

Please note this is a rough draft/outline of an article about UniswapX. I will make edits in real time. The article breaks down the White Paper focusing on a high level summary, reviewing the protocol architecture, and then diving into order types. Finally, I add some analysis as well as resources for review. Feel free to discuss below.

## TLDR

- UniswapX enables off-chain counterparty discovery with on-chain settlement
- The protocol uses dutch orders to facilitate competitive pricing for swappers via various sources of on and off-chain liquidity provided by Fillers
- Abstracts fee payment, reduces MEV, enables cross-chain swaps, including fast L1<>L2 asset movement
- Compliments Uniswap V4's hooks approach

## Summary

UniswapX is a Dutch auction-based decentralized trading protocol using signed off-chain orders that are executed and settled on-chain. The protocol provides swappers with a gasless experience, MEV protection, and access to arbitrary liquidity sources. Swappers generate signed orders ("intents") which specify their preferences, and fillers compete using arbitrary fill strategies to satisfy these orders. UniswapX does not specify how fillers fill swappers' orders: liquidity can be sourced from a combination of on-chain liquidity venues like Uniswap or other DEXs, off-chain liquidity, or from other UniswapX orders. Multiple orders can be bundled into the same transaction, which can also execute other actions atomically on-chain.

- Outsources routing and batching to a permissionless set of Fillers who can route orders to a combination of on-chain and off-chain liquidity
- Trades use Permit2 executable off-chain signatures, which allow swappers to pay tx fees implicitly as part of their swap thereby avoiding needing the chain's native token for gas fees.
- Swappers never pay gas costs for failed swaps
- Internalizes MEV, reducing value lost by returning any surplus generated by an order back to the swappers in the form of price improvement.
- Supports cross-chain trading, allowing swappers to trade assets on an origin chain for desired assets on a destination chain.

## Architecture

[

1490x762 174 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/dbabdd1c50214e0a48e44f4641b00cdc01951184.jpeg)

Source: [UniswapX](#)

- Swapper
  - users of the application who have intents. Always receive the best possible execution on their orders
- Fillers
  - a combination of MEV searchers, market makers, and/or other on-chain agents who pick up user orders and send them to the reactor contract. By submitting swappers' orders on-chain, fillers pay gas fees on their behalf. These costs are then recouped by factoring gas fees into the execution price
- [Reactor](#)
  - this contract is responsible for checking that the execution of a trade against the filler's strategy matches the parameters users expect, and reverting trades that do not
- Executor
  - Order fill-contracts which fill

UniswapX orders. They specify the filler's strategy for fulfilling orders and are called by the reactor

## Single Chain Tx Lifecycle

1. User signs order specifying their preference
2. Filler listens to the API and picks up the order submitting it to the relevant reactor contract
3. The reactor contract calls the Filler's executor contract which specifies the fill logic
4. Once assets have been sourced, the executor sends assets to the swapper, and the swapper pulls funds from the swapper's address
5. The Reactor checks that the order's conditions have been met

## Dutch Order

UniswapX uses an order type, a Dutch order, which closely resembles a Dutch auction. The decaying nature of Dutch orders creates a competitive market among fillers to find the best possible price for swappers as soon as possible while keeping some small profit margin for themselves.

Dutch orders execute at a price that depends on the time of its inclusion in a block. The order starts at a price that is estimated to be better for the swapper than the current estimated market price. The order's price then decays over time until it hits the worst price the swapper would be willing to accept. Fillers are incentivized to fill an order as soon as it is profitable for them to do so. If they wait too long, they risk losing the order to another filler willing to take a smaller profit.

This is similar in concept to a fee escalator design put forth in [EIP -2593](#), [the Agoric papers](#), and [here](#).

