Kaiko

DEX Methodology

A comprehensive overview of decentralized exchanges, swaps and liquidity events

OVERVIEW

Patent pending on DEX data methodologies.

This report covers all aspects of Kaiko's DEX trade and liquidity data, providing examples for analysis and a detailed overview of our collection methodology and normalization procedures. We also provide an overview of the five Ethereum-based DEX and DEX aggregators in our coverage.

Written by Anastasia Melachrinos, Clara Medalie, and Riyad Carey

About Kaiko

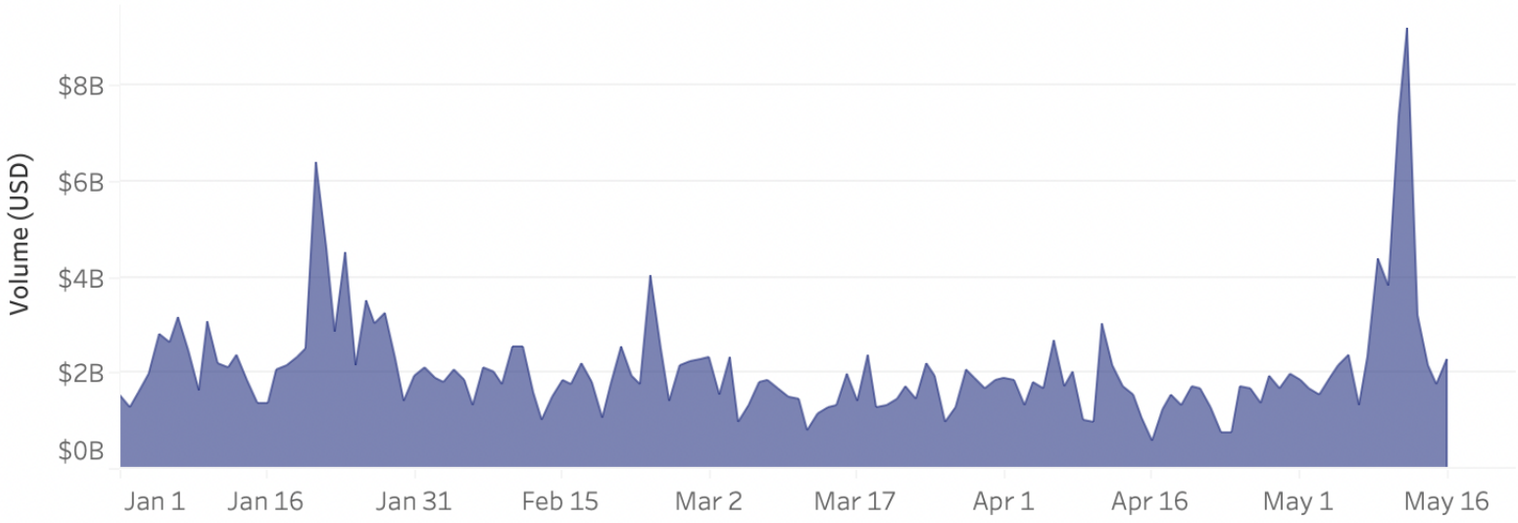
Kaiko is the leading cryptocurrency market data provider for institutional investors and enterprises. We empower market participants with accurate, transparent, and actionable financial data to be leveraged for a range of market activities. Our institutional grade data services enable seamless connectivity to historical and live data feeds across 100+ centralized and decentralized exchanges. Contact us at www.kaiko.com to learn more about our data and research services.

I. Introduction to DEXs

Decentralized exchanges (DEXs) have soared in popularity since 2020, accounting for hundreds of billions of dollars in trade volume over the past year alone, making them a formidable contender to centralized exchanges (CEXs). It is critical to understand and evaluate DEX data to fully understand how cryptocurrency markets are structured today. However, DEX data is uniquely complex to collect, clean, and normalize, and requires direct interaction with the Ethereum blockchain. This report will explore how the biggest Ethereum-based DEXs operate, with a focus on Kaiko's methodology and normalization procedure for swaps and liquidity data.

2022 DEX Trade Volume

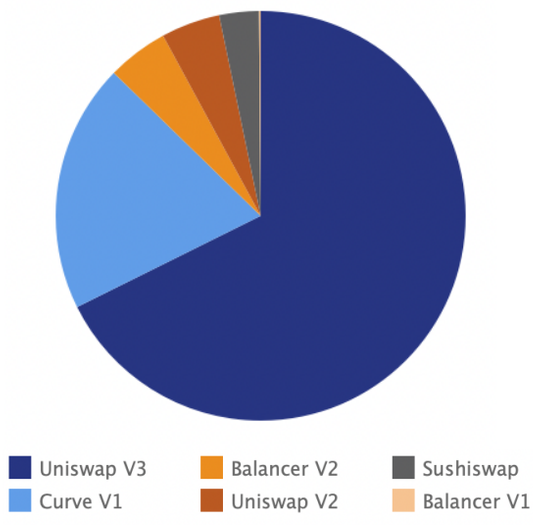
On average, more than $2 billion is traded every day on Ethereum-based DEXs



Coverage

Kaiko's DEX Data Feed covers the most liquid Ethereum-based DEXs: Uniswap (V2 + V3), SushiSwap, Balancer (V1+V2), Curve (V1 + Factory Pools), and DEX aggregator 1inch (V3). Combined, these DEXs account for the majority of decentralized trading activity across all blockchain networks. From these DEXs, Kaiko collects four data types: trades, liquidity events, liquidity snapshots, and pool reference data. Today, Uniswap possesses the largest market share of volume out of all Ethereum-based DEXs.

DEX Market Share of Volume

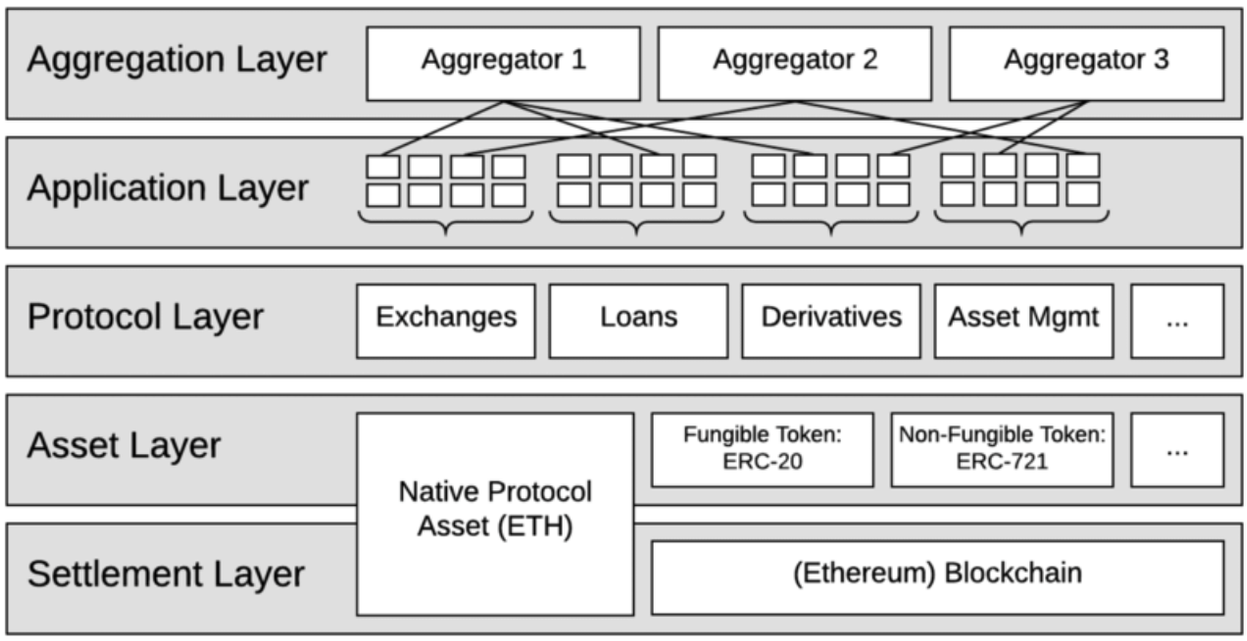


*As of June 1, 2022

DEX Infrastructure: An Overview

DEXs enable direct peer-to-peer transactions through automated smart contracts, allowing anyone to create trading pairs, provide liquidity, and swap tokens without the need for any centralized intermediaries. To understand how data is collected from a DEX, it is necessary to have a high-level overview of blockchain architecture, which today comprises five defined layers. DEXs exist on just one part of this multi-layer stack, of which blockchain protocols like Ethereum are the foundation.

Blockchain Architecture



Source : Fabien Schär, Decentralized Finance: On Blockchain- and Smart Contract-Based Financial Markets, 2021, FRED Saint Louis.

The blockchain protocol, being the settlement layer (Layer 1), defines the entire ecosystem's infrastructure, security and critical functionality. The asset layer (Layer 2) sets up the standards used to create tokenized assets, such as ERC-20 tokens or NFTs (Non-Fungible Tokens), in addition to scaling solutions such as Optimism or Arbitrum, which allow transactions to be executed at a lower cost than those on the main blockchain network.

Decentralized finance occurs on top of the asset layer, with utility-oriented protocols such as DEXs (Uniswap, Sushiswap), lending/borrowing protocols (Aave, Compound), and derivatives exchanges (Synthetix). Above the protocol layer is the application layer, which is dedicated to user interfaces that allow DeFi protocols and smart contracts to be used by end-users. Finally, the aggregation layer builds user-centric systems with connections to a variety of protocols or applications.

