Introduction

Reactor Testnet Bridge: https://reactor.testnet.apexfusion.org

The document explains the technical architecture and operation of Reactor Bridge within the Apex ecosystem, comprising the Prime (L1) chain and the Vector and Nexus (L2) chains. The Vector chain utilizes the UTXO model, while the Nexus chain is based on the EVM. Bridging is achieved by locking tokens on the source blockchain and releasing an equivalent amount on a destination blockchain. Locked APEX tokens are held at a multisig address or a smart contract controlled by bridge validators. Figure 1 illustrates the chain interactions with the bridge as a mediator.

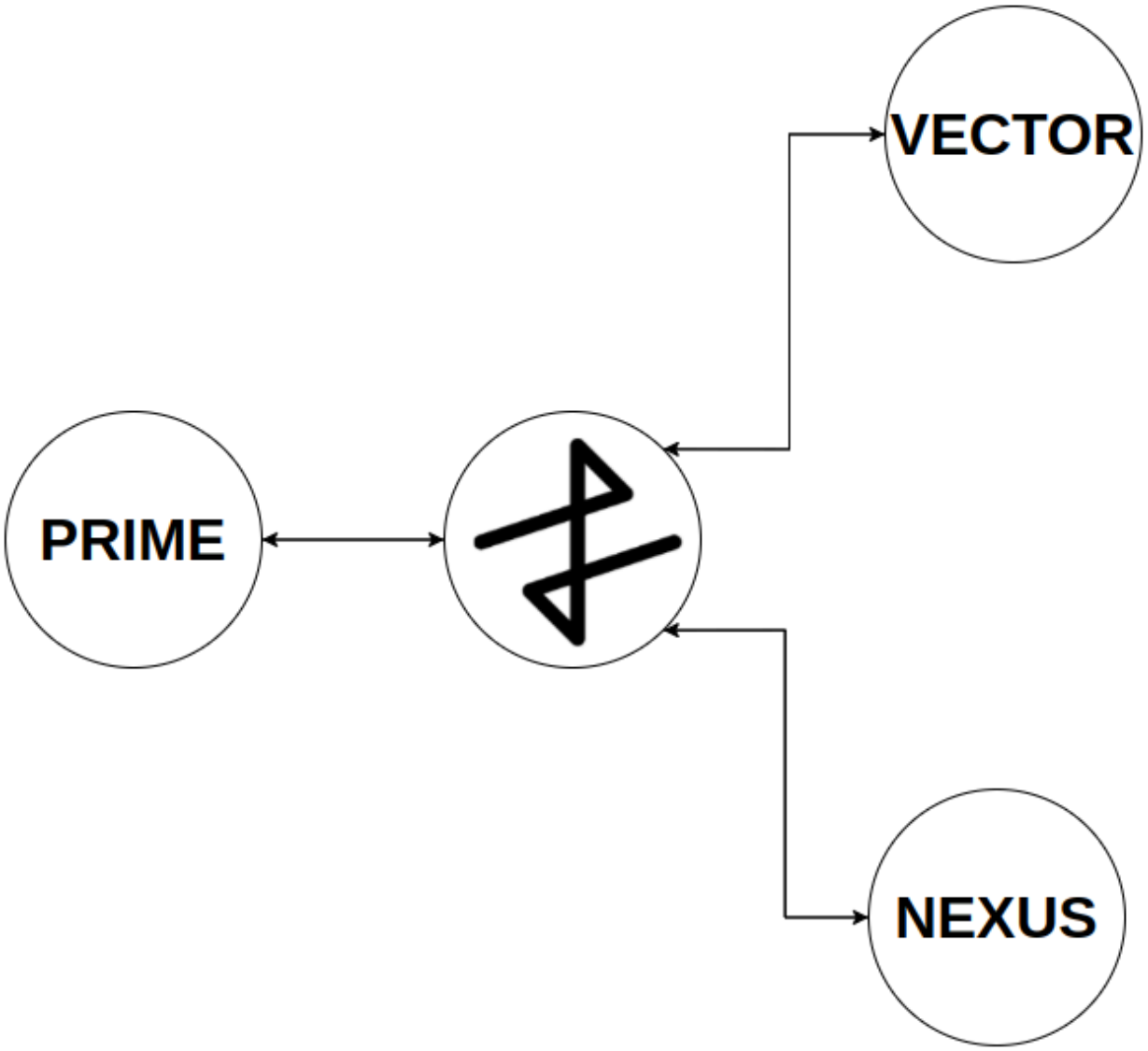

Figure 1: High-level Overview

All blockchains in the Apex ecosystem use the APEX token minted exclusively on the Prime chain. Tokens are transferred between L1 and L2 chains via the trusted two-way Reactor Bridge. Thus, the locked tokens on the L1 chain always equal the released tokens on the L2 chains.

