Thanks to Quintus, Tina, Josh, Calvin and others for insight and/or review.

Cross-domain MEV pushes permissionless systems towards centralization, opening them to capture and weakening credible neutrality. Re-staking and cross-staking lets single entities order blockspace in different domains, earning MEV inaccessible to others.

In Cosmos, block building is not evenly dispersed among nodes. Withmesh security, a few nodes would control even more of their networks thanks to x-domain MEV. With EigenLayer (EL), sophistication, capital, and good infra will produce similar outcomes, possibly entrenching players and harming neutrality.

[

Screenshot 2023-03-10 at 10.38.37 AM

794×794 39.6 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/09bcca5501b631c3560c468f5ef265f9d719345f.jpeg)

The basics were laid out here, but the space for X-domain MEV goes beyond this initial example that explored cross-domain MEV with oracles. EL encourages a new type of integration within its 3-role MEV market where nodes get self-referential MEV. E.g., nodes can manipulate state prior to proposing a block, creating a major incentive for exclusive order flow and for proposers collusion.

[

Screenshot 2023-03-10 at 10.38.57 AM

1306×646 65.7 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/ef0e4acc864e5a99907146cea07ac9d832d99f92.png)

Swap out Chainlink for re-staked nodes and EL can generate exclusive order flow and encourage proposers (who may just be themselves) to collude to get unique MEV.

While mitigation mechanisms exist, most fall short. And due to delayed slashing on EL, mitigation using fallbacks (e.g., user Oracle 2 when Oracle 1 censors) aren't very powerful, meaning EL can withhold services until collusion is viable. In the following, I run through an Oracle example in depth, then offer a few brief instances of other x-domain opportunities with EL. Then, I consider what to do about these issues, considering Barnabe's Protocol-Enforced Proposer Commitments and Flashbot's SUAVE among other things. But before all that, using research from Flashbots and other MEV folks, we examine T-Shaped collusion.

Cost of Collusion vs the Incentive to Collude

Collusion between proposers and EL nodes or between EL nodes is an expense, namely it requirestrust, splitting MEV rewards and other coordination costs. This cost is greatly reduced when collusion is internal, but it still exists.

A rule put forth in the cross-domain paper fromObadia, et. al., 2021 is that when MEV earned from operating nodes on two domains is greater than the MEV from ordering a single domain, there is an incentive for node to centralize.

[

Screenshot 2023-03-10 at 10.40.46 AM

1238×660 251 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/7c6a499c995a5a4d78b310397c56ae001f42e003.png)

Source: Flashbots.

There is a non-zero incentive for proposers to collude with EL nodes. The above inequality holds for EL nodes and proposers for at least arbitrage, but likely also for liquidations and other MEV activities. Since re-staking uses zero-cost capital but extraction still requires sophistication and lots of capital, well equipped EL nodes will have lots of exclusive information and MEV to extract along with the ability to easily collude with proposers (themselves + weaker nodes).

EL encourages T-shaped integration

, where proposers are incentivized to let EL nodes build or indirectly propose most blocks and builders and searchers sign deals for exclusive order-flow thanks to information and high MEV potential transactions EL controls.

T-shaped integration is really bad. It includes problems fromexclusive order flow (while the builder and searcher market is competitive, T-shaped integration will make it less competitive as builders and searchers can dominate with a proposer cartel) and multiblock MEV.

Multiblock MEV

is MEV from when a node orders concurrent blocks. This is more profitable than building the same number of standalone blocks non concurrently. Proposers increase profit from arbitrage and sandwich attacks by excluding transactions (minus transfers) from the first block. It also lets nodes manipulate oracles and harness more of their MEV (for an explainer on multiblock MEV, check out this excellent thread by 0x94305).

This further encourages centralization, letting EL nodes earn more MEV than other nodes and incentivizing vanilla nodes to surrender block production to cartel(s). Multiblock MEV will be fostered by re-staking, possibly leading to instability and finality issues. In sum, EL nodes will benefit from exclusive order flow, cross-domain MEV, and help foster multiblock MEV, entrenching the entities behind EL nodes on Ethereum.