This report specifically focuses on Ethereum DEXs, which operate on the **protocol layer** of a blockchain. While DeFi protocols are often deployed on other blockchains or Layer 2 scaling solutions, highlighting the composable nature of this technology, Ethereum-based (mainnet) DEXs dominate DeFi activity today.

Centralized vs. Decentralized Exchanges

Historically, the majority of crypto market activity has happened on centralized exchanges (CEXs), which serve as the main fiat gateway to cryptocurrency markets. CEXs operate much like traditional financial marketplaces with a centralized order book that enables market makers to submit bids and asks, which are in turn matched with market buy and sell orders. The key difference between a CEX and a DEX lies in who controls a user's assets: on a CEX, traders relinquish control of their crypto to a centralized intermediary that manages custody and order matching. As such, all market activity happens "off-chain", whereas on a DEX all trades are executed and recorded directly on the blockchain.

With the use of smart contracts, DEXs operate entirely autonomously and without any centralized intermediary, meaning traders fully control their assets. For example, while Uniswap's website serves as the frontend for the exchange (and the Uniswap team can delist tokens from this website) it is always possible to interact directly with Uniswap's contracts on the Ethereum blockchain, even for tokens delisted from the frontend. Because DEXs are fully decentralized and native to a blockchain, users are only able to swap cryptocurrencies, not fiat.

Centralized (CEX)	Decentralized (DEX)
- Centrally managed (off-chain) - Exchange controls your assets - Listing fees and due diligence for new trading pairs - Regulatory jurisdictions and KYC - Barriers to market making - Data provided through public REST/ WebSocket APIs	- Decentralized protocol (on-chain) - You control your assets - Anyone can list a pair by creating a new smart contract - No regulation or KYC - Anyone can be a market maker - Data recorded directly on the blockchain

The main differentiators between centralized and decentralized exchanges

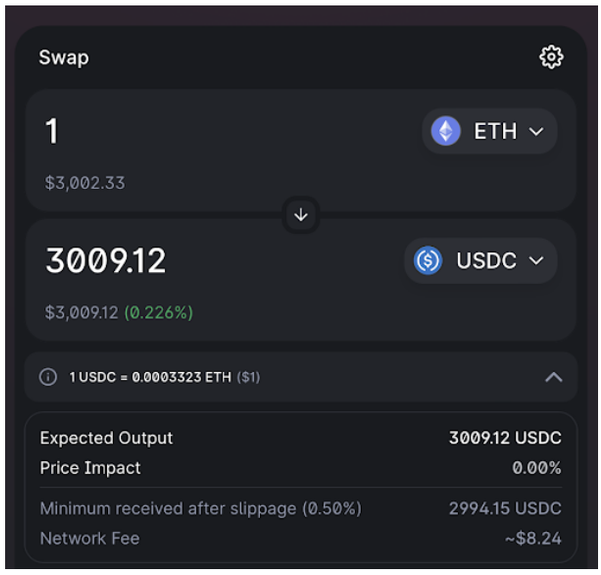
DEX Market Events

On a DEX, a market event is classified as either (i) **token swaps**: the exchange of one token for another, analogous to a "trade" on a centralized exchange or (ii) **mint/burn events**: corresponding to token adds/removals to/from liquidity pools, which are initiated by liquidity providers and analogous to "order book events" on a CEX. Trade ("swaps") data is very similar in both form and structure on both types of exchanges. Liquidity data is where the biggest differences lie, reflecting the distinct order matching mechanisms each type of exchange relies on.

Swapping

Swaps on a DEX function much the same as trades on a CEX: a user chooses which token they would like to swap, is presented with important information (current price, slippage and network fee), and is prompted to confirm the swap. The swap is analogous to a market order being matched with a bid or ask on a centralized order book, except in this case, the exchange occurs with a token lying in what is called a "liquidity pool." A liquidity pool can be thought of as a vault where users can securely store their tokens, which are used as a source of liquidity. Swaps always occur between just two tokens and involve the exchange of one token for another lying in the pool.

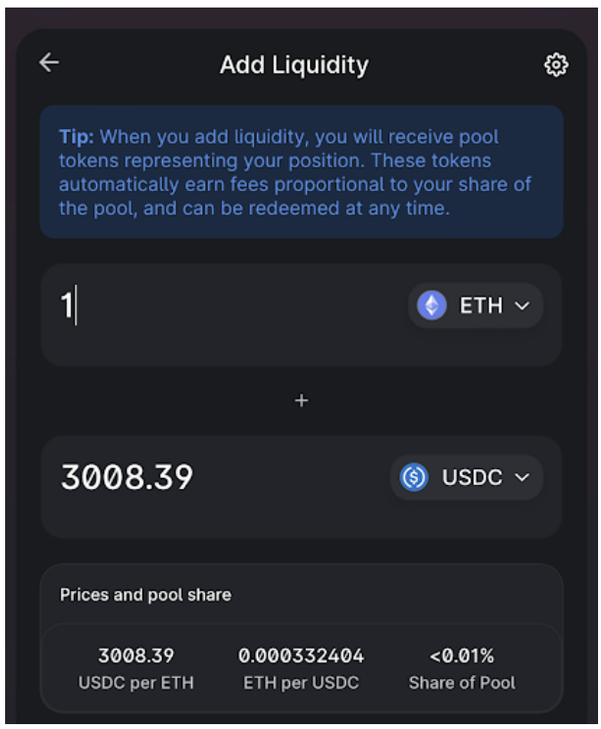
Swapping ultimately enables the price discovery process because every time a token is added or removed from a pool, the price of the tokens adjust.



Swaps interface on Uniswap

Providing Liquidity

Liquidity pools generally contain **two or more** tokens and most DEXs require liquidity providers (LPs) to deposit pairs of tokens reflecting the current composition of the pool. For example, for the USDC-WETH pool on Uniswap V2, a liquidity provider would deposit both WETH and USDC in equal value, thus maintaining the current price of the assets [more information on how assets are priced is detailed in section VII]. LPs on a DEX can be compared with market makers on a CEX: LPs deposit tokens into a pool— which is called a "mint" — after which they receive liquidity pool tokens (LP tokens) and earn swap fees. LP tokens can be "burned" at any time, which allows the liquidity provider to receive the underlying tokens plus any accrued swap fees.

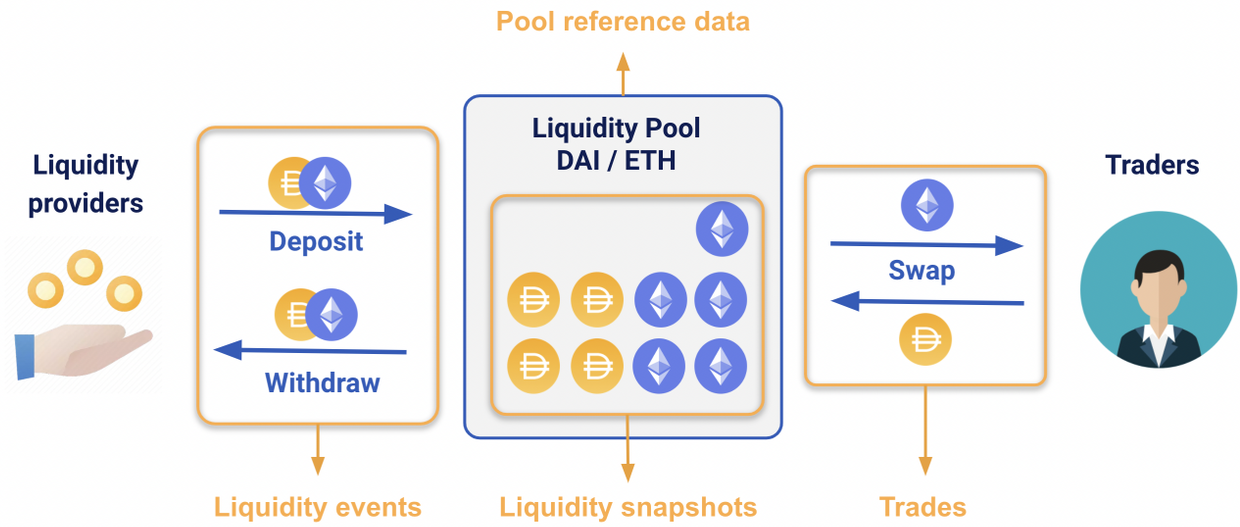


Liquidity provider interface on Uniswap

These two interactions: swaps and the addition (mints) or removal (burns) of liquidity comprise the two distinct types of DEX data that Kaiko provides.

II. DEX Data Overview

Today, Kaiko's DEX Data Feed includes four types of data: trades, liquidity events, liquidity snapshots, and reference pool data. These data types are collected block-by-block from the Ethereum blockchain, and normalized into a single format across all DEXs.



Trades

Kaiko produces DEX trade ("swap") data exactly as it is formatted on a CEX, including the price, amount, timestamp, and trade direction. Swaps always occur between two tokens, even if a liquidity pool contains more than two, enabling us to assign a consistent "base" and "quote" to a recurring combination of tokens.

Liquidity Events

Liquidity events denote when a liquidity provider adds or removes tokens from a liquidity pool. This data includes the pool's name, address, exchange, type (mint [add] or burn [remove]), amount per token, and details about the token. This data provides a tick-level overview of all liquidity added or removed to a pool.

Liquidity Snapshots

Kaiko collects and provides block-by-block liquidity snapshots for each pool in our coverage. The data output contains the amount of tokens available in each pool at the time of the snapshot. This data can be used to calculate the "total value locked" per liquidity pool by applying a USD conversion.

