LIP-6: In-protocol coverage proposal

Motivation

This improvement proposal provides a way for Lido governance to burn stETH as a means to distributed cover for losses to staking. It doesn't oblige Lido to cover for losses or introduce any auto-cover mechanism; just provides a clean option to use cover if DAO ever deems it necessary.

Currently, Lido has no adopted and well-defined mechanism of applying coverage for stakeholders' losses due to validators penalties, slashing, and other conditions.

There are discussions and research on coverage options that took place earlier:

- Should Lido use third party insurance providers?
- Offline & Slashing Risks: Are Self-Cover Options Enough?

To precisely specify coverage application without breaking existing principles, agreements, and integrations with stETH token (as a clear example we have in mind Anchor/bETH) we propose an in-protocol solution. The proposal is aimed to improve the overall technical transparency and completeness of the Lido protocol regarding a coverage application.

Proposed solution

We propose an on-chain in-protocol mechanism of applying coverage by burning stETH token (which may sound frightening at the first glance). The burn is limited by design with governance permissions hardened with fine-grained control: it could be invoked only by the proposed dedicated contract and only on its address.

To mitigate possible issues with external integrations we define exact time slots for the coverage application and provide externally available counters of ever burnt stEth.

Use cases

The proposed solution can be used to implement the cover cases (I mean after mass slashing/penalties events) and the non-cover cases (possible underperforming cases or rare penalties when there is no balance decrease between the consecutive Lido oracle reports).

After all, to be precise, it's an application mechanism definition, not a cover solution itself. The proposed solution is suitable for enacting both self-cover options, so for external 3rd party insurance providers.

Specification

Formal specification and further details provided on theLIP GitHub repo.