

Introducing Brink Intents

Brink is building the most advanced intents protocol for Web3.

[We see a future where intents are much more powerful than they are today](#) With Brink Intents, users sign their desired outcomes instead of sending transactions directly to a blockchain. A single signature can delegate multi-action workflows to solvers on any network. Signatures are composed with any number of conditional state checks and single or cross-chain actions such as market swaps, limit swaps, and transfers. DApps and wallets that build on Brink Intents offer massive UX benefits to their users by abstracting away transaction submission and gas payments, and by simplifying many time consuming steps into one signature. It is a completely new paradigm for interacting with Web3, and it's powered by account abstraction.

Intents built with our protocol have three core properties. They are:

- Composable:

Actions such as swap, transfer, and bridge can be composed together into a single signed message allowing for infinite combinations of intended outcomes.

- Conditional:

Intents can be enabled or disabled by state conditions, such as market spot prices and deltas, wallet balance changes, governance voting results, or off-chain analytics brought on-chain via oracles.

- Continuous:

A single intent signature can permit continuous outcome solving over a period of time, such as the intent to swap 1 ETH for DAI at market price once every 50,000 blocks or at a specified time interval.

Our protocol lets users delegate any workflow to solvers within a single signature: from simple limit order routing to extremely complex multi-swap, state-conditional actions across different networks.

Composable Intents

Web3 users often have complex workflows that require many transactions to achieve their desired outcomes. Definition of these outcomes can be broken down into Outcome Primitives

. These primitives define a user's desired state, and solvers are incentivized to find the most optimal way to fulfill that state. Brink Intents supports outcome primitives such as:

- Swap:

"My desired outcome is to have Asset A, I'm willing to pay with Asset B"

- Transfer:

"My desired outcome is to move assets from one address to another"

- Bridge:

"My desired outcome is to move assets from the same address on one chain to another chain"

For all of these state-changing outcome primitives, the user signs what they want to happen, but it's up to solvers to deal with the complexity of where to swap, how to transfer, or what bridge to use.

Brink Intents also supports Conditional Primitives

. These primitives are used to determine which outcomes are currently allowed by a user. All conditional primitives can be represented as boolean state. Logic around questions like "has my swap been completed?" and "is the ETH/DAI price less than 1,800?" can be represented with conditional primitives, since these ultimately resolve to either true or false.

Conditional and outcome primitives can be composed together into signatures that delegate user workflows to solvers. Consider a simple workflow where a user wants to swap from DAI to ETH at a low market price and then swap from ETH back to DAI at a slightly higher market price. A user can compose this intent with two limit swap outcome primitives and a conditional primitive to check whether the first limit swap outcome has been solved. Solvers that understand how these composable primitives work can monitor and solve them independently.

In this workflow, the user only signs once, but solvers will have to “solve” this workflow with two transactions that would likely happen many hours apart from each other. A solver may have never seen this particular composition of primitives before, but the solver only needs to understand how to interpret and solve each primitive individually. Using the composable primitives that Brink Intents provides, an infinite set of workflows can be composed, signed, and delegated for solving.

State Conditions

Brink Intents can support conditional primitives for any on-chain state. State can represent anything from price thresholds to complex off-chain analytics data (provided this data is brought on-chain with an oracle). Conditional primitives check state and always resolve to either true or false.

Using a price condition as an example, a conditional primitive could check if the current TWAP (Time Weighted Average Price) on a Uniswap pool is less than the user’s signed price. The conditional primitive forces solvers to prove that the condition is true by reading the TWAP value and comparing it to the user’s signed price threshold.

The potential set of state is infinite. Brink Intents supports a subset of conditional primitives for state such as:

- Market prices and deltas
- Blocks mined / not mined
- Blocks / time elapsed
- Gas prices
- Address balances
- Intent outcomes, such as swap fill state

More complex state conditions can be integrated to support more powerful cases. For example:

- Account actions, such as swaps and transfers
- DAO proposal voting results
- DeFi data, such as liquidity or yield
- Any off-chain data brought on-chain

Our core protocol is un-opinionated about what state actually represents, so conditional primitives can be built around any state check that resolves to a boolean value. Ideally, conditional primitives will be built around state that is not easily manipulated by a centralized entity, but ultimately this trade-off is left up to users of the protocol.

Continuous Delegation

Intent-based protocols in Web3 today only support signed limit orders, which delegate a single action. Brink Intents enables Continuous Delegation:

a single signature can delegate a workflow with any number of outcomes over any period of time.

One example of continuous delegation is the intent to make a recurring purchase or sale, which is sometimes referred to as DCA (Dollar Cost Averaging). In the context of Web3, a user might want to swap 3 ETH for DAI. Instead of doing this all at once, they want to average the swap over 150k blocks. They’re willing to pay a 1% fee for each swap. This can be composed with standard Brink Intents primitives into a single signature. Solvers can use the same signature to solve the user’s desired outcome multiple times.

This particular intent to swap will end after about 3 weeks (assuming 12s block times), but intents can also be cyclical and continue indefinitely. Consider a market maker who has the intent to buy and sell on the ETH/DAI market at a range between 1,800 DAI and 2,000 DAI. Their intent workflow might look something like this:

This intent would only have to be signed once. It could be valid until canceled, or have an explicit expiry block, but otherwise it will be executed continuously by solvers in a loop for as long as market conditions allow, using the same initial signature.

