

Introduction to Brink

[Suggest Edits](#)

Welcome to Brink Intents

Brink Intents lets you build [powerful intent-based features](#) into your Web3 application, without having to build smart contracts or run solvers. You can integrate intents with just a few lines of code.

We recommend reading our introduction post [Introducing Brink Intents](#), for an overview of Brink Intents.

Disclaimer

The Brink engineering team is still working hard on the products explained in these docs. Therefore, you may come across some inconsistencies or stale/missing information. We are working hard to make the Brink Intents Protocol fully feature-complete, and refining the documentation is part of that process. If you have any questions, please reach out to us on [Discord](#).

What can I build with Brink Intents?

You can build any intent-based workflow, from basic swap functionality to complex workflows that execute continuously based on state conditions. Here are just a few examples:

- Recurring swaps (dollar cost averaging)
- Subscriptions with recurring transfers
- [Stop-loss](#)
- and [bracket orders](#)
- Collateral liquidation protection
- Yield/rewards auto-compounding
- Portfolio rebalancing

Your users can permit workflows like these with only a single signature. Brink solvers will run them continuously.

Intent-based workflows can be designed to run immediately after the user signs, or conditionally based on market price fluctuations, governance proposal results, wallet balance deltas, or any on-chain state change that you define.

How do I integrate intents?

You can define intents with a few lines of JSON using the [Brink DSL](#):

```
JSON // swap 5,000 USDC to ETH every 10,000 blocks up to 5 times {replay : { // signer can cancel the intent at any time
nonce :123 ,runs : 'UNTIL_CANCELLED' },conditions : [{type : 'blockInterval' , // signer allows the intent to run every 10,000
blocks up to 5 times interval :10_000 ,maxIntervals :5 }],actions : [{type : 'marketSwap' , // signer allows swap of 5,000 USDC
for ETH at market price tokenInAmount :5000.00 .tokenIn : 'USDC' ,tokenOut : 'ETH' ,fee :2.5 // incentivize solver with 2.5% of
the swap }] } Then use the Brink API to get intent data for the user to sign and submit their intent. Once a valid signed intent
is submitted, solvers will monitor and execute the intent automatically.
```

Read about [how to create your first intent](#).

Why should I use Brink Intents in my DApp?

Intents simplify interactions for your users. Here's a short list of benefits of using signed intents over standard transactions.

- Users don't need ETH or other native tokens in their account to pay gas fees
- Intents don't have any risk of revert, which can be costly and annoying
- Protection against common MEV "attacks" such as sandwiching
- Users can sign one intent to accomplish multiple future actions (example: swap once a day for 30 days)
- One intent can operate across multiple networks without the user having to "switch" networks manually

How do users interact with Brink Intents?

All intent-based workflows can be permitted by users with a single signature. Signatures can be issued from either an [EOA](#) or an [Account Abstraction](#) wallet such as [Safe](#) .

If you're building a standard DApp, you can prompt users to sign intents using their EOA with [EIP-712](#) , a common standard supported by [Metamask](#) and many other wallets. Intents can also be signed with [EIP-1271](#) , a signature validation method for smart contracts.

If you're integrating intents with an [ERC-4337](#) compliant wallet, either signature scheme can be used to create a UserOperation for an intent.

Where does liquidity for swaps come from?

When your users sign an intent, they are tapping into all liquidity across the entire Web3 ecosystem.

Your user does not need to specify a liquidity source. Intent-based swaps only require users to specify their desired outcome. For example, if a user wants to swap WETH for DAI at a better price than current market conditions will allow, this can be defined as a limitSwap :

JSON {type : 'limitSwap' , tokenIn : 'DAI' , tokenInAmount : 1300.00 tokenOut : 'WETH' , tokenOutAmount : 1.0 } Solvers will monitor this intent and compare pricing across all liquidity sources. They compete to find the most efficient path to source this swap and cover their costs.

How can I start building?

Read the [Quick Start](#)

How can I contribute to Brink?

We welcome open source contributions. Browse our code on github.com/brinktrade and reach out to the team on the [Brink Discord](#) Updated 5 months ago

[Quick Start](#) Did this page help you? Yes No * [Table of Contents](#) * * [Welcome to Brink Intents](#) * * [What can I build with Brink Intents?](#) * * [How do I integrate intents?](#) * * [Why should I use Brink Intents in my DApp?](#) * * [How do users interact with Brink Intents?](#) * * [Where does liquidity for swaps come from?](#) * * [How can I start building?](#) * * [How can I contribute to Brink?](#)