DeFi | Academy

Decentralized Exchanges

In Ethereum

By Edson Alcala

Course content

You will learn:

- How decentralized exchanges platforms work
- Different types of decentralized exchanges
- Design considerations and patterns
- How to create a decentralized exchange platform in Ethereum

Pre requisites

You will need

- Introduction to smart contracts course
- Introduction to tokens course

Decentralized exchanges

A quick introduction to decentralized exchanges in Ethereum

What is a Decentralized Exchange?

Traditional Exchanges

- **Binance**
- **Coinbase**
- **Bitfinex**

Traditional Exchanges: Problems

Unfortunate May: BlockFi Suffers Breach, BitMEX Trading Engine Fails

By **Linas Kmieliauskas** • May 19, 2020

BlockFi's Data Breach May Allow Criminals to Extort Rich Clients

BlockFi disclosed a data breach that potentially leaked the physical addresses and account activity of its customers, highlighting the risks of KYC finance platforms.

What is a Decentralized Exchange?

A cryptocurrency exchange that works without any central authority.

Decentralized Exchange characteristics?

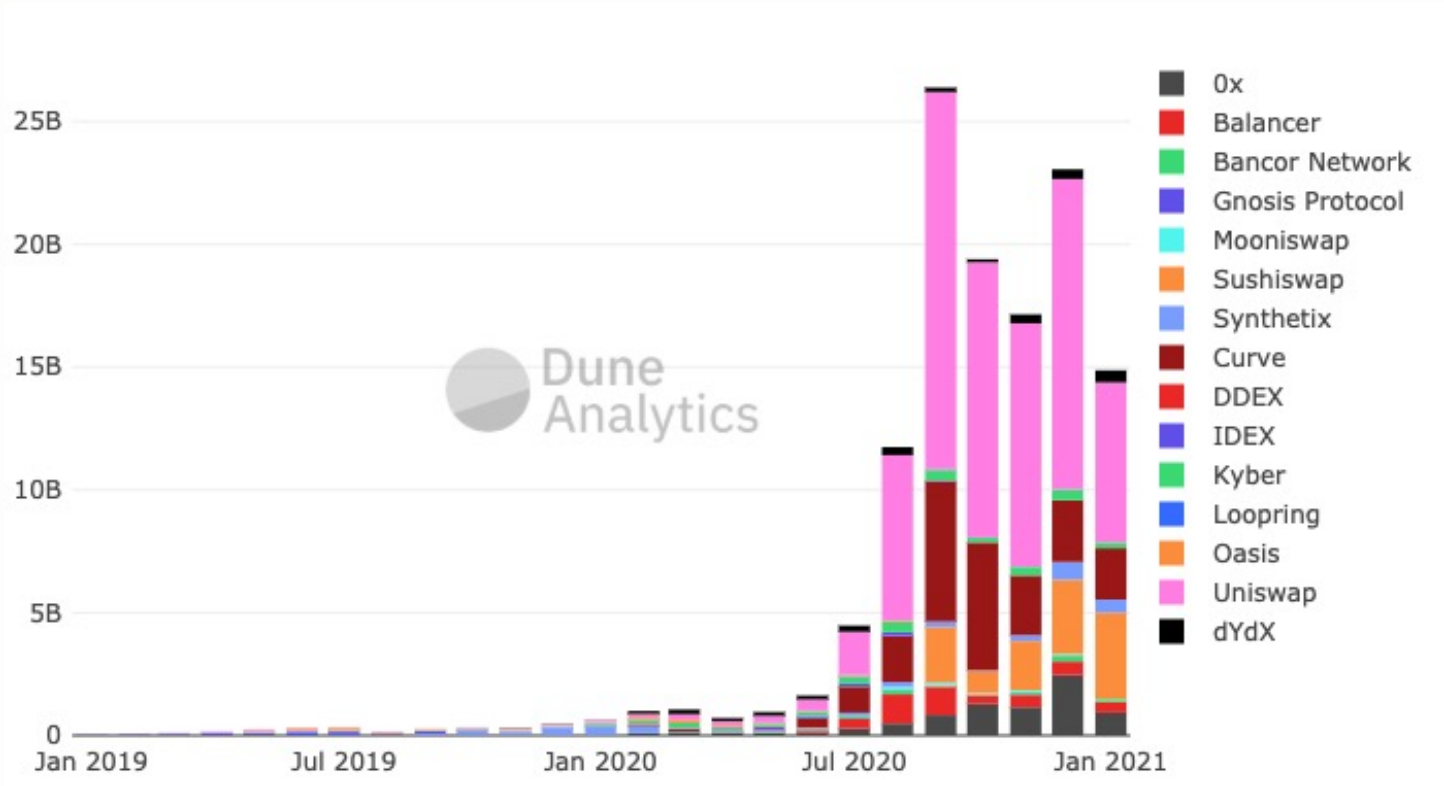
- Non custodial
- Automated
- Globally accessible
- Pseudo anonymous

Most used Decentralized Exchanges

- **Uniswap** - https://uniswap.org
- **Curve** https://www.curve.fi/

- **Aggregators**
  - **Matcha** https://matcha.xyz/
  - **1inch** - https://1inch.exchange

Most used Decentralized Exchanges



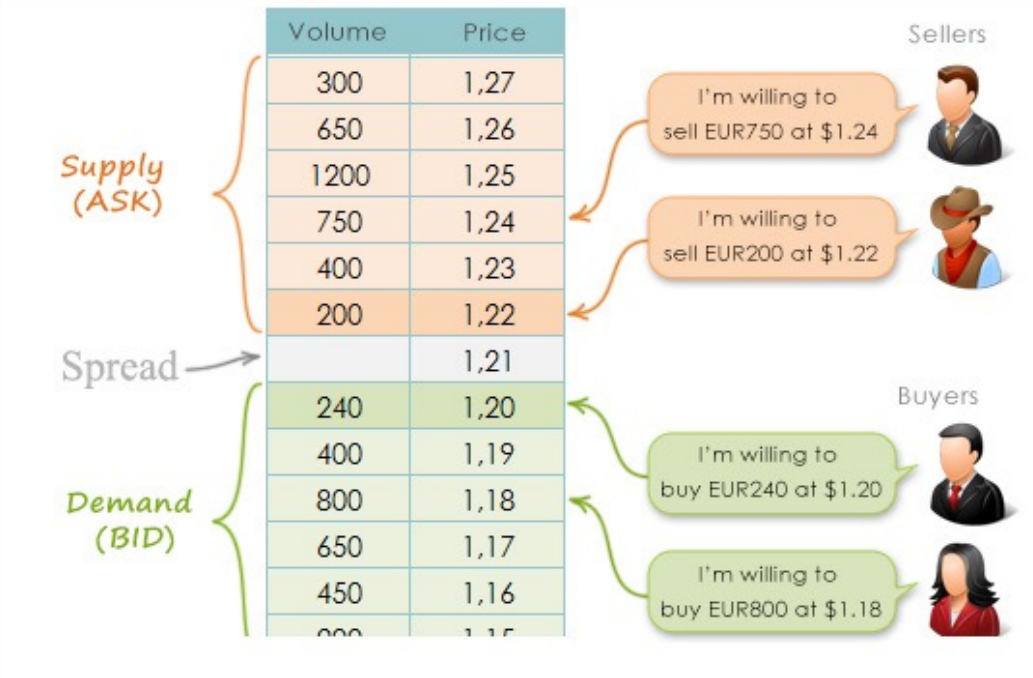
Types of Decentralized Exchanges

- Order books
- Automated market makers
- Dutch auction

Order books

- It is an exchange design
- Mostly used in centralized exchanges
- We keep a list of all open offers from buyers and sellers in the system
- The exchange operator (the central authority) matches orders from buyers and sellers

