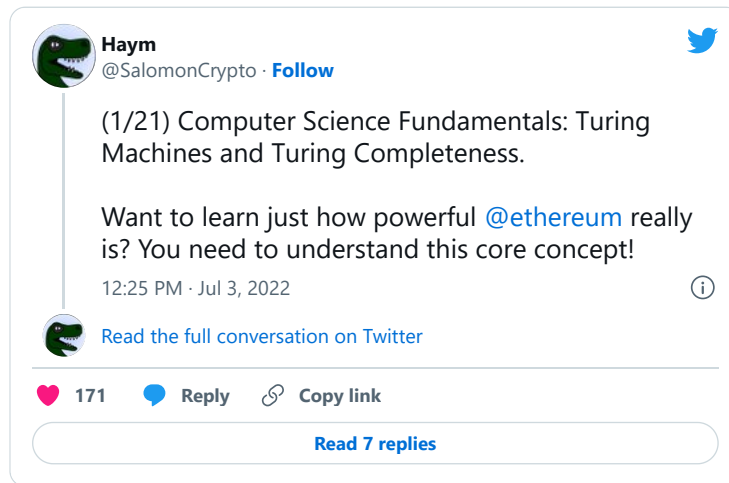
(5/24) [@ethereum](#) replaces Bitcoin's application-specific computing environment with a generalized one.

Bitcoin can only add or subtract. Ethereum is Turing-complete.



(6/24) Turing-completeness is a concept that boils down to "if a system is Turing-complete it can do anything that any other Turing-complete system can do"

We can prove that [@ethereum](#) is Turing-complete and therefore we know it is capable of everything your Macbook is capable of



(7/24) [@ethereum](#) allows a developer to deploy a decentralized application without having to build out decentralized trust network.

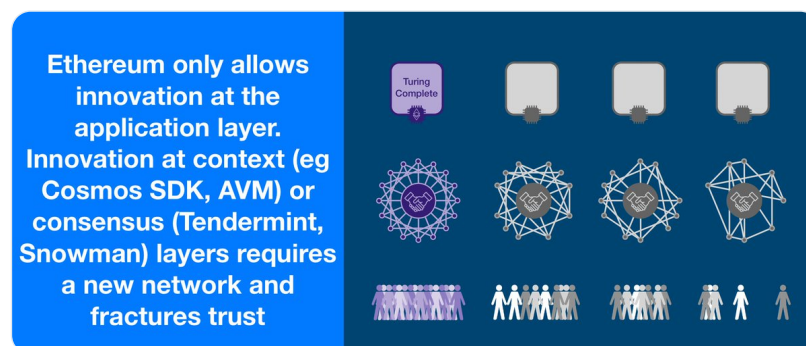
Decentralized trust becomes a resource supplied by Ethereum. All the technology, infrastructure and participants get abstracted away.

(8/24) From the perspective of an application, [@ethereum](#) provides a decentralized trust module. Developers are freed up for innovation.

Before they had to rebuild the wheel and nurture a network before they even got started. Today all that comes in the base package.

(9/24) However, [@ethereum](#) didnt solve the trust problem. The trust module is built on the first three layers of the system (trust, consensus, context).

This created a great platform for innovation, but requires a complete rebuild for changes to those 3 layers.



(10/24) Turns out [@ethereum](#) only provides trust around block production, but there are lots of other applications that need trust.

Any other trust we must supply directly through middleware.

(11/24) Middleware is any software that provides a service, information or method of communication between chains.

Middleware exists between chains and therefore must secure its own trust.

(12/24) A good example of middleware is oracles, software that transfers data in and out of [@ethereum](#)'s computing environment.

These protocols must bootstrap their own trust network... which is very hard and expensive. There's a reason why [@chainlink](#) is the only game in town.

**Haym**
@SalomonCrypto · [Follow](#)



(1/20) [@ethereum](#), Oracles and [@chainlink](#): The Communication Layer of Web3

How do smart contracts get info from outside the Ethereum blockchain? How can a protocol interact with a web2 service? How will The World Computer integrate with The Real World?

This thread has answers!



Blockchain Oracles

The diagram illustrates the role of oracles as a communication layer between the real world and the blockchain. On the left, a globe represents the 'Real World'. In the center, a wizard with a staff represents the 'Oracle'. On the right, a blue diamond shape represents the 'Blockchain'. Orange lines show data flowing from the globe to the wizard and from the wizard to the blockchain. A blue line shows data flowing from the blockchain back to the globe. The text 'Twitter: @SalomonCrypto' is at the bottom right of the diagram.

