The concept behind Dymension is to support multiple DA (Data Availability) layers, allowing a rollapp creator to choose the most suitable one based on criteria such as decentralization, costs, and other factors.

Currently, the initial version only supports Celestia, and the fault proving mechanism is governed by the governance module, making the system permissioned rather than permissionless.

Fault proving involves multiple steps, but for this discussion, we are focusing on proving the inclusion of a specific block in a DA layer, which is the initial step in the fault proving process.

Therefore, we face two primary issues:

1. The lack of choices for DA Layers.
2. This is due to the absence of standardization for how a DA layer can be represented arbitrarily.

Standardizing a DA Layer in Dymension

The first step is to understand how a DA (Data Availability) Layer is utilized in Dymension. There are two main actors that interact with the DA Layer interface:

1. Dymension network: The settlement layer uses the DA layer in a “lazy” manner by storing a commitment indicating that a sequencer has posted data on the DA. This commitment can later be used to dispute the committed data, the end state, and other relevant aspects.
2. Dymint: This component actively uses the DA layer to store blocks, compute the commitment and send it to the Dymension network.

From Dymint’s perspective, the standardization is straightforward: the DA layer serves as a repository for storing arbitrary data.

However, from the Dymension network’s perspective, the interaction becomes more complex and intriguing.

DA Layers as IBC Clients

From the Dymension network’s perspective, a DA (Data Availability) Layer can be abstracted into two core functions:

1. UpdateDAState: Given a trusted block from the Dymension network, this function verifies a more recent block and updates it as the latest trusted block.
2. VerifyMembership: This function validates, given a block and an inclusion proof, that the state (in the form of a key and value) of that inclusion proof belongs to the given block.

These two functions are already managed by IBC (Inter-Blockchain Communication).

The DA Layer can thus be abstracted as an IBC client implementing a consensus validation mechanism. We can then build the membership verification functionality on top of this.

Practically, when a sequencer updates the state of the DA, it provides Dymension with the latest block of the DA layer it posted to. The IBC client implementation of the DA within Dymension will update its internal latest block state accordingly. This updated state can then be used to verify membership proofs when the sequencer posts blob commitments.

Da Modularity: Interchain DA

Interchain-DA is the first new pluggable DA (Data Availability) layer built on top of Dymension, utilizing the generic DA Layer standard.

Interchain-DA aims to align Dymension with the existing Cosmos ecosystem. It is a Cosmos SDK module that can be integrated into any Cosmos chain, enabling that chain to become a DA provider.

This integration allows Cosmos chains to offer DA services to Dymension rollapps.

The module provides integrators with flexibility to decide on:

- Fees for blob posting
- Data retention time

Both of these parameters can be controlled through governance.

Advantages of using cometbft and cosmos-sdk powered chains

The Cosmos ecosystem has an abundance of block space. Instead of producing empty blocks, chains can allocate a portion of their block size to provide DA (Data Availability) services to rollapps.

CometBFT-powered chains offer immediate finality, allowing the Dymension network to promptly settle the blob inclusion aspect. This ensures that, in case of disputes, only the execution needs to be proven, not the availability of the block itself.

Efficiency

The Interchain-DA module is designed to be as lightweight as possible. It only stores fingerprints of the blobs in the Merklized state, preventing increased node running requirements for the chain.

The actual blob data is stored in a separate, non-Merklized state, similar to how CosmWasm blobs are stored.

Blobs have a time-to-live (TTL), and once this period expires, the blobs are garbage collected. This ensures that running the interchain DA module does not cause linear increase in node running requirements.

Usage in Dymension

The DA provider chain connects to Dymension through an IBC connection. As sequencers post data on the DA provider, they also update the state of the IBC client.

This setup enables Dymension to instantly verify the correct inclusion of the Blob when a rollapp state is updated using the Interchain-DA module.