Sequencers and Maximal Extractable Value

FranklinDAO

Table of Contents

Abstract

We discuss the role sequencers, both centralized and decentralized, play in terms of Maximal Extractable Value (MEV). With the rise of new blockchains and L2s, the problem of MEV has only become more prominent. Each of these new ecosystems attempts to mitigate MEV in their own ways. We provide case studies on the Cosmos[1] ecosystems Sei[2] and Skip[3], Ethereum L2 Arbitrum[4], sequencer network Espresso[5], and Solana[6]. We discuss the approaches each ecosystem has taken and their implications. We find that, generally, centralized sequencers can eliminate some forms of harmful MEV (eg. frontrunning) but have a monopoly threat and single point of failure. In contrast, Byzantine Oligarchy decentralized sequencers encourage frontrunning and so either require a PBS-like mechanism[7] or an ordering constraint to prevent centralization of MEV profits. We further consider Solana's Proof of History as a way to correlate risk with computing MEV and the developing literature around Byzantine democracy.

This report assumes familiarity of blockchain infrastructure, consensus, and MEV. Consider reviewing the Appendix sections for definitions of core concepts in MEV prior to reading this report.

Introduction

Layer 2 (L2) scaling solutions, also known as rollups, were designed to alleviate high execution costs on Ethereum. They function by processing transactions outside of Ethereum (the L1) and recording condensed information about the transactions on Ethereum. These L2s often employ a unique software component called a sequencer to order and process incoming transactions before they are written to the L1. The majority of such rollups, like Arbitrum, depend on a centralized entity to perform sequencing[8]. While this simplification of the protocol allows for various benefits during the developing days of the blockchain, centralization introduces several risks, including censorship, MEV concerns, and lack of reliability. Research has been conducted in various directions to address these issues, including decentralizing sequencers, proposer-builder separation (PBS), and alternative rollup designs.

Due to the unique environment a centralized sequencer creates, the MEV opportunities that have arised in L2s are largely different from those in Ethereum L1. Thus, our research is focused on determining what impact the type of sequencer has on MEV within an ecosystem. Furthermore, we lay some basic guidelines for chains in determining what tradeoffs each type has, in hopes that it may inform the future decisions made by the L2 community.

[1] https://cosmos.network/
[2] https://www.sei.io/
[3] https://skip.money/
[4] https://arbitrum.io/
[5] https://www.espressosys.com/
[6] https://solana.com/
[7] https://notes.ethereum.org/@vbuterin/pbs_censorship_resistance
[8] https://docs.arbitrum.io/sequencer

Layer-2 Sequencer Landscape

As L2s have become more prominent, a common design pattern has emerged: centralized sequencers. While centralized sequencers are often a first step to help initially scale network throughput, they can result in a multitude of negative externalities. The most notable concerns are that centralized sequencers are single points of failure, lead to latency games, and provide monopolistic powers.

If a centralized sequencer has an outage, this results in a network-wide outage, something that has happened multiple times on networks such as Solana[9] and Arbitrum[10]. Validators are unable to get transactions included in blocks, meaning the blockchain is at a standstill until the sequencer is brought back up. Centralized sequencers are also particularly vulnerable to failures caused by hacking or government regulatory action and censorship.

Perhaps the most concerning problem with centralized sequencers is that the party that controls the sequencer effectively is able to behave as a monopolist. While this threat has not materialized in any of the blockchains that operate centralized sequencers, it is well within the strategy space of the owner of the sequencer. There are many actions the owner could implement, but the most adverse are censorship and unfair prioritization of transactions. For example, the owner could implement a policy that all transactions with a fee less than $10 are always rejected, or could always prioritize their own transactions. Generally, if we consider the owner as purely economically rational, it is in their best interest to set a minimum fee that maximizes their payoff. Of course, the social payoff for doing so is strictly unfavorable.

Moreover, due to geographic centralization, a unique MEV ecosystem arises. This, combined with low fees, results in what has been termed as “latency games”. These games are quite similar to the current high frequency trading environment as they both rely on being able to receive information quicker than other parties. We discuss the implications of centralized sequencers in more detail in the Results section. Overall, the outcome of high frequency trading results in power heavily centralized among parties that can afford specialized hardware. Extrapolating the same end result to centralized sequencers, the need to address this problem becomes clear. However, there exist many avenues for doing so, each having their own nuances.

Our work discusses potential solutions in detail, both existing and theoretical. While we do not discuss the technical implications of migrating from a centralized sequencer, we discuss various end goals and their implications. Specifically, we focus on protocol-owned decentralized sequencers, public decentralized sequencers, Solana’s Proof of History, and more nascent literature on Byzantine democracy.

[9] https://www.coindesk.com/tech/2023/02/28/solana-developers-say-reason-for-network-outage-still-unclear/
[10] https://www.coindesk.com/tech/2023/06/07/arbitrum-temporarily-stopped-processing-due-to-software-bug/

Research Problem & Objectives

Given increased demand for scalable infrastructure without compromising on fairness, this report aims to accomplish the following objectives:

1. **Evaluating Layer-2 Centralization Risk**: To rigorously assess whether centralization in L2 solutions is a primary contributing factor to negative MEV. This involves analyzing the implications of centralized sequencer designs and the resulting incentives of different actors in the MEV ecosystem.
2. **Classification and Comparison of Practical Approaches to Decentralization**: To survey and categorize existing and emergent protocols that attempt to solve the issue of L2 centralization. Each approach will be evaluated based on its effectiveness in mitigating centralization and its impact on both positive and negative MEV.
3. **Broader Analysis and Discussion on Decentralized Sequencer Trade-offs**: To scrutinize the trade-offs involved in implementing decentralized sequencers as opposed to centralized ones. This report provides additional considerations and suggestions for decentralizing sequencers as observed during the survey process.

The scope of this report is confined to a select number of widely-adopted L2 solutions and employs statistical analysis, case studies, and expert interviews in order to provide both qualitative and quantitative responses to the above objectives.

In addressing the objectives, the report aims to provide actionable insights and recommendations for developing decentralized L2 solutions that can mitigate the risks associated with centralization and negative MEV behavior.

Methodology

Research Design

This research adopted a qualitative approach, utilizing case study research design to gather in-depth insights from founders and lead engineers of various L2 and adjacent protocols. The case study method allowed for an exploration of the nuances and complexities surrounding these protocols from the perspectives of their key developers, thereby facilitating a deep understanding of each protocol's technical components, primary value, and orientation towards decentralization.

