Uniswap

Decentralized Finance Study Case

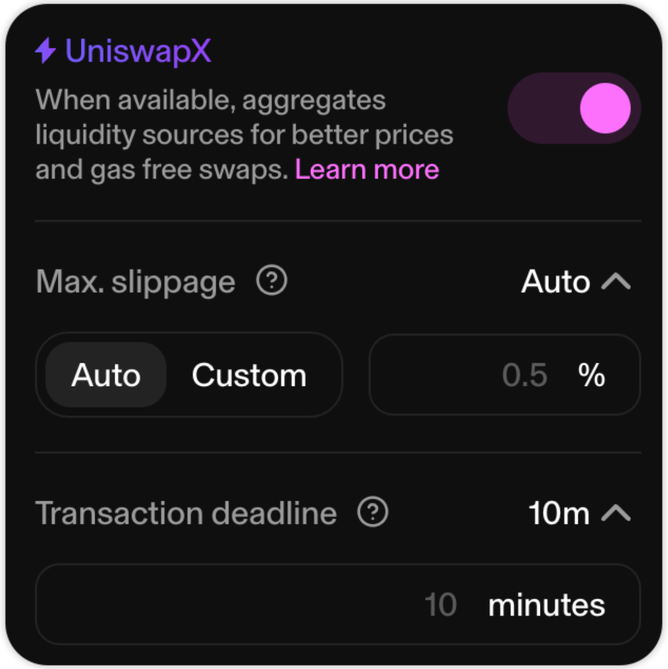
By 0xqige.eth

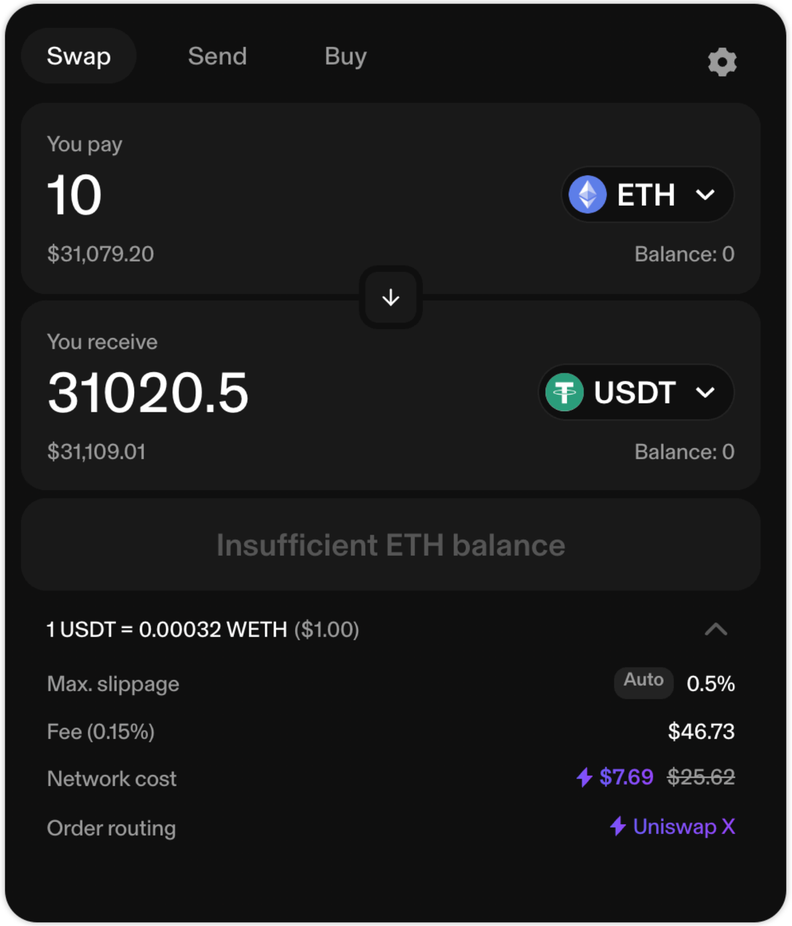
Decentralized Exchange

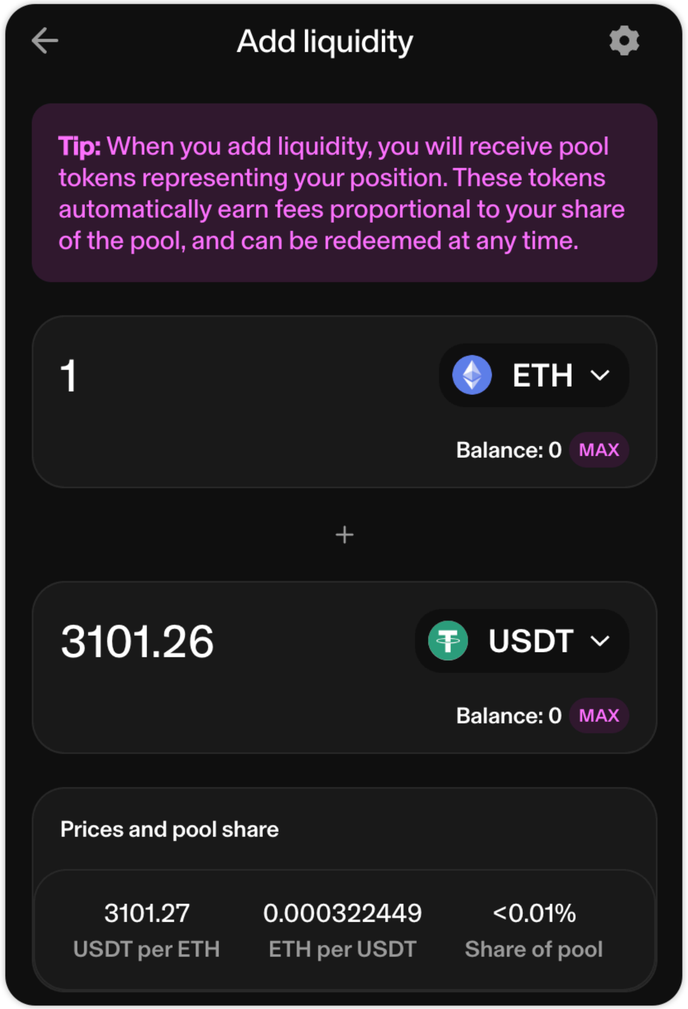
- ## 24H Trading Volume
  - 26/03/2024: 645 decentralized crypto exchanges with a total 24h trading volume of $3.43 Billion, a 18.92% change in the last 24 hours. Currently, the 3 largest decentralized exchanges by volume are Uniswap V3 (Ethereum), Pancakeswap V3 (BSC), and Uniswap V3 (Arbitrum One).
- ## Permissionless
  - Anyone can create a trading pair for any two ERC-20 tokens
- ## Non-custodial
  - Nobody can shut it down or steal funds(unless the blockchain is compromised or the smart contract have a bug)
- ## Censorship-resistant
  - Any one can send transactions, not need KYC

