Given the possibility of yet more changes to the sharding 1.1 spec, and developers' concerns that they are building something that could get changed again, I wanted to offer something that is worthwhile as a development target to shoot for right now

, and will be on the path toward implementing the final protocol:

1. Anyone can call addHeader(period_id, shard_id, chunks_root)

at any time. The first header to get included for a given shard in a given period gets in, all others don't. This function just emits a log.

1. For every combination of shard and period, N collators (now called "notaries") are sampled. They try to download the collation body corresponding to any header that was submitted. They can call a function submitVote(period_id, shard_id, chunks_root)

. This function just emits a log.

1. Clients read logs. If a client sees that in some shard, for some period, a chunk has been included and >= 2N/3 notaries voted for it, it accepts it as part of the canonical chain.

Notice that this protocol is extremely simple, and lacks "notary skin in the game" (slashing conditions that make it risky to vote for collations unless you actually downloaded the full data at that time) but it is under some assumptions a complete protocol, and offers an opportunity to build and test all of the base infrastructure, including:

- The capability of having 100 separate shard p2p networks, and building and sending collations across those networks
- The ability to read logs emitted by an SMC
- The ability to send transactions that call functions of the SMC
- The ability for a client to maintain a database of which collation roots it has downloaded the full body for
- The ability of a validator to (i) log in, (ii) detect that it has been randomly sampled, switch to the right p2p network, and start doing stuff, (iii) log out