Blobstream: Streaming modular DA to Ethereum

What is Blobstream?

Blobstream is the first data availability solution for Ethereum that securely scales with the number of users. Formerly known as theQuantum Gravity Bridge (QGB) , Blobstream relays commitments to Celestia's data root to an onchain light client on Ethereum, for integration by developers into L2 contracts. This enables Ethereum developers to build high-throughput L2s using Celestia's optimised DA layer, the first with Data Availability Sampling (DAS). Any ecosystem can deploy a Blobstream light client onchain to allow L2s and L3s to access DA from Celestia.

An implementation of Blobstream, bySuccinct , calledBlobstream X , is out and will be used in the upcoming deployments. This implementation proves the validity of Celestia block headers on a target EVM chain using zero-knowledge (ZK) proofs, which allow inheriting all the security guarantees of Celestia.

The latest implementation of Blobstream X isSP1 Blobstream , which is written in Rust for the SP1 zkVM. SP1 Blobstream offers improved performance and efficiency while maintaining the security guarantees of the original Blobstream X.

Please note: Blobstream remains early-stage, experimental software and users should use Blobstream at their own risk.

Implementations of Blobstream

- SP1 Blobstream
- (an implementation of Blobstream X)
- Blobstream X

Blobstream vs. data availability committees (DACs)

Decentralization and security

Blobstream is built on Celestia, which uses a CometBFT-based proof-of-stake system. Blobstream shares the same security assumptions as Celestia. In contrast, data availability committees (DACs), are typically centralized or semi-centralized, relying on a specific set of entities or individuals to vouch for data availability.

Mechanism of verification

Blobstream uses data availability attestations, which are Merkle roots of the batched L2 data, to confirm that the necessary data is present on Celestia. The L2 contract on Ethereum can check directly with Blobstream if the data is published on Celestia. Similarly, a DAC would rely on attestations or confirmations from its permissioned members.

Flexibility and scalability

Blobstream is designed to offer high-throughput data availability for Ethereum L2s, aiming to strike a balance between scalability and security. It operates independently of Ethereum's gas costs, as Celestia's resource pricing is more byte-focused rather than computation-centric. On the other hand, the scalability and flexibility of a DAC would depend on its specific design and implementation.

In summary, both Blobstream and DACs aim to ensure offchain data availability, but Blobstream offers a more decentralized, secure, and scalable solution compared to the potential centralized nature of DACs.

What is SP1 Blobstream?

SP1 Blobstream is the latest implementation of Blobstream in Rust using theSP1 zkVM.

SP1 Blobstream is the latest implementation of Blobstream with a ZK light client that bridges Celestia’s modular DA layer to Ethereum to allow high-throughput rollups to use Celestia’s DA while settling on Ethereum.

Optimistic or ZK rollups that settle on Ethereum, but wish to use Celestia for DA, require a mechanism forbridging Celestia’s data root to Ethereum as part of the settlement process. This data root is used during inclusion proofs to prove that particular rollup transactions were included and made available in the Celestia network.

Bridging Celestia’s data root to Ethereum requires running a Celestialight client as a smart contract on Ethereum, to make the latest state of the Celestia chain known on Ethereum and available to rollups. SP1 Blobstream uses the latest advances in ZK proofs to generate asuccinct proof that enough Celestia validators have come to consensus (according to the CometBFT consensus protocol) on a block header, and verifies this proof in the SP1 Blobstream Ethereum smart contract to update it with the latest Celestia header.

The SP1 Blobstream ZK proof not only verifies the consensus of Celestia validators, but it also merkelizes and hashes all

the data roots in the block range from the previous update to the current update, making accessible all Celestia data roots (verifiable with a Merkle inclusion proof against the stored Merkle root) to rollups.

If you're looking to deploy SP1 blobstream to a new chain, seenew Sp1 Blobstream deployments .

Learn more at thesp1-blobstream repo.

NOTE

The current Blobstream deployments all use SP1 Blobstream.

Integrate with SP1 Blobstream

The following docs go over how developers can integrate SP1 Blobstream.

You canfind the repository for SP1 Blobstream along with code for:

- The SP1 Blobstream smart contract -SP1Blobstream.sol
- The SP1 program
- The SP1 Blobstream contract Golang bindings

The first deployments of SP1 Blobstream will be maintained on the following chains: Arbitrum One, Base and Ethereum Mainnet. Every 1 hour, the prover/relayer will post an update to the Blobstream contract that will include a new data commitment range that covers a 1-hour block range from thelatestBlock in the contract. On Ethereum Mainnet, the contract will be updated every 4 hours.

How to integrate with Blobstream

Integrating your L2 with Blobstream requires two components: youronchain smart contract logic , and youroffchain client logic for your rollup . The next three sections cover these topics:

- Integrate with Blobstream contracts
- Integrate with Blobstream client
- Querying the Blobstream proofs

Blobstream rollups

More on the different ways to build a blobstream rollup can be found in theblobstream rollups documentation.

Deployed contracts

You can interact with the SP1 Blobstream contracts today. The SP1 Blobstream Solidity smart contracts are currently deployed on the following chains:

Contract EVM network Contract address Attested data on Celestia Link to Celenium SP1 Blobstream Ethereum Mainnet 0x7Cf3876F681Dbb6EdA8f6FfC45D66B996Df08fAe Mainnet Beta Deployment on Celenium SP1 Blobstream Arbitrum One 0xA83ca7775Bc2889825BcDeDfFa5b758cf69e8794 Mainnet Beta Deployment on Celenium SP1 Blobstream Base 0xA83ca7775Bc2889825BcDeDfFa5b758cf69e8794 Mainnet Beta Deployment on Celenium SP1 Blobstream Sepolia 0xf0c6429ebab2e7dc6e05dafb61128be21f13cb1e Mocha testnet Deployment on Celenium SP1 Blobstream Arbitrum Sepolia 0xc3e209eb245Fd59c8586777b499d6A665DF3ABD2 Mocha testnet Deployment on Celenium SP1 Blobstream Base Sepolia 0xc3e209eb245Fd59c8586777b499d6A665DF3ABD2 Mocha testnet Deployment on Celenium [][Edit this page on GitHub] Last updated: Previous page Transaction resubmission guidelines Next page Integrate with Blobstream contracts []