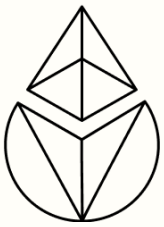
Lido V3 Technical Paper

December 2025



Authors

Eugene Pshenichnyy - Eugene Mamin - Max Merkulov - Alexey Potapkin

Abstract

LIDO V3 introduces stVaults — a new primitive that enables staking through user-defined validator setups, with optional stETH liquidity on top. stVaults are non-custodial smart contracts that delegate ETH to the chosen node operator while maintaining withdrawal credentials control. Stakers define key parameters as well: fees, MEV, sidecars, custody, insurance, and more.

Ether remains staked through the stVault and earns rewards, while the staker can mint stETH against it. Minting is overcollateralized: part of the stake is held back to mitigate slashing risk and preserve stETH fungibility. Liquidity on top of stVaults can be used to hedge price risk, participate in DeFi, or build complex staking strategy on top. The original model via the Core Pool — formerly known as Lido V2 — where users stake ETH and receive 1:1 stETH, remains fully supported.

stVaults are modular, sovereign building blocks for the staking economy. They are upgradable, but designed to let users opt out of Lido DAO governance in edge cases, such as governance capture. Adding stVaults as an additional option to mint stETH alongside the Core Pool makes the system more expressive and resilient: stakers can "vote with their setup" instead of relying on token governance to shape the validator set. The stVault architecture is open and composable, enabling node operators, protocols, institutions, and DeFi strategies to build on top — pushing Ethereum staking toward greater flexibility and decentralization.

Contents

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

1 INTRODUCTION

Ethereum staking presents a familiar dilemma: control or liquidity. Native staking gives users full control — over node operators, fees, legal arrangements, client software, and sidecar modules — but it locks up their capital. Liquid staking unlocks capital by issuing a fungible token (e.g. stETH) for staked ETH, but in exchange it limits the staker's ability to choose and configure validators.

Lido V2 is built around a shared pool idea. This design prioritizes liquidity and simplicity, but doesn't allow users to choose how their ETH is staked — the protocol selects the validator set and operating model. For many stakers, this approach works: it's easy, integrated, and liquid. Since mid-2024, ETH in liquid staking protocols stabilized in absolute terms, while its relative share slightly declined compared to native staking (by late 2025 nearly 30ETH is staked, but liquid staking now accounts for a smaller share of that total than it did in 2023).

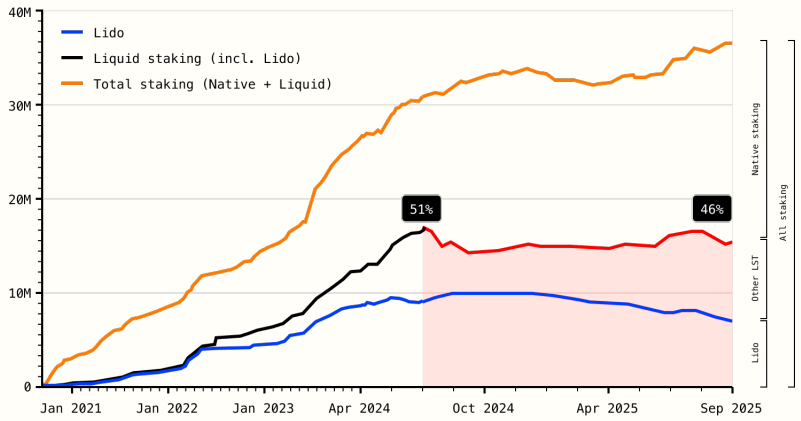


Figure 1: *Evolution of staked ETH by modality (Lido, other LST/LRT, native). Volumes in millions of ETH. Red area marks liquid staking from mid-2024 to mid-2025, with percentages indicating its relative share at the start and end of this period.*

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

This reflects a broader trend toward customization. Institutional participants require legal agreements with specific operators, and there is growing demand for custom validator setups and sidecars — whether for DVT, restaking, or other specialized use cases.

This tension between control and liquidity is now fragmenting the staking ecosystem. Users who want flexibility are forced to build custom infrastructure, often sacrificing access to shared liquidity. But the traditional shared staking pool model doesn't support customization directly: it assumes fungibility across all stakers, making it impossible to isolate how individual capital is staked or to reflect different preferences on risk, rewards, and validator choice. A more adaptable design is needed — one that decouples staking logic from liquidity, enabling diverse staking setups while preserving the benefits of a liquid token like stETH.

These considerations have shaped the foundation of LIDO V3 — a design that resolves the control-versus-liquidity dilemma in staking. Its core innovation is the stVault — an isolated staking position that gives users control over validator choice, fee terms, and infrastructure, while retaining access to stETH liquidity. stETH is minted against the stVault position in an overcollateralized manner, ensuring safety and maintaining liquid token fungibility.

stVaults can be used to support a wide range of staking use cases that were previously incompatible with a shared pool model:

1. **Delegation Liquid Staking:** Node operators can now attract stake directly and launch their own stVaults, while offering their stakers access to stETH liquidity, enabling delegation models without compromising on control or integration.

2. **Custom Staking Strategies and LRTs:** stVaults provide a base layer for building custom staking portfolios — from automated on-chain strategies to setups with off-chain governance. Capital can be split across multiple vaults, diversifying validator exposure and strategy risk. Protocols like Symbiotic can integrate restaking, platforms like Mellow can wrap stVaults, and markets like Pendle can unlock early access to rewards — all while maintaining local validator logic.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

3. **Leverage Staking:** Stakers can mint stETH against their staked position and use it in lending protocols to borrow more ETH and stake it back to stVault, creating capital-efficient loops. As long as staking yield exceeds borrowing costs, this structure increases the effective APR.

4. **Institutional Staking:** stVaults allow institutional players — such as ETFs, ETPs, liquid funds, or custodians — to retain full control over validator operations, without holding idle liquidity buffers. When needed, stETH can be minted and redeemed on secondary markets to meet withdrawals, while exposure remains intact. Inflows can later rebalance the position without operational disruption.

LIDO V3 supports this expanded range of use cases while continuing to offer the existing V2 staking path — a simple, default 1:1 flow into a shared staking pool. On top of that, stVaults extend the system into a modular, flexible, and sovereign staking infrastructure. Sovereignty means that stakers retain an exit path — similar in spirit to Dual Governance — allowing them to opt out of Lido governance in case of governance capture. stVaults are not just isolated configurations, but foundational components that other protocols, products, and institutions can build on.

2 STETH AS THE LIQUIDITY LAYER FOR ETHEREUM STAKING

There are two ways to mint stETH in LIDO V3 : Core Pool and stVaults. This section covers how they link up and how stETH stays backed across both.

The Core Pool (formerly known as Lido V2) is the classic way to stake: you send ETH and get stETH back at a 1:1 ratio. The ETH you stake goes into the Staking Router, which

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

spreads it across a set of Node Operators. Redemption works through an asynchronous withdrawal queue: first you request to exit, and once the request is processed, you can claim your ETH — also at a 1:1 basis.

stVaults extend the protocol's capabilities by enabling users to choose node operators and stake on customizable, non-custodial terms — while preserving access to stETH liquidity. In effect, stVaults turn staking into a marketplace where users can select who to stake with and under what conditions.

Each stVault is operated by a specific node operator and comes with its own rules — like fee structure, reputation, or validator preferences (for example, non-censorship). A staker selects a vault, stakes ETH, and can unstake at any time. Optionally, they can mint stETH against their vault position, unlocking liquidity without exiting. This stETH can be used freely — whether in DeFi, for hedging, or to build leverage. To exit a vault where stETH was minted, the borrowed stETH (plus fees) must be repaid first.

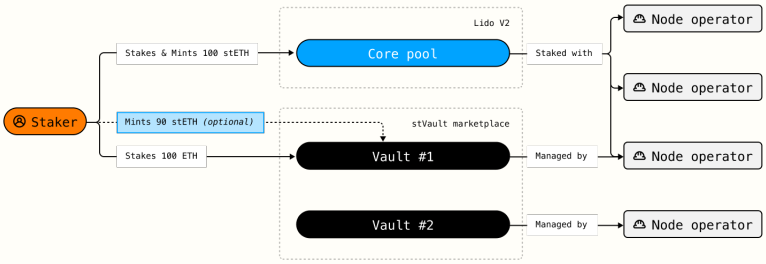


Figure 2: *Two ways to stake with LIDO V3. The Core Pool mints stETH 1:1 and delegates to a diversified validator set, while stVaults let stakers choose operators on flexible terms with optional stETH minting.*

The fundamental principle of stETH as a liquid staking token remains unchanged in LIDO V3: one stETH still represents one staked ETH and can be redeemed through the withdrawal queue. What has changed is how that ETH is backed. Previously, all backing came from the Core Pool. Now, with stVaults allowing stETH to be minted

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

against collateral, the system includes two types of backing: internal (ETH held within the Core Pool) and external (ETH staked through stVaults). Both contribute to the same unified stETH supply.

