Chain Architecture

This page contains information about the components of the rollup protocol and how they work together to build the layer 2 blockchain from the Chain Operator's perspective. The OP Stack is built in such a way that it is as similar to Ethereum as possible. Like Ethereum, the OP Stack has execution and consensus clients. The OP Stack also has some privileged roles that produce L2 blocks. If you want a more detailed view of the OP Stack protocol, check out theOP Stack section of our documentation.

Permissioned Components

These clients and services work together to enable the block production on the L2 network. Sequencer nodes (op-geth +op-node) gather proposed transactions from users. The batcher submits batch data to L1 which controls the safe blocks and ultimately controls the canonical chain. The proposer submits output roots to L1 which control L2 to L1 messaging.

op-geth

op-geth implements the layer 2 execution layer, withminimal changes(opens in a new tab) for a secure Ethereum-equivalent application environment.

op-node

op-node implements most rollup-specific functionality as the layer 2 consensus layer, similar to a layer 1 beacon-node. Theop-node is stateless and gets its view of the world fromop-geth .

op-batcher

op-batcher is the service that submits the L2 Sequencer data to L1, to make it available for verifiers. To reduce the cost of writing to the L1, it only posts the minimal amount of data required to reproduce L2 blocks.

op-proposer

op-proposer is the service that submits the output roots to the L1. This is to enable trustless execution of L2-to-L1 messaging and creates the view into the L2 state from the L1's perspective.

Ingress Traffic

It is important to setup a robust chain architecture to handle large volumes of RPC requests from your users. The Sequencer node has the important job of working with the batcher to handle block creation. To allow the Sequencer to focus on that job, you can peer replica nodes to handle the rest of the work.

An example of this would be to configureproxyd(opens in a new tab) to route RPC methods, retry failed requests, load balance, etc. Users sendingeth_sendRawTransaction requests can have their requests forwarded directly to the Sequencer. All other RPC requests can be forwarded to replica nodes.

proxyd

This tool is a RPC request router and proxy. It does the following things:

1. Whitelists RPC methods.
2. Routes RPC methods to groups of backend services.
3. Automatically retries failed backend requests.
4. Track backend consensus (latest, safe, finalized blocks), peer count and sync state.
5. Re-write requests and responses to enforce consensus.
6. Load balance requests across backend services.
7. Cache immutable responses from backends.
8. Provides metrics to measure request latency, error rates, and the like.

Sequencer

The Sequencer node works with the batcher and proposer to create new blocks. So it should handle the state changing RPC requesteth_sendRawTransaction . It can be peered with replica nodes to gossip newunsafe blocks to the rest of the network.

Replica node

The replica nodes are additional nodes on the network. They can be peered to the Sequencer, which might not be

connected to the rest of the internet, and other replicas. Additional replicas can help horizontally scale RPC requests.

Next Steps

- Find out how you can support snap sync
- on your chain.
- Find out how you can utilize blob space
- to reduce the transaction fee cost on your chain.

Overview Creating Your Own L2 Rollup Testnet