Participant Selection

Participants were purposively selected to include founders and lead engineers from various L2 and adjacent protocols. The selection was grounded on the participants' intrinsic knowledge and firsthand experience in the development and operation of the protocols. For this project, we talked to Maghnus Mareneck (CEO of Skip Protocol), Jayendra Jog (CEO of Sei Protocol), Ed Felten (Co-Founder of Arbitrum/OffChain Labs), Benedikt Bunz (Co-Founder of Espresso Systems), and Alex Stokes (Researcher at ETH Foundation). You can find some of our discussions on our YouTube here.

Data Collection

The primary data collection tool was semi-structured interviews, which were facilitated through virtual platforms to accommodate different geographical locations of the participants. Each protocol under study was represented by at least one founder or lead engineer, with whom at least one interview was conducted. The interviews were guided by a set of open-ended questions, encouraging participants to describe the following:

1. Key technical components of the protocol
2. The primary value added by the protocol to the industry
3. The protocol's orientation towards decentralization

Post-interview, further discussions were facilitated to elucidate any areas that needed more depth, or to explore emergent themes and insights not initially covered in the interview.

The interview guide was piloted and reviewed to ensure the questions elicited detailed responses and fostered open discourse. This iterative process ensured a refined data collection tool that facilitated rich, insightful responses from the participants.

To ensure the trustworthiness of the study, several strategies were employed, including member checking and peer debriefing. Member checking involved sharing the findings with the participants to ensure that their perspectives were accurately represented, while peer debriefing facilitated an external check on the research process.

Results

Centralized Sequencer

Almost all L2s within the Ethereum ecosystem have opted for a centralized sequencer. As mentioned before, there are clear negatives of doing so: the threat of monopoly and acting as a single point of failure. The centralized sequencer could behave maliciously by leveraging monopolistic powers, e.g. censor transactions below a certain fee, frontrun, backrun, etc. However, these sequencers are typically run by the organization that initially created the chain, so as long as there is trust in that organization, it is unlikely these threats will materialize. They arguably have incentive to act with integrity, as malicious acts can hamper community perception and use of the chain. The centralized sequencer is also a single point of failure; if it goes down, the whole network is down.

In addition to sequencer failure and malicious behavior, using a centralized sequencer also results in latency games. Latency games have been extensively studied in the traditional finance field, particularly around high frequency trading (HFT). Firms participating in HFT are heavily geographically centralized[11]: a firm with access to specialized hardware and geographical proximity has an advantage over its competitors. If a firm is X milliseconds closer to a source of truth than a competitor, they effectively are able to see and act X milliseconds into the future compared to the broader market. Very similar games arise within ecosystems that have centralized sequencers. MEV extractors attempt to put their machines as physically close to the sequencers as possible in order to ensure they have the lowest latency and thus are able to execute latency-sensitive HFT-like strategies. If this were to continue, it would only be a matter of time before the existing HFT monopolies enter the crypto markets.

Generally, the firms that are able to extract the most value in latency games are the ones with the closest geographical location as well as the most specialized hardware. If a blockchain continues to use a centralized sequencer, one can expect an end state where these forces act as centralizing powers, resulting in MEV being controlled by a select few capable of making large infrastructure investments. Thus, a need arises to move away from centralized sequencers.

However, a centralized sequencer does have some advantages over a decentralized sequencer. Blocktime latency is minimal and ordering constraints are easily enforced. Many L2s and rollups currently use centralized sequencers because they are relatively simple to build and avoid the need for complex consensus algorithms. Specifically, centralized sequencers can act as a temporary solution to scale a blockchain. Additionally, through ordering mechanisms like FIFO, it can prevent frontrunning and provide fast transaction confirmations.

We look towards Arbitrum as a case study. When talking to Ed Felten of Offchain Labs, he mentioned that when determining what sort of sequencer to use, latency and MEV mitigation were the main issues they wanted to address. Ed stated that users value latency highly and that they also wanted to prevent negative types of MEV such as frontrunning. As a result, a centralized sequencer seemed like the ideal choice. Latency was clearly better as there was no consensus protocol overhead. Moreover, enforcing a first-come-first-serve ordering on the sequencer also prevented frontrunning. MEV searchers no longer would be able to guarantee transaction ordering in order to frontrun orders. However, as we saw with Arbitrum, this type of system gives rise to other types of MEV such as latency games and spam attacks.

There are several directions that Arbitrum has explored to mitigate MEV, particularly latency games. Arbitrum's proposals to address the centralizing power around latency games and spam attacks have merit but are not perfect. One proposal is called Time Boost[12], which is a

[11] https://lime.co/high-frequency-trading-colocation-and-the-limits-of-the-speed-of-light/

[12] https://medium.com/offchainlabs/time-boost-a-new-transaction-ordering-policy-for-arbitrum-5b3066382d62

modification of first-come-first-serve. It allows transactions to pay a priority fee in exchange for an earlier position in the transaction ordering. This results in an auction mechanism which is intended to mitigate spam attacks. However, while this may reduce latency spam, some latency games will still remain since earlier transactions require lower bid amounts to win the race. Other studies[13] have analyzed this mechanism from a game-theoretic perspective.

Another option is geographic decentralization of the sequencer. Geographic decentralization could mitigate MEV by making latency racing more challenging by requiring a larger infrastructure investment, but does not fully eliminate the problem. However, it could also lessen the value of Time Boost due to the increased latency between decentralized sequencer nodes.

The Cost of Byzantine Oligarchy

In this section, we argue that any blockchain with a Byzantine oligarchy consensus mechanism requires proposer-builder separation in order to alleviate MEV centralization. Byzantine oligarchy protocols achieve consensus through a lottery-like system and can be thought of as a "rotating dictator". Each period, a leader is chosen by some sort of random process. This leader can then act monopolistically when creating the next block. Anything goes as long as the block is valid: they can censor pending transactions, insert new transactions, and order transactions in any way. Since there is no punishment for doing so, leaders (also referred to as block producers), are in fact *incentivized* to act maliciously and extract as much value as possible. Thus far, almost all decentralized sequencers used in practice rely on a Byzantine oligarchy.

The value that can be realized through these powers is a large portion of MEV (arguably all MEV). However, not every producer is able to realize these profits. In order to do so, a producer must be able to calculate and simulate vast amounts of state. This typically requires an incredibly high amount of compute that acts as a large barrier to entry. As a result, MEV in a blockchain with a Byzantine oligarchy based sequencer is strongly centralized among the parties that are able to afford high compute machines.

