Cross-Domain Overview

This overview provides a detailed walkthrough of the lifecycle of cross-chain transactions, covering deposits, withdrawals, and transaction flows between L1 and L2. The diagram below illustrates the main components and steps involved.

Figure 1: The Lifecycle of an OP Stack Crosschain Transaction

Cross-domain communication in the OP Stack involves moving assets and messages between L1 and L2. Key components, such as bridges, messengers, and portals, ensure these transactions are executed securely and transparently. This page breaks down the lifecycle of a cross-chain transaction into three main flows: Deposit Flow, Transaction Flow (Tx Flow), and Withdrawal Flow.

Deposit Flow

Depositing assets from L1 to L2 follows a structured process that ensures funds are securely transferred and credited to the user's L2 account. The steps are as follows:

1. User Initiation (L1 Standard Bridge):
2. The user initiates a deposit by sending assets (e.g., ETH or tokens) to the L1 Standard Bridge contract. This contract receives the assets and prepares a message to relay to L2.
3. Message Relaying (L1 CrossDomain Messenger):
4. The L1 Standard Bridge sends a message to the L1 CrossDomain Messenger, which is responsible for handling inter-layer communication. The messenger emits aTransactionDeposited
5. event to signal the start of the deposit process.
6. Processing on L2 (OptimismPortal):
7. The message is received by the OptimismPortal on L2. Here, the deposited assets are held securely until the transaction is finalized. The portal initiates the deposit transaction, updating the user's balance on L2.
8. Finalization (L2 CrossDomain Messenger):
9. The L2 CrossDomain Messenger processes the deposit by updating the user's account balance, and making the assets available for use on L2.

Transaction (Tx) Flow

The transaction flow covers the steps involved in cross-domain message passing and state updates between L1 and L2:

1. Message Queuing (L2ToL1MessagePasser):
2. During cross-layer communication, certain messages are queued for processing. TheL2ToL1MessagePasser
3. prepares these messages for state updates or withdrawals, ensuring they are available for proving and relaying.
4. State Reading and Proving (L2OutputOracle):
5. TheL2OutputOracle
6. plays a critical role in validating state changes between L1 and L2. It reads the current state and submits a proposal usingproposeL2Output()
7. . This proposal includes information about queued messages or state changes that need to be relayed to L1.
8. Message Relay (CrossDomain Messengers):
9. Messages are relayed between the L1 and L2 CrossDomain Messengers as part of transaction execution. This includes updating state or finalizing transactions on the target layer.

Withdrawal Flow

Withdrawing assets from L2 back to L1 involves a multi-step process to ensure the transaction is validated and executed correctly:

1. User Initiation (L2 Standard Bridge):
2. The withdrawal process starts when the user calls thewithdraw()
3. function on the L2 Standard Bridge, specifying the amount and asset to be withdrawn.
4. Message Relay (L2 CrossDomain Messenger):
5. The L2 CrossDomain Messenger receives the withdrawal request and relays the message for processing. It may involve queuing the message in theL2ToL1MessagePasser
6. for further steps.
7. Proving and Finalization (L2OutputOracle):
8. The withdrawal message is proven using theL2OutputOracle
9. , which submits a state output proving that the withdrawal is legitimate. This step involves reading the state and generating the required proofs during the proving time.
10. Finalization (L1 CrossDomain Messenger and L1 Standard Bridge):
11. Once the withdrawal is proven, the message is finalized by the L1 CrossDomain Messenger. The L1 Standard Bridge completes the process by releasing the withdrawn assets to the user's L1 account.

Forced transaction Transactions