MEV Driven Centralization in Ethereum: Part 2

Simon Brown

Follow

--

2

Listen

Share

This is the second part of the two part exploration of how MEV in PoS Ethereum is acting as a centralizing force on the ecosystem. The first part of this series was written before the merge, and sought to speculate as to how Ethereum might become more centralized due to the effects of MEV. The second part is written after the merge, and looks at what actually transpired in the months just after the merge, the state of the ecosystem today, and where we go from here.

All the information referenced in this post comes from various public sources, and was collected by a whole bunch of very smart, diligent and committed people. If you are interested in this topic, I would encourage you to take a look at the various data sources I have referenced.

Validator Centralization

Recall that the initial rationale behind mev-boost was to try to mitigate the economies of scale that large staking pools would bring to bear on MEV extraction, which would inevitably lead to significant centralization. Mev-boost was designed to allow ANY validator to gain access to as much MEV as was available for the block they were proposing, regardless of their resources. This would mean that a solo staker would have as much chance of obtaining MEV as a large staking pool, thereby maintaining a level playing field.

Despite the mitigating factors of mev-boost, there is still significant centralization in the validator set. Below is a snapshot of the breakdown of the validator set taken from Etherscan on the 25th November:

What jumps out is the fact that Lido has a 29.8% share of the validator set, which comes worryingly close to crossing a consensus threshold.

This is somewhat misleading though, as Lido itself is a DAO, which means it is decentralized by definition. Lido has transparent policies around the distribution of their node operators, which you can read more about it in their Department of Decentralization post [1].

If we break down Lido into it's node operators instead of just regarding it a one single entity, then the composition of the validator set looks a little different:

This still isn't great, especially considering that Coinbase and Kraken (both US registered companies) together have over 21% of the validator set. Furthermore, Danny Ryan argues that there are risks associated with any liquid staking derivative protocol gaining a share of the validator that crosses a consensus threshold, even if that protocol itself is decentralized. You can read more about it in his post The Risks of LSD — HackMD [2].

How much of this centralization is driven by MEV though? Given that mev-boost gives equal access to all MEV to all validators, then surely this centralization must be down to other factors. Well yes and no, remember that the more validators you have, the more often you have a validator selected as a proposer. Currently the probability of being selected as a proposer at least once a week is 10%:

This increases to 315% for a staking pool with 30% of the validator set, meaning you are guaranteed to get at least three blocks a week, and three times as likely to land one of those intermittent high-value MEV blocks. Check out Flashbots' excellent analysis [3] of this phenomenon.

Given that validators didn't start getting access to MEV until they were able to receive execution layer rewards (i.e. after the merge), we would assume that if MEV was the driving force behind validator centralization, it would follow that we would see an increase in centralization post-merge. By looking at the actual data, this does not seem to be the case. While the amount of ETH staked has been increasing overall over time, the relative share between the main players hasn't changed that much at all since the merge.

This would suggest that mev-boost is doing its job, and distributing MEV evenly between all the validators in the set. Of course, we could argue that the lack of change in validator share since the merge could be because of low-adoption of mev-boost, or indeed low amounts of MEV being paid to the validators. Taking a close look at both of those metrics tells us that these are not the reasons for the lack of centralization.

First of all, mev-boost adoption has grown significantly over time, to the extent that ~90% of the validator set has installed mev-boost and has registered with one or more relays, which suggests that the absence of a change in the relative share of staking pools post-merge, is not because of a low adoption of mev-boost.

Second, we can look at the validator execution layer rewards over time. The below chart shows the total execution layer rewards paid to validators, which includes MEV. This data is taken from Chainsight Analytics' excellent MEV-Boost Dashboard.

Of course the execution layer rewards consist of the priority fees of all transactions in the block as well as any MEV that is captured on top of that. In order to ascertain whether the lack of change in validator centralization post-merge is down to a low amount of MEV, we would need to look at MEV in isolation from the priority fees. Fortunately our good friends at Flashbots have done just that [4]. Their data analysis shows that MEV accounts for ~73% of all payment rewards to validators.

Putting all these metrics together paints a picture of a fairly even distribution of MEV across the entire validator set, which mitigates centralization.

How will withdrawals affect distribution of staking?

