Ensuring Compatibility with Ethereum's Pectra Upgrade

This post outlines the technical details of the proposed upgrade to the Lido protocol to ensure continuous operation with Ethereum's upcoming Pectra hardfork. The focus is exclusively on compatibility, rather than integrating any new features introduced by Pectra. Pectra-related new functionalities, such as Triggerable Withdrawals via EIP-7002 and increasing the MAX_EFFECTIVE_BALANCE

& validator consolidations via EIP-7251, will be addressed in separate posts.

Following the forum discussion, this proposal will be presented on Snapshot to seek DAO support for preparing Lido on Ethereum for the Pectra upgrade. If the proposal is supported, two on-chain votes will follow: one before and one after the Pectra hardfork.

The proposal's technical scope consists of three parts:

1. Deployment of a new version of CS Verifier Contract;

2. Items for the further on-chain vote before

the hardfork;

1. Items for the further on-chain vote after

the hardfork.

The on-chain vote before

the hardfork includes only the minimal changes necessary to keep the Lido protocol functioning correctly after the Pectra upgrade. The on-chain vote after

the hardfork includes optimizations that have no strict timeline and can be implemented without a set deadline post-upgrade.

The following sections outline how updates to Lido's Oracles

, Oracle Report Sanity Checker

, and CS Verifier

contracts are proposed. Details of the upcoming Pectra upgrade, its impact on the Lido protocol, and recommended protocol changes can be found in LIP-27: Ensuring Compatibility with Ethereum's Pectra Upgrade.

Valset technical contributors are available to address any questions that may arise.

Part 1. Deployment

CS Verifier Contract

Remove Initial Slashing Penalty Report

Under Pectra, the penalty for newly slashed validators is roughly 0.0078 ETH for a 32 ETH validator, rather than ~1 ETH.

Therefore, it's proposed to remove the initial slashing penalty report from the CS Verifier contract before epy Pectra hardfork activation. Accounting of losses for initial slashing will take place at the moment of reporting the full withdrawal of the validator.

Update gIndexes

It is proposed to redeploy the contract with two sets of gIndexes: one for the current (Deneb) fork version and one for the Electra fork version, along with a specific slot number for the hardfork activation.

// gIndexes calculation: // https://github.com/lidofinance/community-staking-module/blob/4d5f4700e356dc502c484456fbf924cba56206ad/script/gindex.mjs#L36

GI_FIRST_WITHDRAWAL_PREV = 0x0000000000000000000000000000000000000000000000000000000000e1c004; GI_FIRST_WITHDRAWAL_CURR = 0x0000000000000000000000000000000000000000000000000000000000161c004;

GI_FIRST_VALIDATOR_PREV = 0x0000000000000000000000000000000000000000000000000056000000000028; GI_FIRST_VALIDATOR_CURR = 0x0000000000000000000000000000000000000000000000000096000000000028;

GI_HISTORICAL_SUMMARIES_PREV = 0x0000000000000000000000000000000000000000000000000000000000003b00; GI_HISTORICAL_SUMMARIES_CURR = 0x0000000000000000000000000000000000000000000000000000000000005b00;

// The first slot of the Deneb activation epoch https://eips.ethereum.org/EIPS/eip-7569#activation FIRST_SUPPORTED_SLOT = 8626176;

// TBA after the Pectra activation epoch for mainnet is known // first slot of the Electra activation epoch PIVOT_SLOT =

Part 2. Vote before the Pectra hardfork

Oracles consensus version

It is proposed to increment the consensus version for Accounting, Validator Exit Bus, and CSM oracles.