User Interface

Uniswap





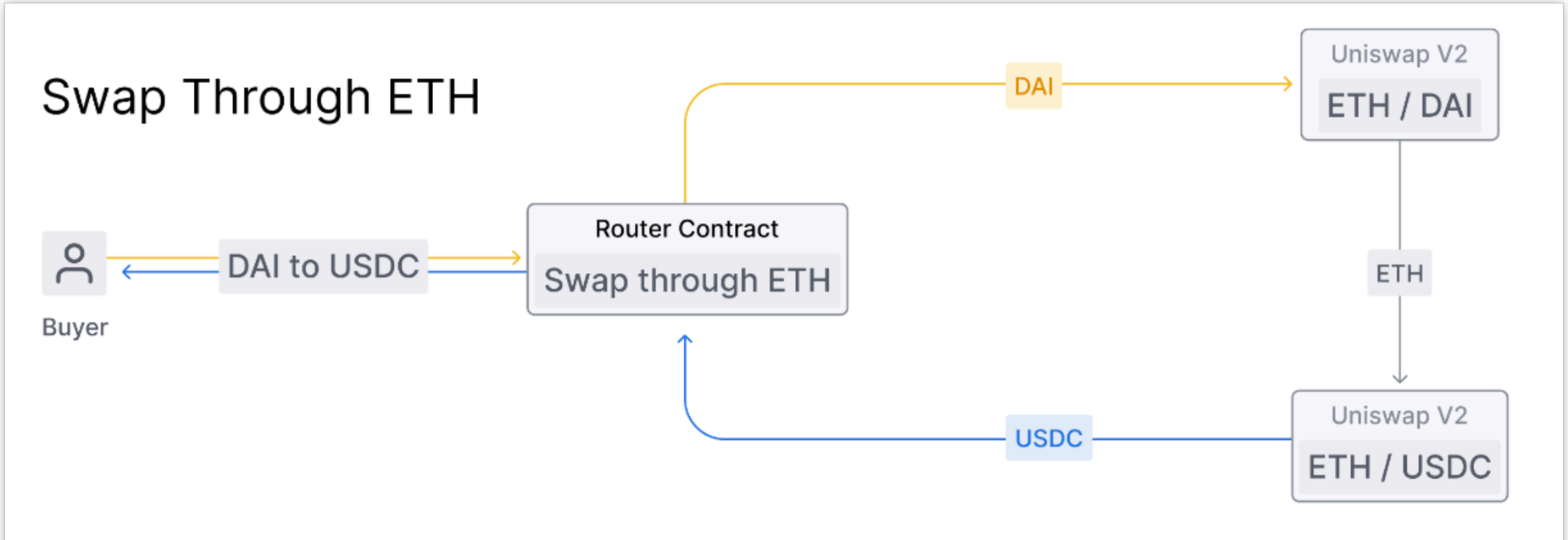


Swap

Uniswap Feature

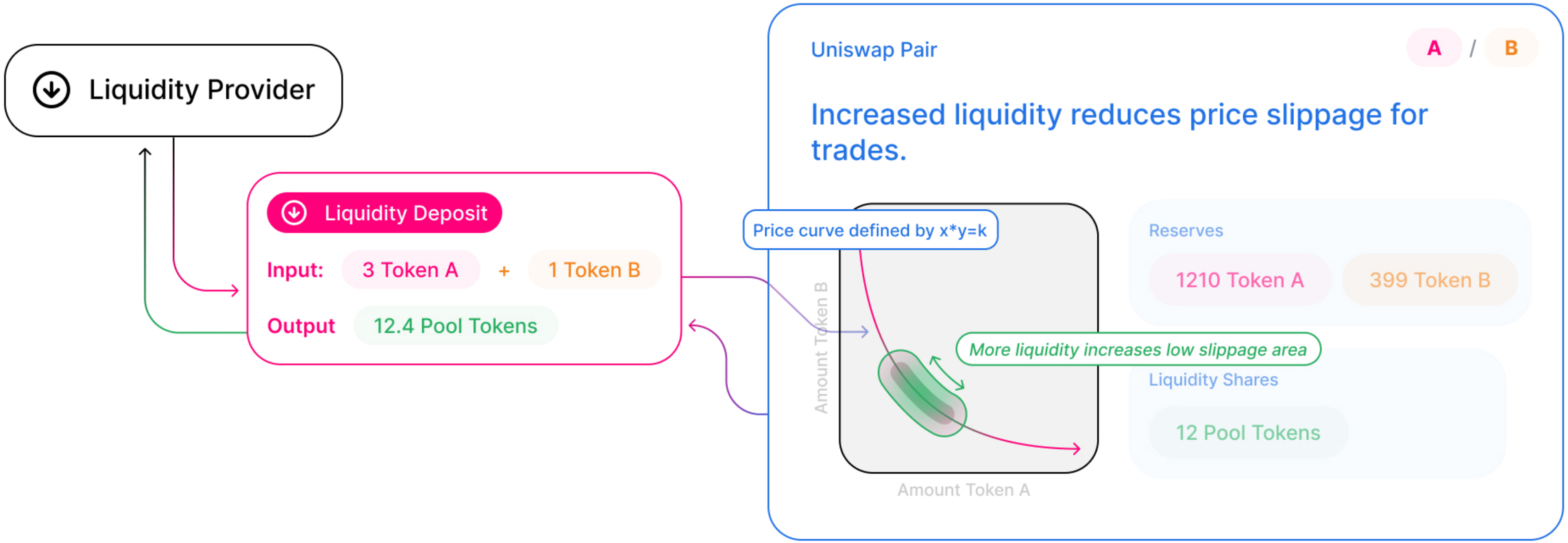
Direct Swap

Swap Through ETH



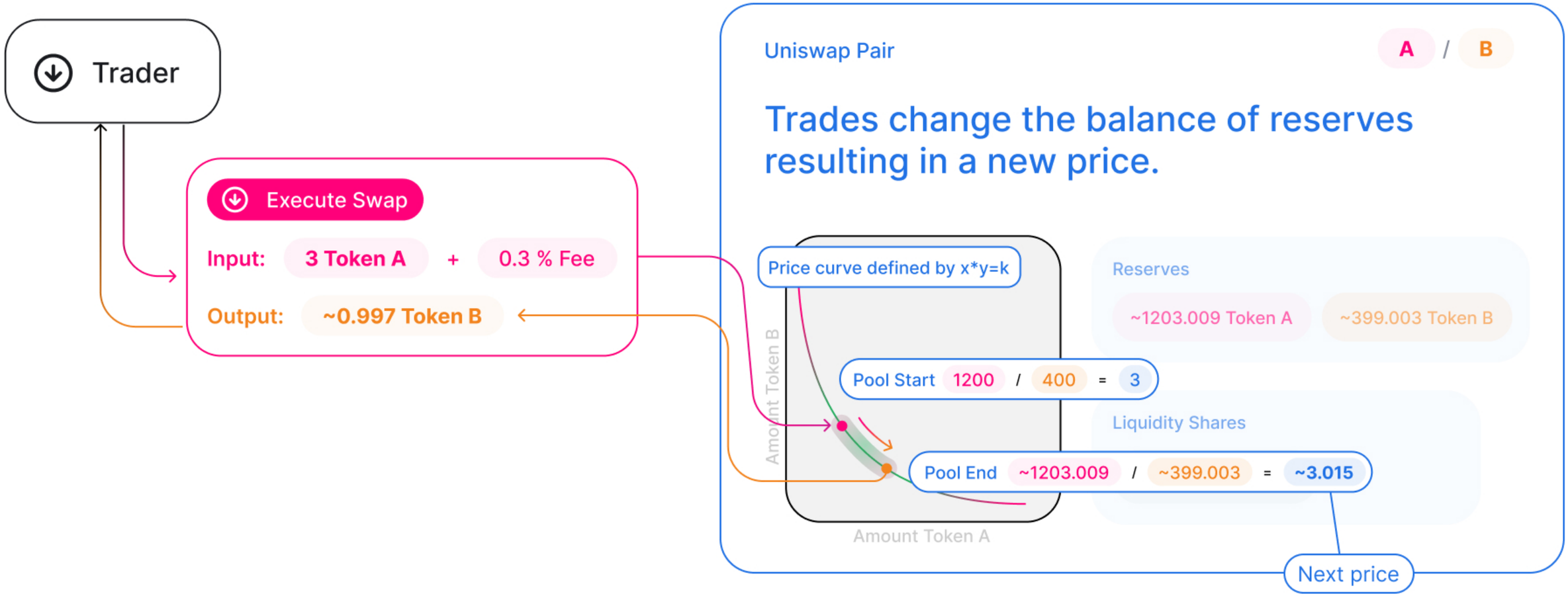