This is an interesting question. Will we see stakers moving from one staking provider to another? Currently all stakers are currently locked-in. Once withdrawals are enabled, competition between staking pools should increase. At this point, will may see a spontaneous redistribution of the share of the staking.

If and how this will happen is anyone's guess, but we may very well see many people moving based solely on APR, or we may see more market share gravitating to liquid staking derivatives. By the time withdrawals are enabled on Ethereum, we may see Distributed Validator Technology up and running, see Obol or SSV Network.

Relay Centralization

As of the 23rd November 2022, Flashbots'

Relay is processing approximately 80% of all mev-boost blocks, whereas BloXroute's

relays together process around 14% of all blocks.

This clearly shows a high level centralization in a very critical component of Ethereum's infrastructure. This is less than ideal, and the potential issues with this level of centralization was brought into sharp focus in August when the US Treasury Department (specifically OFAC) placed sanctions on the Tornado Cash dapp [5]. They essentially added the Ethereum addresses of a whole bunch of smart contracts related to the Tornado Cash dapp to their SDN list. This had a chilling effect, and caused a number of organizations to remove the dapp's UI, the GitHub code repositories, and of course, for RPC endpoint providers to censor any transactions being sent to Tornado Cash.

It's not clear if OFAC actually realized that they can't actually stop access to Tornado Cash (spoiler alert: it's still being actively used), but the placing of those contract addresses on the SDN list had the desired chilling effect. Many of the mev-boost relays started censoring Tornado transactions (rather than risk going to jail indefinitely, or paying hefy fines), and one of those relays is the Flashbots relay. This means that at the time of writing, 75% of all blocks on Ethereum are being censored.

It is important to note that this censorship does not prevent access to Tornado Cash, (or any other contract that happens to be censored), it merely delays transactions en route to Tornado Cash, by a few blocks. Justin Drake [6] refers to this as "weak censorship", as opposed to "strong censorship" whereby access is completely prevented. However, the access to Tornado Cash isn't the real issue at the moment, the issue is that Ethereum cannot claim to be a credibly neutral platform if it implements the decrees of authoritarian governments. This makes the centralization of mev-boost relays a bit of a problem.

So what's causing the centralization among mev-boost relays?

Recall that in part 1 I described the phenomenon of mev-hiding and how it would likely lead to trusted relationships between staking pools and relays? Staking pools generally prefer their node operators to connect to specific relays so that they can keep track of the amount of MEV being paid to its validators. This is to identify any potential payments to node operators that aren't passed on to the validators. This is certainly one contributing factor to relay centralization, as well as simple economies of scale, i.e. once a builder is producing valuable blocks, it's easier to just use an established and trusted relay.

Block Builder Centralization

Surprisingly, the distribution of blocks proposed to the network from third party block builders is actually fairly even. This is somewhat surprising given that I had projected that network effects would drive the emergence of only a handful of dominant

block builders, and in fact, I even went so far as to not completely rule out the emergence of one single block builder.

What we've seen instead is the emergence of a number of block builders. There still exists some centralization in that 50% of all mev-boost blocks are created by 2 builders: Flashbots and 0x69. There are roughly 8 or 9 active block builders, and a long tail of dozens of smaller builders, that together win less than 2% of all blocks proposed to the network.

This could still be improved upon, but at least it's not much worse than the centralization we had in PoW whereby there were two major mining pool operators that produced most of the blocks.

Interestingly, initially in the period after the merge, what transpired was the emergence of a single dominant block builder as I and others had predicted, but this changed

over time. The chart below shows the breakdown of the share the blocks being proposed to network by various builders over time.

As you can see, at one point, Flashbot's were building on average over 60% of the mev-boost blocks. If you overlay these numbers with the total adoption of mev-boost by the validator set, you can start to see that at one point, Flashbots were producing roughly 30% of all blocks on Ethereum. This was somewhat expected, but also quite alarming to see, because at that point, it seemed as though that trend would continue, which was a bit worrying.

The chart below shows the percentage share of mev-boost blocks proposed to the network that were built by Flashbots, represented by the blue line. The red line shows the overall adoption of mev-boost by the validator set, and the yellow line represents the percentage blocks built by Flashbots as a share of the overall adoption mev-boost.