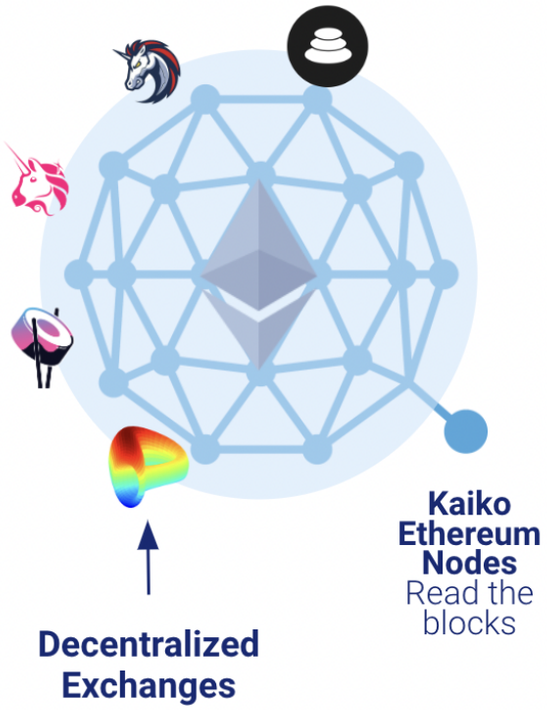
Pool Reference Data

This data contains reference information about each DEX pool, including the pool address, pool names, the type of pool, percentage weights of tokens in a pool, and the associated fee with using the pool. Reference data reveals information about pre-programmed rules for each liquidity pool.

DEX Data Collection

Centralized exchanges such as Coinbase, Binance, and Gemini offer data collection and resolution using public REST or WebSocket APIs, which makes the collection and normalization of market data feeds relatively straightforward. In contrast, DEX data collection requires direct interaction with the Ethereum blockchain.

To collect DEX data, Kaiko operates several full archive Ethereum nodes, which store a snapshot of all blocks since the origin of the chain. These allow us to collect all historical and live data from Ethereum blocks with a very low risk of downtime and no risk of missing data.



Available Now:
Available Soon:

DEX Data Coverage

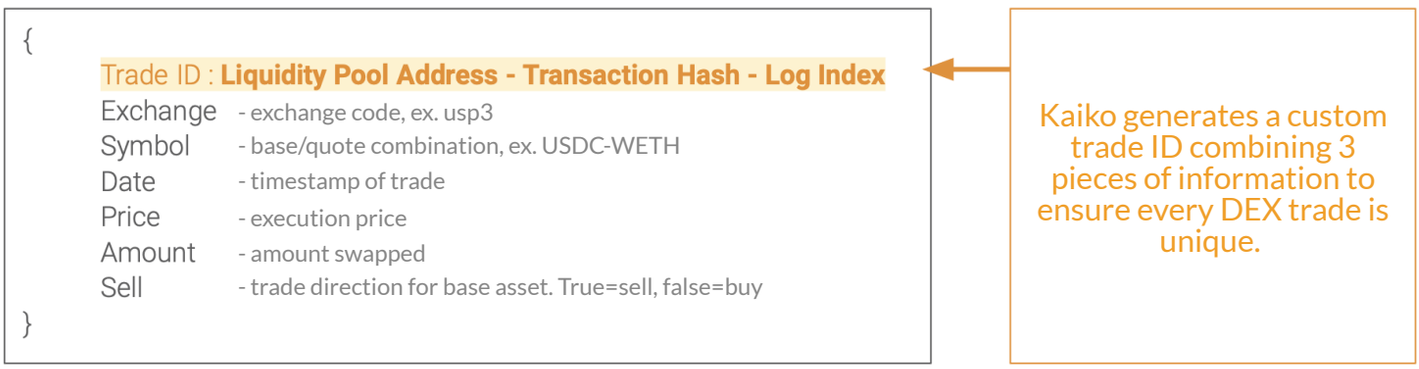
Exchange	Versions Covered	Blockchain	Trades Live	Trades Historical	Liquidity Pool Live	Liquidity Pool Historical
Uniswap	V2, V3	Ethereum	✅	✅	✅	✅
Sushiswap		Ethereum	✅	✅	✅	✅
Curve	V1, Factory Pools, V2	Ethereum, Polygon	✅	✅	✅	✅
Balancer	V1, V2	Ethereum	✅	✅	✅	✅
1inch	V3, V4	Ethereum, Avalanche, BNB Chain, Polygon	✅	✅	❌	❌

All DEX data is collected block-by-block and normalized into the four data types we provide. While the final data output is the same across all DEXs, there are many unique considerations that we take into account. CEX data collection is very similar across all exchanges, while DEX data collection requires protocol-specific and blockchain-specific infrastructure and collection methodologies. Kaiko will soon expand collection to other blockchains and Layer 2 protocols, which will increase the number of DEXs covered.

III. Trade Data ("Swaps")

On-chain trade data sourced from DEXs requires a thorough understanding of how swapping and liquidity pools work, which can vary depending on the protocol (detailed in section VII). However, the base layer data remains the same: trades on both centralized and decentralized exchanges include the price, amount, direction and timestamp of the transaction. The normalization procedure for assigning a base/quote direction to each swap and for assigning a trade ID are the main differentiating factors, described below.

Trade Data Structure



Trade ID

Trade IDs on centralized exchanges are easy to understand because most exchanges automatically generate a unique ID with every trade and there is only one trading pair per base/quote combination. On DEXs, Trade IDs are not so simple: sometimes multiple pools can exist for the same asset pair and some trades recorded on a blockchain can have the same "transaction hash."

This makes it complicated to differentiate trades, which is why developing a unique trade_ID for DEX trades is a particularly important problem to solve. Kaiko generates a custom DEX trade ID based on three pieces of information: the transaction hash, the pool address and the log index. This ensures that no matter what, all trades are unique and also enables traders to identify the liquidity pool that a trade originated from, even when multiple pools on a DEX may exist with the same tokens.

"trade_id": "0x397ff1542f962076d0bfe58ea045ffa2d347aca0-0x45a4db43b2cabcf01f85c87784950f69ff38eb306a74dbf84f43cc89933e456a-0x61"

[ ] = pool address [ ] = transaction hash [ ] = log index

Pair vs. Pool

On DEXs, the notion of a "base" and "quote" currency for a pair of assets is not applicable. For example, on a CEX both a buy and sell trade for ETH to USDT will be denoted as occurring on the 'ETH-USDT' pair. On a DEX, a sell trade for (W)ETH to buy USDT will be denoted as (W)ETH-USDT on the blockchain, whereas a buy trade for WETH to sell USDT will be denoted as USDT-WETH. This raises the question of what constitutes a "base" and a "quote" currency for a pair on a DEX. To maintain consistency in our data, we take each asset's Ethereum address and order them alphabetically to determine which token in a pool will be denoted as the base or quote. Our normalization system automatically maintains this order and inverts the price and amount of a trade if necessary to match all previous trades.

Another complexity lies in the amount of assets that can exist in a liquidity pool. For example, on Curve, liquidity pools can contain more than two assets. However, all swaps still occur between just two assets, which means we need to split each pool into individual pairs depending on the number of asset combinations possible. For example, a Curve pool containing 3 assets— USDC, USDT, and DAI—would generate three pairs: DAI-USDC, DAI-USDT, and USDC-USDT, reflecting all possible swap combinations.

View appendix for our full methodology on assigning trade IDs and base/quote pairs.

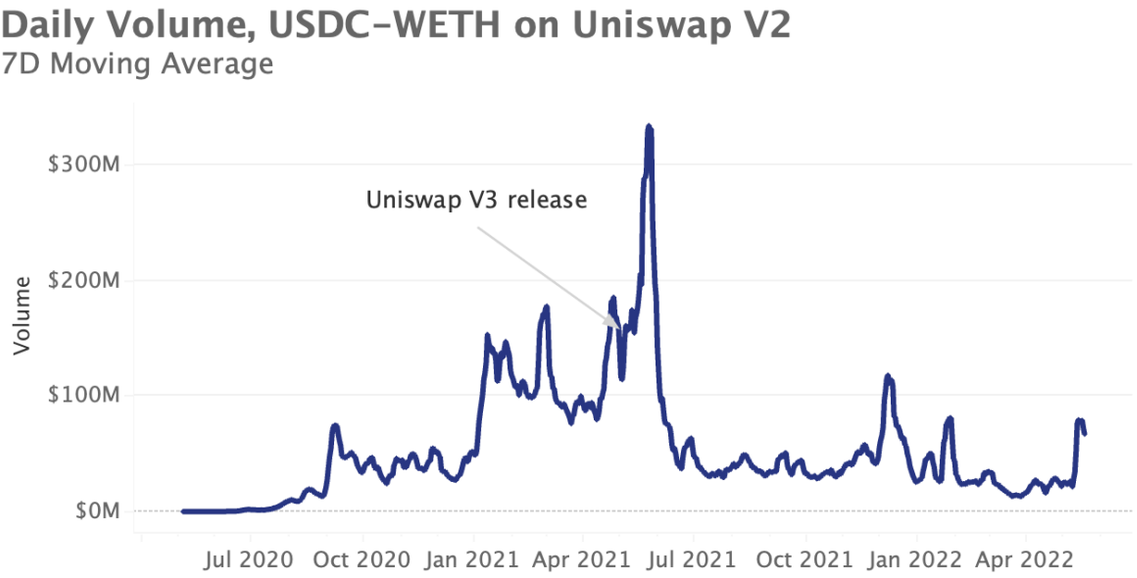
Exploring DEX Trade Data

Trade data provides a tick-level overview of all swaps happening on a DEX. With this data, we can study patterns in volume, trade size, trade frequency, and price discovery. To the right, we show an example trade data point for the USDC-WETH pair on Uniswap. The *sell* variable refers to the trade direction, which when false means that the trader "bought" USDC in exchange for WETH.

