Introduction to MEV

1.1 What is MEV?

Listen to "the MEV Audiobook":

Maximal extractable value ("MEV")

is defined as the value that block proposers (miners or validators) can extract from a blockchain by using their ability to order, insert and censor transactions within the blocks that they produce.

MEV's tentacles touch the entire transaction supply chain of blockchains and significantly impact their user experience, consensus mechanisms, and design considerations. Every application and protocol developer needs to consider how their design choices might expose them and their users to the extractors of value that patiently lie in wait for new opportunities.

MEV is a double-edged sword. It can be a hindrance to the adoption of blockchain applications if value extraction comes at a cost to the user experience, but also a powerful incentivization mechanism that developers can leverage as an integral part of their protocols.

In this report, we explore how the extraction of MEV has evolved into a highly complex game where (teams of) crypto natives and quant trading firms compete for the chance to capture a sliver of the value available.

We have chosen to focus on the Ethereum ecosystem in this report. We appreciate that MEV exists and needs to be addressed in other blockchain ecosystems as well. We will dive deeper into MEV outside Ethereum in our forthcoming reports.

To understand how MEV is extracted, let's start with the fundamentals. Let's consider the lifecycle of a standard Ethereum Transaction (post-Merge) to see where and how value gets extracted:

1. 1.

When a user has the intent to use some blockchain application, they need to translate this intent, say swapping ETH for USDC, to some sequence of instructions that the Ethereum Virtual Machine ("EVM") can understand. Because most users do not speak in bytecode, the user interacts with a front-end UI that captures the user's inputs (e.g., how much ETH they want to sell) and creates calldata that achieves the user's goal when executed by the EVM.

Before the application can interact with the user's account (based on the calldata created through the front end), the chain needs to know that the user has in fact authorized the interaction. The user's wallet takes the dApp calldata and allows the user to sign it cryptographically to create a transaction, which when executed fulfils the user's intent.

1. 2.

After signing, the transaction is forwarded to some validator node to be distributed in the wider peer-to-peer gossiping network for transactions. The receiving node conducts checks to confirm that the transaction is valid and, after certifying validity, adds the transaction to its collection of pending transactions, the so-called mempool. [1]

After inclusion into the mempool, transactions stay in a pending state until they are selected to be a part of the next block to be added to the blockchain by the block proposer.

On Ethereum, for each slot, this proposer is chosen randomly from the set of validators, nodes with 32 ETH staked on the network.

On Ethereum, the default way a block proposer sequences transactions in their block is by a gas auction - including the highest tip paying transactions in decreasing order. When sending a transaction, a user needs to pay a gas fee based on the computational complexity of the transaction and the current per unit cost of gas. In addition to this, they can include a priority fee (tip), to incentivize a validator to include their transaction ahead of other pending transactions.

The priority fee paid by an unsophisticated retail user might not actually be a good indicator of the "true value" of that transaction. Sometimes the true value of a transaction is much higher than what the user has promised to pay, as we will see later. This value can be captured using different MEV strategies

(section 5). Block proposers have full autonomy on what sort of blocks they add to the blockchain (not just the default ordering by priority fees paid). This includes reordering, not including/censoring and adding transactions into the blocks they submit, which enables them to extract this value.

Consider our earlier swap transaction: since there are multiple other DEX liquidity pools trading the ETH/USDC pair, our trade could create an arbitrage opportunity, where the price in the pool after our trade would be lower than in a pool on another DEX. The block proposer could include a transaction immediately following ours, which would buy some ETH from the pool we just traded with and sell on the higher-priced pool, arbitraging the difference in price.

We will dive deeper into this in the later sections, but while block proposers can extract MEV in the way illustrated above, the level of required sophistication to do so effectively means that the proposers are better off letting other sophisticated actors called searchers

(section 4.1) suggest to them how to extract value from the transactions available (what transactions to include and in which order).

This is a win-win situation, since validators get to keep a portion of the MEV profits (often a significant majority, up to 99% in atomic MEV - section 3) without investing significant amounts to strategy development, and the searchers get to constantly extract value out of transactions without staking 32 ETH and having to wait to be selected as a block proposer.

1. 3.

These searchers constantly scan the public mempool for new transactions, simulate their effect on their own instance of the EVM and, if there is a profitable strategy, create bundles of transactions, by inserting their own transactions before and/or after the target transaction, and submit them to one or more block builders

for inclusion in the next block, not directly to the proposer.

