# Lido Dual Governance

# Design Review & Risk Analysis

April 2024

*Prepared for:*
**Lido**

*Prepared by:*
**Tomer Ganor**

# Table of Contents

# Executive Summary

## Project Scope

| Repository | Last Reviewed Commit |
|---|---|
| https://github.com/lidofinance/dual-governance | b29d7055f6b9281b3c9e097e7ce4f77c4da03312 |

## Design Review Goal

This review aims to identify potential risks within the proposed Dual Governance design. By exposing unexplored vulnerabilities, we hope to enable the Lido team to adjust the design and mitigate these risks accordingly.

## Project Overview

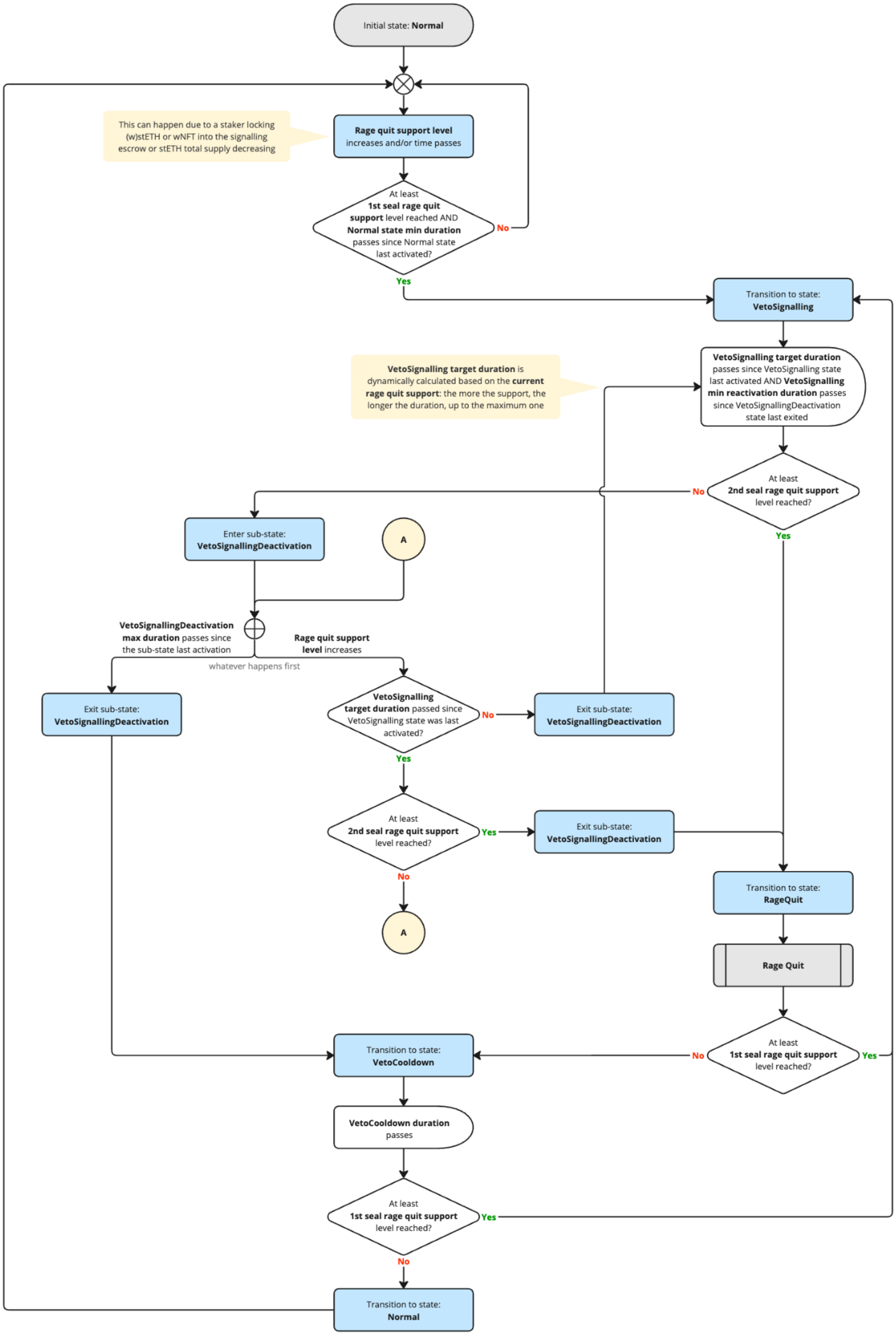
Currently, the governance framework of the Lido protocol consists of the Lido DAO, utilizing LDO voting for approving DAO proposals. Additionally, it incorporates an optimistic voting subsystem known as Easy Tracks for minor adjustments to parameters, defaulting to LDO voting in case of any objections from LDO holders.