How Works-Pool

UniswapV2 Mechanics



How Works-Swap

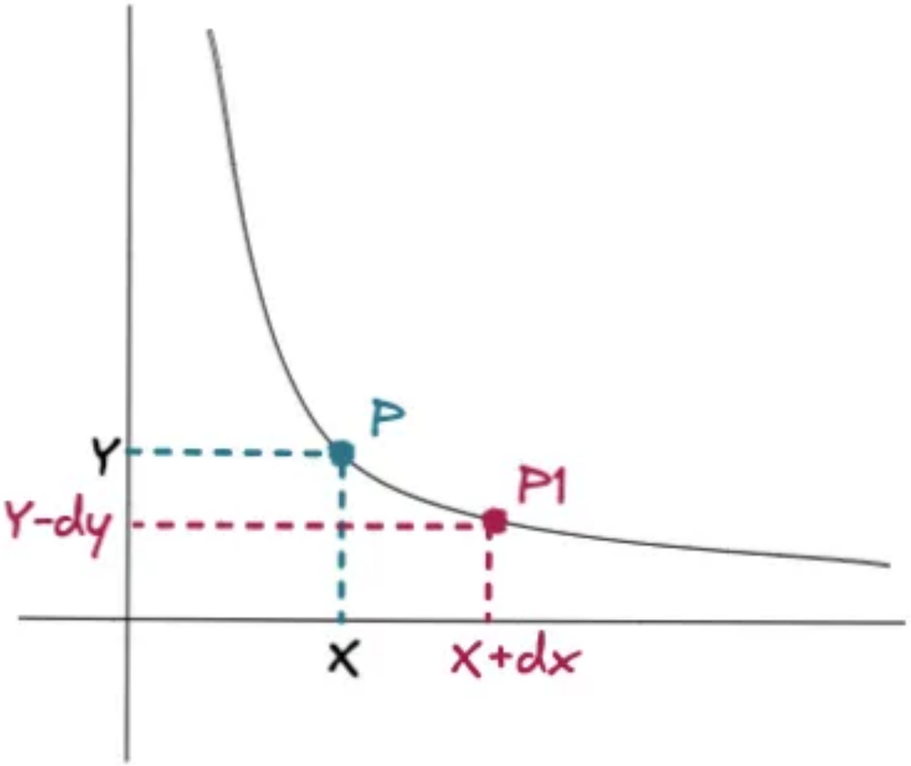
Uniswap Mechanics



How Works-Swap

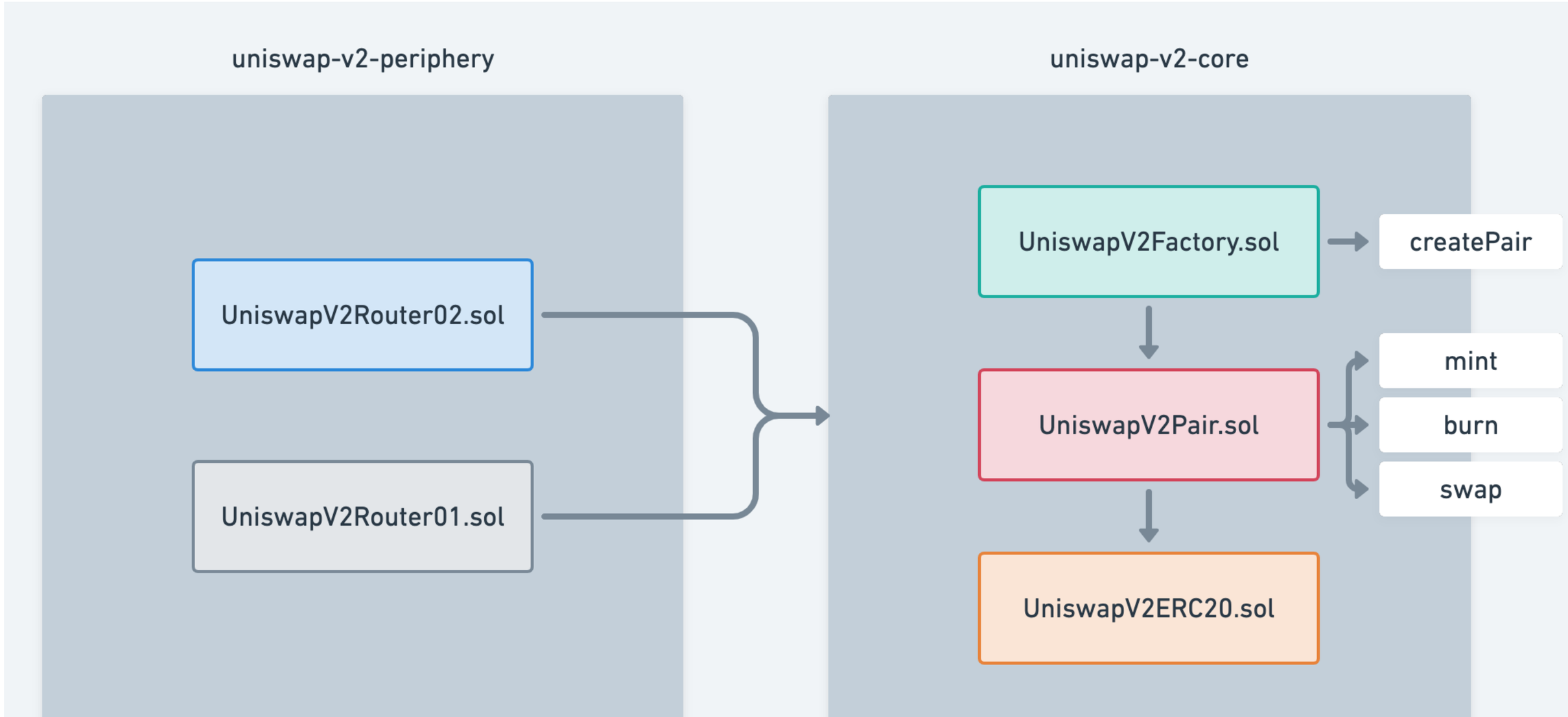
Price Cure: x*y=k

- $x * y = K$
- $p_x = \frac{y}{x}$
- Swap Δx tokenA —> Δy tokenB
  - $x * y = (x + \Delta x) * (y - \Delta y)$
  - $\Delta y = \frac{y * \Delta x}{x + \Delta x}$
  - $\frac{\Delta y}{\Delta x} = \frac{y}{x + \Delta x}$



