

Rütli-Coin



Table of Contents



Project Idea



Overview



React



Smart Contract



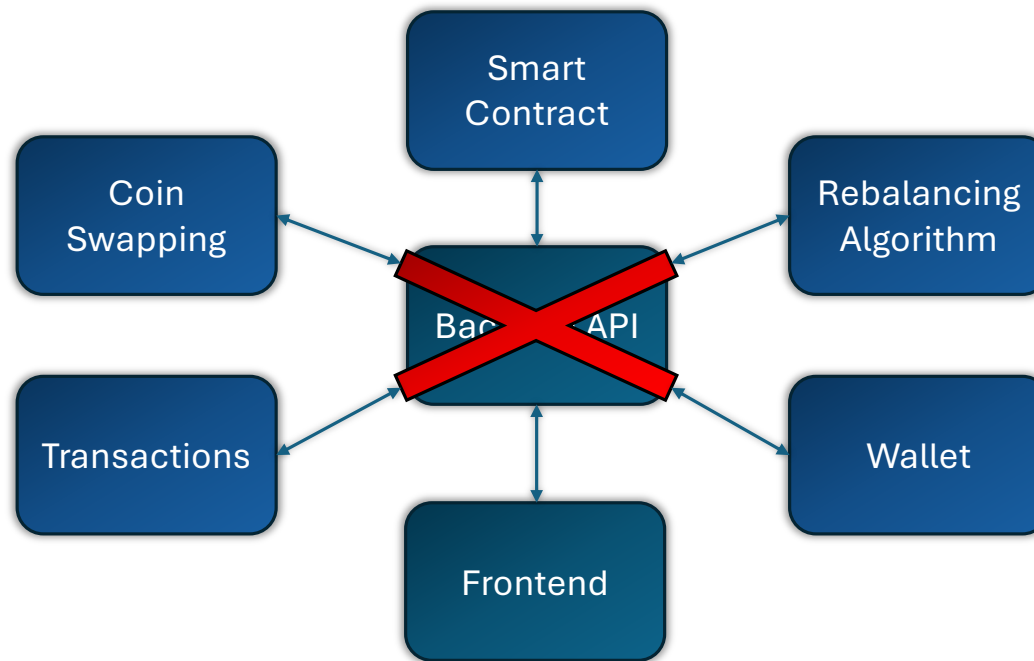
Uniswap Pool



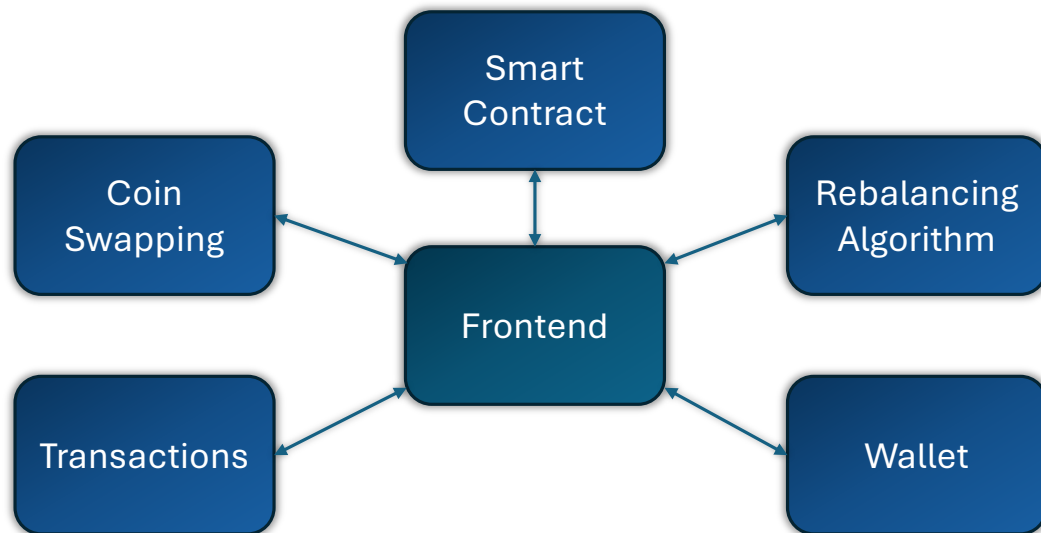
Rebalancing



Project Idea - First Iteration



Project Idea - Frontend-only approach



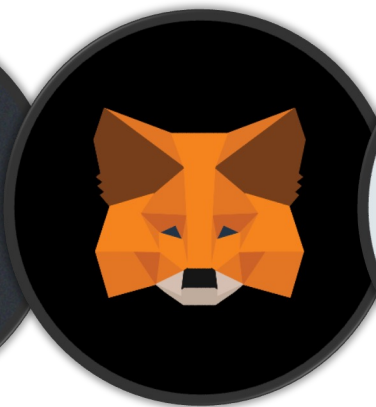
- Lower architecture complexity
- Ethers.js library
 - Interaction with SmartContract
- MetaMask integration
- Seemed to be the simpler approach

Overview - Tools



ViteJS

Development
server



MetaMask

Wallet



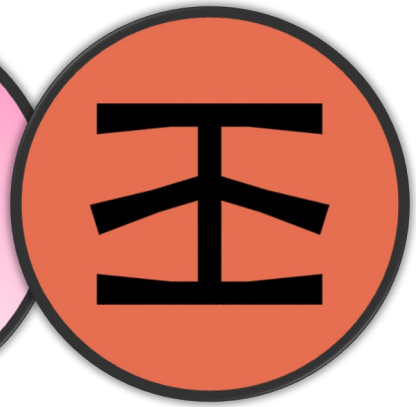
Remix IDE

Solidity/
Smartcontract
IDE



Uniswap

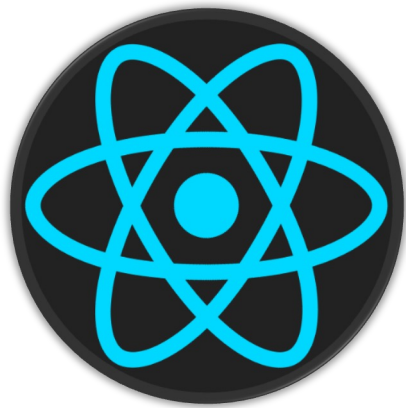
DEX
(Decentralized
Exchange)