The Dual Governance mechanism (DG) represents an evolution in protocol governance, empowering stakers to influence decisions by enabling them to veto DAO proposals. It serves as a negotiation tool between stakers and the DAO.

Dual Governance can be viewed as implementing the following:
1) a flexible, user-expandable timelock on DAO decisions.
2) a withdrawal mechanism tailored to the nuances of Ethereum withdrawals, accommodating staker preferences.

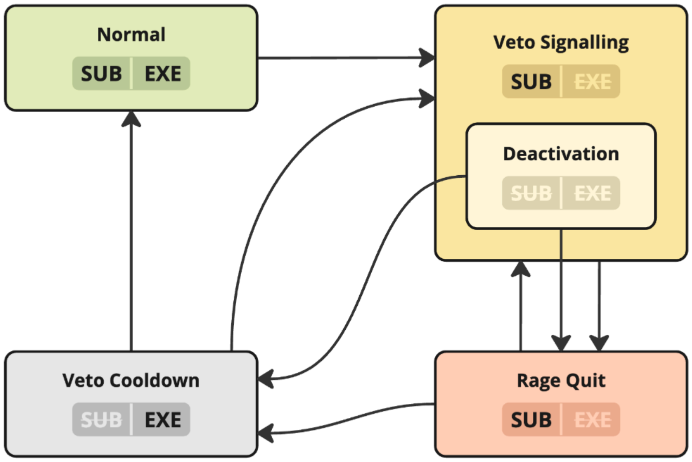
In the happy flow, if the DAO passes a bad or malicious proposal, stETH holders should be able to veto it by depositing 10% (current value, can be changed) of the stETH to the VetoSignaling escrow, waiting for 45 days  (current value, can be changed)  in which no proposal can be executed, and then withdraw the money in the RageQuit state. (Note that also during the RageQuit no proposal can be executed.) Upon completion of RageQuit, the system goes back to normal without the opposing stETH holders as part of the total stakes. The stakers can also manage negotiation with the DAO in order to cancel opposed proposals and continue as usual once they succeed.

The diagram on the next page describes the flow/stats transitions of the proposed design:


Initial state: **Normal**
This can happen due to a staker locking (w)stETH or wNFT into the signalling escrow or stETH total supply decreasing
**Rage quit support level** increases and/or time passes
At least **1st seal rage quit support** level reached AND **Normal state min duration** passes since Normal state last activated?
No
Yes
Transition to state: **VetoSignalling**
**VetoSignalling target duration** is dynamically calculated based on the **current rage quit support**: the more the support, the longer the duration, up to the maximum one
**VetoSignalling target duration** passes since VetoSignalling state last activated AND **VetoSignalling min reactivation duration** passes since VetoSignallingDeactivation state last exited
At least **2nd seal rage quit support** level reached?
No
Yes
Enter sub-state: **VetoSignallingDeactivation**
A
**VetoSignallingDeactivation max duration** passes since the sub-state last activation
**Rage quit support level** increases
whatever happens first
Exit sub-state: **VetoSignallingDeactivation**
**VetoSignalling target duration** passed since VetoSignalling state was last activated?
No
Exit sub-state: **VetoSignallingDeactivation**
Yes
At least **2nd seal rage quit support** level reached?
Yes
Exit sub-state: **VetoSignallingDeactivation**
No
A
Transition to state: **RageQuit**
**Rage Quit**
At least **1st seal rage quit support** level reached?
No
Yes
Transition to state: **VetoCooldown**
**VetoCooldown duration** passes
At least **1st seal rage quit support** level reached?
Yes
No
Transition to state: **Normal**

