Lido tokens integration guide

This document is intended for developers looking to integrate Lido's stETH or wstETH as a token into their dApp, with a focus on money markets, DEXes and blockchain bridges.

Lido

Lido is a family of liquid staking protocols across multiple blockchains, with headquarters on Ethereum. Liquid refers to the ability of a user's stake to become liquid. Upon the user's deposit Lido issues stToken, which represents the deposited tokens along with all the rewards & penalties accrued through the deposit's staking. Unlike the staked funds, this stToken is liquid — it can be freely transferred between parties, making it usable across different DeFi applications while still receiving daily staked rewards. It is paramount to preserve this property when integrating stTokens into any DeFi protocol.

This guide refers to Lido on Ethereum (hereinafter referred to as Lido).

Lido tokens

stTokens: stETH and wstETH

For ether staked in Lido, the Lido protocol gives users stETH that is equal to the amount staked. For easier DeFi integrations,stETH has a non-rebasable value-accruing counterpart called 'wrapped stETH' (or just wstETH).

Lido's ERC-20 compatible stTokens are widely adopted across the Ethereum ecosystem:

- The most important on-chain liquidity venues include:* stETH/ETH liquidity pool on Curve
- wstETH/ETH Composable stable pool on Balancer v2
- wstETH/ETH pool on Uniswap V3
- wstETH is listed as a collateral token on the following AAVE v3 markets:* Ethereum mainnet
- Arbitrum
- Optimism
- Polygon
- stETH is listed as a collateral token on the AAVE v2 Ethereum mainnet
- market
- wstETH is listed as a collateral token on Maker
- steCRV (the Curve stETH/ETH LP token) is listed as a collateral token on Maker
- there are multiple liquidity strategies built on top of Lido's stTokens, including Yearn
- and Harvest Finance

Integration utilities: ChainLink price feeds

- There are live ChainLink stETH/USD
- and stETH/ETH
- price feeds on Ethereum.
- There are live ChainLink stETH/USD price feeds on Arbitrum
- and Optimism
- . These also have ChainLink wstETH-stETH exchange rate data feeds.

LDO

LDO is a Lido governance ERC-20 compliant token derived from the MiniMe Token . Thus, LDO holder balances are queryable for an arbitrary block number, an essential security feature for the Lido voting mechanics.

unstETH

A non-fungible token (NFT) is used to represent a withdrawal request position in the protocol-level withdrawals queue when a stToken holder decides to redeem it for ether via the protocol.

note Unlike the other Lido's tokens (stETH ,wstETH , and LDO), unstETH is non-fungible, and implements the ERC-721 token standard.

stETH vs. wstETH

There are two versions of Lido's stTokens, namely stETH and wstETH. Both are fungible tokens, but they reflect the accrued staking rewards in different ways. stETH implements rebasing mechanics which means the stETH balance increases periodically. On the contrary, wstETH balance is constant, while the token increases in value eventually (denominated in stETH). At any moment, any amount of stETH can be converted to wstETH via a trustless wrapper and vice versa, thus tokens effectively share liquidity. For instance, undercollateralized wstETH positions on Maker can be liquidated by unwrapping wstETH and swapping it for ether on Curve.

stETH

What is stETH

stETH is a rebaseable ERC-20 token that represents ether staked with Lido. Unlike staked ether, it is liquid and can be transferred, traded, or used in DeFi applications. The total supply of stETH reflects the amount of ether deposited into protocol combined with staking rewards, minus potential validator penalties. stETH tokens are minted upon ether deposit at 1:1 ratio. Since withdrawals from the Beacon chain have been introduced, it is also possible to redeem ether by burning stETH at the same 1:1 ratio (in rare cases it won't preserve 1:1 ratio though).

Please note, Lido has implemented staking rate limits aimed at reducing the post-Merge staking surge's impact on the staking queue & Lido’s socialized rewards distribution model. Read more about ithere .

stETH is a rebasable ERC-20 token. Normally, the stETH token balances get recalculated daily when the Lido oracle reports the Beacon Chain ether balance update. The stETH balance update happens automatically on all the addresses holding stETH at the moment of rebase. The rebase mechanics have been implemented via shares (seeshares).