3:44 AM · Aug 10, 2022

[Read the full conversation on Twitter](#)

 378  Reply  Copy link

[Read 21 replies](#)

(13/24) Fortunately we have [@eigenlayer](#)! So now I can only find 3 videos.

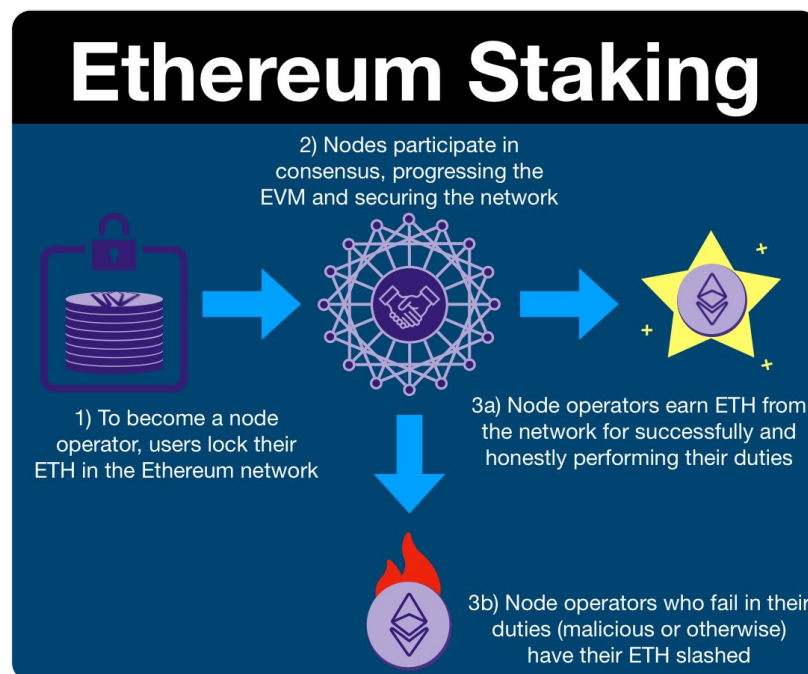
But just one is all it takes. Once you hear [@sreeramkannan](#) walk you through you'll see that this technology will eventually be in the core [@ethereum](#) protocol.



https://www.youtube.com/embed/-V-fG4J1N_M

(14/24) [@eigenlayer](#) integrates at the node level of [@ethereum](#). Nodes are where \$ETH staking takes place.

Node operators lock \$ETH in exchange for the right to operate a node. If the operator behaves and fulfills his duties, he earns \$ETH.



(15/24) If he fails in his duty (or worse, takes malicious action), a portion of the \$ETH that he locked up is slashed (permanently taken) and they are ejected from the network.

This is the mechanism by which [@ethereum](#) secures and delivers trust.

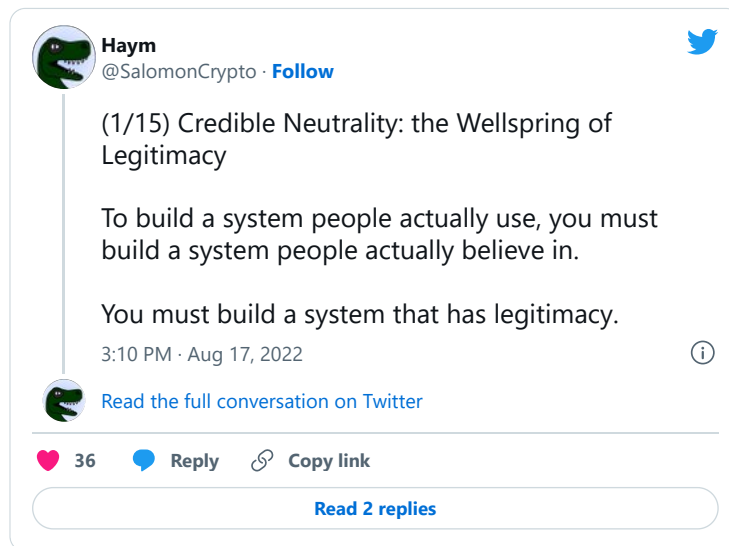
(16/24) And so, a node operator puts capital at stake and operates a node.

Nodes are real computers scattered across the globe. The requirements for running an [@ethereum](#) node are EXTREMELY low, especially considering the capital at stake.

(17/24) Requirements are kept low deliberately; every person, whether MEGACORP CEO or Aunt Phillis, can become a node operator and keep the network honest.

From decentralization flows credible neutrality.

