CoW Grant - Automated fee withdrawal

Author

I am meetmangukiya. I am a software developer, have been in DeFi and web3 space for over 3 years now. I also have familiarity with CoW protocol technicals.

Grant Category

Other/Misc.

Grant Description

Automated fee withdrawal for solver fee distribution.

Grant Goals and Impact

Current process is automated but requires manual inspection of calldata everytime its ran by the core team members. It also requires the signers to sign the transactions everytime its ran. With the safe module contract code it won't require dilligence since the contract code will enforce it.

Scope

Create a Safe module that can be enabled on the Safe 0x423cEc87f19F0778f549846e0801ee267a917935

that accepts an array of tokens to be sold. The orders created must adhere to the aforementioned limitations (i.e. receiver

).

The grant would specifically cover the work for handling automated fee collection. Precisely:

1. A Safe module smart contract, which creates orders on behalf of the smart contract.
2. A script that triggers order creation onchain.

The Safe module is required to:

- Be deployed with the receiver

(address

), to-token

(address

), and keeper

(address

) set as immutable variables.

- Provide a permissioned drip

function (callable only by keeper

) for "dripping" protocol fees into the receiver

's account by means of CoW Protocol orders. * Accepts an array of structs, with the struct being defined as:

struct SwapToken { IERC20 sellToken; uint256 sellAmount; }

- Creates an order selling the tokens from the GPv2Settlement

contract to the receiver

that is partially fillable and has buy amount 1.

- Calls settle

and triggers the orders via setPreSignature

.

- Accepts an array of structs, with the struct being defined as:

struct SwapToken { IERC20 sellToken; uint256 sellAmount; }

- Creates an order selling the tokens from the GPv2Settlement

contract to the receiver

that is partially fillable and has buy amount 1.

- Calls settle

and triggers the orders via setPreSignature

.

- Provide a permissioned approve

function to approve the vault relayer from the settlement contract. * Accepts an array of IERC20 tokens.

- Approves the vault relayer on behalf of the settlement contract to spend type(uint256).max
- Accepts an array of IERC20 tokens.
- Approves the vault relayer on behalf of the settlement contract to spend type(uint256).max
- Provide a permissioned revoke

function to revoke approvals from the settlement contract to any

contract. * Accepts an array of structs, with the struct being defined as:

struct Revocation { IERC20 token; address spender; }

- Approves the vault relayer on behalf of the settlement contract to spend 0.
- Accepts an array of structs, with the struct being defined as:

struct Revocation { IERC20 token; address spender; }

- Approves the vault relayer on behalf of the settlement contract to spend 0.
- Gas consumption is kept to a minimum. A suggestion here is to not sell the entire balance and leave 1 wei so as to not reset the slot to zero, and waste a refund.
- Be written in Solidity v0.8.25 using the foundry smart contract development framework.
- Have comprehensive test coverage.
- A script, readily packaged in a docker container, writting in typescript and ethers v6 that:
- Handles similar parameters as the self-sell

script above.

- Creates the transaction for calling drip

by the keeper

EOA.

- Calls the API to post the orders.
- Handles similar parameters as the self-sell

script above.

- Creates the transaction for calling drip

by the keeper

EOA.

- Calls the API to post the orders.

The script is required to:

- Determine which tokens should be withdrawn from the settlement contract. This is all tokens whose balance on the settlement contract is worth more than a fixed, configurable amount in USD.

- Determine which tokens need to be approved for trading.

- Post orders to the API in the format that is accepted by the Safe module for each token to be withdrawn.

- Send a transaction from an EOA to the module that sets any needed approval and calls drip

for all orders created in the previous step.

Milestones

- The safe module

- Tests

- Script

. Will probably deliver this under 2 weeks.

Funding Request

5000 xDAI. Full payment can be made on delivery is fine.

Budget breakdown

- Development - 5000 xDAI

Gnosis Chain Address

0xd33d055ea52ec5a685995d40f19fa98d0744df0c

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.