So, the validators have not only outsourced the extraction of MEV to searchers but also the construction of blocks from the searchers' bundles and other transactions in the mempool [2] to a second group of sophisticated third parties, the block builders. This is called proposer-builder separation

("PBS"

) more on that later (section 6.3).

These block builders try to assemble a block that allows them to maximize the amount they pay to the proposer of the next block (from the MEV available). Builders need to build blocks using non-overlapping bundles, meaning that the same transaction cannot be included twice in the block. Thus, the builder usually

selects the searcher whose bundle promises to pay it the most (the searcher auction, more in section 4.1).

1. 4.

The block proposer for that slot selects the most valuable block submitted to it by the block builders (via relays

, section 6.2). To do this, the proposer runs a software, MEV-boost

(section 6.4), on top of their consensus client that processes the blocks it receives and automatically submits the most valuable one it received to the network.

The commitment to submit the most valuable block forces the builders to try to maximize the amount they pay to the proposer. Otherwise, they risk not having their block proposed and losing in the block-level auction (section 4.2).

Figure 1 summarizes the current transaction flow and the actors that interact with a user transaction before it gets added to the blockchain.

The key takeaway here is that the extraction of MEV has created a competitive supply chain for user transactions, with many actors vying for a share of value. Whenever a validator running MEV-boost is proposing a block, all transactions in that block have gone through at least one competitive auction (block level auction), usually two (searcher auction) and possibly even three (order-flow auction, “OFA

”, see our report on OFAs for further reading).

It is paramount to have an understanding of this supply chain and the fundamental motivators behind the actors in it, for how we address MEV will determine whether crypto will ever be able to reach its potential as a truly decentralized and open ecosystem.

In the rest of the article, we will dive deeper into how the externalities of MEV extraction have shaped the market into what it is today, what are the downsides of the current system, and how we might see a shift from this simple “MEV supply chain” into a “transaction supply network” [3].

A Brief History of MEV

Blockchain transactions have not always been subjected to such a game. So how did we arrive at the current MEV ecosystem? In this section, we provide a brief overview of the story so far.

Block proposers (miners/validators) having the ability to extract value out of pending transactions through reordering has been a concern even before smart contracts and DeFi were a thing. The idea of miners potentially being able to use their privileged position in transaction inclusion to extract profit was first discussed in 2013.

Back then, Peter Todd, a Bitcoin Core developer, offered a bounty of 2.48 BTC for anyone who was able to show a hash collision against the SHA-1 cryptographic algorithm [4]. The proof was to be submitted on-chain, which proved to be problematic. A miner aware of the bounty could see the bounty-collecting transaction, reorganize the chain to block the transaction and thus claim the reward for themselves. While it was not called MEV then, this was one of the first known times miner’s ability to extract value out of users’ transactions was acknowledged.

For the reorganization of transactions to be worthwhile, there needs to be some sort of value to capture. The lack of more complex use cases than peer-to-peer transactions, like DEX trading or lending protocols, on native Bitcoin means that miners do not have much reason to reorganize blocks. Because of this lack of expressivity in Bitcoin transactions, there are few ways to extract MEV on Bitcoin without attacking the consensus of the entire chain (e.g, intentional reorgs). Extracting MEV by attacking consensus is an order of magnitude more difficult than extracting MEV on Ethereum. The small size of the opportunity and the difficulty of extraction have meant that MEV extraction has not garnered a similar level of attention on Bitcoin as it has on other ecosystems. [5]

While the threat of MEV was known before smart contract blockchains like Ethereum existed, the concerns around reordering attacks became much more serious after Ethereum and other general purpose blockchains started gaining traction.

In the Ethereum ecosystem, reordering attacks were first thought of in 2014 in a post on the Ethereum subreddit[6], in which an ETH researcher, pmcgoohan, wondered what was to stop a miner from frontrunning traders on marketplaces built on top of Ethereum, given the miners autonomy in block construction. There were many different suggested solutions, even by Vitalik, but no clear answer to the problem was identified.

While MEV was identified as a problem (without the term being coined yet), discussion around the topic only really started picking up steam after the first DeFi applications gained adoption and brought with them the first major sources for extractable value: liquidations (DAI in 2017) and DEX arbitrages (0x and EtherDelta in 2017, first wave of AMMs in 2018 with Bancor and Uniswap).

