Sreeram presented this idea at the Science of Blockchain Conference in August. I am sharing it here for ongoing research and discussion. The following is copied over from [this HackMD document

](https://hackmd.io/@layr/SkBRqvdC5).

TL;DR

- MEV-Boost only allows for full-block MEV, thus removing any agency for expressing preferance on inclusion of transactions from block proposers.
- We re-introduce this agency by enabling partial-block MEV via restaking through EigenLayer. By also allowing proposers to commit to a backup block, the trust on relay for validity check is also removed.

Introduction

In MEV-Boost's current design, block proposers who want to participate in MEV-Boost have to auction off the right to make the entire block in order to capture any MEV. The key reason for this is that MEV-Boost only allows full-block building in the MEV market. This limitation stems from the fact that, with the current architecture of MEV-Boost, block proposers can only attest to the block headers while selecting the highest bid.

Having MEV-Boost provision for only full-block building leads to a restriction on the set of features it can guarantee:

1. No freewill for block proposers.

With MEV-Boost, the contents of blocks are completely determined by block builders and relays who are likely to be centralized organizations whose incentives are very different from decentralized collectives of block proposers. One way for the block proposers to exercise freewill in construction of the block involves not participating in MEV-Boost and instead constructing blocks locally. Unfortunately, this pits the economic incentives of block producers against their agency to participate in decentralized block production. As a concrete example, suppose all block builders collude to operate a censorship market, then some transactions can be censored for extortion. Whereas if the block producers were highly decentralized, such collusion attacks become very difficult to coordinate.

1. Limited slashing capability.

With full-block building, MEV-Boost features only one slashing capability – specifically when a block proposer proposes a different block than the commited one. This slashing is piggybacked on ETH2.02.0's slashing for equivocation. However, with our proposed upgrade for decentralization of MEV-Boost, there is necessity for a much richer trapestry of slashing capabilities.

We propose an incremental upgrade on MEV-Boost that employs EigenLayer for increased decentralization in block production while using a broader set of slashing primitives to secure the system.

Upgrade: Partial-block MEV-Boost via EigenLayer

MEV-Boost, in its current format, features only full-block building. The reason behind having that restriction is that MEV-Boost piggybacks on slashing in ETH2.02.0. More explicitly, when the block proposer signs the header provided to them by the relay, the proposer binds themselves to it (indicated by Step [6] in following figure). If they ever sign the header of a different block at the same height, they are slashed according to Ethereum's slashing conditions for equivocation. This binding gives the relay the assurance that the builder's block will be either proposed or the proposer must forfeit their slot entirely or the proposer will propose a a different block in which case it will be slashed according to ETH 2.02.0's slashing primitve. Since Ethereum only slashes block proposers for signing two block headers at the same height, MEV-Boost only features builders having build the entire block in order to provide a header to the proposer.

[

4115×1714 455 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/9dc9d85ad09164e1d26bfb47d9b76ce17b979a8d.jpeg)

Fig1. An overview of MEV-Boost.

However, this full-block building removes any power from the block proposers (a highly decentralized set) to express any opinion on the composition of the block. To remedy this situation, we propose partial block MEV-Boost using EigenLayer

.

Protocol description

See Fig. 2 for an illustration. The primary steps involved in this upgrade are:

1. Staking with EigenLayer.

Block proposers must have restaked their ETH with EigenLayer in order to participate in this upgraded protocol for MEV-Boost. Block proposers can either restake their stake on the beacon chain by pointing their withdrawal credentials to the EigenLayer contracts or they can have people delegate to them on Ethereum's execution layer in exchange for MEV rewards.

1. Block builder assembling partial block.

In partial block MEV-boost, block builders assemble the portion of the block they are intereseted in creating. This could be the entire-block or could be a portion of the block. We note that due to EIP-1555 it is not possible for the builders to continually build full blocks (as price escalates exponentially). This ensures that even when given full freedom, blocks will always have excess space. For the purpose of explanation, we assume block builders assemble half of a block builder_part

(we note again this will not be half of a block each time but due to 1559, this will be on an average half a block) and compute a merkle_root

of the transactions contained in this half. The builders then send these transactions along with the associated merkle_root

and bid

to the relay.

1. Centralized relay provisioning DA only.

The relay provisions data availability (DA) by storing the transactions and communicates the (merkle_root

, bid

) to the block proposer.

1. Commitments by block proposer.

The MEV-Boost protocol continues as before with the block proposer selecting the highest bid. The proposer also assembles an alternative block B_alt

of their own. The block builder then sends an attestation to the winning bid's merkle root merkle_root

concatenated to a commitment commit_B_alt

(not the header, but perhaps the transaction root) to their own block B_alt

.