Hopefully the trend we've seen of more builders entering the space, and winning more of a share of the blocks proposed, will continue to evolve over time. To their credit, Flashbots have decided to try to encourage this by open sourcing their block builder [7], which should make it even easier for block builders to compete.

Moving Forward — What's in Store for Ethereum?

It is clear that mev-boost is more than a piece of software, it is a critical piece of ethereum infrastructure. What Flashbots have created is something that fundamentally changes the design philosophy of the network. Going forward, Flashbots are eager that the stewardship and governance of mev-boost lies with the community.

At the start of October, Flashbotsmade a call for more involvement from the community[8], and was met with a positive response. A number of organizations have stepped forward to offer to contribute to the ongoing stewardship and development of mev-boost.

Moving forward there are decisions to be made around how to implement partial block auctions, transaction inclusion lists, new transaction types, etc. and hopefully we should see more participants engaging and contributing to the process.

There are a number of innovations in the space that aim to mitigate against the risks we have seen from the effect of MEV as a centralizing force in Ethereum, and a number of these innovations promise to be fairly successful. In fact, MEV mitigation has been given its own swimlane in the Ethereum roadmap, aptly labeled "The Scourge" . At the center of this swimlane of activity on the roadmap lies "In-Protocol PBS".

In-Protocol PBS (Proposer Builder Separation)

PBS is a design philosophy, and there are multiple approaches to achieving PBS and several reasons why it is being researched.

PBS was first proposed as an approach to mitigating against the centralizing effect of MEV. The idea was that if we outsource the MEV extraction to sophisticated specialists, and give every validator equal access to the outsourced services, it would prevent staking pools with more resources than solo stakers, from benefiting from economies of scale, and obtaining a larger and larger share of the entire validator set.

PBS also has implications for scalability — i.e. [danksharding9]. Making a danksharding megablock is not something that all validators can do. Without PBS, the number of validators on the network will reduce, and potentially significantly, therefore PBS is seen as crucial to enable danksharding.

Mev-boost can be thought of as a sort of "proto-PBS", or PBS that exists outside of the protocol. In a way, it can be seen as a pretty good test of the idea. However, the problem with this approach is that it places the mev-boost relay in a trusted position. It is trusted by both the block builder and the proposer. Without some sort of enshrined PBS, there does not exist a trustless way for proposers to be confident that the builder's block will be released and that they will receive payment, and builders have no trustless way to be confident that the MEV in their block isn't stolen.

Therefore, it seems likely that PBS will be enshrined in the protocol in some way, though as of now, it's unclear exactly how this will happen, as there are still a number of proposals for doing this.

For more information on what the current thinking is around how PBS can be implemented in protocol, refer to the ideas described in these threads:

- Proposer/block builder separation-friendly fee market designs — Economics — Ethereum Research
- Two-slot proposer/builder separation — Proof-of-Stake — Ethereum Research
- Single-slot PBS using attesters as distributed availability oracle — Proof-of-Stake — Ethereum Research

Other Directions: Partial Block Auctions

As we have seen, outsourcing the construction of the entire contents of the block to a third party block builder can lead to misaligned preferences over the content of the block, for example, builders try to avoid any potential issues with the US government and thereby censoring certain transactions.

The solution then seems to be to outsource the construction of only part

of the contents of the block, an idea which is commonly referred to as partial block auctions.

There are a number of approaches to this, but they all seem to center around one idea: where the proposer creates either a block prefix or a suffix containing transactions, and the rest of the transactions in the block come from the block builder. At the moment, we have yet to see a proposal that allows for multiple block builders to contribute transactions to a single block.

One of the approaches that seems to be gaining traction has been proposed byEigenlayer, using their restaking mechanism.

Partial Block Auctions through Restaking

The restaking idea is very simple. When a validator registers with the protocol, they deposit 32 ETH into the deposit contract, and supply their withdrawal credentials which specify an address to withdraw the validator's balance to, in the case that the validator wants to unstake.

Restaking is built on the idea of supplying a smart contract address as the withdrawal address. Once the validator unstakes, they withdraw their balance into this restaking smart contract, and must then withdraw from that smart contract in order to access their stake and validator rewards.

