Note: this is a copy of a document on hackmd which can be found (and commented on, if preferred to writing in the forum) here . It is suggested to use the hackmd version of the document to read the draft, as the table of contents and internal links make it easier to navigate.

The purpose of this document is to attempt to join the the discussions we've previously had in two separate topics, Ethereum MEV Extraction and Rewards - Discussion & Policy Groundworkand [Proposal] Optimal MEV Policy for Lido , into one policy document that can be considered by the DAO.

Apart from discussion, improvements, changes, and finalization of TBD/pending elements in the draft, I think that the following topics are also open and should be settled before the policy is finalized:

- If PBS-lite infrastructure is adopted, when should it be enabled? (e.g. at Merge, slightly after the Merge)
- Testing any required monitoring setups
- Identifying, vetting, and testing relays before the PBS-lite infrastructure is enabled
- Any other open topics/items identified by rest of the community

Lido on Ethereum Block Proposer Rewards Policy

STATUS: DRAFT Open for comment

A. Overview

As a DAO and a protocol, Lido on Ethereum should have a transparent, enforceable, and monitorable policy with regards to how its constituent Node Operators are expected to behave with regards to rewards that accrue from the activity of producing blocks for validators that they run as a part of Lido, namelypriority fees and MEV extraction, and how the protocol will distribute the rewards that accrue to the protocol as a result of these activities.

B. Purpose

This policy aims to outline how Node Operators participating in Lido should distribute rewards obtained due to block production (including potential MEV rewards), what mechanisms or infrastructure may be used in execution of this, how rewards will be distributed, and how these activities will be monitored.

C. Scope

This policy applies to the Lido on Ethereum protocol, theNode Operators that participate in the protocol, and the validators that they operate as a part of the protocol.

D. Policy Statement

D.1 General goals

There are two key questions that this policy addresses:

- How should validators run by Node Operators propose blocks? and
- How should validators run by Node Operators distribute the rewards that result from block proposals?

Since the consensus within the Ethereum community is to move towards aPBS system where the building of blocks is separated from the proposal of blocks, Lido should seek to adopt an approach as close to a most likely draft of PBS as possible. Lido could become a testbed for this feature which is easier to roll back than a full-fledged implementation if something goes wrong.

On the rewards side, validators are powered by deposits fromstakers, and thus stakers should always receive the lion's share of rewards.

To that end, Lido as a protocol should:

1. Enforce - to the technical extent possible - that blocks are not built by Node Operators, but instead built by third parties and obtained from open and transparent builder markets.
2. Require that Node Operators ensure that relevant rewards are routed to the [Lido Execution Layer Rewards vault

](https://docs.lido.fi/guides/node-operator-manual/#execution-layer-feesrewards-configuration-priority-fee-and-mev-rewardscollection-and-distribution) which, as per LIP-12, will periodically be triggered to stake the rewards (less the Lido protocol fees), thereby essentially increasing staker rewards.

1. Ensure, to the extent possible, that proposed blocks are valid and do not damage the proper and sustainable operation of the underlying protocol.

2. Monitor Node Operator behavior (monitoring will be publicly available) and penalize Node Operators who are found to not be in accordance with the above stated goals.

D.2 Node Operator requirements

D.2.I Priority Fees

Priority fee receipt and distribution

Priority fees are sent to the fee recipient

configured for a validator. In general Node Operators should ensure that priority fees should be sent to the Lido Execution Layer Rewards Vault

.

In the case that the fee recipient

is otherwise overridden (e.g. due the execution of a block with extracted MEV where the fee recipient

points to a block builder's address instead), a transaction within the payload should exist from the block builder to the Lido Execution Layer Rewards Vault with at least the sum-total of priority fees for that block (for transactions sourced from the public mempool), plus any additional rewards from MEV extraction.

D.2.II MEV and MEV rewards

The optimal MEV policy for Lido is to maximize staking APR and Ethereum security while minimizingMEV hiding and decreasing the power that Lido and Node Operators have over the network by virtue of running validators (i.e by separating block building from block proposal).

[

1068×707 140 KB

](https://europe1.discourse-cdn.com/business20/uploads/lido/original/2X/c/c8c9a485d236a9f269bf7075eab0096140e88f37.jpeg)

Maximize Staking APR

Extracting MEV is good for Lido because as a staking protocol it competes for stake from users on the one side and quality Node Operators on the other. Both are maximized when staking APR is maximized. Under an alternative policy that doesn't maximize APR, stakers would be more likely to delegate to a different provider, and Node Operators would be more likely to circumvent Lido and market to users directly, and/or participate in Lido but attempt to extract the MEV in a hidden way and hide the rewards from Lido and stakers (i.e. "MEV hiding").