Early analysis of these decentralized exchanges showed ways in which users could suffer from frontrunning due to miners reordering transactions to their advantage. In the paper “The Cost of Decentralization in 0x and EtherDelta” [7], a team of researchers described the same frontrunning phenomenon as pmcgoohan did 3 years earlier. The authors estimated that in practice they could be able to extract nearly 1 million dollars a year running an arbitrage bot on these early exchanges. James Prestwich was also early to raise the issue of miners potentially extracting value from users in his post “Miners aren’t your friends" [8] from January 2018, illustrating how miners could reorder, insert and censor transactions to extract value from users.

Still, discussion around these issues did not really kick into full gear until 2019 and the Flash Boys 2.0 paper[9] by Daian et al., where the term Miner Extractable Value was first coined and formalized. In the paper, the researchers documented the practice of bots searching

for pure-profit opportunities (searchers), like arbitrages across decentralized exchanges, and raised concerns over the potential risks for Ethereum, if these bots were left unchecked. At this point, the focus was more on the consensus-level risks to Ethereum rather than on negative externalities to the user experience. The fear was that the existence of MEV could make a 51% attack profitable if a miner was able to use significant resources to reverse the chain's history and extract all the available MEV in previous blocks for themselves (so-called time-bandit attacks [10]).

This was before there was any off-chain infrastructure to settle execution rights between searchers, which meant that competition for opportunities was happening on-chain either through priority gas auctions (“PGAs

” - see section 6.1) or backrunning. This on-chain competition had a few undesirable externalities that affected not just searchers, but Ethereum users as a whole, including network and blockspace congestion (including more volatile/higher gas prices). Because there was no open market for searchers to compete for opportunities as is today, there was a trend towards vertical integration between miners and searchers. Some mining pools started cooperating with trading firms to ensure that specific searchers' strategies were included by their miners over others.

The negative externalities of handling MEV on-chain led to the founding of Flashbots[11] (section 6) in late 2020. Flashbots is a research and development organization which was formed to illuminate the dark forest of MEV, democratize the extraction of it and distribute the gains. The main concerns that Flashbots wanted to alleviate at the time were congestion of both blockspace and the networking layer due to searchers competing in PGAs (more on this later) and the aforementioned time-bandit attacks.

The first solution that Flashbots released was MEV-geth (section 6.2), an off-chain auction for MEV settlement. It allowed any miner to connect with searchers who were seeking to extract MEV.

Before Flashbots, the only stakeholders in the ecosystem were the searchers and the miners/mining pools. After MEV-geth was released, relays were added to the MEV extraction supply chain as a trustless communication channel between miners and searchers. The move to Proof of Stake brought with it fears of sophisticated staking providers causing consensus-level centralization. To address this, after the merge, Flashbots released MEV-boost to separate the roles of the block proposer and the block builder to increase specialization and avoid centralization at the consensus level (section 6.4).

Categorization & Size of the MEV Market

We have now covered the definition of MEV, the history of value extraction on blockchains and how that has lead us to the current supply chain on Ethereum. Seems like MEV is one of the most important topics for protocol and application designers, but how common is this value extraction from users really and what does it look like in practice?

MEV comes in many shapes and sizes. Thinking about how we can classify different phenotypes of MEV allows us to better understand what is creating MEV, who suffers from the negative externalities of MEV extraction and what solutions would be feasible and most effective to address them.

While there are many different ways to think about the MEV market, here we present the following categorizations of MEV:

1. 1.

EV_ordering & EV_signal

1. 2.

On-chain MEV & off-chain MEV

1. 3.

Mafia, Moloch, & Monarch EV

One way to categorize MEV, presented by Frontier Research in "A new game in town[12], is to consider what kind of information the extraction of value requires, and where this information is sourced from. Based on this, we can split MEV into EV_ordering

and EV_signal

.

The key differentiator between these two categories is whether the extraction of value can be done purely with information relating to the blockchain, its current state and pending transactions in the mempool, or whether information outside the system is required.

- In the first case, the value is called EV_ordering

because extraction happens purely by reordering, adding, or removing transactions in a bundle/within a block(s[13] and is based on the current state of the blockchain and its mempool.

