Kwil Rewards Protocol

The Kwil Rewards Protocol is a protocol for administering on-chain rewards based on the state of a Kwil database. Projects can use the Kwil rewards protocol to:

- pay users for usage of their data
- incentivize users for uploading data to a database
- reward node operators for functioning properly and validating the network

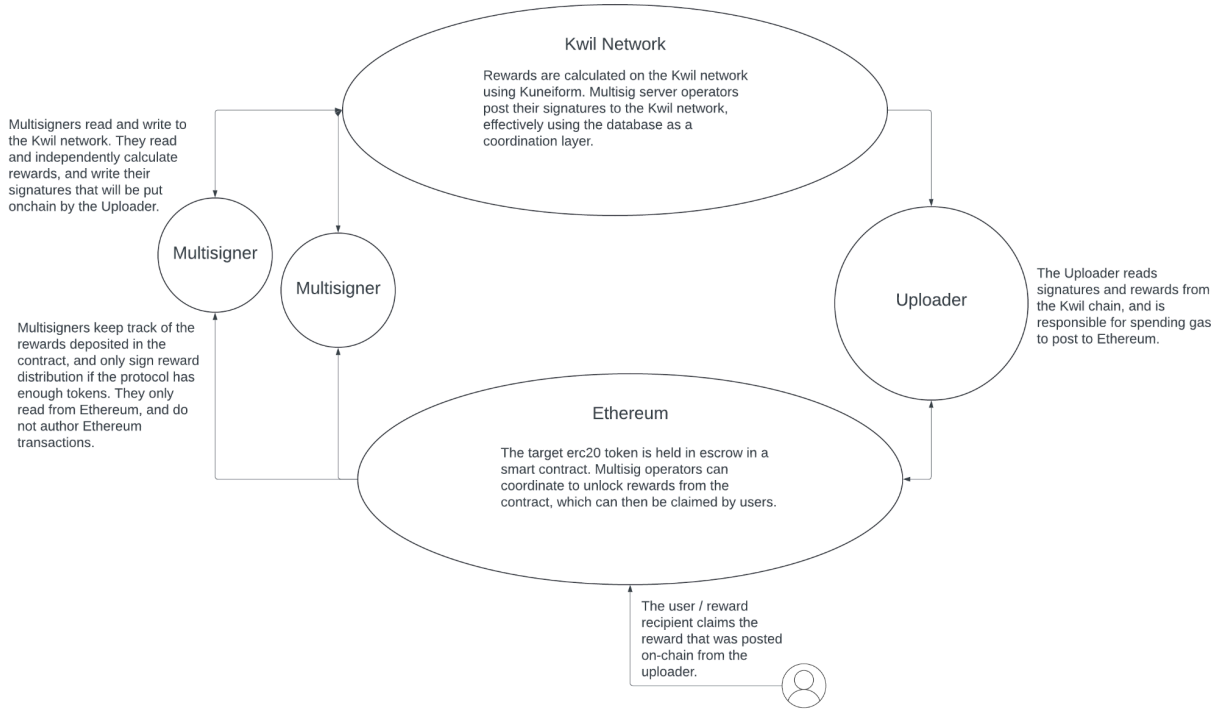
Terminology

- "Kwil Database" / "Kwil Network": A deployment of 1 or more Kwil nodes agreeing to database state. These terms are used interchangeably, as the reward protocol should function regardless of how many nodes exist.
- "Multisigner": A server that is responsible for signing reward distribution messages and posting them to the Kwil Network. Multisigners use a secp256k1 private key for signatures.
- "Uploader": A service that is responsible for reading multisigner signatures from Kwil and uploading them to Ethereum. It is responsible for expending gas and ensuring that the transaction is included in a block.
- "Ethereum" / "L1": The EVM network where the target erc20 token lives. We will generally consider this to be Ethereum, but it can be any EVM network, and thus these terms may be used interchangeably.
- "Reward Root": A Merkle root where each leaf node is a hash of a wallet address and a reward amount.
- "Sign Hash": A unique message that is signed by multisigners and verified on-chain. It is a hash of a reward root, the total amount of rewards in a reward root, a reward nonce, and the Solidity smart contract address where the signatures will be verified.

Technical Overview

The Kwil Rewards Protocol has 3 main functions:

- Holding rewards in escrow within a smart contract.
- Deterministically calculating rewards according to an algorithm defined by the database deployer on the Kwil network.
- Unlocking tokens held in escrow on-chain upon receiving enough signatures from a multisig.



Components

There are four main components that are necessary for the Kwil rewards protocol:

1. Escrow Rewards Contract: A Solidity contract that is capable of holding rewards in escrow, and only releasing them with the signatures of a sufficient number of signers.
2. Kwil Rewards Extension: A Kuneiform Precompile that aggregates rewards for periods. It should provide public actions that can be used by multisigners and the uploader to read and post data accordingly.
3. Multisig Uploader: A server that can read signatures posted to the Kwil network and upload those signatures to the L1.
4. Multisigner Server: A server that holds a secp256k1 private key which can read data from the Kwil network and sign reward messages. It should also be able to read data from Ethereum to know if the contract does not have enough funds or if its configured private key not a signer for the contract.