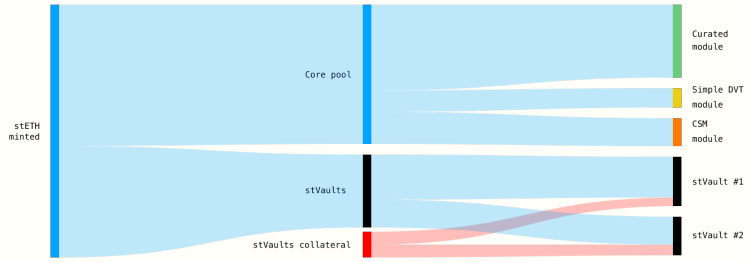


Figure 3: *stETH backing splits between ETH delegated through Core Pool modules and ETH staked in stVaults, the latter being additionally collateralized.*

The figure highlights the collateral structure behind stETH in LIDO V3 . The Core Pool is fully controlled by the protocol and its ETH is directly redeemable via the withdrawal queue. stVaults, by contrast, allow optional minting of stETH against deposited ETH. A portion of the ETH in each stVault is explicitly reserved as collateral and does not participate in stETH minting (shown in black), reinforcing the stVault's risk isolation and accountability.

The Core Pool, in addition to being the default staking path, acts as a stabilizing force within the system. It provides predictable liquidity, ensures that stETH can be redeemed directly through the protocol, and serves as an APR oracle for stVaults. Without the Core Pool, managing redemptions solely through stVaults would be difficult, as each stVault is isolated and its operator has no incentive to provide system-wide liquidity.

However, if a large number of users attempt to redeem their stETH at once, the Core Pool alone may not be sufficient. To handle this, LIDO V3 introduces a safety mechanism called liquidity-driven force rebalancing. In such cases, the protocol can pull ETH from

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

stVaults with outstanding minted stETH, effectively closing their positions and freeing up liquidity. Vault owners do not lose principal, only the opportunity cost of having their ETH temporarily locked. While this mechanism is designed for edge cases and is unlikely to activate under normal conditions, it acts as a critical backstop to ensure the system remains solvent and responsive even under stress.

stVaults give stakers more control — they can choose who runs their validators and under their terms. But that control comes with structure: each stVault is still tied to a specific operator. Within stVaults, stake centralization is mitigated through economic measures. Node operators with a larger share of backing face higher fees and reduced mintable collateral. The system is designed so that users who choose node operators that improve validator set diversity receive better terms. This creates a natural incentive to support decentralization through individual staking choices.

3 STVAULTS

3.1 Introduction

stVaults are the core innovation of the LIDO V3 — a modular, non-custodial staking primitive designed to meet the needs of both stakers and node operators. Stakers retain control over their staking position; node operators get access to capital for validation — without taking custody or bearing ownership risk. A staker creates a vault, deposits ETH, and links it to a chosen operator — delegating staking responsibilities, not ownership or control.

An operator may influence rewards or penalties because the staked ETH is subject to validator performance risks — including missed duties and slashing. However, they cannot access or move the principal. Control is preserved via mechanisms introduced in

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

recent Ethereum hard forks: a secure deposit flow using EIP-4788 (beacon block root in the EVM) and EIP-2537 (BLS12-381 signature verification), while validator exits remain staker-triggerable through EIP-7002. Custodial risk is removed by design — not by trust.

The independence of each stVault allows for granular customization of its validation setup, including client software, MEV policy, including relay selection, and sidecars, alongside a tailored fee structure. Its rewards are decoupled from stETH APR and benefit from automatic compounding, enabled by the adoption of EIP-7251. This EIP allows validators to accumulate balances up to 2048 ETH, significantly reducing the need for frequent reward skimming and increasing capital efficiency throughout the validator lifecycle.

The owner of a stVault has the option to mint stETH against their vault, unlocking liquidity from their staking position. To mitigate risks such as validator underperformance, a corresponding portion of ETH in the vault is locked as collateral. Typically, this collateral exceeds the minted amount by 5–50%, depending on parameters. Because stETH is a productive, rebasing token, the locked portion naturally grows over time with stETH APR.

The ability to mint stETH against a stVault introduces flexibility tailored to different use cases. Users can hedge against ETH price fluctuations, use the minted stETH across DeFi to earn additional rewards, and manage entry and exit from staking without waiting for deposit or withdrawal queues — a feature especially valuable for protocols, structured strategies, and ETF/ETP products built on top. Owners may also choose to leverage their staking position to amplify rewards.

stVaults are not only modular — they are sovereign. No vault can be forced to accept an upgrade, allow more than a user-defined portion of its ETH to be locked, or remain under protocol governance once the owner decides to opt out. In edge cases where governance becomes a risk, the ability to exit without unstaking offers a strong safety valve. And because validator accounting can be complex, asynchronous, and partially off-chain, stVaults are built to put risk management in the hands of the user. This reflects

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

a core Lido principle: ensuring control and resilience stay with the user.

3.2 Anatomy of stVault

A stVault is the minimal unit of staking logic in LIDO V3 — a self-contained contract that connects one staker with one node operator under clearly defined rules. It handles the entire validator lifecycle: from ETH deposit and validator registration to reward restaking, exit execution, and ETH withdrawal. Everything is on-chain, transparent, and non-custodial by default.

Conceptually, a stVault establishes a tripartite relationship where the staker, the node operator, and the protocol are interconnected through explicit on-chain rules. This structure ensures that ownership, liquidity provision, and validator control are clearly separated yet reliably coordinated.

The diagram below shows how the components interact:

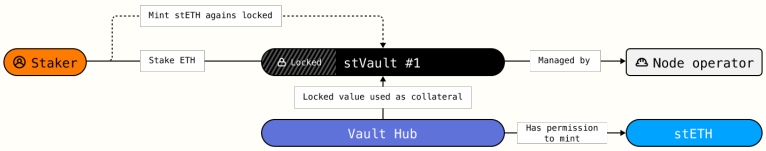


Figure 4: *The stVault is a coordination mechanism between the staker and the node operator. ETH is deposited by the staker and staked via the node operator. A portion of the vault's balance is locked by the Core Pool and used as collateral — this locked amount enables the staker to mint stETH against it.*

- **Staker/Owner** is the actor who provides ETH and manages the staking position within the vault. On-chain, this can be an externally owned account (EOA), a multisig, a DAO treasury, or any smart contract — such as a vault strategy, structured product, or staking aggregator. The staker initiates the vault, selects the

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

node operator, and controls validator actions, exits, and stETH minting. A single staker can own and operate multiple vaults in parallel, constructing diversified staking positions by distributing their ETH across different node operators and setups — tailoring risk, performance, and strategy to their specific needs.

- **Node Operator** is the actor responsible for running validator infrastructure on behalf of the vault. On-chain, this is represented by an Ethereum address — which may belong to a solo staker, a company, or a smart contract managing a staking pool or DVT cluster. The operator generates validator credentials and performs consensus duties, while custody of funds remains with the vault. At setup, the operator defines their service fee and selects a risk tier — a configuration that influences how much of the vault's ETH can be used as collateral during stETH minting. A single node operator can manage multiple stVaults in parallel, each with different stakers, validator clients, geographies, and terms — enabling scalable, modular delegation across the network.

- **stVault** is a smart contract that facilitates coordination between a staker and a node operator. It manages staking operations, enforces permissions, and defines control boundaries. Each vault is single-staker, single-operator, and fully non-custodial. Withdrawal credentials are set to an address controlled by the vault's internal logic, and exits can be triggered via EIP-7002 — ensuring operational control stays with the staker.

- **Core Pool** plays a critical role in liquidity issuance. It requests the VaultHub (on-chain ledger tracking every stVault's total value and locked value, serving as the central coordination point for the multi-vault ecosystem) to lock a portion of ETH within a stVault and treats this locked amount as collateral backing newly minted stETH. The staker can then mint stETH against this collateralized portion.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

3.3 Shared layers

While each stVault is independent and autonomous, it still relies on a set of critical protocol-level services that are too complex or infeasible to integrate directly into the vault. These services are referred to as "shared layers" and include capital-efficient consensus layer data oracles, stETH minting and burning, secure deposit flow, and a registry of Node Operators. On top of that, the protocol maintains a governance layer that helps roll out stVault upgrades.

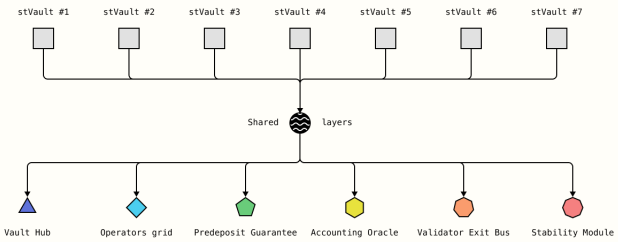


Figure 5: *This diagram illustrates the architecture where multiple stVaults (top row) interact with a common set of shared layers (bottom row).*

Shared layers:

- **Vault Hub** is a smart contract that vaults can call to mint and burn stETH. It enforces various parameters such as minting limits, reserve sizes, fees, and more. By centralizing these critical functions, VaultHub ensures that every stETH operation adheres to the protocol's rules, contributing to the overall security and efficiency of the system.
- **Operators Grid** is a smart contract that maintains the grouping of stVaults by risk tier and associated Node Operators. It also assigns a risk tier for operators who joined permissionlessly. Operators Grid is tightly integrated with VaultHub:

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

the rules governing how much stETH a vault can mint are determined by the operator's tier and the cumulative stETH minted across all their stVaults.

- **Predeposit Guarantee (PDG)** is a service designed to protect against front-running vulnerabilities during Deposit Contract calls — a risk that could allow node operators to steal deposited funds and/or lead to undercollaterized stETH minting. This module is crucial for maintaining the non-custodial nature of the setup, ensuring that node operators have no way to misappropriate users' ETH and mint stETH against non-stVault withdrawal credential validators. PDG functions for stVaults as an alternative DSM-like mechanism, based on the assumption that a 1 ETH pre-deposit verification can serve as a safe proof for activating a new validator, which in turn enables seamless stETH minting against ETH on that validator.