The Bridge Architecture

Reactor functions as a bridge blockchain, where validators collect, verify, and act on environmental events. Figure 2 illustrates the main components of the Reactor Bridge and their interactions during chain transmissions.

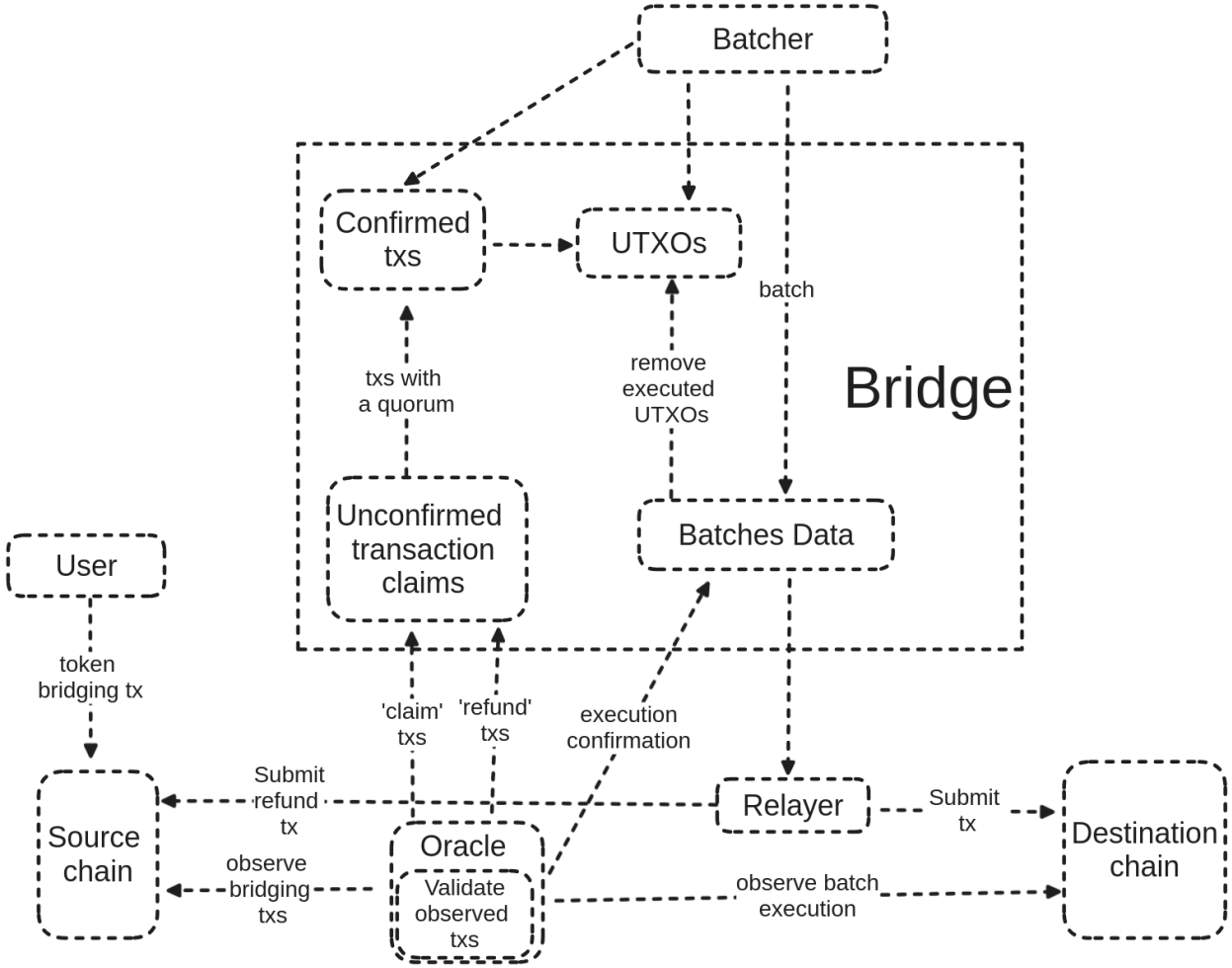

Figure 2: Reactor Components

The main components:

1. **Oracle** is an off-chain component run by each validator of the bridge blockchain. It monitors key events crucial to bridge operations, such as bridging requests, batch executions, and refund executions. These events are processed to facilitate accurate bridging decisions.

2. **Bridge Smart Contracts** are designed to implement on-chain bridging logic. The processing of events sbmitted by Oracles takes place whithin.

3. **Batcher** is another trusted off-chain of a validator. It monitors the bridge blockchain to determine when it's time to initiate batch creation. It then generates a batch instance representing the transaction that will be submitted to a destination blockchain.