Contract Structure

UniswapV2 Mechanics



Contract Structure

UniswapV2 Mechanics

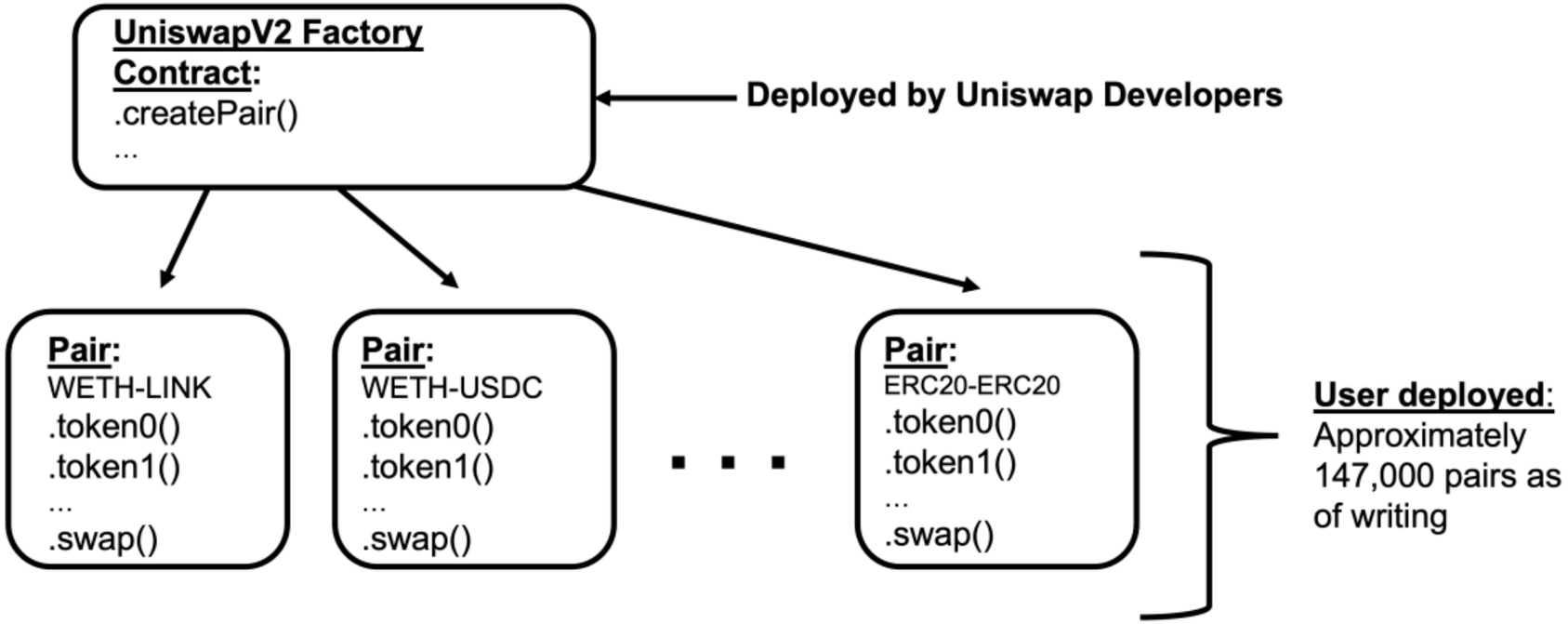


Figure 1: Relationship of UniswapV2 Factory and Token Pair contracts.

Contract: Router

UniswapV2 Mechanics

```
import '@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol';
```

```
function swapExactETHForTokens(uint amountOutMin, address[] calldata path, address to, uint deadline)
    external
    payable
    returns (uint[] memory amounts);
function swapTokensForExactETH(uint amountOut, uint amountInMax, address[] calldata path, address to, uint deadline)
    external
    returns (uint[] memory amounts);
function swapExactTokensForETH(uint amountIn, uint amountOutMin, address[] calldata path, address to, uint deadline)
    external
    returns (uint[] memory amounts);
function swapETHForExactTokens(uint amountOut, address[] calldata path, address to, uint deadline)
    external
    payable
    returns (uint[] memory amounts);
```

Contract: Factory

UniswapV2 Mechanics

```
1   pragma solidity >=0.5.0;
2
3   interface IUniswapV2Factory {
4       event PairCreated(address indexed token0, address indexed token1, address pair, uint);
5
6       function feeTo() external view returns (address);
7       function feeToSetter() external view returns (address);
8
9       function getPair(address tokenA, address tokenB) external view returns (address pair);
10      function allPairs(uint) external view returns (address pair);
11      function allPairsLength() external view returns (uint);
12
13      function createPair(address tokenA, address tokenB) external returns (address pair);
14
15      function setFeeTo(address) external;
16      function setFeeToSetter(address) external;
17  }
```

Contract: Pair

UniswapV2 Mechanics

```
interface IUniswapV2Pair {
    event Approval(address indexed owner, address indexed spender, uint value);
    event Transfer(address indexed from, address indexed to, uint value);

    function name() external pure returns (string memory);
    function symbol() external pure returns (string memory);
    function decimals() external pure returns (uint8);
    function totalSupply() external view returns (uint);
    function balanceOf(address owner) external view returns (uint);
    function allowance(address owner, address spender) external view returns (uint);

    function approve(address spender, uint value) external returns (bool);
    function transfer(address to, uint value) external returns (bool);
    function transferFrom(address from, address to, uint value) external returns (bool);

    function DOMAIN_SEPARATOR() external view returns (bytes32);
    function PERMIT_TYPEHASH() external pure returns (bytes32);
    function nonces(address owner) external view returns (uint);

    function permit(address owner, address spender, uint value, uint deadline,
```