Note on ERC-20 compliance

stETH does not strictly comply with ERC-20. The only exception is that it does not emitTransfer() on rebase asERC-20 standard requires.

Accounting oracle

Normally, stETH rebases happen daily when the Lido oracle reports the Beacon chain ether balance update. The rebase can be positive or negative, depending on the validators' performance. In case Lido's validators get slashed or penalized, the stETH balances can decrease according to penalty sizes. However, daily rebases have never been negative by the time of writing. The accounting oracle has sanity checks on both max APR reported (the APR cannot exceed 27%, which means a daily rebase is limited to27/365%) and total staked amount drop (staked ether decrease reported cannot exceed 5%). Currently, the oracle report is based on five oracle daemons hosted by established node operators selected by the DAO. As soon as five out of nine oracle daemons report the same data, reaching the consensus, the report goes to the Lido smart contract, and the rebase occurs. There is adedicated oracle dashboard to monitor current accounting reports.

Oracle corner cases

- In case oracle daemons do not report Beacon chain balance update or do not reach quorum, the oracle does not submit the daily report, and the daily rebase doesn't occur until the quorum is reached.
- Oracle report might be delayed, but it will include values actual for the reporting refSlot. So, even if reported 2 hours late, it will include only rebase values for the original period.
- In case the quorum hasn't been reached, the oracle can skip the daily report. The report will happen as soon as the quorum for one of the next periods will be reached, and it will include the incremental balance update for all periods since the last successful oracle report.
- Oracle daemons only report the finalized epochs. In case of no finality on the Beacon chain, the daemons won't submit their reports, and the daily rebase won't occur.
- In case sanity checks on max APR or total staked amount drop fail, the oracle report cannot be finalized, and the rebase cannot happen.

stETH internals: share mechanics

Daily rebases result in stETH token balances changing. This mechanism is implemented via shares. Theshare is a basic unit representing the stETH holder's share in the total amount of ether controlled by the protocol. When a new deposit happens, the new shares get minted to reflect what share of the protocol-controlled ether has been added to the pool. When the Beacon chain oracle report comes in, the price of 1 share in stETH is being recalculated. Shares aren't normalized, so the contract also stores the sum of all shares to calculate each account's token balance. Shares balance by stETH balance can be calculated by this formula:

shares [account]

=

balanceOf (account)

* totalShares / totalPooledEther

1-2 wei corner case

stETH balance calculation includes integer division, and there is a common case when the whole stETH balance can't be transferred from the account while leaving the last 1-2 wei on the sender's account. The same thing can actually happen at any transfer or deposit transaction. In the future, when the stETH/share rate will be greater, the error can become a bit bigger. To avoid it, one can usetransferShares to be precise.

Example:

1. User A transfers 1 stETH to User B.
2. Under the hood, stETH balance gets converted to shares, integer division happens and rounding down applies.
3. The corresponding amount of shares gets transferred from User A to User B.
4. Shares balance gets converted to stETH balance for User B.
5. In many cases, the actually transferred amount is 1-2 wei less than expected.

The issue is documented here:lido-dao/issues/442

Bookkeeping shares

Although user-friendly, stETH rebases add a whole level of complexity to integrating stETH into other dApps and protocols. When integrating stETH as a token into any dApp, it's highly recommended to store and operate shares rather than stETH public balances directly, because stETH balances change both upon transfers, mints/burns, and rebases, while shares balances can only change upon transfers and mints/burns.

To figure out the shares balance,getSharesByPooledEth(uint256) function can be used. It returns the value not affected by future rebases and it can be converted back into stETH by callinggetPooledEthByShares function.

See all available stETH methodshere . Any operation on stETH can be performed on shares directly, with no difference between share and stETH.

