

The original [Easy Track Motions proposal](#) received generally positive feedback from the community, so we've spent some time re-shaping the implementation to make it even more scalable and secure.

First, let us remind you what Easy Track motions are all about. If you are aware of the general concept, you can skip to [the new implementation breakdown](#).

## Problem and proposed solution

Lido DAO governance currently relies on Aragon Voting App model. This means DAO approves or rejects proposals via direct governance token voting. Though transparent and reliable, it is not a convenient way to take actions only affecting small groups of Lido DAO members. These actions hardly ever spark any debate in the community, and votings on such decisions often struggle to attract wider DAO attention and thus, to pass.

We propose Easy Track motions as a solution to this problem.

Easy Track motion is a lightweight voting considered to have passed if the minimum objections threshold hasn't been exceeded. As opposed to traditional Aragon votings, Easy Track motions are cheaper (no need to vote 'pro', token holders only vote 'contra' if they have objections) and easier to manage (no need to ask broad DAO community vote on proposals sparking no debate).

There are three types of votings run periodically by the Lido DAO that are proposed to be wrapped into Easy Track motions:

- Node Operators increasing staking limits
- Funds being allocated into reward programs
- Funds being allocated to LEGO program

More types of motions can be added later, e.g.:

- Topping up Lido staking referral program
- Purchasing staking cover

Creating and maintaining the listed Easy Track motions falls within the remits of two Lido sub-governance groups:

1. Node Operators Committee

to start staking limit increase motions.

1. Financial Operations Team

to start funds allocation motions (i.e. LEGO and reward programs).

Node Operators Committee should include addresses of active node operators partnering with Lido, the list is easy to look up in the [Node Operators Registry](#) Aragon app.

Financial Operations Team should consist of established DAO members' addresses and can be represented as a Gnosis Safe Multisig with access to corresponding Easy Track features.

## Proposed implementation

To create a feature as flexible and extendable as possible, we propose implementing Easy Track functionality within several contracts rather than building a single contract for it.

The statement below is abstract, please visit [specification.md](#) for full detailed specification.

## Easy Track workflow and motion life cycle

A group of core smart contracts will implement basic logic and store data related to Easy Track motions, and a series of separate EVM Script Factory smart contracts will represent all the supported types of Easy Track motions.

This architecture aims to support further extension by adding new types of Easy Track motions without replacing the core of the proposed system.

1. A motion can be started by calling createMotion()

function on EasyTrack

core contract.

1. Upon motion creation, EasyTrack

calls `createEvmScript()`

function on the corresponding `EVMScriptFactory`

contract passing all the data required to generate a motion enactment EVM script.

1. `EVMScriptFactory`

smart contract generates the script and returns it to the EasyTrack

core contract.

1. EasyTrack

contract conducts the motion according to the motion settings (i.e. motion duration, objections threshold and max active motions limit). While active, the motion can be canceled at any time by the address that has started it.

1. As soon as the motion duration expires, it becomes possible to enact the motion. Anyone can enact the motion from that moment on.
2. To enact the motion, the EasyTrack

contract passes the enactment EVM script to the `EVMScriptExecutor`

smart contract.

1. `EVMScriptExecutor`

holds all the permissions to run valid enactment EVM scripts passed from the EasyTrack

smart contract (or from Aragon Voting App).

## Changes from v1

The biggest change from the v1 architecture proposed in [the original LIP-3](#) is how we reduce interactions between core contracts and Easy Track type-specific contracts to only passing EVM scripts rather than distributing motion type logic among multiple Easy Track Executor contracts. This change was made to stick with uniform EVM Script Factory design and foster future Easy Track scalability.

The newly proposed EVM Script Factories cannot actually enact motions, while Easy Track core contract applies checks to make sure the submitted script does exactly what it is expected to do.

Another point regards using Aragon Voting App as a fallback option if Easy Track is attacked or misused. With EVM script based model, Aragon Voting App will be able to conduct votings on any proposal normally wrapped in an Easy Track motion.

## Sanity and security

The proposed Easy Track motions design involves timelock as a native security feature. Any suspicious motion can be rejected or canceled within a relatively long motion duration period. To prevent malicious actions, we propose putting the following additional limitations in place (numeric values fixed based on community feedback from v1):

- It should be impossible to set objections threshold at more than 5% of the total LDO supply. Nevertheless, the default objections threshold should be set a lot lower at 0.5% of the total LDO supply. Easy to start – easy to reject.
- It should be impossible to set motion duration at less than 48 hours. For any motion, there should be enough time for DAO to submit objections and reject the motion. For more urgent issues there's always an option to start a regular Aragon voting.
- It should be impossible to spam motions. The default limit for simultaneously active motions will be set at 12 and it can only be increased by the DAO to up to 24 motions at a time.

As an additional safety measure, EasyTrack

core contract is proposed to be pausable. If something goes wrong, the DAO will also be able to cancel any active motions without them reaching the objections threshold.