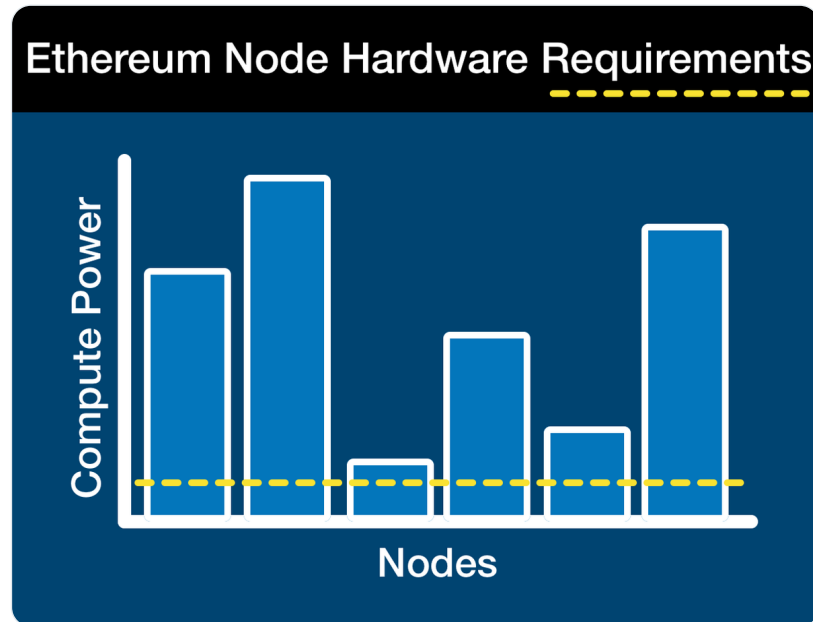
From credibly neutrality comes global dominance.



(18/24) Many node operators have extra computational power, some MUCH more.

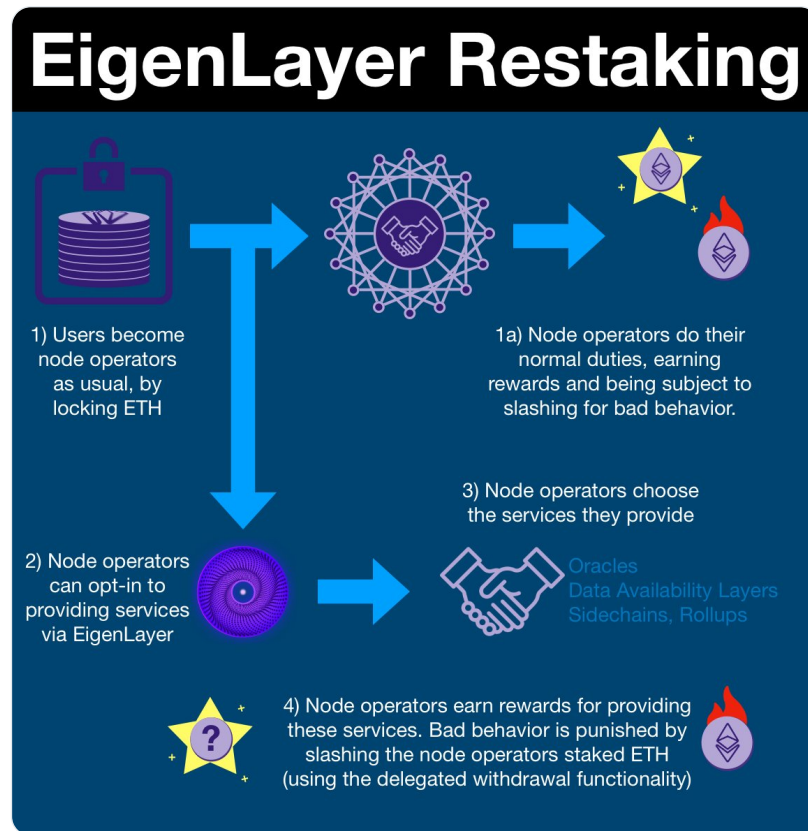
[@eigenlayer](#) allows node operators to deploy their excess computing power for middleware layers while tapping into the trust base of [@ethereum](#).

The process is called restaking.



(19/24) Node operators begin the process like normal, locking \$ETH in exchange for the right to operate a node.

Then node operators opt-in to being a service provider for [@eigenlayer](#).



(20/24) Restaking means your \$ETH is put at risk for additional slashing if the node operator misbehaves in providing these additional services

This isn't a liquid staking product, this is managed through the delegated withdrawal address.