- **Accounting Oracle** is an efficient and well-served Lido core part that delivers essential data from the consensus layer to the protocol. In V3 update, it also enables VaultHub to make informed decisions based on up-to-date network conditions and provides the updated accounting data for stVaults.

- **Validator Exit Bus** decides on the validators that need to be exited to fulfill the stETH withdrawal requests and V3 update will enable it to maintain stETH redeemability, also, by rebalancing stVaults.

- **Stability Module** controls economic incentives to balance stVaults and Core Pool, slowly increasing the fees on stVaults and using it to increase stETH APR until it becomes balanced again.

3.4 stETH minting against stVault

The concept of stVaults was introduced earlier as a mechanism for non-custodial staking, where ETH remains under the staker's control. This section breaks down how overcol-

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

lateralized stETH minting works: how collateral and liabilities are structured, how they evolve over time, and what risks and opportunities they create for the vault owner.

When a vault owner mints stETH, they create a liability for the stVault: the protocol records the amount of stETH minted and locks an equivalent amount of ETH, plus an additional reserve, as collateral. The reserve serves as a safety buffer that protects stETH holders from potential risks related to validator underperformance, penalties, or slashing.

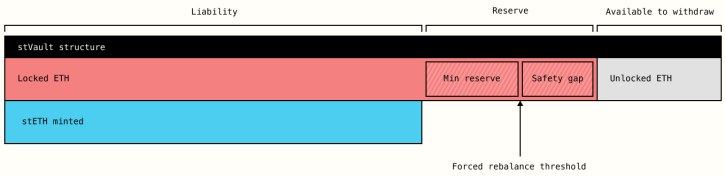


Figure 6: *The figure illustrates the structure of a stVault's staking position along with its additional stETH leverage. The locked staked ETH backs the minted stETH and includes a minimum reserve and safety gap to manage risks. The reserve serves as a safety buffer, while surplus value beyond reserve requirements is available for withdrawals from stVault.*

All of these values are dynamic. As staking rewards in the vault accrue, the amount of ETH in the vault grows at a rate that depends on the validation performance and the sidecar setup. The minted stETH amount increases as usual with the rate defined by the Lido Core APR, and the liability (as well as the collateral) amount increases due to stETH rebase and some fees charged by the protocol. As a result, the well-performing stVault maintains its position to be healthy over time or even can earn some upside by optimizing the setup.

The reserve ratio is an inherent stVault parameter that defines the relationship between the stVault's collateral and outstanding liability. It reflects how much extra ETH

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

is locked beyond what is needed to back the minted stETH. A higher reserve ratio indicates a safer position for the protocol with a stronger protection against validator risks, while a lower reserve ratio signals increased exposure.

If the total vault value falls below the forced rebalancing threshold, the vault becomes a target for the forced rebalancing. During this process, a portion of the collateral is transferred to Lido Core, while the vault's outstanding liability is reduced.

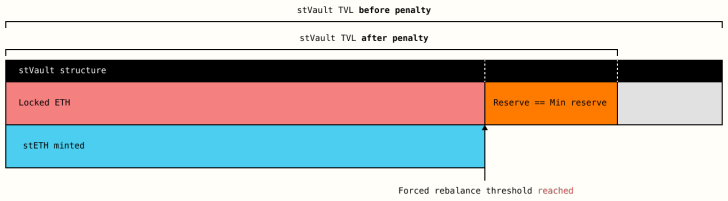


Figure 7: *Structure of an stVault during a slashing event. The locked ETH buffer absorbs penalties, with the minimum reserve serving as the final protection layer. If the penalty depletes the locked ETH down to the minimum reserve, the vault reaches the forced rebalance threshold, triggering forced rebalancing*

This adjustment restores the reserve amount to a healthy level without requiring the full liquidation of the position and without the vault owner losing the control over the capital. Forced rebalancing acts as a soft liquidation mechanism: the vault owner loses some upside — as part of their principal is moved to the common pool — but retains most of their capital in the vault and continues to benefit from staking rewards. The threshold is designed to be chosen in a manner that does not allow the vault to be rebalanced because of bad luck but primarily because of slashing or negligence to the performance of the validation.

Additionally, a small penalty is applied during forced rebalancing to cover gas costs and reward the actor executing the operation, ensuring the mechanism remains efficient

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

and economically viable.

To unlock the collateral and regain the ability to withdraw ETH from the vault, the stVault owner must repay their outstanding liability. Repayment involves returning the minted stETH adjusted for rebases and any accrued protocol liquidity fees. Once the liability is fully covered, the locked collateral becomes eligible for withdrawal or reuse. It is important to note that the release of collateral happens asynchronously, synchronized with the oracle updates (typically once per day), following the same principle used for withdrawals in the Core Pool.

Summarizing the actions available to a stVault owner:

1. **Deposit** — Add ETH to the stVault, increasing the position value and improving the health factor.

2. **Withdraw** — Retrieve available (unlocked) ETH from the stVault at any time. To unlock the remaining collateralized ETH, all outstanding liabilities must first be repaid.

3. **Mint** — Mint stETH immediately after depositing ETH into the stVault, independent of whether the node operator has already staked the funds. It's important to note that until the ETH is fully staked and starts earning rewards, the liability will continue to grow while the underlying ETH does not, which can slightly degrade the health factor over time.

4. **Repay** — Return the minted stETH adjusted for rebase (plus any accrued fees) to repay liabilities. Some operations, such as collateral release, occur asynchronously and are synchronized with oracle updates (similar to the withdrawal queue mechanism in the Core Pool).

5. **Rebalance** — Settle a portion of the outstanding liability at the current rebase-adjusted rate by reallocating a part of the stVault's collateral to the Core Pool. Rebalancing can be initiated manually by the stVault owner or triggered automat-

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

ically if reserve ratios fall below critical thresholds or liquidity conditions deteriorate (see details in the following sections). It's important to note that no token conversion occurs during rebalancing: the principal capital remains intact, while only the leverage and the effective size of the underlying position are adjusted.

Detailed user flows for each action are provided in A

3.5 Fees

The system defines two categories of fees: Node Operator Fees and DAO Fees. All fees are configured granularly per stVault and are automatically collected by the protocol.

1. **Node Operator Fee** are charged on actual staking rewards earned by the staked ETH within the stVault and collected by the assigned Node Operator.
2. **DAO Fees** consist of two parts:
    - **Infrastructure Fee:** charged on deemed staking rewards — a synthetic calculation based on the stVault's total value and the Core Pool's daily reward rate. This approach decouples DAO rewards from validator self-reporting (since Execution Layer rewards are difficult to verify on-chain). In the simplest version, the infrastructure fee can be calculated as a fixed percentage of the stETH APR.
    - **Liquidity Fee:** charged on the amount of minted stETH and accumulated over time while the stETH remains outstanding. These fees compensate the DAO for providing liquidity services.

In addition, a **Reservation Liquidity Fee** may be applied if a stVault owner wants to secure the right to mint stETH at a specific reserve ratio in the future. Since high stake concentration under a single operator increases slashing risk (necessitating higher reserve ratios), reservation fees allow vaults to lock in future minting capacity without competing against other stVaults for limited liquidity.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

3.6 Balancing stETH risk profile and redeemability with staker-driven Node Operator selection

Previously, stETH was solely collateralized by the Core Pool — a diversified set of validators coordinated by the Staking Router. Now, stETH can also be minted against isolated stVaults operated by individual Node Operators, which have different security properties and liquidity characteristics compared to the Core Pool. This shift creates a fundamental challenge: the protocol must reconcile these two distinct sources of backing to maintain stETH's simplicity, fungibility, and reliable redeemability for holders.

To address this, we separate the problem into two parts: managing the slashing risks associated with stETH backed by heterogeneous validator sets, and maintaining stETH's redeemability across a system now composed of both Core Pool and stVaults. Each presents its own challenges and requires targeted solutions, which we explore in the following sections.

3.6.1 Designing the Risk Framework for stETH Slashing Exposure

One major systemic risk for stETH is mass slashing: if a significant portion of validators backing stETH — potentially across one large or multiple Node Operators — are slashed within a short timeframe, the resulting losses are distributed across all stETH holders. Even a relatively small percentage of slashed validators can lead to outsized penalties due to the quadratic penalty escalation built into Ethereum's slashing rules.

Although slashing events have been rare historically, and no mass slashing event has ever occurred on the network, they represent a major systemic risk. As correlation penalty grows linearly with share of network slashed, the total risk impact on this component is quadratic dependence on stake concentration. This risk must be carefully considered in the system's design.

For the curated parts of the validator set in the Core Pool — where the protocol can reliably assess the independence of validators, operators, technologies, or modules

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

(e.g., curated operator sets or known DVT modules) — slashing risk is mitigated by distributing stake across distinct risk domains. Wherever possible, ETH is allocated across different entities, consensus clients, execution clients, geographies, and staking architectures to minimize the likelihood of simultaneous failures. Risk profiles are tracked at the level of each diversification dimension, allowing the system to spread exposure intelligently rather than blindly.

Within the Core Pool, where validator independence cannot be reliably verified — as in permissionless modules — the protocol cannot rely on diversification to mitigate correlated slashing risk. Instead, validators participating through such modules are required to post bonds to internalize potential losses. To further limit systemic exposure, the total size of the validator set under these assumptions is restricted, treating all participants as a single correlated entity in worst case.