## Description of the State Machine

Dual Governance state machine has 4 different states:



1. **Normal:** in this state the system operates as usual: the DAO approves proposals, waits for the minimum timelock (currently 3 days), and executes the proposals. This state should be active 99% of the time,

2. **VetoSignaling:** the system transitions to this state when there is an opposition to a submitted proposal. In this state, no proposal execution is possible.
   In order to sustain this state, stakers must deposit at least 10% of total stETH, otherwise the system transitions to VetoSignalingDeactivation substate.
   - VetoSignalingDeactivation substate - in this substate there is no execution and no submission of proposals. This is the De-escalation phase.
     If RageQuit support increases while the system is in this state, it is possible to exit the VetoSignalingDeactivation substate and go back to regular VetoSignaling state.

3. **VetoCoolDown:** In this state, there is no VetoSignaling and no proposal submissions allowed. It is only possible to execute proposals that were submitted already. This state exists solely to enable the DAO to execute proposals after a successful de-escalation or a successful RageQuit. Note that this only happens if there aren't sufficient funds in the new veto escrow to start another VetoSignaling.

4. **RageQuit:** In this state there is no proposal execution and this is in order to prevent the DAO from influencing the stakers funds while they are rage quitting. This state may persist for a long time because the system waits for the complete withdrawal of RageQuit funds plus the additional time to claim the funds.

## Possible Required Paths (State Transitions Sequences)

These are the paths required to enable all the desired attributes of the Dual Governance:

1. Continuous Normal state
2. Normal → VetoSignalling → RageQuit → Normal
3. Normal → VetoSignalling → VetoSignallingDeactivation → VetoCooldown → Normal

# Found risks

## Risk-01: RageQuit loop can DoS the DAO

### Path

Normal → VetoSignalling → RageQuit → Normal

### Scenario Description

This path enables to loop between the VetoSignalling and the RageQuit. In both states there is no proposal execution possible (thus it is DoS of the DAO).
This can happen if someone deposits a minimum amount needed for RageQuit (currently 10% Of stETH), then after 45 days the RageQuit happens, once the attacker withdraws the money, he deposits again to the new veto escrow and does the same again and again to create the loop.
There is a protection from cycling: prolonging timelock on the assets withdrawn during the RageQuit. But an attacker that has two times the amount needed for the first attack can ignore this timelock by splitting his money into two attacks. As a last resort, the Lido team defined the Tiebreaker Committee.
In order to break the loop, the committee can execute pending proposals (that were submitted by the DAO). This solved this DoS and also solved a permanent deadlock that may have been caused by an error/bug in the withdrawal queue.
The execution power of the committee is only enabled after 1 whole year of the system not being in Normal / VetoCooldown states.

### Found Risk Scenario

The attacker will perform the above attack continuously (loop between the VetoSignalling and the RageQuit), when once a year, when the committee is about to get the execution power, at the end of the VetoSignalling-RageQuit cycle, the attacker will stop in Normal state for a Midstate in the transaction and return to the loop (This will cause a permanent DoS to the DAO due to the rest of the committee timelock).

### Mitigation

It was decided to change the flow of path 2:
Previous flow: Normal → VetoSignalling → RageQuit → Normal

New Flow : Normal → VetoSignalling → RageQuit → **VetoCoolDown** → Normal
This new flow prevents the raised risk since instead of the Normal state that can be exited immediately, the VetoCoolDown has a timelock of VetoCooldownDuration (currently 5h)
The Lido team implemented that solution to the current design.

**Note**:
There is still an option to DoS the system for a period of up to 1 year until the tiebreaker committee timelock passes but the stETH required for that is doubled than the RageQuit amount (currently 20%)

## Risk-02: VetoSignallingDeactivation loop DOS

### Path

Normal → VetoSignalling → VetoSignallingDeactivation → VetoCooldown → Normal

### Scenario Description

Let's assume a malicious DAO proposal. Once a malicious proposal has been submitted, someone may deposit stETH to the VetoSignalling escrow (triggering VetoSignalling) and withdraw the stETH right back. This should cause a cooldown in which you can run the malicious proposal with no veto option.
VetoSignalling deactivation substate was defined by the Lido team in order to prevent a situation in which the cooldown state is mistakenly achieved. This mistake transition to cooldown state can be done by 3 different paths:

- Malicious veto done by an attacker (as described above in the risk description).
- The time limit expired and the money came too late (we wouldn't want to cancel the veto when there is genuine support for this veto).
- Some of the people actually decided to withdraw the money back, but we still want to have a transition period to enable a fluent flow and not a drastic change with 0 response time.

