title: Weak subjectivity description: An explanation of weak subjectivity and its role in PoS Ethereum. lang: en

Subjectivity in blockchains refers to reliance upon social information to agree on the current state. There may be multiple valid forks that are chosen from according to information gathered from other peers on the network. The converse is objectivity which refers to chains where there is only one possible valid chain that all nodes will necessarily agree upon by applying their coded rules. There is also a third state, known as weak subjectivity. This refers to a chain that can progress objectively after some initial seed of information is retrieved socially.

Prerequisites {#prerequisites}

To understand this page it is necessary to first understand the fundamentals ofproof-of-stake.

What problems does weak subjectivity solve? {#problems-ws-solves}

Subjectivity is inherent to proof-of-stake blockchains because selecting the correct chain from multiple forks is done by counting historical votes. This exposes the blockchain to several attack vectors, including long-range attacks whereby nodes that participated very early in the chain maintain an alternative fork that they release much later to their own advantage. Alternatively, if 33% of validators withdraw their stake but continue to attest and produce blocks, they might generate an alternative fork that conflicts with the canonical chain. New nodes or nodes that have been offline for a long time might not be aware that these attacking validators have withdrawn their funds, so attackers could trick them into following an incorrect chain. Ethereum can solve these attack vectors by imposing constraints that diminish the subjective aspects of the mechanism—and therefore trust assumptions—to the bare minimum.

Weak subjectivity checkpoints {#ws-checkpoints}

Weak subjectivity is implemented in proof-of-stake Ethereum by using "weak subjectivity checkpoints". These are state roots that all nodes on the network agree belong in the canonical chain. They serve the same "universal truth" purpose to genesis blocks, except that they do not sit at the genesis position in the blockchain. The fork choice algorithm trusts that the blockchain state defined in that checkpoint is correct and that it independently and objectively verifies the chain from that point onwards. The checkpoints act as "revert limits" because blocks located before weak-subjectivity checkpoints cannot be changed. This undermines long-range attacks simply by defining long-range forks to be invalid as part of the mechanism design. Ensuring that the weak subjectivity checkpoints are separated by a smaller distance than the validator withdrawal period ensures that a validator that forks the chain is slashed at least some threshold amount before they can withdraw their stake and that new entrants cannot be tricked onto incorrect forks by validators whose stake has been withdrawn.

Difference between weak subjectivity checkpoints and finalized blocks {#difference-between-ws-and-finalized-blocks}

Finalized blocks and weak subjectivity checkpoints are treated differently by Ethereum nodes. If a node becomes aware of two competing finalized blocks, then it is torn between the two - it has no way to identify automatically which is the canonical fork. This is symptomatic of a consensus failure. In contrast, a node simply rejects any block that conflicts with its weak subjectivity checkpoint. From the node's perspective, the weak subjectivity checkpoint represents an absolute truth that cannot be undermined by new knowledge from its peers.

How weak is weak? {#how-weak-is-weak}

The subjective aspect of Ethereum's proof-of-stake is the requirement for a recent state (weak subjectivity checkpoint) from a trusted source to sync from. The risk of getting a bad weak subjectivity checkpoint is very low because they can be checked against several independent public sources such as block explorers or multiple nodes. However, there is always some degree of trust required to run any software application, for example, trusting that the software developers have produced honest software.

A weak subjectivity checkpoint may even come as part of the client software. Arguably an attacker can corrupt the

checkpoint in the software and can just as easily corrupt the software itself. There is no real crypto-economic route around this problem, but the impact of untrustworthy developers is minimized in Ethereum by having multiple independent client teams, each building equivalent software in different languages, all with a vested interest in maintaining an honest chain. Block explorers may also provide weak subjectivity checkpoints or a way to cross-reference checkpoints obtained from elsewhere against an additional source.

Finally, checkpoints can be requested from other nodes; perhaps another Ethereum user that runs a full node can provide a checkpoint that validators can then verify against data from a block explorer. Overall, trusting the provider of a weak subjectivity checkpoint can be considered as problematic as trusting the client developers. The overall trust required is low. It is important to note that these considerations only become important in the very unlikely event that a majority of validators conspire to produce an alternate fork of the blockchain. Under any other circumstances, there is only one Ethereum chain to choose from.

Further Reading {#further-reading}

- Weak subjectivity in Eth2
- Vitalik: How I learned to love weak subjectivity
- Weak subjectivity (Teku docs)
- Phase-0 Weak subjectivity guide
- Analysis of weak subjectivity in Ethereum 2.0