

Grant Title:

MEV Blocker bundle verification tool

Author:

[@bleu](#) [@yvesfracari](#) [@ribeirojose](#) [@mendesfabio](#)

About You:

[bleu

](<https://github.com/bleu-studio>) collaborates with companies and DAOs as a web3 technology and user experience partner. We're passionate about bridging the experience gap we see in blockchain and web3.

Additional Links:

Our work for CoW so far:

- [CoW] [AMM Deployer](#):

a Safe app to deploy new CoW AMM pools from a Safe Wallet.

- [CoW] [MilkmanApp](#):

a Safe App designed for creating and managing Milkman orders within the CoW ecosystem. It empowers DAOs to sell tokens while deferring the price determination to execution time, leveraging price checkers for optimal pricing strategies.

- [CoW] [Python SDK](#) (ongoing):

we're helping CoW put together a Python SDK to provide for developers for query on-chain data, managing orders, and integrating with the CoW Protocol's smart contracts.

- [CoW] [Stop-Loss](#) (closing):

a Safe app UI to allow for multisig wallets to fully manage stop-loss orders.

- [CoW] [Have I been MEV'd](#):

a suite of bots and a web application integrated with the ZeroMEV API to inform the Web3 community about potential MEV losses.

Grant Category:

Developer tools (SDK)

Grant Description:

This open-source project aims to develop a verification tool for MEV Blocker transactions. The tool will analyze blockchain data, simulate transaction bundles, and compare simulated results with actual outcomes to ensure users receive maximum refunds and to identify any rule violations.

The system will automatically collect data for each new block, perform simulations, and send alerts for suboptimal payments or missing transactions to a Discord/Telegram channel. It will also feature an API for historical block verification. By enhancing transparency and optimizing user returns, this tool will contribute to the integrity of MEV Blocker operations in the broader blockchain ecosystem.

Grant Goals and Impact:

The goal is to create a publicly available software tool that verifies MEV Blocker transactions, ensuring users receive the highest possible refunds while identifying any instances of rule breaches.

Milestones:

Milestone

Due Date

Payment

Data fetching

3 days

1k xDAI + 2.5k COW

Bundle simulation

1 week

2k xDAI + 5k COW

Deployment

1 week

1k xDAI + 2.5k COW

Data Collection

1. Query Dune for all bundles in a specified block.
2. Identify the first transaction position (p
) using the MEV Blocker in this block.
1. Develop a method to retrieve backrun transactions related to a specified transaction.

Bundle Simulation

1. Simulate all bundles at position p
where the transaction occurred using RPC calls.
1. Order bundles by payment value and save the state (s
) after each: * Discard failed bundles.
1. Discard failed bundles.
2. For each bundle, simulate the backrun transaction at p+1
using state s
: * If the backrun fails, try including it at the end.
1. If the backrun fails, try including it at the end.
2. Verify all logged transactions against the actual block.
3. Compare simulated vs. actual payment values for each transaction and the entire block.

System Deployment

1. Create a container for data collection and simulation.
2. Set up automatic runs for each new block (after a set delay) to log alerts on Discord/Telegram for:
3. Suboptimal payments
4. Missing transactions
5. Suboptimal payments
6. Missing transactions
7. Develop an API to verify payments and missing transactions for any historical block.

Funding Request:

We propose that milestone payments be released upon each milestone's approval.

Budget Breakdown:

The budget includes the hourly rates of a developer during the execution and a project manager on a need basis. The xDAI part of the budget shall be paid after each milestone's completion and the COW shall be vested over 1 year to cover diluted

maintenance and related costs for the same period.

While creating this proposal, we discovered that Dune queries will significantly exceed the free limit of 2500 credits/month, with our tests showing 5-10 credits spent per query per block. Dune's pricing structure is complex, depending on the number of columns, rows, and bytes returned. To create a sustainable long-term service, we need CoW's input on this matter. You can review the estimated datapoints for each block in the [provided Dune query](#) (note: 1000 datapoints = 1 credit on the free plan). Consequently, we'll require at least the Dune Plus plan (\$349/month) and potentially additional credits. A decision is needed on whether CoW will cover this cost or if we can leverage the existing CoW Dune team's resources.

Gnosis Chain Address (to receive the grant):

0x554866e3654E8485928334e7F91B5AfC37D18e04

Other Information:

- All the code will be open-source from day 0. We're open to feedback during PRs as well;
- We're happy to answer any questions and are open to feedback about this proposal;

Terms and Conditions:

By submitting this grant application, I acknowledge and agree to be bound by the [CoW DAO Participation Agreement](#) and the [CoW Grant Terms and Conditions](#).