Decentralized Exchanges (DEXs)

Ali Yahya
Why try to "decentralize" an exchange?

• Composability (brief rant)
• Credibly Neutral
• Security
• Global Reach
What is a DEX?

A decentralized exchange (or DEX) is an online marketplace where transactions occur directly between participants, without the aid of any trusted intermediaries.

Key Properties

• Composable / Programmable
• Credibly Neutral
• Non-Custodial
• Permissionless
First Approach: Order Book Based DEXs
Order Book Based DEXs

The Relayer Model

• Matching is done off-chain by a centralized “Relayer”
  • The relayer crafts a transaction off-chain that resembles an atomic-swap, then submits it to the blockchain
• Trade settlement is done on-chain

Many examples of DEXs that initially worked this way:
• 0x protocol
• EtherDelta
• Kyber
• Airswap
Order Book Based DEXs

Limitations of the Relayer Model

• Less programmable/composable
• Depends on the presence of a centralized party
• Peer-to-peer—hard to bootstrap liquidity
• It's expensive with today's blockchains because of gas

Great resource:
Front-Running, Griefing, and the Perils of Virtual Settlement, by Will Warren
Is there a simpler way to build a DEX?
A Bit of History: Automated Market Makers (AMMs)

- Pricing shares in prediction markets — Hanson’s Market Scoring Rules
  - Also used to price online ads
- Idea first explored in crypto in 2016 by:
  - Vitalik Buterin — [reddit post](#)
- Then generalized by Alan Lu and Martin Koppelman:
  - Blogpost: [Building a Decentralized Exchange in Ethereum](#)
High Level Aspiration

Two-Sided Marketplace

Liquidity Providers

Traders

Smart Contract

ETH: 10
DAI: 12
$xy = k$

![Graph showing the relationship between ETH and DAI with $xy = k$.]

 Trader

Liquidity Providers

4.0 ETH

0.012 DAI

8.0 DAI

Smart Contract

10.0 ETH $\times$ 20.0 DAI $= 120$
Uniswap V2

Demo: app.uniswap.org
\[ (x - \Delta x)(y + \Delta y) = k \]
\[ xy = k \]

**Simple Pricing Rule**

\[(x - \Delta x)(y + \phi \Delta y) = k\]

where \((1 - \phi)\) is the percentage fee that is paid to liquidity providers, and where \(\Delta x > 0\) and \(\Delta y > 0\).
\( xy = k \)

**Simple Pricing Rule**

\[(x - \Delta x)(y + \phi \Delta y) = k\]

\[\phi \Delta y = \frac{xy}{x - \Delta x} - y\]

\[= \frac{xy - y(x - \Delta x)}{x - \Delta x}\]

\[= \frac{xy - xy - y\Delta x}{x - \Delta x}\]

\[\Delta y = \frac{1}{\phi} \cdot \frac{y\Delta x}{x - \Delta x}\]
Simple Pricing Rule

\[ \Delta y = \frac{1}{\phi} \cdot \frac{y \Delta x}{x - \Delta x} \]

This rule specifies the price of buying \( \Delta x \) in terms of \( y \).

A similar exercise (swapping \( x \)s and \( y \)s) produces a rule that specifies the price of selling \( \Delta x \) in terms of \( y \):

\[ \Delta y = \frac{y \phi \Delta x}{x + \phi \Delta x} \]
Example where the contract contains 4.0 ETH and 30.0 DAI and charges a fee for liquidity providers of 30 bps.

\[ \Delta y = \frac{y \phi \Delta x}{x + \phi \Delta x} \]

\[ \Delta y = \frac{30 \times 0.997 \times \Delta x}{4 + 0.997 \times \Delta x} \]

Say a trader wants to sell 8.0 ETH to the contract. How much DAI should she get in return?

\[ \Delta y = \frac{30 \times 0.997 \times 8}{4 + 0.997 \times 8} = 19.98 \]

(The fee to liquidity providers is 0.02.)
In the Wild: Uniswap

Selling $x$ for $y$

$$\Delta y = \frac{y \phi \Delta x}{x + \phi \Delta x}$$

Buying $x$ for $y$

$$\Delta y = \frac{1}{\phi} \cdot \frac{y \Delta x}{x - \Delta x}$$

UniswapV2Library.sol

```solidity
// given an input amount of an asset and pair reserves, returns the maximum output amount of the other asset
function getAmountOut(uint amountIn, uint reserveIn, uint reserveOut) internal pure returns (uint amountOut) {
    require(amountIn > 0, 'UniswapV2Library: INSUFFICIENT_INPUT_AMOUNT');
    require(reserveIn > 0 && reserveOut > 0, 'UniswapV2Library: INSUFFICIENT LIQUIDITY');
    uint amountInWithFee = amountIn.mul(997);
    uint numerator = amountInWithFee.mul(reserveOut);
    uint denominator = reserveIn.mul(1000).add(amountInWithFee);
    amountOut = numerator / denominator;
}
```

```solidity
// given an output amount of an asset and pair reserves, returns a required input amount of the other asset
function getAmountIn(uint amountOut, uint reserveIn, uint reserveOut) internal pure returns (uint amountIn) {
    require(amountOut > 0, 'UniswapV2Library: INSUFFICIENT_OUTPUT_AMOUNT');
    require(reserveIn > 0 && reserveOut > 0, 'UniswapV2Library: INSUFFICIENT LIQUIDITY');
    uint numerator = reserveIn.mul(amountOut).mul(1000);
    uint denominator = reserveOut.sub(amountOut).mul(997);
    amountIn = (numerator / denominator).add(1);
}
```
Quick Demo: https://app.uniswap.org/
How to Think about an AMM’s Price