Infura

API for
Blockchains

Overview - Frameworks



React

Javascript
Framework



ethers.js

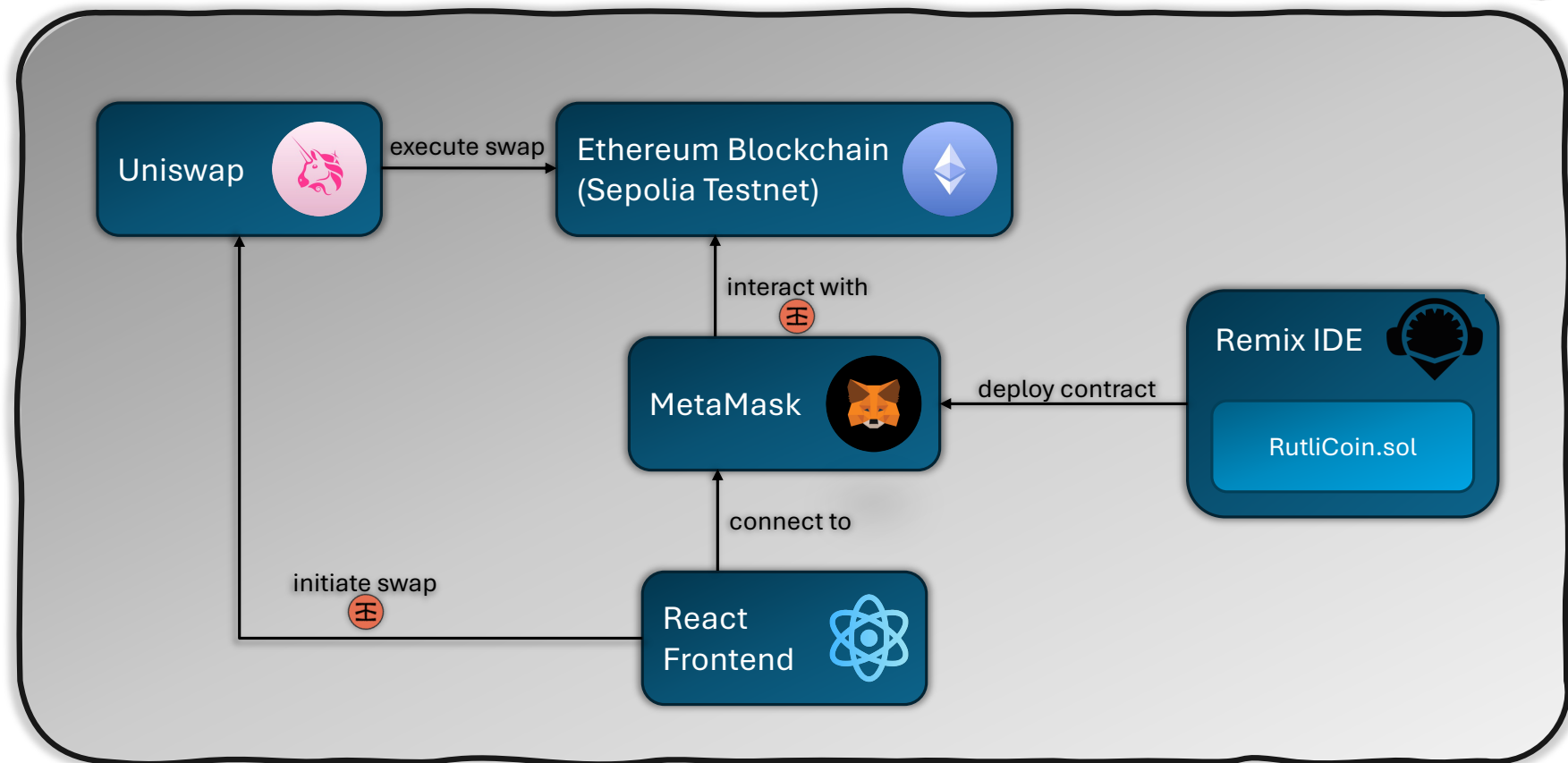
Interaction with
Ethereum
Blockchain



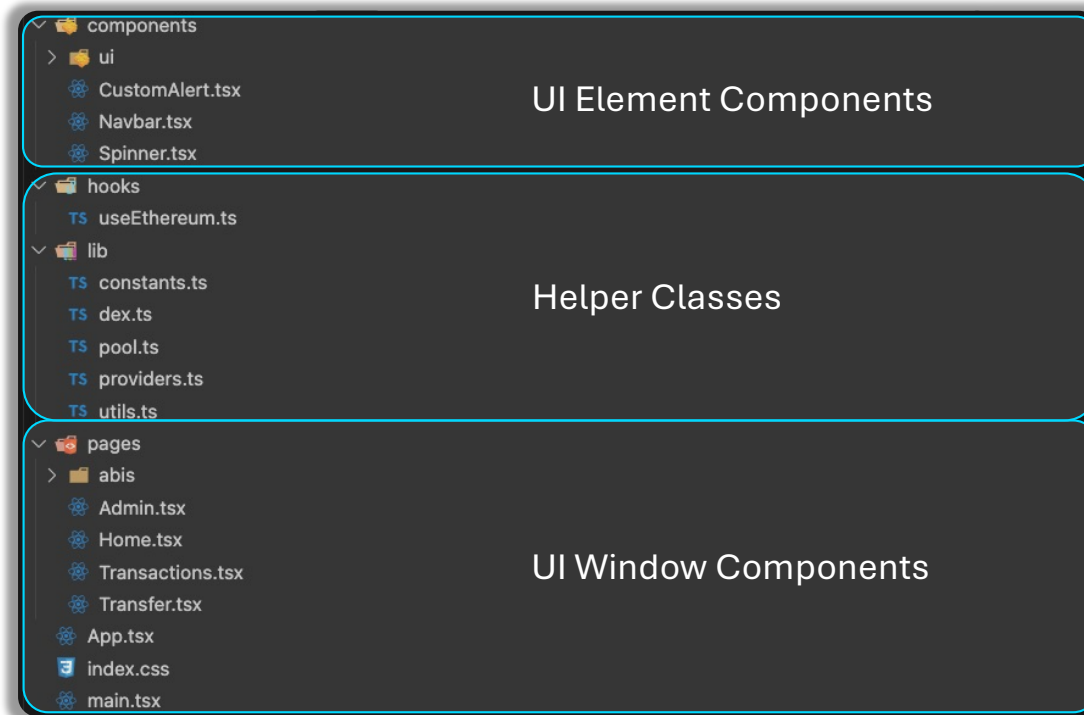
