title: Proof-of-stake (PoS) description: An explanation of the proof-of-stake consensus protocol and its role in Ethereum. lang: en

Proof-of-stake (PoS) underlies Ethereum's consensus mechanism. Ethereum switched on its proof-of-stake mechanism in 2022 because it is more secure, less energy-intensive, and better for implementing new scaling solutions compared to the previous proof-of-work architecture.

Prerequisites {#prerequisites}

To better understand this page, we recommend you first read up onconsensus mechanisms.

What is proof-of-stake (PoS)? {#what-is-pos}

Proof-of-stake is a way to prove that validators have put something of value into the network that can be destroyed if they act dishonestly. In Ethereum's proof-of-stake, validators explicitly stake capital in the form of ETH into a smart contract on Ethereum. The validator is then responsible for checking that new blocks propagated over the network are valid and occasionally creating and propagating new blocks themselves. If they try to defraud the network (for example by proposing multiple blocks when they ought to send one or sending conflicting attestations), some or all of their staked ETH can be destroyed.

Validators {#validators}

To participate as a validator, a user must deposit 32 ETH into the deposit contract and run three separate pieces of software: an execution client, a consensus client, and a validator client. On depositing their ETH, the user joins an activation queue that limits the rate of new validators joining the network. Once activated, validators receive new blocks from peers on the Ethereum network. The transactions delivered in the block are re-executed to check that the proposed changes to Ethereum's state are valid, and the block signature is checked. The validator then sends a vote (called an attestation) in favor of that block across the network.

Whereas under proof-of-work, the timing of blocks is determined by the mining difficulty, in proof-of-stake, the tempo is fixed. Time in proof-of-stake Ethereum is divided into slots (12 seconds) and epochs (32 slots). One validator is randomly selected to be a block proposer in every slot. This validator is responsible for creating a new block and sending it out to other nodes on the network. Also in every slot, a committee of validators is randomly chosen, whose votes are used to determine the validity of the block being proposed. Dividing the validator set up into committees is important for keeping the network load manageable. Committees divide up the validator set so that every active validator attests in every epoch, but not in every slot.

How a Transaction Gets Executed in Ethereum PoS {#transaction-execution-ethereum-pos}

The following provides an end-to-end explanation of how a transaction gets executed in Ethereum proof-of-stake.

1. A user creates and signs a transaction with their private key. This is usually handled by a wallet or a library such as ether.js, web3js, web3py etc but under the hood the user is making a request to a node using the Ethereum JSON-RPC API. The user defines the amount of gas that they are prepared to pay as a tip to a validator to encourage them to include the transaction in a block. The tips get paid to the validator while the base fee gets burned.
2. The transaction is submitted to an Ethereum execution client which verifies its validity. This means ensuring that the sender has enough ETH to fulfill the transaction and they have signed it with the correct key.
3. If the transaction is valid, the execution client adds it to its local mempool (list of pending transactions) and also broadcasts it to other nodes over the execution layer gossip network. When other nodes hear about the transaction they add it to their local mempool too. Advanced users might refrain from broadcasting their transaction and instead forward it to specialized block builders such as Flashbots Auction. This allows them to organize the transactions in upcoming blocks for maximum profit (MEV).
4. One of the nodes on the network is the block proposer for the current slot, having previously been selected pseudo-randomly using RANDAO. This node is responsible for building and broadcasting the next block to be added to the Ethereum blockchain and updating the global state. The node is made up of three parts: an execution client, a consensus client and a validator client. The execution client bundles transactions from the local mempool into an "execution payload" and executes them locally to generate a state change. This information is passed to the consensus client where the

execution payload is wrapped as part of a "beacon block" that also contains information about rewards, penalties, slashings, attestations etc. that enable the network to agree on the sequence of blocks at the head of the chain. The communication between the execution and consensus clients is described in more detail in Connecting the Consensus and Execution Clients.
5. Other nodes receive the new beacon block on the consensus layer gossip network. They pass it to their execution client where the transactions are re-executed locally to ensure the proposed state change is valid. The validator client then attests that the block is valid and is the logical next block in their view of the chain (meaning it builds on the chain with the greatest weight of attestations as defined in the fork choice rules). The block is added to the local database in each node that attests to it.
6. The transaction can be considered "finalized" if it has become part of a chain with a "supermajority link" between two checkpoints. Checkpoints occur at the start of each epoch and they exist to account for the fact that only a subset of active validators attest in each slot, but all active validators attest across each epoch. Therefore, it is only between epochs that a 'supermajority link' can be demonstrated (this is where 66% of the total staked ETH on the network agrees on two checkpoints).

More detail on finality can be found below.

Finality {#finality}

A transaction has "finality" in distributed networks when it is part of a block that can't change without a large amount of ETH getting burned. On proof-of-stake Ethereum, this is managed using "checkpoint" blocks. The first block in each epoch is a checkpoint. Validators vote for pairs of checkpoints that it considers to be valid. If a pair of checkpoints attracts votes representing at least two-thirds of the total staked ETH, the checkpoints are upgraded. The more recent of the two (target) becomes "justified". The earlier of the two is already justified because it was the "target" in the previous epoch. Now it is upgraded to "finalized".