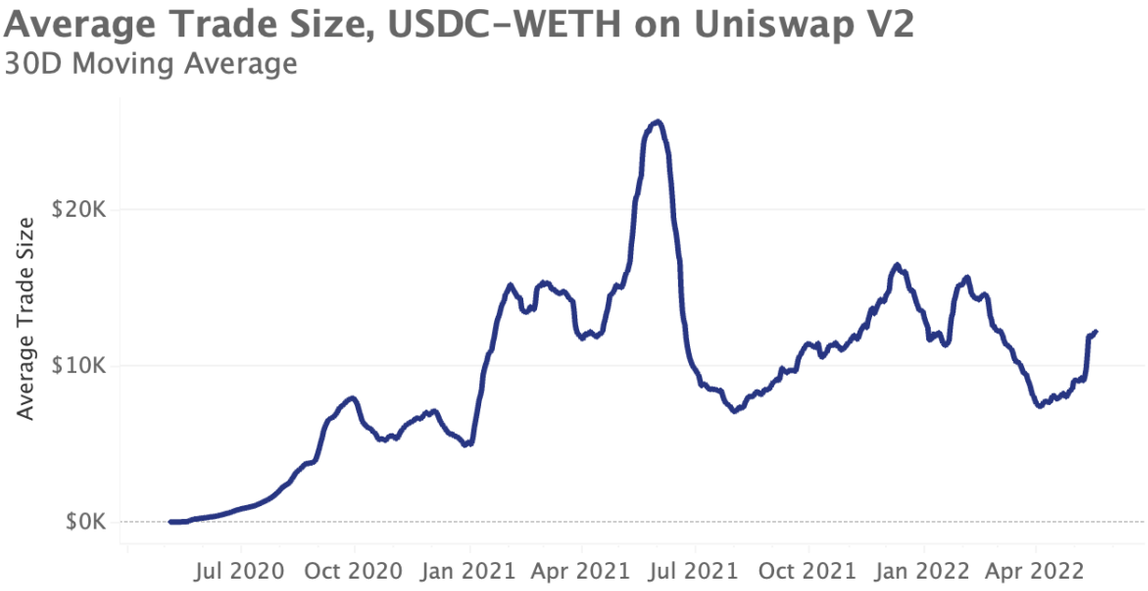

```
"timestamp": 1653299260000,
"trade_id":
"0x88e6a0c2ddd26feeb64f039a2c41296fcb3f5640-
0x7d2a2c40d01498b043906991af144586e153b886
05efb440363e146a4bc4511a-0x3",
"price": "0.000485270774782244",
"amount": "811.949914",
"sell": false
```


Trade data example, USDC-WETH

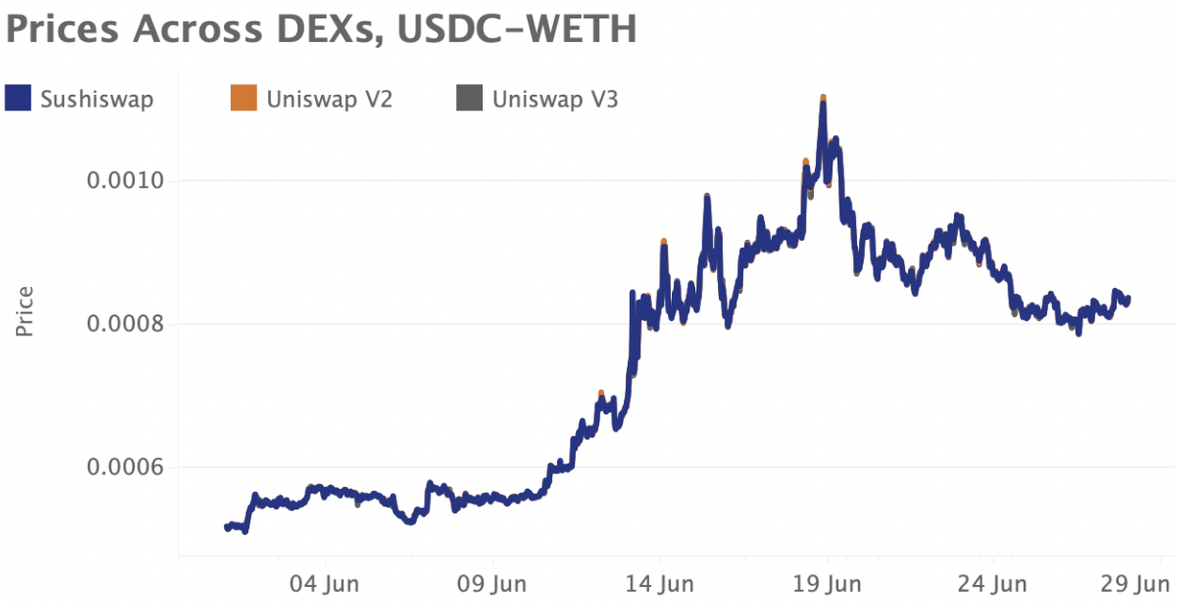
Using trade data, we can chart the daily volume for the USDC-WETH pool on Uniswap V2, and can observe how volume plummeted following the release of V3 of the protocol.



Average trade size shows that trades are between $10-$20k for this pool on Uniswap, far higher than the average CEX trade, which is between $2-$3k.



Examining the USDC-WETH pool on top DEXs, we can see that decentralized markets are very efficient for more liquid tokens.
Kaiko maintains the USDC-WETH base/quote order as part of our normalization methodology (pricing 1 USDC in WETH), although this can easily be inverted to achieve the price of 1WETH in USDC.



IV. Liquidity Events

Liquidity events refer to all information on the addition or removal of tokens to/from a liquidity pool by liquidity providers (LPs). This data does not include changes to the composition of a pool that occurs during swaps. Most DEXs require LPs to deposit liquidity in token pairs. For example, an LP will need to deposit both WETH and USDC into the USDC/WETH pool in order to maintain balanced prices. However, there are some exceptions to this such as Curve, which doesn't require deposits to be balanced.

When a user deposits tokens into a liquidity pool, they receive liquidity pool tokens that represent their pool share. Thus this process is referred to as a "mint", as the user is minting a liquidity pool token by adding tokens to a pool. In the reverse, a user will "burn" their liquidity pool token and receive the underlying tokens in return, plus any transaction fees that they accrued. All mints and burns are recorded directly on the blockchain and Kaiko collects this information, as well as the associated information included below.

Liquidity Events Data Structure

{

Block	- block height of the Ethereum blockchain for the event
Timestamp	- timestamp of the event
Pool Name	- ex. USDC - WETH - .001 (referring to fee)
Pool Address	- pool address on the Ethereum blockchain
Exchange	- exchange code, ex. usp3
Transaction Hash	- unique transaction hash of the event
Price	- the price at the time of the event, as collected on-chain. No normalization is applied.
Type (mint/burn)	- mint=add liquidity, burn=remove
Amounts	- includes the token, token address, and amount of token added to the pool

}

Our liquidity events data includes an optional metadata field giving an LP's specified concentrated liquidity price levels on Uniswap V3 for the mint/burn event.

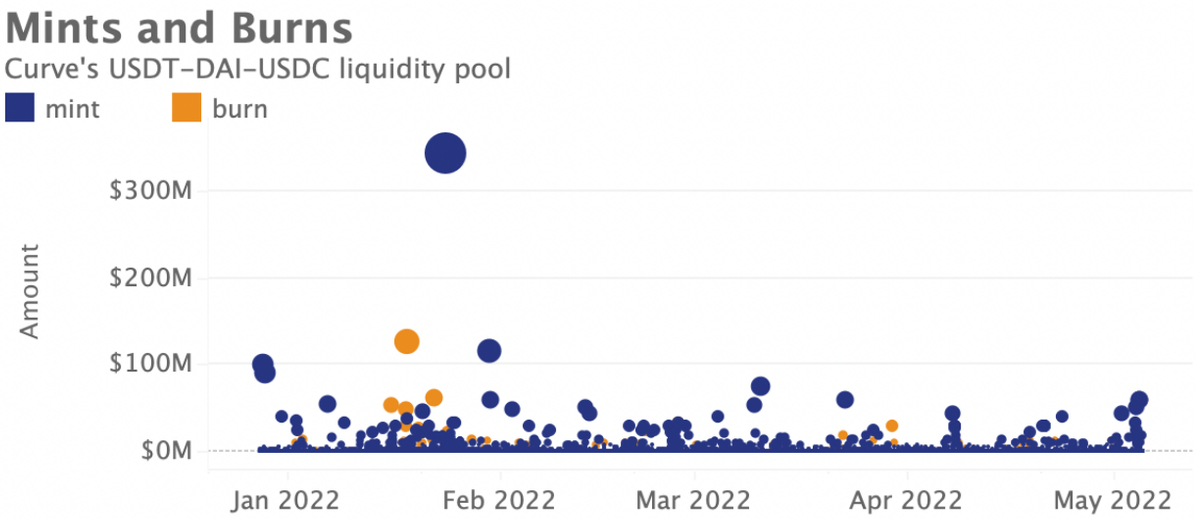
Exploring DEX Liquidity Events

Liquidity events provide a tick-level overview of how LPs are interacting with liquidity pools and how this liquidity evolves over time. Every mint and burn for a pool is available in our database, enabling the reconstructing of historical pool activity. This data can be aggregated at different intervals and a USD exchange rate applied to the token amounts minted or burned. An example datapoint is on the right, containing the amounts of USDC and WETH "minted" (deposited) into the Uniswap V3 pool. Below, we explore various visualizations of this data.

