Required reading:

- Understanding the validator lifecycle
- A note on Ethereum 2.0 phase 0 validator lifecycle - HackMD
- eth2.0-specs/beacon-chain.md at f2eb5e38e2ad651afe1d55eece2e3d28260c2b4a · ethereum/eth2.0-specs · GitHub (namely, initiate_validator_exit

, slash_validator

, and process_slashings

are most important)

- GitHub - lidofinance/lido-dao: Lido DAO smart contracts

Withdrawals in liquid staking

Withdrawals in liquid staking is a fairly complex mechanics, because it needs to take into account a number of simultaneous, rarely communicating processes in a way that results in the fair distribuitions of rewards and penalties for all the users.

Here are events and processes that need to be accounted for for a purpose of a withdrawal process in Lido.

1. Withdrawal request from a staker
2. Withdrawal signal from a validator
3. Oracle reports on network state
4. Ongoing rewards and penalties
5. Ongoing slashings
6. Unbonding period
7. New penalties and slashings during onbonding
8. Network turbulence (e.g. lack of finality)

Most of these points are fairly comlicated by themselves in eth2. It would be fair to say that handling slashings during withdrawals is the main problem for Lido

Moving parts of ETH2 withdrawal from Lido

Lido’s design choices

- slashings and rewards are socialized between all stakers
- node operators don’t have collateral
- slashing risks are covered
- stETH is minted immediately on deposit and starts to receive socialized rewards/is under slashing risks, even if the validators deposited with that eth are still in the entry queue

Withdrawal request from a staker

---.----------.---------- withdrawal request tx withdrawal requests finalized mined (t+0) (~t+300+ until phase 1.5, ~t+720 after)

additional issue is that withdrawals can be for any amount

Withdrawal signal from a validator

There is an issue that withdrawal txs can be in any amount, and validator can only proceed it in terms of distinct validators (~30-40eth chunks)

----.----------.----------.---------- staker withdrawal validators stop tx validators stop tx requests finalized (~t+720) issued (~t+720+) finalized (~t+1440+)

Oracle reports on network state

Oracle reports on network state are done about once/day when there is a finality on the network.

Ongoing rewards

Rewards and offline penalties continue to accrue during all the time it takes from a withdrawal request to the exit from withdrawal queue - need to distribute them fairly between withdrawal requesters and general stakers.

Ongoing slashings

Slashing is fairly complicated in eth2. The timeline looks like that:

1. An offesne is made by a validator (t+0)
2. An offense is reported (t2=t+???)
3. A small fixed initial slashing happens (t2)
4. About 18 days (maybe more but unlikely) of inactivity for the validator, during which midpoint slashing size (see process_slashings

in the spec) is determined

1. A midpoint slahsing happens, that can range from 0 to 100% of validator balance, depending on share of validators slashed (at this point slashing effects can be considered known and resolved)
2. about 18 days after (maybe more but unlikely), the remaining funds are withdrawable

Unbonding period

Unbonding takes validator going through exit queue (from 30m to a few weeks) and additional ~27 hours until funds are withdrawable. When in exit queue validator can be slashed or penalized. If it's slashed, see above.

Network turbulence

In case of unforeseen problems with client logic ETH2 can lose finality at any time for a long period of time, which will make unclear what's actually happening there re: slashings and exits until finality is in.

Out of scope, for now

Algorithm to select validators to unstake; gas concerns; the details of implementation. I believe it can be abstracted away for now.

Strawman withdrawal methods

I'll go through a couple strawman methods to show where they break, and then present three I consider to be working ones.

Simple burn and withdraw

Staker sends his stETH to the withdrawal smart contract. The amount of stETH is stored and node operators withdraw enough validators to have that amount filled. When validator withdrawals are claimed, and there's enough eth in withdrawal contract, withdrawer can claim their eth.

Failure scenario

Imagine there's an ongoing slashing that has not been reported yet but is already known. People who withdraw are not subject to it, and people who don't take the full effect when the oracle reports the slashing. This leads to a bank run of sorts, where people that are most important for the platform (long term stakers) take the most amount of harm.

Minor issues

While waiting for an exit queue to clear withdrawers do not get rewards.

Batched burn and withdraw after an oracle report

Staker sends his stETH to the withdrawal smart contract. The amount of stETH is noted. After a successful oracle report all stETH amounts are adjusted and than withdrawals for the day are batch-processed. Nnode operators withdraw enough

validators to have the burned amount filled. When validator withdrawals are claimed, and there's enough eth in withdrawal contract, withdrawer can claim their eth.