4. **Relayer** submits confirmed batches to the destination chain. It is a dedicated, trustless component operated as a standalone instance that can be run by any party. In essence, the Relayer retrieves batches from the bridge blockchain and submits the transactions to the destination chain.

The Bridge Workflow

Based on the components described in the previous section, a deposit/withdrawal workflow includes the following steps (Figure 3):

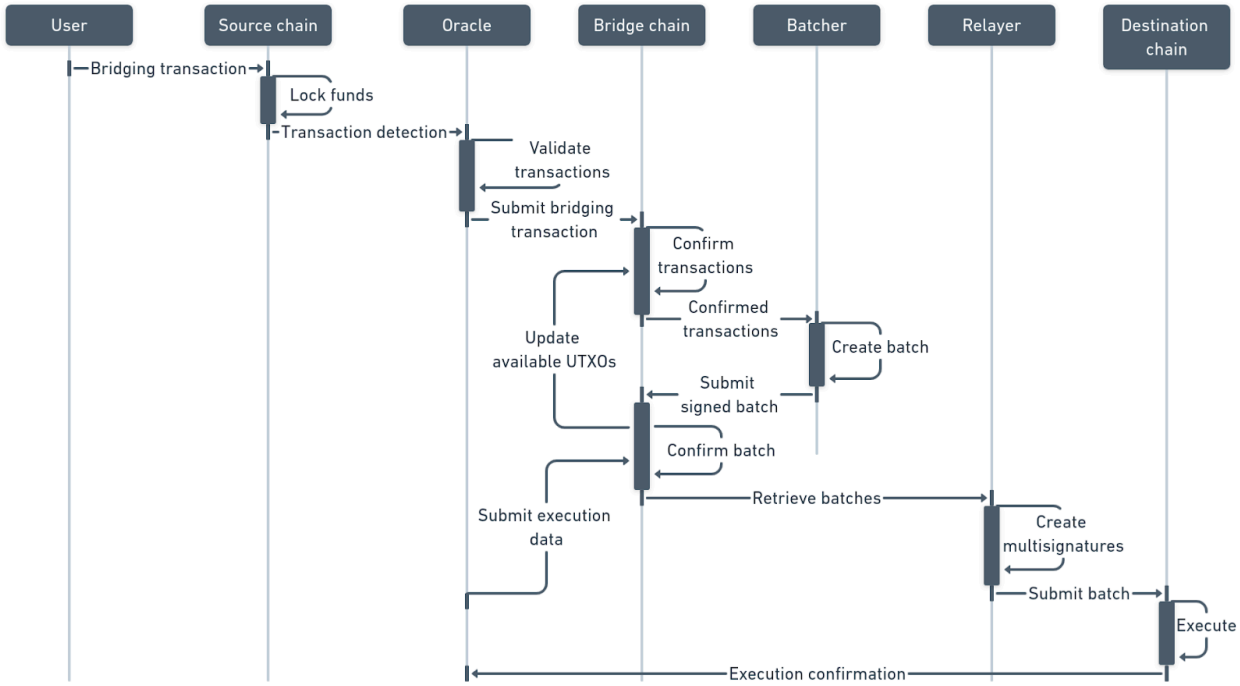

Figure 3: Reactor Workflow

1. **Bridging Initiation**: A transaction on the source chain is subimtted in order to to cover fees and the bridged funds. The funds are locked at a bridge-controlled multisig address. The transaction metadata includes details like the destination address, chain ID, and amount.

2. **Detection**: Validators run an off-chain Oracle to monitor bridging requests by observing source chain transactions that output UTXOs to the bridge's multisig address.

3. **Witnessing**: Oracles submit a transaction (a cliam) to the bridge blockchain to confirm they have observed the bridging request.

4. **Batching**: Confirmed bridge transactions are stored on the bridge blockchain until a batch is created. A batch, a transaction executed on the destination blockchain, forms when enough transactions are confirmed or the maximum time limit between batches is reached. The batch may be UTXO or EVM transaction depending on the type of the destination chain.

5. **Batch Confirmation**: Validators confirm the batch by signing it with their private keys controlling the destination blockchain's multisig address or smart contract.

6. **Submission**: The batch transaction is submitted to the destination blockchain. The funds are unlocked and transferred to the destination addresses.

The Bridge User Interface

To ensure a seamless and intuitive user experience, a meticulously designed, user-friendly interface has been developed for interacting with the Reactor Bridge. To use the bridge, users should connect to the application using their source chain wallet.

If the source chain is either the Prime and Vector network, users should open their Eternl Beta wallet, make sure they are connected to the desired source network, and connect their wallet as a dApp account.