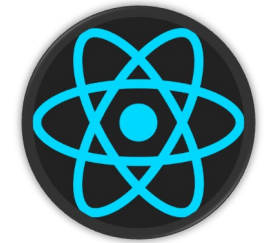
Tailwind CSS

CSS
Framework

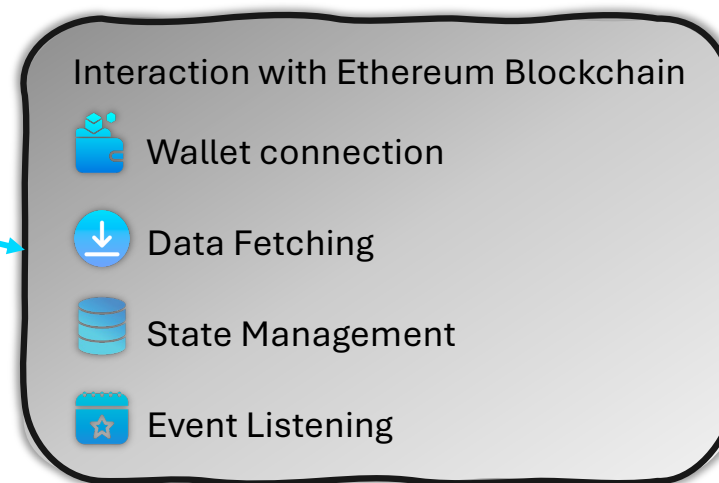
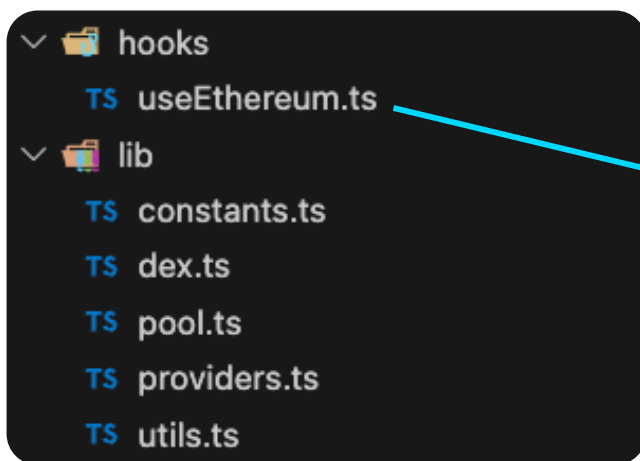
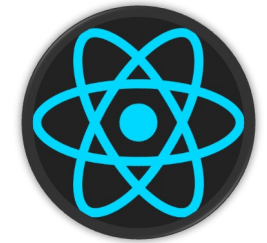
Overview - Infrastructure