The preferred way of operating stETH should be:

1. get stETH token balance;
2. convert stETH balance into shares balance and use it as a primary balance unit in your dApp;
3. when any operation on the balance should be done, do it on the shares balance;
4. when users interact with stETH, convert the shares balance back to stETH token balance.

Please note that 10% APR on shares balance and 10% APR on stETH token balance will ultimately result in different output values over time, because shares balance is stable, while stETH token balance changes eventually.

If using the rebasable stETH token is not an option for your integration, it is recommended to use wstETH instead of stETH. See how it workshere .

Transfer shares function for stETH

TheLIP-11 introduced thetransferShares function which allows to transfer stETH in a "rebase-agnostic" manner: transfer in terms ofshares amount.

Normally, we transfer stETH using ERC-20transfer andtransferFrom functions which accept as an input the amount of stETH, not the amount of the underlying shares. Sometimes we'd better operate with shares directly to avoid possible rounding issues. Rounding issues usually could appear after a token rebase. This feature is aimed to provide an additional level of precision when operating with stETH. Read more about the function in theLIP-11 .

Also, V2 upgrade introduced atransferSharesFrom to completely match ERC-20 set of transfer methods.

Fees

Lido collects a percentage of the staking rewards as a protocol fee. The exact fee size is defined by the DAO and can be changed in the future via DAO voting. To collect the fee, the protocol mints new stETH token shares and assigns them to the fee recipients. Currently, the fee collected by Lido protocol is 10% of staking rewards with half of it going to the node operators and the other half going to the protocol treasury.

Since the total amount of Lido pooled ether tends to increase, the combined value of all holders' shares denominated in stETH increases respectively. Thus, the rewards effectively spread between each token holder proportionally to their share in the protocol TVL. So Lido mints new shares to the fee recipient so that the total cost of the newly-minted shares exactly corresponds to the fee taken (calculated in basis points):

shares2mint * newShareCost = (_totalRewards * feeBasis) / 10000 newShareCost = newTotalPooledEther /

(prevTotalShares + shares2mint) which follows:

$$shares2mint = \frac{_totalRewards * feeBasis * prevTotalShares}{(newTotalPooledEther * 10000) - (feeBasis * _totalRewards)}$$

How to get APR?

Please refer tothis page for the correct Lido V2 APR calculation.

It is worth noting that with withdrawals enabled, the APR calculation method for Lido has changed significantly. When Lido V2 protocol finalizes withdrawal requests, the Lido contract excludes funds from TVL and assigns to burn underlying locked requests' stETH shares in return. In other words, withdrawal finalization decreases both TVL and total shares. The old V1 formula isn't suitable anymore because it catches TVL changes, but skips total shares changes.

Do stETH rewards compound?

Yes, stETH rewards do compound.

All rewards that are withdrawn from the Beacon chain or received as MEV or EL priority fees (that aren't used to fulfill withdrawal requests) are finally restaked to set up new validators and receive more rewards at the end. So, we can say that stETH becomes fully auto-compounding after V2 release.

wstETH

Due to the rebasing nature of stETH, the stETH balance on the holder's address is not constant, it changes daily as oracle report comes in. Although rebasable tokens are becoming a common thing in DeFi recently, many dApps do not support rebasing. For example, Maker, UniSwap, and SushiSwap are not designed for rebasable tokens. Listing stETH on these apps can result in holders not receiving their daily staking rewards which effectively defeats the benefits of liquid staking. To integrate with such dApps, there's another form of Lido stTokens called wstETH (wrapped staked ether).

What is wstETH

wstETH is an ERC20 token that represents the account's share of the stETH total supply (stETH token wrapper with static balances). For wstETH, 1 wei inshares equals to 1 wei in balance. The wstETH balance can only be changed upon transfers, minting, and burning. wstETH balance does not rebase, wstETH's price denominated in stETH changes instead. At any given time, anyone holding wstETH can convert any amount of it to stETH at a fixed rate, and vice versa. The rate is the same for everyone at any given moment. Normally, the rate gets updated once a day, when stETH undergoes a rebase. The current rate can be obtained by callingwstETH.stEthPerToken()

Wrap & Unwrap

When wrapping stETH to wstETH, the desired amount of stETH is locked on the WstETH contract balance, and the wstETH is minted according to theshare bookkeeping formula.