Maximize Ethereum security

Extracted MEV helps the economic security of the protocol "A system where validators 'leave MEV on the table' is one where an obvious attacker subsidy is readily available. This is self-defeating and degrades security in a pure-economic-rationality model, as it causes centralization/oligopoly." To make blockchains maximally robust, all honest actors need to extract MEV at the maximum rate, otherwise they dramatically lower the cost for a dishonest actor to attack the system.

Minimize MEV hiding

The MEV hiding problem is defined as follows: Node Operators handle the staking in return for a fee on the overall staking reward. However, because Node Operators can receive money in direct transfers or off-chain, the actual total rewards received by virtue of running a validator can carry a lot of uncertainty. Lido can ensure that stakers get their cut of the rewards it can see, but not of the rewards it cannot see – thus, as a protocol it should attempt to ensure that all rewards are being recorded on chain.

An example MEV-hiding scenario is where Node Operators report 1 ETH reward per block to Lido (of which they receive 5%) but hide 0.5 ETH reward from Lido (of which they get to keep 100%). This is a large carrot for Node Operators at the expense of Lido and its stakers.

The ability to surreptitiously steal decreases as confidence about the true size of the reward increases. In other words, it is an oracle problem. Staying with the above example, if Lido knows the true value of the block is 1.5 ETH, the Node Operator can still try to report 1 ETH to Lido, but the theft would be easily detectable.

To resolve the MEV-hiding problem, you need a reliable oracle on the true block reward (or the block reward needs to be guaranteed in-protocol, e.g. via two-slot PBS). Until such a solution is possible in-protocol, infrastructure needs to be used to achieve similar outcomes to PBS but within the current technical limitations of the Ethereum protocol.

Minimize power over blocks

Lido's potential to grow as a staking protocol is inversely proportional to the risk it poses potentially resulting from unchecked control over block production due to the amount of stake that runs through the protocol. By minimizing and separating the power that comes from running validators (i.e. segregating proposal and building), Lido can focus on its remit: maximizing staking rewards and cultivating an excellent validator set. The underlying problem is not that Lido wants to hurt Ethereum users but that it can't prove that it doesn't, or at some point won't; this is a trust problem. Therefore the optimal course of action is to minimize how much users and the Ethereum ecosystem need to trust Lido and Node Operators.

The requirement that Node Operators source blocks from open markets effectively removes the potential for the protocol to coordinate or exercise influence over the content of the blocks being built by the Node Operators, and thus reduces the overall risk that the protocol may pose to the underlying network.

D.3 Implementation

The DAO should periodically identify and evaluate what solutions are available on the market to achieve the above-stated goals. Additionally, the infrastructure and overall solution(s) adopted should:

- Be transparent and open (i.e. ideally open source) and monitorable
- Be fully tested by all Node Operators prior to usage on mainnet
- Not reduce the safety, security, or effective operation of the Lido protocol or the underlying network (Ethereum)

D.3.I Available infrastructure

As at the time of the most recent policy update, contributors to Lido have identified the following infrastructure that can satisfy the goals and implementation requirements outlined:

Infrastructure

Description

Implementation Details

MEV-Boost

MEV-boost is an out-of-protocol implementation of PBS by Flashbots. It works by separating concerns across at least three roles:

- Block Builders (responsible for assembling block content and sending them along with bids to relays)
- Relays (responsible for aggregating, sorting, and simulating bids from builders, sending them to Block Proposers (first encrypted, then revealed once they have been accepted right before the block is published)
- Block Proposers (responsible for interfacing with relays and selecting block proposals based on the bids that have been received).

Notes:

(1) There may be more actors (especially upstream of Block Builders) but they're not especially important for this policy.

(2) It's possible that a Node Operator may vertically integrate several of the above roles, but in the case of Lido Node Operators are asked to not run their own relays (as this might allow for MEV hiding or cheating).

See Appendix A.1

Default block-building

Normal block building via the public mempool using standard / un-modified Execution Layer client software.

