Simple Summary

A very important component of CoW protocol are the so-called solvers. CoW protocol is truly unique in the sense that it allows independent off-chain solvers to compete for the right to execute a batch of trades.

Solvers are rewarded by CoW protocol with CoW tokens every time they executed a batch of trades:

CIP-2: Solver Rewards [Closed Proposals

](/c/cow-improvement-proposals-cip/closed-proposals/13)

CIP: 2 title: Solver Rewards author: Felix Leupold status: Active created: 2022-02-21

Update 11/03/2022 Vote passed with 65M votes in favour (134k against) and has been executed at:Ethereum Transaction Hash (Txhash) Details | Etherscan Simple Summary The purpose of this proposal is to ensure continued solution submission by solvers powering CoW Protocol by refunding them the gas cost for solution submission plus a fixed 100 CoW reward for every winning settlement. This proposal is aiming a…

At the moment, the following questions are at the top of mind of any new solver developer:

- How do I get my solver deployed in production?
- How good is my solver?
- Say the CoW team is ready to review my solver, which rules will they enforce to approve it or reject it?
- Why are there already solvers in production that did not have to go through the approval process all other new solvers have to go through?

Idea

The idea is to build an indipendent testing and benchmarking playground for solver developers that will let them test their solutions on the fly and without having to overload the CoW team with requests to test and approve their solvers.

The playground can be built in two different ways; the later more complex than the other.

Approach 1

The first approach would consist on collecting sets of trade batch test instances in an AWS S3 bucket every end of month. Solver developers can tap into the bucket of the most recent month and attempt to solve the trade batch test instances. Once the solver is finished, the solution is submitted to a solution scoring service that will provide a score to the solver. Solvers are allowed to submit their solutions several times a month. This setup is similar to a Kaggle competition, in which solvers will need to achieve the highest score to climb up the ranking and be considered for deployment by the end of the month.

Approach 2

The second approach would consist on deploying a websocket that will emit trade batches to listening solvers. Solvers must be prepared to receive the trade batch and submit a solution as soon as possible to the scoring service. This setup will allow to benchmark solvers in their ability to achieve maximum utility and the speed at which they do so. Similar to approach 1, the best solvers are shortlisted for deployment by the end of the month.

Either approach 1 or 2 are fairly simple to implement and would dramatically improve the solver development and approval process.

Rationale

1. Since all solvers utilize the same trade batch test instances, it can be ensured they are fairly benchmarked and, later on, screened for deployment.
2. Solver developers can benchmark their solvers without having to wait until they are whitelisted.
3. On the fly benchmarking and scoring can help solvers understand how they can improve their solutions.
4. Total transparency in the selection and deployment of solvers. No solver should be running in production “just because”. Solvers running in production should be selected based on a fair comparison of the actual utility they can achieve or the unique attributes they bring into the mathmaking process of CoW protocol.
5. If the performance of a solver that has been deployed decreases, then space should be made for a different solver that can achieve a higher performance score.
6. Simplify the whitelisting process for the DAO.

Discussion Points

The exact scoring methodology remains an open question:

- How should trade batch solutions be scored?
- Should different scoring functions be implemented? One for speed, one for utility, one for amount of matches? Should an average total score be calculted?

Implementation

For the implementation, it would be suggested to build a small task force of 5 community members that would build an MVP of the service and ranking dashboard.