TLDR; Usage of coincidences of wants in cross-chain transfers might help reduce gas fees and wait time

Background

We are building Fetcch, we have an inbuilt cross-chain settlement layer that works based on stablecoin liquidity pools and DEX aggregators that enables any token to any token transfers across multiple blockchains.

Cross-chain CoWSwap?

CoWSwap model is simple, they collect orders and share them with multiple solvers, solver's job is to generate a batch of transactions to be executed on-chain which helps user swap their tokens. It can decide to swap the user's tokens in 2 ways, using the DEX aggregator or finding CoWs or Coincidences of want between orders.

What is Coincidences of want?

Coincidences of want or CoW is when 2 buy and sell order match in a batch, for example, a batch contains 3 transactions

- User A wants to swap 10 MATIC → USDC
- User B wants to swap 13 USDC → MATIC
- User C wants to swap 13 DAI → USDT

Now, with the above intents,

User A wants to sell MATIC and buy USDC and User B wants to buy MATIC and sell USDC. This is a CoW, instead of routing their transactions through the DEX aggregator (which will also cut fees for providing liquidity for swap), we will just do a simple transfer between both users, which will save them DEX, Gas, and other fees

[

Cross Chain CoW Swap

3808×1268 102 KB

](https://ethresear.ch/uploads/default/original/2X/4/49d281227207949fca336a2ae48795a7fca73616.png)

CoWs aren't just direct transfers, they can also match ≥ 2 orders with each other if we modify the above orders

- User A wants to swap 10 MATIC → DAI
- User B wants to swap 13 USDC → MATIC
- User C wants to swap 13 DAI → USDC

Now, User A will sell their MATIC to User C, User C will their USDC to User B, and User B will sell their MATIC to User A, so now they are fulfilling each other's orders.

This resembles the order book architecture.

How will cross-chain CoWs work?

There are X major problems in cross-chain CoWs

- In a swap, all of the execution is on the same chain, so we can execute transactions atomically, this will help us verify if User A has sent their tokens or not, but in a cross-chain context, we can't atomically execute transactions, and cannot verify if a user actually has sent tokens on their blockchain or not
- This will all be a volume game, if there are a lesser number of transactions per batch, it directly will affect the number of CoWs that we will find, so more users will use the bridges and add those batch auction times to bridging time, meaning that, if volumes are low, we can't provide the same experience as bridges provide
- This is not a tech problem, but it is very important in this context
- This is not a tech problem, but it is very important in this context

There are some possible solutions for the above problems, let's discuss them further

Conducting Solver Auction

We will need to conduct an auction between various solvers who have created order bundles that will be executed on-chain.

CoWSwap handles this auction off-chain through a centralized server and so do various other similar protocols because having a decentralized matching algorithm is too hard to scale on a blockchain

The matching algorithm will still be run off-chain on a server and doesn't need to be decentralized in the traditional style of a smart contract or some consensus-based state machine if you think it through, this problem resembles PBS's problem, instead of giving the power of block building to a single entity, we will give the power of building to block to anyone and those anyone can propose to the proposer (previous block builder) to add it to the blockchain. In a similar way, solvers will receive a list of swaps and will solve it locally and then propose it to the auctioneer directly, auctioneer will then choose the best bundle with probably the most valid CoWs and execute it on-chain.

We need to decentralize the auction and auctioneer part but for the sake of PoC, we will be running it on a centralized server, but possible solutions include PoS roll app or app chain, smart contract deployment on very cheap L2/L3, using intent-centric protocols like anoma, suave etc

Some solutions for cross-chain value transfers

Hashed Timelock Smart Contract (HTLC)

HTLC is a pretty old and tested practice, this was one of the first cross-chain solutions to be built, and it worked something like this

- User A generates a private key and hashes it, then it will create a contract by passing hashed key and some tokens to it on Chain A
- User A shares the hash with User B and User B does the same on Chain B
- User A uses the private key and claims the token deposited by User B on Chain B
- Now, the private key is publically available on the blockchain, User B uses that and claims its token on Chain A

HTLC provides kind-of atomic transactions, here, atomicity isn't dependent on the system, but on the user.

Some disadvantages of this issue with regard to CoWs

- Somebody can front-run your claim transaction using the already publically available private key
- Increased gas fees for setting up HTLC and then claiming the tokens
- The token settlement will become complicated if CoWs are matched between ≥ 2 orders

AMBs or Oracle based solution

We can Arbitrary Message Passing Bridges (AMBs) to send cross-chain messages across blockchains to lock and unlock tokens.

- User A will deposit tokens on Chain A
- We will send a cross-chain message from Chain A to Chain B proclaiming that we have received tokens
- If User B has already deposited tokens on Chain B, those tokens are sent to User A, if not, then we wait for User B to send tokens and upon receiving them, we directly send those tokens over to User A
- Chain B sends a cross-chain message to Chain A that it has released those tokens to User A and Chain A does the same and releases tokens to User B

It is very easy to implement and guarantees more security than the HTLC solution, but it has its own issues

- AMBs cost money, there are bridge fees + gas fees that the user/infra has to provide, as we are sending 2 cross-chain messages, the cost to operate this kind of solution becomes a lot higher
- If by any chance, AMBs fail to deliver the message, we will need to retry (if the AMB layer doesn't have a mitigation for that)

Centralized Watcher

We can have a specialized server that watches over the cross-chain messages and makes sure both parties receive their share of the tokens

- All users will deposit their tokens with their order id in a smart contract

- Our watcher will match orders and release tokens accordingly in a single transactions

We are batching a lot of transactions into one, so gas fees will be much lower and execution will be a lot faster, but it has its own issues

- Settlement depends on centralized watcher, which obviously has centralized issues attached to it

Intents

We can also use an intent-centric protocol like SUAVE or Anoma for matching CoWs and executing them in a decentralized manner, but SUAVE is in the research phase and Anoma is still in the development/testnet phase, so we will need to wait for them

Apart from these above solutions, we can also build something using zk or fraud proofs, which can help verify the deposit of assets from Chain A on Chain B in a non-trusted manner

For volumes, there are no other solutions than having a greater distribution and providing possibly the lowest fees and fastest execution possible, in the coming weeks, we are going to build a small PoC for this and lookout for possible fee patterns which will help us understand its viability

Not all orders are complementary to one another, there will be some orders which can be partially fulfilled by CoWs but will need some liquidity networks to fill the remaining order, in this case, how do we ensure the user receives all of the tokens in a single transaction, it doesn’t make sense to send 60% tokens in one transaction and other 40% tokens after 5 minutes in another transaction.

A solution is to use an order id and make our contract on the destination chain aware of how many tokens we need to satisfy that order id, if it receives all of the tokens related to the order, it releases the token, but this introduces another transaction which can be batched and done by some service (like how wormhole does it)

Ref blog - How to think about Cross Chain CoWSwap? | Fetcch