Rollup protocol overview

The big idea that makes Optimism possible is the Optimistic Rollup. We'll go through a brief explainer ofhow Optimistic Rollups work at a high level. Then we'll explainwhy Optimism is built as an Optimistic Rollup and why we believe it's the best option for a system that addresses all of our design goals. Check out theprotocol specs(opens in a new tab) , if you want to more detail about the rollup protocol.

Optimistic rollups TL;DR

Optimism is an "Optimistic Rollup," which is basically just a fancy way of describing a blockchain that piggy-backs off of the security of another "parent" blockchain. Specifically, Optimistic Rollups leverage the consensus mechanism (like PoW or PoS) of their parent chain instead of providing their own. In OP Mainnet's case, this parent blockchain is Ethereum.

Block storage

In Bedrock, L2 blocks are saved to the Ethereum blockchain using a non-contract address (0xff00..0010 on Ethereum(opens in a new tab)) to minimize the L1 gas expense. As these blocks are submitted as transactions using EIP-4844blobs , there is no way to modify or censor them after the "transaction" is included in a block that has enough attestations. This is the way that OP Mainnet inherits the availability and integrity guarantees of Ethereum.

Blocks are written to L1 ina compressed format(opens in a new tab)to reduce costs. This is important because writing to L1 isthe major cost of OP Mainnet transactions.

Block production

Optimism block production is primarily managed by a single party, called the "sequencer," which helps the network by providing the following services:

- Providing transaction confirmations and state updates.
- Constructing and executing L2 blocks.
- Submitting user transactions to L1.

In Bedrock, the sequencer does have a mempool, similar to L1 Ethereum, but the mempool is private to avoid opening opportunities for MEV. In OP Mainnet, blocks are produced every two seconds, regardless of whether they are empty (no transactions), filled up to the block gas limit with transactions, or anything in between.

Transactions get to the sequencer in two ways:

1. Transactions submitted directly to the sequencer.
2. These transactions are a lot cheaper to submit, as they do not require the expense of a separate L1 transaction. However, they cannot be made censorship resistant since the sequencer is the only participant that knows about them.
3. Transactions submitted on L1 (calleddeposits
4.) are included in the chain in the appropriate L2 block.
5. Every L2 block is identified by the "epoch" (the L1 block to which it corresponds, which typically has happened a few minutes before the L2 block) and its serial number within that epoch.
6. The first block of the epoch includes all the deposits that happened in the L1 block to which it corresponds.
7. If the sequencer attempts to ignore a legitimate L1 transaction, it ends up with a state that is inconsistent with the verifiers, same as if the sequencer tried to fake the state by other means.
8. This provides OP Mainnet with L1 Ethereum level censorship resistance.
9. You can read more about this mechanismin the protocol specifications(opens in a new tab)
10. .

For the moment,The Optimism Foundation(opens in a new tab)runs the only block producer on OP Mainnet.

Block execution

The execution engine (implemented as theop-geth component) receive blocks using two mechanisms:

1. The execution engine can update itself using peer to peer network with other execution engines.
2. This operates the same way that the L1 execution clients synchronize the state across the network.
3. You can read more about itin the specs(opens in a new tab)
4. .
5. The rollup node (implemented as theop-node
6. component) derives the L2 blocks from L1.
7. This mechanism is slower, but censorship resistant.
8. You can read more about itin the specs(opens in a new tab)

9. .

Bridging ETH or tokens between layers

Optimism is designed so that users can send arbitrary messages between smart contracts on L2 (OP Mainnet, OP Sepolia, etc.) and the underlying L1 (Ethereum mainnet, Sepolia, etc.). This makes it possible to transfer ETH or tokens, including ERC20 tokens, between the two networks. The exact mechanism by which this communication occurs differs depending on the direction in which messages are being sent.

OP Mainnet uses this functionality in the Standard bridge to allow users to deposit tokens from Ethereum to OP Mainnet and also allow withdrawals of the same tokens from OP Mainnet back to Ethereum. See thedeveloper documentation and examples for details on the inner workings of the Standard bridge.

Moving from Ethereum to OP Mainnet

In Optimism terminology, transactions going from Ethereum (L1) to OP Mainnet (L2) are calleddeposits .

You useL1CrossDomainMessenger (opens in a new tab) orL1StandardBridge (opens in a new tab) . Deposit transactions become part of the canonical blockchain in the first L2 block of the "epoch" corresponding to the L1 block where the deposits were made. This L2 block will usually be created a few minutes after the corresponding L1 block. You can read more about thisin the specs(opens in a new tab).

Moving from OP Mainnet to Ethereum

Withdrawals (the term is used for any OP Mainnet to Ethereum message) have three stages:

1. You initialize withdrawals with an L2 transaction.
2. Wait for the next output root to be submitted to L1 (you can see this onthe SDK(opens in a new tab)
3.) and then submit the withdrawal proof usingproveWithdrawalTransaction
4. .
5. This new step enables offchain monitoring of the withdrawals, which makes it easier to identify incorrect withdrawals or output roots.
6. This protects OP Mainnet users against a whole class of potential bridge vulnerabilities.
7. After the fault challenge period ends (a week on mainnet, less than that on the test network), finalize the withdrawal.

You can read the full withdrawal specifications here(opens in a new tab)

Fault proofs

In an Optimistic Rollup, state commitments are published to L1 (Ethereum in the case of OP Mainnet) without any direct proof of the validity of these commitments. Instead, these commitments are considered pending for a period of time (called the "challenge window"). If a proposed state commitment goes unchallenged for the duration of the challenge window (currently set to 7 days), then it is considered final. Once a commitment is considered final, smart contracts on Ethereum can safely accept withdrawal proofs about the state of OP Mainnet based on that commitment.

When a state commitment is challenged, it can be invalidated through a "fault proof" (formerly known as a "fraud proof" (opens in a new tab)) process. If the commitment is successfully challenged, then it is removed from theStateCommitmentChain to eventually be replaced by another proposed commitment. It's important to note that a successful challenge does not roll back OP Mainnet itself, only the published commitments about the state of the chain. The ordering of transactions and the state of OP Mainnet is unchanged by a fault proof challenge.

The fault proof process is currently undergoing major redevelopment as a side-effect of the November 11th, 2021EVM Equivalence(opens in a new tab) update.

Next steps

- If you want to learn more about rollup protocol, check out the guides ondeposit flow
- ,withdrawal flow
- , ortransaction flow
- .
- To learn about operating your own L2 rollup, see the guide onstarting a self-hosted chain
- or go directly to the tutorial oncreating your own L2 rollup
- .

Rollup Derivation pipeline