Order books


Volume
Price
Sellers
300
1,27
650
1,26
1200
1,25
750
1,24
400
1,23
200
1,22
Supply (ASK)
Spread
1,21
240
1,20
400
1,19
800
1,18
650
1,17
450
1,16
Demand (BID)
Buyers
I'm willing to sell EUR750 at $1.24
I'm willing to sell EUR200 at $1.22
I'm willing to buy EUR240 at $1.20
I'm willing to buy EUR800 at $1.18

Order books


8,229.99 USD Last trade price
+0.24% 24h price
11,244 BTC 24h volume
Order Book
Order Book
Trade History
Market Size
Price (USD)
My Size
0.0019 8230.49 -
0.0090 8230.02 -
0.0012 8230.01 -
6.8006 8230.00 -
USD Spread 0.01
21.9712 8229.99 -
0.1216 8229.66 -
0.5000 8229.00 -
0.0200 8228.99 -
Aggregation 0.01
Price Charts
5m
Candle
Overlay
O: 8,269.99 H: 8,270 L: 8,263.48 C: 8,263.49 V: 6
$8,400
$8,200
$8,000
11 AM
12:05pm
1 PM
2 PM
3 PM
8,229.995
Mid Market Price
$8,110
$8,150
$8,190
$8,230
$8,270
$8,310
$8,350
Open Orders
Open
Fills
Size
Filled (BTC)
Price (USD)
Fee (USD)
Status

AAM: Automated Market Maker

Exchange design that pools liquidity reserves and makes markets according to a deterministic algorithm.

Users can trade against these reserves at **prices set by an automated market making formula.**

Example:

- Uniswap
- Curve
- Balancer

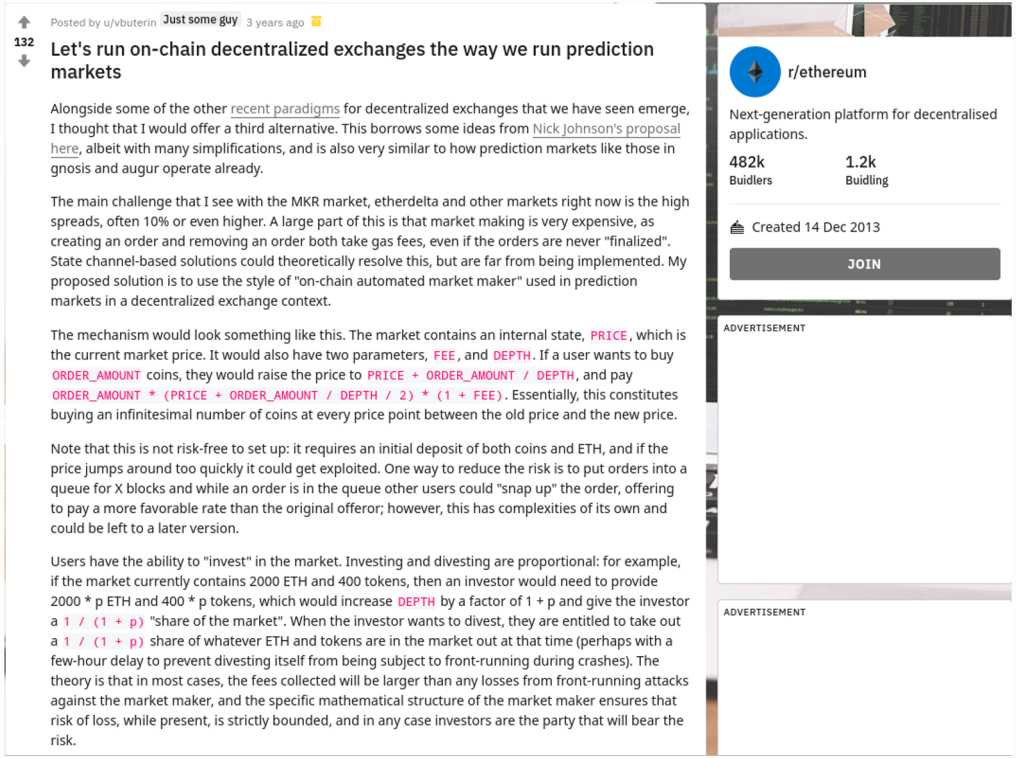
Uniswap

Automated Liquidity Protocol

What is Uniswap?

Uniswap is a fully decentralized protocol for automated liquidity provision that allow users to swap tokens on Ethereum.

Background: Vitalik Post



Posted by u/vbuterin Just some guy 3 years ago

132

Let's run on-chain decentralized exchanges the way we run prediction markets

Alongside some of the other recent paradigms for decentralized exchanges that we have seen emerge, I thought that I would offer a third alternative. This borrows some ideas from Nick Johnson's proposal here, albeit with many simplifications, and is also very similar to how prediction markets like those in gnosis and augur operate already.

The main challenge that I see with the MKR market, etherdelta and other markets right now is the high spreads, often 10% or even higher. A large part of this is that market making is very expensive, as creating an order and removing an order both take gas fees, even if the orders are never "finalized". State channel-based solutions could theoretically resolve this, but are far from being implemented. My proposed solution is to use the style of "on-chain automated market maker" used in prediction markets in a decentralized exchange context.

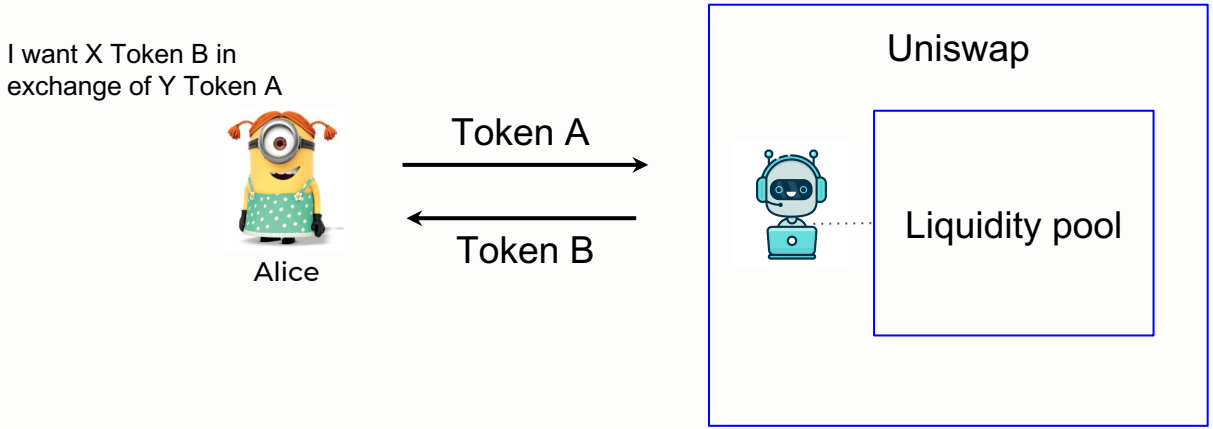
The mechanism would look something like this. The market contains an internal state, `PRICE`, which is the current market price. It would also have two parameters, `FEE`, and `DEPTH`. If a user wants to buy `ORDER_AMOUNT` coins, they would raise the price to `PRICE + ORDER_AMOUNT / DEPTH`, and pay `ORDER_AMOUNT * (PRICE + ORDER_AMOUNT / DEPTH / 2) * (1 + FEE)`. Essentially, this constitutes buying an infinitesimal number of coins at every price point between the old price and the new price.