When unwrapping, wstETH gets burnt and the corresponding amount of stETH gets unlocked.

Thus, the amount of stETH unlocked when unwrapping is different from what has been initially wrapped (given a rebase happened between wrapping and unwrapping stETH).

wstETH shortcut

Note, that the WstETH contract includes a shortcut to convert ether to wstETH under the hood, which allows you to effectively skip the wrapping step and stake ether for wstETH directly. Keep in mind that when using the shortcut,the staking rate limits still apply.

Rewards accounting

Since wstETH represents the holder's share in the total amount of Lido-controlled ether, rebases don't affect wstETH balances but change the wstETH price denominated in stETH.

Basic example :

1. User wraps 1 stETH and gets 0.9803 wstETH (1 stETH = 0.9803 wstETH)
2. A rebase happens, the wstETH price goes up by 5%
3. User unwraps 0.9803 wstETH and gets 1.0499 stETH (1 stETH = 0.9337 wstETH)

Goerli wstETH for testing

The most recent testnet version of the Lido protocol lives on the Goerli testnet (see the full list of contracts deployed here). Just like on mainnet, Goerli wstETH for testing purposes can be obtained by approving the desired amount of stETH to the WstETH contract on Goerli, and then callingwrap method on it. The corresponding amount of Goerli stETH will be locked on the WstETH contract, and the wstETH tokens will be minted to your account. Goerli ether can also be converted to wstETH directly using thewstETH shortcut – just send your Goerli ether to WstETH contract on Goerli, and the corresponding amount of wstETH will be minted to your account.

wstETH on L2s

Currently, wstETH token is present on Arbitrum, Optimism, and Polygon PoS with bridging implemented via the canonical bridges.

note Unlike on the Ethereum mainnet, wstETH on L2s is a plain ERC-20 token and cannot be unwrapped to unlock stETH on the corresponding L2 network. Without the shares bookkeeping, the token cannot provide thewstETH/stETH rate and the rewards accrued on-chain. However, there're live Chainlink wstETH/stETH rate feeds forArbitrum andOptimism that can and should be used for this purpose.

LDO

What is LDO

LDO is a governance token used for the Lido DAO's voting process (both off-chain and on-chain). The token is widely available in DeFi and CeFi ecosystems.

LDO has internal mechanics of the balance snapshots (balanceOfAt andtotalSupplyAt) to allow voting power not being manipulated within the time of the ongoing vote.

Note on ERC-20 compliance

Although the LDO is fully compliant with ERC-20, it is worth noting that the token doesn't revert a transaction on all of the failure paths inside boththransfer andtransferFrom methods returning thefalse status instead.

note It's critical to check the return status for external integrations as the ERC-20 token standardrequires to prevent various attack vectors (e.g. token deposits in vaults):

Callers MUST handlefalse fromreturns (bool success) . Callers MUST NOT assume thatfalse is never returned!

ERC20Permit

wstETH and stETH tokens implement the ERC20 Permit extension allowing approvals to be made via signatures, as defined inEIP-2612 .

Thepermit method allows users to modify the allowance using a signed message, instead of throughmsg.sender . By not relying onapprove method, you can build interfaces that will approve and use wstETH in one tx.

Staking rate limits

In order to handle the staking surge in case of some unforeseen market conditions, the Lido protocol implemented staking rate limits aimed at reducing the surge's impact on the staking queue & Lido's socialized rewards distribution model. There is a sliding window limit that is parametrized with_maxStakingLimit and_stakeLimitIncreasePerBlock . This means it is only possible to submit this much ether to the Lido staking contracts within a 24-hours timeframe. Currently, the daily staking limit is set at 150,000 ether.