## Order Parameters

- Input token
- Output token
- Input (output) amount
- Starting output (input) amount
- Minimum output (input) amount
- Decay function
- Claim deadline
- Authorization for the UniswapX reactor contract

## Example

[Apriori wants to swap 1 ETH for the most USDC](#) they can receive. Apriori requests a quote from (potential fillers) Bengt and Carlo:

- Bengt offers to buy Apriori's ETH for 1,000 USDC
- Carlo offers 999 USDC
- Apriori can also swap their 1 ETH directly through Uniswap v3 for 997 USDC

Apriori accepts Bengt's quote for 1,000 USDC, and signs the order. This order includes a maximum value (set by Bengt's quote of 1,000 USDC) and minimum value of 997 USDC (set by the Uniswap smart order router API). Bengt could fill Apriori's order using his own USDC or by routing Apriori's 1 ETH to a variety of on-chain liquidity venues (Uniswap V3, V2, Curve, Sushi). Bengt decides to fill Apriori's order using some of his own USDC, and sends Apriori 1,000 USDC in exchange for their 1 ETH.

Alternatively, If Bengt had decided to fall through on his offer, Apriori is not required to submit a new order and signature. Instead, their existing order automatically updates, offering their 1 ETH to any filler who can give them 999 USDC in return. One block has passed and none of the other fillers are willing to fill Apriori's order at 999 USDC. After another Ethereum block (12 seconds) has expired, Apriori's 1 ETH becomes available for 998 USDC. Suddenly, Carlo realizes that he can fill Apriori's 1 ETH sell order for 998 USDC while still collecting a 1 USDC profit by sending Apriori's trade to a combination of Uniswap v3 and Sushiswap. Carlo sends Apriori 1 ETH to Uniswap v3 and Sushiswap on Apriori's behalf, returning 998 USDC to Apriori and keeping the remaining 1 USDC output.

This price improvement over Uni V3 is achieved by sourcing liquidity from multiple venues and capturing surplus value from advanced order routing that would otherwise be left for a MEV searcher to capture.

## Order Parameterization

The UniswapX Protocol does not enforce a specific decay function. One way of parameterizing the Dutch order starting price is to poll a selection of fillers through an off-chain Request For Quote (RFQ) system.

In order to incentivize this network of fillers to offer their best possible price, UniswapX allows orders to specify a filler that receives the exclusive right to fill the order for a brief duration, after which the Dutch auction begins and any filler is able to execute the order.

An RFQ system may benefit from using an accompanying reputation or penalty system to limit abuse of the free option that this exclusivity provides fillers and to ensure that swapper user experience does not suffer.

## Cross-chain Orders

The UniswapX Protocol can be extended to support cross-chain trading, where a swapper trades assets they hold on an origin chain for desired assets on a destination chain. This is possible as long as there is a message passing protocol between any two chains. Swapping and bridging are combined into a single action, removing the need for swappers to interface directly with bridges, maintain gas tokens on either chain, or wait for settlement delays. Swappers can specify that they receive native assets on the destination chain, rather than a bridged asset. For example, USDC on mainnet can be swapped directly for ARB on Arbitrum. Swappers do not assume any exposure to a bridge when swapping native assets, and fillers only take on bridge risk while rebalancing between chains through bridges.

### Simplified Cross-Chain Orders

To initiate a cross-chain order, the swapper signs an off-chain order that includes the same parameters as a single-chain order described above, alongside the following parameters:

- Settlement oracle
- a one-way oracle that can attest to events occurring on some destination chain.
- Canonical rollup bridge
- Light client bridge
- Third party bridge
- Canonical rollup bridge
- Light client bridge
- Third party bridge
- Fill deadline
- time before which the order must be filled on the destination chain
- Filler bond amount and filler bond asset

— the bond that the filler must deposit on the origin chain

- Proof deadline

— the time before which the filler must prove their fill on the origin chain

### Cross-Chain Tx Lifecycle

1. The Filler executes an order by first claiming the swapper's order and submitting the filler bond to the origin chain reactor contract, and then by transferring assets to the swapper's address on the destination chain via the destination chain reactor contract.
2. The reactor contract records that the order was filled before the fill deadline and relays a message through the settlement oracle back to the reactor contract on the origin chain confirming fulfillment of swapper's order.
3. The swapper's assets, alongside the bond, are then released to the filler on the origin chain. In the event that the filler does not execute the order before the proof deadline, the swapper receives both their input assets and the filler's bond from the reactor contract on the origin chain.