It’s worth noting that opportunities from multiblock MEV are underexplored. Since it will likely occur more frequently (and not randomly) once EigenLayer is live, it should be more actively explored.

Oracle Updates (in depth example)

This type of MEV is called “spike” MEV on account of irregular occurrences with large profits. Oracles have along history of cheating to win liquidations on their own protocols and it makes sense - tinkering with price updates can be profitable. Now, EL nodes will capture that MEV.

[

Screenshot 2023-03-10 at 10.44.06 AM

1328×634 130 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/b5e2b7e819a598047b05449696d6646bc2b4eb01.png)

Liquidations and arbitrage yield chunky MEV opportunities to EL nodes. So

urce: Flashbots.

An oracle run with EL, while having more sustainably than a 3rd party oracle network, could extract MEV by 1) colluding with the current block proposer / builder to land the transaction in a particular position and/or 2) withholding price updates for a few blocks.

Here’s a spec of how it could work:

[

Screenshot 2023-03-10 at 10.45.32 AM

1278×768 83.5 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/5b9cc4f30d935c2fcdfc15fde1de22a4fc46cc3e.png)

Option A:

If an EL node is also the proposer, the update can be withheld (under some reasonable bound) until the node can build locally, scooping up liquidation MEV and splitting it with EL nodes that colluded to withhold the update.

Option B:

If an EL node is not the current proposer, they can withhold the update within some bound until an EL node is the proposer (reverts to A)

Option C:

If no EL node is the proposer within a time span, the oracle nodes still get paid for inclusion with searchers, increasing their yield substantially over other nodes. EL nodes can collude with the current proposer or builder (regardless if they are re-staking) to get even more MEV, entrenching a cartel over other nodes.

Historically oracles can withhold updates for a few blocks. The greater the adoption of the oracle the greater the incentive to withhold updates.

One workaround is to fall back to another oracle in the event of censorship (dapps like MakerDAO already do this), discouraging delays. The incentive for vertical integration is still large under this construction.

One idea is to auction the update through SUAVE. Liquidation MEV would still pass to the oracle nodes though (this is a recurring theme with oracles and re-staking discussed below). So, basically, no matter how it’s implemented, EL inserts proposers into the front of the 3-role MEV market, giving them partial power over transactions with high MEV potential.

But x-domain MEV isn’t limited to oracles - possibilities of extraction via re-staking are endless.

Bridges & Other VMs

Arbitrage makes up the vast majority of MEV profits today. Cross-domain arbitrage would give EL nodes an advantage over other nodes. This could be via a bridge, a DEX built on EL nodes (an Orderbook comes to mind), or an entire other blockchain built on EL. (e.g., Solana). (As a side note, building any of the above would require altering the base layer of Ethereum quite a bit, as something like a bridge would need consensus messages in the execution client.

[

Screenshot 2023-03-10 at 10.46.30 AM

922×588 73.9 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/561923e585c7e1f26e07214fabedb985d2726704.png)

2-domain arbitrage on Ethereum and Polygon possible with EL. Source:Flashbots/Westerngate

Drawing from Unity is Strength, sophisticated nodes with capital on both ends and good infra will win most MEV. EL nodes will be particularly advantaged by the simple fact that sometimes they will dictate sequencing on both domains. Repeating history, nodes will specialize to earn more and incentivize collusion. This blockspace (if it ever built, which is a big technical if

) needs mitigation mechanisms of its own. But even with them, EL nodes would be privileged in this environment.

Sequencing

Decentralizing sequencers is a work in progress, but one idea is to use EL nodes to order L2 state. This sequencer would have exclusive information on L2 state and can collude with proposers (itself) to earn more MEV. Collusion between two distinct entities running a sequencer and a proposer separately entails more friction, so it would be less lucrative.

