Data retrievability and pruning

The purpose of data availability layers such as Celestia is to ensure that block data is provably published, so that applications and rollups can know what the state of their chain is, and store that data. Once the data is published, data availability layersdo not inherently guarantee that historical data will be permanently storedand remain retrievable.

In this document, we discuss the state of data retrievability and pruning in Celestia, as well as some tips for rollup developers in order to ensure that syncing new rollup nodes is possible.

Data retrievability and pruning in celestia-node

As of version v0.13.0, celestia-node has implemented a light node sampling window of 30 days, as specified inCIP-4 . This means that once pruning is implemented, light nodes will now only sample blocks within a 30-day window instead of sampling all blocks from genesis. This change introduces the concept of pruning to celestia-node, where data outside of the 30-day window may not be stored by light nodes, marking a significant update in how data retrievability and storage are managed within the network (v0.13.0 release notes).

Data blobs older than the recency window will be pruned by default on light nodes, after pruning is fully implemented, but will continue to be stored by archival nodes that do not prune data. Light nodes will be able to query historic blob data in namespaces from archival nodes, as long as archival nodes exist on the public network.

Once pruning is fully implemented, light nodes will only perform data availability sampling for blocks within the data recency window of 30 days.

Suggested practices for rollups

Rollups may need to access historic data in order to allow new rollup nodes to reconstruct the latest state by replaying historic blocks. Once data has been published on Celestia and guaranteed to have been made available, rollups and applications are responsible for storing their historical data.

While it is possible to continue to do this by using theGetAll API method in celestia-node on historic blocks as long as archival nodes exist on the public Celestia network, rollup developers should not rely on this as the only method to access historical data, as archival nodes serving requests for historical data for free is not guaranteed. Below are some other suggested methods to access historical data.

- Use professional archival node or data providers.
- It is expected that professional infrastructure providers will provide paid access to archival nodes, where historical data can be retrieved, for example using theGetAll
- API method. This provides better guarantees than solely relying on free archival nodes on the public Celestia network.
- Share snapshots of rollup nodes.
- Rollups could share snapshots of their data directories which can be downloaded manually by users bootstrapping new nodes. These snapshots could contain the latest state of the rollup, and/or all the historical blocks.
- Add peer-to-peer support for historical block sync.
- A less manual version of sharing snapshots, where rollup nodes could implement built-in support for block sync, where rollup nodes download historical block data from each other over a peer-to-peer network.* Namespace pinning.
-
    - In the future, celestia-node is expected to allow nodes to choose to "pin" data from selected namespaces that they wish to store and make available for other nodes. This will allow rollup nodes to be responsible for storing their data, without needing to implement their own peer-to-peer historical block sync mechanism. [][Edit this page on GitHub] Last updated: Previous page The lifecycle of a celestia-app transaction Next page Data availability FAQ []