ReFuel Wallet

Easy Fuel Onboarding with EVM Wallets

ReFuel Wallet Components

The JSON-RPC Wrapper

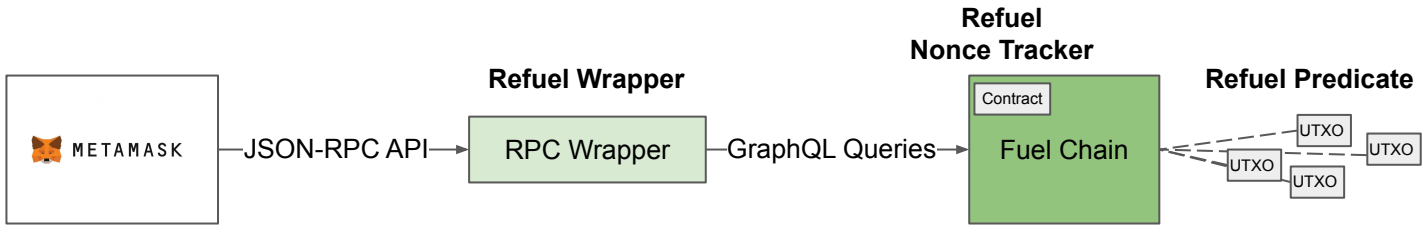
Allows for pointing existing wallets to a known interface while serving up Fuel chain data instead

The ReFuel Predicate

Allows for UTXOs to be controlled by signed Ethereum style transactions

Nonce Manager Contract

Prevents replay attacks by using a contract that issues tokens representing allowed account nonces



The JSON-RPC Wrapper

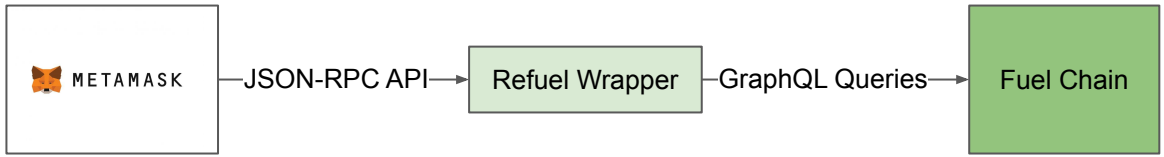
The JSON-RPC Wrapper

Emulates the common JSON RPC API calls that wallets use to connect to EVM chains.

For example:

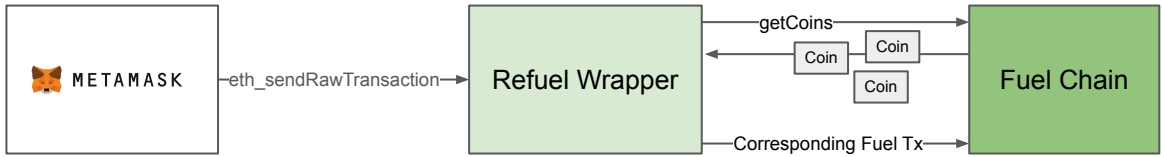
- eth_getBalance
- eth_blockNumber
- eth_getBlockByNumber
- eth_call
- eth_estimateGas
- eth_feeHistory

https://ethereum.org/en/developers/docs/apis/json-rpc



The JSON-RPC Wrapper

One of the key methods to emulate is the ***eth_sendRawTransaction*** method. The wrapper will have to decode the message (RLP encoding) and construct a corresponding Fuel transaction that translates the Ethereum addresses into ReFuel addresses and finds the appropriate coins which can be unlocked under the ReFuel Predicate with the signed EVM raw transaction.

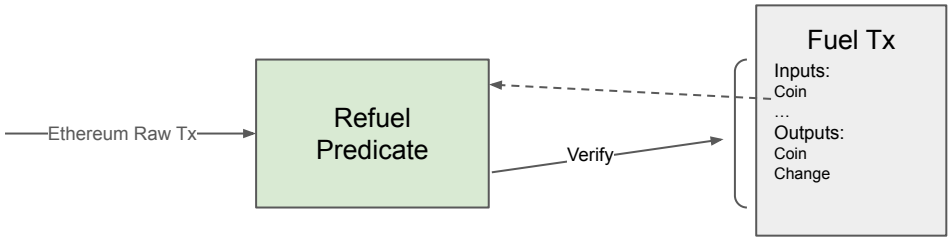


The ReFuel Predicate

The ReFuel Predicate

Coins held by an Ethereum address are locked behind a ReFuel Predicate that only allows coins to be spent if they are part of a specific transaction construction that matches a signed Ethereum transaction. This predicate acts as the corresponding ReFuel address on the Fuel side and is specific to a corresponding Ethereum address.

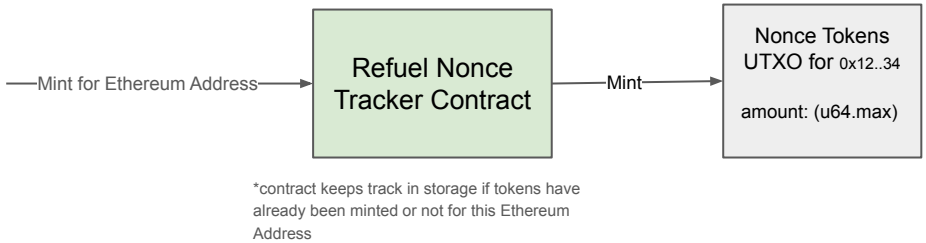
The predicates main job is to decode the signed tx (RLP encoding) and validate that the transaction inputs and outputs, as well as the signer, matches the given tx.



The Nonce Tracker

The Nonce Tracker

The nonce tracker contract mints a set of tokens for a given Ethereum address in a deterministic way so that the ReFuel Predicate can know what the token asset_id will be before they're even minted. Before a ReFuel wallet can facilitate transactions for an Ethereum address, it needs to be initialized and have these tokens minted for the predicate to see. Interaction with "uninitialized" ReFuel predicates, like sending coins to them, can still be done but will require "initialization" in order for the receiver to use them.



The Nonce Tracker

When the ReFuel Predicate validates a signed tx it looks at the nonce and checks that a nonce coin UTXO is included in the tx inputs with the amount [***u64.max - nonce***]. It then also checks the nonce tokens are included in the output with 1 token burned and the rest put into a single output for verification of the next tx nonce. In this setup, even if a tx fails, the nonce is always tracked correctly and increment accordingly. Additionally, it doesn't require the predicate to know any contract state or force special scripts that interact with a central contract.