* [Accounting Oracle 0x852deD011285fe67063a08005c71a85690503Cee

](https://etherscan.io/address/0x852deD011285fe67063a08005c71a85690503Cee#readProxyContract#F16): from 2

to 3

* [Validator Exit Bus Oracle 0x0De4Ea0184c2ad0BacA7183356Aea5B8d5Bf5c6e

](https://etherscan.io/address/0x0De4Ea0184c2ad0BacA7183356Aea5B8d5Bf5c6e#readProxyContract#F13) from 2

to 3

* [CSM Fee Oracle 0x4D4074628678Bd302921c20573EEa1ed38DdF7FB

](https://etherscan.io/address/0x4D4074628678Bd302921c20573EEa1ed38DdF7FB#readProxyContract#F16) from 1

to 2

The new consensus version will signal the oracles to switch to the new algorithm that considers the transition through Pectra: before the hardfork, the algorithm collects the report in the old way, after the hardfork, in the new way.

Update CS Verifier Contract

It is proposed to revoke the verifier role on the Community Staking Module contract from the old CS Verifier contract and grant it to the new CS Verifier contract, which will be deployed in advance (as described in the "Deployment" section above).

This update involves two steps:

1. Revoke

the [VERIFIER_ROLE (0x0ce23c3e399818cfee81a7ab0880f714e53d7672b08df0fa62f2843416e1ea09

)](https://etherscan.io/address/0xdA7dE2ECdDfccC6c3AF10108Db212ACBBf9EA83F#readProxyContract#F16) on the [Community Staking Module (0xdA7dE2ECdDfccC6c3AF10108Db212ACBBf9EA83F

)](https://etherscan.io/address/0xdA7dE2ECdDfccC6c3AF10108Db212ACBBf9EA83F) contract from the old [CS Verifier (0x3Dfc50f22aCA652a0a6F28a0F892ab62074b5583

)](https://etherscan.io/address/0x3Dfc50f22aCA652a0a6F28a0F892ab62074b5583) contract.

1. Grant

the [VERIFIER_ROLE (0x0ce23c3e399818cfee81a7ab0880f714e53d7672b08df0fa62f2843416e1ea09

)](https://etherscan.io/address/0xdA7dE2ECdDfccC6c3AF10108Db212ACBBf9EA83F#readProxyContract#F16) on the [Community Staking Module (0xdA7dE2ECdDfccC6c3AF10108Db212ACBBf9EA83F

)](https://etherscan.io/address/0xdA7dE2ECdDfccC6c3AF10108Db212ACBBf9EA83F) contract to the new CS Verifier contract. The address of the new CS Verifier will be posted in this thread after deployment.

Part 3. Vote after the Pectra hardfork

Sanity Checker

It is proposed to update the following parameters in the Oracle Report Sanity Checker contract [0x6232397ebac4f5772e53285b26c47914e9461e75

](https://etherscan.io/address/0x6232397ebac4f5772e53285b26c47914e9461e75):

- exitedValidatorsPerDayLimit

: from 9000

to 3600

;

- appearedValidatorsPerDayLimit

: from 43200

to 1800

;

- initialSlashingAmountPWe

: from 1000

to 8

.

exitedValidatorsPerDayLimit = 3600

This parameter defines the maximum number of validators that may be reported as "exited" per day, depending on the consensus layer churn limit.

In Pectra, a limit MAX_PER_EPOCH_ACTIVATION_EXIT_CHURN_LIMIT

is introduced on the amount of ETH that can be exited per epoch. Therefore, the maximum number of validators that can exit the network per day can be simplified and calculated as:

https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#time-parameters

SLOTS_PER_EPOCH = 32

https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#time-parameters-1

SECONDS_PER_SLOT = 12

https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#validator-cycle

EJECTION_BALANCE = 16 * 10 ** 9

https://github.com/ethereum/consensus-specs/blob/dev/specs/electra/beacon-chain.md#validator-cycle

MAX_PER_EPOCH_ACTIVATION_EXIT_CHURN_LIMIT = 256 * 10 ** 9

epochs_per_day = 24 * 60 * 60 / SECONDS_PER_SLOT / SLOTS_PER_EPOCH = 225 exited_validators_per_epoch_limit = MAX_PER_EPOCH_ACTIVATION_EXIT_CHURN_LIMIT / EJECTION_BALANCE = 16 exited_validators_per_day_limit = exited_validators_per_epoch_limit * epochs_per_day = 3600

appearedValidatorsPerDayLimit = 1800

This parameter defines the maximum number of validators that can be reported as "appeared" in a single day, limited by the maximum daily deposits via the Deposit Security Module.

