CometBFT

IPC Spec - CometBFT

This document explains the relationship between Fendermint and CometBFT.

What is CometBFT?

CometBFT is ageneric blockchain engine using the Tendermint consensus. What this means for us has been argued in the originalsolution concept that introduced it (still as Tendermint Core at the time) for use for IPC.

Bygeneric we mean that CometBFT doesn't say anything about the format and content of thetransactions and theledger of the replicated state machine; that is completely within the jurisdiction of theapplication , which is Fendermint.

The two communicate over the Application BlockChain Interface (ABCI). We are currently usingversion 0.37 withABCI v1 ; version 0.38 moves toABCI v2 , which will unlock new capabilities.

To get an idea of how little CometBFT says about what the transactions and the ledger should look like, have a look at thekvstore example in the guide; there are no signatures, the transactions are literally just strings formatted as"key=value" . Check outthese steps to see how to get started with CometBFT by running the built-inkvstore application.

A really good overview of the Tendermint consensus lifecycle in CometBFT is provided by the following diagram:

?

What is Fendermint?

Fendermint implements a ledger using the ABCI interface, driven by CometBFT. It contains the definition of transactions and queries, and uses the FVM to execute messages and store the data as IPLD.

See theIPS Spec - Executions andIPC Spec - Snapshots to get a detailed overview of how the ABCI methods are implemented, and theArchitecture diagram for the component layout.

Common Misconceptions

There were some questions we should clarify, as they are not immediately clear for someone not familiar with how CometBFT works.

How many CometBFT instances per Fendermint?

There is a 1-to-1 relationship between CometBFT and Fendermint. CometBFT feeds the blocks to Fendermint for execution. Fendermint maintains the application state with deterministic execution of transactions. Fendermint is not a standalone application, with CometBFT as some kind of a front-end, like the Ethereum API facade. It is just an execution engine.

Can Fendermint instances collude and lie to CometBFT?

This assumes that Fendermint instances communicate between each other in a Byzantine way, present their version of truth to CometBFT, and then the question is what happens if Fenderrmint isn't majority honest. No, it doesn't work that way.

Fendermint instances do not run consensus between themselves, that is precisely the job of CometBFT. Each CometBFT is associated with a single dedicated and trusted Fendermint instance that it owns, and communicates with other CometBFT instances about the results of their respective transaction executions.

How many subnets does Fendermint run?

Each CometBFT+Fendermint chain is a specific subnet. To run a subnet, the operators must launch a separate blockchain.

Does Fendermint call CometBFT through ABCI?

No, the ABCI application does not normally make RPC calls to CometBFT, the communication is strictly initiated by CometBFT, the application acts as the server reacting to events happening in consensus.

Fendermint itself calls CometBFT via JSON-RPC to enquire about the history of the power table, and the status of the chain syncing, but that has to happen outside the ABCI cycle.

Process Model

CometBFT is written in Go, while Fendermint is written in Rust. Had Fendermint been written in Go, we would have the opportunity to compile afendermint binary that runs CometBFT with an in-process implementation of the Fendermint ABCI ledger. This is howCosmos SDK applications work. In our case, however, we have to run two distinct processes:cometbft

andfendermint , with the communication happening over TCP. They can be deployed on the same machine, or separately.

In Docker they are two distinct containers, withcometbft being one of the official images. This has the advantage that there is no need to audit it, because we haven't changed anything.

Sometimes the question comes up whether thefendermint process should somehow owncometbft and try to hide the fact that it even exists to simplify its operation and deployment - in this regard it would make it more similar to the experience of native Cosmos SDK apps written in Go, which are executed in the CometBFT process itself.

It is worth keeping in mind in this context thatsecurely deploying CometBFT in production is a task left for engineering. Fendermint doesn't contain any tools or documentation to do this for the user.

For this reason it is worth highlighting the need for operators to familiarise themselves with CometBFT. Trying to hide CometBFT as an implementation detail would risk obscuring it, and making it more difficult for them to find expert advice on operating their blockchain.

When in need for inspiration, one can look at thePenumbra deployments , which have a similar split between a Rust application and the CometBFT engine.

Previous Addressing Next IPLD Resolver Last updated5 days ago On this page *IPC Spec - CometBFT * What is CometBFT? * What is Fendermint? * Common Misconceptions * Process Model