In stVaults, the situation is both similar and different: there are verified Node Operators as well as permissionless participants, but stakers, not the Staking Router, decide how stake is allocated across vaults. From the perspective of minimizing systemic risk for stETH, the critical factor is not the stake distribution itself — especially given that an operator's total exposure may extend beyond Lido protocol — but ensuring that each staking position is sufficiently bonded to protect stETH holders against catastrophic consequences.

To anchor the risk framework, the protocol adopts a conservative invariant: each Node Operator is assumed to have **up to 1% of the total Ethereum network stake outside of their activity within stVaults (ETH staked which may be used as a collateral)**. This assumption captures external exposures that are not directly visible within the protocol but could contribute to correlated slashing risk. It provides a baseline for determining the amount of collateral that must be locked when minting stETH. The chart below shows the reserve ratio — the percentage of collateral required relative to minted stETH — needed to cover the correlated slashing risk of a single Node Operator based on their concentration.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

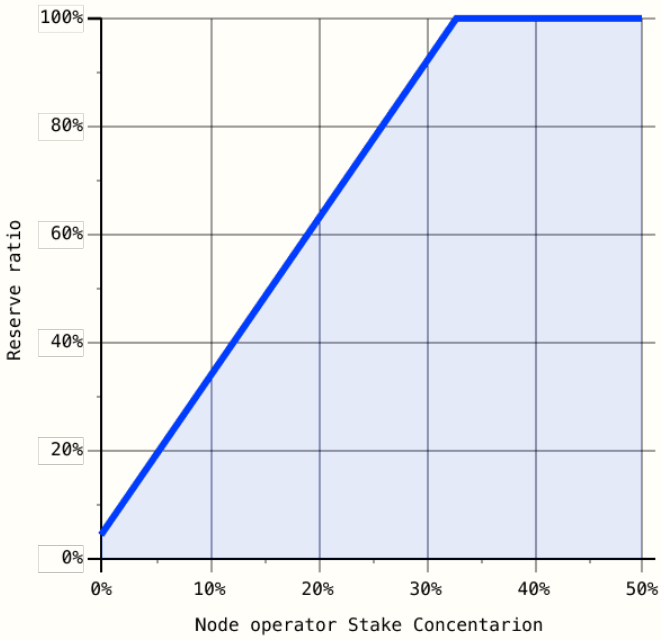


Figure 8: *Reserve ratio required to cover correlated slashing losses for a single Node Operator assuming 1% external network stake exposure.*

While this approach works for calculating the baseline reserve ratio for a single Node Operator, it does not fully capture the dynamics introduced as an operator grows and manages multiple stVaults and stakers. As the number of stVaults and the volume of stETH minted under a single operator increases, the effective reserve ratio weakens, amplifying systemic risk. This creates a dilemma: either the reserve ratio must be made dynamic — allowing the actions of one staker to affect others and potentially triggering forced rebalances — or the protocol must conservatively fix the reserve ratio upfront and restrict the operator's growth, thereby incentivizing Sybil behavior to bypass exposure limits.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

To resolve this, a different principle is introduced. Early stakers preserve the conditions they entered under, including their original reserve ratio and risk assumptions. New stakers joining later also preserve their entry conditions, but they must absorb the additional systemic risk created by the increased cumulative exposure. This approach fairly internalizes the risk externalities: participants who add new risk to the system are the ones who bear its cost, rather than spreading it across earlier participants.

Figure 9: *composed of six panels, demonstrates approach to setting the Reserve Ratio (RR) across successive Tiers, showing the potential risks of calculating RR independently for each Tier.*

While this principle ensures fairness and strong economic alignment, applying it directly would make the system overly complex and difficult to navigate for users. This naturally creates a competition, as tiers offering the most favorable conditions on reserve ratio are limited, representing the described structure of transferring growth in risk exposure to subsequent tiers. On a baseline level it's a fair system with stakers

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

coming earlier having the advantage, with a possibility of market build on top in the future.

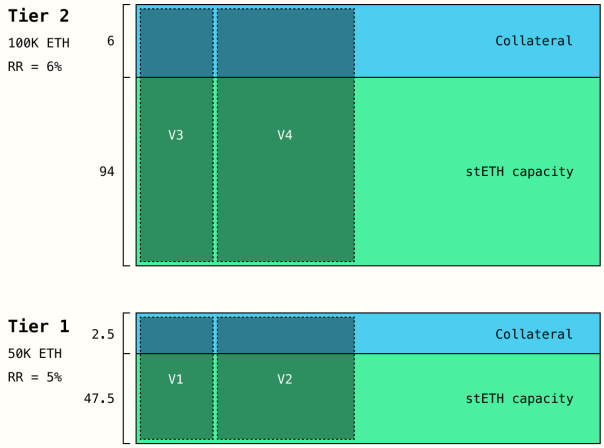


Figure 10: *Figures 10: Illustrative example of tier sizes and RRs.*

This tiered approach introduces a natural economic disincentive to centralization. As a Node Operator's exposure grows, the increasing bond requirements make it progressively more difficult to offer competitive terms for minting stETH. This dynamic encourages stake to flow toward smaller or less concentrated operators, mitigating systemic risk and promoting a more decentralized validator set.

A key invariant of the collateralized tiering system is that the collateral associated with each stVault is used to cover slashing penalties related to that stVault's validators. Collateral is always available to absorb validator losses, but is utilized in full — particularly for tiers with lower reserve ratios — only if a correlated slashing event occurs. If a stVault's validators are unaffected, or if only isolated slashing incidents happen, the collateral remains largely intact. Even if a major Node Operator experiences widespread slashing and correlated penalties are triggered, a stVault's collateral will not be touched

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

unless its own validators are directly impacted.

Although there is a tech for Risk transferring represented on Figure 9. As slashing penalties related to particular stVault's validators could also include increase in correlation penalties for stVaults in previous tiers, but only in cases where:

1. Collateral from previous stVaults tiers is fully utilized
2. Slashing of validators from stVaults in further Tier directly affected correlation penalties for validators from stVauts from previous Tiers

In summary, the Core Pool manages systemic risk through diversification and targeted bonding requirements, while stVaults address risk via collateralization and structured reserve ratios. The design further discourages stake concentration through progressively increasing bond demands, counteracting the natural centralizing tendencies of delegated staking systems with free operator choice.

3.6.2 Maintaining stETH Redeemability with stVault Constraints

Redeemability refers to the ability of all stETH holders to exchange their tokens for ETH on a 1:1 basis through the protocol's internal mechanisms, independent of the size of their position or the way in which the stETH was acquired.

In the Core Pool model, redeemability was maintained through the withdrawal queue and the Validator Exit Balancing Oracle (VEBO), which optimized validator exits and the fulfillment of withdrawal requests efficiently and fairly. Since all ETH was socialized within the Core Pool, the protocol managed validator exits centrally, while stakers did not need to be concerned with which specific validators or Node Operators were exited to satisfy withdrawals.

The introduction of stVaults complicates this dynamic. Part of the ETH backing stETH now resides within vaults controlled by individual vault owners. These owners expect to benefit from the liquidity of stETH without having their staking positions forcibly unwound to satisfy unrelated withdrawal requests. This creates a fundamental

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

tension between maintaining universal redeemability for all stETH holders and protecting the autonomy of individual vaults.

To illustrate the challenge, consider a simple model: suppose the Core Pool holds 50 ETH, and there is a single stVault holding 100 ETH. The vault owner mints 90 stETH against their vault and sells it on the secondary market for DAI. A trader who acquires the 90 stETH subsequently submits a withdrawal request through the withdrawal queue. However, the Core Pool's available liquidity of 50 ETH is insufficient to fulfill the request.

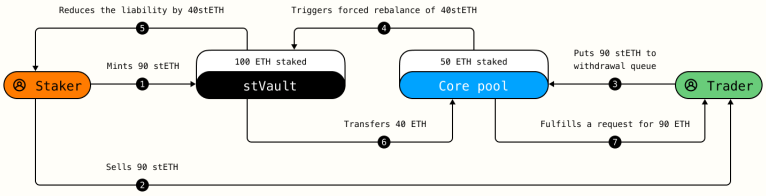


Figure 11: *Force rebalancing flow for stVault.*

In this case, to uphold the redeemability guarantee for the trader, the protocol must initiate a forced rebalancing of the vault. This process allows a portion of the vault's collateral to be used to fulfill the withdrawal, ensuring that the 1:1 exchange rate between stETH and ETH is maintained system-wide.

In situations where the Core Pool lacks sufficient liquidity to fulfill withdrawal requests, the protocol introduces a secondary rebalancing trigger: **liquidity-related forced rebalancing**. This mechanism follows the same logic and structure as reserve-ratio-driven forced rebalancing but is activated by liquidity shortages rather than stVault health factor degradation.

The key challenge in liquidity-driven rebalancing is determining which vaults should be adjusted first, while also giving vault owners a fair opportunity to proactively manage their positions before forced intervention. To address this, liquidity-related forced

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

rebalancing is subject to two constraints:

1. **Rebalancing delay:** A delay period is introduced before forced rebalancing is executed, allowing vault owners to deleverage, repay liabilities, or otherwise adjust their positions voluntarily.

2. **Risk prioritization:** Vaults with the highest outstanding liabilities relative to their collateral are rebalanced first. This ensures that those who have minted the most stETH and contributed most to the system's liquidity stress bear the primary responsibility for restoring balance.

Liquidity-related forced rebalancing is intended strictly as a **last-resort mechanism**. The design of LIDO V3 introduces a dedicated mechanism to ensure that the Core Pool retains sufficient liquidity to fulfill withdrawal requests under normal conditions. Measures to maintain this invariant are discussed in the following section.

