

The Reservoir Relay

Reservoir's production grade relaying service

The Reservoir relay is a production-grade relayer currently supporting Relay. Reservoir acts as a trusted party that performs cross-chain relays and order validation.

How does it work?

When a user wishes to bridge, they submit an order and a transfer to the Reservoir relayer, who validates the order and sends the funds to the user on the destination chain. Reservoir maintain a balance on all chains in order to make executions, and then periodically rebalances to ensure there is sufficient liquidity on all chains. In the rare case of a transaction failure, Reservoir refunds the user directly as soon as possible.

While this model requires trust in Reservoir as a service provider, it reduces the costs of executing cross-chain actions to an absolute minimum. Since each cross chain action is simply an origin chain transfer and a destination chain transfer (42000 gas total), it is substantively cheaper than bridging on other bridges.

Why start with a trusted system?

The current state of Relay, which requires trust in Reservoir's relayer, is an important part of the protocol development process. It is not intended as a final state of the system, but as a way to introduce the core concepts, and learn from users, developers, and relayers about what is most important in the future decentralized system. If you are interested in learning more about where Relay is headed, you can read the protocol overview [here](#).

Technical overview

Here we walk through some technical details about how the Reservoir relayer functions.

Getting a quote

When a user wishes to Relay, the application receives an order (quote) from the [Relay SDK/API](#). The quote returns both a cost for execution, and an expected time. The quote is a total value required by the user for the action to proceed. The quote encompasses the execution value, the estimated destination chain gas fee, and the relayer fee. This quote covers the gas cost on the destination chain. The transfer call data for order acceptance is also included in the returned data.

Onchain order attribution

When the user accepts the order, she submits a transfer to the solver. An orderId associated with the quote is appended in the tx calldata. This orderId will be used on the destination chain to link fills across chains.

Order validation and filling

Upon validation of the inbound transfer transaction, the Reservoir relayer will fill the order on the destination chain. Included in the fill transaction is the orderId associated with the the quote. Reservoir maps the origin and destination chain transactions together by orderId to ensure the transaction was filled appropriately.

Rebalancing

The Reservoir relayer uses a robust rebalancing algorithm to make sure funds are as available as possible for cross-chain executions. The user is not responsible for any rebalancing activity. Should the Reservoir relayer become "off-balance", the capacity for instant relaying may be limited.

Fallbacks (coming soon)

Since transfers are sent directly to the Reservoir relay by users. The relay may opt to use alternative methods of cross-chain bridging in the case of low instant capacity. Under these conditions, the relay will quote the user a longer timeframe within which to expect the execution to occur. For example, should a user desire a withdrawal from an L2 onto Ethereum, the relay may opt to use a canonical bridge to bridge the user's funds. The relay attempts to provide the best possible quote (in price and in time) to the user, who may choose to accept the proposed quote or not.

Refunds

If for some reason the quoted execution is no longer available by the time of relay submission, the user may be refunded. Refunds are as instant as possible and are refunded on the destination chain where possible, and the origin chain otherwise.

Partial fills

If the user submits the transfer event after quote expiration, the relay may not be able to fill the order at the quoted price if gas prices on the destination chain have increased substantively. In this case, the relay will attempt to fill the order at the best possible price, and give the rest of the bridged value to the user.

Disputes

Due to the trusted nature of the current Relay design, disputes (i.e. cases where a user did not receive the expected outcome and was not refunded) should be brought directly to the Reservoir support team. [Relay Protocol Supported Chains](#) [twitter](#) [Powered by Mintlify](#)

- [How does it work?](#)
- [Why start with a trusted system?](#)
- [Technical overview](#)
- [Getting a quote](#)
- [Onchain order attribution](#)
- [Order validation and filling](#)
- [Rebalancing](#)
- [Fallbacks \(coming soon\)](#)
- [Refunds](#)
- [Partial fills](#)
- [Disputes](#)