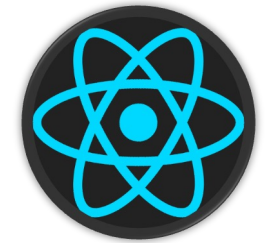
React - Project Structure



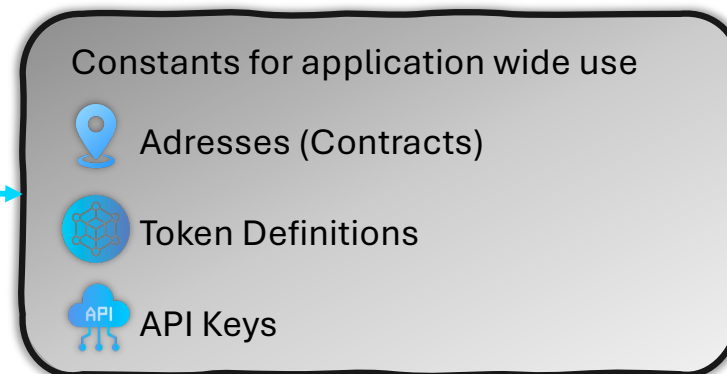
React - useEthereum



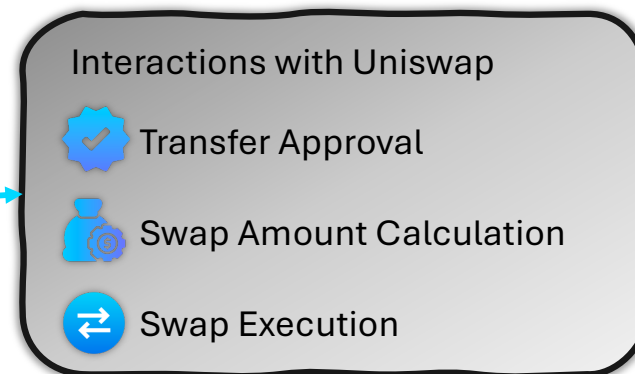
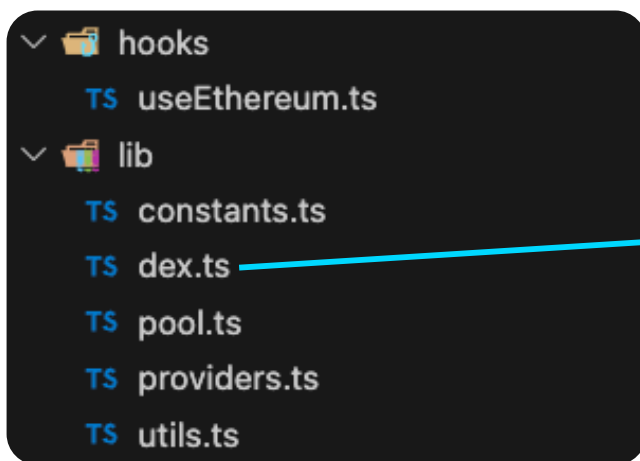
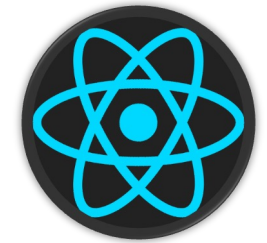
React - constants



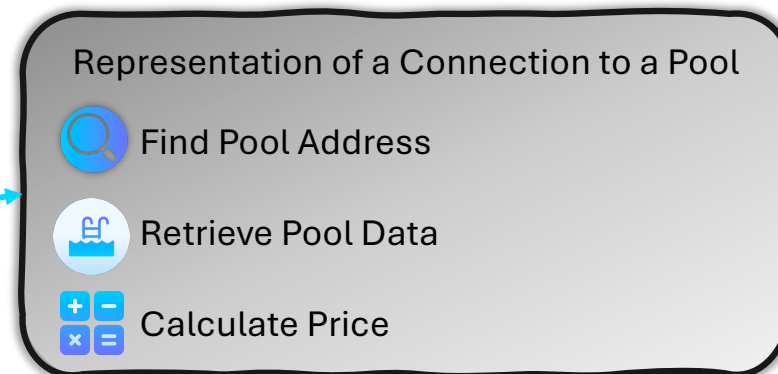
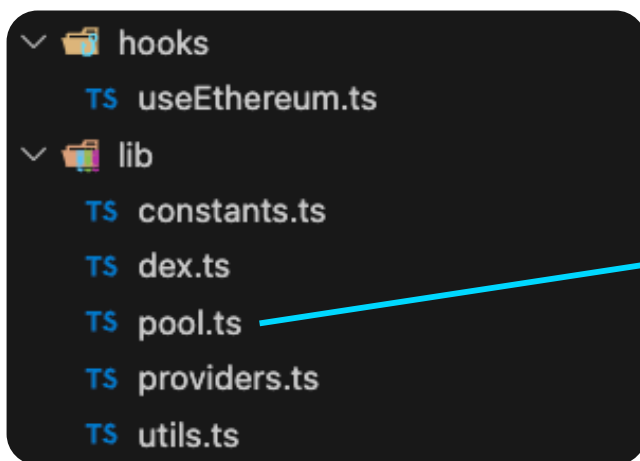
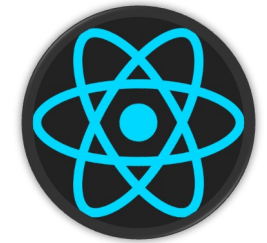
```
✓ hooks
  TS useEthereum.ts
✓ lib
  TS constants.ts
  TS dex.ts
  TS pool.ts
  TS providers.ts
  TS utils.ts
```



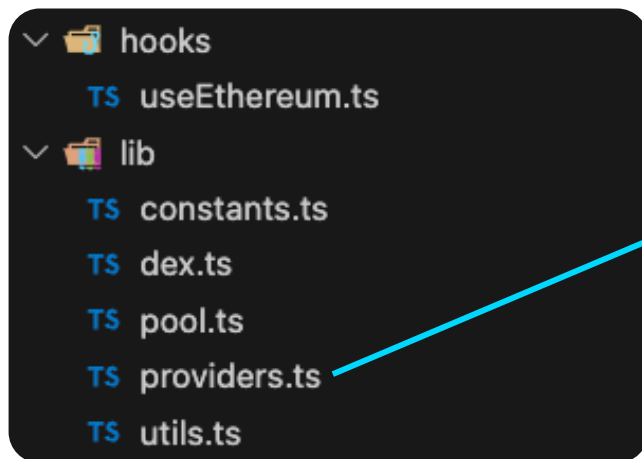
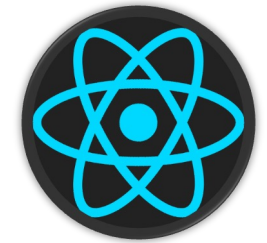
React - dex



