History Expiry @ f23g

This is mostly a *forward looking* session on how clients will expand upon their ability to prune pre-merge history and begin pruning post-merge history.

Agenda

Decide data format for post-merge history

- Major areas that will guide the decision:
    - Main use cases for files — is it archive sync, ad hoc access, or other? If only archive sync, there is little value for including receipts because they will be regenerated. What amount of unused data is acceptable to EL, e.g. suppose we use Era format with beacon block data and state and this is 50% of data required to start EL archive node.
    - Value of EL <> CL alignment — every additional format adds overhead at every step. We should strive for the minimal number of formats that achieve our goals.
    - Value of uniform format for pre and post merge — do clients value this?
- Era1 format — current data format for pre-merge data. Some unneeded data types, but they are small, e.g. 32 byte root for accumulator, 8192*32=262kb of values for total difficulty. We could potentially reuse this directly.
- Era format — the original Era format has traditionally been used for sharing CL history, but post-merge it also includes all data necessary to re-execute EL blocks. We may consider an optional receipts extension so consumers can choose to download directly or recompute, depending on their use case.
    - Note: this depends somewhat on the size of CL to EL data. Since most people will only care about EL data to recreate archive indexes, we shouldn't force them to download a lot of CL data that they won't use. However there is economies of scale for using the same format across the stack.

- E2hs format — a newer format designed to seed Portal network via bridge nodes. It has proofs included, however the proof format changes depending on block height. The general file structure is consistent from genesis to tip though.
- Parquet —
- Some combination of above — we can create something new based on one of the above formats

Choose an expiry strategy

- Rolling expiry — requires online pruning of data after the cutoff. No coordination needed beyond initial data format and retrieval sources.
  - If we do this, do we need to target a coordinated release date or can clients add support when they have availability?
- Fork-based expiry — use our tried and true method of shipping: hard forks. Higher coordination burden, but also higher guarantee support across all clients.

Portal as a retrieval mechanism

- EF discontinuing support of Portal
  - The standalone client Trin has been discontinued, efforts shall be redirected towards client integration.
  - Geth will continue integrating Portal history network, other clients welcome to join.
- Other mechanisms will exist and be supported by various client teams, institutions, and bittorrent which are based around the agreed upon archival data format, e.g. `era1` , etc.
- At what point do clients view Portal, or another decentralized p2p style retrieval mechanism a requirement? Will clients proceed with rolling expiry without integration in Portal?

Non-blocking progress

- Pre-merge history had some headwinds which required us to generally work together
    - Clients had no good way to indicate what history they are actually storing. eth/69 will resolve this going forward, but we also decided later that this may not have been as big of an issue as originally thought.
    - Receipts were a critical mechanism for bootstrapping CL deposit roots and we didn't want to inhibit that process. Now with EIP-6110 this is no longer a concern.
    - Clients chose to wait for a robust decentralized p2p solution, but when combined with above, led to multi-year delay in making progress on history expiry.
- Going forward, what are client's lines-in-the-sand regarding history expiry?
    - Assume agreement is reached on the format and strategy (i.e. rolling vs. fork-based), is it fair to say that clients can begin expiring history as they see fit?
    - What are factors that would impact this decision? Any specific metrics or requirements clients would want to see from era providers? Is a single client supporting Portal enough to have confidence?