Note that this is not risk-free to set up: it requires an initial deposit of both coins and ETH, and if the price jumps around too quickly it could get exploited. One way to reduce the risk is to put orders into a queue for X blocks and while an order is in the queue other users could "snap up" the order, offering to pay a more favorable rate than the original offeror; however, this has complexities of its own and could be left to a later version.

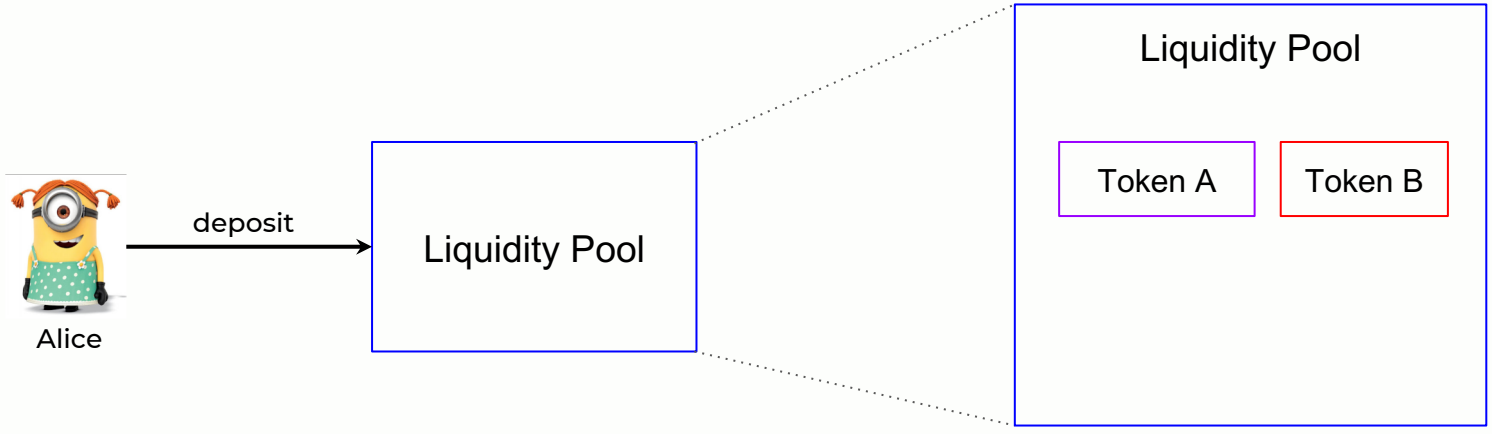
Users have the ability to "invest" in the market. Investing and divesting are proportional: for example, if the market currently contains 2000 ETH and 400 tokens, then an investor would need to provide 2000 * p ETH and 400 * p tokens, which would increase `DEPTH` by a factor of 1 + p and give the investor a `1 / (1 + p)` "share of the market". When the investor wants to divest, they are entitled to take out a `1 / (1 + p)` share of whatever ETH and tokens are in the market out at that time (perhaps with a few-hour delay to prevent divesting itself from being subject to front-running during crashes). The theory is that in most cases, the fees collected will be larger than any losses from front-running attacks against the market maker, and the specific mathematical structure of the market maker ensures that risk of loss, while present, is strictly bounded, and in any case investors are the party that will bear the risk.

https://www.reddit.com/r/ethereum/comments/55m04x/lets_run_onchain_decentralized_exchanges_the_way/

AMM

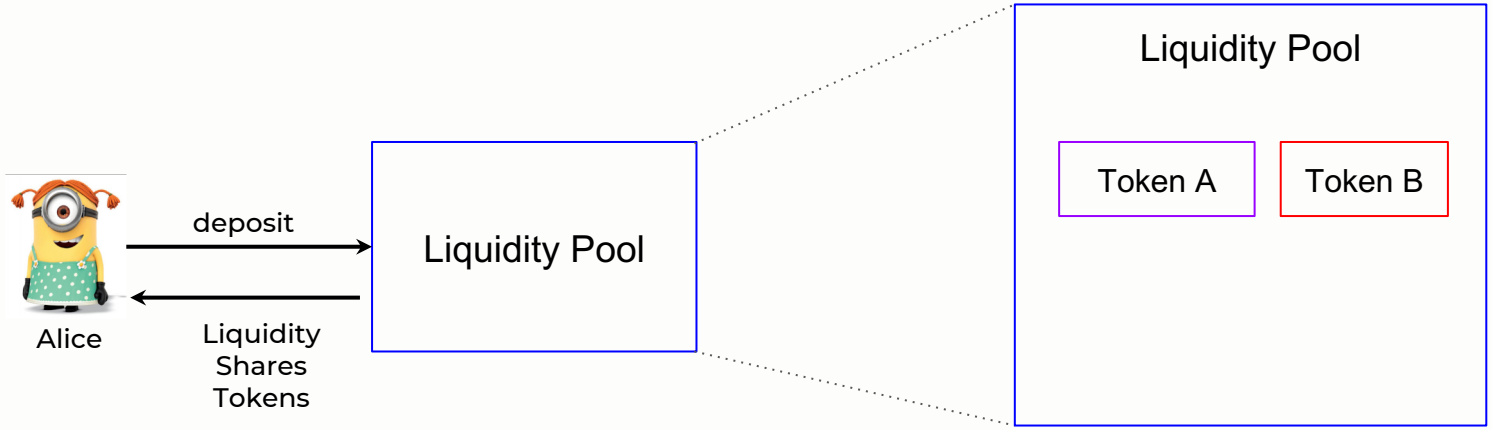

I want X Token B in
exchange of Y Token A
Alice
Token A
Token B
Uniswap
Liquidity pool