Failure scenario

Imagine there's an ongoing slashing. People who withdraw will suffer an effect of an initial slashing penalty, but not midpoint slashing penalty. People who don't take the full effect when the oracle reports the midpoint slashing. This leads to a bank run of sorts, where people that are most important for the platform (long term stakers) take the most amount of harm.

Minor issues

While waiting for an exit queue to clear withdrawers do not get rewards.

Batched burn and withdraw after an oracle report + delay if there's a an ongoing slashing

Staker sends his stETH to the withdrawal smart contract. The amount of stETH is noted. After a successful oracle report all stETH amounts are adjusted and than withdrawals for the day are batch-processed. If there are ongoing slashings, withdrawals initiated in the day it happened or after, are not processed until the midpoint slashing takes effect. Node operators withdraw enough validators to have the burned amount filled. When validator withdrawals are claimed, and there's enough eth in withdrawal contract, withdrawer can claim their eth.

Failure scenario

Imagine there's an ongoing slashing. People who withdraw will suffer an effect of an initial slashing penalty, and midpoint slashing penalty, and effects of an initial slashing penalty for any new slashings, but not midpoint slashings for any new slashing. People who don't take the full effect of both the original and new slashings. This leads to an incetive to derisk your eth as soon as there is a slashing. This might result in a bank run of sorts, where people that are most important for the platform (long term stakers) take the most amount of harm. This is much less pronounced than previous options though, almost ignorable.

Minor issues

While waiting for an exit queue to clear withdrawers do not get rewards.

Non-strawman withdrawal methods

There are three options I see for withdrawals that can work with this, both fairly complicated.

They have in common:

1. No withdrawals are possible in time of no finality in eth2.
2. All of the day's withdrawals are batched and processed after an oracle report for an epoch that is at least N hours after withdrawal requests (between 1h and 24h, think 8h is good enough). That is done to be sure that request is final (no reogrs are possible) and that all pre-withdrawal slashings can be reasonably considered to be reported.

Withdrawal method A

Staker sends his stETH to the withdrawal smart contract. The amount of stETH is noted. After a successful oracle report all stETH amounts are adjusted and than withdrawals for the day are batch-processed. If there are ongoing slashings, withdrawals initiated in the day it happened or after, are not processed until the midpoint slashing takes effect.

If there are any new slashings during that time, processing is delayed to take new midpoint slashings too, up to a full slashing duration of 36 days.

Node operators withdraw enough validators to have the burned amount filled. When validator withdrawals are claimed, and there's enough eth in withdrawal contract, withdrawer can claim their eth.

If there's no slashings, a regular withdrawal will take between 28 hours and three days, depending on withdrawal congestion, and relative time of withdrawal signal and oracles report.

Withdrawal method A.1

Staker sends his stETH to the withdrawal smart contract. The amount of stETH is noted. After a successful oracle report all stETH amounts are adjusted and than withdrawals for the day are batch-processed. If there are ongoing slashings, withdrawals initiated in the day it happened or after, are not processed until the midpoint slashing takes effect.

The stETH amount noted in the first step limits the amount of eth a person can withdraw - so they don't have any staking

upside but bear the slashing risk. It prevents a “bank run” scenario from the last strawman method.

Node operators withdraw enough validators to have the burned amount filled. When validator withdrawals are claimed, and there’s enough eth in withdrawal contract, withdrawer can claim their eth.

If there’s no slashings, a regular withdrawal will take between 28 hours and three days, depending on withdrawal congestion, and relative time of withdrawal signal and oracles report.

Withdrawal method B

The core of that method is to unsocialize withdrawals.

If there are no unresolved slashings at batch processing time (resolved slashing is the one that got its midpoint balance decrease):

During batch processing, every withdrawal is associated with a set of validators exited to process it. From this point on their risks and rewards are not shared with the main staking pool and are depending on validators assigned to the specific withdrawal. May be made more safe by purchasing a separate slashing cover (e.g. Unslashed’s cover).

When these validators exit, every withdrawal request is fullfilled separately.

If there are unresolved slashings, withdrawals are locked until they are resolved and that slashing effect is socialized, but new slashings after that for general lido pool do not impact earlier withdrawals.

If there’s no slashings, a regular withdrawal will take between 28 hours and three days, depending on withdrawal congestion, and relative time of withdrawal signal and oracles report.

Conclusion

After talking to some people about this I came to conclusion that method B is not the way people want their withdrawals to work.

Between methods A and A.1 they almost to person have chosen A.1 as universallly more preferable (losing on a bit of rewards is better than waiting for extra weeks in the worst case).

Methods A and A.1 can be implemented optionally (e.g. default is A1 and A is opt-in). I don’t think it’s a good idea to start with implementing both, but we can have A.1 implemented first and then bolt on A as an option if there’s some demand for that.