This allows the smart contract to impose additional slashing conditions on the validator, so that in order to access their stake + validators rewards in their entirety, they will need to have fulfilled any commitments that they made as a validator. These commitments can be anything. In fact, Eigenlayer allows various restaking contracts to be made, and refers to them as "middlewares". In this way Eigenlayer can be thought of as a "programmable slashing protocol".

One example of the sort of commitments that validators can sign up to, is partial block auctions

, i.e. validators can create part of the block themselves, and allow the other part to be created by a block builder. The validators can allow the block builder to create any size portion of the block, and propose the rest themselves.

In this setup, the mev-boost relay stores the transactions of the builder part of the block, and forwards the merkle root of the transactions to the validator. The setup as described still relies on a central trusted relay, but

validators maintain a backup block that they can propose if anything goes wrong. Furthermore, Eigenlayer can eliminate the trusted relayer entirely using another middleware, which was designed as a data availability layer.

Using this data availability middleware, builders send their portion of the block to a "data availability quorum". They do this by secret sharing their block to the nodes in the DA quorum, such that no one node can access any information about the block. The quorum nodes sign the secret share they receive and return it to the builder, who then creates a "certificate" in the form of an aggregate signature, which it includes with its bid to the proposer. The proposer selects the highest bid of all the bids that have valid certificates, and signs the header contained in that bid. The proposer then sends this signed header to the DA quorum, who releases the respective secret shares to the proposer, who can now re-assemble the builder's portion of the block.

This approach is very interesting for two reasons: it means that the proposer can build part of the block themselves, which will help mitigate against the censorship problem affecting Ethereum at the moment, and also help to decentralize the mev-boost infrastructure, eliminating the need for centralized, trusted relays.

Partial Block Auctions through PEPC

Note that there is also an exploration by Barnabé Monnot to enshrine this form ofrestaking at the protocol level [10]. This

would allow for validators to enter into any sort of general commitment with any third party, and for this to be enforced at the protocol level, by attestation committees, foregoing the need for Eigenlayers restaking contracts / middlewares. This idea has been called Protocol Enforced Proposer Commitments, or “PEPC”. The main rationale behind this approach is that, as Barnabé argues, when the protocol is no longer aware of how much validators effectively have at stake, it could eventually destabilize consensus (though I believe Eigenlayer has another idea for mitigating this, involving triggering validator exits programmatically).

The PEPC idea would facilitate partial block auctions in exactly the same way that restaking would, except it would allow the protocol to keep track of which validators had been slashed and to what extent. This of course assumes that Protocol Enforced Proposer Commitments would be more appealing to validators and third parties than restaking smart contracts.

Protocol Level Partial Block Auctions

Note that this form of partial block auction is proposed as an alternative to crLists in PBS, which I’ll dive into a little later. In this scheme, the partial block auction would be facilitated by the protocol itself. With this idea, the proposer can either provide a prefix or suffix, basically meaning the builder will provide part of the block, and the proposer will provide the rest. In his exploration [11], Vitalik talks about the trade-offs between the two approaches, such as the fact that it places an extra burden on the proposer which could hamper the progress toward eventual statelessness.

Transaction Inclusion Lists / crLists / Hybrid PBS

Transaction Inclusion Lists, Censorship Resistance Lists, or “crLists” are a method to mitigate against the censorship of transactions by block builders, without requiring the proposer to actually supply any part of the block themselves.

At a high level the idea is to allow proposers to create a list of valid transactions (i.e. valid nonce, signature, balance, maxFeePerGas etc.) that they have observed in the public mempool which are deserving of being included in a block, based on gas price.

This is not as straightforward as it might seem, and there are various different variations on the approach. All the variations on the approaches to crLists seem to converge on a central tenet that the protocol will force builders to either produce a full block, or a block that accepts the proposers inclusion list.

The rationale for this is that if a builder creates a block that does not use all available blockspace, despite the fact that there are transactions in the mempool that could be included, then can we assume that a rational builder is censoring transactions for some reason, because not including all available transactions is simply leaving money on the table, which is not rational. If the block builder produces a block that does not use all available space, then there is no reason that the block they produce should not include the transactions in the crList.

