Data availability FAQ

What is data availability?

Data availability answers the question, has this data been published? Specifically, a node will verify data availability when it receives a new block that is getting added to the chain. The node will attempt to download all the transaction data for the new block to verify availability. If the node can download all the transaction data, then it successfully verified data availability, proving that the block data was actually published to the network.

As you'll see, modular blockchains like Celestia employ other primitives that allow nodes to verify data availability more efficiently. Data availability is critical to the security of any blockchain because it ensures that anyone can inspect the ledger of transactions and verify it. Data availability becomes particularly problematic when scaling blockchains. As the blocks get bigger, it becomes impractical for normal users to download all the data, and therefore users can no longer verify the chain.

What is the data availability problem?

The problem with data availability occurs when the transaction data for a newly proposed block cannot be downloaded and verified. This type of attack by a block producer is called a data withholding attack, which sees the block producer withhold transaction data of a new block.

Since transaction data is withheld, nodes cannot update to the latest state. Such an attack can have numerous consequences, from halting a chain to gaining the ability to steal funds. The severity of the consequences will depend on the type of blockchain (L1 or L2) and whether data availability is kept onchain or offchain. The data availability problem commonly arises around L2 scaling solutions like rollups and validiums.

How do nodes verify data availability in Celestia?

In most blockchains, nodes that verify data availability do so by downloading all transaction data for a block. If they are able to download all the data, they have verified its availability. In Celestia, light nodes have access to a new mechanism to verify data availability without needing to download all the data for a block. This new primitive for verifying data availability is called data availability sampling.

What is data availability sampling?

Data availability sampling is a mechanism for light nodes to verify data availability without having to download all data for a block. Data availability sampling (DAS) works by having light nodes conduct multiple rounds of random sampling for small portions of block data. As a light node completes more rounds of sampling for block data, it increases its confidence that data is available. Once the light node successfully reaches a predetermined confidence level (e.g. 99%) it will consider the block data as available.

Want a simpler explanation?Check out this thread on how data availability sampling is like flipping a coin.

What are some of the security assumptions that Celestia makes for data availability sampling?

Celestia assumes that there is a minimum number of light nodes that are conducting data availability sampling for a given block size. This assumption is necessary so that a full node can reconstruct an entire block from the portions of data light nodes sampled and stored. The amount of light nodes that are needed will depend on the block size - for bigger blocks more light nodes are assumed to be running.

A second notable assumption that is made by light nodes is that they are connected to at least one honest full node. This ensures that they can receive fraud proofs for incorrectly erasure coded blocks. If a light node is not connected to an honest full node, such as during an eclipse attack, it can't verify that the block is improperly constructed.

Why is block reconstruction necessary for security?

In Celestia, blocks need to be erasure coded so that there is redundant data to aid the data availability sampling process. However, nodes tasked with erasure coding the data could do so incorrectly. Since Celestia uses fraud proofs to verify that erasure coding is incorrect, the full block data is needed to generate a bad encoding fraud proof.

There could be a situation where validators only provide data to light nodes and not full nodes. If the full nodes don't have the ability to reconstruct the full block from the portions of data stored by light nodes, they wouldn't be able to generate a bad encoding fraud proof.

What is data storage?

Data storage is concerned with the ability to store and access past transaction data.

Data storage and retrieval is needed for multiple purposes, such as:

- Reading the information of a previous transaction
- Syncing a node
- Indexing and serving transaction data
- Retrieving NFT information

What is the problem around data storage?

The issue with data storage is whether past transaction data can be stored and successfully retrieved at a later time. The inability to retrieve historical transaction data can cause problems, such as users being unable to access information about their past transactions or nodes that cannot sync from genesis. Luckily, the assumptions around storing and accessing past data are weak. Only a single copy of a blockchain's history needs to be accessible for users to gain access to historical transaction data. In other words, data storage security is a 1 of N honesty assumption.

What is the difference between data availability and data storage?

Data availability is about verifying that transaction data for a new block is public and available. In contrast, data storage involves storing and accessing past transaction data from old blocks.

Where does blockchain state fit into this?

Up until now it's been all about transaction data, but blockchain state is a related topic. The state is different from transaction data. Specifically, the state is like a current snapshot of the network, which includes information about account balances, smart contract balances, and validator set info.Problems that arise from the size of the state are different in nature than those around data availability and retrievability.

Why doesn't Celestia incentivize storage of historical data?

Most blockchains don't incentivize storage of data because it shouldn't be the responsibility of a blockchain to guarantee past data will be retrievable forever. In addition, the data storage problem only requires a single party to store and provide the data for users, which is not a strong problem. As such, Celestia's purpose is to provide a secure and scalable way to verify the availability of data. Once data has been verified as available, the job of storing and retrieving historical data is left up to other entities that require the data. Luckily, there are natural incentives for outside parties to store and serve historical data to users.

Who may store historical data if there is no reward?

There are multiple types of actors that may be likely to store historical data. Some of those include:

- Block explorers that provide access to past transaction data.
- Indexers that provide API queries for past data.
- Applications or rollups that require historical data for certain processes.
- Users that want to guarantee that they will have access to their transaction history.

What are some things blockchains can do to provide stronger assurances of data retrievability?

- Reward nodes based on the amount of transaction data they store and requests for data they serve (this is the case with some data storage blockchains, likeFilecoin
-).
- Publish transaction data onto a data storage blockchain that incentivizes storing and serving requests for historical data. [][Edit this page on GitHub] Last updated: Previous page Data retrievability and pruning Next page Overview of TIA []