To revert a finalized block, an attacker would commit to losing at least one-third of the total supply of staked ETH. The exact reason for this is explained in this Ethereum Foundation blog post. Since finality requires a two-thirds majority, an attacker could prevent the network from reaching finality by voting with one-third of the total stake. There is a mechanism to defend against this: the inactivity leak. This activates whenever the chain fails to finalize for more than four epochs. The inactivity leak bleeds away the staked ETH from validators voting against the majority, allowing the majority to regain a two-thirds majority and finalize the chain.

Crypto-economic security {#crypto-economic-security}

Running a validator is a commitment. The validator is expected to maintain sufficient hardware and connectivity to participate in block validation and proposal. In return, the validator is paid in ETH (their staked balance increases). On the other hand, participating as a validator also opens new avenues for users to attack the network for personal gain or sabotage. To prevent this, validators miss out on ETH rewards if they fail to participate when called upon, and their existing stake can be destroyed if they behave dishonestly. Two primary behaviors can be considered dishonest: proposing multiple blocks in a single slot (equivocating) and submitting contradictory attestations.

The amount of ETH slashed depends on how many validators are also being slashed at around the same time. This is known as the "correlation penalty", and it can be minor (~1% stake for a single validator slashed on their own) or can result in 100% of the validator's stake getting destroyed (mass slashing event). It is imposed halfway through a forced exit period that begins with an immediate penalty (up to 1 ETH) on Day 1, the correlation penalty on Day 18, and finally, ejection from the network on Day 36. They receive minor attestation penalties every day because they are present on the network but not submitting votes. This all means a coordinated attack would be very costly for the attacker.

Fork choice {#fork-choice}

When the network performs optimally and honestly, there is only ever one new block at the head of the chain, and all validators attest to it. However, it is possible for validators to have different views of the head of the chain due to network latency or because a block proposer has equivocated. Therefore, consensus clients require an algorithm to decide which one to favor. The algorithm used in proof-of-stake Ethereum is called LMD-GHOST, and it works by identifying the fork that has the greatest weight of attestations in its history.

Proof-of-stake and security {#pos-and-security}

The threat of a 51% attack still exists on proof-of-stake as it does on proof-of-work, but it's even riskier for the attackers. An attacker would need 51% of the staked ETH. They could then use their own attestations to ensure their preferred fork was the one with the most accumulated attestations. The 'weight' of accumulated attestations is what consensus clients use to determine the correct chain, so this attacker would be able to make their fork the canonical one. However, a strength of proof-of-stake over proof-of-work is that the community has flexibility in mounting a counter-attack. For example, the honest validators could decide to keep building on the minority chain and ignore the attacker's fork while encouraging apps, exchanges, and pools to do the same. They could also decide to forcibly remove the attacker from the network and destroy their staked ETH. These are strong economic defenses against a 51% attack.

51% attacks are just one flavor of malicious activity. Bad actors could attempt long-range attacks (although the finality gadget neutralizes this attack vector), short range 'reorgs' (although proposer boosting and attestation deadlines mitigate this), bouncing and balancing attacks (also mitigated by proposer boosting, and these attacks have anyway only been demonstrated under idealized network conditions) or avalanche attacks (neutralized by the fork choice algorithms rule of only considering the latest message).

Overall, proof-of-stake, as it is implemented on Ethereum, has been demonstrated to be more economically secure than proof-of-work.

Pros and cons {#pros-and-cons}

| Pros | Cons | | --- | -- | | Staking makes it easier individuals to participate in securing the network, promoting decentralization. validator node can be run on a normal laptop. Staking pools allow users to stake without having 32 ETH. | Proof-of-stake is younger and less battle-tested compared to proof-of-work | | Staking is more decentralized. Economies of scale do not apply in the same way that they do for PoW mining. | Proof-of-stake is more complex to implement than proof-of-work | | Proof-of-stake offers greater crypto-economic security than proof-of-work | Users need to run three pieces of software to participate in Ethereum's proof-of-stake. | | Less issuance of new ETH is required to incentivize network participants | |

Comparison to proof-of-work {#comparison-to-proof-of-work}

Ethereum has not always been a proof-of-stake network. When Ethereum started, it used proof-of-work. The switch from proof-of-work to proof-of-stake happened in September 2022. Proof-of-stake comes with benefits over proof-of-work:

- better energy efficiency – there is no need to use lots of energy on proof-of-work computations
- lower barriers to entry, reduced hardware requirements – there is no need for elite hardware to stand a chance of creating new blocks
- reduced centralization risk – proof-of-stake should lead to more nodes securing the network
- because of the low energy requirement less ETH issuance is required to incentivize participation
- economic penalties for misbehavior make 51% style attacks more costly for an attacker compared to proof-of-work
- the community can resort to social recovery of an honest chain if a 51% attack were to overcome the crypto-economic defenses.

Further reading {#further-reading}

- Proof of Stake FAQ *Vitalik Buterin*
- What is Proof of Stake *ConsenSys*
- What Proof of Stake Is And Why It Matters*Vitalik Buterin*
- Why Proof of Stake (Nov 2020) *Vitalik Buterin*
- Proof of Stake: How I Learned to Love Weak Subjectivity*Vitalik Buterin*
- Proof-of-stake Ethereum attack and defense
- A Proof of Stake Design Philosophy *Vitalik Buterin*
- Video: Vitalik buterin explains proof-of-stake to Lex Fridman

Related topics {#related-topics}

- Proof-of-work