User Experience Benefits

Brink Intents protocol lets users put complex on-chain workflows into motion with very little effort, instead of having to deal with direct blockchain transactions that require deep protocol knowledge, MEV awareness, gas, transaction risk, and lot of time. We believe that the composable intents layer is key to onboarding future users by removing a big part of the current learning curve, and reducing complexity for those already using Web3 applications. Brink Intents can benefit both existing and future users.

Users who primarily use EOA's

can delegate any of their existing transaction-based workflows to solvers.

Multi-Sig / AA Wallet Holders

can sign and approve continuous actions that execute over time, rather than having to define every individual transaction and coordinate signing when an action is needed.

DAOs

can create proposals that result in intent-based delegation instead of transaction based actions to ultimately reduce the number of proposals they need to approve.

Whether these users interact with Brink Intents directly using our builder tools, or indirectly through a Web3 application that has integrated Brink Intents, these benefits are the same.

Solver Incentives

In all intents-based systems, solvers need incentives to cover their operational cost and earn profit. To capture incentives, solvers need to compete with other solvers as well as generalized MEV bots running front-running strategies like sandwich attacks. The solvers themselves are part of the MEV ecosystem since they are attempting to extract the maximum value from every intent they solve. The most optimal solver incentive designs work by aligning solver incentives with optimal user outcomes, so that more MEV for solvers actually results in faster and better execution for users.

Brink Intents is designed to accommodate any solver incentive design. Our primitives use two common incentive structures: implicit and explicit. In both cases, fees are taken from the tokens involved in the intent and the user does not have to hold ETH or pay directly for gas.

Implicit Incentives

are used in our limit swap primitives. For a limit swap where the user is willing to pay 1 ETH and requires receipt of 2,000 DAI, a solver attempts to find a route that will return more than 2,000 DAI. For example, if the solver finds a route from 1 ETH to 2,050 DAI, they can take the additional 50 DAI to cover the transaction gas cost and earn profit. Solvers compete to be the first to find the most optimal route that covers their costs. The user specifies a desired input and output, but does not need to sign an explicit fee.

Explicit Incentives

are used for our market swap and transfer primitives. For these, the user needs to sign an explicit fee that they are willing to pay to the solver. The fee can be a fixed amount, or can include some dynamic computation based on state, such as current gas cost plus a fee percentage. Using a recurring transfer as an example, a user who wants to transfer 7,000 USDC to another account once every 100,000 blocks might sign an explicit fee of gas cost + 2% paid to the solver in USDC at ETH/USDC market rate.

Protocol Architecture

Brink Intents Protocol's on-chain and off-chain components work across EVM networks to securely execute desired outcomes defined by users.

- Brink

Intent API:

A hosted service for building, signing, storing, and reading intents

- Brink Intent Pool:

An off-chain intents data store

- Brink Smart Contracts:

A smart contract layer that solvers must interact with. Enforces security of intent execution and verification

- Brink

Solvers:

A network of bots that resolve intent outcomes on EVM-compatible chains

- Brink UI:

An app built on the Brink Intents infrastructure for end-users to create intents

[

1643×1872 307 KB

](https://europe1.discourse-cdn.com/standard20/uploads/anoma1/original/1X/64475e4384a70f117df5120d49d0c09a4a6a84a0.png)

Intent API

The primary interaction layer for front-end DApp developers and solvers to use the Brink Intents protocol. Front-ends use the API to build the intents data that their users need to sign, and to submit user signed intents to the Intent Pool. Solvers use the API to query the Intent Pool in order to discover intents to solve. The API also provides endpoints for interacting with Brink smart contracts and building transactions that solve signed intents.

Intent Pool

A data store for signed intents. This is a permissioned intent pool specifically designed for the Brink Intents protocol. All signed intents are public by default. Since intents are public, our intent pool treats them as immutable and does not allow them to be modified or removed by users. While the intent pool is permissioned, signing and solving in Brink Intents protocol is permissionless. Users and DApps using Brink Intents can choose to broadcast intents to other intentpools. The protocol is designed to accommodate decentralized intentpools and we hope these will be available in the near future.

Brink Smart Contracts

A smart contract layer that all solvers must call when executing intent solving transactions. The Brink smart contracts validate intent signatures, verify signed intent conditions, control execution of conditional and outcome primitives, and check that solver transaction outcomes meet the expected outcomes signed by users. These smart contracts ensure that solvers can only take actions that users authorize.

Solvers

A network of bots that solve user intents. They are incentivized by MEV. Solvers are not permissioned at the smart contract layer, but users can choose to limit solving of their intents to a specific solver or group of solvers.

Brink UI

A DApp to help users create and sign intents for common use-cases, such as stop-loss orders and recurring swaps (Dollar Cost Averaging), and to build custom intents without having to write code or interact with an API. The Brink UI is a web application that works with Metamask and other standard Web3 wallets.

Account Abstraction

Brink Intents supports both EOA's and smart contracts as signers, allowing intents to be signed with either EIP-712 or EIP-

1271. Brink Intents also supports creation of intents as an EIP-4337 UserOperation. Part of the Brink smart contract layer is an extremely light abstracted proxy account that EOA's and smart contract wallets can either deposit to or approve to.

Multi-Chain Support

Brink Intents works on any EVM-compatible chain. The Intents API and Intent Pool allow intents to be signed for any chainId. Users can also create intents that involve multiple chains.

Next Steps

Brink Intents is currently only available to [early access users](#). Please reach out on Discord or Twitter [@brinktrade](#) if you're interested in building.