N/A

D.4 Monitoring & Penalties

Monitoring

Appropriate monitoring should be implemented (or leveraged, e.g. from third parties) in order:

- Determine whether the Node Operators are behaving as intended
- Participants (e.g. block builders, relays) in the the block building process are acting in good faith and as expected
- Normal validator and protocol operations are not adversely affected by the running of MEV-related infra

Penalties

Node Operators who are found to be engaging in activities against the spirit of this policy and its stated goals will be subject to:

- Potential refund requests for amounts found to be incorrectly or inappropriately not distributed to the protocol
- Potential cessation of new stake deposits to their validators
- Potential stake withdrawal and validators being exited (voluntarily or via triggerable exits, when possible) and off-boarding from the Lido on Ethereum set
- Potential adverse effects on their general standing within the Lido ecosystem (e.g. having stake delegated to them in other Lido on X networks reduced, or being offboarded)

D.4.I Infrastructure and Availability

Monitoring with regards to Node Operator block rewards performance and possible MEV extracted may be made available through two avenues:

- Infrastructure managed by contributors to Lido; this infrastructure should be open and publicly accessible
- Infrastructure managed by third parties which includes Lido-related data; contributors to Lido should work with these third parties to ensure data related to Node Operators is easily accessible so that it may be in integrated into third party analytics and reporting tools

For details about monitoring implementations for different proposed infrastructure solutions, please see the relevant Appendix per infrastructure.

D.4.II Monitoring & Penalties Specification

Lido should monitor, record, and assess non-compliance with this policy. Monitoring and follow-up actions should include at least the following:

- Assessing whether the correct fee recipient

has been configured by the Node Operators for the validators that they operate

- Assessing from which source proposed blocks by ethe validators have been received (and if that source is appropriate)
- Assessing the general availability of the allowed sources for block bids around the time of each block being proposed
- Assessing, in cases where blocks were not constructed via chosen MEV infrastructure, whether the block proposed contained transactions not available in the public mempool
- Recording, on a per-block basis, the most valuable known block bids offered via accepted MEV infrastructure (and block bid sources, e.g. relays), i.e. the reference block
- Recording, on a per-block basis, the blocks actually produced by the validators and their total value, i.e. the actual block
- Analyzing, on a per-block basis and historically, the difference between the reference block

and the actual block

for each Node Operator participating in Lido

Some leeway should be given to Node Operators for potential network problems, faulty relays, or buggy software outside their control. It should be enough for the Lido DAO to punish gross misbehavior when a Node Operator is deviating more than ALLOWED_VALUE_DEV

% from the reference block, on a frequency basis more than ALLOWED_FREQ_DEV

% of the time over a rolling time window DEV_WINDOW

. These parameters can be reviewed and tweaked via governance, if needed.

MEV Monitoring & Penalty parameters

Variable

Value

Description

History

ALLOWED_VALUE_DEV

TBD

In percentage terms, how much less value an actual block

must contain compared to the respective reference block

in order to be considered inappropriate

ALLOWED_FREQ_DEV

TBD

In percentage terms, how often a Node Operator may deviate from the reference block

within window DEV_WINDOW

before being considered in breach of policy

DEV_WINDOW

TBD

In proposed blocks, the length of the rolling window

D.5 Currently prescribed solution(s)

This section will be reviewed by the DAO and updated on an at-least yearly basis and more often if needed. It details which block production solutions may be used by Node Operators at the current time.

Applicability period: YYYY/MM/DD - YYYY/MM+6/DD Unless otherwise overriden by a more recent DAO vote

Lido should aid Ethereum in moving towards its stated goal, PBS. To that end:

1. Validators operatoed by Node Operators participating in the Lido protocol should produce blocks via the MEV-Boost infrastructure as detailed in Appendix A.1, by obtaining sealed blocks from the maximum possible number of relays (to be determined by each Node Operator based on their own risk and legal assesment) from Lido's "must-include list" and an optional number of relays from the "allow list".

2. If using MEV-boost infrastructure leads to any operational failures/difficulties (e.g. failing to produce valid blocks, blocks at all, received rewards being incorrect, or lack of availability of appropriate relays), the Node Operator may fall back to building blocks via the "Default" block-building method.

3. Blocks produced by the validators will be monitored as perMonitoring & Penalties section and the monitoring section of the relevant Appendices.

4. Node Operators acting in violation of the policy are subject to penalties as described in theMonitoring & Penalties section.

E. Definitions

Block builder

The agent responsible for assembling the potential contents of a block (i.e. the set of transactions that makes up the “payload” of a block). In Proof of Work Ethereum it’s the miner. In Proof of Stake Ethereum it’s the validator, but this activity can separated from the validator via PBS, in which case builders need to submit bids for blocks to be included to theblock proposer for inclusion.