You can picture this as a health globe from Diablo 2 with a maximum of_maxStakingLimit and regenerating with a constant speed per block. When you deposit ether to the protocol, the level of health is reduced by its amount and the current limit becomes smaller and smaller. When it hits the ground, the transaction gets reverted.

To avoid that, you should check ifgetCurrentStakeLimit() >= amountToStake , and if it's not you can go with an alternative route. The staking rate limits are denominated in ether, thus, it makes no difference if the stake is being deposited for stETH or usingthe wstETH shortcut , the limits apply in both cases.

Alternative routes

1. Wait for staking limits to regenerate to higher values and retry depositing ether to Lido later.
2. Consider swapping ETH for stETH on DEXes like Curve or Balancer. At specific market conditions, stETH may effectively be purchased from there with a discount due to stETH price fluctuations.

Withdrawals (unstETH)

V2 introduced the possibility to withdraw ETH from Lido. A high-level upgrade overview can be found in the blog post . Withdrawals flow is organized as a FIFO queue that accepts the requests with stETH attached and these requests are finalized with oracle reports as soon as ether to fulfill the request is available.

So to obtain ether from the protocol, you'll need to proceed with the following steps:

- request the withdrawal, locking your steth in the queue and receiving an NFT, that represents your position in the queue
- wait, until the request is finalized by the oracle report and becomes claimable
- claim your ether, burning the NFT

Request size should be at least100 wei (in stETH) and at most1000 stETH . Larger amounts should be withdrawn in multiple requests, which can be batched via in-protocol API. Once requested, withdrawal cannot be canceled. The withdrawal NFT can be transferred to a different address, and the new owner will be able to claim the requested withdrawal once finalized.

The amount of claimable ETH is determined once the withdrawal request is finalized. The rate stETH/ETH of the request finalization can't get higher than it's been at the moment of request creation. The user will be able to claim:

- normally – the ETH amount corresponding to the stETH amount at the moment of the request's placement

OR

- discounted - lowered ETH amount corresponding to the oracle-reported share rate in case the protocol had undergone significant losses (slashings and penalties)

The second option is unlikely, and we haven't ever seen the conditions for it on mainnet so far.

The end-user contract to deal with the withdrawals isWithdrawalQueueERC721.sol , which implements the ERC721 standard. NFT represents the position in the withdrawal queue and may be claimed after the finalization of the request.

Let's follow these steps in detail:

Request withdrawal and mint NFT

You have several options for requesting withdrawals, they require you to have stETH or wstETH on your address:

stETH

- CallrequestWithdrawalsWithPermit(uint256[] _amounts, address _owner, PermitInput _permit)
- and get the ids of created positions, wheremsg.sender
- will be used to transfer tokens from and the_owner
- will be the address that can claim or transfer NFT (defaults tomsg.sender
- if it’s not provided)
- Alternatively, sending stETH on behalf ofWithdrawalQueueERC721.sol
- contract can be approved in a separate upfront transaction (stETH.approve(withdrawalQueueERC721.address, allowance)
-), and therequestWithdrawals(uint256[] _amounts, address _owner)
- method called afterwards

wstETH

- CallrequestWithdrawalsWstETHWithPermit(uint256[] _amounts, address _owner, PermitInput _permit)
- and get the ids of created positions, wheremsg.sender
- will be used to transfer tokens from, and the_owner
- will be the address that can claim or transfer NFT (defaults tomsg.sender
- if it’s not provide)
- Alternatively, sending wstETH on behalf ofWithdrawalQueueERC721.sol
- contract can be approved in a separate upfront transaction (wstETH.approve(withdrawalQueueERC721.address, allowance)
-), and therequestWithdrawalsWstETH(uint256[] _amounts, address _owner)
- method called afterwards

PermitInput structure is defined as follows:

struct

PermitInput

{ uint256 value ; uint256 deadline ; uint8 v ; bytes32 r ; bytes32 s ; } After requestERC721 NFT is minted to_owner address and can be transferred to the other owner who will have all the rights to claim the withdrawal.