Even if producers are benevolent, a Byzantine oligarchy has other issues. If pending transactions are readily available in a public mempool, profit opportunities are also public knowledge, e.g. DEX-DEX arbitrage. Due to the discrete nature of a blockchain, this results in what has been termed as Priority Gas Auctions (PGAs). Specifically, if there is a profit opportunity, multiple parties send the same transaction attempting to capture this profit. In order to ensure their transaction runs before anyone else's, each party's best strategy is to set a fee as high as possible, capped at the profit being made from the transaction. As a result, the chain is flooded with transactions that will almost all fail, driving up transaction fee prices and preventing legitimate users from having their transactions executed.

While MEV itself is unlikely to be eliminated, the centralization of who can realize MEV can be targeted.

[13] https://flashbots.notion.site/Batching-Bidding-and-Latency-db23db03ca4745df9409289283cd4d1b

Proposer Builder Separation

These issues with Byzantine oligarchy are well studied, although the term was coined later. Notably, they were first identified by Daian et. al.[14] which resulted in the inception of Flashbots. Flashbots attempts to alleviate these problems through a separation of powers, specifically separating the power of *producing* a block from *building* a block. This has been aptly termed “Proposer Builder Separation”, or PBS.

The main idea behind PBS is to create an auction around block proposals. There are two main parties involved: “builders” and “proposers”. Builders determine the contents and ordering of a block. They submit these blocks to a trusted party along with a fee that is paid to them if their block is selected. Proposers are validator nodes in the blockchain. If they are selected to propose a block, they can ask the trusted party for the block with the highest profit (transaction fees earned from publishing the block less the fee paid to the builder). This reduces the proposer’s role to simply publishing blocks offering the highest fees.

By separating the roles of proposers and block builders, PBS encourages a more competitive and inclusive market for transaction inclusion. By mitigating the exclusive control over MEV previously held by validators, PBS democratizes the block building process and incentivizes a broader range of participants to partake in it. Profits from block publishing are now able to be distributed to any validator, effectively distributing the value of MEV across a larger validator set. Additionally, one no longer needs to be a validator to build blocks to extract MEV, allowing for parties that specialize in certain types of MEV which helps reduce the compute barrier.

Thus, if a centralized sequencer wishes to transition to any decentralized sequencer that has a Byzantine oligarchy, it must incorporate some sort of PBS-like system. This may not be necessary in the initial design, but it must be included in the end state. It is likely there are other mechanisms that distribute MEV. However, it remains an open question if these alternatives are able to distribute MEV in a more equitable manner.

Skip Protocol

Our description of PBS thus far is highly general and specific implementations of PBS vary. In practice, there are numerous considerations, such as the requirement of a trusted party and the privacy of pending transactions. Here, we discuss Skip, an MEV infrastructure on Cosmos, and the implications of some of the particular design choices made by them with the intent of informing any L2 looking to migrate to a decentralized sequencer.

Skip is building MEV infrastructure that helps blockchains leverage MEV to promote financial sustainability and good user experiences while enabling searchers to execute more sophisticated and profitable strategies. The main products Skip provides are Protocol-Owned Builder (POB), Skip Select, and ProtoRev. POB can be thought of as an upgrade to Skip Select and both are very similar to Flashbots Auction, specifically being a top of blockspace auction.

[14] https://arxiv.org/abs/1904.05234

Searchers submit bundles that capture MEV along with a bid. The winning bundle(s) is then included by the block proposer at the top of the block. However, because Skip focuses on the Cosmos ecosystem, there are some interesting differences worth discussing, namely frontrunning protection and the removal of relays. Moreover, because Cosmos and Tendermint are constantly evolving, Skip is able to provide more granular configurations than Flashbots. ProtoRev is a module deployed on Osmosis that backruns transactions at execution time and passes MEV back to the protocol, dictated by governance.

Chains using Tendermint have mempools that locally order transactions in a first come first serve manner, meaning that frontrunning MEV is already very difficult to achieve when a validator is honest. However, with the addition of Skip's infrastructure, frontrunning bundles become well within possibility. As a result, Skip provides a feature called Frontrun-protect. Specifically, Frontrun-protect is a set of rules that prevent certain classes of bundles from being accepted. Namely, any bundles that have submitter transactions sandwiching user transactions or submitter transactions before user transactions are rejected[15].

Skip also differs from Flashbots in that they have removed trusted relays from their ecosystem. This was enabled when Cosmos SDK released ABCI++ (version 0.47 and above), a change that effectively allowed arbitrary metadata to be passed along with a proposed block. As a result, other validators can verify the output of an auction winner by including something like the root of the Merkle-tree of all bids along with the proposed block. If other validators disagree with the metadata they see, they can revoke the block proposed and start a new auction. As a result of this, Skip has branded their solution as Protocol-Owned Builders, or POB. It is important to note, though, that the core mechanisms used by Skip are effectively still PBS.

The ProtoRev module[16] is an interesting attempt at mitigating MEV on Osmosis, a Cosmos chain focused on liquidity provisioning, trading, and lending. As with any AMM, swaps on Osmosis can result in backrunning opportunities. Skip partnered with Osmosis to build a module that automatically backruns any transactions within the Osmosis validation code, relying upon the PostHandler functionality Cosmos SDK provides. ProtoRev is able to calculate any three-hop arbitrage cycles or below. Of the profits made from the module, 20% is to be given to Skip in the first year, 10% in the second, and 5% thereafter. Any remaining profits are sent to a DAO governed address. According to the ProtoRev dashboard[17] roughly $100,000 worth of MEV has been captured since launch. As of the time of writing, it does not appear that the profits have been used in any way yet.

Protocol-Owned Decentralized Sequencer

One approach to decentralizing sequencers is to establish a protocol-owned sequencer network, where all sequencers for a protocol are only able to sequence for that protocol. Notably, this is how Ethereum and most layer 1 networks currently operate. This method avoids

[15] https://twitter.com/SkipProtocol/status/1602372364884328449
[16] https://osmosis.zone/blog/osmosis-protorev-by-skip-protocol-on-chain-app-directed-arbitrage
[17] https://osmosis-skip-protorev-dashboard.vercel.app/

several failure modes caused by centralized sequencers and provides an avenue for tailoring sequencers specifically to optimize performance for dedicated tasks. A notable example of this is the Sei Protocol[18].

Sei is a Cosmos based layer-1 blockchain with the goal of becoming the infrastructure for trading applications of all types, primarily swapping but also derivatives and prediction markets. Their tenet is that swapping assets is the most important application in crypto. Notably, they are backed by prominent figures in the trading space, including Jump Crypto, Flow Trading, and Delphi Digital. Sei cites what they call the "Exchange Trilemma": exchanges are only able to offer two of the following three properties: decentralization, scalability, and capital efficiency[19]. Their goal is to solve this trilemma by building a decentralized exchange focused blockchain compared to the general purpose blockchains we see today, such as Ethereum. Sei claims to be the fastest blockchain so far, with a minimum blocktime of 300ms[20]. This is achieved mainly through optimistic block generation, intelligent block propagation, and parallel processing of transactions.