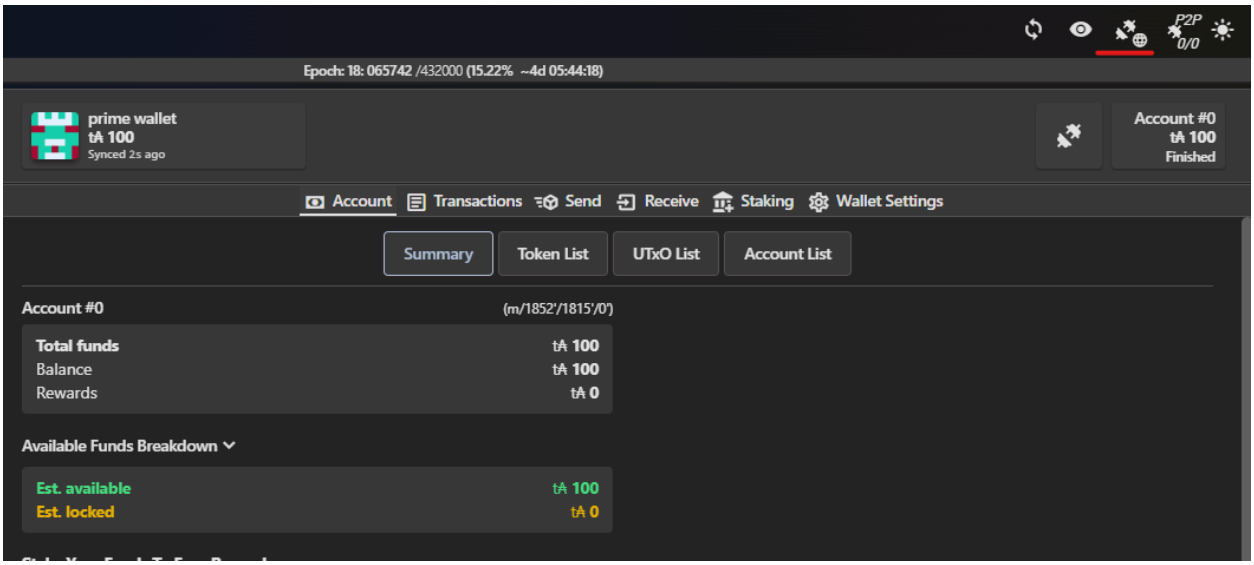

Figure 4: Eternl user interface

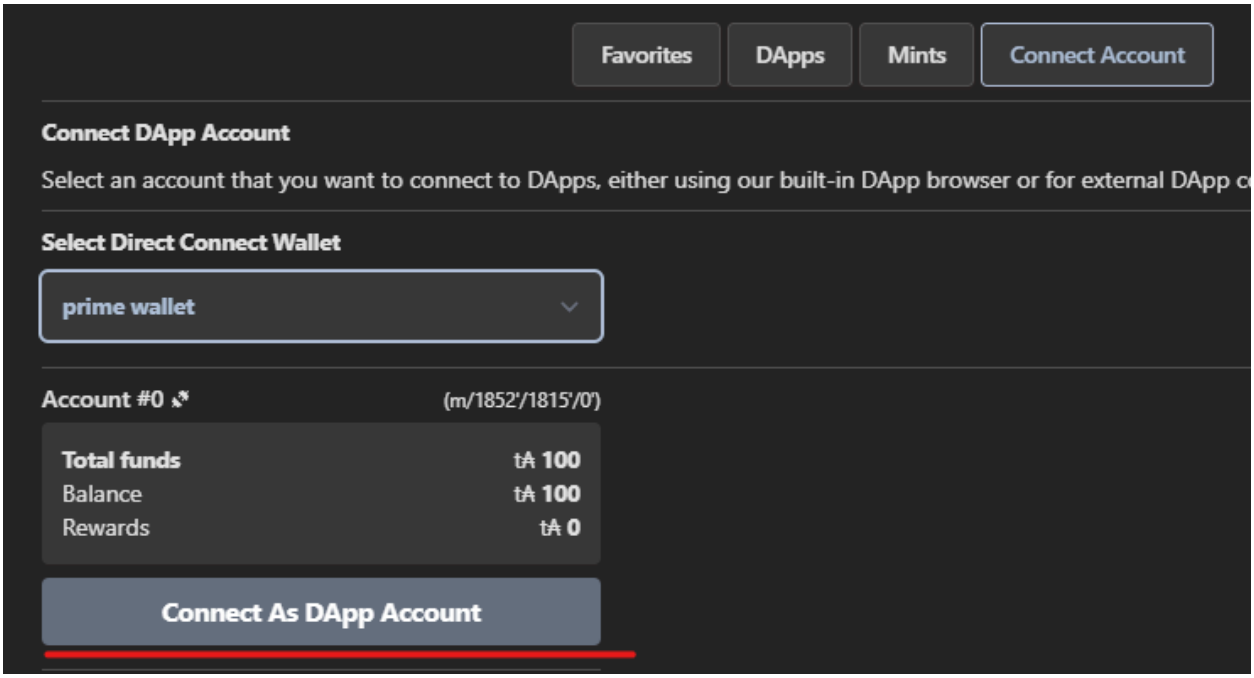

Figure 5: Eternl - Connect DApp Account

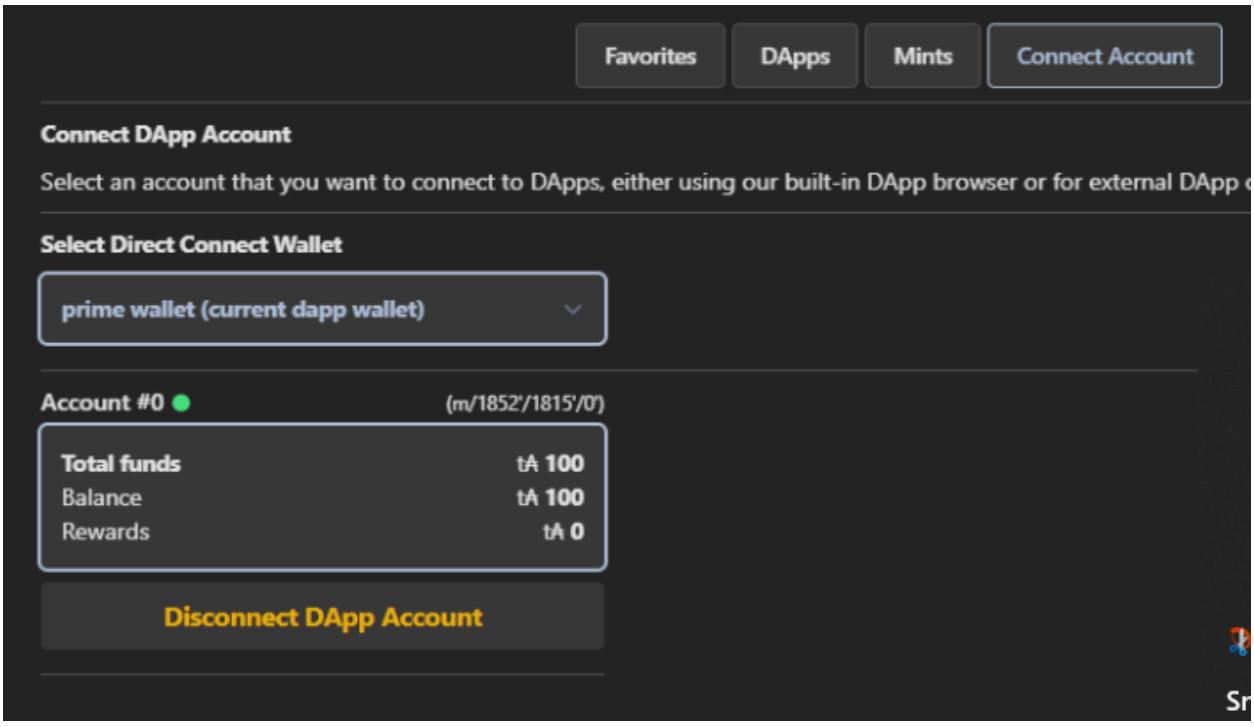

Figure 6: Eternl - DApp account connected

After this, they should navigate to the reactor bridge, select their desired source network and destination network. After this, they should click "Connect Wallet" and

follow the browser wallet prompts.

