

tldr; intents are hot right now, but no one has idea what it is and how to decentralize it, we will focus on that

[

proposed-intent-mempool

1720×1057 79.5 KB

](https://ethresear.ch/uploads/default/original/2X/1/1cb6f9f4f1debd9b3fcee3896d982ebd2c417837.png)

What are Intents?

Plain and simple, intents are nothing but predicates/conditions. User/dapp submits their “conditions” to a “computer” called solver. Solver understand these conditions and uses this constraints to design a transaction which will pass all of these “conditions”. Ordering of conditions matter here.

Hurdles in decentralising them

Some major hurdles

- Usage of LLMs to process natural language to Intent
- Lack of proper structure for intents
- Current solver design
- Closed pools of intents
- Auction Layer

Usage of LLMs to process natural language to Intent

LLMs like ChatGPT or Langchain are heavily used by recent projects to convert natural language to their internal structured intent, it will be very hard to decentralize this part, so we can leave this upto the dApps and users

Lack of proper structure for intents

Any system that is trying to be decentralized has faced this issue, if all of the nodes(computers) in a network don't agree on a structure(s), then they can efficiently communicate between each other and this might lead to dangerous behaviour.

A proper structure for intents is required for it to be decentralized, this is the main part of decentralizing intents. But how can we structurized intents?

Structure For Intents

What are the important parts that define a intent from any other type of data structure?

- Conditions
- User's intent
- will be a dictionary
- User's intent
- will be a dictionary
- Pre conditions
- this should be true before solver tries solving the intent
- for example - i want to swap 100 ETH only if price of ETH > 100\$ and fees should be lowest
- here, want to swap 100 ETH & fees should be lowest is condition and ETH > 100\$ is pre condition
- will be a dict
- this should be true before solver tries solving the intent
- for example - i want to swap 100 ETH only if price of ETH > 100\$ and fees should be lowest

- here, want to swap 100 ETH & fees should be lowest is condition and ETH > 100\$ is pre condition
- will be a dict

Conditions are checked at runtime while solving the intent and Pre conditions are checked before even solving the intent

Intents will be a very open ended structure, so we can't really encapsulate every type of intent in a single structure, so we will have to go multi-structure with conditions and pre-conditions

We will have types of intents like swap-

, it can define its own types of conditions and pre-conditions, something like this

SCENARIO --> A user who wants to swap 100 MATIC (POL) -> USDC (ETH), only wants slippage up to 1% and should at least receive 100 USDC for 100 MATIC, current price 1 MATIC -> 0.99 USDC

CONDITIONS --> amountIn - 100 tokenIn - MATIC (POL) tokenOut - USDC (ETH) slippage - 1%

PRE-CONDITIONS --> usdcPrice - >= 1 MATIC maticPrice - >= 1 USDC

TYPE: swap-

Likewise there can be an intent type for payments

SCENARIO --> A user wants to send 100\$ of ETH to vitalik.eth

CONDITIONS --> tokenIn - ETH amountInToken - null amountInDollar - 100 receiver - vitalik.eth sender - someone.eth tokenOut - null amountOutToken - null amountOutDollar - null

PRE-CONDITIONS -->

TYPE - payments-

Closed Pool of Intents

Current Intent implementations are all centralized and so are their mempools, each application has their own storage for storing intents which only their solvers can access, limiting the decentralization and generic nature of the whole ecosystem

We need open mempools which supports various types of above intents and any solver can run or connect to a mempool and start solving them

Open Mempool

A open mempool is needed for intents if we want to promote decentralization.

Operators running this mempool will be able to define what type of intents they want to support for storage or they can store every type of intent, its upto the operator, we did this because some operators might only want to run mempool supporting their custom types, so that any solver can solve them.

[

how-mempool-will-work

2072x852 78 KB

](<https://ethresear.ch/uploads/default/original/2X/7/7415c345dd1b079f5226c3060d6f8e679999d2c7.png>)

Mempool can also connect to a P2P gossip network to gossip received intents with other mempools, this will probably be a libp2p or waku implementation in practice.

Mempool will usually be run alongside solver, so solver can get fast access to intents they care about.

Current Solver Design

Solver's are vital part of intents, they solve an intent by calling some APIs or contracts and build transaction(s) for it which can be executed by the user

These solvers are currently centralized and support only handful of types of intents, we can take this structure and make it pretty decentralized.

New Solver Design

Solvers can run a mempool or can connect with some open mempool, it will subscribe to certain types of events which it supports like with above case, a solver can subscribe to swap-

and it will then receive a stream of these events.

If a solver supports an intent type, meaning that they can also solve that intent, then it will subscribe to their pre-conditions by some internal logic and then try to solve the intent based on conditions

, the implementation of how to solve the intent can be decided by the developer but at the end of it we should receive transaction(s) or userops which can be executed by the user

[

how-solver-will-work

1964x687 96.6 KB

](<https://ethresear.ch/uploads/default/original/2X/6/68ab4d03692453aa40352769a4d7f8230e9de38c.png>)

Auction Layer

Currently, no intent implementation uses an decentralized

auction layer to settle bids between multiple solvers, CoW Swap does but its closed and centralized.

What are some good auction layers that are decentralized and as well as usable enough?

- Existing Blockchains (Ethereum, Polygon, L2s, etc)
- New Blockchain (OP Stack or any other kind of rollup or L1 like SUAVE)

In short, the answer is blockchain, now it is upto the builder where they want to have their auction layer but because of our type based architecture, there can be multiple auction layers supporting various types of intent types on various blockchains.

Once everything is done, intent is added to mempool, its solved by a solver, user has accepted a bid on auction layer, they can directly receive their built transaction in their wallet as a notification which they can approve using Pull Payments

In my previous post, I was discussing about [Cross chain CoW Swap](#) and while building that, I thought to myself, this can be generalized for any type of intent that we want, and then I wrote this post, now back to code!

Would love to have a healthy discussion here and understand the possible flaws in this system!