### Found Risk Scenario

An attacker flashloans an amount of money that meets the threshold of the SecondSealRageQuitSupport, deposits it to the escrow, and right after, withdraws the money and repays the flashloan.

Now the state is VetoSignallingDeactivation.
The attacker will repeat the flashloan deposit and withdraw whenever the deactivation time is going to end.
In this state nothing can be done, no proposal submission and no execution. (the time duration it can be done in is up to 45 days.)

Once this attack is done, there is 1 day of cooldown (currently changed to 5 hours) in which we can only execute but cannot submit.

This can be done over and over again and cause DoS* to the DAO.

*There is a workaround in which the user in Lido can still submit during the deactivation state by doing the same attack but once depositing the money - submit the proposal and then withdraw the money. Of course, this is not the desired flow.

## Mitigation

In order to prevent this attack , the Lido team added a condition before the transition to VetoSignallingDeactivation that validates that the VetoSignalling state has been active for at least 5 hours.
The Lido team also added a timelock for each user, so a user now cannot deposit and withdraw in the same block, making the flashloan path irrelevant.
This way the described risk of this path does not prevent the submission of the proposal.

# Risk-03: Delaying VetoSignalling by increasing the needed stETH

## Path

Normal → VetoSignalling → (submit) → VetoSignallingDeactivation → VetoCooldown → Normal

## Scenario Description

Let's assume a malicious DAO. The DAO just deposits the secondSeal amount of funds and waits. Right before the VetoSignalling is over, the DAO submits the malicious proposal and withdraws the deposited stETH. Now the VetoSignallingDeactivation duration is 3 days, the same amount of time for the timelock of the submitted proposal. If stETH holders want to veto this new proposal they need to bring the secondSeal amount in 3 days and not the firstSeal which is much easier. They need to bring the secondSeal amount since we are at the VetoSignalling max duration (meaning 45 days have passed)  and the amount needed to be deposited to get back to the VetoSignalling state should be equal to the previous amount (before the withdrawal) and not the 1% as the firstSeal.
To address this scenario, the Lido team changed the deactivation time to be max of 3 days and 45 days from the last submission.
It seems like the DAO has no interest in doing that (submit at the end of the VetoSignalling state and prolong the deactivation duration) since it only delays itself (the DAO)

## Found Risk Scenario

If the above scenario takes place and the DAO submits a proposal, the attacker can exploit this stage by depositing 1% (firstSeal) and withdrawing once it's allowed. Now (after the above fix) the deactivation state duration will be ~45 days and not 3 days.
With only 1% of the money the attacker can hold back the DAO for 45 days and not as initially designed which was 8 days (5 days of VetoSignalling + 3 days of Deactivation).

## Mitigation

In order to prevent this attack, the Lido team changed the condition of the transition to a Deactivation state.
The initial condition was: T(R) passed since the VetoSignalling activation when R is the deposited amount.
Current condition: T(R) passed since the VetoSignalling activation or the last submit (the latest of them).

Now, the longest extra delay of the DAO that can happen is the delta between the activation and the submission which is very short if the DAO doesn't want to delay itself.

## New Risk Found

If the DAO is malicious, only 10% of the users can RageQuit together.
Let's assume a malicious DAO with 1% of the stETH, the DAO deposits to the escrow, and every 5 days submitted to a new standard proposal (not a malicious one), after 45 days, the condition for the transition to the RageQuit is almost met, now, once we have 10% of stETH deposited the RageQuit starts immediately.
In BAU (business as usual), we would be able to RageQuit with more than 10% because we had the first 45 days to wait for them, and if we have more than 10% we still wait for the 45 days to end. Now, since we already spent the first 45 days while the DAO delay loop happened, the RageQuit starts immediately when 10% is reached and therefore only 10% can RageQuit together at once. If there were 19% that wants to Ragequit only the first 10% can and the rest 9% are stuck.