Sei's consensus mechanism is similar to other Cosmos chains. Sei is built using Cosmos SDK and Tendermint Core and, most importantly, includes a custom built-in Central Limit Order Book (CLOB) module. The CLOB itself is built from the blockspace, meaning in order to submit an order, a user must broadcast a transaction stating their intention. Transaction execution on Sei consists of two steps. The first step is processing any transactions not related to the order book, e.g. sending/receiving tokens, staking, function calls on smart contracts. Orderbook related transactions are dealt with as no-ops in this step. The second step is processing the order book related transactions. Within this step, transactions are grouped by market first, then the order matching engine is run.

Because Sei is building their infrastructure from the ground up, they are able to leverage the use of Frequent Batch Auctions (FBAs) to completely mitigate certain types of intrablock negative MEV. Their implementation of FBAs allows for makers to get at least the price they listed, while takers get the average clearing price of each auction. As a result, Sei claims that frontrunning attacks are completely mitigated. FBAs are baked within the order matching logic engine itself, which is as follows for each block[21]:

1. Order cancellations are processed.
2. Limit orders are added to the CLOB but not processed.
3. Market orders are processed.

[18] https://www.sei.io/

[19] https://www.nasdaq.com/articles/exchange-trilemma%3A-the-challenge-of-balancing-scalability-decentralization-and-capita

[20] https://youtu.be/ePt7pTuTmE8

[21] https://medium.com/venture-beyond/fba-clobs-mev-and-hfts-how-seis-on-chain-orderbook-cracks-liquidity-dd247d66e90c

    a. Market orders are technically IOC orders with a worst price.
    b. Orders are processed in order from worst to best price. The average clearing price is calculated once all orders are processed, and then takers are given the average clearing price while makers get the price they listed.

4. Order book is checked for crossed orders.
    a. Repeat 3b to process crossed orders. Crossed limit orders are also filled at the average clearing price in this case.

The average clearing price is best explained through an example. Let us assume the current order book has two orders: a limit buy for 1 ETH at $100 USDC from Alice and a limit buy for 1 ETH at $110 USDC from Bob. Evelyn and Jasper now both submit a market order to sell 1 ETH at worst for $100. Alice and Bob are the makers and Evelyn and Jasper are the takers. The resulting block has filled all four orders. Alice's order is filled and she receives $100 USDC. Bob's order is filled and he receives $110 USDC. To determine Evelyn and Jasper's orders, we calculate the average clearing price. In total, 2 ETH were sold for $210 USDC, so the average clearing price is $105. Thus, both Evelyn and Jasper receive $105 USDC for the sale of their 1 ETH.

Given Sei's FBA mechanism, MEV attacks such as sandwiching are no longer possible; there is no way for a malicious party to make pure profit within a single block in the context of the on-chain CLOB[22]. Further, attacks to falsify the order matching engine itself are infeasible. Since the CLOB is a part of the state of the blockchain and the order matching function is deterministic, any deviation will be caught by the consensus protocol, resulting in slashing of that validator's stake.

Because Sei launched very recently, it is difficult to predict how effective its design will be in mitigating MEV. Moreover, there may be various opportunities for MEV that have yet to be realized. One general example of this is censorship of cancellations. Let's say a user has submitted a limit sell order and the price of the asset has moved significantly higher. The user may wish to submit a cancellation in order to prevent losing out on potential upside. However, a validator can censor this cancellation and include an order to fill the stale sell for a profit opportunity. Additionally, FBA also does not protect against multi-block MEV strategies. However, it is unclear whether or not these types of strategies result in negative externalities. By accruing risk for multiple blocks, parties executing these strategies may be providing a service to users, similar to how market makers in traditional finance are able to offer more competitive prices and instant liquidity by taking on inventory risk.

While FBA does mitigate various types of negative MEV, Sei still suffers from the potential of malicious validators censoring transactions. Various other transactions not included within the CLOB still provide opportunity for MEV through the typical actions of a validator being able to include, exclude, and reorder transactions. Thus, the separation of builder and proposer roles is

[22] https://twitter.com/jayendra_jog/status/1566948998472011777

still needed in order to reduce the power validators have. Sei is looking to take an approach similar to Flashbots Auction, a marketplace of searchers and block builders.

Public Decentralized Sequencer

Another option for blockchains is to rely on an external, shared sequencing layer. One of the leading platforms in the space is Espresso, which aims to generally solve the problem of decentralizing sequencers, focusing on L2s[23,24].

At its core, consensus is a problem of transaction ordering; once the order of transactions is agreed upon, it does not matter within what virtual machine they are executed, the end state will always be the same. Specifically, Espresso will be a PoS sequencer network that acts purely as a sequencer. Other client blockchains will submit transactions to Espresso and through Espresso's consensus protocol, these transactions will be published in blocks that the client blockchains will use as a source of truth for ordering. From here, client blockchains will now only be responsible for execution of transactions. They will simply get the ordering of transactions from Espresso and then execute the transactions and validate state transitions.

This design avoids some MEV concerns caused by centralization, but still poses several challenges. In any consensus mechanism, MEV is the result of the manipulation of the ordering, insertion, or censoring of transactions into a block. Chains that rely on a shared sequencer effectively forfeit their control over transaction ordering to the sequencing layer. Thus, in a world where various L2s are using Espresso as their sequencer, MEV is not mitigated in any way, it is simply *pushed up* from the L2 to Espresso. As a result, it becomes clear that a PBS like mechanism is still needed in order to prevent externalities that persisted on Ethereum before Flashbots arrived.

One unique aspect that Espresso introduces is the possibility of *atomic cross-chain* MEV*.* If there are multiple chains using Espresso and a PBS like mechanism is implemented, one can imagine that the bundles submitted to be built can include transactions across multiple chains. The PBS mechanism Espresso uses could ensure that the transactions included in this bundle are included within the same block, effectively allowing for searchers to efficiently look for cross-chain MEV. Moreover, the bundling of cross-chain transactions could also allow for secure bridging without the need of a centralized entity for liquidity. A user could bridge their assets by bundling two transfers together; one of which sends assets on chain A and one of which results in the user receiving assets on chain B.