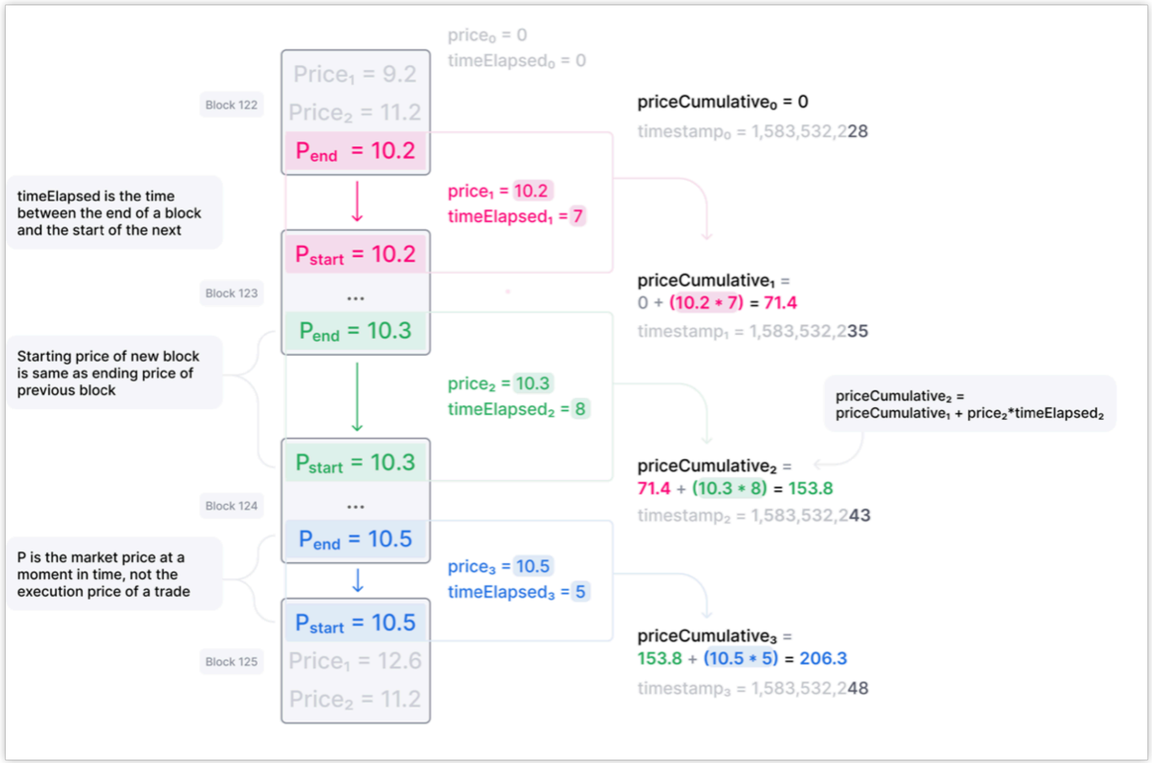
```
function factory() external view returns (address);
function token0() external view returns (address);
function token1() external view returns (address);
function getReserves() external view returns (uint112 reserve0, uint112 reserve1, uint32 blockTimestampLast);
function price0CumulativeLast() external view returns (uint);
function price1CumulativeLast() external view returns (uint);
function kLast() external view returns (uint);

function mint(address to) external returns (uint liquidity);
function burn(address to) external returns (uint amount0, uint amount1);
function swap(uint amount0Out, uint amount1Out, address to, bytes calldata data) external;
function skim(address to) external;
function sync() external;

function initialize(address, address) external;
```

