

Problem statement

The [Treasury Management Committee \(TMC\)](#) was introduced last April as a result of [Snapshot vote](#). The committee's purpose is to develop constrained strategies, execute actions within [Treasury Management Principles](#), and eventually automate itself away.

Key aspects include keeping ETH as the primary unit of account, supporting the Lido protocols' integrity, growth, and robustness, minimizing risk of loss, and ensuring all treasury ETH is staked using Lido.

Its capabilities are presently confined to proposing votes via [Aragon](#), such as [TMC-0 \(vote #161\)](#), highlighting the need for more flexible tools to enhance treasury agility and rebalancing. [TMC-1](#) introduces the notion of an automated swapper triggered through an optimistic governance (like [Easy Track](#)).

A treasury swapper is an example of on-chain tooling that can implement TMC-1 and execute constrained actions, with optimistic purview by Lido DAO token holders. The swapper should adhere to the following requirements:

General requirements

- proposed solution should be able to be used by the TMC multisig
- Lido DAO token holders should always be able to veto any actions
- funds should only be available for a limited set of transactions, pre-approved by Lido DAO token holders;
- proposed solution should have restricted access to the Lido DAO treasury.

Technical requirements

- every trade should be MEV-protected;
- every trade should be "price-guaranteed" (the trade is expected to be cancelled if the minimum exchange amount is not received);
- beneficiary of every trade should be Lido DAO [treasury](#);
- variety of tokens for trade (both input and output) should be strictly limited by DAO;
- TMC should have the ability to operate trades (move funds from treasury, place orders, move funds back to treasury in case of unsuccessful trade);
- TMC should never take custody of Aragon funds;

Primary use-cases

- selling stETHs to stables (DAI/USDC/USDT)
- rebalancing the amounts of different stables (DAI/USDC/USDT)

Considering mentioned requirements it is proposed to develop a new technical solution *Stonks*, which will contain a fixed set of necessary operations for exchanging funds and managing the treasury.

Solution components

1. [Easy Track](#) motion or Aragon vote (in case of emergency) for token transferring. These contracts will be used as is.
2. [Stonks](#) - a set of contracts which act as a receiver of tokens from the first step and a container of swap operations set by Lido DAO;
3. [Cow Protocol](#) - a fully permissionless trading protocol that enables batch auctions to maximize liquidity via Coincidence of Wants (CoWs) in addition to tapping all available on-chain liquidity;
4. [ChainLink](#) - a decentralized oracle network that provides the USD price of Ethereum's native cryptocurrency and tokens.

There are two directions of tokens movement between components:

1. Swap (happy path) - when tokens are transferred from the DAO Treasury to Stonks for swapping using CoW Protocol. Swap proceeds are sent directly to the DAO Treasury (CoW Protocol API allows it).
2. Recovery (contingency) - if the swap doesn't execute for any reason, tokens must be returned to the DAO Treasury.

Happy path illustration

[

890×895 26.9 KB

](https://europe1.discourse-cdn.com/business20/uploads/lido/original/2X/3/339c7268c69ff0dca76180a946794e5dfb8bd23a.png)

Step 0

. A new Stonks instance is created with specific params and the DAO is proposed to add that contract to the Easy Track allowed recipients.

Step 1

. Tokens are transferred from the DAO Treasury to the Stonks instance via Easy Track top up factories or Aragon vote.

Step 2

. Order placement. TMC requests Stonks to deploy a new Order contract via placeOrder function and it automatically sends all available assets there. After deployment the Order emits an event about it's creation, sets an allowance to the CoW vault relayer contract and waits until this order is completed.

Step 3

. Order creation. Easy Track UI checks for Order Created

events and if it detects one, allows TMC to create an offchain order on Cow Protocol. This UI is trustless and can be used by anyone. The order price is checked against price tolerance parameters to simultaneously provide some margin to cover CoW Protocol fees + minor price fluctuations, but restrict from executing significantly unfavourable swaps (more details are in the [specification](#)).

Step 4

. The Swap. At the moment of order fulfilment CoW Protocol debits funds from Order contract & sends all exchanged assets to the DAO Treasury.

Recovery (contingency) flow illustration

[

834×860 24.8 KB

](https://europe1.discourse-cdn.com/business20/uploads/lido/original/2X/e/ea4390d859fc8b2180306afa645ae7027cae8637.png)

If for any reason swap isn't going through, there must be a method to recover the tokens. Tokens can be at Stonks (before the swap is requested) or at Order (once the swap is requested on Stonks).

Step 4

. The swap hasn't happened and the order has expired. Anyone can return all funds to the Stonks contract for further actions in a permissionless manner. From this moment Order contract becomes inactive.

Step 5

. At the moment, in case of market turbulence the DAO can decide what to do next:

- send all funds to the DAO Treasury back;
- create a new order again.

For more details about each step please read the [specification](#).

Voting actions

Lido DAO is proposed to add following contracts to the set of Lido on Ethereum protocol in case of successful Aragon vote:

new TopUpAllowedRecipient ET factory addresses

These factories are used to request tokens from the Treasury, to be swapped in Stonks orders (TMC multisig selects a

factory according to the swap direction, for example stETH → DAI, and enters the amount to be swapped).

TopUpAllowedRecipient for STETH → DAI / USDC / USDT (ADDRESS_TBA)

parameters:

- allowedRecipientsRegistry (ADDRESS_TBA
-)
- Token: stETH ([0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84](#));
 - Top up limit: $9_000 * 10^{18}$ (\$30M equivalent using [30d TWAP](#) rounded to the nearest thousand);
 - Limit refresh frequency: every six months

TopUpAllowedRecipient for DAI /USDC / USDT → USDC / USDT / DAI (ADDRESS_TBA)

parameters:

- allowedRecipientsRegistry (ADDRESS_TBA
-)
- allowedTokensRegistry (ADDRESS_TBA
-)
- Top up limit: $10_000 * 10^{18}$;
 - Limit refresh frequency: every three months

Stonks contracts

These contracts create order instances initiating CoW swap orders using tokens and parameters received from TopUpAllowedRecipient and AmountConverter. CoW orders are created via ET UI, following a preliminary double check of the minimal amount that the treasury should receive in case of success.

- STETH→DAI (STONKS_STETH_DAI_ADDRESS_TBA
-)
- STETH→USDC (STONKS_STETH_USDC_ADDRESS_TBA
-)
- STETH→USDT (STONKS_STETH_USDT_ADDRESS_TBA
-)
- DAI→USDC (STONKS_DAI_USDC_ADDRESS_TBA
-)
- DAI→USDT (STONKS_DAI_USDT_ADDRESS_TBA
-)
- USDT→USDC (STONKS_USDT_USDC_ADDRESS_TBA
-)
- USDT→DAI (STONKS_USDT_DAI_ADDRESS_TBA
-)
- USDC→USDT (STONKS_USDC_USDT_ADDRESS_TBA
-)
- USDC→DAI (STONKS_USDC_DAI_ADDRESS_TBA
-)

)

This contract provides functionality to retrieve expected token conversion rates based on the Chainlink Price Feed. It also sets immutable variables:

- [illegible]

Audit of Stonks contracts is being finalized and the report to be provided in the comments to this post before the Aragon vote starts.