[23] Interview with Benedikt Bünz, Chief Scientist at Espresso Systems
[24] https://hackmd.io/@EspressoSystems/EspressoSequencer#D-Compatibility-with-Proposer-Builder-Separation

Proof of History

An alternative consensus protocol that is worth discussing is Solana's, particularly because its design challenges the need of PBS. Solana's consensus protocol is a combination of Proof of Stake (PoS) and a novel mechanism they coined as Proof of History (PoH)[25].

We first describe the PoH mechanism at a high level[26]. The usual definition of "block" does not directly translate to Solana. Instead, Solana has *slots*[27], which is a period of time where a leader ingests transactions. Similar to Byzantine oligarchy, a PoS protocol determines a leader node who decides what transactions to include in the current slot. The transactions output during this slot are usually referred to as a block, but it is important to note that these transactions are output in *realtime*. This is achieved through PoH. The slot begins with some initial state, which is represented by its hash. Each subsequent update in the slot requires the *previous* hash as input along with the transaction to execute, which creates a pseudo verifiable delay function. The duration of a slot is a fixed number of hashes, currently 800,000 which equates to roughly 400 ms slot times. Slot leaders do not have to process a transaction every tick, they can emit a no-op by simply hashing the previous state.

Due to the sequential nature of Solana's PoH, a unique MEV ecosystem arises. In essence, unlike Ethereum, a validator trying to extract MEV does not necessarily make more profit than a naive validator. Let us assume at the start of a new slot, there is a set of pending transactions T to be output. We compare two validators, one that is naive and one that is adversarial. The naive validator simply outputs transactions in order of fee. We assume the adversarial validator will attempt to make as much profit as possible. However, note that if the adversarial validator attempts to compute MEV opportunities on the set T, it can not output any of those transactions while it is calculating, thus potentially already losing compared to the naive validator. While the adversarial validator could do something such as calculate MEV on a subset of T and output the other transactions in parallel, the fact remains that by doing any sort of MEV computation, the adversarial validator is risking on losing out on fees if the MEV opportunity is small.

Interestingly, although this MEV game appears to be "fairer" than Ethereum's ecosystem, there has been a push by JitoLabs to introduce a PBS-like system in Solana. This may be due to the fact that fees on Solana are cheap, which leads to similar spam attacks like the ones in Arbitrum. In order to address this, JitoLabs has created an MEV infrastructure similar to Flashbots. Jito effectively creates discrete blocks by consolidating transactions into what they term a block engine. From here, the block engine forwards the most profitable bundles and transactions to slot leaders. The adoption of a PBS-like system in Solana is potentially due to the fact that, while PoH appears to reduce the power of a high compute validator, vanilla PoH does not provide an avenue to distribute MEV across a larger validator set.

[25] https://solana.com/solana-whitepaper.pdf
[26] https://docs.solana.com/
[27] https://github.com/solana-labs/solana/blob/master/sdk/program/src/clock.rs

Ordering Consensus

As we have seen thus far, the main property of a blockchain that affects MEV is the sequencer. While the main axis we have considered is if the sequencer is centralized or not, we can in fact break this down more granularly. Specifically, sequencers can be either centralized or decentralized as well as either enforcing an *ordering constraint* on transactions or not. Note that a centralized sequencer without an ordering constraint does not make much practical sense, as it would result in the centralized entity constantly reordering transactions to extract MEV. Thus, there are three categories of sequencers to consider: centralized sequencers with an ordering constraint, decentralized sequencers without an ordering constraint (Byzantine oligarchy), and decentralized sequencers with an ordering constraint.

In this section, we consider the implications and challenges of implementing a decentralized sequencer with an ordering constraint. Both the private and public decentralized sequencers we have discussed do not have an ordering constraint. At the time of writing, there does not appear to be a blockchain in practice that builds consensus upon an ordering constraint. However, there has been academic work in the space. Generally, this problem is known as *Byzantine ordered consensus*, first introduced by Zhang et al[28].

In general, the motivation to incorporate consensus on an ordering constraint is to prevent negative MEV such as frontrunning. We have seen that for centralized sequencers, frontrunning becomes near impossible due to their ordering constraint. If, for example, consensus can be built on a first-in-first-out (FIFO) ordering of transactions, then the same consequence should hold for decentralized sequencers: frontrunning becomes near impossible. To illustrate the difficulty of the problem, let us assume that FIFO ordering is determined based on reported timestamps by nodes. If a majority of nodes receive transaction A before transaction B, then A should come before B in the final ordering. However, it turns out that this is impossible to achieve and can be proved using the Condorcet paradox from the social choice literature.[29]

Not all hope is lost, though. Kelker et al[30] discuss the implications of relaxing the ordering constraint to be at the block level. That is, instead of dictating that A comes before B within the block ordering, they simply require that no node can deliver A in a block after B. Zhang et al further describe and implement a mechanism coined as Pompe in which nodes effectively vote on orderings. Pompe is able to enforce that the earliest transaction that an honest node timestamps will precede the latest transaction an honest node timestamps. Zhang et al coined this type of consensus as *Byzantine democracy*, as much of the work draws from the existing social choice literature.

[28] "osdi20-zhang_yunhao_0.pdf." n.d. https://www.usenix.org/system/files/osdi20-zhang_yunhao_0.pdf.

[29] Kelkar, Mahimna, Fan Zhang, Steven Goldfeder, and Ari Juels. 2020. "Order-Fairness for Byzantine Consensus." In *Advances in Cryptology – CRYPTO 2020*, 451–80. Lecture Notes in Computer Science. Cham: Springer International Publishing.

[30] Ibid

However, in general, it is known that fully eliminating Byzantine influence is provably impossible. While this is an unfavorable result, the literature has shown that the current status quo of Byzantine oligarchy is far from optimal. Instead, if the crypto community truly values decentralization, they should move towards consensus protocols that use Byzantine democracy.

Discussion

Sequencer	Pros	Cons
Centralized Sequencer	● Simpler to build ● Can avoid frontrunning MEV via ordering constraints	● Single point of failure ● Monopolist behavior ● Latency games
Protocol-Owned Decentralized Sequencer	● No single point of failure ● Can be tailored for specific use cases, e.g. Sei order book	● Encourages MEV ● Requires PBS to distribute MEV more equitably
Public Decentralized Sequencer	● No single point of failure ● Allows atomic cross-chain execution, which may have economic benefits	● Greenfield project, technical challenges ● Cross-chain MEV becomes easily achievable
Proof of History	● MEV has clear risk tradeoffs ● Continuous transaction output vs block based output	● Low fees result in spam attacks ● Hardware bar to entry for validators is exceptionally high
Byzantine Ordered Consensus with Byzantine Democracy	● No single point of failure ● Can potentially eliminate frontrunning	● Greenfield project, technical challenges ● Theory is still being developed

Overall, we see that the mitigation of MEV can be addressed with a wide array of solutions. All the MEV mitigations we analyze are focused around atomic frontrunning, although their ideologies differ. Non-atomic (or multi-block) MEV is much more complex to analyze from a sequencer perspective. However, it does not seem to be something that yet needs addressing. In order to execute multi-block strategies, parties typically need to publicly expose themselves to risk for a period of time and the payout is not guaranteed, making them similar to traditional market making strategies, but arguably more exposed.

