

# Introduction to Arbitrum rollups with Celestia as DA

## Overview

The [integration of Celestia with Arbitrum Orbit](#) and the Nitro tech stack marks the first external contribution to the Arbitrum Orbit protocol layer, offering developers an additional option for selecting a data availability layer alongside Arbitrum AnyTrust. The integration allows developers to deploy an Arbitrum Rollup that uses Celestia for data availability and settles on Ethereum.

[Arbitrum Orbit](#) is a framework that enables the creation of customized, self-managed Arbitrum Rollup and AnyTrust chains. Key highlights of Arbitrum Orbit include:

1. Creation of custom chains
2. : Orbit allows the creation of dedicated chains that settle to Arbitrum's Layer 2 chains (Arbitrum One, Nova, Goerli, Sepolia), with customizable features like throughput, privacy, gas token, and governance.
3. Solving Ethereum's scalability
4. : Orbit addresses Ethereum's congestion and high demand for block space by enabling the creation of personal rollups, which offer scalable, secure alternatives to Ethereum's public chains.
5. Decentralized application development
6. : Orbit chains provide dedicated throughput, EVM+ compatibility, independent roadmaps, and reliable gas prices enhancing the development and operation of decentralized apps.
7. Benefits to the Ethereum ecosystem
8. : Orbit contributes to a multi-chain future for Ethereum, enhancing scalability, offering flexible security models, and enabling experimentation with execution environments and governance models.
9. Versatility and interoperability
10. : Orbit chains can be used for a range of purposes, from hosting a single dApp to an ecosystem of dApps, with the capability to communicate with other Orbit chains.

## Blobstream

### UPCOMING INTEGRATION

The Blobstream section is currently under development. Please note that the integration with Blobstream is not yet finalized, but it is planned for implementation in the near future. The Celestia and Arbitrum integration also [includes Blobstream](#), which relays commitments to Celestia's data root to an onchain light client on Ethereum. This allows L2 solutions that settle on Ethereum to benefit from the scalability Celestia's data availability layer can provide.

As part of this integration, Blobstream is being called from the `ArbitrumSequencerInbox.sol` contract.

In the `SequencerInbox.sol` contract, the `validateBatchData` modifier has been designed to authenticate that the data root is on Celestia when reading a batch of data. This is achieved by the [following code in SequencerInbox.sol by @Ferret-San](#):

Note that the data above is the bytes serialized version of this struct in Go:

```
go type
```

```
BlobPointer
```

```
struct { BlockHeight uint64 Start uint64 SharesLength uint64 Key uint64 NumLeaves uint64 TupleRootNonce uint64  
TxCommitment [ 32 ] byte DataRoot [ 32 ] byte SideNodes [][] 32 byte } type
```

```
BlobPointer
```

```
struct { BlockHeight uint64 Start uint64 SharesLength uint64 Key uint64 NumLeaves uint64 TupleRootNonce uint64  
TxCommitment [ 32 ] byte DataRoot [ 32 ] byte SideNodes [][] 32 byte }
```

## Ethereum fallback mechanism in Nitro

Another feature of this integration is the [Ethereum fallback mechanism](#), which enables Ethereum L2s (or L3s) to “fall back” to using Ethereum calldata for data availability in the event of downtime on Celestia Mainnet Beta.

In the case of Celestia downtime or temporary unavailability, L2s can fallback to posting transactions as calldata on Ethereum or another DA layer for data availability instead of posting to Celestia. This mechanism ensures users can continue to transact securely and seamlessly, preventing disruptions and helping to ensure user funds do not get stuck in the L2's bridge on Ethereum. This feature is available for the [Arbitrum Orbit integration](#).

By default in [Arbitrum Nitro](#), the [Ethereum fallback mechanism in the BatchPoster function](#) is handling the process of storing data, with a fallback mechanism to store data onchain if the primary data availability storage fails.

The [@celestiaorg/nitro](#) integration [uses the same fallback mechanism](#) .

The fallback logic for Celestia DA is configurable, providing an alternative to the previous default fallback mechanism. Additionally, an ability has been added to the Arbitrum node software which allows the sequencer to call `VerifyAttestation` to check if a data root has been posted on Blobstream or not, before it sends the sequencer message (data pointer) to the underlying chain.

## Next steps

In the next page [learn how to deploy an Arbitrum rollup devnet using Celestia as DA](#). [\[ \[ Edit this page on GitHub \]](#) Last updated: [Previous page Ethereum fallback mechanism](#) [Next page Deploy an Arbitrum rollup devnet](#) [\[](#)