## Optimistic Cross-Chain Orders

Some settlement oracles may be slow or expensive to use. For example, executing swaps from one optimistic rollup to another may be expensive for fillers, taking over seven days and involving at least one transaction on the L1 before they are allowed to take custody of the swapper's input tokens and their initial bond. An optimistic cross-chain protocol can alleviate these settlement delay issues, effectively constructing a fast and cheap bridge on top of any slow bridge. The optimistic protocol includes the same parameters as the simplified protocol, and these additional parameters:

- Challenge bond amount and challenge bond asset
- the amount that a challenger must post as a bond on the origin chain.
- Challenge deadline

— the deadline before which a challenger can challenge a fill. This must be before the proof deadline.

### Optimistic Cross-Chain Tx Lifecycle

In the optimistic case everything is the same as described above except, if the filler fills the swapper's order on the destination chain before the fill deadline, and no one challenges the fill before the end of the challenge period, the filler receives the swapper's funds, alongside their filler bond, on the origin chain.

### Challenge game

- Anyone can challenge the filler after the fill deadline but before the challenge deadline expires
- Challenge is made by using the reactor contract on the origin chain
- If the fill is challenged, the filler must provide a proof before the proof deadline from settlement oracle
- If the filler proves that they filled the order before the proof deadline they receive the challenger's bond
- If the filler fails to provide a valid proof, the filler's bond is split b/w challenger and swapper
- The swapper's funds are returned to them on the origin chain

### Fees

UniswapX has the same fee switch present in Uniswap V2 which governance has control over. The DAO can vote to turn on a 5 bp fee for each swap of a given token pair. The fee would need to be activated on each chain individually.

### Analysis

It's clear that Uniswap has much order flow that comes directly to their front-end. It likely will not be difficult to bootstrap UniswapX especially if Fillers provide consistently better fills than passive LPs. Not only was this a competitive move to match CowSwap's product but also it's evident that many applications can be built on top of UniswapX.

Discussion on different twitter spaces (AMM/MEV community) this week focused on how UniswapX would be bad for passive LPs who would now strictly eat unwanted or 'toxic' orderflow. I tend to agree with this analysis at face value. However, with custom Hooks from v4, this does not need to be the case. Toxic flow can often be identified by the timing with which it arrives. There could exist a hook which raises its fee dynamically if it identifies toxic flow for example. Note I'm handwaving a bit but it is my understanding that V4 hooks can be designed in such a way.

Though it is possible that this new off-chain liquidity system accelerates Builder centralization because it is likely that many of the existing Builder-searcher entities will be incentivized to act as Fillers. These entities often have access to inventory from a variety of sources as they regularly compete in top of the block CEX-DEX arbitrage games. One could see this driving Builders on Ethereum to become rollup sequencers for similar reasons.

### Conclusion

The protocol is ground-breaking for Ethereum and has a lot of similarities to how one would envision an Anoma DEX. However, the protocol lacks a way to provide users information flow control and does not have a unifying message passing protocol to rely on for cross-chain orders. As of writing it appears the cross-chain messaging is only compatible with EVM chains.

### Resources

- [UniswapX White Paper](#)
- [Uniswap X Repo](#)

- [How does UniswapX work?](#)
- [UniswapX Docs](#)
- [Introducing Uniswap X](#)
- [EthCC Reveal - On-chain Trading](#)
- [UniswapX revealed - A game changer for DeFi](#)
- [UniswapX is a Powerplay](#)
- [Uniswap V4 White Paper](#)
- [How to Integrate with Permit 2](#)
- [UniswapX and the DAO](#)
- [Next Gen AMMs: Hooks, Intent Based Liquidity & More](#)
- [LVR Reduction; The Biggest Open Problem in DeFi](#)