

Simple Summary

This document proposes the implementation of a set of factories for Easy Track. The set is designed to transfer Treasury funds within periodic on-chain limits.

Motivation

To streamline routine governance operations, some Lido DAO activities are governed by [committees](#). For ease of operations, transactions from committees go through Easy Track. Voting on Easy Track is based on the principle of vetoing. Among other operations, committees can transfer funds from the Treasury. Since the number of committees and the amount of spending is growing significantly, it's highly desirable to have on-chain limits.

Easy Track motions already have the limit on single token transfer size (see [LIP-13](#)). In addition, there is a need to have separate limits for every committee that are set per period and correlate with the real budgets of the committee.

The proposal describes the design for periodic on-chain limits for committees' payments.

Specification

Overview

The idea of using limits is relatively straightforward:

- The DAO sets an on-chain limit per period and period duration for each committee via Aragon voting.
- Every time the committee wants to create a motion, it should be checked whether motion amount is within the committee budget.
- Every time the motion is enacted, the check is performed again. An attempt to overspend must lead to a revert.
- The amount of the enacted motion is automatically added to the expenses of the current period.
- The expenses of the current period are reset to zero at the beginning of a new period.

Technical specification

It is recommended to read [Easy Track Specification](#) proceeding with this section.

It is proposed to create a new special set of [EVMScriptFactories

](<https://github.com/lidofinance/easy-track/blob/master/specification.md#evmscript-factories>) and a new LimitsChecker contract which is inherited by AllowedRecipientsRegistry

Contracts based on the existing RewardPrograms set (RewardProgramsRegistry

, AddRewardProgram

, RemoveRewardProgram

, TopUpRewardProgram

) have the following differences from the existing ones:

- The term "RewardProgram" was substituted for a more general "AllowedRecipient",
- AllowedRecipientsRegistry

(ex-RewardProgramsRegistry

) stores and manages limits,

- TopUpAllowedRecipients

(ex-TopUpRewardProgram

) can check if the payment satisfies the balance for the current period on motion creating and enactment.

The following flow is proposed for storing and interacting with limits:

- As the TopUpRewardProgram

contract, the new TopUpAllowedRecipients

contract stores a token

for payments. It's set on deployment. If it is necessary to make payments in different tokens, then a separate set of factories for each token should be deployed and a limit must be set for each token.

- The TopUpAllowedRecipients

contract stores limit parameters: the period duration in months (periodDurationMonths

) and the total limit (limit

) - the maximum amount of funds that can be spent within the period.

- The expenses of the current period (spentAmount

) are also stored in the contract and is updated every time a new motion is enacted.

- During checks, the current payment amount is compared with the "spendable balance" (limit - spent amount).

- When a new period starts, the spentAmount

is reset to zero with the first motion's enactment in the period.

- To track when the new period has begun, the contract stores the time stamp of the end of the current period (currentPeriodEndTimestamp

).

The following figure provides an overview of the limits flow:

[

3420×1561 168 KB

](https://europe1.discourse-cdn.com/business20/uploads/lido/original/2X/f/f8e64827c09802c6a46e4468308ede656c2ccb02.png)

Design tradeoffs

Please proceed to the full LIP-19 text published on GitHub [for further details](#).

Links

- [Easy Track limits and budgets: ADR](#)
- [LIP-13: Easy Track payments limits](#)
- [Easy Track Guide](#)
- [Easy Track Specification](#)