React - pool



React - provider



Connection with Browser Extension



Creation of Web3Provider



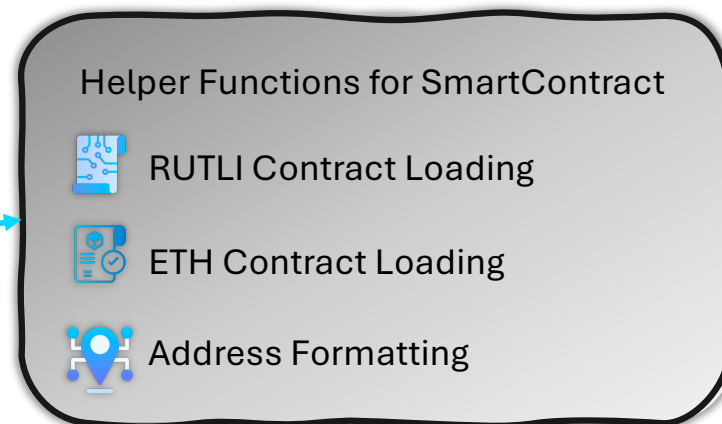
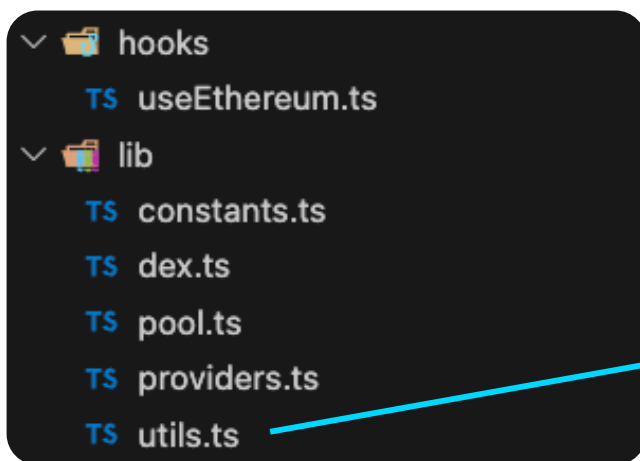
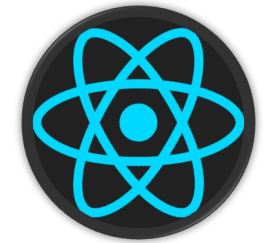
More Later

Allows access to ethereum object in Window object

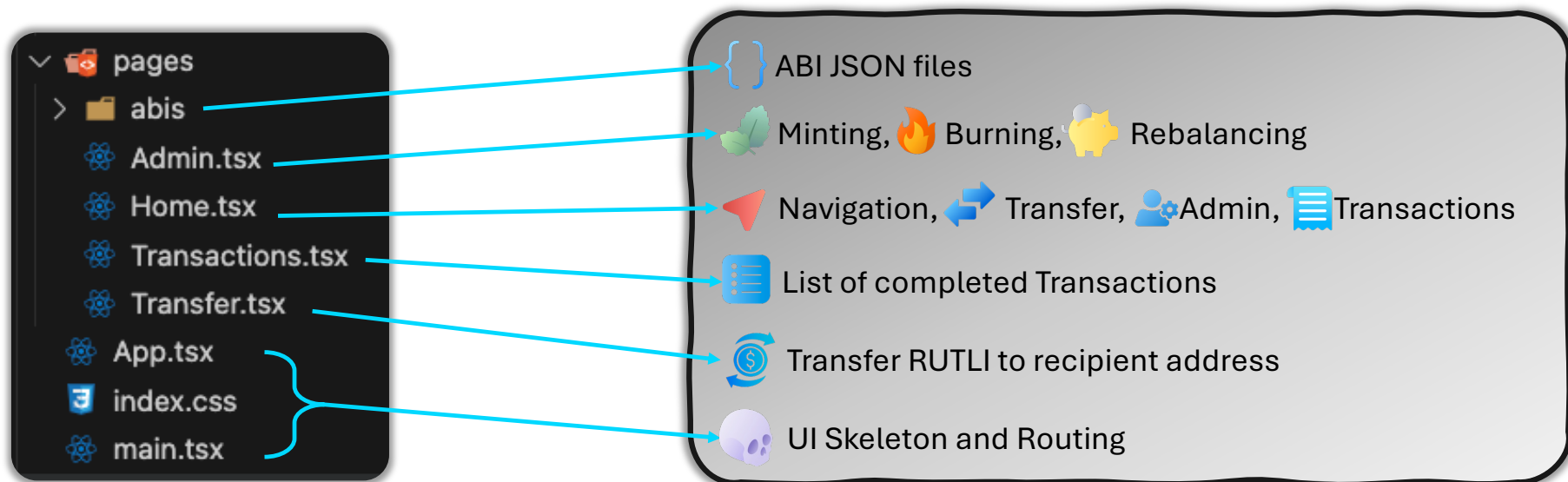
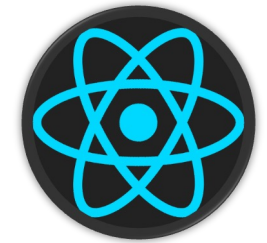