3.7 Core Pool Sustainability Incentives

As discussed in the previous section, stETH can be minted against both Core Pool deposits and stVault collateral, but all redemptions must be processed exclusively through the Core Pool. This structural asymmetry creates pressure on Core Pool liquidity that must be addressed by protocol mechanisms. The following section outlines the mechanisms designed to balance this asymmetry and align the incentives between vault owners and the Core Pool.

Vault owners prefer to avoid having their positions used for fulfilling withdrawal requests, except as a last resort through forced rebalancing. This places the primary burden of satisfying withdrawals on the Core Pool. To ensure that the Core Pool can reliably perform this function, its size must be supported through explicit mechanisms.

A critically depleted Core Pool would not only delay withdrawals but would also collapse the isolation guarantees that stVaults rely on. Without sufficient Core Pool liquidity, withdrawal pressure would be absorbed directly by stVaults through forced

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

rebalancing, exposing vault owners to external risks and eroding the economic viability of stVaults as a product. In this sense, vault owners themselves are structurally incentivized to support the Core Pool as a shared liquidity buffer.

Maintaining a sufficiently large Core Pool is critical not only because of liquidity considerations but also because the Core Pool serves as the protocol's APR oracle — a topic discussed later in the Core Pool section.

Given this natural alignment of incentives, the protocol may formalize the relationship between stVaults and the Core Pool by introducing an economic mechanism where a portion of stVault rewards is redirected back into the Core Pool via a liquidity fee. This would allow vault owners to continue benefiting from Core Pool liquidity while proportionally contributing to its maintenance, encoded directly in smart contracts.

In practice, Lido DAO should define a policy — implemented either as governance rules or an on-chain controller — to target a desired Core Pool size. The target must combine two factors:

1. **Relative share** of minted stETH (e.g., ≥30% of TVL), based on the need to withstand maximum expected liquidity outflows.

2. **Absolute minimum** balance in ETH (e.g., ≥1M ETH), ensuring that the Core Pool remains large enough to serve as a reliable APR oracle.

If the Core Pool balance falls below the target threshold, the controller can adjust the liquidity fee applied to stVaults, increasing it until stakers are economically incentivized to migrate liquidity back into the Core Pool — and decreasing it when Core Pool liquidity exceeds targets. The economic effect here is on both sides - as fee is directly increasing stETH APR, it's simultaneously improving Core Pool market proposition, and decreasing expected user APR across stVaults, as DAO fee components there are using stETH APR as Oracle. If liquidity fees alone prove insufficient to restore balance, the protocol retains the option of applying forced rebalancing as a last resort to protect redeemability and system stability.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

The mechanism operates similarly to a central bank interest rate policy: liquidity costs are adjusted progressively to stabilize the system around the intended equilibrium. Over time, the liquidity fee structure could be further refined to account for differences in collective risk contribution and network health across various stVaults (e.g., validator client diversity, Node Operator concentration, geographic distribution), making the liquidity market more fair and balanced for all participants.

In the initial deployment of LIDO V3 , Core Pool sustainability will be preserved by hard-capping the mintable capacity of stVaults to 30% of Lido Core. This simple constraint allows the protocol to maintain sufficient liquidity for withdrawals without relying on more complex dynamic balancing mechanisms during the early stages of system growth.

3.8 Sovereignty

LIDO V3 assumes trust between stVault owners and the protocol governance, but embeds sovereignty guarantees to protect against governance and oracle risks. Sovereign stVaults not only shield individual stakers, but also minimize the value of governance capture: even if control over governance is seized, the economic upside is limited by the autonomy of individual stVaults.

Two risks make sovereignty essential:

1. **Oracle-driven liability** — stVaults rely on asynchronous oracle updates to determine the amount of locked collateral. Despite decentralization, oracle mechanisms are inherently subject to bugs, delays, or manipulation. An inaccurate oracle reading could cause incorrect minting of stETH, exposing vault owners to liabilities they cannot control.

2. **Governance capture** — The Lido DAO retains the ability to upgrade stVault code. Although critical for security patches and network compatibility, this introduces a systemic risk: if the value locked in stVaults exceeds the effective cost of

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

controlling governance (e.g., acquiring sufficient LDO voting power), malicious or misguided upgrades could harm vault owners. Potential risks include forced collateral increases, fee changes without consent, or even seizure of vault assets.

To address these risks, LIDO V3 introduces two sovereignty tools: an immutable collateral cap and an escape hatch, giving vault owners fine-grained control over their trust exposure.

3.8.1 Immutable collateral cap

Since each stVault operates as an independent contract, it can enforce an immutable cap on how much of its ETH stake is eligible both to back minted stETH and to be subject to forced rebalancing. For example, a vault owner may set a hard limit of 40%. ETH above this threshold remains completely outside the oracle's accounting reach and cannot be locked or reallocated by protocol mechanisms.

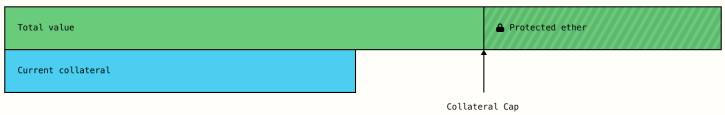


Figure 12: *Collateral cap limits how much of a vault's ETH can back stETH or be force-rebalanced.*

This cap acts as a strict on-chain constraint: no oracle manipulation can cause the stVault to expose more than the agreed share. Owners should size the cap carefully, as a lower cap reduces the collateral buffer for covering penalties and may trigger forced rebalancing sooner under adverse conditions.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

3.8.2 Escape hatch

Each stVault remains upgradable by the Lido DAO while stETH is outstanding, allowing critical updates like hard-fork support and risk parameter adjustments. However, upgradeability introduces a tail risk for stakers, as analyzed in the Dual Governance research, where governance-driven changes could alter the social contract between Lido DAO and protocol participants.

If a vault has no outstanding stETH, its owner may trigger a one-time opt-out. The escape hatch freezes the vault's code at the last trusted version and prevents future DAO upgrades from applying to it. From that point, the vault operates fully under the owner's control — at the cost of losing the ability to mint new stETH until opting back in.

Unlike Dual Governance models, the decision is unilateral and requires no coordination with other stakers, node operators, or voters. As a result, governance cannot impose new code or collateral changes, and contentious DAO votes cannot trigger mass validator exits or destabilize Ethereum consensus. Another advantage of the escape hatch is fork-friendliness: vault owners can detach from protocol governance and gradually withdraw their stake elsewhere without disrupting network operations.

Both the escape hatch and the collateral cap are embedded directly in each stVault's code and verifiable on-chain. Together, they allow stakers to define their own balance between liquidity and sovereignty: shielding individual positions from unwanted governance changes, turning trust assumptions into enforceable code, and improving fork resilience.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

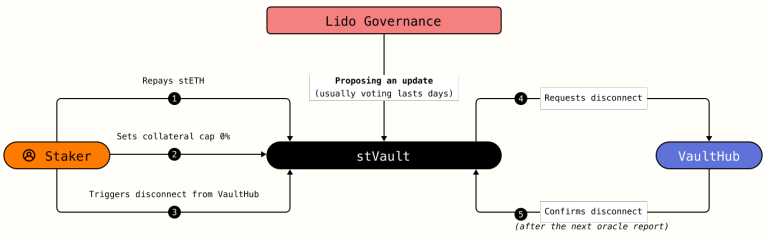


Figure 13: *stVault opt-out and disconnection from governance upgrade flow*

Practically, a stVault owner wishing to disconnect can fully repay outstanding stETH, set the collateral cap to protect all the stVault's ETH, and trigger the escape hatch. After the next oracle update, the vault becomes fully sovereign — isolated from governance upgrades and forced rebalancing, while continuing to operate independently.

stVaults are not only modular — they can also stand on their own. With sovereignty features built in, stVaults are not only safer for individual stakers but can also serve as foundational building blocks for new protocols. By weakening or making optional their connection to Lido governance, they remove a critical bottleneck for scaling and integration across the broader staking economy.

4 CORE POOL

4.1 Current State

The Core Pool remains the backbone of Lido's liquid staking ecosystem. While LIDO V3 introduces stVaults for customizable staking setups, the Core Pool continues to offer a simple, decentralized, and intuitive way to stake ETH and mint stETH at a 1:1 ratio.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

Stakers don't need to choose node operators, compare rewards, or manage configurations — they just stake. The Core Pool focuses on vanilla Ethereum staking and handles everything under the hood by distributing stake across a diverse set of node operators. It also enables straightforward redemption of stETH at a 1:1 ratio through the in-protocol withdrawal queue.

The validator set of the Core Pool is modular by design and managed by the Staking Router. As of December 2025, stake is allocated across a highly diverse set of over 680 Node Operators, spread across three active modules: the Curated Module, Simple DVT, and the Community Staking Module (CSM). These operators include home stakers, semi-professionals, and professional validators, many of whom enter through modules that leverage Distributed Validator Technology (DVT) and bonded entry mechanisms.

This section explains how the Core Pool interacts with stVaults — the other major component of the protocol — and how their relationship balances flexibility with stability. It also outlines what's next for the Core Pool: what remains to be improved, which modules are planned to add, and what trust assumptions might be hardened as the system continues to evolve.

4.2 How the Core Pool Anchors the Protocol

4.2.1 Core Pool as a Liquidity Buffer

In Lido's architecture, anyone can withdraw from the Core Pool, while withdrawals from stVaults are restricted to their respective owners. This introduces a structural asymmetry: if the Core Pool is depleted, it may be unable to fulfill withdrawal requests — even if ample ETH is locked in stVaults. This creates a liquidity mismatch that must be resolved for the protocol to remain robust.