How it works



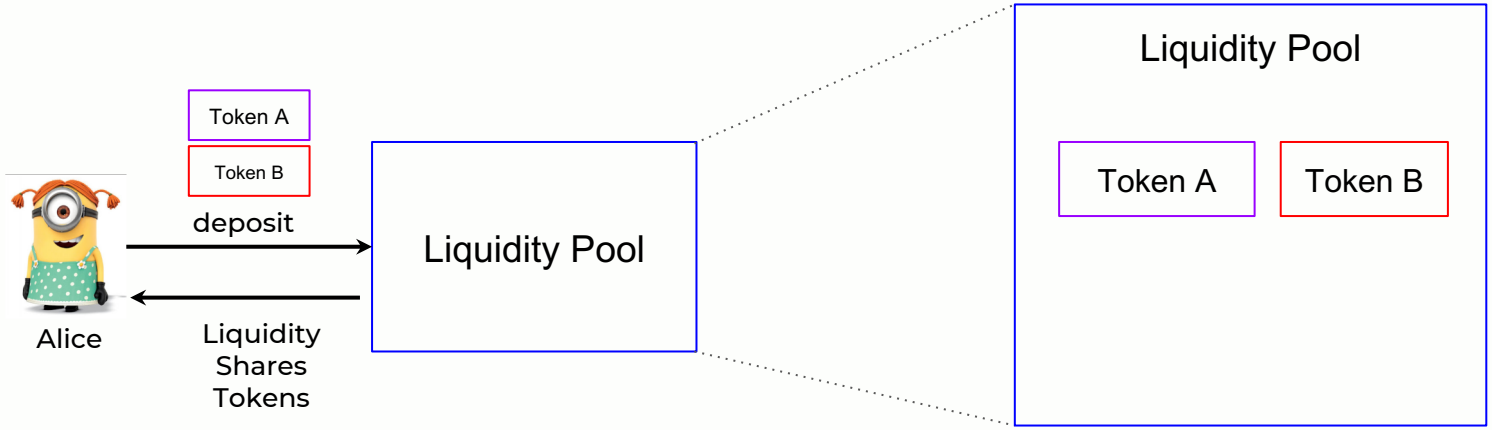
Liquidity Pool is just a smart contract that keep balances of two unique pair of tokens

How it works



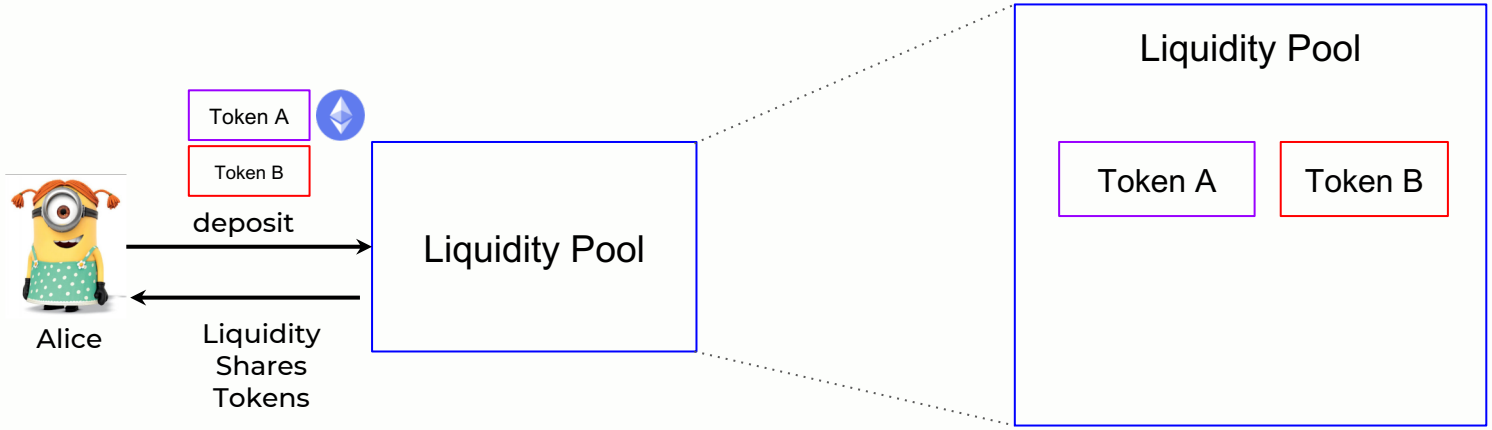
Users who deposit to the Liquidity Pool are called **Liquidity Providers.**

How it works



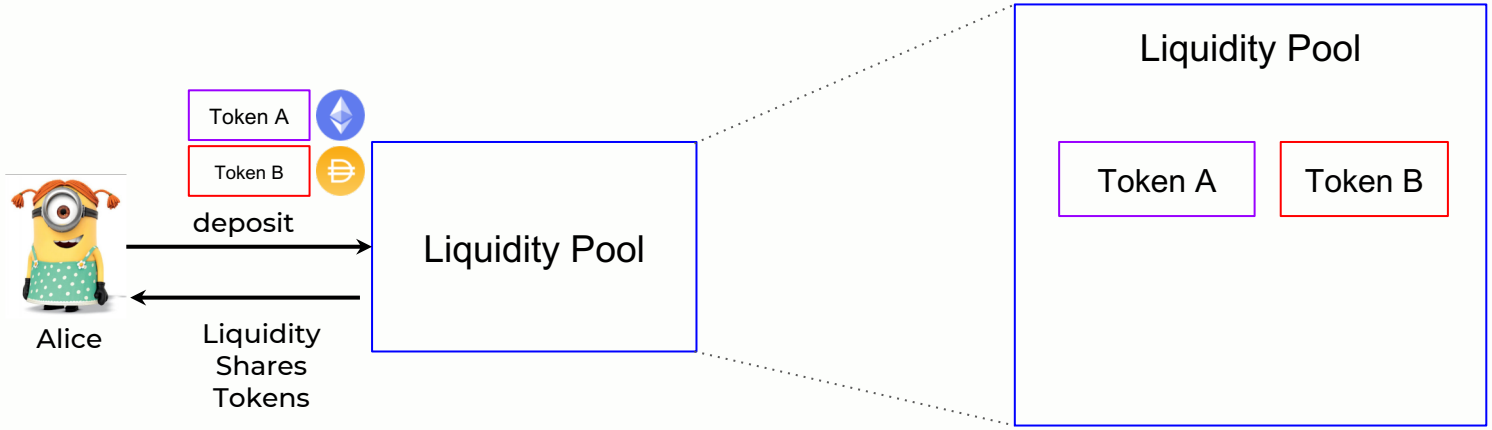
Users who deposit to the Liquidity Pool are called **Liquidity Providers.**

How it works



Users who deposit to the Liquidity Pool are called **Liquidity Providers.**

How it works



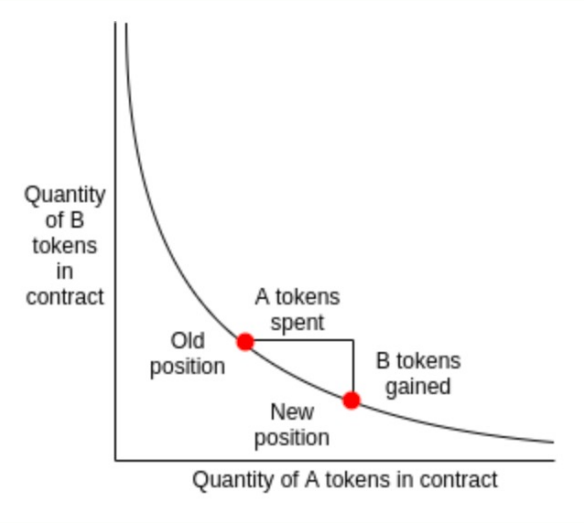
Users who deposit to the Liquidity Pool are called **Liquidity Providers.**