Allows Connection to Wallet

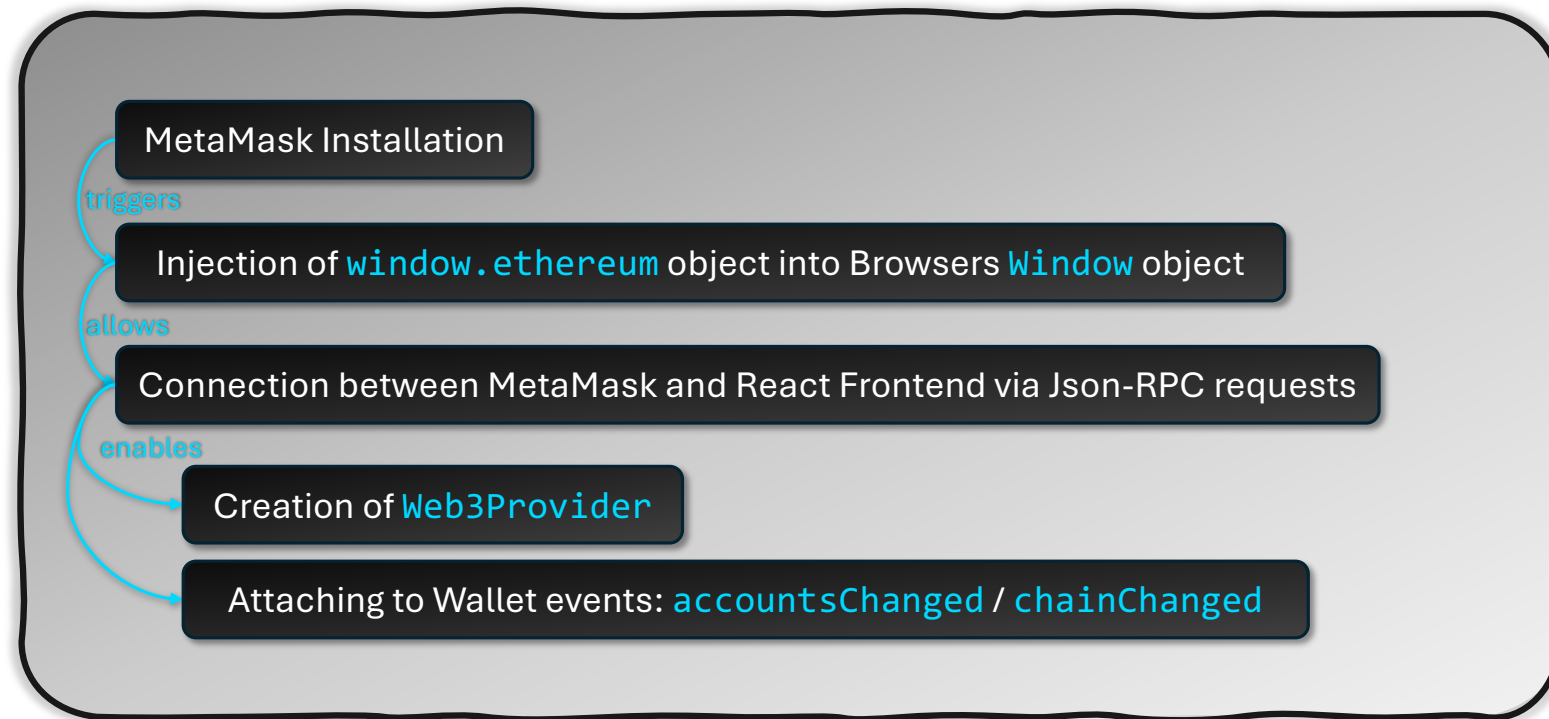
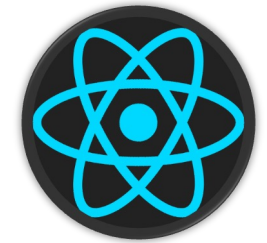
React - utils



React - Pages / Components



React – Wallet connection



Infura - Connection



Setup

Created Account



Retrieved API Key



Setup JsonRpcProvider

Setup Signer / Wallet

```
export const INFURA_PROVIDER = new ethers.providers.JsonRpcProvider("https://sepolia.infura.io/v3/API_KEY");
```

```
export const WALLET = new ethers.Wallet(Config.wallet.privateKey, INFURA_PROVIDER);
```

SmartContract

Specification

- ERC-20
- 18 decimals
- Initial supply = 1'000'000
- Ownable Contract

Minting 🌱

- No more than initial supply

Burning 🔥

- Not less than balance

Pausing ⏸

- Not really necessary

```
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/access/Ownable.sol";

contract ButliCoin is ERC20, Ownable {
    uint256 public constant WEI_MULTIPLIER = 10**18; //10^18 for wei
    uint256 public constant INITIAL_SUPPLY = 1000000 * WEI_MULTIPLIER; // 1 million tokens

    bool public paused = false;

    event TokensMinted(address indexed to, uint256 amount);
    event TokensBurned(address indexed from, uint256 amount);

    constructor() {
        _mint(msg.sender, INITIAL_SUPPLY);
    }

    function mint(uint256 amount) public onlyOwner {
        require(amount > 0, "Amount must be greater than 0");
        require(amount <= INITIAL_SUPPLY, "Amount too large"); // Safety cap
        _mint(msg.sender, amount);
        emit TokensMinted(msg.sender, amount);
    }

    function burn(uint256 amount) public {
        require(amount > 0, "Amount must be greater than 0");
        require(balanceOf(msg.sender) >= amount, "Insufficient balance to burn");
        _burn(msg.sender, amount);
        emit TokensBurned(msg.sender, amount);
    }

    function togglePause() public onlyOwner {
        paused = !paused;
    }
}
```



