Integrate Celestia for service providers

This document is for third-party service providers, such as custodians and explorers, integrating the Celestia network.

Getting started

When getting started Celestia, we recommend checking out these resources first:

- Introduction to Celestia
- Monolithic v. Modular
- Celestia's DA Layer
- Learn modular
- Overview to running nodes on Celestia
- Build whatever

Celestia service provider notes

Celestia is a fairly standard Cosmos-SDK based chain. We use the latest version of Tendermint and the Cosmos-SDK, with only minor modifications to each. This means that we are:

- Using the default Cosmos-SDK modules: auth, bank, distribution, staking, slashing, mint, crisis, ibchost, genutil, evidence, ibctransfer, params, gov (limited in some TBD capacities), upgrade, vesting, feegrant, capability, and payment.
- Use the standard digital keys schemes provided by the Cosmos-SDK and Tendermint, those being secp256k1 for user transactions, and tm-ed25519 for signing and verifying consensus messages.

While exactly which modules used is subject to change, Celestia aims to be as minimal as possible.

Custody and key management

Celestia supports many already existing key management systems, as we rely on the Cosmos-SDK and Tendermint libraries for signing and verifying transactions. Learn more in the Cosmos-SDK documentation

RPC and querying

In celestia-app, only the standard RPC endpoints for Tendermint and the Cosmos-SDK are exposed. We do not currently add or subtract any core functionality, but this could change in the future. The same goes for querying data from the chain.

In celestia-node, the Data Availability node client, there is a JSON-RPC API that allows you to interact directly with Celestia's Data Availability layer. Learn how to use the API in this tutorial.

Compatibility

Linux, particularly Ubuntu 20.04 LTS, is the most well tested. Potentially compatible with other OSs, but they are currently untested. Some of the cryptography libraries used for erasure data are not guaranteed to work on other platforms.

Syncing

Since we utilize Tendermint and the Cosmos-SDK, syncing the chain can be performed by any method that is supported by those libraries. This includes fast-sync, state sync, and quick sync.

Notable exceptions relative to other blockchains

Relative to other Tendermint based chains, Celestia will have significantly longer blocktimes of roughly 12* seconds. The reason behind this block time is to optimize the bandwidth used by light clients that are sampling the chain, and is not because we have modified Tendermint consensus in any meaningful way. Validators will likely download/upload relatively large blocks. It should be noted that while these blocks are large, very little typical blockchain state execution is actually occurring on Celestia. Meaning that the bandwidth requirements will likely be larger than that of a typical Cosmos-SDK based blockchain full node, the computing requirements should be similar in magnitude.

*Subject to Change [][Edit this page on GitHub] Last updated: Previous page Integrating Wallets for developers Next page celestia-node api documentation []