How it works


Uniswap Pool
Reserves
100
Token A
10
Token B
Liquidity Shares
12
Pool Tokens
Liquidity Provider
Deposit
Input:
10 Token A
+
1 Token B
Output
4 LP Shares
Trader
Swap
Input:
10 Token A
+
0.03 % Fee
Output
1 Token B

Constant product formula

X * Y = K



Constant product formula: Trade

A trade must satisfy $$\boldsymbol{X} * \boldsymbol{Y} = \boldsymbol{K}$$

Note: Considering a fee of 0

Constant product formula: Trade

A trade must satisfy $$X * Y = K$$

$$if\ X = R_{TA}\ ;\ Y = R_{TB}$$

Note: Considering a fee of 0

Constant product formula: Trade

A trade must satisfy $$X * Y = K$$

$$if\ X = R_{TA}\ ;\ Y = R_{TB}$$

Where

R_{TA} are the reserves of the Token A

R_{TB} are the reserves of the Token B

Note: Considering a fee of 0

Constant product formula: Trade

A trade must satisfy $$X * Y = K$$

$$if\ X = R_{TA}\ ;\ Y = R_{TB}$$

Where

R_{TA} are the reserves of the Token A

R_{TB} are the reserves of the Token B

$$R_{TA} * R_{TB} = K$$

Note: Considering a fee of 0

Constant product formula: Trade

When we trade, we want to get Δ tokens.

Either Δ_{TA} in exchange of Δ_{TB} or vice versa

Note: Considering a fee of 0

Constant product formula: Trade

When we trade, we want to get Δ tokens.

Either Δ_{TA} in exchange of Δ_{TB} or vice versa

For example.

I want to get Δ_{TB} by putting Δ_{TA}

Note: Considering a fee of 0

Constant product formula: Trade

When we trade, we want to get Δ tokens.

Either Δ_{TA} in exchange of Δ_{TB} or vice versa

For example.

I want to get Δ_{TB} by putting Δ_{TA}

Then we have:

$$(\boldsymbol{R_{TA}} + \Delta_{TA})\,(\boldsymbol{R_{TB}} - \Delta_{TB}) = \boldsymbol{K = R_{TA} * R_{TB}}$$

Note: Considering a fee of 0

Constant product formula: Number of Tokens

Following past example: I want to get Δ_{TB} by putting Δ_{TA}

How do we calculate the number of tokens?

$$(R_{TA} + \Delta_{TA})\,(R_{TB} - \Delta_{TB}) = R_{TA} * R_{TB}$$

Note: Considering a fee of 0

Constant product formula: Number of Tokens

Following past example: I want to get Δ_{TB} by putting Δ_{TA}

How do we calculate the number of tokens?

$$(R_{TA} + \Delta_{TA})\,(R_{TB} - \Delta_{TB}) = R_{TA} * R_{TB}$$

Note: Considering a fee of 0

Constant product formula: Number of Tokens

Following past example: I want to get Δ_{TB} by putting Δ_{TA}

How do we calculate the number of tokens?

$$(R_{TA} + \Delta_{TA})\,(R_{TB} - \Delta_{TB}) = R_{TA} * R_{TB}$$

$$\frac{1}{(R_{TA} + \Delta_{TA})}(R_{TA} + \Delta_{TA})\,(R_{TB} - \Delta_{TB}) = \frac{1}{(R_{TA} + \Delta_{TA})}\,R_{TA} * R_{TB}$$

Note: Considering a fee of 0

Constant product formula: Number of Tokens

Following past example: I want to get Δ_{TB} by putting Δ_{TA}

How do we calculate the number of tokens?

$$(R_{TA} + \Delta_{TA})\,(R_{TB} - \Delta_{TB}) = R_{TA} * R_{TB}$$

$$\frac{1}{(R_{TA} + \Delta_{TA})}(R_{TA} + \Delta_{TA})\,(R_{TB} - \Delta_{TB}) = \frac{1}{(R_{TA} + \Delta_{TA})}\,R_{TA} * R_{TB}$$

$$(R_{TB} - \Delta_{TB}) = \frac{R_{TA} * R_{TB}}{(R_{TA} + \Delta_{TA})}$$

Note: Considering a fee of 0

Constant product formula: Number of Tokens

Following past example: I want to get Δ_{TB} by putting Δ_{TA}

How do we calculate the number of tokens?

$$(R_{TA} + \Delta_{TA})(R_{TB} - \Delta_{TB}) = R_{TA} * R_{TB}$$

$$\frac{1}{(R_{TA} + \Delta_{TA})}(R_{TA} + \Delta_{TA})(R_{TB} - \Delta_{TB}) = \frac{1}{(R_{TA} + \Delta_{TA})} R_{TA} * R_{TB}$$

$$(R_{TB} - \Delta_{TB}) = \frac{R_{TA} * R_{TB}}{(R_{TA} + \Delta_{TA})}$$

$$\Delta_{TB} = R_{TB} - \frac{R_{TA} * R_{TB}}{(R_{TA} + \Delta_{TA})}$$

Note: Considering a fee of 0

Constant product formula: Number of Tokens

Following past example: I want to get Δ_{TB} by putting Δ_{TA}

How do we calculate the number of tokens?

$$(R_{TA} + \Delta_{TA})(R_{TB} - \Delta_{TB}) = R_{TA} * R_{TB}$$

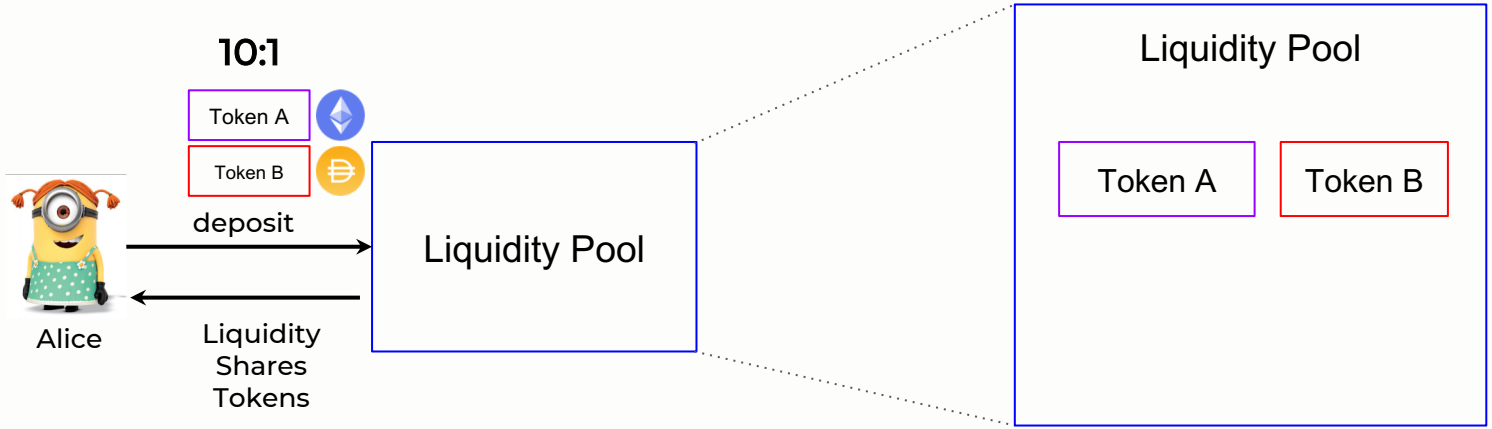
$$\frac{1}{(R_{TA} + \Delta_{TA})}(R_{TA} + \Delta_{TA})(R_{TB} - \Delta_{TB}) = \frac{1}{(R_{TA} + \Delta_{TA})} R_{TA} * R_{TB}$$