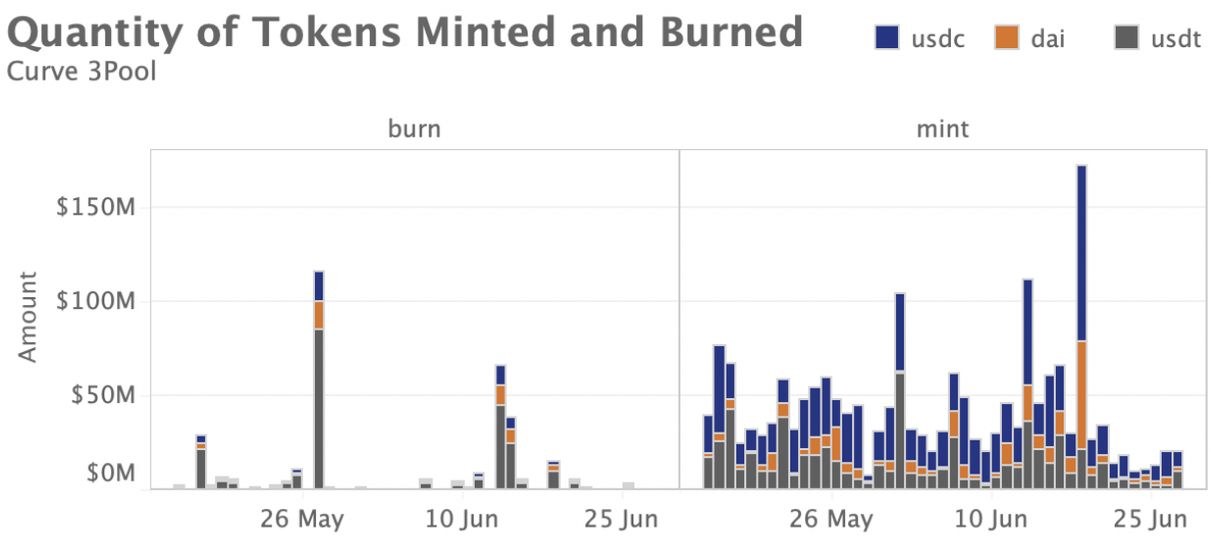

```
"block_number": 14841857,
"type": "mint",
"pool_name": "USDC-WETH-.001",
"pool_address": "0xb4e16d0168e52d35cacd2c6185b44281ec28c9dc",
"exchange": "usp3",
"transaction_hash":
"0x051c347ed328f606a4a18c12d761c457338496216457840dea167c16e58
",
"price": 0.000514657801896448,
"amounts": [

{"symbol": "USDC",
"address": "0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48",
"amount": 194.195646},
{"symbol": "WETH",
"address": "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2",
"amount": 0.09994430430822067} ]
```


Liquidity events example, USDC-WETH

Every mint and burn for Curve's 3pool is represented as an individual data point since the start of 2022. Each data point takes the sum of all tokens minted or burned, with a USD conversion applied. During times of volatility, mints and burns will increase.



We chart the sum of each individual token minted and burned for the 3pool. More USDC was minted compared with USDC and DAI whereas more USDT was burned. Curve enables LPs to deposit unbalanced tokens, whereas Uniswap and Sushiswap do not.



We can also chart the cumulative year-to-date change in liquidity in the Sushiswap USDC-WETH pool by taking the net amount of tokens minted and burned. If charted separately, USDC and WETH would have equal USD amounts minted/burned because the DEX only allows LPs to mint/burn a ratio of tokens based on the current price.

V. Liquidity Snapshots

Liquidity pool snapshots provide a bird's eye view of liquidity pools, much in the way order book snapshots provide context to tick-level order book events. Kaiko takes snapshots of all liquidity pools at each Ethereum block (generally every 12-14 seconds). Combined with liquidity events data, this provides a full view of each liquidity pool and enables the calculation of measures like "total value locked" for a protocol.

Liquidity Snapshots Data Structure

{

Block	- height of the Ethereum blockchain at the snapshot
Timestamp	- timestamp of the snapshot
Pool Address	- pool address on the Ethereum blockchain
Exchange	- exchange code, ex. usp3
Transaction Hash	- unique transaction hash of the event
Amounts	- includes the token, token address, and amount of tokens in the pool at the time of the snapshots

}

Block-by-block liquidity snapshots contain a list of the quantity of each asset within a pool at the moment of the snapshot.

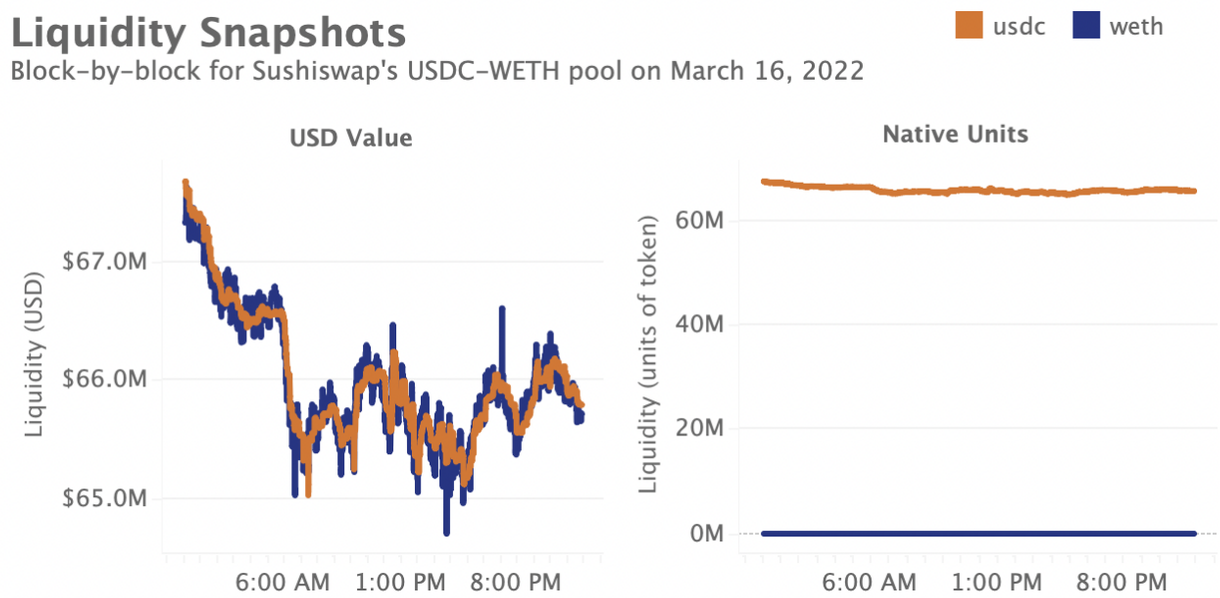
Exploring DEX Liquidity Snapshots

Liquidity snapshots reveal the block-by-block composition of a liquidity pool over time. For most DEXs, the value of each individual asset within a liquidity pool is designed to be equal with the others, thus any divergence in the value of the quantity of one asset is used by arbitragers (swappers) to figure out how to re-balance a pool and arbitrage away any price differences. The bigger the quantity of assets in a pool, the more liquid and more difficult it is for a swap to change the price. Thus, a common indicator for a pool's liquidity is the "total value locked", which can be calculated by taking the sum of all assets in a pool at the time a snapshot is taken.

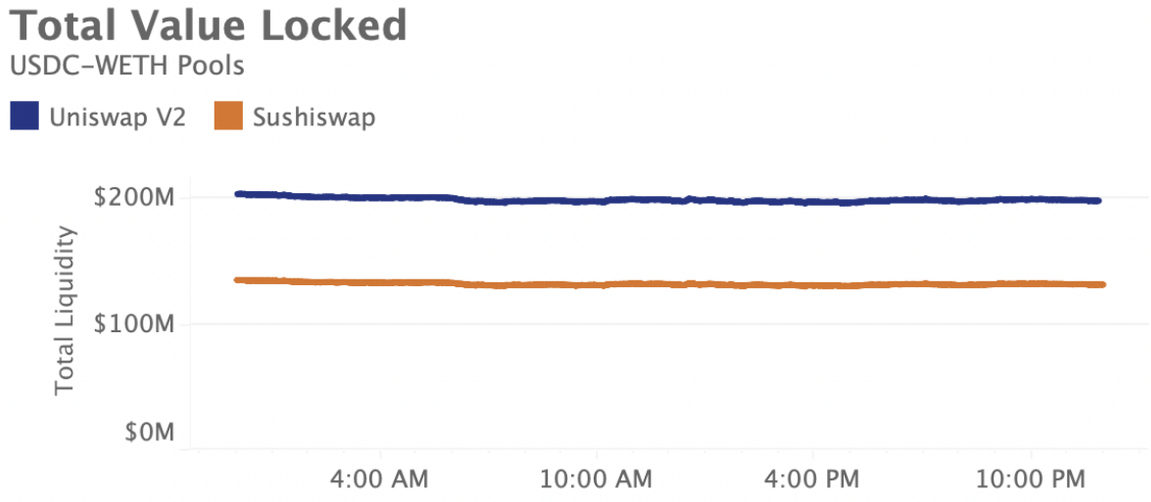

```
"block_number": 14842582,
"pool_name": "3pool",
"pool_address": "0xbebc44782c7db0a1a60cb6fe97d0b483032ff1c7",
"exchange": "curv",
"amounts": [
{ "symbol": "DAI",
"address": "0x6B175474E89094C44Da98b954EedeAC495271d0F",
"amount": 205758309.7472583},
{"symbol": "USDC",
"address": "0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48",
"amount": 208783335.335607},
{"symbol": "USDT",
"address": "0xdAC17F958D2ee523a2206206994597C13D831ec7",
"amount": 1150101805.156811} ]
```