Since EL nodes have much lower collision costs and have self-referential MEV they are more competitive than other sequencer(s). This could lead to cartelization for exclusive MEV. And even without a cartel, self-referential MEV lets EL nodes get exclusive x-domain MEV…

Greater profit opportunities incentivizes collusion between EL nodes but also between proposers and EL nodes. With a certain level of adoption, cartelization could be minimized, but that introduces other risks from a majority of nodes re-staking.

Other MEV Opportunities

I’ll keep this short and simple since it largely lays outside my knowledge and is both highly technical and highly speculative. It’s possible EigenDA can earn MEV from things like validiums by withholding data unless bribed. While work is being done on this (S/O proof of custody) and it won’t necessarily happen (the ability to extract depends on the settlement layer’s fork-choice rule and with all the DA layers launching this year, extraction won’t be that viable) it’s still possible this sort of MEV is exclusively produce with EigenDA nodes.

Wat do?

Having talked about this centralizing force, it’s now worth asking the age old question of MEV - Wat do? Below I examine a few ideas for workarounds.

Spreading Awareness

Spreading awareness of MEV is the first step towards mitigating, capturing, and controlling externalities. To spread awareness, EL will run a dashboard showing re-stakers positions, limiting the leverage nodes have over middleware. This dashboard could also help identify opportunistic nodes acting in different roles across domains (e.g. sequencer on Optimism, oracle on Arbitrum).

Unfortunately, obscuring if you run different nodes isn’t hard, so identifying these actors and their actions will be hard if not impossible. Regardless, spreading awareness and information is the first step towards illuminating the space between the dark forests. The next step is to research and develop methods and products capable of harnessing and mitigating MEV.

Can SUAVE save us?

EL will be run by either a token controlled DAO or a committee of various crypto community members (e.g., Vitalik, Vitalik's mom, Vitalik's dad). This group will white-list projects wanting to use EL, so some MEV mitigation can happen at the front-end.

Once listed, these products will be in the wild, meaning they can only be punished by the DAO/veto group if they break specific rules specified prior to launch.

[

Screenshot 2023-03-10 at 10.48.23 AM

1304×634 182 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/8c64d4f97a91731791077fe1b5713b1c909f5e2f.png)

SUAVELayer.

If EigenLayer wants to use SUAVE or be a part of SUAVE, there is likely 1) a coordination issue between on-boarding all the different middlewares post-launch and 2) a incentive issue of getting middleware to on-board to SUAVE to basically give up some MEV for the user.

Besides these problems, as mentioned above, SUAVE doesn't solve all the externalities created by EL. While it could help with something like arbitrage or front-running, it won't stop exclusive information flow or limit nodes from harnessing MEV created from transactions like liquidations.

Limiting EigenLayer

At a certain participation rate and level of economic value, EL poses a big risk to Ethereum's health. If a middleware arises with high resource requirements that produces a lot of revenue/MEV, EL nodes could form a cartel, controlling big parts of Ethereum. It may be best for EL to limit staking to ETH, not re-staked ETH, so nodes don't get a dual role. But that will only introduce friction by not letting nodes re-use capital - cross-domain capture would still be possible, just at a higher capital cost, so this would really de-democratize access.

Another alternative would be forcing re-staked nodes to give up construction of their block completely, but collusion likely limits the effectiveness of doing this. This also forces EL yields to compete with ETH yields, something that won't happen and also weakens the biggest selling point for EL to begin with (e.g., bye bye high economic security at zero marginal cost).

One idea is to force EL nodes running services to pass off their proposer responsibilities whenever they have both roles. But again, collusion between the base and EL nodes is still incentivized [mev(a+B) > mev(a) + mev(b)] and EL still gets exclusive MEV.

Whatever the case may be, we likely can't stop building EL or limit the re-stakers from their roles. Ethereum is open and permissionless so someone will build this to be maximally expressive, letting nodes chase yields and use idle resources.