Additionally, this NFT implements theERC4906 standard and it's recommended to rely on

event

BatchMetadataUpdate (uint256 _fromTokenId ,

uint256 _toTokenId) ; to update the NFT metadata if you're integrating it somewhere where it should be displayed correctly.

note Withdrawal transactions made withrequestWithdrawalsWithPermit orrequestWithdrawalsWstETHWithPermit might fail due to being front-run by stealing the user-provided signature to executetoken.permit method. It does not impose any fund loss risks nor blocks the capability to withdraw, but it affects the UX. For the details, seethis issue .

It's recommended to mitigate the issue, e.g. by utilizing the approach used inLido staking widget . Shortly, the idea is as follows. If the initial...WithPermit transaction fails, immediately resent the request but viarequestWithdrawals/requestWithdrawalsWstETH method this time, seamlessly relying on the allowance already provided as a result of the griefing transaction. For the specific example, seethe following code .

Any other viable approach for mitigation might be used as well. As one more example, deploy a wrapper smart contract that triesrequestWithdrawalsWithPermit/requestWithdrawalsWithPermitWstETH and ifcatches the revert error, continues withrequestWithdrawals/requestWithdrawalsWstETH , checking the allowance is enough.

Checking the state of withdrawal

- You can check all the withdrawal requests for the owner by callinggetWithdrawalRequests(address _owner)
- which returns an array of NFT ids.
- To check the state of the particular NFTs you can callgetWithdrawalStatus(uint256[] _requestIds)
- which returns an array ofWithdrawalRequestStatus
- struct.

struct

WithdrawalRequestStatus

{ /// @notice stETH token amount that was locked on withdrawal queue for this request uint256 amountOfStETH ; /// @notice amount of stETH shares locked on withdrawal queue for this request uint256 amountOfShares ; /// @notice address that can claim or transfer this request address owner ; /// @notice timestamp of when the request was created, in seconds uint256 timestamp ; /// @notice true, if request is finalized bool isFinalized ; /// @notice true, if request is claimed. Request is claimable if (isFinalized && !isClaimed) bool isClaimed ; } NOTE: Since stETH is an essential token if the user requests a withdrawal using wstETH directly, the amount will be nominated in stETH on request creation. You can callgetClaimableEther(uint256[] _requestIds, uint256[] _hints) to get the exact amount of eth that is reserved for the requests, where_hints can be found by callingfindCheckpointHints(__requestIds, 1, getLastCheckpointIndex()) . It will return a non-zero value only if the request is claimable (isFinalized && !isClaimed)

Claiming

To claim ether you need to call:

- claimWithdrawal(uint256 _requestId)
- with the NFT Id on behalf of the NFT owner
- claimWithdrawals(uint256[] _requestIDs, uint256[] _hints)
- if you want to claim multiple withdrawals in batches or optimize on hint search* hints =findCheckpointHints(uint256[] calldata _requestIDs, 1, lastCheckpoint)
-
    - lastCheckpoint =getLastCheckpointIndex()

General integration examples

stETH/wstETH as collateral

stETH/wstETH as DeFi collateral is beneficial for several reasons:

- stETH/wstETH is almost as safe as ether, price-wise: barring catastrophic scenarios, its value tends to hold the ETH peg well;
- stETH/wstETH is a productive token: getting rewards on collateral effectively lowers the cost of borrowing;
- stETH/wstETH is a very liquid token with billions of liquidity locked in liquidity pools onCurve
- ,Balancer v2

-)

Lido's staked tokens have been listed on major liquidity protocols:

- On Maker,wstETH collateral (scroll down to Dai from WSTETH-A section)
- can be used to mint DAI stablecoin. SeeLido's blog post
- for more details.
- On AAVE, multiple tokens can beborrowed against stETH
- . SeeLido's blog post
- for more details. Please note: stETH is only supported on AAVE as lending collateral. Borrowing stETH on AAVE is not currently supported. However, any asset can be borrowed on AAVe via a flashloan. Due to a known 1-2 wei corner case)
- article for more details.