Block proposer

The agent responsible for proposing new blocks in a blockchain. In Proof of Work Ethereum it’s the miner. In Proof of Stake Ethereum it’s the validator.

MEV

Originally “Miner Extractable Value” as applied to Proof of Work blockchains, the more all-encompassing “Maximal Extractable Value” terminology is now used to refer to the potential value that may be extracted from activity of the selection, ordering, and insertion of transactions into a block by block proposers.

Node Operator

An organization/entity/person that operates validators. In the context of Lido, ones that do so as a part of the Lido protocol.

PBS

Proposer Builder Separation, see: State of research: increasing censorship resistance of transactions under proposer/builder separation (PBS) - HackMD and https://members.delphidigital.io/reports/the-hitchhikers-guide-to-ethereum/ (section “In-protocol Proposer-Builder Separation”)

Priority Fees

See Gas and fees | ethereum.org

Validator

See Proof-of-stake (PoS) | ethereum.org . In the context of Lido, validators operated by 3rd partyNode Operators for the Lido protocol.

Staker

A user/organization/etc. which holds (w)stETH.

F. Related Documents and References

I. Revision History

Version

Date

Description

0.1

Aug 10, 2022

Initial policy creation

Appendix A - Detailed MEV Lido Implementations

Appendix A.1 MEV-Boost

MEV-boost is an out-of-protocol implementation of PBS by Flashbots. It works by separating concerns across at least three roles:

- Block Builders (responsible for assembling block content and sending them along with bids to relays)
- Relays (responsible for aggregating, sorting, and simulating bids from builders, sending them to Block Proposers (first encrypted, then revealed once they have been accepted right before the block is published)

- Block Proposers (responsible for interfacing with relays and selecting block proposals based on the bids that have been received).

Notes:

(1) There may be more actors (especially upstream of Block Builders) but they're not especially important for this policy.

(2) It's possible that a Node Operator may vertically integrate several of the above roles, but in the case of Lido Node Operators are asked to not run their own relays (as this might allow for MEV hiding or cheating).

A.1.I Configuration

A.1.II Relays

In MEV-Boost relays can be configured at a per-validator level. The expectation is that Lido Node Operators would ensure that for all validators that are being run as a part of the Lido on Ethereum protocol, the appropriate list of relays is regularly updated and correctly configured.

Currently, MEV-boost software run by the Node Operator is not able to evaluate and assess (e.g. via simulation) whether the reward in the block received from the block builder is correct, so it cannot drop "bad" or "underpaid" blocks.

As an immediate practical solution, Relays will basically need to be trusted to do this, which is in part why Flashbots aims to build a monitoring + reputation system for relays.

As a result, if using MEV-boost, Lido should adopt the following requirements with regards to use of relays by Lido Node Operators:

1. The Lido DAO, either directly or via an appointed committee/sub-group, will maintain two lists of MEV-Boost relays:
2. A "must-include list" of relays - DAO-vetted relays that are considered trustworthy, well-operated, and reliable.
3. An "allow list" of relays - DAO-approved but perhaps less well-known or battle-tested relays which can be trialed prior to inclusion in the "must-include list".
4. A "must-include list" of relays - DAO-vetted relays that are considered trustworthy, well-operated, and reliable.
5. An "allow list" of relays - DAO-approved but perhaps less well-known or battle-tested relays which can be trialed prior to inclusion in the "must-include list".
6. For validators which they run as a part of the Lido protocol, Node Operators should configure their MEV-Boost instances to connect to as many of the relays included in the "must include list" as possible, based on each Node Operator's risk and legal assessment. Additionally and optionally, they may include any of the relays included in the "allow list". Node Operators should not source blocks from relays not included in either list.
7. Both the "must include list" and the "allow list" should be maintained in such a way that they are:
8. Relatively easily amendable (in case relays should be considered for addition, e.g. via public request, or relays should be removed e.g. for bad performance), and
9. Publicly listed and easily retrievable (by the public, by block-builders, and by Node Operators).
10. Relatively easily amendable (in case relays should be considered for addition, e.g. via public request, or relays should be removed e.g. for bad performance), and
11. Publicly listed and easily retrievable (by the public, by block-builders, and by Node Operators).

As a starting point, it is highly suggested that relays considered for inclusion in these lists must be:

1. publicly available,
2. publicly listed & maintained,
3. open source,
4. transparent about what (if any) transaction or address filtering they enforce.

A.1.III Monitoring

Spec is being refined as MEV-Boost specs and Relay API are being finalized, meanwhile please refer toMEV Monitoring Design - HackMD for the general gist.