

This is a technical proposal related to Lido contracts implementation details. It doesn't change product logic.

In a nutshell, we propose an approach for writing initializer functions in Lido proxy contracts which might need to be upgraded. We also propose corresponding changes to LidoOracle

contract.

Much much more detailed reasoning and specifications of the proposal are placed in [Lido LIPs repo](#) as usual. This topic-starter is intended to give a quick impression on the topic and help to decide whether you'd like to participate in the discussion. The full LIP specification seems for me to be too cumbersome to be posted here.

Motivation

There are proxy contracts in Lido which might need upgrades from time to time. The contracts might need additional initializations during the upgrades. These contracts also need to be deployed from scratch from time to time (local testing, testnets, etc).

When all these requirements meet [common initialization pattern](#) doesn't help and we'd like to devise a single approach to the problem to be shared between all Lido contracts.

For now, only contract LidoOracle

has undergone such upgrades and it lacks capability to be deployed from scratch.

Summary

We propose an approach and a corresponding upgrade of LidoOracle contract.

The approach in a nutshell touches these questions:

- when to add one more initializer function and when to add some setters;
- contract versioning;
- keep vs. don't keep initializer functions of every upgrade;
- restrictions on calling initialization functions.

The LidoOracle proposed changes in a nutshell:

- add function initialize()

which performs initialization from scratch up

- change contract version in storage from 1 right to 3 via new function finalizeUpgrade_v3

References

- [full LIP-10 specification](#)
- [LidoOracle contract](#)
- [Pull request with the proposed LidoOracle upgrade](#)
- [OpenZeppelin proxy pattern](#)