$$(R_{TB} - \Delta_{TB}) = \frac{R_{TA} * R_{TB}}{(R_{TA} + \Delta_{TA})}$$

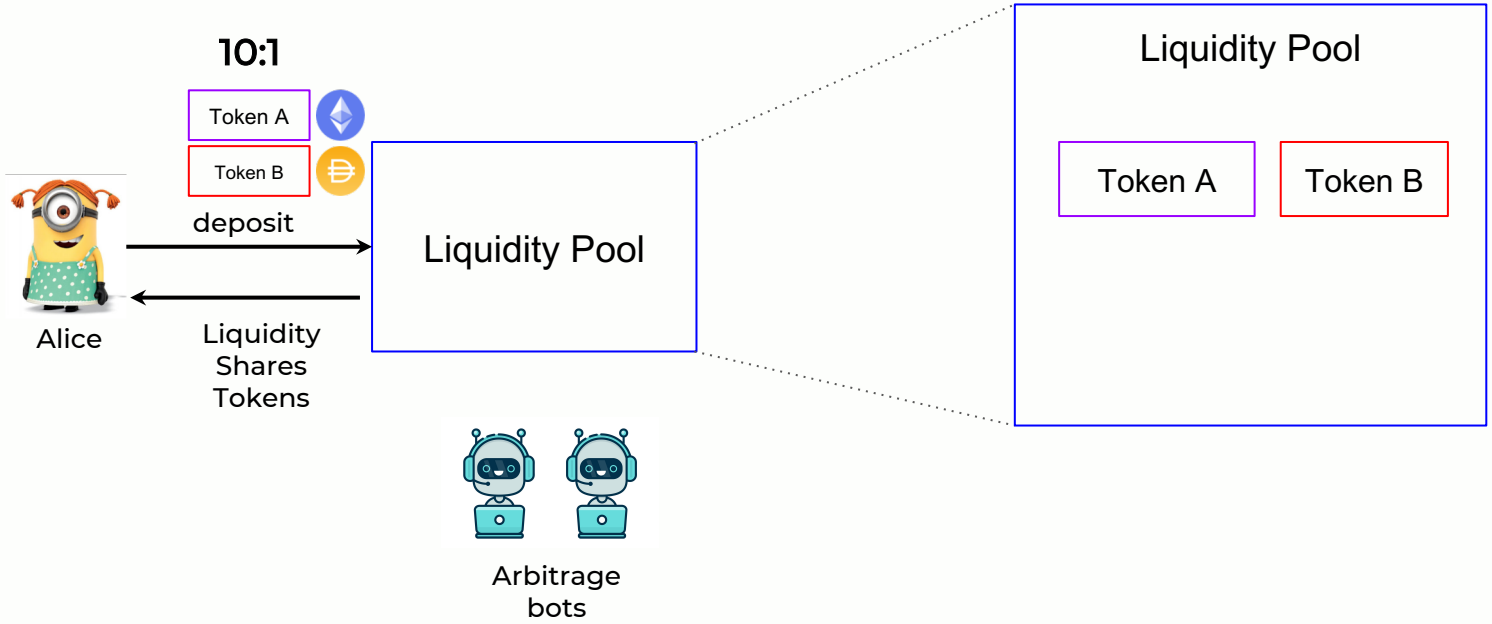
$$\Delta_{TB} = R_{TB} - \frac{R_{TA} * R_{TB}}{(R_{TA} + \Delta_{TA})} \quad \rightarrow \quad \Delta_{TB} = \frac{\Delta_{TA} * R_{TB}}{(R_{TA} + \Delta_{TA})}$$