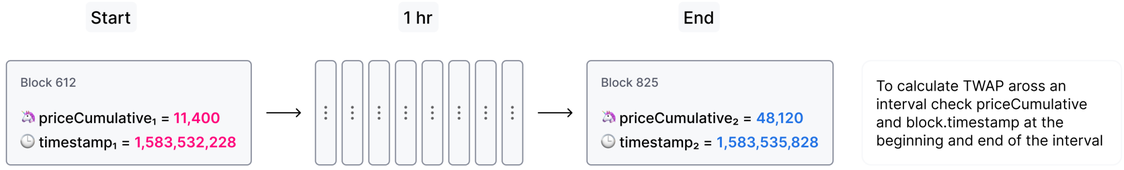
Oracle: TWAP

Uniswap Feature



Uniswap V2

Time-Weighted Average Price using Uniswap V2

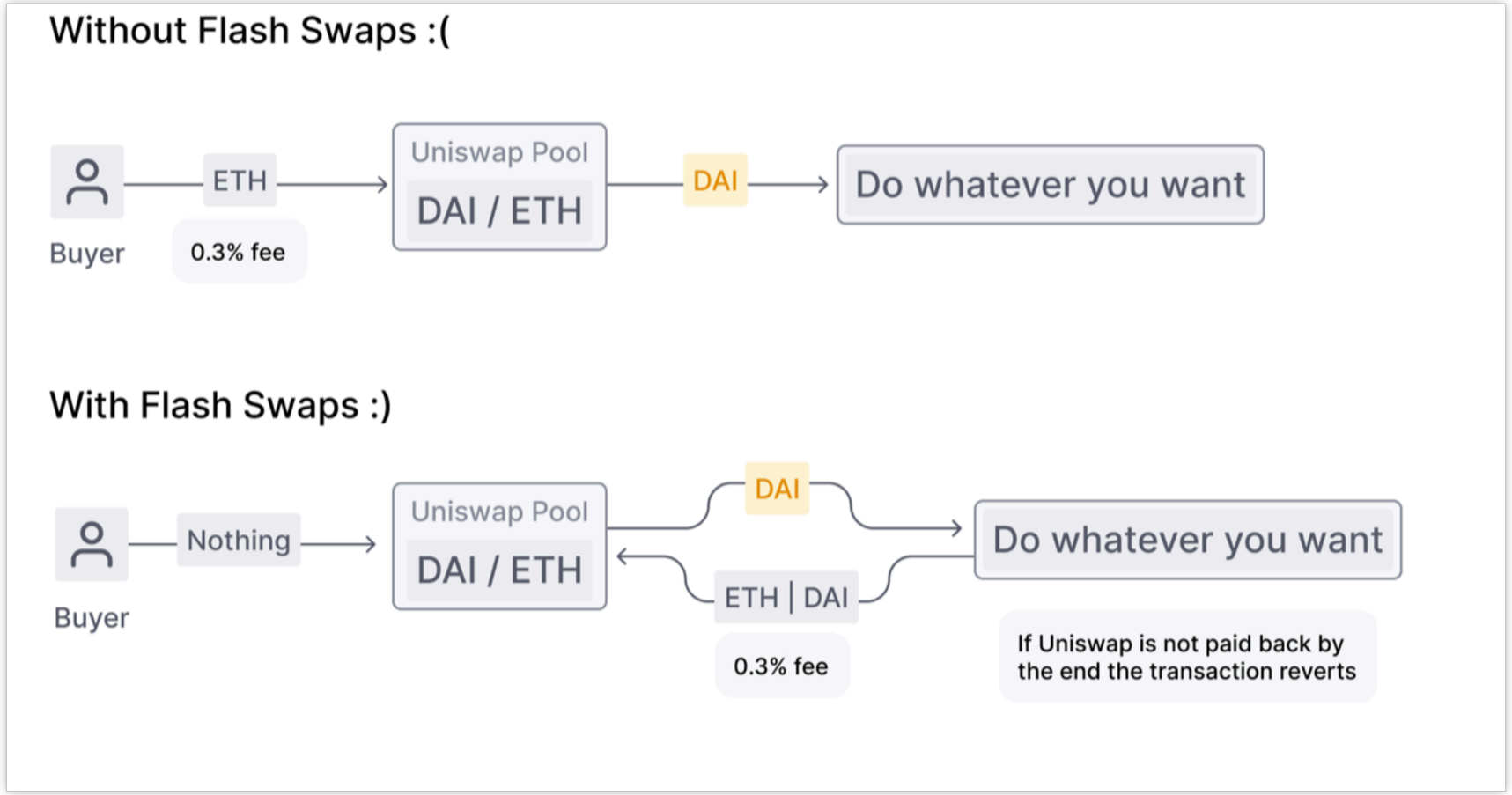


To calculate TWAP aross an interval check priceCumulative and block.timestamp at the beginning and end of the interval

$$\text{TWAP} = \frac{priceCumulative_2 - priceCumulative_1}{timestamp_2 - timestamp_1} = \frac{48{,}120 - 11{,}400}{1{,}583{,}535{,}828 - 1{,}583{,}532{,}228} = 10.2$$

FlashSwap

Uniswap Feature



Use case ?

Exercise

- Deploying Your UniswapV2 Dex (foundry)
- NFTMarket supports the buying of NFTs with any ERC20