### Offered Mitigation

Once the RageQuit is guaranteed, we enable users to join the RageQuit by waiting 2 additional days. The RageQuit will protect them from DAO influence.

## Risk-04: RageQuitAccumulation loop DoS the DAO

### Path

Normal → VetoSignalling → RageQuitAccumulation → Ragequit → VetoCooldown → Normal

### Scenario Description

The Lido team implemented the offered mitigation of Risk-03 but in their solution the RageQuit is not yet guaranteed when they enable users to join the RageQuit by waiting 3 additional days.
In the implementation, since the RageQuit is not yet guaranteed, you can unlock the stETH and go to the VetoSignallingDeactivation sub-state for up to 3 days.
Let's assume malicious stakers that want to DoS the system. They can do VetoSignalling for 45 days, reach the 3 days waiting for additional stakers to join the RageQuit, and then they unlock the stETH to get to the VetoSignallingDeactivation substate. Every 2 days, they lock the stETH to return the RageQuitAccumulation substate or unlock the stETH to move to the VetoSignallingDeactivation sub-state. This loop can go on forever and DoS the system.

### Offered Mitigation

In order to prevent the above risk and also simplify the whole system we offer to revert all changes done in risk #3 which are:
1. The change of the condition of the transition to Deactivation state from the initial condition which was: T(R) passed since the VetoSignalling activation when R is the deposited amount to the current condition which is: T(R) passed since the VetoSignalling activation or the last submit (the latest of them).
2. The change that once the RageQuit condition is met, users are enabled to join the RageQuit by waiting 3 additional days.

Instead of those changes we offer to implement one small change to the VetoCooldown state:

Instead of being able to execute any proposal that was submitted more than 3 days ago, we only allow to execute proposals that were submitted before the start of the VetoSignalling state (and more than 3 days ago).
By adding this change we achieve the following:

1. We enable the existing pre-VetoSignalling proposals to be executed as intended (if no one vetos them)

2. We validated that each proposal will get at least 3 review days by the stakers. If at the end of those days the firstSeal amount (currently 1%) of stETH is locked in the escrow, then the VetoSignalling state starts (as designed) and no ploys can happen.

**Note:**

In this suggested design we need to consider that the max delay time of a proposal is doubled in comparison to the previous design (since we may wait for the second iteration of the VetoCooldown).

This issue was discussed with the Lido team and it was decided to choose the parameters of the VetoSignaling duration accordingly.

## Risk-05: Tiebreaker committee gets power before it should

### Path

Normal → VetoSignalling → RageQuit → Normal

### Scenario Description

If 100% of stETH wants to RageQuit, the withdrawal time to finalize the RageQuit might take more than 1 year. Then the tiebreaker committee will get execution power although the system is not stuck (it is supposed to get the power only once the system is stuck).

### Lido's Response to this Scenario

It is a very rare use case (calculated time of more than a year to RageQuit) and we trust the tiebreaker committee to not abuse their power in such cases because it comprises lots of trusted entities.

# Recommendations

We specified important considerations that the Lido team should note in any future change to the design:

1. The defined response time of the stakers should always be preserved in any design change and taken into consideration in all veto-blocking following states (meaning, to consider the dependency of all states)
2. Need to validate execution time and submission time in every flow (to prevent DoS)
3. VetoSignallingDeactivation duration is also relevant to the response time of the stakers. (as changed from 1 to 3 days.)
4. When and if additional power to the DAO is added, it should be done very carefully because of the case in which the DAO may become malicious or by using some exploit an attacker may give themself more LDO tokens than exist and they get complete control of the DAO voting.

# Disclaimer

The Certora Prover takes a contract and a specification as input and formally proves that the contract satisfies the specification in all scenarios. Notably, the guarantees of the Certora Prover are scoped to the provided specification and the Certora Prover does not check any cases not covered by the specification.

Even though we hope this information is helpful, we provide no warranty of any kind, explicit or implied. The contents of this report should not be construed as a complete guarantee that the contract is secure in all dimensions. In no event shall Certora or any of its employees be liable for any claim, damages, or other liability, whether in an action of contract, tort, or otherwise, arising from, out of, or in connection with the results reported here.