This is the continuation of the initial thread on dual governance. The linked thread contains an important context so please give it a read if you have time.

Since the last mechanism design version was proposed in this post, the protocol contributors working on DG have made several iterations to incorporate the received feedback and make the mechanism simpler, less fragile, and more efficient.

Before presenting the updated version, let me outline the problem we're trying to solve and briefly trace the chain of reasoning that has led us to the solution being proposed.

The problem

Currently, the Lido protocol code and its parameters are controlled by the Lido DAO via LDO token voting. The protocol takes a 5% fee from the staking rewards and directs it to the DAO treasury (another 5% being distributed to node operators participating in the protocol).

While LDO holders should generally be motivated to maintain the protocol's well-being since it's reflected in the LDO token price, it doesn't necessarily mean that LDO holders efficiently represent the protocol users. For example, imagine that LDO holders collectively decide to increase the protocol fees: while this might have a positive effect on the LDO holders' immediate well-being, it is clearly against the interests of at least some portion of the protocol users.

This can be generalized as a principal-agent problem (PAP)

between the DAO (the agent) and the protocol users (the principal). The problem exists because LDO holders don't have the exact same incentives as users.

Moreover, as Vitalik highlights in his Moving beyond coin voting essay, the PAP is exacerbated by the fact that economic interest in the protocol's revenue can be unbundled from the governance power:

one could skew the incentives of the DAO token holders by bribing them or borrow the DAO voting token on the open market to try getting enough voting power for pushing a change that's against the interests of both the DAO and the protocol users.

The presence of the PAP is not great but one can argue that, if users realize that the current agent doesn't represent them well enough, they can always leave the protocol and choose another agent that's better aligned with their interests or even decide to remove the agent completely via solo-staking.

This is a very important mechanism generally known as foot voting

. In theory, it should protect users from the negative effects of any incentive misalignment between them and the DAO or any attack on the DAO. However, in practice and in the specific case of the Ethereum liquid staking, the efficiency of foot voting is limited due to a number of factors at play.

The first factor is the specifics of how the Ethereum PoS works. To unstake ETH from a validator, one has to wait until the validator is fully exited, and all Ethereum validator exits are processed through a single queue with limited throughput. This means that the time required for leaving the protocol depends on external out-of-protocol factors and can vary by orders of magnitude. This, in turn, implies that imposing a static timelock on DAO decisions cannot guarantee that any user has enough time to leave the protocol before the DAO applies a change that is not in the user's interests.

The second factor is that a significant portion of users choose liquid staking because they want to re-deploy the staked capital to other forms of economic activities, resulting in liquid staking tokens (LSTs) being widely used in the DeFi, including the protocols that require additional time to withdraw from (e.g. lending markets). This adds one more external dependency that can prevent users from leaving the protocol within a pre-defined timeframe.

The third factor stems from the information asymmetry between the passive majority and active educated minority of users: correctly assessing all risks associated with a particular governance decision, including tail risks, requires the knowledge most of the users don't possess. Communicating the potential adverse effects of a DAO decision via the social layer takes additional time, reducing the probability of the passive majority leaving the protocol before the decision becomes executable.

Lido DAO has established a number of governance protocols for reducing the information asymmetry (e.g. the GOOSE framework, the Node Operators Sub-Governance Group, the LIP framework, the commitment for the minimum number of audits of any mainnet code change) but they are all social layer agreements between the current LDO holders and thus cannot protect from an external attack on the DAO.

Towards the solution

The ultimate solution to the problem is governance minimization and eventual ossification of the protocol code and parameters

. There's no governance risk if nothing is being governed.

Gradually minimizing the governance scope is something that the protocol contributors see as a necessity in the coming years. However, until the Ethereum specification ossifies, the code upgradeability can only be reduced up to a certain extent (e.g. see EIP-7002, EIP-7251). Additionally, any immutable code has to be formally verified on the bytecode level to exclude the possibility of a compiler bug producing a non-fixable vulnerability.

There's also the fungibility layer of the protocol that serves as the risk/reward assessment engine and distributes ETH between different validator subsets in a way that balances the yield and the risks of the resulting validator set. The risks here include the tail risks the validator set creates for the Ethereum network, e.g. censorability and correlated slashing risks. There's ongoing research (see this report for the latest iteration) on whether these risks can be estimated by the protocol with the help of a trustless oracle gadget bringing the required information onchain but it's a long-term endeavor and it's not yet clear how the desired outcome can be practically accomplished. Until the protocol has such a trustless mechanism implemented, there has to be some governance at the fungibility layer.

One more potential area of research is looking for ways of introducing an explicit opt-in to new code and parameter set versions for stETH holders and integrations. It's not yet clear whether it can be done without breaking the LST fungibility and the resulting liquidity fragmentation which, given that liquidity is one of the main factors driving users to LSTs, would destroy the protocol's competitiveness against other decentralized and centralized liquid staking providers. Nevertheless, it is an interesting research direction.

Now that we've established that the protocol will have to live with some kind of governance at least in the medium term, let's see how we can minimize the risks this governance creates.

Dual governance

As highlighted in the first section, the general problem can be decomposed into 1) the presence of PAP, and 2) the limited efficiency of foot voting. So ideally we'd want to introduce some mechanism that improves both the alignment between the DAO and the protocol users and the efficiency of foot voting.

That's where we arrive at the proposed dual governance design. It aims at the following improvements:

1. Give stakers a way to credibly signal their disagreement with the DAO and the commitment to leave the protocol if the DAO doesn't cooperate in resolving the incentives conflict.
2. Provide a negotiation device between the stakers and the DAO.
3. Introduce an extended dynamic timelock on DAO decisions that can be triggered by an active minority of stakers and prolonged as more stakers participate.
4. Improve foot voting efficiency by allowing stakers to exit the protocol without being subject to new and pending DAO decisions.

An overview of the proposed mechanism design and some ideas for future research on governance risk minimization can be found in this note: Dual Governance design overview - HackMD

.

It should be noted that stakers are not the only category of protocol users; there are also node operators. One potential future research direction is looking for ways to also improve the efficiency of foot voting by node operators, e.g. allowing a subset of stakers and node operators to coordinate a protocol and DAO fork by re-pointing validator withdrawal credentials to a new contract (not currently supported by the consensus layer).

Another direction of future research is exploring thenon-token and hybrid governance.

Next steps

From here, several things have to happen before the design is finalized, resulting in a more formal Lido Improvement Proposal (LIP) that will be submitted for a DAO vote and the associated Architecture Decision Record (ADR) document:

1. Evaluating the robustness of the proposed mechanism via scenario and attack modelling.
2. Evaluating the practicality of the mechanism via prototyping the code.
3. Gathering the community feedback.

This thread is aimed at accomplishing 3 while the protocol contributors work on 1 and 2 (both being currently in progress) so any feedback is highly appreciated!

It's important to highlight that, though dual governance is (in my opinion) an important step in reducing governance risks of the protocol, it's in no way the final step. Some of the ideas for further improvements can be found in the mechanism design document linked above, and I invite everyone interested to discuss those and any other potential improvements by posting a

topic on this forum.