This post explores taking a “stateful” shard and removing its state trie, effectively abstracting away the EVM. The idea is to decouple collation validity from consensus, thereby putting more emphasis on log ordering, availability and bandwidth. The end result is a “log shard” with is a feature subset of “phase 1” shards, with much of the potential for scalability thanks to various state-minimisation strategies (TrueBit, Plasma, cryptoeconomic accumulators), and with significantly less implementation complexity.

The EVM of a log shard has a single LOG opcode that pushes a blob of data to a witness-friendly log accumulator. In such a log shard, a collation is an ordered list of logs. (The LOG opcode can be implicit by serialising log arrays with a separator symbol.) The only restriction for collations is a collation limit (say, 1MB). So any

blob of data with size ≤1MB is a valid collation. The fork choice rule for validators is simplified to only include:

- availability of collations
- enforcement of the collation limit
- basic housekeeping of VMC collation headers, which include the log accumulator root

To maintain the same cryptoeconomic incentives as in stateful sharding, we still need to give validators coinbase rewards and transaction fees:

- Coinbase rewards

: This is done by issuing coinbase rewards in the parent chain, in our case the main shard.

- Transaction fees

: The collation limit induces a fee market, and fees for inclusion of logs can be paid for out-of-band, e.g. using payment channels from users to validators. (I am exploring an offchain design similar to Raiden where users can pre-purchase generic “stateless gas” to trustlessly compensate any validator for including logs in collations. This is the topic of another post.)

Conclusion

We have a log shard which limits itself to log ordering, friendly log witnesses, and log real-time availability. It is useful for scalability when used in combination with a state-minimisation strategy. Maintenance of the shard requires marginal storage and CPU. As a “pure bandwidth” shard, it sits somewhere between a stateful shard and something like BitTorrent or IPFS. Compared to stateful shards, implementation is greatly simplified and the design lends itself to networking optimisations to push collation limits to the extent allowed by bandwidth resources.