

We developed the MEV Blocker RPC endpoint in collaboration with Agnostic Relay and Beaverbuild in order to provide a very simple Pareto improvement over transaction submission via the public mempool. MEV Blocker offers frontrunning and revert protection as well as MEV kickbacks for backruns on all kinds of raw Ethereum transactions.

While MEV Blocker achieves a very good level of protection from MEV, we mainly built it as a complement to CoW Swap for transaction intents that are not yet supported by the protocol (e.g. providing liquidity, NFT trades, etc...) and as a stepping stone for integrations (giving users protection is as simple as changing your RPC endpoint).

Of course, we believe that CoW Protocol itself is still a much better approach for the DEX trading use case in particular and as a solution for MEV as a whole (wasting less block space, as well as giving more value back to users, better prices from batching and a better user experience via signed intents). In a sense, CoW Protocol is the dApp layer solution to MEV, whereas MEV Blocker is an infrastructure addition to the PBS "solution" to MEV.

Moreover, CoW Protocol itself can use MEV Blocker's infrastructure to benefit CoW Protocol solvers and therefore also CoW Swap users. In the last week alone, solvers paid more than \$20k for failed transactions ([dune](#)). Solvers are also still occasionally getting sandwiched. Note that these attacks don't come at a direct cost to the user — who is guaranteed the price that was promised in the off-chain competition. Instead, it is more of a calculated cost solvers carry when deciding what slippage tolerance to set for AMM interactions. Depending on the current gas price & volatility setting, a slightly higher than necessary slippage tolerance can reduce the cost from failed transactions.

Therefore, we have changed the submission logic to default to using the MEV Blocker endpoint until solvers are enabled to implement their own solutions submission strategy.

Another aspect of MEV Blocker is its kickback mechanism for backruns. Searchers can bid to be placed directly behind a transaction to potentially equalise AMMs that have been touched disproportionately. The originator of the transactions receives 90% of the searcher's bid as a refund ETH transfer within the same block.

Unfortunately, even CoW Protocol settlements are sometimes prone to backruns. The most common case is large trades which don't attract enough private market maker liquidity and thus move on-chain liquidity sources to a point where external venues (such as Binance) offer an arbitrage opportunity. The ENS trade was such an example:
<https://twitter.com/Ox94305/status/1623664463109816322>

It is of course the goal of CoW Protocol to provide the best price across all different venues for users on Ethereum (a task solvers compete and are rewarded for). However it is the reality that those backrunning opportunities still exist today. We think it is clear that capturing 90% of such opportunities by using MEV Blocker is better than letting them go to validators, but this begs the question: how to utilize the refunds that arise? This is the goal of this post: to kickstart the discussion, which should likely result in a CIP vote. For now, the refunds are collected in [this Safe](#) which is controlled by CoW DAO. An overview of backrun transactions can be seen here: <https://dune.com/queries/2403907>

While we believe this value ultimately belongs to CoW Protocol users, it is hard to properly attribute and distribute the value in an automated way for multi-order batches. We also think that it would be wrong to send those refunds to the affected solvers, as they should have an incentive to improve their matching strategies in protocol and ensure users get the best price.

In the meantime, the funds could potentially be used as a revenue stream for CoW DAO to further foster the solver competition and incentivize more competitive solutions (e.g. by buying back CoW tokens to increase the economic incentive of the competition).

Please comment with suggestions and arguments on how you think these kickbacks should be used!