Build a Rollup With Avail

Overview

Integrating with Avail enhances transaction processing by keeping data off-chain while ensuring its availability and validity. Avail's role as an optimized blockchain for data availability is central to this adaptation, offering a robust and modular design for diverse use cases.

If you are looking to build with Avail at a hackathon, you can check out a curated list of ideas on thisNotion doc(opens in a new tab) .

System Operations

- Transaction Processing and Sequencing
- : In the rollup framework, transactions are processed, sequenced, and readied for submission to Avail.
- Data Submission to Avail
- : This processed data is securely transferred to Avail, following a specific protocol designed for efficient and secure data handling.
- Configuration and Connection
- : Rollup systems are configured for seamless integration with Avail, ensuring smooth data flow and interaction.
- Smart Contract Interaction
- : Users engage with on-chain contracts, providing Merkle proofs for actions like withdrawals. These contracts interact with Avail to authenticate and process these transactions.

Attestation Bridge

AVAILABLE ON TESTNET Avail's Attestation Bridge is available on testnet for Ethereum. * Streamlined Verification with Attestation Bridge * : Avail's Attestation Bridge simplifies the verification process on L1. This bridge facilitates the direct posting of data availability attestations to the L1 blockchain, thereby easing the workload of the verification contract. * Role of the Verification Contract * : With the Attestation Bridge in place, the verification contract's primary role is to check these on-chain attestations, ensuring data availability and integrity.

Interaction with the L1

- Verification Contract Functionality
- : Situated on the L1, this contract plays a dual role—it verifies transaction accuracy and checks data availability, utilizing Avail's attestations.
- L1 Contract Dynamics
- : Rollups maintain a communicative relationship with L1 via dedicated contracts. The main attestation contract stores state commitments (Merkle data roots) from block producers. Parallelly, a verification contract handles state transition validity checks.

Security and Finalization

- Validator Consensus
- : Avail's validators, part of a Nominated Proof-of-Stake system, reach consensus on the transaction batches.
- GRANDPA Finality Gadget
- : The consensus is solidified using the GRANDPA finality gadget, guaranteeing the availability of data.
- Data Root Publication
- : Sequencers publish the data root on the L1, linking Avail's data availability with L1's security.

Avail Light Clients and Data Verification

- Independent Data Verification
- : Avail's light clients enable verification of data availability without relying on the majority of nodes.
- Data Sampling
- : Light clients can sample from the blocks on Avail's blockchain using an AppId to validate data availability.

Validium Architecture with Avail

In the Validium model, transactions are collected by a sequencer, which batches them together. The sequencer's role extends beyond transaction ordering—it prepares a Merkle tree of transactions, where each leaf represents a transaction or a set of transactions. The root of this Merkle tree, known as the batch hash, is crucial for ensuring the integrity of the transaction batch and for constructing inclusion proofs.

Avail comes into play as the recipient of this transaction data. Upon receiving the batched transactions, Avail executes erasure coding.

An inclusion proof is a Merkle proof generated by Avail to attest that a specific transaction is part of the batch and has been recorded on Avail's blockchain. It's essential for verifying the presence of transactions without downloading the entire dataset.

Simultaneously, the state of the system is computed by executing these transactions. A Prover, which is a computational entity, takes the state transitions and generates cryptographic proofs—such as zk-SNARKs or STARKs—to attest to the validity of state changes without revealing the underlying data.

A Sequence Sender is responsible for communicating with L1. It takes the inclusion proofs and batch hashes from Avail, along with validity proofs from the Prover, and submits them to the L1 chain. This submission process usually involves smart contract calls that log the transaction data's hash and the validity proofs onto the L1 blockchain.

The L1 acts as the main layer for dispute resolution and finality. While it doesn't store the transaction data itself, it retains the cryptographic commitments to the data. This ensures that if there's ever a dispute or need for verification, the proofs can be checked against the commitments to ascertain the validity of state transitions and the availability of data.

This setup leverages L1's security while offloading the data-intensive work to Avail, which is optimized for handling vast amounts of data efficiently and securely. By doing so, Validium chains can significantly reduce their costs and improve scalability, all the while maintaining a high level of trust and security.

Optimium Architecture with Avail

In the Optimium model, transactions are similarly aggregated by a sequencer. This sequencer organizes transactions into batches and computes a data root, a Merkle tree root representing the batch, crucial for integrity and proof of inclusion.

Avail is integrated as a data availability layer. Once the sequencer sends transaction batches to Avail, it employs erasure coding to ensure data redundancy and integrity. Avail then generates KZG polynomial commitments and the data root, essential for confirming data availability.

The next phase involves state computation of the system, executed on the rollup, depending on the chain's architecture. Avail's data availability solution ensures that the transaction data is readily accessible for any necessary computation or verification.

A Sequence Sender, in this architecture, is responsible for submitting proofs to the main chain. These include the data root from Avail, ensuring that the data availability is anchored to the security of Ethereum or the corresponding L2.

This architecture provides the dual benefits of the main chain's security for settlement and dispute resolution, and Avail's efficiency in handling data. By offloading data availability to Avail, Optimium chains can achieve higher scalability and efficiency while maintaining robust security and decentralization.

DA Solution Suites

Avail Uncharted is a core initiative within the Avail ecosystem dedicated to exploring uncharted territories in modular blockchain technology. Driven by the core Avail team, the mission is twofold: to nurture innovative projects and to cultivate a close-knit collaboration with the community.

Project Description Repository Avail-Powered Optimistic EVM Rollup A sovereign EVM-compatible optimistic rollup construction. op-evm (opens in a new tab) DA Adapter for Sovereign SDK An adapter enabling modular sovereign rollups using the Sovereign Rollup SDK. sovereign-da-adapter (opens in a new tab) DA Interface for Madara Starknet A unified DA interface allowing the Madara Starknet Sequencer to publish data onto Avail. madara-da-interface (opens in a new tab) Avail-Powered zkEVM-Based Validium A Validium based on the Polygon zkEVM stack that uses Avail instead of the native DAC for data availability. -validium-node (opens in a new tab)

-validium-contracts (opens in a new tab) DA Adapter for Optimism SDK An adapter facilitating Avail DA's integration with Optimism's Rollup SDK op-stack. avail-op-stack-adapter (opens in a new tab) DA Adapter for Rollkit An adapter designed for Rollkit's modular rollup framework that enables ABCI-compatible solutions. rollkit-da-adapter (opens in a new tab)

Build with Avail Quickstart