

So I've read and watched a lot of discussion about the data availability problem where an attacker holds on to some data at time of creation, but I don't remember seeing any discussion about making sure that the data is evenly distributed across all nodes for all time (just only at the start).

What I mean is, if you have a large number of shards (~1000+), for which most clients only download some small fraction of the total data, how can you be sure some of data isn't permanently lost if all clients eventually have to clear their data and re-download or prune to save space. I'm thinking on the decades to centuries timescale.

I guess a small number of archival nodes might be setup purely by altruistic people. Is this a threat to the network? In a highly sharded network, don't you need the history to rebuild the state to act on it? For popular parts of the state, I can see some always sticking around, but what if there are parts of the history that haven't been touched in a very long time. Like someone wants to make a transaction to a contract that hasn't been used in decades.

I know you can use eraser codes to regenerate parts of blocks, but what if whole blocks (or the shard equivalent of blocks) get forgotten? It could create usability or consistency issues for long range contracts.