I am putting some thoughts down after a conversation with@fiiiu and @socrates1024, in which @fiiiu refused to accept my opinions without elaboration. My basic point is that we should implement two types of kettle communication - no-guarantees/best-effort and consensus - before we consider anything else.

I haven't dug too deeply on this.

By "no guarantees "I mean that we build something that just enables basic message passing. Perhaps this can be done in multiple ways if needed. The most basic use case I have in mind is a kettle running a solver sending an Eth transaction to an Eth block builder. RPC calls as they work today are probably fine. The things which seem like we may want over and above RPC transaction submission is a protocol for discovering kettles of a certain type and a protocol for proving identity+what application is being run.

Consensus I think is relatively well defined.

AIUI, in communication protocols, there is usually a performance/security/guarantees tradeoff where having more of one usually means you get less of the others. Consistency in a database (I think) is the strongest guarantee a SUAPP use case could need. The "no guarantees" case is the best performing we could get. Of course, we know you can't have consistency and availability but since Ethereum every new protocol has favoured consistency and I would imagine this trend to continue for us.

My argument consists of two points:

- we cover all use cases relatively well by implementing (n=2f+1 BFT) consensus and a best effort protocol. Every use case likely has some protocol that makes the best tradeoffs for its use case and the ideal case would be having that available for that use case.
- Since consistency is the strongest thing we can provide, it always satisfies the "communication guarantee" part.
- Consensus to get consistency might be costly in the latency category. We can accommodate for this by reducing the security parameter, n (and maybe moving nodes closer together). One motivation for why we could do this is that requiring consensus replicas to run in TEEs makes byzantine safety consistency unlikely
- For the many use cases where we just need simple message passing, we don't force the overhead of consensus so the worst case is avoided.
- Since consistency is the strongest thing we can provide, it always satisfies the "communication guarantee" part.
- Consensus to get consistency might be costly in the latency category. We can accommodate for this by reducing the security parameter, n (and maybe moving nodes closer together). One motivation for why we could do this is that requiring consensus replicas to run in TEEs makes byzantine safety consistency unlikely
- For the many use cases where we just need simple message passing, we don't force the overhead of consensus so the worst case is avoided.
- in the space of things we could implement these two types of protocols seem to have the highest bang for their buck. We have mature off-the-shelf consensus protocols that we can probably run mostly out the box (tendermint being the top example). We know that consistency is a useful property and covers many applications. We also don't have a lot of evidence that suggests there are many use cases that could benefit significantly from us putting effort into developing something weaker than a consensus protocol.
- We know from our ongoing research grant on censorship resistant DA that there are options we could pursue outside of consensus, but it seems these may require a lot of work.
- The one example we have of a decentralised orderbook (for which we speculate consensus is too expensive) is something which dYdX has spent over a year trying to figure out and feels far from a low hanging fruit.
- We know from our ongoing research grant on censorship resistant DA that there are options we could pursue outside of consensus, but it seems these may require a lot of work.
- The one example we have of a decentralised orderbook (for which we speculate consensus is too expensive) is something which dYdX has spent over a year trying to figure out and feels far from a low hanging fruit.

Of course there is a complex space of possible protocols and the jury is still out on what the CR-DA grant with Common Prefix will return, but at least this is my position for now. It might make sense to give grants to do work on other communication protocols without distracting our internal resources.