We found that the properties of a blockchain's sequencer dictate the type of mitigations used, specifically if the sequencer is centralized or decentralized. We posit that the most crucial aspect of a chain's consensus mechanism is in fact the ordering constraint of transactions. Centralized sequencers inherently enforce an ordering constraint (unless they act as a malicious dictator) such as first come first serve. On the other hand, of the case studies done, decentralized sequencers do not enforce any sort of ordering mechanism through consensus. We discuss the properties of both systems but also consider some other approaches, such as Solana's Proof of History and a decentralized sequencer with an ordering constraint.

Decentralized Sequencing

Decentralized sequencing can be thought of as the traditional consensus mechanisms most are familiar with. Generally speaking, some lottery-like mechanism picks a random validator to be the next block proposer. This proposer effectively acts as a dictator: they can include anything they want in the block as long as they are valid transactions. While this seemingly prevents the threat of monopoly compared to a centralized sequencer, it does not come without faults. With Byzantine oligarchy, validators are *incentivized* to extract as much value as possible when creating a block. However, block building is an extremely computationally expensive operation due to the necessity of simulating as many transactions as possible. Thus, a validator that is computationally advantaged will be able to extract much more value than others.

In response to this, auction based mechanisms such as PBS have been introduced in order to help mitigate the centralization of power among high compute validators. Flashbots Auction does this by creating a marketplace in which three distinct groups participate: searchers, builders, and proposers. At a high level, searchers submit bundles of transactions directly to builders who attempt to build the most profitable block. Builders then submit these blocks along with a fee (the auction bid) to relays. These relays then forward the block with the highest fee to the proposer, who picks the block with the highest fee to propose. While high compute validators are still able to extract a large amount of value, all other validators are able to reap comparable rewards by simply participating in the market. Moreover, the fees builders pay should approach the value they gain from the block they build due to the openness of the market. Thus, a PBS mechanism more equitably distributes MEV across the validator set.

However, in a system with PBS, frontrunning is not just allowed but *encouraged* when Byzantine oligarchy is implemented. While this may be beneficial for MEV such as backrunning, it is purely harmful for users when frontrunning is possible. In our case studies, we see that Skip and Sei attempt to prevent frontrunning, but both do not fully eliminate it.

To prevent frontrunning, Skip relies on what can be described as social slashing. Validators within Cosmos are mostly public known entities. If the community sees a validator frontrunning or doing other malicious activities, they can decide to exclude that validator. However, this process is not codified in any way, meaning it relies upon social movements. This idea of social slashing is effectively the opposite of what the Ethereum community expects; Ethereum validators are *assumed to be malicious*, extracting as much value as possible. As a result, the

Ethereum ecosystem has approached MEV from the validator level, whereas for Skip it appears that MEV is approached from the user/searcher level.

Due to this, Skip's Frontrun-protect inherently relies upon the integrity of the validators. Frontrun-protect does not appear to be validated through consensus and thus, the constraints that Skip enforces can be circumvented if validators act maliciously. It is possible that a malicious validator could disregard the frontrun-protect setting and accept frontrunning bundles. Even if other validators realized this, a malicious validator could provide a service to obfuscate frontrunning by splitting it across multiple bundles. While complete obfuscation may be impossible, the point here is that frontrunning is not completely eliminated. Furthermore, nothing prevents a validator from ordering, censoring, or adding transactions within a block in such a way that extracts MEV. Tendermint is still a Byzantine oligarchy protocol, and other validators have no way of validating the order in which the proposer receives transactions. Skip has claimed that the amount of frontrunning happening on Osmosis is next to none[31]. However, there may be additional difficult-to-detect MEV taking place, as the analysis was only done for one month and the total value on Osmosis is much smaller than that of the Ethereum ecosystem. Overall, though, the idea of retroactively punishing frontrunning does have merit in reducing negative MEV.

Sei is worth mentioning for its use of FBAs to mitigate frontrunning, although this mitigation is not easily replicable in existing chains. Sei's FBA is baked into the virtual machine itself, so something like Ethereum would not be able to implement it without drastic changes. The FBA only prevents frontrunning specifically against swaps, so frontrunning is still possible in other types of transactions. However, swaps are arguably the most crucial type of activity to protect against frontrunning. In other scenarios, though, Sei still requires PBS to prevent general frontrunning as well as network spam. Jay Jog has mentioned that Sei will most likely implement something similar to Flashbots in the near future as they approach a mainnet launch.

However, while general purpose chains like Arbitrum or Polygon may not be able to easily implement something like an FBA, there are still contract level mitigations that can be put in place to prevent frontrunning. For example, Uniswap pools allow for a slippage tolerance above or below a specified price. This puts the onus upon the developers and users, though, which is a separate issue altogether.

An interesting middle ground we consider is Solana's consensus mechanism. Proof of History creates a game where malicious validators inherently need to make a tradeoff between extracting MEV and losing out on collecting transaction fees. Yet, due to the low fees on Solana, it appears that PoH does not have the impact on MEV as intended. JitoLabs did a study[32] and found that effectively 58% of validator's compute time was spent on failed arbitrage transactions. They attribute this to the low fees charged on Solana, which allows spam attacks to be a profitable strategy. More research should be done to determine if Solana's PoH can help

[31] https://twitter.com/SkipProtocol/status/1602372364884328449
[32] https://www.jito.network/blog/solving-the-mev-problem-on-solana-a-guide-for-stakers/

mitigate MEV, particularly around if higher transaction fees could help by combating spam attacks.

Finally, we do a brief review of the literature around *Byzantine ordered consensus,* particularly how it can be implemented without the use of a Byzantine oligarchy. While there are still caveats to consider, such as transaction ordering being enforced on blocks instead of *within* blocks, it has been shown that achieving Byzantine ordered consensus is possible. Thus, while the technical implementations may be arduous, we ask that L2s looking to move away from centralized sequencers consider this as a possible solution.