Under this scheme, a censoring builder that wants to avoid including transactions from the crList would need to fill a block up to the gas limit in order to have their block accepted by the proposer and by the network. In order to do this, they would need to fill up the block themselves with random transactions. This may well be economically viable to do for one or two blocks, but remember: under EIP-1559, once a block limit has been reached, the basefee increases, which means the builder will need to pay increasing amounts of gas for the transactions that they are using to fill the block up to it’s limit, to avoid having to include the crLists. Over time, the increase in basefee will mean that most normal transactions won’t be able to be included in the block, and this in turn makes the amount of space that the builder has to “stuff” even bigger, and exponentially more expensive.

A more likely scenario is that the block builder will just forgo producing a block until the crLists do not contain sanctioned transactions, which should re-balance the block builder landscape in favor of non-censoring builders.

The main idea, and all variations of it, all seem to rely on altruism. The various designs for crLists all make sure that it’s not expensive for proposers to create crLists, so in theory it’s not going to cost them anything, however there is also no clear incentive for them to create crLists either.

If the proposer for the current slot is responsible for making the crList for that slot, then there incentive is to create an empty list, as this will ensure that the block builders will continue to build the most profitable block. This is especially true when the main dominant block builders (who routinely build the highest value blocks) are all censoring.

Therefore, crLists are envisioned to be created for future slots. For example, the proposer for the current slot 2n, creates a crList for 2n+2. This has been referred to as “forward inclusion lists”. This way, the proposer for the current slot incurs no risk of financial disadvantage from creating a crList. This also has the nice property of being compatible with single-secret-leader-election, or SSLE, as the proposer doesn’t dox themselves by publishing a crList ahead of their slot.

This document outlines a non-exhaustive list of variations on the design of a crLists scheme: https://notes.ethereum.org/@fradamt/H1TsYRfJc [12].

Open Questions

All the ideas outlined above are being actively discussed and some of them I find to be very promising. However, one question that I have is in relation to what incentive the validator has to not just outsource 100% of the block to the builder. Are we just relying on altruism?

What incentive is there for the proposer to do any work at all in order to provide part of the block? Why would a validator go to the trouble of creating crLists? If it is because Ethereum clients implement this and do it by default, will the validators have the option of disabling it? I would imagine that node operators of some staking pools would be reluctant to publish a crList or a block prefix / suffix containing sanctioned transactions.

What about mis-aligned incentives? If the proposer is supplying part of the block, does this take away valuable block space that the builder could use to derive MEV? Even if there aren't enough transactions available to fill a block, the proposer will need to be careful to select transactions that won't conflict with a builder's bundle (i.e. starting with simple transfers).

In terms of crLists, what happens if a proposer doesn't publish a crList on the P2P network? Can this be enforced by consensus? This has strong synchrony assumptions which adds complexity, which has been touched upon in some of the original material I've linked to.

Also, how do crLists work with partial block auctions i.e. block prefixes / suffixes? If for example, a number of validators sign up to partial block auctions through restaking, what happens when crLists are implemented?

Conslusions

The above are just a few of the questions that occurred to me as I wrote this, and there exist more challenging questions posed by people with bigger brains than me. So as you can see, there are still lots of open questions and potential concerns to iron out before any specific direction or approach can be settled upon. In this regard, it could be a couple of years before we begin to see a preferred solution emerge, by which time the entire MEV landscape could be quite different.

The main conclusion I have from researching this post is that I'm more bullish on Ethereum since the merge, from seeing the wealth of ideas of innovations that have emerged to tackle the emergent centralization that we've seen. Furthermore, a number of those ideas are clearly working. We've seen a trend of progressive decentralization of various key parts of the ecosystem, and those trends look set to continue, which will put Ethereum in a far stronger and more robust position going forward, giving confidence and encouragement to those that are building tools to improve people's lives.

If you are interested in diving in bit more in PBS and censorship resistance, here are some good resources to get started with:

crLists resources:

- How much can we constrain builders without bringing back heavy burdens to proposers? — Proof-of-Stake — Ethereum Research
- Forward inclusion list — HackMD
- State of research: increasing censorship resistance of transactions under proposer/builder separation (PBS)
- Notes on Proposer-Builder Separation (PBS)
- PBS censorship-resistance alternatives — HackMD