In Pectra, validators are added to the registry after their deposit passes through the pending_deposits

queue, which is limited by the [MAX_PER_EPOCH_ACTIVATION_EXIT_CHURN_LIMIT

](https://github.com/ethereum/consensus-specs/blob/dev/specs/electra/beacon-chain.md#validator-cycle). Given that Lido validators are deposited exclusively with 32 ETH each, the maximum number of Lido validators that can appear on the network per day can be calculated as:

https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#time-parameters

SLOTS_PER_EPOCH = 32

https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#time-parameters-1

SECONDS_PER_SLOT = 12

https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-

chain.md#validator-cycle

EJECTION_BALANCE = 16 * 10 ** 9

https://github.com/ethereum/consensus-specs/blob/dev/specs/electra/beacon-chain.md#validator-cycle

MAX_PER_EPOCH_ACTIVATION_EXIT_CHURN_LIMIT = 256 * 10 ** 9

https://github.com/lidofinance/core/blob/901c0e19b752f5ea03118d71881d8b72ccee44c0/contra

DEPOSIT_SIZE = 32 * 10 ** 9

epochs_per_day = 24 * 60 * 60 / SECONDS_PER_SLOT / SLOTS_PER_EPOCH = 225 appeared_validators_per_epoch_limit = MAX_PER_EPOCH_ACTIVATION_EXIT_CHURN_LIMIT / DEPOSIT_SIZE = 8 appeared_validators_per_day_limit = appeared_validators_per_epoch_limit * epochs_per_day = 1800

initialSlashingAmountPWei = 8

(0.008 ETH)

This parameter defines the maximum initial slashing penalty for validators in the case of misbehavior. It ensures that a slashed validator can be penalized up to 0.008 ETH, contributing to the overall negative rebase calculation during the sanity check process.

In Pectra, MIN_SLASHING_PENALTY_QUOTIENT_ELECTRA

increases by 128 times from 32

to 4096

, reducing the initial penalty size by 128 times. Since Lido validators exclusively use the [ETH1_ADDRESS_WITHDRAWAL_PREFIX

](https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/validator.md#eth1_address_withdrawal_prefix), their effective balance [is limited to MIN_ACTIVATION_BALANCE

](https://github.com/ethereum/consensus-specs/blob/dev/specs/electra/beacon-chain.md#new-get_max_effective_balance). Therefore, the maximum initial slashing penalty is calculated as:

https://github.com/ethereum/consensus-specs/blob/dev/specs/electra/beacon-chain.md#gwei-values

MIN_ACTIVATION_BALANCE = 32 * 10 ** 9

https://github.com/ethereum/consensus-specs/blob/dev/specs/electra/beacon-chain.md#rewards-and-penalties

MIN_SLASHING_PENALTY_QUOTIENT_ELECTRA = 4096

initial_slashing_amount_gwei = MIN_ACTIVATION_BALANCE / MIN_SLASHING_PENALTY_QUOTIENT_ELECTRA = 0.0078125 initial_slashing_amount_pwei = initial_slashing_amount * 10 ** 3 = 7.8125 # ~8

Fallback Plan

If any risks related to the on-chain voting before the Pectra upgrade are identified, or if the DAO does not approve the proposed update, the fallback plan will be implemented.

One day before the Pectra upgrade, deposits will be paused, and the unpause of deposits will be put to a separate vote for the DAO to decide after Pectra. This measure is necessary to ensure the stable operation of the Lido protocol.

This scenario is unlikely, but if it materializes, a detailed action plan will be outlined separately on the forum.

Coming Soon: Deployed Contracts and Audit Verifications

After the deployment, addresses for the newly deployed contracts will be provided. These contracts will undergo verification by external auditors, audit reports will be released, detailing the findings and mitigations before the on-chain votes.

Additionally, descriptions of the items up for vote will be published, offering the community a clearer understanding of the proposed changes and their implications for the protocol.