To address this, the system can introduce economic incentives. For example, as stVaults grow in size, a portion of the fees collected from stETH usage could be redirected to the Core Pool as a reward. In effect, stVaults would "pay for the privilege

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

of not being withdrawal-liable" in the first place, contributing to the liquidity that the Core Pool must guarantee on their behalf. In early versions of the protocol, however, such stabilization mechanics may not be in place. Instead, stVaults may be required to remain smaller than — or at most, equal to — the size of the Core Pool, ensuring that liquidity obligations can still be met under stress. This conservative constraint helps safeguard withdrawals until more dynamic rebalancing mechanisms are deployed.

With such incentives in place, the Core Pool serves as a shared liquidity buffer for all staking activity. It ensures seamless withdrawals, even if most ETH is concentrated in stVaults. This strengthens the protocol's resilience and reliability, giving users confidence that stETH can be redeemed at any time.

However, if the Core Pool is exhausted despite incentives, the system must initiate controlled withdrawals from stVaults backing outstanding stETH. These withdrawals should be prioritized by risk, starting with vaults closest to their rebalancing thresholds. This is an edge-case scenario, but one that vault owners must understand and account for — even if such withdrawals are only triggered under extraordinary withdrawal pressure. This mechanism helps preserve systemic stability while resolving liquidity shortages in critical moments.

4.2.2 Core Pool as APR Oracle

Beyond its role as a liquidity buffer, the Core Pool also serves as a performance oracle for the entire protocol. It provides a transparent reference point for the baseline rewards of vanilla Ethereum staking — including MEV — and is used to calculate infrastructure fees for stVaults, since directly measuring hidden MEV is often infeasible.

However, most of the Core Pool's stake is still concentrated within large node operators from the Curated Module. This creates a potential attack vector: a group of operators could collude to slightly underreport performance — even by a few basis points — effectively reducing their liability for infrastructure and liquidity fees, both of which are influenced by the protocol-wide stETH APR.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

To mitigate this, it's critical to maintain a sufficiently large and diverse Core Pool, making it harder to manipulate and ensuring that the reference APR remains smooth and resistant to MEV windfall spikes. The more stake distributed across independent modules — including those using DVT or bonded access — the harder it becomes for any subgroup to meaningfully bias the oracle. Additionally, the protocol can implement a wide-scope performance oracle that aggregates validator performance from multiple sources, and introduce disincentives for underperformance, further reducing the risk of manipulation.

4.2.3 Modules vs stVaults

With the introduction of the Staking Router, Lido modularized the validator set — enabling stake to be distributed across pluggable staking modules, each supporting different types of Node Operators. This was a major architectural shift that separated the stETH token and accounting logic from the process of stake delegation.

Now, with the introduction of stVaults, the protocol gains a second modular primitive — one that allows stakers to define their own validator sets, fee structures, and risk parameters. Naturally, this raises the question: can we converge on a single primitive? It's possible that in future versions of the Lido protocol, the Staking Router will be extended to delegate stake directly to stVaults, effectively combining the two abstractions and migrating from staking modules to a unified stVault-based model.

However, this change won't happen quickly. While the Staking Router manages stake allocation across modules, it does not handle intra-module stake delegation prioritization, reward distribution, or exit management for Node Operators. Transitioning fully to a stVault-based architecture would require major upgrades or extensions to these systems. One potential path forward is to design an adapter layer that conforms to the module interface and allows stVaults to integrate with the Staking Router, making them usable as a backend. It's better to first see how stVaults perform in mainnet conditions before moving fully to stVault-based architecture.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

4.3 Strengthening the Core Pool

Below is a list of protocol upgrades introduced in LIDO V3 that are specifically aimed at improving the efficiency, decentralization, and trust properties of the Core Pool.

4.3.1 MaxEB (Maximum Effective Balance) Support

Ethereum's Pectra hard fork, activated in May 2025, introduced EIP-7251, which raised the maximum effective balance (MaxEB) per validator from 32 ETH to 2048 ETH. This allows validators with withdrawal credentials 0x02 to consolidate their stake, reducing the total number of active validators. This change benefits the network by streamlining validator management and paving the way for faster block finalization. Lido's Staking Router is being upgraded to support these larger validators, aligning with Ethereum's roadmap and reducing the operational burden on node operators who currently manage thousands of validator instances.

4.3.2 Oracles and the zkOracle "second-opinion"

Lido's accounting layer hinges on an **oracle** that feeds two critical pieces of data into the protocol:

- **Validator balances** – the beacon-chain truth that drives stETH rebases.
- **Withdrawal-queue (WQ) state** – which requests can finalise, whether bunker-mode is on, how much ETH is ready and should be requested from validator exits, etc.

As long as the oracle behaves, stETH supply reflects reality and withdrawals flow. If it stalls or mis-reports, the protocol is blinded: withdrawals freeze and negative rebases could be mis-computed. V3 therefore treats the oracle subsystem as a *first-class trust-minimisation target*.

Tentatively structured upgrades (rolled out in stages):

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

- **zkOracle (second opinion)** – specified in LIP-23. *A zero-knowledge program digests a beacon-block root plus validator proofs and spits out a succinct proof that the balance delta is correct.* On-chain verification is cheap and deterministic while any mismatch — especially one that would trigger a negative rebase — auto-pauses stETH token rebases, giving the DAO time to react.
- **Multi-prover fallback** — the zkOracle contract will accept proofs from several independent prover clusters. If one cluster disappears, the others keep liveness—no committees required.
- **Emergency WQ push** — If oracles fail for a grace period, anyone can advance the Withdrawal Queue by submitting a beacon-root plus a minimal Merkle proof. It's slow and pricey, but exits never get stuck by being not finalized.
- **zk Exit-bus oracle** — Long term, a second circuit will prove that a validator's voluntary-exit message was signed and included (or wasn't and should be force-triggered from the EL side). This removes the bespoke "validator exit bus oracle" entirely.

Step by step, "trust the multisig" is replaced by "verify the proof", ensuring users can withdraw even under the worst oracle-failure scenario.

4.3.3 Algorithmic Stake Rebalancing

Today, the selection of node operators and their commission rates is done within separate votes by Lido DAO — a necessary limitation while the infrastructure for a more diverse validator set is still being built. In future versions of the protocol, this process is expected to transition to an **algorithmic solution**, abstracting optimization problem of balancing APR, risk, and network health into a repeatable, smart contract–based, on-chain system.

The vision here is the DAO providing the functional form, evaluating both the different options on the validator set and representing the dynamic structure of the system,

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

where transition between states may lead to additional transaction costs.

Therefore Lido DAO defines **strategy** for validator set, while execution is done by solving a dynamic optimization problem with a necessary tooling to collect and store on-chain data on performance metrics, fee structures, client diversity, geographic distribution, and participation history.

That would lead to a transparent structure on decision making and execution, where Node Operators could adjust their staking setup and fee structure based on the effect on expected stake distribution, defined by algorithm.

> As externalities exist (both in terms of risk concentration and systematic network effects), within the Lido Core distribution algorithm they could be internalised utilizing the function form itself, building corresponding alignment for incentives directly.
>
> For the stVaults stake distribution is defined by the market, hence the correct incentives through fee structure and risk mitigation levels would be built, indirectly affecting the overall protocol validator set.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

5 KNOWN RISKS

While LIDO V3 is designed for robust security and decentralization, certain risks are inherent to any staking protocol. This section outlines the main risks identified and the mitigation strategies implemented. It does not attempt to exhaustively capture all possible edge cases, but focuses on the principal areas of concern relevant to stETH holders and protocol sustainability.

5.1 Smart Contract Complexity and Systemic Risk

The modular architecture of LIDO V3 underpinning stVaults increases protocol complexity, broadens the attack surface, and introduces the potential for unexpected interactions between components — especially around protocol upgrades. This is particularly relevant following Ethereum network hard-forks, where previously stable assumptions or invariants may become fragile or invalid. **Mitigation:** Lido contributors follow a conservative development process: multiple independent audits by leading security firms (including Certora, Consensys Diligence, MixBytes, and others), fuzzing, invariant checks, and formal verification of critical code paths. Lido V3 has undergone over seven independent security audits and formal verification reviews prior to mainnet deployment. Public testnets are used to simulate real-world conditions and expose edge cases. The protocol has maintained a zero-incident record with no monetary impact on users since launch. The primary objective is to maintain core safety properties — most importantly, to keep stakers safe.

5.2 Governance Capture and Parameter Misconfiguration Risk

The Lido DAO governance system, while designed with checks and balances, carries inherent risks: misconfigured parameters (e.g., undercollateralized vaults or misaligned validator incentives) or malicious proposals could destabilize both individual stVaults, Core Pool and the broader stETH ecosystem. **Mitigation:** The presence of Dual Gov-

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

ernance, which gives stETH holders veto power over critical protocol-level decisions, reduces the risk of unilateral misconfigurations. stVaults can opt out of Lido DAO governance entirely via a permissionless "governance escape hatch" provided no stETH has been minted.

