Architecture

The Connext protocol is composed of a set of smart contracts and offchain agents.

Smart Contracts

Connext's smart contracts are the interfaces between the protocol and the wide gamut of different users in the ecosystem. There are contracts for handlingxcall , managing asset registration, and provisioning liquidity for routers and stableswap LPs.

The full contract stack consists of the following components.

Connext

Dispatches and handles messages related to sending funds across chains. Custodies funds for canonical assets, fast liquidity, and stable swaps.

The Connext contract uses theDiamond pattern so it comprises a set of Facets that act as logical boundaries for groups of functions. Facets share contract storage and can be upgraded separately.

Diamond Facets

TokenFacet

Manages asset enrollment, stores mappings of adopted <-> local assets, exposes liquidity caps functions, and specifies stableswaps for assets.

BridgeFacet

Implementsxcall and enables destination-side calldata execution.

InboxFacet

Holds all the functionality needed for Connext's messaging layer to reconcile cross-chain transfers.

ProposedOwnableFacet

Provides a basic access control mechanism.

RelayerFacet

Manages whitelisting of relayers.

RoutersFacet

Manages whitelisting of routers and keeps track of router owners/recipients.

StableSwapFacet

A StableSwap implementation that custodies closely pegged assets (eg. group of stablecoins).

SwapAdminFacet

Manages only-admin controls for the StableSwapFacet.

DiamondCutFacet

Functions for adding, removing, and replacing facets.

DiamondLoupeFacet

Required by the Diamond standard. Implements the DiamondLoupe interface which allows for inspection of a Diamond contract's various facets and their functions.

Messaging

The various contracts required to manage merkle roots containing hashed transfer data and send them through a hub-and-spoke architecture. The messaging architecture includes:

- Connector.
- A connector is an abstraction around an underlying transport layer. TheIConnector
- interface requires aprocessMessage
- method implemented for handling incoming messages.Connector
- is an abstract contract that is inherited by the following contracts:
-
    - SpokeConnector.
-
    - TheSpokeConnector
-
    - is deployed on spoke domains and implements asend
-
    - method to send the Merkle root of all the messages that originate from the spoke domain to the hub domain. For example,ArbitrumSpokeConnector
-
    - is deployed on the Arbitrum L2.
-
    - HubConnector.
-
    - TheHubConnector
-
    - is deployed on hub domains for each spoke and implements asendMessage
-
    - method to send the aggregated Merkle root of all the received spoke Merkle roots to the configured destination domain. For exampleArbitrumHubConnector
-
    - is deployed on Ethereum L1.
- *
-

Each AMB implementation requires us to create and deployHubConnector andSpokeConnector contracts for that flavor of AMB, calling into the internal bridge infrastructure.

Offchain Agents

Routers

Routers are liquidity providers that enable instant liquidity for the user on the destination chain in return for a fee. Anybody can participate in the protocol as a router and there is no minimum liquidity required! Routers provide a crucial service to the Connext protocol.

Learn how to run one in theRouters section.

Sequencer

The sequencer collects bids from all chains and randomly selects router(s) to fulfill them. Any number of routers can fulfill a single transaction, which is especially useful for large transfers. The sequencer will post batches of these bids to a relayer network to submit them to chain.

Relayers

Relayers are a decentralized network of infrastructure operators that can execute smart contract transactions on behalf of a user in exchange for a small fee. Because the last leg of a cross-chain transaction requires execution on the destination domain, relayers play an important role in completing the full flow.

We are currently usingGelato as our relayer service.

Previous How It Works Next Transaction Lifecycle Last updated9 months ago On this page *Smart Contracts * Connext * Messaging * Offchain Agents * Routers * Sequencer * Relayers

Edit on GitHub