Today Flashbots and our community are excited to launch a*public beta* of **Flashbots Protect**, a new product line focused on allowing everyday users and developers to easily submit transactions to Flashbots.

Since launch anyone has been able to submit transactions to Flashbots, but Flashbots Protect expands access by providing simple ways for users and developers to integrate. In turn the benefits of the Flashbots Auction, like frontrunning protection, no cost for failed transactions, and priority in blocks, will be accessible by a much wider set of stakeholders.

Moreover, our release today is a foundation for future releases: with over 90% of the hashrate of Ethereum accepting bundles and the introduction of [megabundles](#) we have plans for greatly increasing the number of transactions we can support per block and for powerful new features for end users.

Flashbots Protect is composed of an API and an RPC:

- **Flashbots Protect API** is a simple API that developers can integrate into their applications to enable users to have their transactions sent to the Flashbots Network.

- **Flashbots Protect RPC** is an RPC endpoint that users can enter into their wallet and have their transactions sent to the Flashbots Network.

In this blogpost we briefly introduce each product.

## Flashbots Protect API for developers

The Flashbots Protect API is a simple tool that developers can use to give their users the ability to submit transactions to Flashbots. Importantly, users do not need to change anything to be able to use the Flashbots Protect API on your applications! Here's how it works:

- A user executes a transaction

- Instead of being sent to the public mempool — where hungry sandwich bots are watching — it is sent to Flashbots Protect

- We smartly manage the submission of that transaction to the Flashbots network, resubmitting the transaction and monitoring for its inclusion in a block

- Updates on the transaction are streamed to your application, where they can be displayed to users

- Your user can submit a cancel request at any time and we'll pause submission of that transaction

There are two ways of using the Flashbots Protect API. First, you can emit transactions over a websocket and listen for updates. We envision this first option being used by most applications. Second, transactions can also be posted to an endpoint with the `eth_sendRawTransaction` json rpc method.

**How the Flashbots Protect API makes integrating with Flashbots easier**

Flashbots allows users to specify granular preferences about their transactions, such as the exact block number that they want their transaction included in. While this granularity is great for searchers, it can be cumbersome for the average developer or user because it requires sending transactions in new ways.

The Flashbots Protect API abstracts away all of that complexity. Simply send us transactions and we'll manage the submission to Flashbots. Specifically, we format transactions as bundles, simulate them to make sure that they won't fail, and keep submitting them to Flashbots for every block. We'll also check for gas prices and stop submitting your transaction in the case that your transaction is underpriced relative to where base fees are.

We have some other product features on our roadmap that will make sending us transactions even better still yet. If you have any feature requests please let us know!

**Flashbots & mistX collaboration**

The mistX team, leveraging their experience building a DEX that lets users trade with Flashbots, built the technology behind the Flashbots Protect API in collaboration with the Flashbots team. **The Flashbots Protect API is hosted and operated by Flashbots**, and the codebase is maintained jointly with mistX.

**How can I use the Flashbots Protect API?**

Documentation for the Flashbots Protect API can be found here: docs.flashbots.net

**Where can I ask questions?**

The #flashbots-protect channel in the Flashbots Discord

## Flashbots Protect RPC for users

The Flashbots RPC endpoint is a simple tool that anyone can add to their wallet and have their transactions sent to Flashbots. Everything about your normal web3 experience will stay the same, except your transactions will be sent directly to miners via Flashbots instead of going to the public mempool. Moreover, you benefit from two other key features of Flashbots: no cost for failed transactions and priority in blocks.

Please be aware that this is a public beta and your transaction may occasionally time out and never be mined. To add the Flashbots RPC please see our onboarding guide on the Flashbots docs here.

The Flashbots Protect RPC is a collaboration from the following communities: MiningDAO for significant code contributions, mistX for their work on the Flashbots Protect API and help testing, Nethermind for help with devops and testing. Thank you also to the early adopters of this product for help testing.

**How does the Flashbots RPC work?**

Whenever you make a transaction in your preferred wallet that transaction will be sent to the Flashbots RPC. Upon receipt we validate that this is a valid transaction and evaluate whether that transaction needs frontrunning protection. Then we send the transaction to the Flashbots Protect API, which manages the submission of that transaction.

**How do we evaluate what needs frontrunning protection?** We maintain a list of functions that don't need frontrunning protection, such as simple ETH transfers or ERC20 approvals, and if and only if your transaction is directly calling one of these functions then we'll send it to the mempool instead. That is because these pre-approved functions are "safe" from frontrunning bots, and it is cheaper and likely more quick to get a transaction approved via the mempool.

## Stewarded by Flashbots, built with community, for our ecosystem

Enabling a permissionless, transparent, and fair ecosystem for MEV is a shared objective across Flashbots' active community of researchers and builders.

We further our mission to democratize MEV in part by working closely with an ecosystem of value aligned contributors from various backgrounds and projects. We would like to give a special shoutout to *Don & Dire from mistX, Swaraj from Nethermind, and Ivan from MiningDAO, in addition to several Flashbots mates, for contributing to the Flashbots Protect API and Flashbots RPC.*