5.3 stETH Performance and Slashing Risks

Poor validator performance can cause stETH staking returns to fall below the Ethereum network average, reducing expected APR for holders. In severe cases — which have not historically occurred during Lido's operation but remain theoretically possible — such as extended validator downtime, slashing events, or coordinated failures, this underperformance may trigger a negative rebase, shrinking the total stETH supply and directly impacting holders. For example, if a large portion of validators (e.g., 80%) were to go offline or be slashed simultaneously — a scenario that has never historically occurred — the ETH backing stETH could drop sharply, potentially wiping out up to 100deposits. Beyond immediate financial losses, correlated validator failures can also amplify systemic risks across the Ethereum network. **Mitigation:** LIDO V3 is designed to minimize negative rebase risks through a multi-layered approach. First, the protocol prioritizes building a diverse validator set and distributing risk across different domains — such as clients, geographies, and operator types — to limit exposure to correlated failures. Node Operator performance is continuously monitored, with regular tracking of uptime, effectiveness, and client diversity to detect early warning signs. Additionally, dynamic per-vault bonding scales collateral requirements with stake size, ensuring that larger positions concentrated under a single Node Operator bear proportionally higher responsibility. As a final backstop, the Coverage Fund — whose use must be explicitly approved by DAO governance — can be deployed to absorb part of the losses and mitigate the impact on stETH holders in extreme scenarios.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

5.4 Liquidity, Withdrawals Delay and stVault Force Rebalancing Risks

Under extreme market conditions or synchronized redemption waves, market liquidity for stETH may dry up, leading to slippage or inability to swap stETH for ETH. At the same time, redemptions via the protocol's internal withdrawal queue may experience delays — due to validator exit bottlenecks, oracle issues, or sustained demand spikes. If Core Pool TVL drops too low, vault owners may be subject to force rebalancing, where their stake is reallocated to satisfy withdrawal requests. **Mitigation:** The protocol always guarantees 1:1 redemption via the withdrawal queue. This hard guarantee creates arbitrage opportunities, incentivizing liquidity providers to supply stETH-ETH liquidity on secondary markets and help restore supply. If the withdrawal queue slows down — due to validator exit delays, oracle disruptions, or surging demand — Lido includes fallback mechanisms such as triggerable exits to preserve validator withdrawal functionality. When Core Pool liquidity is insufficient, LIDO V3 can trigger force rebalancing: stake is withdrawn from stVaults and redeposited into the Core Pool to fulfill redemptions. Vault owners do not lose principal, but bear opportunity cost, as the leverage behind their minted stETH is being removed. But in the end of the day all withdrawals ultimately depend on Ethereum's validator exit queue. In periods of network-wide congestion, this can become the bottleneck, introducing delays regardless of internal protocol mechanisms.

5.5 stVault Node Operator Misbehavior Risk

Node Operator risk — i.e., the risk that a node operator harms the stVault's capital — arises when validator misbehavior leads to penalties or slashing. While operators cannot directly withdraw principal, they can indirectly cause a loss if their validators go offline or are slashed. In typical scenarios, slashing is limited to a small percentage of the validator's balance. However, in edge cases — such as extreme operator centraliza-

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

tion or a catastrophic event (e.g., a critical bug in a widely used client) — losses may, in theory, reach up to 100% of the affected stake. Such events have not occurred on Ethereum mainnet but remain a non-zero tail risk. Additionally, an operator may attempt to extract execution layer rewards or hide MEV from stakers. **Mitigation**: LIDO V3 is architected to disincentivize stake centralization through protocol-level controls, market-based stake routing, and operator self-bonding mechanics in stVaults. The node operator marketplace surfaces reputation and performance data for curated operators, providing stakers with the context needed to assess and price operator risk effectively.

5.6 Oracle Manipulation Risks

LIDO V3 oracles are responsible for bringing consensus layer data on-chain and coordinating key protocol actions. If oracle data is delayed, unavailable, or incorrect, it can result in Lido protocol entering invalid states, including paused withdrawals or disruption of protocol operations. Even though oracles are operated by independent entities, there remains risk of coordinated misbehavior or software compromise. **Mitigation:** Lido protocol employs a quorum-based relay architecture, aggregating inputs from multiple sources to reduce reliance on any single actor. Built-in safeguards include deviation checks and update rate limits to detect anomalies and reject faulty data. The system is designed to remain functional as long as a threshold of oracles remains honest. Additional mitigation includes hardened oracle implementations, including requirements to prove anomalies with zero-knowledge proofs for some cases.

5.7 Hidden Stake Share (Sybil and Obfuscation Risk)

The validator risk framework relies on accurate input data — such as Node Operator independence, infrastructure characteristics, and validator share within the network. If this data is incorrect or incomplete (due to Sybil attacks, misreporting, or obfuscation), the framework may misattribute risk levels, leading to suboptimal stake distribution and hidden concentration risks. **Mitigation:** LIDO V3 incorporates transparency require-

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

ments for curated operators, planning metadata aggregation in future through systems like NodeOperatorAtlas, and dynamic risk scaling (e.g., bonding and stake caps). However, some level of attribution error remains inevitable in open systems, and continuous monitoring and framework updates are critical to minimize its impact over time.

6 REFERENCES

All references in this document are provided as inline hyperlinks to their respective sources.

7 ACKNOWLEDGEMENTS

We thank the Lido community for their valuable ideas, feedback, discussions, support, and help with writing, editing, and preparing the LIDO V3 whitepaper for publication. Special thanks to Isidoros Passadis, Sam Kozin, Sacha Saint-Leger, Victor Suzdalev, Mikhail Gurevich, Hasu, Stanislav Buynovskiy, Katya Pavlenko, Daria Kiseleva, Sam Morozov, Armin Satzger, Dmitrii Vulbrun, Azat Serikov, Eugene Emelyanov, Mariya Muzyko, and Tomer Ganor.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

8 DISCLAIMER

This document is provided “as is”, without warranties of any kind, express or implied, for informational purposes only. It does not constitute accounting, legal, or tax advice, nor does it contain investment recommendations. Nothing in this document should be construed as a commitment, promise, or the basis for any binding obligation.

The views expressed are those of individual contributors to the Lido protocol and do not necessarily reflect the outcomes of Lido DAO governance or the views of affiliated organizations. The opinions reflected herein are subject to change without notice.

This whitepaper describes conceptual elements of the protocol and should not be interpreted as a definitive technical specification. Always refer to the Lido documentation, Lido Improvement Proposals (LIPs), and the source code of deployed contracts for the most up-to-date information.

This PDF was auto-generated from lidofinance/whitepaper, commit ‘95b7d82‘.

A USER FLOWS FOR STVAULTS

This appendix outlines the main user flows for stVaults, highlighting their key interactions and implications. Unlike traditional liquid staking, where stETH is minted in exchange for ETH at a 1:1 ratio, stVaults mint stETH against staked ETH held within the stVault, considering the Reserve Ratio. This position is overcollateralized, with only a portion of it available for stETH minting, while the remaining part is held as a Reserve.

Disclaimer: we intentionally leave out of context both the creation of the stVault itself and the assignment of the Reserve Ratio, which defines the Total stETH Minting Capacity. The Reserve Ratio is determined by the parameters of the Node Operator's tier selected during stVault creation. We also leave the details of proving validators through the Predeposit Guarantee mechanism out of context, assuming the validators are already proven in these described scenarios.

The core mechanics of stVaults can be understood through three primary scenarios:

1. **Opening position and minting stETH** – ETH is deposited into an stVault, staked, and used as collateral to mint stETH, unlocking liquidity for staked assets.

2. **Closing a staking position** – The minted stETH must be repaid before the stVault can be closed, allowing the withdrawal of the staked ETH and rewards.

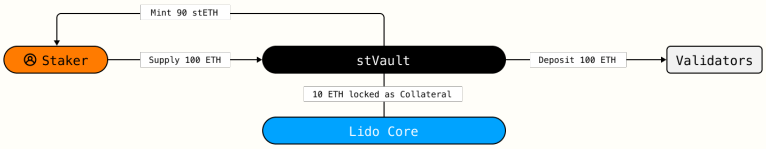
3. **Voluntary and forced measures to restore the stVault's Health Factor in case of Node Operator underperformance or slashing penalties** - Discrepancies between stETH performance and the underlying staked position may necessitate intervention. If a validator underperforms or incurs slashing penalties, the Vault Owner can take proactive measures to restore the balance between Total Value and stETH Liability. Additionally, forced rebalancing mechanisms can be triggered to adjust vault positions, mitigating risks and preserving the protocol's financial stability.

Each of these user flows is examined in detail in the following sections.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

A.1 Opening position and minting stETH

Opening a position within an stVault involves a series of steps aimed at coordinating the staker and the node operator. Once the staking position is established, the staker can mint stETH against it as collateral. The process unfolds as follows:



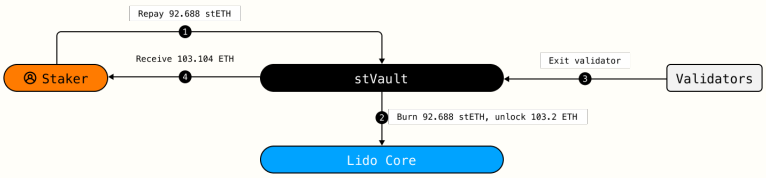
1. Supplying ETH:
    - The staker supplies 100 ETH to stVault.
    - The stVault now holds the ETH and is ready for delegation to a node operator.
2. Depositing ETH to validators:
    - The node operator determines which validator will receive the 100 ETH delegation (utilizing the target validator pubkey) and tops up the validator through the Predeposit Guarantee contract. The validator should have been proven by the PreDeposit Guarantee mechanism to maintain the non-custodial nature of the setup.
3. Minting stETH and locking collateral:
    - Minting stETH Capacity = 90 stETH (in this example, we assume a 10
    - The vault owner mints 90 stETH, locking 100 ETH as collateral, consisting of 90 stETH Liability and a 10 ETH Reserve.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

- The minted stETH provides the staker with on-demand liquidity while maintaining a buffer in the stVault for risk management, while also offering an opportunity for subsequent use of stETH in DeFi to add an additional layer of yield.

