TLDR

: We suggest a scheme for proposers to omit witnesses in the context of stateless validator execution.

Construction

Let P

be a proposer proposing a collation C

for execution by stateless validators. Instead of providing a “witness object” W

to validators, the proposer provides a “state object” S

which only contains the storage elements (storage locations and data) accessed by transactions in C

. Given C

, S

and the pre-stateroot, a stateless validator can compute a corresponding post-stateroot (or raise an exception if S

has missing or extraneous storage elements).

If a particular storage element s \in S

is invalid (i.e. does not correspond to the pre-stateroot) the next proposer can slash P

by providing a Merkle path for the correct storage data at the storage location of s

. In general, proposers are responsible for slashing other proposers in their windback, and are themselves liable to slashing if they don’t.

Discussion

When sharing a state object S

the proposer is making an easily-refutable cryptoeconomic claim that storage elements in S

faithfully corresponds to the pre-stateroot (Merkle root) without the need to share the witnesses (intermediate Merkle nodes) for storage elements (Merkle leaves).

The state object S

handles the availability part of the witness object W

, and the cryptoeconomic claim handles the validity part of the W

. Because the bulk of W

is intermediate Merkle nodes using S

instead of W

allows for synchronous stateless execution to consume significantly less bandwidth (~10x less bandwidth).