Uniswap Pool - Setup



Pair Selection

 Pair Selection

 Fee

 Low & High price ranges

Liquidity

 Starting price selection

 Add liquidity

Add liquidity Clear all ⚙️

Select pair

 ETH ⌵  RUTLI ⌵

0.30% fee tier Hide

Not created

0.01% ☐ 0.05% ☐ 0.30% ☒ 1.00% ☐

Not created Not created Not created Not created

Set price range Full range RUTLI ETH

Low price +

7491.7218 -

RUTLI per ETH

High price +

12475.58 -

RUTLI per ETH

10000

Starting ETH Price: 10000 RUTLI per ETH

Deposit amounts

0.5  ETH

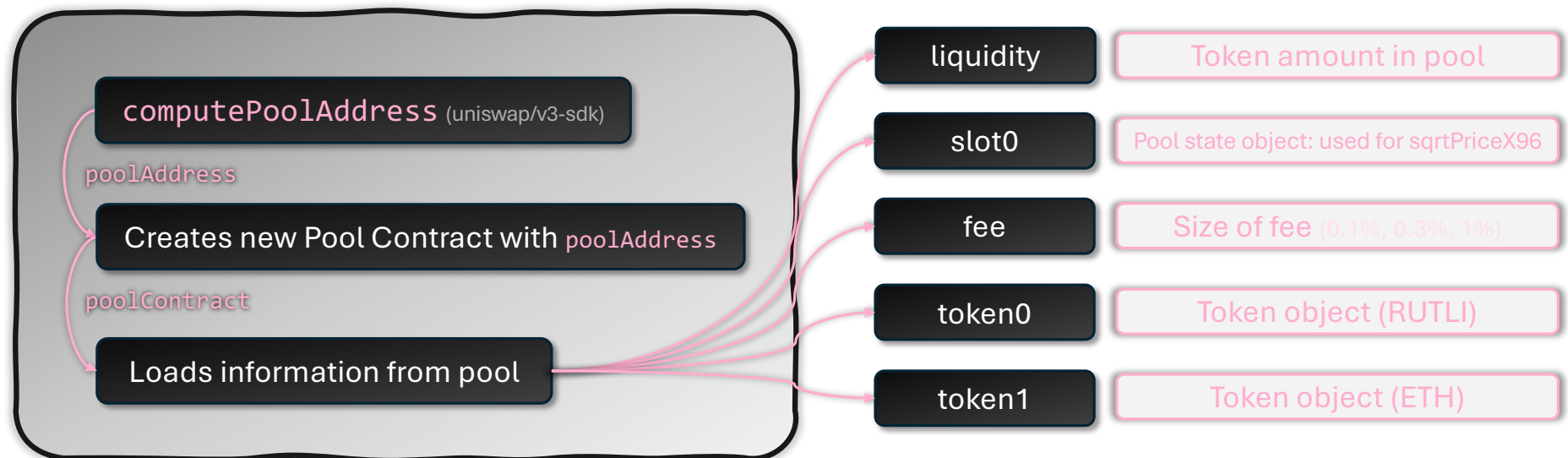
\$13,132.70 Balance: 3.96 MAX

6426.39  RUTLI

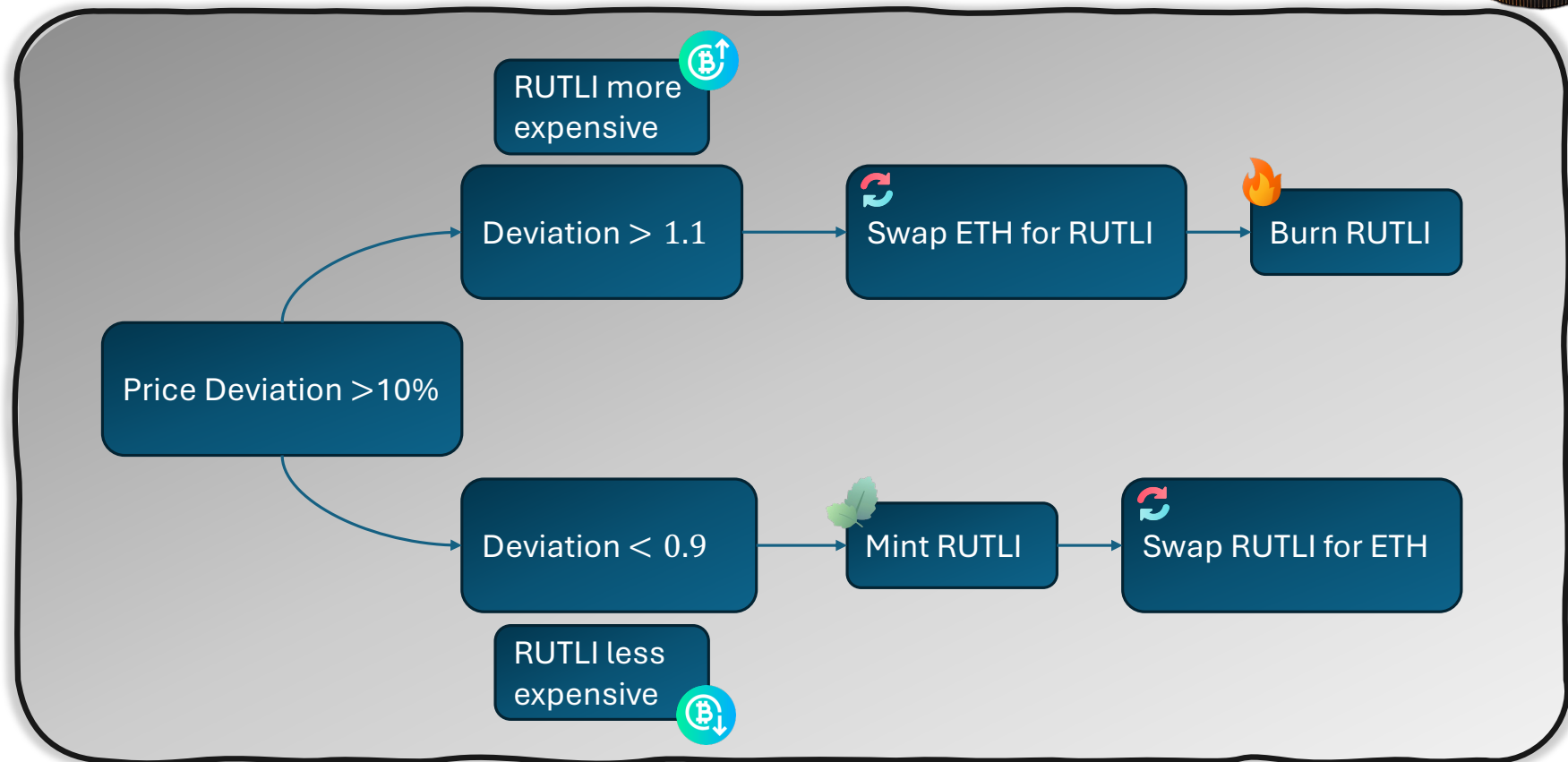
- Balance: 1.00M MAX

Approve RUTLI

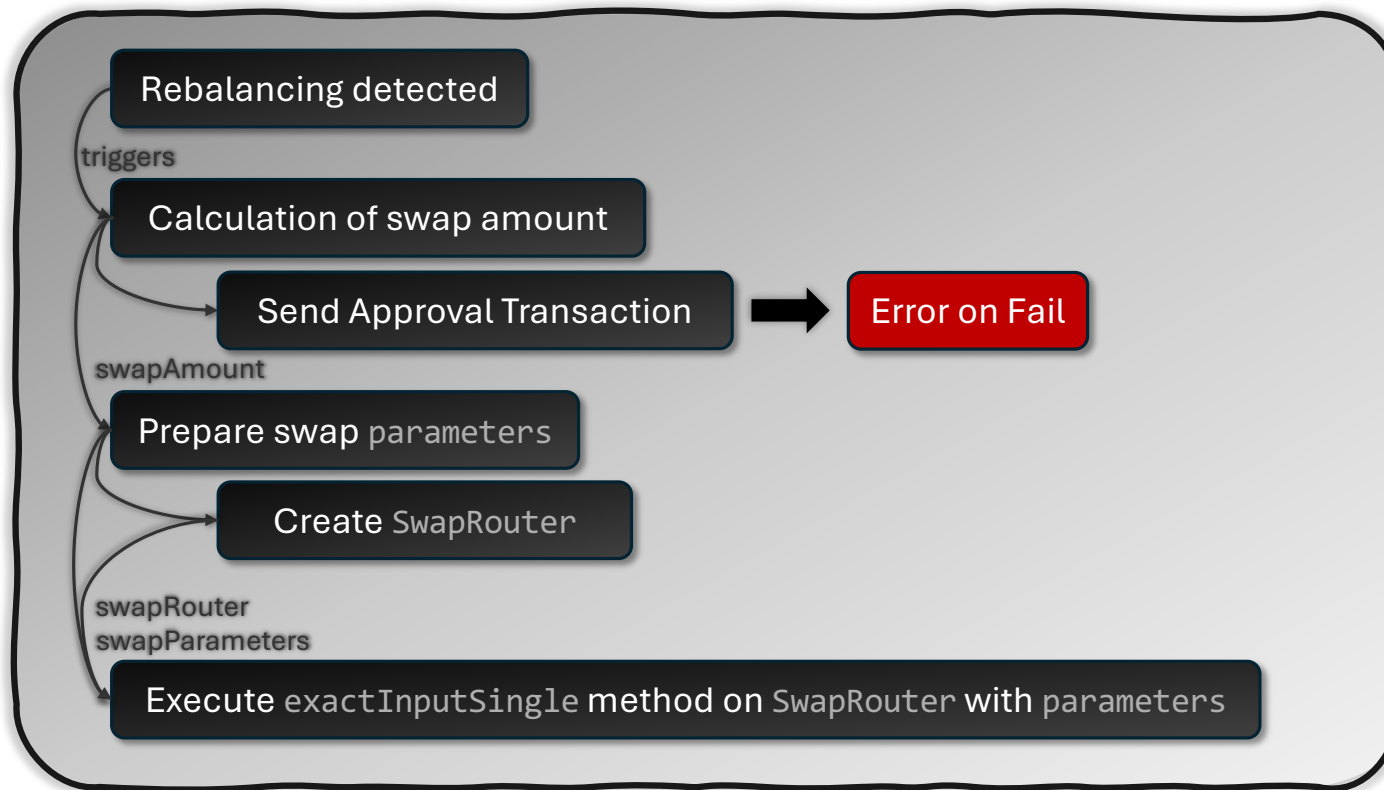
Uniswap Pool - Connection



Rebalancing Algorithm



Rebalancing – Swap



Rebalancing – Failing Transaction



- Automatic execution of Swap failed (Example: [Etherscan Transaction](#))
- Approval before transaction seems good
- Params seem good

Rebalancing – Failing Transaction



Transaction Hash:	0x8aeec0071d7717de6070ab2831a078e48445a2c04789b8d7c059622747c50135