Price is the ratio between assets (e.g. **DAI**) paid and assets (e.g. **ETH**) received.

If I pay 100 **DAI** for 4 **ETH**, then my price per ETH is 25 **DAI**.

In our notation, this is given by |Δy/Δx|.

---

**Selling x for y**

\[ Δy = \frac{yφΔx}{x + φΔx} \]

**Buying x for y**

\[ Δy = \frac{1}{φ} \cdot \frac{yΔx}{x - Δx} \]

Divide both sides by Δx to get Δy/Δx.

\[ \frac{Δy}{Δx} = \frac{yφ}{x + φΔx} \]

\[ \frac{Δy}{Δx} = \frac{1}{φ} \cdot \frac{y}{x - Δx} \]
\[ xy = k \]

**Marginal Price & Slippage**

\[
\frac{\Delta y}{\Delta x} = \frac{y\phi}{x + \phi\Delta x} \quad \text{Selling } x \text{ for } y
\]

\[
\frac{\Delta y}{\Delta x} = \frac{1}{\phi} \cdot \frac{y}{x - \Delta x} \quad \text{Buying } x \text{ for } y
\]

**Observation #1**

Pricing depends on the size of the trade, \( \Delta x \).

For example with **20.0 ETH** * **6.0 DAI** = **120**,  

Buying **10 ETH** (i.e. \( \Delta x = 10 \)) costs **6.02 DAI**  

Or **0.602 DAI** per ETH  

Whereas buying **5 ETH** costs **2.006 DAI**  

Or **0.401 DAI** per ETH

* assuming \( \phi = 0.997 * a16z
$xy = k$

### Marginal Price & Slippage

**Selling $x$ for $y$**

$$
\frac{\Delta y}{\Delta x} = \frac{y\phi}{x + \phi \Delta x}
$$

**Buying $x$ for $y$**

$$
\frac{\Delta y}{\Delta x} = \frac{1}{\phi} \cdot \frac{y}{x - \Delta x}
$$

In the limit, as $\Delta x$ approaches 0:

$$
\lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \phi \frac{y}{x}
$$

$$
\lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \frac{1}{\phi} \frac{y}{x}
$$

And, if we set the fee to zero ($\phi = 1$), then:

$$
M_p = \left| \frac{y}{x} \right|
$$

where $M_p$ denotes marginal price

$M_p$ is equal to the magnitude of the slope of the tangent line.
\[ xy = k \]

**Marginal Price & Slippage**

\[
\frac{\Delta y}{\Delta x} = \frac{y\phi}{x + \phi \Delta x} \quad \text{Selling } x \text{ for } y
\]

\[
\frac{\Delta y}{\Delta x} = \frac{1}{\phi} \cdot \frac{y}{x - \Delta x} \quad \text{Buying } x \text{ for } y
\]

**Observation #2**

Pricing depends on the size of \( x \) and \( y \) (i.e. \( k \))

It’s straightforward to see that, as \( k \) increases, the effective price of the AMM is less sensitive to \( \Delta x \).
Incentives for Liquidity Providers
Alice deposits 10 ETH and 12 DAI of liquidity, which implies:

\[ M_p = 1.2 \]

where \( M_p \) denotes marginal price

Alice waits for a month, during which traders drive $700 worth of volume through the AMM.

At the end of the month, Alice withdraws her ETH and DAI. By that time, the price of ETH has gone up 4x. The marginal price is now:

\[ M'_p = 4.8 \]

**What is Alice’s return?**

Assume: \((1 - \phi) = 0.003\)
First, what does Alice earn from liquidity provider fees?

\[ V(1 - \phi) = 700 \times 0.003 = $2.1 \]

where \( V \) denotes trading volume

Second, how many ETH and DAI does Alice get back?

\[ x' = 5 \text{ ETH} \]
\[ y' = 24 \text{ DAI} \]
Impermanent Divergence Loss

So, how did Alice do?

Measured in DAI, Alice now has:

$$R = 5 \text{ ETH} \times \left(\frac{4.8 \text{ DAI}}{\text{ETH}}\right) + 24 \text{ DAI} + 2.1 \text{ DAI}$$

$$R = 50.1 \text{ DAI}$$

Not bad, but how would she have done if she had just held onto her 12 ETH and 10 DAI?

$$R_B = 12 \text{ ETH} \times \left(\frac{4.8 \text{ DAI}}{\text{ETH}}\right) + 10 \text{ DAI}$$

$$R_B = 67.6 \text{ DAI}$$

This is called impermanent loss divergence.
**Impermanent Divergence Loss**

What if volume had been higher?