Conclusion & Future Directions

In general, we categorize the case studies done into two buckets: decentralized sequencers with no ordering constraint and centralized sequencers with an ordering constraint. We see that for a blockchain with a decentralized sequencer with no ordering constraint, frontrunning is *encouraged*. In this system, without a mechanism like PBS, there is a centralizing power among high compute validators to be able to extract the most value. Using PBS, MEV can be distributed more equitably across the validator set but importantly *MEV still comes at the expense of individual users in the form of frontrunning*. Thus, it seems reasonable to claim that any chain using a Byzantine oligarchy consensus mechanism should implement a PBS-like system to at least combat centralization of MEV among high compute validators.

For a centralized sequencer with an ordering constraint, frontrunning becomes vastly more difficult to do and blocktime latency is minimal. *A centralized sequencer can effectively eliminate atomic frontrunning at the cost of a monopoly threat and single point of failure.* Moreover, if the ordering constraint is time based like Arbitrum’s first-come-first-serve, it results in latency games and geographic centralization among validators. It is not clear that moving a centralized sequencer to a decentralized sequencer is optimal. Unless the decentralized sequencer is able to enforce an ordering constraint, frontrunning would then become much easier to do. However, one may be able to argue that a decentralized sequencer combined with application level frontrunning mitigations may be a reasonable tradeoff.

As we have seen, the ordering constraint chosen by a blockchain is the most important factor in determining what types of MEV are possible. Overall, it does not appear that there is a clear winner; both approaches have different tradeoffs. Thus, when considering what type of sequencer to implement, developers need to be wary of the types of MEV they promote and/or mitigate.

We suggest that there needs to be more research done on the theoretical and practical implications of Byzantine democracy. Building consensus in a trustless environment to enforce a time based ordering constraint is known as a hard problem. Generally, a decentralized sequencer with an ordering constraint requires consensus to be built upon some global binary relation that dictates the ordering of transactions in a mempool, but alternative approaches such as voting on proposed blocks or Solana’s Proof of History may have merit as well.

Appendix

Block Building

Concepts are initially explained with reference to Ethereum's implementation, observing analogies across the world of implementations. However, we will denote differences across various blockchain implementations (Cosmos, Solana, etc.) throughout the paper.

As of the writing of this paper, transactions are standardly initiated by externally-owned accounts (those owned by non-smart contract actors), which will prompt a change in the state of the Ethereum Virtual Machine (EVM)[33]. The request for transaction execution is broadcasted, then it is added to the transaction pool (mempool) alongside other pending transactions. A transaction relies on a validator to select it for inclusion in a particular block, at which point the transaction will be verified and added to the chain.

Transactions include a number of attributes, including most intuitively a sender, recipient, signature, and value/data. Additionally, as stated in Ethereum's developer documentation[34], transactions also include attribute related to gas:

- gasLimit: maximum amount of gas units that can be consumed by the transaction
- maxPriorityFeePerGas: the maximum price of the consumed gas to be included as a tip to the validator
- maxFeePerGas: the maximum fee per unit of gas willing to be paid for the transaction (optional, and inclusive of baseFeePerGas and maxPriorityFeePerGas)

Since the London upgrade[35] (EIP-1559), blocks have been created with variable-sizes. On Ethereum, the target is 15 million gwei but may increase or decrease to adjust for network demand. The upper limit is 30 million gwei and, while it is not enforced, a lower limit technically exists since there is a minimum amount of gas required for a transaction. For a transaction to be included in a block, the offered price per gas unit must be equal to or greater than the base fee. Base fee is a function of the previous block, with a ceiling of 12.5% increase between blocks to ensure block size does not remain indefinitely high. Prior to the London Upgrade, validators would receive the total gas fee of transactions included in a block. Post-London, base fees are burned to be removed from circulation, so a priority fee or top was introduced in order to incentivize validators to include a transaction in a block.

Blocks are created by a validator selected at random and contain a strict ordering of transactions. This is because each transaction alters the state of the EVM, and therefore must be executed with respect to the state induced by the transaction just before it. Validators who

[33] https://ethereum.org/en/developers/docs/evm/
[34] https://ethereum.org/en/developers/docs/transactions/
[35] https://eips.ethereum.org/EIPS/eip-1559

order transactions (and hence build the block) are incentivized to do so in such a way that their earnings from priority gas fees is as high as possible.

Maximal Extractible Value

Overview

A comprehensive overview of different types of MEV strategies[36] is not necessary for this paper, though we provide resources below for further review. Instead, we classify certain types of MEV strategies as value additive and value extractive.

In general, MEV is value additive when it encourages a secure network of validators that will include transactions in blocks instead of generating empty blocks or repeating blocks for their own profit. Further, MEV encourages market participants (especially in decentralized finance) towards economic efficiency[37] via rapid liquidations and accurate pricing.

However, there are forms of MEV which are value extractive. Most notably, sandwich and frontrunning attacks during which externally-owned accounts with access to non-private information precede large transactions, drive up prices, and thereby induce slippage and slowed execution for the larger/other transactions. Here the MEV actor is creating an opportunity for gain at the expense of another user.

Types

Atomic Strategies: These strategies fully execute within a single transaction or bundle and generally require low capital, mainly because of flash loans. Atomic strategies can involve arbitrage, sandwich attacks, and liquidations.

- **Arbitrage**: This involves buying an asset at a lower price on one platform and selling it at a higher price on another[38]. Atomic arbitrage, thanks to smart contracts, can be executed risk-free within a single transaction, even when borrowing large sums via flash loans. This is generally considered as a positive.
- **Sandwich Attacks**: In this strategy, a malicious actor spots a pending transaction that is likely to affect the price of a token[39]. They then place a buy order before that transaction and a sell order after it, essentially "sandwiching" the original transaction to benefit from the price swing. This is considered negative MEV.
- **Liquidations**: In DeFi lending protocols, if the value of the collateral falls below a certain threshold, it can be liquidated to maintain system solvency[40]. Liquidators can use flash loans to fund the liquidation and earn a profit from the fees and collateral.

[36] https://www.blocknative.com/blog/what-is-mev
[37] https://en.wikipedia.org/wiki/Economic_efficiency
[38] https://eigenphi-1.gitbook.io/classroom/mev-types/arbitrage
[39] https://eigenphi-1.gitbook.io/classroom/mev-types/sandwich-mev
[40] https://eigenphi-1.gitbook.io/classroom/mev-types/liquidation

Non-Atomic Strategies: These strategies involve multiple chains or centralized exchanges, also known as cross-domain arbitrage, as well as statistical strategies. They require significant capital to hold a position and can't generally be executed with flash loans.

