The new solver rewards mechanism has been ongoing for a couple of months. It's only until recently that we got the time to dig into the way in which our solutions are being assessed and how our participation in the solver competition translates into P&L based on the new (more complex) rewards mechanism.

Part of the new rewards mechanism consists of penalizing solvers when their solutions fail at execution time. In this case, the solver is penalized based on the score of the third best solution (see [this post

](https://forum.cow.fi/t/cip-20-auction-model-for-solver-rewards/1405) for more detail) reference solution. While in theory this seems appropriate, the reward mechanism fails to consider cases in which the third best solution would also revert at execution time.

Thanks to the new auction result [notification service

](https://github.com/cowprotocol/services/pull/1689) deployed by CoW Protocol we can look into some examples of seemingly unfair penalties for our solver. The service notifies us if our settlement suceeded or failed at the execution block. We consider the following three auctions:

Auction 7000251

- Solver wins auction.
- Solution proposed simulates successfully at the pending block in which the solution was submitted to the Driver (17788186).
- Solution proposed simulates successfully one block after submission to the Driver (17788187).
- Driver attempts to settle solution at block 17788188 (two blocks after original submission), but slippage limits are triggered.
- Solutions of next two best solvers also fail at block 17788188. Their slippage tolerance limits are also triggered.

Auction 7000269

- Solver wins auction.
- Solution proposed simulates successfully at the pending block in which the solution was submitted (17788200).
- Solution proposed simulates successfully at blocks 17788201, 17788202, 17788203, 17788204.
- Solution is flagged as failed even though it could have been executed.

Auction 6987617

- Solver wins auction.
- Solution proposed simulates successfully at the pending block in which the solution was submitted to the Driver (17777715).
- Solution proposed simulates successfully one block after submission to the Driver (17777716).
- Driver attempts to settle solution at block 17777717 (two blocks after original submission), but slippage limits are triggered. In order for the settlement to be successful, the solver would need to accept a negative slippage larger than 0.01 ETH, which is the maximum penalty attributable to the solver. Hence, the resolver is acting reasonably by taking the rewards penalty as opposed to the negative slippage loss.
- The solutions from the 4 next best solvers (except for 1inch solver which allowed for negative slippage larger than 0.01ETH) also fail.

Conclusion

In summary, we identify the following practical imperfections from the current rewards mechanism:

1. It assumes the score used to compute the penalty in case of failure is feasible at the execution block. However, as it can be seen by the examples provided, this is not always the case.
2. It does not take into account the feasibility of a trade to determine if the solver should be attributed a penalty or not. For instance, unfeasible trades could be flagged based on the number of failing solutions across all solvers that participated in an auction.
3. It does not take into account protocol induced failures. For instance, the solution for auction 7000269 could have been executed throughout several blocks. The solutions for auctions 6987617 and 7000251 were feasible for two successive blocks, but the lag between submission and execution triggered a reasonably high the slippage tolerance.

This being said, we propose to shift from the theoretical definition of failure (which is oblivious to all these practical nuances) to a practical definition of failure in which the following questions are addressed:

- What is a solver induced failure? If solver can prove solution succeeds at the pending block she submitted to the Driver (and one block thereafter), then should it be considered a solver induced failure? If solution fails due to extreme slippage (i.e. slippage beyond max reward of solver or something like that), should it be considered a solver induced failure?
- What is a failure that should be attributed to the protocol? If solution simulates successfully x blocks after submission to the Driver, should the protocol pay the penalty instead of the solver?