Note: Considering a fee of 0

How pricing works



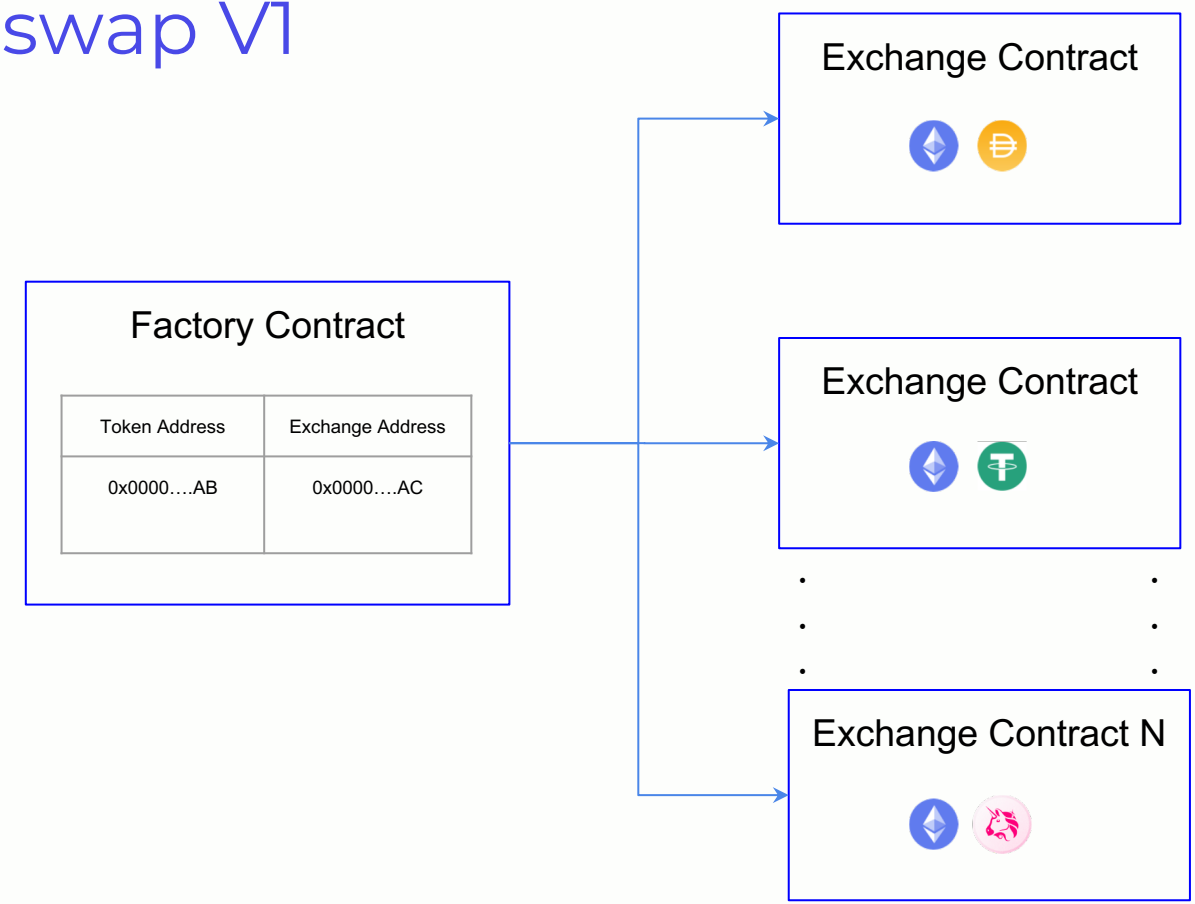
How pricing works



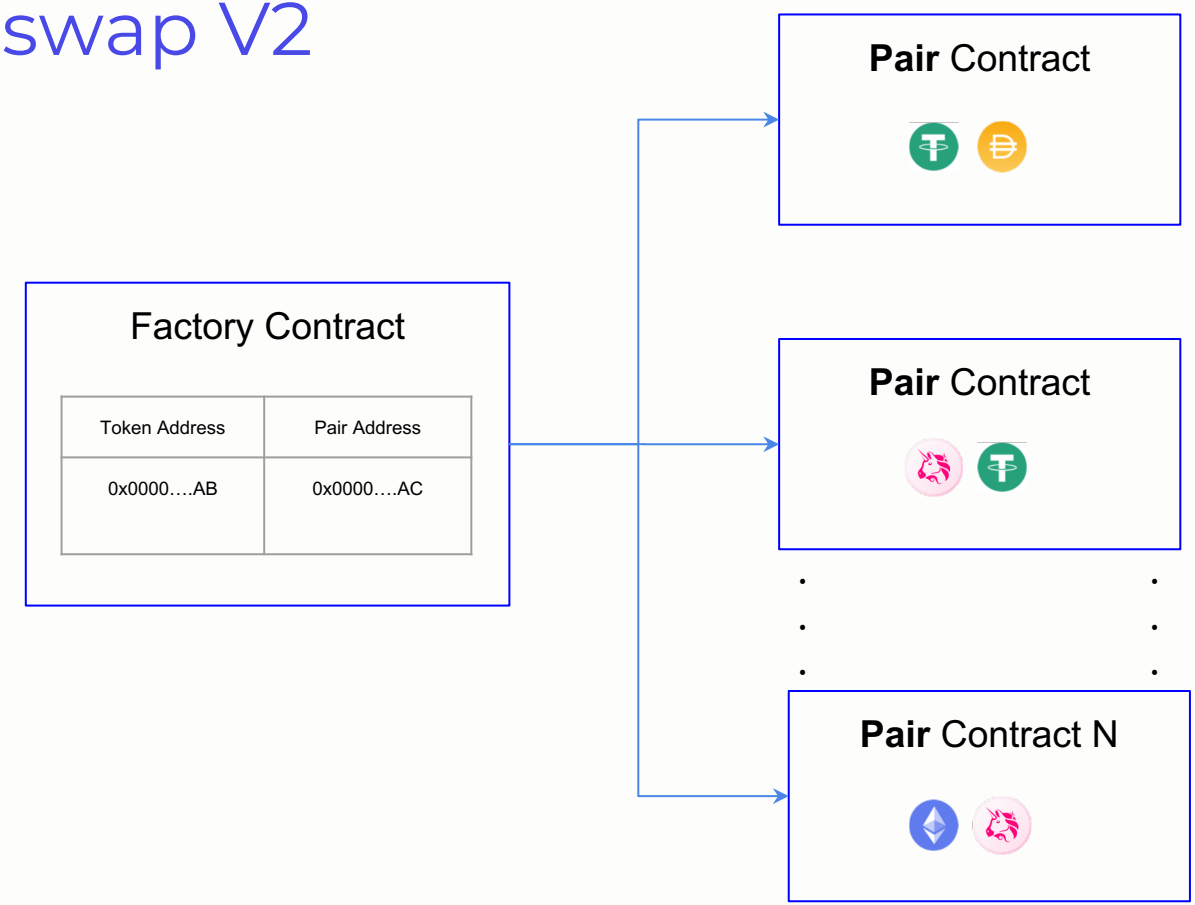
Uniswap Architecture

- Uniswap V1
- Uniswap V2

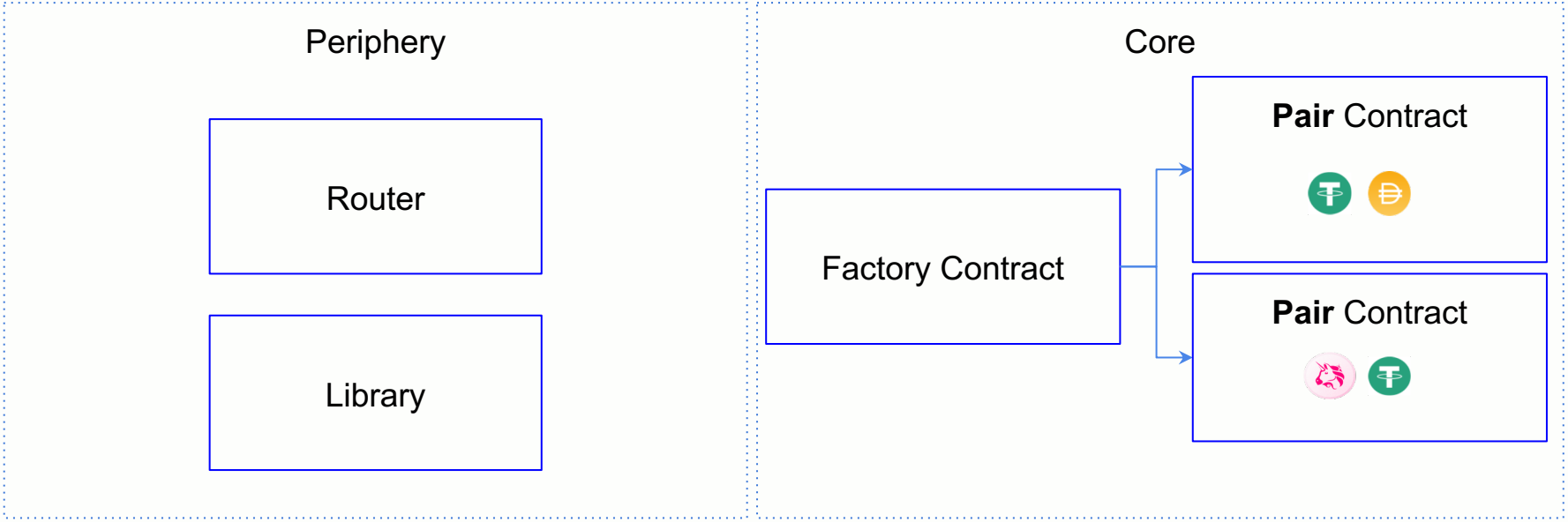
Uniswap V1


Exchange Contract
Factory Contract
Token Address
Exchange Address
0x0000….AB
0x0000….AC
Exchange Contract
Exchange Contract N

Uniswap V2


Pair Contract
Factory Contract
Token Address
Pair Address
0x0000….AB
0x0000….AC
Pair Contract
Pair Contract N