- **Centralized Exchange Arbitrage**: This involves executing simultaneous buy and sell orders of the same asset across different exchanges to profit from price differences. This strategy is capital intensive due to the absence of flash loans. These strategies, along with hybrid arbitrage described below, help to equalize prices across different exchanges.
- **Hybrids**: These strategies use both on-chain and centralized exchanges. They often involve integrating decentralized exchanges into the cross-exchange arbitrage and executing buy-sell orders with inventory held in an on-chain wallet.
- **Just-In-Time (JIT) Liquidity**: This strategy is capital intensive and involves providing liquidity for a specific transaction to earn a substantial part of the transaction fees. After the transaction, the liquidity is removed, mitigating the risks associated with providing liquidity over longer periods.

As we shall discuss in this article, one of the leading suggestions for mitigating adverse effects of MEV within the Ethereum ecosystem is Proposer-Builder Separation (PBS)[41]. This draws a distinction between block builders and validators, where block builders capture MEV opportunities and bid in order to have their block selected by a validator. The validator selects a block and receives the bid as a reward, and thereby is no longer focused on optimizing MEV income, which prevents malicious validator behavior.

Roles & Stakeholders

There are some key roles we have already identified and summarize here:

- **Builders:** builders are responsible for receiving transactions from users and searchers, and then organizing and ordering them to construct the most profitable block. These blocks are then transmitted to validators via relays. Builders are incentivized by the fees that they collect from users who want to extract MEV.
- **Searchers:** searchers are responsible for finding MEV opportunities. This involves things like analyzing blockchain data, identifying arbitrage opportunities, and predicting future price movements. Searchers often write automated programs, or bots, to extract MEV. Searchers are incentivized by the profits that they can make from extracting MEV. Searchers can help increase the market's efficiency by bundling transactions in an efficient manner.
- **Validators:** validators are responsible for including transactions in blocks. They can extract MEV by including MEV-filled transactions in their blocks. Validators are incentivized by the block rewards that they receive for including transactions in blocks.

[41] https://boost.flashbots.net/

- **Users:** users initiate state changes on public blockchains like Ethereum via transactions. These state changes generate opportunities for actors in the MEV supply chain to capture MEV generated by the transactions.

Flashbots

Flashbots[42] creates protocols and tools for various MEV players including block builders, validators, etc. to engage with block construction in a more value additive manner. The MEV-Boost protocol (described below) was designed to mitigate the unintended consequences of not yet enshrined consensus design that MEV extracts value from. We review some of their key repositories and products here as context to the solutions designed for other ecosystems below.

Flashbots Builder[43]

- **Problem**: MEV extraction is difficult and time-consuming for developers.
- **Solution**: Flashbots Builder provides a library of functions that can be used to extract MEV, as well as a dashboard that allows developers to track their MEV earnings.
- **Technical summary**: Flashbots Builder uses a private transaction relay to allow MEV participants to compete for the right to include MEV-filled transactions in blocks. This allows developers to extract MEV without the need to frontrun or sandwich other transactions.

Flashbots Builder is a powerful tool that makes it easy for developers to extract MEV. It provides a library of functions that can be used to extract MEV, as well as a dashboard that allows developers to track their MEV earnings. This makes it possible for developers to extract MEV without the need to learn about the complex mechanics of MEV extraction.

Flashbots Builder has been used by a wide range of developers, including DeFi protocols, exchanges, and gaming applications. It has helped to make MEV extraction more accessible and efficient, and it has played a major role in the growth of the MEV market.

Flashbots MEV-Boost

- **Problem**: Validators without high compute power are not able to earn as much fees as they could by including MEV-filled transactions in their blocks.
- **Solution**: Flashbots MEV-Boost allows validators to source high-MEV blocks from a competitive builder marketplace. This effectively distributes profits made from MEV across the entire validator set.
- **Technical summary**: Flashbots MEV-Boost uses a private transaction relay to allow MEV searchers to auction for the ability to capture MEV transactions in blocks. This allows validators to earn more fees by including the winning transactions in their blocks.

[42] https://www.flashbots.net/

[43] https://github.com/flashbots/builder

Flashbots MEV-Share[44]

- **Problem**: MEV extraction is often seen as a negative externality of the Ethereum blockchain because the value comes at the cost of users.
- **Solution**: Flashbots MEV-Share allows MEV participants to share their earnings with the Ethereum community. This is done by burning a portion of the MEV that is extracted through Flashbots.
- **Technical summary**: Flashbots MEV-Share uses a smart contract to track MEV earnings and distribute them to the Ethereum community. This ensures that MEV extraction is not a zero-sum game and that the Ethereum community benefits from the activity.

Flashbots MEV-Share is a program that allows MEV participants to share their earnings with the Ethereum community. This is done by burning a portion of the MEV that is extracted through Flashbots. Flashbots MEV-Share has been used by a wide range of MEV participants, and it has helped to make MEV extraction more beneficial to the Ethereum community.

Flashbots SUAVE[45]

- **Problem**: MEV extraction is currently limited by the need of a trusted private transaction relay. This relay technically has the ability to frontrun everyone.
- **Solution**: Flashbots SUAVE is a new protocol that will allow MEV participants to extract MEV without the need for a trusted relay. This will make MEV extraction more efficient and transparent.
- **Technical summary**: Flashbots SUAVE leverages Intel SGXs that replace the trusted private transaction relay. This allows MEV extraction to be done in a public and transparent way.

Flashbots SUAVE is a new protocol that will allow MEV participants to extract MEV without the need for a private transaction relay. This will make MEV extraction more efficient and transparent. Flashbots SUAVE is still under development, but it has the potential to revolutionize the way that MEV is extracted on Ethereum.

Flashbots Protect[46]

Flashbots Protect is a valuable tool for users who want to benefit from MEV extraction without having to worry about the risks. It is also a valuable tool for developers who want to make their applications more MEV-resistant.

Here are some of the benefits of using Flashbots Protect:

- **Frontrunning protection**: Your transactions will not be frontrun by other users.

[44] https://docs.flashbots.net/flashbots-protect/mev-share
[45] https://writings.flashbots.net/the-future-of-mev-is-suave/
[46] https://docs.flashbots.net/flashbots-protect/overview

- **Sandwich protection**: Your transactions will not be sandwiched by other users.
- **No failed transactions**: Your transactions will only be included in blocks if they do not include any reverts.
- **Priority in blocks**: Your transactions will be mined at the top of blocks, giving them priority.
- **Etherscan integration**: Anyone with your transaction hash can see the status of their transactions on Etherscan.

Flashbots Protect is a valuable tool for users and developers who want to benefit from MEV extraction without having to worry about the risks. It is a key part of the future of Ethereum, and it will help to ensure that the Ethereum blockchain remains secure and efficient.