(21/24) \$ETH stakers can set the withdrawal address of their node to a 3rd party address. When they opt-in to [@eigenlayer](#), they set their withdrawal address to an EigenLayer smart contract. This contract can then deduct any \$ETH before returning it to you (effectively slashing).

(22/24) In exchange for committing to operate the middleware and taking this extra risk, the node operator is compensated by the service provider.

Thus, the middleware can tap directly into \$ETH's trust base.


(23/24) Merge mining refers to the act of mining two or more cryptocurrencies at the same time, without sacrificing overall mining performance.


Restaking is Proof of Stake's version: merge staking.

(24/24) Below is [@ethereum](#)'s roadmap. As much as it's full of detail, it's even more full of questions.

The more I ask, the more I realize I still have to learn.

And the more I realize that Ethereum is inevitable.

**Haym**
@SalomonCrypto · [Follow](#)



(1/27) [@ethereum](#) Roadmap

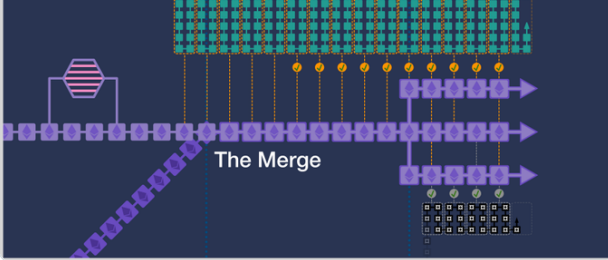
In ~24 hours, Ethereum be changed forever. But The Merge is not the end, it simply marks a new chapter; one with many more improvements. Ethereum is becoming the World Computer!

A guide to the plan that will take Ethereum from 12 to 100,000 txns/sec.


Ethereum: The World Computer


Roadmap to 100k Txns/Sec




PAST	Present	Future
State Channels Plasma EIP-1559	Optimistic Rollups ZK-Rollups MEV-Boost	AppChains Enshrined PBS Danksharding



The Merge

1:57 AM · Sep 14, 2022 


 [Read the full conversation on Twitter](#)

 1K  Reply  Copy link

[Read 49 replies](#)

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.

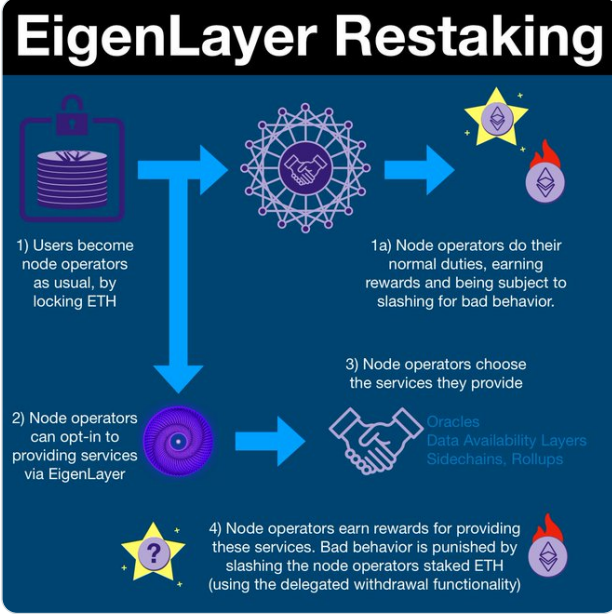
**Haym**
@SalomonCrypto · [Follow](#)

(1/24) Are you liking my [@ethereum](#) roadmap posts? Do you want to further into the future, beyond the roadmap?

Bitcoin was our 0 to 1 for trustless applications.
Ethereum for trustless computing.

And soon, [@eigenlayer](#) will extend [\\$ETH](#) to provide generic, extendable trust.

EigenLayer Restaking




The diagram illustrates the EigenLayer Restaking process in four steps:


- 1) Users become node operators as usual, by locking ETH. (Icon: ETH stack with padlock)
- 2) Node operators can opt-in to providing services via EigenLayer. (Icon: Node operator with gear)
- 3) Node operators choose the services they provide. (Icon: Handshake with list: Oracles, Data Availability Layers, Sidechains, Rollups)
- 4) Node operators earn rewards for providing these services. Bad behavior is punished by slashing the node operators staked ETH (using the delegated withdrawal functionality). (Icon: Star with question mark and ETH stack with flame)


Additional details from the diagram:


- 1a) Node operators do their normal duties, earning rewards and being subject to slashing for bad behavior. (Icon: Node operator with star and ETH stack with flame)

5:26 AM · Sep 20, 2022

 [Read the full conversation on Twitter](#)

 261

 Reply

 Copy link

[Read 9 replies](#)

...