[

Screenshot 2023-03-10 at 10.49.28 AM

1326×768 92.7 KB

](https://global.discourse-cdn.com/standard14/uploads/eigenlayer/original/1X/6144c149a10427154511a8d0d0e09f423ca49ada.jpeg)

So, if EL is built to be maximally expressive and SUAVE doesn't fix every issue, what else can we do?

Re-visiting proposer commitments

Barnabe Monnot from Ethereum's Robust Incentives Group has proposed making the Ethereum protocol aware of EL node balances via protocol-enforced proposer commitments on EL. This lets Ethereum check commitments EL nodes enter into are fulfilled, regardless if the node checking also runs middleware. This would give middleware better assurances and minimize the risk EL poses to Ethereum.

While proposer commitments can minimize symptoms from MEV and limit opportunistic nodes attempting to re-org EL space and steal funds, it does make reorgs and theft from middleware much more challenging.

For example, on EigenDA, nodes lacking blops can attest to holding data without being slashed (in fact, in some scenarios they are economically rational to do so). Something like Dankrad Feist's Proofs of Custody can eliminate this so-called lazy validation. But, generally because slashing conditions can be or are delayed, nodes can act maliciously and/or siphon MEV over the delay period.

Dapp Design

Until research and new methods or products are introduced, social norms can limit MEV's effects today. But Ethereum needs neutrality credibly backed by cryptographic and economically robust systems , not neutrality backed by social consensus. The key to this, ultimately, is dapp designs with MEV in mind.

Protocols grant proposers a monopoly on creating blockspace and users initiate MEV opportunities, but current dapp designs are ultimately reasonable for most MEV. Better designs have already been introduced (e.g., Uniswap V3 generates a lot less MEV than V2), but dapps need to go further.

Ideas include internalizing MEV to the protocol (Maker's dutch auction does this, pushing keepers to compete on price instead of latency) and passing explicit MEV back to the user via something like afee escalator. In sum, we can tinker with incentives and dapp behavior endlessly, limiting MEV. Altering consensus is harder and limited. EL should hone in on on-boarding middleware building with MEV in mind and think less about the protocol it's on.

As far as internalizing MEV goes, there is an incentive for middleware to own its own revenue or MEV with pooled security, requiring re-stakers to stake a governance token like Rocketpool does for its LSD. This would put a bigger capital burden on nodes, centralizing EL's node set further, but also would mitigate some of the MEV revenue that disadvantages vanilla nodes. EL will allow this, which is fine - experimentation is good after all - but it shouldn't make it a requirement for re-staking. Rocket Pool did and now Lido is its (much) bigger brother (albeit for other reasons as well).

MEV Smoothing

While rewards won't go to vanilla nodes, EigenLayer cansmooth MEV between its entire node set, limiting the power of nodes running more intense applications. While much can be said about the ethics of this idea, it would undoubtedly limit cartelization within the protocol.

The smoothing of MEV could be especially important for transactions coming from middleware like oracles, where one node gets to influence Ethereum state. Reading prices is light-weight, meaning many nodes can participate. Smoothing MEV from the node selected to update price to all nodes would increase participation, making collusion less likely.

At higher participation rates the cartel is weakened, but importantly for intense, high resource middleware a few well equipped nodes would maintain control. To get specific, EL nodes could offer additional features like pre-confirmations, specific transaction placement in blocks and other features that base layer nodes can't compete with. In turn, this information flow and advantage could make them more effective builders and searchers, letting them extract MEV from weaker nodes even when they aren't the proposer.

Future research likely needs to focus on this specific integration opportunity between power proposers and nodes not running EL.

TL;DR

: EL nodes running middleware will be power proposers, able to harness their MEV and spot as proposer(s) to collude and grow their share of the validator set. Friction (e.g., threshold signatures), minimization, and mitigation can be used, but ultimately most rewards from MEV will go to re-stakers. Dapps on-boarded to EigenLayer need to be designed with MEV in mind from the get-go, while things like MEV smoothing needs to minimize the potential for intra-cartelization.