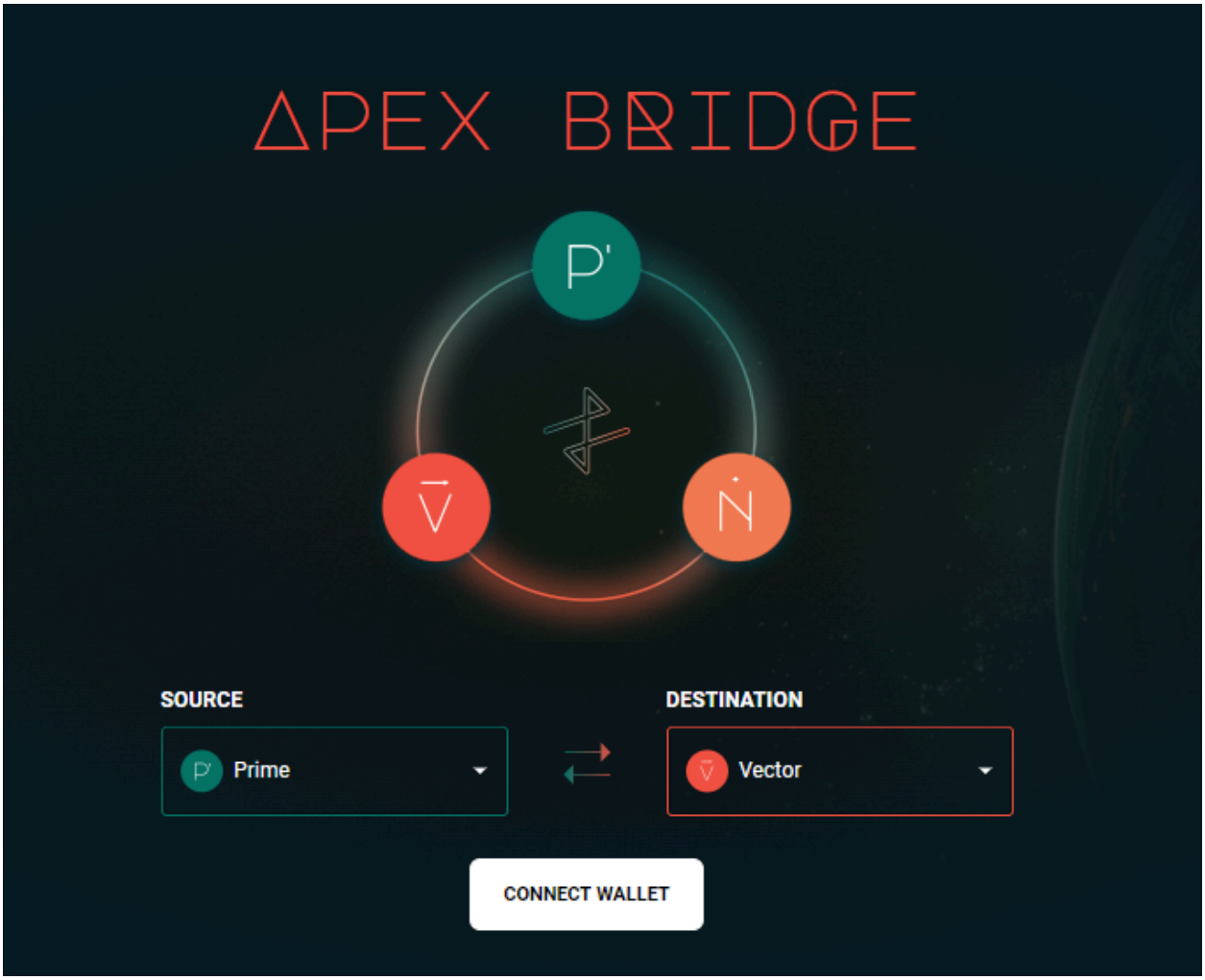

Figure 7: Reactor Bridge - wallet not connected

Once the wallet connection is established, users gain access to a comprehensive platform where they can conveniently input all necessary data to initiate bridging transactions to any destination chain of their choice.

