

Part of this post is to reflect on the draft phase 1 spec and express how it affects my plans for development. These are just my thoughts and plans, getting work done is important, but part of that should be more collaboration. It would be good to get clarification on the spec for phases 2 and 3 of sharding: the EVM and light clients, particularly from a client development perspective (which is what I'm planning on). Obviously it's still an active area of research so you can't give a full spec or much further clarification yet, and reading through prior research will help, which I haven't fully done yet (while it's best if I don't and just implement the draft 1 phase 1 spec). I am asking partly because it's important to know how much of Parity (or other clients) will need to be rewritten. It sounds like it is going to be closer to a complete rewrite than minor changes.

eWASM will make it easier to develop the EVM of a client by providing generic computational opcodes while also providing current EVM-specific opcodes. AIUI, it looks like the wire protocol, web3, and JSON-RPC will be fairly similar, but I haven't looked into these much. But interacting with the sharding manager contract and the SMC itself (where it isn't that complicated, but interacting with it can be) are new additions. There'd possibly be a sharding P2P subprotocol.

For light client support there can be a listener of log events for the SMC.

Later on in phase 5 there's Casper which will add the finality gadget with validators and then pseudo-randomly selected proposers to replace PoW (probably with random beacons like BLS).

There'll be a lot of new additions in different phases like archival accumulators, Justin's Merkle Root Scheme (JMRS) (which AIUI would replace the current scheme), erasure codes plus proofs like zk-S(N/T)ARKs for data availability, light clients and potentially further abstraction with zk proofs, storage rent, shards within shards, load balancing, and others as listed in the phase 1 spec roadmap.

Account abstraction will make it simpler to develop a client.

Light client state protocol mechanisms like stateless clients are also not more complicated for the EVM, but introduce additional complexities such as archival nodes.

<https://ethresear.ch/t/sharding-phase-1-spec/>

The roadmap generally seems to be pretty good in terms of minimizing trade-offs early on. What I think I'll do is implement the phase 1 spec and then read further on research (or earlier or as needed). That will give more time for research for later phases.

I don't expect that it will take long to do test-driven development of the draft phase 1 spec, and do other tests e.g. unit tests. But doing API tests will probably not be practical without a phase 2 EVM, and it seems kind of pointless testing with current Ethereum 1.0 clients.