Status:	Fail
Block:	7197784 2 Block Confirmations
Timestamp:	38 secs ago (Dec-02-2024 06:38:24 PM UTC)
Transaction Action:	Call Exact Input Single Function by 0x11eA0631...45f54a9b9 on 0x3bFA4769...50f7Ae48E
From:	Wallet Address: 0x11eA0631cC21AAcc8e347A214cA74A045f54a9b9
To:	Swap Router Address: 0x3bFA4769FB09eefC5a80d6E87c3B9C650f7Ae48E Warning! Error encountered during contract execution [execution reverted]
Value:	0.000017450594553526 ETH - [CANCELLED]
Transaction Fee:	0.00012296351880402 ETH
Gas Price:	5.28898098 Gwei (0.00000000528898098 ETH)

Rebalancing - Alternative



- Reverted to manual approach
- Application displays swap amount and from which Token in the console
- Admin has to swap manually as soon as token out of balance

Demo



Stablecoin Manager

RutliCoin (RUTLI)

Connected Account:
0x11ea0631cc21aacc8e347a214ca74a045f54a9b9

Balance:
565149.947854854642036818 RUTLI

Current Price:
0.39698991269825024 USD per RUTLI

Transfer

Admin

Recipient Address

Enter recipient's address

Amount

Enter amount

Transfer