Say, volume had been $7,000 instead of $700:

\[ V(1 - \phi) = 7000 \times 0.003 = 21 \]

Therefore,

\[ R = 5 \text{ ETH} \times \frac{4.8 \text{ DAI}}{\text{ETH}} + 24 \text{ DAI} + 21 \text{ DAI} \]

\[ R = 69.0 \text{ DAI} \]

This time, Alice’s returns are greater than her baseline return \( R_B \) of 67.6 DAI. Her profit:

\[ P_L = \frac{R}{R_B} - 1 = 2.1\% \]
**Impermanent Divergence Loss**

More generally

Alice’s return $R$ is given by:

$$R = x'M_p' + y' + V(1 - \phi)$$

Her baseline return $R_B$ is given by:

$$R_B = xM_p + y$$

Her profit, in percentage terms is given by:

$$P_L = \frac{R}{R_B} - 1 = \frac{x'M_p' + y' + V(1 - \phi)}{xM_p + y} - 1$$

Let’s ignore the volume term for now, and simplify:

$$P_L = \frac{x'M_p' + y'}{xM_p + y} - 1 \quad \text{assuming } V = 0 \text{ for now}$$
Impermanent Divergence Loss

Simplifying

Step 1: let’s express everything in terms of $M_p$ and $k$.

$P_L = \frac{x'M_p' + y'}{xM_p + y}$

Step 2: Reintroduce the volume term:

$P_L = \frac{2\sqrt{r}}{r + 1} - 1$

Step 3: Plot this equation

Recall

$xy = k$ and $M_p = y/x$

Thus,

$x = \sqrt{\frac{k}{M_p}}$ and $y = \sqrt{km_p}$

Also,

$x' = \sqrt{\frac{k}{M_p'}}$ and $y' = \sqrt{km_p'}$

Finally, let:

$M_p' = rM_p$
Impermanent Divergence Loss

\[ P_L = \frac{2\sqrt{r}}{r + 1} + \frac{V(1 - \phi)}{c} - 1 \]

Optimal P/L occurs when the final price is equal to that at liquidity provisioning.

P/L percentage of liquidity provision on Uniswap for different scenarios of exchange trading volume and ETH price change.

Image credit: https://www.tokendaily.co/blog/pnl-analysis-of-uniswap-market-making
Quick Demo: [https://zumzoom.github.io/analytics/uniswap/roi/](https://zumzoom.github.io/analytics/uniswap/roi/)
Big Limitation of Uniswap V2

Capital Efficiency
One Approach: **Curve.Fi**
Uniswap V3: Universal AMM

Demo: app.uniswap.org
Concentrate Liquidity

https://uniswap.org/blog/uniswap-v3/
Narrow Activation

https://uniswap.org/blog/uniswap-v3/
Unified Pool

https://uniswap.org/blog/uniswap-v3/
Capital Efficiency: Example

- Alice's V2 Strategy
  - $1M
  - 500,000 DAI
  - 333.33 ETH

- Bob's V3 Strategy
  - $183,500
  - 91,751 DAI
  - 611.17 ETH

- Remaining
  - $816,500

After 1 year
- While ETH ranges between: $1000 ↔ $2500
- 15.9% Effective Capital
- 50% APR

- Alice's V2 Strategy
  - Capital
  - 12 Month Fees

- Bob's V3 Strategy
  - Capital
  - 12 Month Fees

* Liquidity spread across the full price curve
* Concentrated liquidity between $1000 ↔ $2500
* Positions values do not account for fees accrued

https://uniswap.org/blog/uniswap-v3/
# Uniswap v3 Core

March 2021

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hayden Adams</td>
<td><a href="mailto:hayden@uniswap.org">hayden@uniswap.org</a></td>
</tr>
<tr>
<td>Noah Zinsmeister</td>
<td><a href="mailto:noah@uniswap.org">noah@uniswap.org</a></td>
</tr>
<tr>
<td>Moody Salem</td>
<td><a href="mailto:moody@uniswap.org">moody@uniswap.org</a></td>
</tr>
<tr>
<td>River Keefer</td>
<td><a href="mailto:river@uniswap.org">river@uniswap.org</a></td>
</tr>
<tr>
<td>Dan Robinson</td>
<td><a href="mailto:dan@paradigm.xyz">dan@paradigm.xyz</a></td>
</tr>
</tbody>
</table>

[https://uniswap.org/whitepaper-v3.pdf](https://uniswap.org/whitepaper-v3.pdf)
Uniswap’s Metrics To Date

Quick Demo: https://uniswap.info/
Example: Uniswap

Interoperability
DEXs: Concluding Thoughts
Desired Characteristics

• Simple — buildable as a smart contract

• Automated liquidity — no dependence on active market-makers

• No single points of control — no dependence on centralized parties

• Composable/Programmable