Smart contract architecture

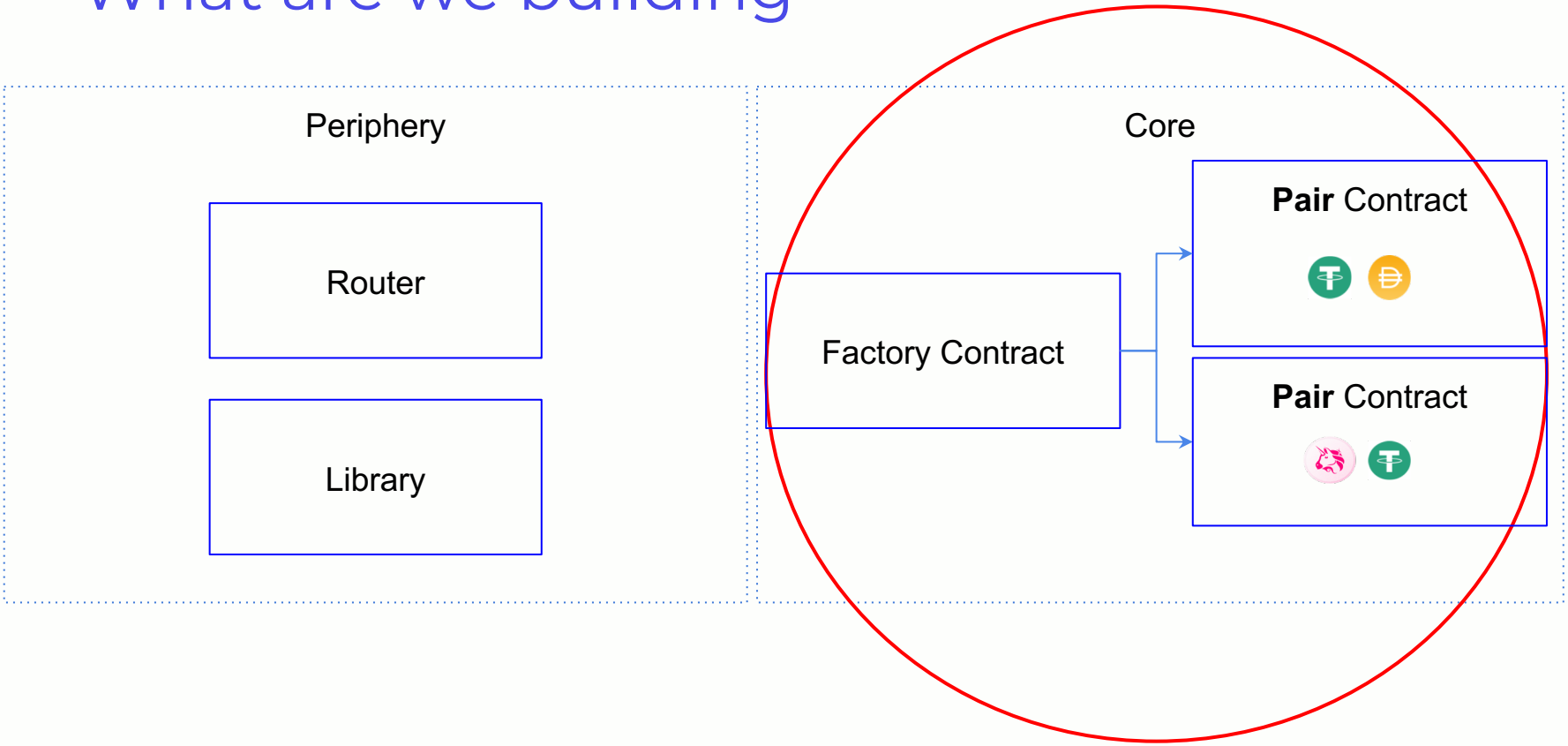


Creating a decentralized exchange protocol

Design considerations

- AMM or Order book?
- What type of tokens are we going to support? ERC20s? ERC721s?
- Are we charging a fee per trade? How much?
- If AMM, how many tokens / percentage are we giving to Liquidity providers?
- If AMM, who will decide which tokens to deposit on the liquidity pool?
- How is the price going to be resolved?

What are we building

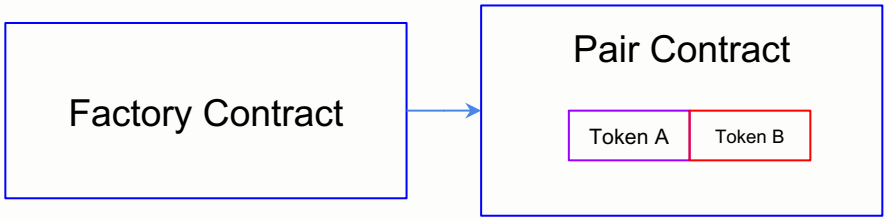


What are we building

- 0 transaction fees
- 1 LP Token per unit of token provided to the liquidity pool

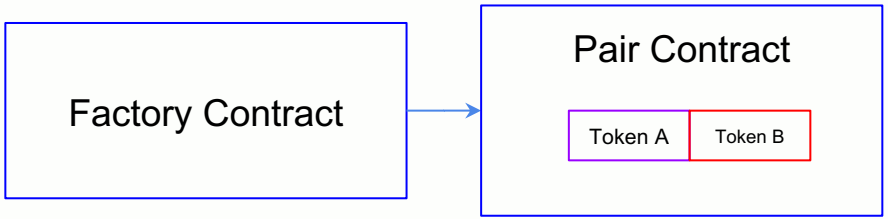
Model smart contracts

- Token A
- Token B
- LP Token
- Pair
- Factory



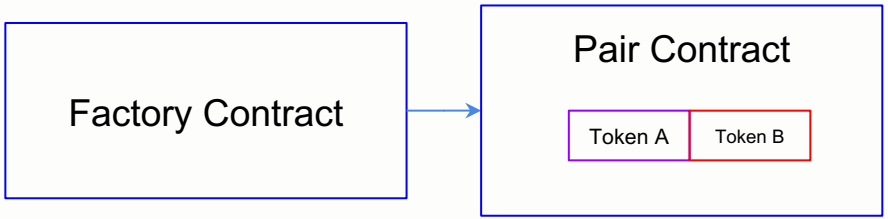
Coding Tokens

- Token A
- Token B
- LP Token
- Pair
- Factory



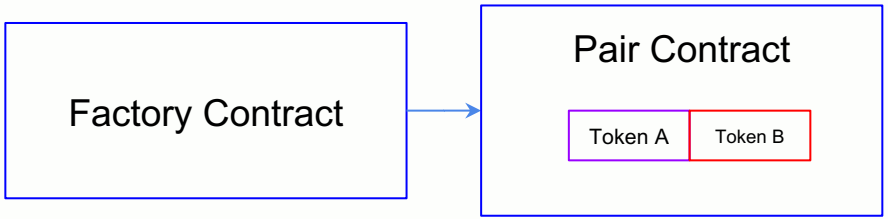
Model Pair contract

- Token A
- Token B
- LP Token
- Pair
- Factory



Model Factory contract

- Token A
- Token B
- LP Token
- Pair
- Factory



Conclusion

- We learned about decentralized exchanges
- Different types of decentralized exchanges
- We analysed and learned how the Uniswap protocol works
- We learned how to create a decentralized exchange in Ethereum

Thanks!

Do you have any questions?

hello@defi-academy.com

Or in our Discord Group

https://discord.gg/xyF9xVTSkY