Liquidity snapshot example, Curve 3Pool

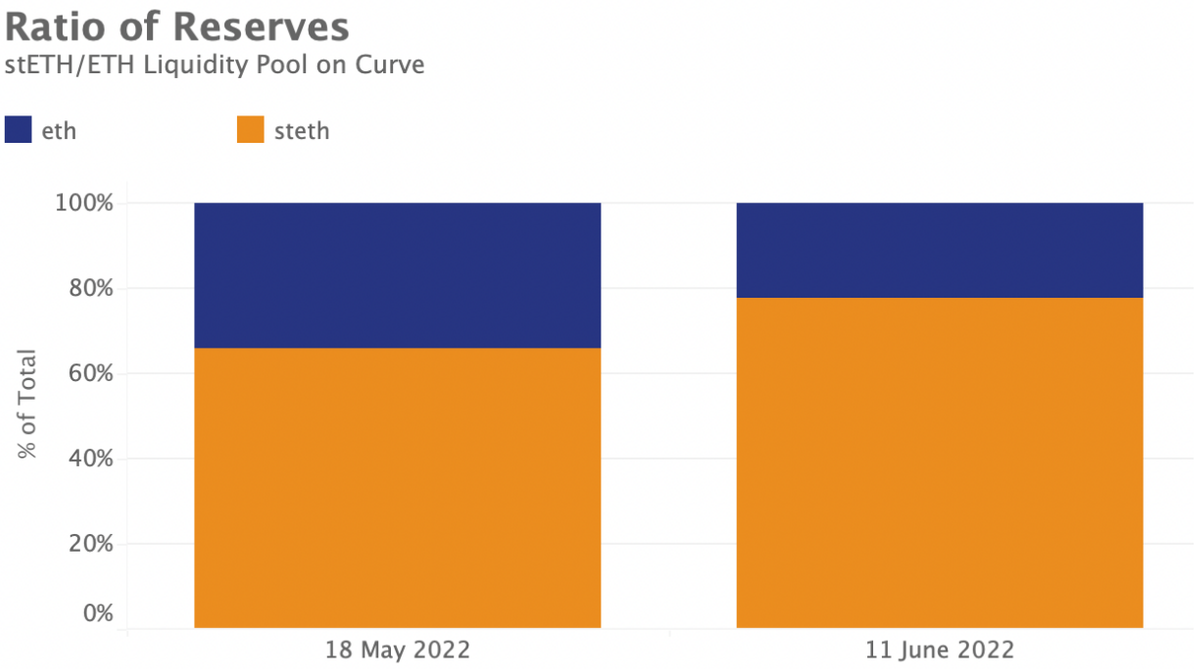
We chart both the USD value and native units of each token in Sushiswap's USDC-WETH pool. When the USD value of the quantity of each token diverges, an arbitrager will swap to make the value equal and rebalance the pool based on current market prices.



We can also compare liquidity across multiple protocols by summing the amount of each asset in similar pools. Uniswap has a higher total value locked in the USDC-WETH pool, which suggests it is more liquid than Sushiswap.



Using liquidity snapshots, we explore the stEth-Eth liquidity pool on Curve before and after stETH diverged from the price of ETH, triggering a wider market collapse. On May 18th, stETH accounted for 66% of reserves and on June 11th, more than 77%. This suggests that swappers were selling stETH at a higher ratio, creating an imbalanced pool.



VI. Pool Reference Data

Pool reference data serves as the guide for all of the pools that are a part of Kaiko's offerings. Because of the complex nature of pools (i.e. different USDC/WETH pools on Uniswap V3 that vary by fee, as well as the USDC/WETH pool on Uniswap V2) it is critical to have reference data for each pool. Kaiko's pool reference data outputs the following information, enabling users to understand exactly which pool a certain liquidity event or trade is referring to. This data will be integrated with Kaiko's Instrument Explorer, allowing users to index and examine pools across the exchanges we cover.

Pool Reference Data Structure

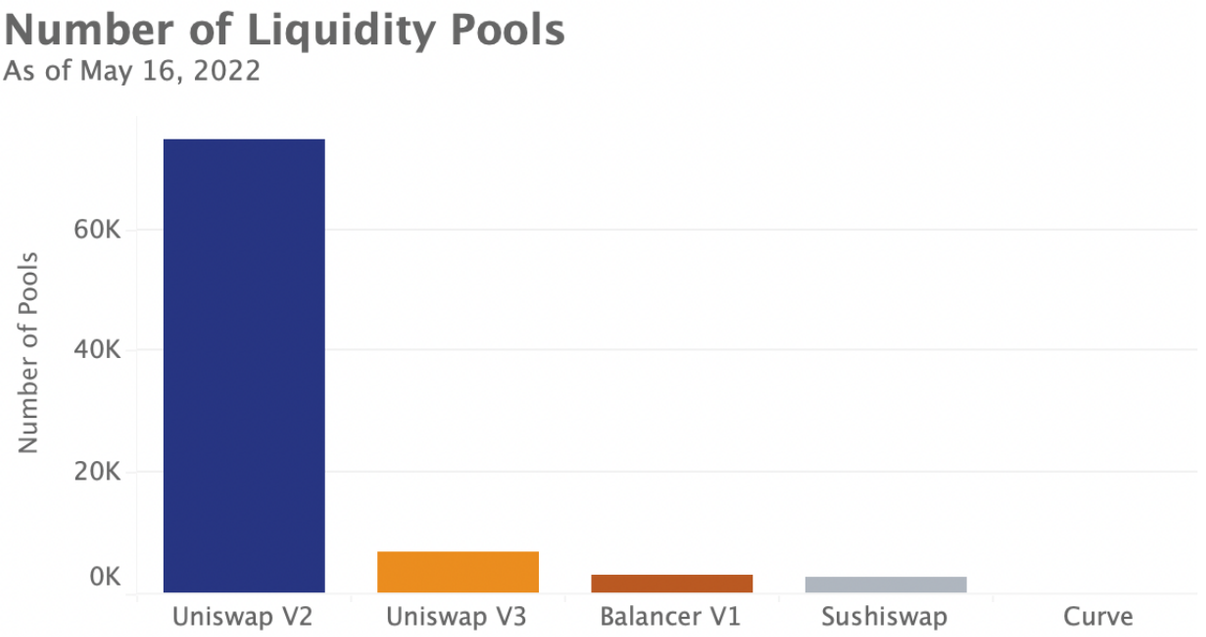
{

Address	- smart contract address on Ethereum
Name	- the pool name as specified on chain
Protocol	- the DEX, such as uniswap v3
Type	- the pool's type, such as metapool on Curve
Fee	- the trading fee for the pool
Tokens	- the tokens in the pool
Underlying Tokens	- specific to Curve, underlying tokens in the pool
Tick Spacing	- specific to Uniswap V3
Weights	- the weight of tokens in a pool, specific to Balancer

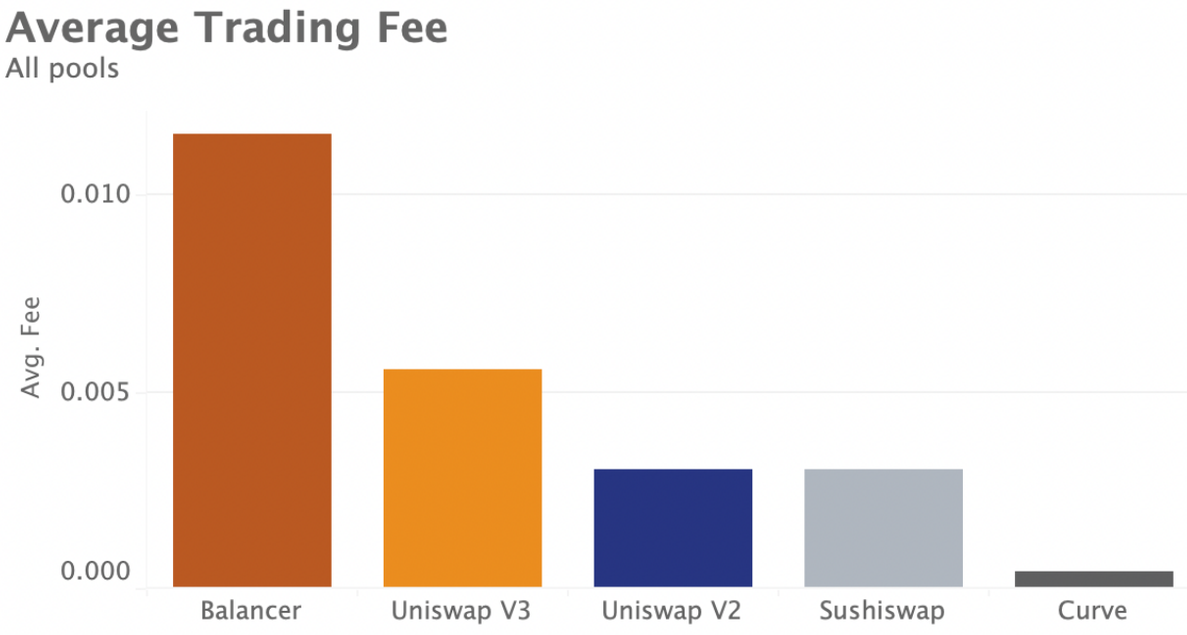
}

Pool reference data can be filtered using our REST API. For example, if you want to search for all pools with a .01 transaction fee or all pools that contain USDC.

