*This is Part 1 of a 3 part series on Intents, byMike Calvanese and the team at Brink.

Intents are going viral. They are one of many UX improvements made possible by account abstraction, another hot topic discussed in Vitalik's talk at EthCC. Intents let users define their desired on-chain outcomes and outsource the hard technical work of achieving these outcomes to 3rd party solvers that interact directly with networks and protocols on the user's behalf. Ultimately this abstraction layer will make Web3 apps feel like normal apps. It reduces the technical learning curve that exists today and will help us onboard millions of new users.

The paradigm shift toward intents means that future users won't need to concern themselves with technical tasks like submitting transactions, paying for gas with ETH, mitigating MEV on AMM's by setting slippage limits, aggregating token swaps for optimal path routing, or bridging assets across networks.

An easy way to understand intents is to think of them as declarative expressions of "what" a user wants to happen, rather than imperative steps of "how" something should happen.

The Current State of Intents

Intents are not a new concept. In the days before Uniswap and AMM's, Ethereum projects like EtherDelta and 0x provided intents-based order books. NFT marketplaces have used signed intents for NFT listings and offers for years. Newer systems like CoW Swap and UniswapX now provide more advanced intents-based infrastructure for ERC20 limit orders. Looking at the current landscape, the term "intents" might seem to be synonymous with "limit orders" because the only intents that are widely supported in Web3 today are "I want X and I'm willing to pay Y".

Intent architectures for limit orders are generally simple and built for a single-purpose: the intent to swap at a better price than what the current market offers. Users sign their limit order intent and solvers look for routes to "fill" the signed intents using AMM's, other liquidity sources, or in some cases other intents. Solvers are incentivized to route-find because they get to keep whatever extra is left over after the user's intent is filled.

[

1712×1674 435 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/9fced1e823769069c64c986f8b9b3f68c51f2679.jpeg)

Limit order intent architecture above.

Many systems have been built to support the limit order use case, but as more advanced intent-based tooling is built we'll start to see more general purpose architectures that will enable more robust use cases.

Some examples of general purpose intents systems are Anoma and Flashbots SUAVE, both actively under development. They will provide intent gossip layers where users broadcast signed intents to gossip nodes. These will be intent specific chains that facilitate the connection between users signing the intents and solvers executing them across different networks.

Another example is Brink, a solution for creating composable intents. Brink allows users and developers to build, sign, and solve intents based on conditional states, across multiple EVM-compatible networks.

The Future of Intents

Let's explore a few ways Web3 intents could evolve beyond simple limit orders. Here are a few new concepts:

- Conditional Intents:

Allow an action when one or more conditions are met

- Continuous Intents:

Express a desire to take a repeated action

- Multi-Step Intents:

When one intent is solved, this opens one or more new intents

- Intent Graphs:

The pathways formed by a group of related intents

Conditional Intents

Current Web3 apps only have one type of conditional intent: limit orders. In the future we'll have intents that can be composed with any number of conditions for any state, resulting in any action.

If you're a Web3 user, you have made many conditional decisions in the past. These could be expressed as intents, signed, and delegated to a 3rd party solver. Here are a few examples:

- Price Thresholds:

"Swap B for A if A/B price drops below X". In traditional finance (trad-fi) this is referred to as a "stop-loss"

- Governance Decisions:

"Sell token A if a governance proposal I don't support passes".

- Wallet Balances:

"If my hot wallet runs low on ETH, transfer more ETH from my cold wallet to my hot wallet".

- Time/Blocks Elapsed:

"If more than X blocks have been mined, transfer ETH to a recipient"

All of these could be signed as a single intent. Solvers would monitor these intents and take action on behalf of the user once the conditions are met. A system where users sign these conditions as part of their intents would force the solver to prove the condition through some on-chain state check.

Intents could be expressed with any number of conditions. For example:

"I want to buy 2 ETH with DAI when 1)

the ETH price is under $1,750, 2)

the average DAI yield for lending ETH is high, and 3)

a whale wallet that I'm watching has bought at least 10 ETH in the last 24 hours"

Users will be able to express their willingness to buy, sell, stake, transfer, or bridge assets when any number of complex state conditions are satisfied. All with a single signed intent.

Continuous Intents

Intents will offer a protocol-agnostic way to perform continuous actions. While intents today generally involve a single signature that corresponds to a single action (limit orders), we will soon see the advent of continuous intents.

Users take continuous actions today by depositing funds into protocols or exchanges. Here are a few examples:

- Cost Averaging:

"Buy ETH with DAI at market price once a month". Users generally use CEX's like Coinbase to accomplish this.

