Grant Title: AccuMoolator

„Automated Dollar-Cost Averaging (DCA) System“

Author:

Maximilian Stahl @GiraeffleAeffle / GitHub: GiraeffleAeffle / maximilian.stahl@blockscape.network / X Giraeffleaeffle / Farcaster giraeffleaeffle

Florian Lüffe @flobrown.eth / GitHub: floberlin / florian.lueffe@blockscape.network / X flobrown_eth / Farcaster: flobrown.eth

About You:

I am a DevOps Engineer specialized in the blockchain area currently working at Blockscape Finance AG. I have experience in running infrastructure for several blockchain networks, as well as oracles (Chainlink, Celo, StakeWise) and indexer (TheGraph) and also contributed to open-source code for Chainlink and StakeWise. I have also worked in a Wallet as a service product as Technical Consultant & Fullstack DevOps Engineer and as a Backend Engineer for a NFT Marketplace project. I studied electrical engineering in my bachelor’s degree and then specialized in information and communication technology.

I am backed and supported by Blockscape in this proposal. They are a validator for several Cosmos networks as well as a Lido operator and run DVT in production. They have experience with proposals due to proposing and building the Gravity Bridge which connects the Ethereum to the Cosmos Ecosystem. In the scope of this proposal I will collaborate with other Blockscape engineers to build this DCA system.

I’ve been using CoW Swap already since it was released and I’m very happy with it. Since about half a year I am also using Monerium with a weekly standing order that mints EURe into my Safe on gnosis chain. Because I always have to set up a TWAP or buy my Tokens every week manually I would like to automate this process with conditions.

For example in Germany you have 1 year of speculation period, which means when you sell your token before you held it for one year you have to pay taxes. To avoid this there could be a condition to hold the tokens for a specified time. Also you want to sell the token with profit, so the price requirement would be to sell the token when the price is higher than for when you bought it.

I’m convinced that a lot of people would be interested in such a feature and that people would be able to save money with a standing order in their favorite tokens and sell them with their set conditions.

Additional Links:

X BlockscapeLab, https://blockscape.network/

Grant Category:

Integrations and protocol order flow

Grant Description:

This project aims to develop an Automated Dollar-Cost Averaging (DCA) System, triggered by repeating EURe bank transfers, to purchase specific tokens. The mechanism ought to be designed to facilitate the establishment of DCA-plans, regardless of whether the user possesses the requisite EURe amount at the time of creation. It is essential that the necessary EURe amount is available at the moment the DCA transaction is triggered. The system would only execute the order when specific conditions are met like holding tokens for a minimum time for tax exemption, sell them at a profit, or (handle tax loss harvesting with customizable conditions).

Use the Programmatic Order Framework from CoW Swap to implement the following Process Flow:

1. Conditions for Token Purchases:

a. Token Balance: Sufficient balance of the selling Token to be able to buy the Token you want.

b. (Token Minting Event): Data from Monerium’s API.

1. Conditions for Token Sales:

a. Hold Period: Maintain ownership of the token for a minimum time to qualify for tax exemption.

b. Price Requirement: Sell the token only if its selling price is higher than the purchase price.

1. Optionally Profit Handling: Automatically transfer any gains back to your bank account in EURe.

(Additional Feature: Tax Loss Harvesting

- Customization: Ability to set specific conditions for this function as per your requirements.)

High Level Workflow Architecture:

[

1600×665 135 KB

](https://europe1.discourse-cdn.com/business20/uploads/cow/original/2X/1/120fdc50396e736b90c9f1903bb32a72fe25b791.png)

Grant Goals and Impact:

The primary goal is to provide an easy way to DCA in and out of a Token without the need to aim for the lows and highs, that maximizes returns and optimizes tax obligations for users. This project will benefit the CoW Protocol ecosystem by enhancing user engagement and offering efficient investment strategies.

Milestones

Milestones

Due Date

Payment

Research, Architecture, and Core Feature Development

~ 01.04.2024

4k xDAI

MVP Front-End & Contract Development and Integration

~ 01.07.2024

15k xDAI

User Feedback & Refinement, Advanced Features and Profit Handling

~ 01.09.2024

6k xDAI

Milestone 1: Research, Architecture, and Core Feature Development

Objectives:

- Research the system architecture, including the necessity of an off-chain service or oracle, and the potential integration with the Monerium API.
- Familiarize with the Programmatic Order Framework.
- Develop and experiment with a Time-Weighted Average Price (TWAP) strategy using the forked CoW Protocol repository.

Deliverables:

A core automated DCA system design capable of:

- Implementing token purchases with EURe in defined time epochs and for different periods based on the TWAP design
- Acquiring API credentials for Monerium and minting testnet EURe tokens
- Propose design for a DCAOrder smart contract based on the composable CoW architecture

A detailed research report outlining the chosen architecture and the decision-making process regarding system components like off-chain services, oracles, and API integrations.

An architecture diagram illustrating the planned system, including all components and their interactions.

Based on the findings:

1. A forked/modified design proposal of the CoW Protocol composable-cow repository with a TWAP strategy adapted for DCA.

or

1. An off-chain/oracle service design proposal

Time Estimate:

- 1-2 Months

Budget on completion:

- 4K xDAI (for research & design)

Milestone 2: MVP Front-End & Contract Development and Integration

Objectives:

- Develop an MVP front-end interface for the Automated DCA System.
- Ensure seamless integration of the front-end with the backend service developed in Milestone 1.
- Enable users to easily create new DCAs and manage their existing ones.

Deliverables:

A user-friendly MVP application that allows users to:

- Create new DCA orders with their EURe token
- View and manage existing DCA orders

Integration of the front-end application with the backend service, ensuring smooth functionality and data flow.

Comprehensive testing documentation demonstrating the functionality and integration of the front-end with the backend.

A user guide or tutorial for the MVP, explaining how to use the interface to create and manage DCA orders.

Time Estimate:

• 2-3 Months (+)

Budget on completion:

- 15K xDAI (for development)

Milestone 3: User Feedback & Refinement, Advanced Features and Profit Handling

Objectives:

Research and implement advanced features for the Automated DCA System, enhancing its functionality and user experience based on community feedback.

Develop a comprehensive profit handling mechanism.

Evaluate and potentially integrate additional tools or services to improve the system’s efficiency and capabilities.

Deliverables:

- Updated/Refined UI, DCA features
- Profit handling mechanism

Time Estimate:

• 1-2 Months (+)

Budget on completion:

- 6K xDAI (for development & marketing)

Funding Request:

The total funding required in xDAI and COW tokens, with justification based on the project's scope and impact.

We request 25000 xDAI for this project. It can have considerable impact on CoW swaps and Moneriums adoption because of the easier way to onboard and create a savings plan.

Budget Breakdown:

Total Budget: 25,000 xDAI

Gnosis Chain Address:

gno:0x10535e219165185FFC3c1354AC39f672100c9951

Other Information:

Opensource contributions:

- stakewise/helm-charts
- telekom-mms/chainlink-helm-charts

Proposal:

- keplr: wallet.keplr.app/chains/gravity-bridge/proposals/170
- ssv: Blockscape - Decentralized Institutional ETH Staking | ssv.network

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.