We can explore the total number of liquidity pools that are active for each protocol. Anyone can create a liquidity pool on a DEX. Uniswap V2 has the most pools, although the majority of these pools have very little activity. Curve only caters for stablecoins, thus has the smallest amount.



Some DEXs like Uniswap V2 and Sushiswap have fixed fees, while others offer dynamic fees. Today, Curve has the lowest average fees across all pools.



VII. Protocols and Collection

There are many unique considerations when collecting and standardizing DEX data and this complexity compounds when collecting from a variety of DEXs. The differences across DEXs impact the methodology of data collection, normalization, and standardization by Kaiko. The below section provides background on each DEX and how they differ based on automated market maker, pools, fees, and more.

Uniswap V2

- **General:** second version of Uniswap protocol
- **Automated Market Maker:** constant product market maker model
- **Pools:** maximum 2 tokens
- **Fees:** 0.3%

Background

Uniswap quickly grew to become the leading DEX in DeFi, enabling the exchange of ERC-20 tokens on Ethereum. In just one year, total daily Uniswap V2 volume grew from $10 billion to $100 billion, positioning the DEX as a serious competitor to centralized cryptocurrency exchanges. Uniswap's story began in 2017 when its creator, Hayden Adams, decided to create an "automated market maker" (AMM) with a proof of concept, smart contracts, and a website. The project was announced a year later on Twitter as a protocol allowing the automated exchange of ERC20 tokens for which token listing is open and free. Since its launch, Uniswap has become the most recognized and used decentralized exchange.

Swapping

On Uniswap V2, a user is able to swap between two ERC20 tokens. If the user wants to swap Token A with Token B, the protocol will check if a Token A/Token B liquidity pool already exists. If so, then the swap can happen directly using this specific pool. If not, the swap will be routed across two or more liquidity pools. Pools on Uniswap V2 always contain two tokens and have a trading fee of 0.3% to incentivize LPs to provide liquidity. LPs always deposit the equivalent value of two tokens in a pool, thus swapping is the only way the price of a token can change.

Prices and Liquidity

Every liquidity pool on Uniswap V2 follows the law of constant product with ***x*** as the quantity of a token (LINK for example) and ***y*** as the quantity of another token (WETH for example). **k** is the constant product of the two quantities and rules price movements between the two tokens of the liquidity pool. Exchange rates are consequently determined according to the token balances on a pool and the following equation: ***x*y = k.*** It is worth mentioning that the invariant ***k***, is not unchanged: the fees paid by swappers to liquidity providers are added to each pool's reserves. As a result, each swap increases ***k*** by the value of the fees paid by swappers, expressed in liquidity pool tokens (LP tokens), which get paid out to LPs.

Data Collection

Kaiko directly extracts trade data from swap and liquidity events on Uniswap V2. Each pool can be identified by its pool address, a unique string of characters which identifies a pool on DEXs and can be considered as the equivalent of a token pair on a centralized exchange. All pools are related to the same smart contract; this contract is a combination of events that rule what is possible to do with Uniswap V2. Kaiko has sourced the Uniswap V2 contract that indexes swap events on each block of the protocol. Kaiko then identifies the pools for each swap event using pool addresses to deliver data from Uniswap V2.

Uniswap V3

- **General:** most recent version of Uniswap protocol
- **Automated Market Maker:** constant product market maker model within concentrated price ranges
- **Pools:** maximum 2 tokens
- **Fees:** multiple fee tiers possible for each token pair

Background

On May 4th 2021, the Uniswap Foundation released the third version of Uniswap's source code: Uniswap V3, designed to progressively replace the second version of the protocol. It has since become the top DEX by volume. Version 3's primary innovation is concentrated liquidity: liquidity bounded within a price range. This enables more efficiency and reduces slippage and volatility.

Swapping

On Uniswap V3, an asset pair can have multiple liquidity pools, each with a different fee model. Swapping works the same as on V2, except that traders must select the liquidity pool they would like to swap with. Generally speaking, users will select 0.05% for pools with similar assets (ex: USDC/USDT), 0.3% for standard pairs (ex: LINK/ETH), and 1% for more volatile or rarely traded tokens (ex: FTM/ETH).

For example, the USDC-WETH pair has three pools, each with a different fee model at .05%, .3%, and 1%. As expected, common pools such as USDC/WETH and WETH/USDT register the most volume at the 0.05% fee tier. In contrast, highly volatile pairs such as SHIB/WETH register the most volume for pools with higher transaction fees (0.3% and 1%).

Prices and Liquidity

Prices on Uniswap V3 are determined using the same liquidity model and equation as Version 2. The main different is how liquidity is provided: LPs can select a price range to provide liquidity for a pool, which ensures that liquidity is concentrated around the current price of a token, enabling more efficiency. LPs will provide liquidity at a fee tier that provides an appropriate level of risk/reward depending on the liquidity and trade volume for each asset.

Data Collection

Having multiple pools for the same asset pair creates an additional layer of complexity for data collection. Taking the example above for WETH/USDT pairs: When collecting data, Kaiko groups each of the three pools together to generate a single data point for USDC-WETH trade volume. In order to differentiate trades for the same WETH/USDT pair but from different pools, traders can look at the trade_Id, which contains the unique pool address. Liquidity events and snapshots are organized per pool, and are not aggregated.

Sushiswap

- **General:** fork of Uniswap V2
- **Automated Market Maker:** constant product market maker model
- **Pools:** maximum 2 tokens
- **Fees:** 0.3%

Background

Sushiswap is a fork of Uniswap V2's source code and uses the same constant product market maker model. The main objective of the project was to create an AMM managed by the community using its governance token, SUSHI.

SushiSwap introduced its governance at a time when Uniswap did not yet have its UNI governance token. In September 2020, a month after the fork, SushiSwap began offering SUSHI incentives to entice liquidity providers to switch from Uniswap to Sushiswap, in what was coined a "vampire attack". The protocol also designated a portion of revenue to go to those who staked their SUSHI tokens. Uniswap released its governance token shortly thereafter. While Sushiswap initially experienced surging growth, it has since lagged Uniswap in trade volumes and total value locked (a measure of the quantity of assets locked in liquidity pools).

Swapping

Because SushiSwap is a direct fork of the Uniswap V2 protocol, swapping on Sushiswap is nearly identical to Uniswap V2, always occurring between two tokens.

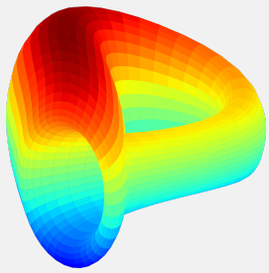
Prices and Liquidity

The main differentiator on Sushiswap is how fees are distributed: liquidity providers earn the majority of the fees (0.25%), and SUSHI stakers earn 0.05%, adding up to the same 0.3% as Uniswap V2. Again, SushiSwap follows the same pricing equation as Uniswap V2. The equation ***x*y = k,*** where ***x*** as the quantity of a token (LINK for example) and ***y*** as the quantity of another token (WETH for example). ***K*** is the constant product of the two quantities and rules price movements between the two tokens of the liquidity pool.

Data Collection

Since data is managed the same way as Uniswap V2, Kaiko handles data collection of Sushiswap's activities using the same methodology. Kaiko directly extracts trade data from swap events on Sushiswap. Each pool can be identified by its pool address, a unique string of characters which identifies a pool on decentralized exchanges and can be considered as the equivalent of a token pair on a centralized exchange.

Curve Finance (V1 + Factory Pools)



- **General:** optimized for stablecoin swaps
- **Automated Market Maker:** sum and product market maker model
- **Pools:** 2+ tokens
- **Fees:** Fees are flexible, with an average of 0.04%

Background

Curve is a DEX optimized for swapping similar assets such as stablecoins or wrapped/staked versions of a token. Curve's unique AMM ensures lower slippage and fees and more efficient liquidity. The DEX also enables interactions with other DeFi protocols, such as the lending protocol Compound.

Swapping

All swaps are made between two tokens, although Curve pools can contain more than 2 tokens. The objective of Curve is to maintain the ratio of prices between similar tokens as close as possible to one.

Pools

Curve has three main pool types:

1. Plain Pools: using the simplest implementation of the StableSwap invariant, containing 2+ tokens.
2. Lending Pools: two or more wrapped tokens (cBTC, aBTC etc.) are paired against one another, while the underlying is lent out on some other protocol, such as Compound.
3. Metapools: enabling the exchange of 1+ tokens with the tokens of one or more underlying pools.

Anyone can create a pool on Curve. However, only pools deployed by Curve are displayed on Curve's user interface. Some exceptions exist, such as Factory Pools, which are deployed by others but verified by Curve and thus promoted to the flagship Curve UI.