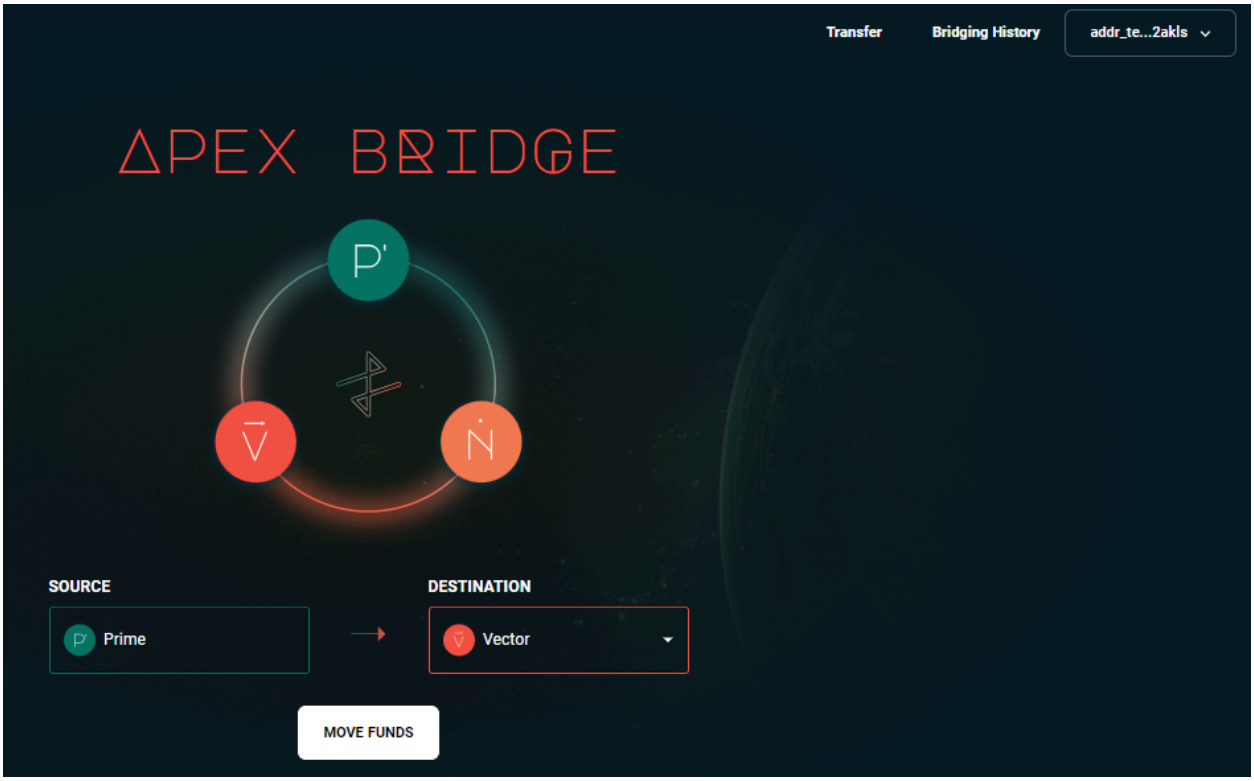

Figure 8: Reactor Bridge - wallet connected

To bridge funds, users should enter the address on the destination chain to which they wish to bridge funds, as well as the amount they wish to bridge. Next, they should click "Move Funds" and follow the wallet prompts to submit the transaction.

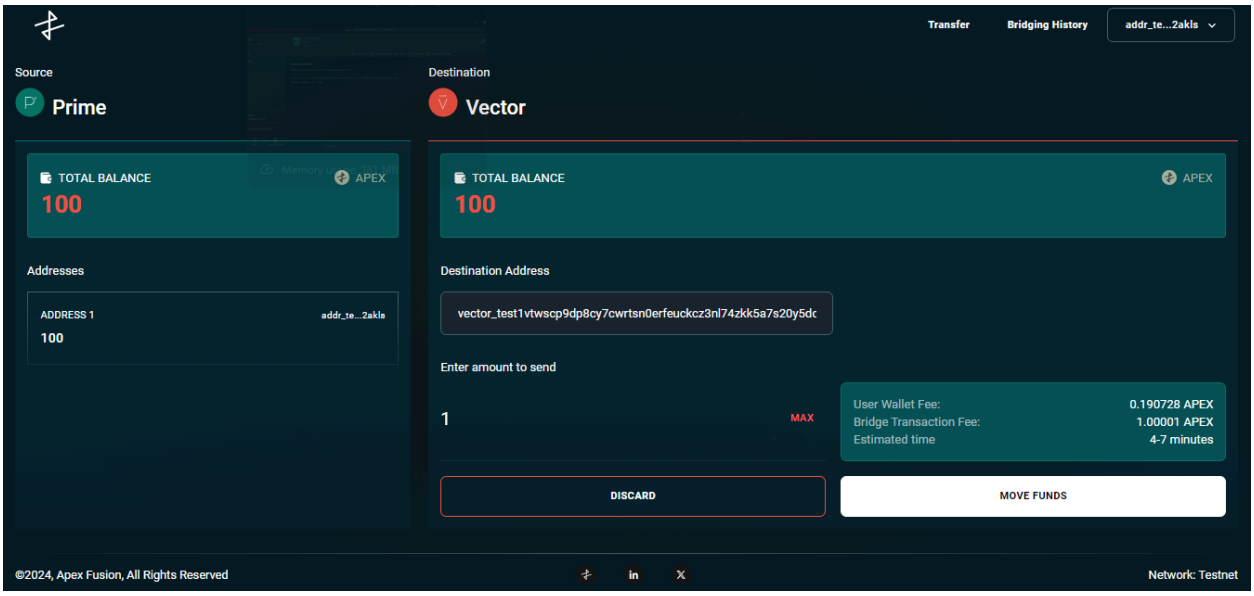

Figure 9: Reactor Request User Interface

Once the transaction has been submitted, the status of the bridging transaction will be visible to the users Upon completion, the funds will be available on the destination chain at the previously entered destination address.