1. Revealing the data.

The relay then releases the underlying transactions builder_part

of the winning bid's merkle root to the block proposer. The block proposer then assembles a new block with the released transactions in the first half and fills the last half with whatever transactions proposer_part

they desire. If the relay does not release the underlying transactions, the block proposer proposes the alternative block B_alt

they assembled.

[

4583×1737 528 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/7b51b71cd67d2fc4abdca84882e5cfe447fb113b.jpeg)

Fig2. An overview of MEV-Boost + EigenLayer.

Analysis

Because the block proposer is not signing the block header in the above protocol description, a natural question that may arise is whether the proposer could steal the transactions released from the relay and assemble a new block that steals all the builder's MEV for themselves. This is a genuine concern as the block propser is not signing on the block header and hence the protocol can't piggyback on ETH 2.02.0's slashing primitive if block proposer sign a block that steals the builder's transactions. However, as the block proposers are staked in EigenLayer, they would get slashed if they ever propose a block that did not include the transactions in builder_part

released to them by the relay (this can be proven on chain via proofs against a block's transactionRoot and the merkle_root

for builder_part

) or if they didn't propose the alternate block B_alt

they attested to in step [8] as shown in Fig. 2.

With EigenLayer, we can now impose a cryptoeconomic cost on the block propser for stealing the block builder's trasactions, thus, allowing block builders to feel comfortable. Just as before, builders can still extract MEV while following whatever regulations their jurisdictions require. More importantly, this system dissolves all economic and political tradeoffs mentioned in the introduction. Now, block proposers can still include MEV-extraction transactions from builders so they do not lose out on economic returns from MEV extraction and, since they can include the second half of their block with whatever transactions they desire, they can contribute to the censorship resistance of Ethereum

.

We note that as a side-effect, the aforementioned system completely mitigates any liveness issues that arise from MEV-Boost: for example, a) the relay may sign an invalid block or b) the relay may not make the block available to the block proposer. This has been a well-documented concern (https://writings.flashbots.net/writings/understanding-mev-boost-liveness-risks/). This problem is completely solved with the proposed approach as the block proposer can just release the block B_alt

in either of the above situations.

Due to the use of alternative block B_alt

, our proposed upgrade requires the relay to serve only data availability for Builder_half

but doesn't require them to check the validity of transactions included in Builder_half

.

We finally note that a downside of the proposed approach (the inclusion of an alternative block) is that the block proposer may fraudulently try to take-over the MEV of the builder in the next block. While we think protecting Ethereum liveness is a better tradeoff, removing the alternative block is an option in this design as well.

Discussion

Latency

Observe that, with MEV-Boost and by extension to all the proposed upgrade in this work, latency $\ell\ell$ from when a builder makes its bid to when the block proposer releases the proposed block to rest of the network is an important quantity. Given that this whole process of block building and proposing should wrap up within the heat of the consensus, it is imperative that the latency $\ell\ell$ is as minimal as possible.

Assume that the network latency is given by $\Delta\Delta$. Suppose that there is negligible latency from intermediate computations in relay and block proposers.

[

Screen Shot 2023-01-15 at 8.49.07 PM

1430×572 30.4 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/eceeeacc6181eae8b7295e9e8ef5f4e84e9f4409.png)

The numbers in [.] at the top of each term indicate the step number in Fig. 2. Observe that the latency incurred in MEV-Boost + EigenLayer is same as that in MEV-Boost.

Penalty for building invalid blocks

In above upgrade, the block builder cannot affect Ethereum's liveness by proposing invalid transactions. However, the block builder doesn't get penalized from attempting to do such malicious action.

Ideally, as a system designer, one would expect the block builder to get penalized for doing so. One way to accomplish that is by having following two additional modifications:

1. Re-staked block builders.

Block builders must have restaked their ETH with EigenLayer in order for it to participate in the upgraded MEV-Boost.

1. Fraudproofs.

Raising disputes on invalid transaction bundles via fraudproofs in interactive challenges, as done in optimistic rollups, should be in place. With this capability, if a block builder proposes an invalid bundle in Builder_part

, then the block proposer can raise a fraudproof once it retrieves Builder_part

and get the block builder penalized.

With these modifications one can achieve a completely cryptoeconomic block production without any surgery on the consensus. The only caveat with this approach is that having interactive challenge mechanism using fraudproofs lend to complexity.

Acknowledgements

We thank Dankrad Feist for suggesting the idea that the builder can build an arbitrary size of the block and the proposer can fill up the rest. This rests on the observation that 1559 ensures that builder will be unable to take up the whole block continually and there will be residual space for proposers.