Prices and Liquidity

Curve's unique AMM does not require LPs to deposit balanced ratios of tokens, even giving the ability to deposit a single token. LPs earn fees in the form of LP tokens and the $CRV token, which can be used for staking and governance. The AMM, using what is known as the StableSwap invariant, proposes a price deterministic model whose objective is to achieve a compromise between low slippage and stable/balanced pools. The StableSwap invariant combines the sum invariant and the product invariant. The first has the potential of offering perfect linearity and thus total lack of slippage, while the other has the advantage of rebalancing pools when needed. To magnify the low slippage portion of the price determination equation, Curve amplifies the constant sum part of the formula with the amplificator coefficient

Data Collection

Data collection for Curve is unique due to how smart contracts are structured. For other DEXs, liquidity pools are all part of the same smart contract, so data collection only requires connecting to one smart contract. For Curve, each liquidity pool has its own smart contract, which means that in order for data providers to capture swap events for all pools, they need to connect to all individual pool smart contracts, which Kaiko does.

Balancer (V1 + V2)

- **General:** enables liquidity providers to gain single-asset exposure
- **Automated Market Maker:** Constant mean market maker
- **Pools:** up to 8 tokens in shared, private, or smart pools
- **Fees:** custom fees from .00001% to 10%

Background

Balancer is an AMM protocol that mainly differs from others on its pools' weightage characteristics and the ability for liquidity providers to provide only one asset in a pool (sing-asset exposure) instead of all the pool's assets. The main difference between Balancer V1 and V2 lies in how assets are stored: V2 holds all assets in a single vault, separating the AMM logic with the custody of assets. V2 now possesses the majority of liquidity.

Swapping

Balancer allows pools to consist of 2 to 8 tokens, each token with a different arbitrary share of the pool (called weight), that can range between 2% and 98%. This directly impacts the way prices are determined based on this AMM, which takes into account weights and transforms the basic constant product 50/50 AMM (***x*y = k),*** to a constant product AMM where prices are totally dependent on pre-determined (not necessary equally distributed) token weights : *krkwk*. For a full explanation on Balancer's AMM, click here. This gives a major importance to arbitrage : With fixed weights and the ability for liquidity providers to have single–asset exposure, market efficiency can only be achieved by arbitrageurs that will take profit from price discrepancies between the pools.

Prices and Liquidity

Smart Order Routing System
To route trades among a multitude of pools, Balancer uses a system called Smart Order Routing (SOR) that sources liquidity from multiple pools to automatically offer the best available price between two assets.

Custom Transaction Fees
In contrast to other protocols, Balancer offers the possibility to pool creators to choose custom transaction fees ranging from 0.00001% to 10%, received by liquidity providers.

Data Collection

Balancer V1 proposes three types of pools: (i) shared pools (also called public pools) (ii) private pools, (ii) and smart pools. All three types of pools are however managed by the exact same smart contract, but smart pools also use another specific contract above the main contract, whose role is to allow smart pools creators to manage the pool's flexible parameters over time. However, the swap events are always read through the main basis contract, for all types of pools. For V2, all events are extracted from the Vault contract.

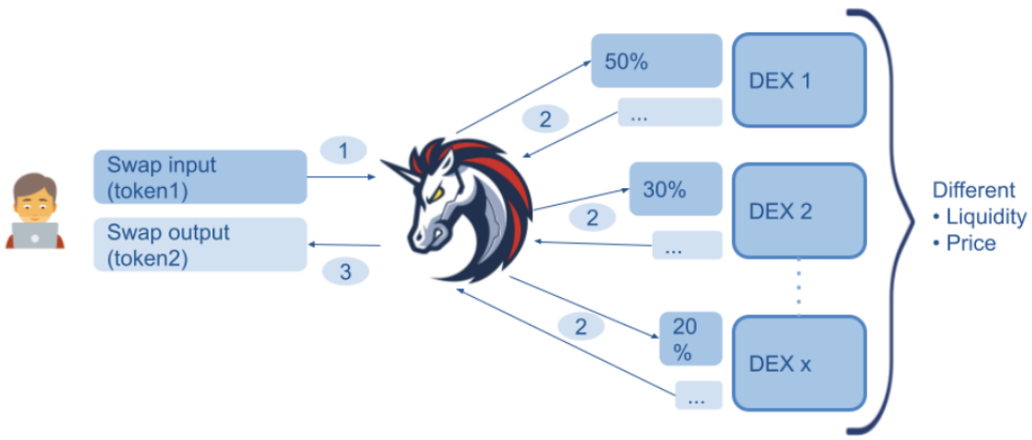
1inch

Background

1inch Aggregator is one of the major DEX aggregators on Ethereum, meaning that it will use a variety of DEXs to find the best execution for a user. As a result, a single trade is split over one or several DEX liquidity pools. The final swap is executed using 1inch Aggregator, but is built using a path of other swaps that use liquidity from other DEX protocols.

Transactions ≠ Trades

Trades that make the final 1inch swap possible share the same transaction hash since they are part of the same block and transaction on Ethereum. A transaction hash is a unique string of characters given to every transaction that is verified and added to the blockchain. On decentralized exchanges, multiple trades can have the same transaction hash if they are part of the same block. Kaiko's trade data for 1inch outputs the main 1inch transaction, and the underlying trades routed through DEXs.



Data Collection

In financial theory, a trade comes before a transaction, which comes before settlement. When considering operations happening on DEXs, trades and transaction steps are closely related, and settlement is made on-chain. A transaction executed on a DEX aggregator like 1Inch is composed of several trades. Consequently, each trade associated with a single transaction can be identified by a single transaction hash in Kaiko's trade data. A trade can therefore be defined as an operation between a sender and a receiver, that concerns a pair of tokens, and that implies a trading amount at a given price. As an illustration, the below shows a USDT/WBTC transaction executed through 1inch Aggregator protocol, using Uniswap V2 and V3's available liquidity. This transaction is made of two trades recorded at the exact same timestamp:

1. A USDT-WETH trade (swap) on Uniswap V3's associated liquidity pool (registered as a trade on Uniswap V3 tick-trades data, and recognizable with the transaction hash).
2. A WETH-WBTC trade (swap) on Uniswap V2's associated liquidity pool (registered as a trade on Uniswap V2 tick-trades data, and recognizable with the transaction hash).

These trades allow the instant creation of a USDT-WBTC transaction on 1Inch. Kaiko's data would contain information about this transaction, in addition to the two trades spread across Uniswap V2 and V3.

VIII. Appendix

Determining Trade IDs

For decentralized exchanges, Kaiko generates a custom Trade ID that incorporates 3 pieces of information: the the pool address, the transaction hash and the log index. An example trade ID is included below, with each of the 3 pieces of information divided by a '-'.

"trade_id":
"0x397ff1542f962076d0bfe58ea045ffa2d347aca0-
0x45a4db43b2cabcf01f85c87784950f69ff38eb306a74dbf84f43cc89933e456a-0x61"

pool address | transaction hash | log index

Pool address: 0x45a4db43b2cabcf01f85c87784950f69ff38eb306a74dbf84f43cc89933e456a

The pool address is a unique string of characters which identifies a pool on decentralized exchanges. Liquidity pools can be composed of two tokens or more, depending on the protocol, and there can also be multiple liquidity pools for the same pair/grouping of assets. This means that the only way to determine where a trade originated is by looking at the pool address. The pool address included in a trade ID differentiates trades that are part of the same pool on protocols in which pools have more than two base tokens or for which multiple pools for the same asset grouping exist.

The transaction hash: 0x397ff1542f962076d0bfe58ea045ffa2d347aca0

A transaction hash is a unique string of characters given to every transaction that is verified and added to a blockchain. On decentralized exchanges, multiple trades can have the same transaction hash, which means that not every trade can be easily differentiated. For example, on Uniswap V2 and V3, if two ERC20 tokens are not paired directly, they can still be swapped as long as a path between them exists. Each step of the multi-step swaps are considered trade events and share the same transaction hash but a different log index (described below). Thus, to differentiate trades, you need to have a combination of both the transaction hash and the log index.

Log index: 0x61

Each block on the Ethereum blockchain is divided into log records that include the transactions. Log records are indexed using log indexes, which can be used to describe an event within a smart contract like a token transfer. The combination of transaction hash and log index is thus unique to each log and allows us to differentiate trades that are part of the same transaction.

Pair Generation Methodology (Pools ≠ Pairs)

For DEXs, the notion of base and quote assets is not applicable. For example, if someone swaps (sells) WETH for USDT, the pair extracted will be WETH-USDT. However, if someone swaps (sells) USDT for WETH, the pair extracted will be USDT-WETH. However, those two pairs have to be considered as one, since they are composed of the same tokens.

To solve this, Kaiko takes each of those token's addresses and orders them alphabetically to determine which token will be the base and quote. For example, USDT's ERC20 token address begins with 0xdac1 and WETH's ERC20 token address is 0xc02a. In this case, since alphabetically 0xc comes before 0xd, the pair's structure will be WETH-USDT.

Thus we combine trades marked WETH-USDT and USDT-WETH into one WETH-USDT pair by inverting data related to the initial USDT-WETH pair:

- newAmount = previousAmount * previousPrice
- newPrice = 1 / previousPrice

Pair Generation Applied to Pools With >2 Tokens

In October 2021 the 3pool registered the most trades on Curve. As its name suggests, this pool is composed of three tokens: DAI, USDT, and USDC. Using the pairs generation methodology described above, Kaiko split this pool into 3 pairs that together represent all pair possibilities in the pool:

Curve 3Pool - 0xbebc44782c7db0a1a60cb6fe97d0b483032ff1c7

DAI-USDC

DAI-USDT

USDC-USDT

To find the total trade volume for the 3Pool, you would then take the sum of trades for all three pair combinations.