A.2 Closing a staking position

Closing a staking position in an stVault involves repaying the stETH liability, unlocking the staked ETH, and withdrawing the final balance, which includes staking rewards. This process ensures that the vault remains fully collateralized while allowing the staker to exit their position. The steps are as follows:



1. Repayment of Minted stETH
    - After one year, the staker is required to repay the originally minted stETH considering that stETH is a rebasing token.
    - Assuming a stETH APR of 2.8totaling 92.52 stETH.
    - The staker repays 92.52 stETH to the stVault.
2. Burning stETH and Unlocking ETH
    - Once the 92.52 stETH repayment is received, the stVault burns the stETH and unlocks the corresponding amount of ETH considering the Reserve Ratio.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

- Accumulated not yet disbursed node operator fee and Lido fees remain locked from withdrawals by the staker to maintain node operator and Lido protocol financial security.

3. Exiting the Validator

   - The stVault initiates the validator exit process, retrieving the staked ETH plus any accumulated staking rewards.
   - The exit is processed through Ethereum's withdrawal queue, ensuring that the validator's funds are returned in compliance with protocol mechanisms.

4. Withdrawing ETH from the stVault

   - Once the exit is finalized, the staker can withdraw their unlocked ETH from the stVault.
   - Assuming stVault validation APR of 3.20.1728 ETH of Lido liquidity fee calculated based on amount of stETH Liability, the available for withdrawals is total of 102.9312 ETH, which includes both the original 100 ETH deposit and 2.9312 net staking rewards.

This structured process ensures that the stVault maintains stability while allowing stakers to exit seamlessly, reclaim their assets, and access their staking rewards.

A.3 Voluntary and forced measures to restore the stVault's Health Factor in case of Node Operator underperformance or slashing penalties

Discrepancies between stETH performance and the underlying staked position may necessitate intervention. If a validator underperforms or incurs slashing penalties, the Vault Owner can take proactive measures to restore the balance between Total Value and stETH Liability, thereby maintaining the position overcollateralized in accordance

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

with the Reserve Ratio. Additionally, forced rebalancing mechanisms can be triggered to adjust vault position (when Force Rebalance Threshold breached), mitigating risks and preserving the protocol's financial stability.

$$HealthFactor = \frac{TotalValue * (1 - ForcedRebalanceThreshold)}{stETHLiability}$$

A.3.1 Voluntary measures to restore health factor

The Vault Owner can initiate voluntary actions to improve the Health Factor when it drops below 100options include:

1. Supplying additional ETH to the vault:
    - The Vault Owner can deposit more ETH into the stVault, increasing the Total Value and thereby improving the Health Factor.
    - This method strengthens the collateralization of the stETH Liability.
2. Repaying stETH liability:
    - The Vault Owner can repay a portion of the stETH Liability to reduce the ratio of Liability to Total Value, enhancing the Health Factor.
3. Rebalancing part of ETH from the vault:
    - The Vault Owner can rebalance by moving a portion of ETH from the stVault to cover the undercollateralization caused by validator penalties or underperformance.
    - This approach directly addresses the discrepancy between stETH and the underlying ETH value.
    - The ability for the Vault Owner to specify the amount of ETH to be rebalanced provides flexibility.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

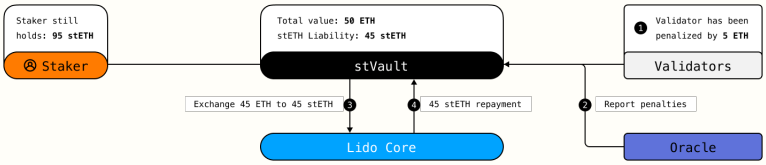
Rebalancing can also be combined with Supplying additional ETH, providing a customized method to restore Health Factor.

Rebalancing mechanism explained

During the rebalancing process, a portion of ETH from the user's vault is sent to the Lido Core protocol. In exchange, stETH is minted at a 1:1 ratio and returned to the vault as repaid stETH. This repayment reduces the user's stETH Liability to the vault.

The original stETH minted by the user remains in their possession. As a result, part of the stETH held by the user is now collateralized by the ETH in the vault considering the Reserve Ratio, while the other part is collateralized by ETH in the Lido Core protocol at a 1:1 ratio.

Example



1. Initial state:
    - Total Value = 100 ETH
    - stETH Liability = 90 stETH
    - Reserve Ratio = 10
    - Health Factor = 100
2. Slashing penalty:

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

- A validator incurs a penalty of 5 ETH.
- New Total Value = 95 ETH.
- Problem: The amount of stETH held by the Vault Owner is now higher than the Total stETH minting capacity of the stVault, considering the reduced Total Value.

3. Rebalancing:
   - The Vault Owner sends 45 ETH from the stVault to Lido Core protocol.
   - In exchange, 45 stETH is minted at a 1:1 ratio and returned to the vault as repaid stETH.
   - This action effectively pays back the stETH Liability of 45 stETH.
4. Resulting metrics:
   - Total Value: 50 ETH (remaining in the stVault)
   - stETH Liability: 45 stETH (original liability remaining)
   - Backed Collateral: 50 ETH (in stVault)
   - Health Factor: 100
5. Vault owner's assets = 90 stETH:
   - 45 stETH is backed by 45 ETH in Lido Core (1:1 ratio).
   - 45 stETH is backed by 50 ETH in the stVault.

This mechanism ensures that the stVault remains sufficiently collateralized even when penalties occur, preserving the protocol's safety and stability.

A.3.2 Forced rebalancing (permissionless)

When the **Forced Rebalance Threshold** is reached, a permissionless rebalancing mode is activated where anyone can initiate the rebalancing process. Unlike voluntary rebalancing where the Vault Owner specifies the amount of ETH to be rebalanced,

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

forced rebalancing uses the entire required amount of ETH to restore the Health Factor, ensuring that the position remains overcollateralized in line with the Reserve Ratio (i.e., Health Factor becomes strictly above 100

This permissionless mechanism allows participants from the market or Lido DAO itself to trigger rebalancing. Automatic mechanisms will be implemented by the DAO or third-party participants to ensure that undercollateralized positions are corrected swiftly and efficiently.

If the necessary ETH for rebalancing is deposited on validators, the protocol will support **Forced withdrawal mechanisms** to bring the ETH back to the vault. This process ensures sufficient funds are available for rebalancing and protects the integrity of the stVault.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

B RISK ASSESSMENT FRAMEWORK

B.1 Purpose and intended application

This Risk Assessment Framework outlines a quantitative approach to:

- Evaluating the effect on risk events, associated with running validators
- Defining risk mitigation strategies accounting for all relevant actors within existing and future Staking Modules and stVaults.
- Unified structure of managing risks, associated with validation. Supporting DAO to make informed and transparent decision on possible risk mitigation designs and level of risk acceptance

The application of framework is intended to be in the process of:

- Identifying a range of risk scenarios, stated in terms of risk events (e.g. slashing for 100% of validators for a particular Node Operator via malicious attack or operational misconfiguration, validators running a specific client going offline due to bug in the software)
- Identifying range of possible external network conditions (e.g. level of CL & EL rewards, or any ongoing slashing for validators out-of-protocol)
- Evaluating the effect of possible mitigation strategies with risk transferring (via bonds, collateral or cover fund provided as mitigation) or risk reduction (via decentralization of Node operators, software or validator setup characteristics)
- DAO makes the decisions on sufficient design both in terms of risk reduction & transferring, taking into account the effect on chosen design on actors in the system.

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

This approach ensures that risk mitigation remains both sufficient and aligned with Lido's principles, while clearly scoping out-of-scenario risks that, though acknowledged, are not directly addressed by the framework.

B.2 Key risk areas and their impact

We consider the following primary risks:

- Validator liveness
- Validator slashings

Each risk is assessed in terms of its potential impact on the protocol, specifically:

- Losses – actual reduction in the total ETH balance (e.g., due to slashing penalties)
- Missed rewards – reduction in expected or achievable staking rewards (e.g., due to offline validators)

B.3 Framework application principles

- **Spec-driven evaluation** - evaluation of risk scenarios is based on current consensus specifications, taking into account future expected upgrades, while they're disclosed in form of EIPs
- **Conservative approach for external network conditions** - risk scenarios includes the possible changes in network conditions that lead to increase in risk effects (e.g. reduction in total stake leading to increase in CL rewards and penalties level)
- **Zero risk tolerance for stakers** - risk scenarios operate in a field of catastrophic events never observed historically, to ensure that mitigation designs are sufficient for the most dramatic events

Generated from repository lidofinance / whitepaper from a commit: 95b7d82

- **Risk attributing** - the proposed risk transfer designs are intended to allocate corresponding risk associated for running validators to actors controlling the source of risk.

B.4 Reasonable risk taking

Utilizing the framework based on principles above, DAO decided on strategy on risk mitigation via reduction, providing incentivization for decentralization, while also creating a multiple level of risk transferring, where, based on scenarios formulated the consequences of the risk event are attributed to:

- Individual actors (e.g. Node Operators, stVaults users)
- Cover fund(s)
- stETH users With the intention that last option could happen only in the scenarios that couldn't be reasonably mitigated, as they are within the field of catastrophic-level network events (e.g. slashing of one of the majority clients)

Particular risk framework application examples are publicly disclosed on protocol research forum:

- Risk assessment for community staking
- Risk Assessment Framework for stVaults

Generated from repository lidofinance / whitepaper from a commit: 95b7d82