Though the eth2 blockchain is going to have a 12 second slot time, and slot times need to be fairly long to ensure that validators can fairly participate without centralization risks, it is theoretically possible to give de-facto time-to-first-confirmation of under 1 second by staggering

shard blocks. If genesis is at time G

, there are S

shards and slot i

starts at time G + 12 * i

, then we could set up the system so that the shard block of shard j

is expected to appear at time G + 12 * (i + \frac{j}{S})

; that is, we could have shard blocks come at even intervals all throughout the slot.

The challenge is how to do this in a way that is game-theoretically stable. There are incentives affecting block proposers: if they wait longer, they can include more transactions that pay higher fees, but if they wait too long, they risk their block not being included on time. If a protocol treats all shards “symmetrically”, then in principle the incentives will be symmetric and so in equlibrium all shard blocks would be published at the same time. How could we adjust the protocol or the incentives to avoid this?

Possible paths for ideas:

- Require a VDF to be run with the shard block commitment as input to enforce a delay (problem: VDFs are very untested technology and we don’t trust their delays on a very fine-grained level)
- Add an incentive for shard blocks to have their headers contained in later shard blocks (eg. an incentive for the commitment to the shard j

block to be included in the shard j+16

block)

- Find some way to create an enforceable market for quick pre-confirmations (if someone wants a block on shard j

committee to quickly, they could somehow pay the proposer to quickly provide a slashing-enforceable pre-confirmation that they will include this block; the challenge is how the market would detect that this pre-confirmation actually was provided quickly)