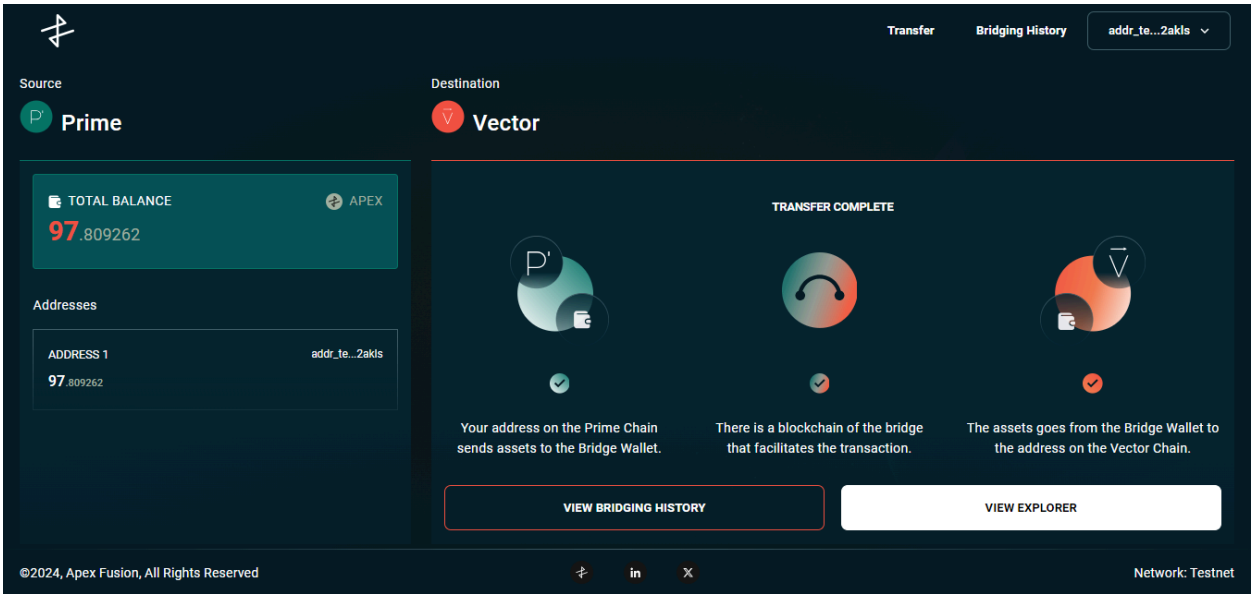

Figure 10: Reactor transaction complete

In order to provide an even richer user experience, users are empowered with the ability to track every bridging transaction they initiate. Through a dedicated transaction tracking feature, users can conveniently monitor the progress and status of each bridging transaction they have sent. Figure 5 exemplifies this feature, displaying a comprehensive view of the status of all bridging requests initiated by the user. This allows users to stay informed about the progress of their transactions and provides transparency throughout the bridging process, ensuring peace of mind and confidence in the platform.

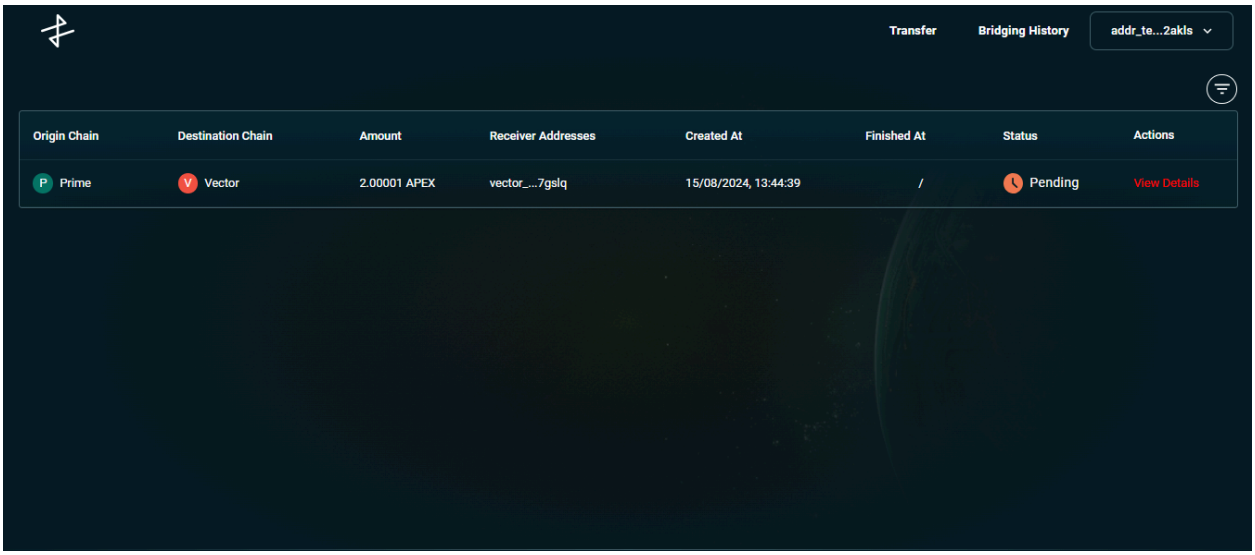

Figure 11: Reactor Requests Current Status