- Yield Compounding (AKA Re-Staking):

"Withdraw rewards in A, swap for B, and re-stake". This is a tedious process that involves multiple transactions and interactions across more than one DeFi protocol.

- Hot Wallet Top-up:

"I have a cold wallet with 50 ETH. Whenever my hot-wallet balance of ETH drops below X, transfer Y ETH from my cold wallet to my hot wallet". Also requires a lot of separate transactions.

- Payment Streaming:

"Transfer X USDC to a recipient address once every 2 weeks". Streaming payment apps support this, but require users to deposit assets into smart contracts

- Market Making:

AMM-based LP positions basically follow the same pattern as two inverted swaps that continue in an infinite loop: "Swap A for B whenever the price of A/B exceeds X, Swap B for A whenever price of A/B is less than X"

As seen in these examples, continuous actions today require users to deposit into specific protocols and submit many transactions. With intents, users could express their desired continuous actions with a single signature.

Cost averaging is a perfect example. A user who intends to cost average (buy or sell repeatedly over time) could express this as a signature that permits 1 ETH to be swapped for DAI at a price provided by a hard-to-manipulate ETH/DAI TWAP oracle, once every 50,000 blocks (about once a week on Ethereum mainnet), for a 1% fee (0.01 ETH). Solvers would monitor this intent and check for a few things:

1. Does the user's account have 1 ETH to swap?
2. Have 50,000 blocks been mined since their last swap?
3. Is 1% (0.01 ETH) enough to cover the gas cost of the swap?
4. After the solver pays gas for the swap, is there ETH left to cover the solver's operating costs, which could include risk of revert after losing a PGA (priority gas auction), or any number of other expenses? Is the potential profit worth it to the solver?

If all of these check out, the solver would send a transaction to fill the user's intent to make the ETH→DAI swap. As long as the user's account has enough ETH, a single signed intent for "cost averaging" could be monitored by solvers and result in a stream of continuous transactions. The user gets what they want without having to interact directly with an EVM network or a specific protocol.

Market making with intents is another example of a continuous and potentially infinite action that could be delegated to solvers. A user who wants to market make ETH/DAI might create an intent that permits a DAI→ETH swap at 1,800 ETH/DAI or less, and an ETH→DAI swap at 2,000 ETH/DAI or greater. With this intent, the user is attempting to lock in a 200 DAI profit every time the market fluctuates between the 1,800 and 2,000 ETH/DAI price point.

Think about this like two inverted limit orders, where if one is filled the other opens and vice-versa. The user could sign a single intent to swap at each price point. Solvers could theoretically fill an infinite number of these orders without the user every taking any action, as long as the market continues to fluctuate across these price points.

Multi-Step Intents

Intents could be composed of multiple steps. You could think about these intents like a state machine, where each transaction transitions from the previous state to a new state. The new state we transition to depends on the conditions defined by the previous state.

One simple example of a multi-step intent is the classic trad-fi Bracket Order. These can vary in their level of complexity, but simple versions are commonly found in trad-fi trading apps. With the composability of Web3 and multi-step intents, we could create extremely powerful bracket orders.

A Web3 bracket order intent could be expressed like this: "I want 1 ETH and I'm willing to pay 1,800 DAI. Once I've paid 1,800 DAI and have 1 ETH, I'm willing to hold that 1 ETH until I can sell it for 2,000 DAI. If the ETH/DAI price drops below 1,620 DAI, I'd like to minimize my loss by selling the 1 ETH, in which case I'll accept 1,600 DAI and I'll give a 20 DAI fee to the solver. If I'm still holding the 1 ETH when [random governance proposal] passes, I want to sell the 1 ETH for whatever the ETH/DAI market will give me, and I'll give 10% of the proceeds to the solver."

This is a relatively simple 3 step order with a few conditional triggers, but as you might imagine, these types of orders could be much more complex.

Intent Graphs

Relationships across user intents can form intent graphs. These graphs represent groupings of user defined conditions and actions that result in swaps, transfers of assets, or other on-chain actions. All of the previous intent examples are actually just names given to specific graph arrangements.

In the same way that a limit order book represents the current state of liquidity for a market, intent graphs could represent not only the current state, but also conditional liquidity under many different potential future states.

Consider an example: In a fictional market where users trade XYZ and ETH, users express intents to buy and sell XYZ under various conditions such as governance proposal results, mining of specific blocks, market price increases and decreases, or whether other intents have been filled.

his graph represents liquidity that exists now, and liquidity that would exist under possible future states. Intent graphs could extend across markets or even across different chains.

Conclusion

Intents are making headway with new developments happening every day.

Powerful Intents: Part 2

will focus on solver incentives.

If you would like to explore the concepts presented here further, come chat with me in the Brink Discord or reach out on Twitter.