Robust price sources are required for listing on most money markets, with ChainLink price feeds being the industry standard. There're live ChainLinkstETH/USD andstETH/ETH price feeds on Ethereum.

Wallet integrations

Lido's Ethereum staking services have been successfully integrated into the most popular DeFi wallets, including Ledger, MyEtherWallet, ImToken and others. Having stETH integrated can provide wallet users with a great user experience of direct staking from the wallet UI itself.

Lido DAO runs a referral program rewarding wallets and other apps for driving liquidity to the Lido staking protocol. At the moment, the referral program is inwhitelist mode . Please contact Lido bizdev team to find out if your wallet might be eligible for referral program participation.

When adding stETH support to a DeFi wallet, it is important to preserve stETH's rebasing nature. Avoid storing cached stETH balance for extended periods of time (over 24 hours), and keep in mind it doesn't necessarily take a transaction to change stETH balance.

Liquidity mining

stETH liquidity is mostly concentrated in two biggest liquidity pools:

- stETH/ETH liquidity pool on Curve
- (contract code
-)
- wstETH/WETH MetaStable pool on Balancer v2
- (read more
-)

Both pools are incentivized with Lido governance token (LDO) via direct incentives and bribes (veBAL bribes coming soon), and allow the liquidity providers to retain their exposure to getting Lido staking rewards.

- Curve pool allows providing liquidity in the form of any of the pooled tokens or in both of them. From that moment on, all the staking rewards accrued by stETH go to the pool and not to the liquidity provider's address. However, when withdrawing the liquidity, the liquidity provider will be able to get more than they have initially deposited.
- Please note, when depositing exclusively stETH to Curve, the tokens are split between ether and stETH, with the precise balances fluctuating constantly due to price trading. Thus, the liquidity provider will only be eligible for about half of the rewards accrued by the stETH deposited. To avoid that, provide stETH and ether liquidity in equal parts.
- Unlike Curve, the Balancer pool is wstETH-based. wstETH doesn't rebase, it accrues staking rewards by eventually increasing in price instead. Thus, when withdrawing liquidity from the Balancer pool, the liquidity providers get tokens valued higher than what they have initially deposited.

Cross chain bridging

The Lido's stTokens will eventually get bridged to various L2's and sidechains. Most cross-chain token bridges have no mechanics to handle rebases. This means bridging stETH to other chains will prevent stakers from collecting their staking rewards. In the most common case, the rewards will naturally go to the bridge smart contract and never make it to the stakers. While working on full-blown bridging solutions, the Lido contributors encourage the users to only bridge the non-rebasable representation of staked ether, namely wstETH.

Risks

1. Smart contract security.
2. There is an inherent risk that Lido could contain a smart contract vulnerability or bug. The Lido code is open-sourced, audited and covered by an extensive bug bounty program to minimize this risk.
3. To mitigate smart contract risks, all of the core Lido contracts are audited. Audit reports can be foundhere

4. .
5. Besides, Lido is covered with a massiveImmunefi bug-bounty program
6. .
7. Beacon chain - Technical risk.
8. Lido is built atop experimental technology under active development, and there is no guarantee that the Beacon Chain has been developed error-free. Any vulnerabilities inherent to the Beacon Chain bring with it slashing risk, as well as stETH balance fluctuation risk.
9. Slashing risk.
10. Beacon chain validators risk staking penalties, with up to 100% of staked funds at risk if validators fail. To minimize this risk, Lido stakes across multiple professional and reputable node operators with heterogeneous setups, with additional mitigation in the form of self-coverage.
11. stETH price risk.
12. Users risk an exchange price of stETH which is lower than inherent value due to withdrawal restrictions on Lido, making arbitrage and risk-free market-making impossible. The Lido DAO is driven to mitigate the above risks and eliminate them to the extent possible. Despite this, they may still exist and, as such, we must communicate them. Edit this page Previous Lido DAO Next General Overview