- The extraction of EV_signal

, requires some information outside the blockchain (a signal

) in addition to the state of the blockchain and any pending transactions a searcher might be targeting. The most common signal is the price of a token on a centralized exchange.

One related concept that is important to understand here is that of atomicity

.

- MEV is atomic

if it consists of only one "leg" of extraction, usually a single bundle that reverts if any of the transactions required for MEV extraction fail. Because of the reversion guarantee, an atomic searcher assumes no risk. EV_ordering is an example of atomic MEV, because the value is extracted immediately in one leg, and the searcher is guaranteed profit if their bundle gets included in a block.

- On the other hand, non-atomic MEV

consists of multiple legs, each of which has to succeed for the MEV extraction to be profitable. A non-atomic searcher takes execution risk. They might take a loss if one of the legs of the extraction fails, e.g., due to competition against other searchers. Profit is only statistically likely, which is why this type of MEV is sometimes called statistical arbitrage

. The main example is CEX-DEX arbitrage more on that in the next section. EV_signal is classified as non-atomic because extraction of value requires multiple legs.

Another important distinction is that of on-chain vs. off-chain MEV.

- On-chain MEV

refers to MEV that is fully extracted on-chain. While all EV_ordering is also on-chain,

these terms are not equivalent because there are also on-chain MEV extraction strategies, where the extraction is not instant which thus fall under EV_signal.

- Off-chain MEV

refers to MEV that where extraction takes place in both an off-chain venue and an on-chain venue. This means that there is more than one leg required for extraction to be successful, and thus all off-chain MEV falls under EV_signal

(but not necessarily the other way around).

- It is significantly easier for us to identify how much MEV is extracted on-chain, because all the information about MEV bundles submitted to the chain is public on the blockchain.
- On the contrary, off-chain MEV is much harder to evaluate, because we usually do not have access to information on the off-chain leg (e.g., trades on Binance) of the extraction.

A third way we can look at MEV is to consider who gets to extract (or whether it is extracted) the MEV and why. This categorization (3EV model

) was first presented by Xinyuan Sun, in 2022.[14] The model identifies three main ways that MEV is created on blockchains:

1. 1.

Mafia EV:

the value that can be extracted due to some parties having asymmetric information about other parties' actions/intents. One example of this is sandwiching users (see next section).

1. 2.

Moloch EV:

the value that could be (but is not) extracted due to a lack of coordination in the system. As an example, non-atomic strategies usually involve some risk which reduces the amount that searchers are willing to bid for opportunities. The difference between their current bid and what they would bid maximally is lost or surrendered to the Moloch.

1. 3.

Monarch EV:

the value that accrues to the party ("monarch") that has the ultimate control over the ordering of transactions and thus the future state of the system, like validators or sequencers.

The definition above is mutually exclusive and completely exhaustive, therefore the total sum of MEV = Mafia + Moloch + Monarch.

MEV is often divided into good and bad types

, based on their externalities for users and the ecosystem as a whole.

- We can think of anything that falls under Mafia and Moloch EV as badMEV

. They arise from undesirable inefficiencies or asymmetries in our systems, which should build solutions to address. Ideally, we can turn both the Mafia and Moloch EV completely into Monarch EV. We can then build solutions that distribute the rewards from the monarch back to the users.

- On the other hand, liquidations (see section 5.3) are a canonical example of a good type

of MEV. As is common with other types of good MEV, liquidations involve incentivizing sophisticated entities to provide a valuable service

in exchange for the MEV they extract,

which would be hard to implement in protocol.[15]

Now that we have an idea of how to categorize different types of MEV, how much value extracted falls under each type?

The amount of MEV available in each block varies widely depending on the general market conditions and transactions available in the mempool. MEV opportunities are usually most abundant in high volume and volatility environments because users are often willing to trade with bad execution (to de-risk quickly) or because prices are moving quickly on off-chain exchanges. These conditions create systemic dislocations that result in arbitrage and liquidation opportunities. Examples of this include the Terra/Luna collapse, USDC depeg or FTX collapse.

It is also important to understand that MEV is defined as maximal

extractable value, which is the hypothetical maximum that can be extracted from the blocks. This is different from the actual extracted value from the system. There are many MEV opportunities that are never extracted, e.g., due to not being profitable enough or the opportunity being too niche. The true extracted value is also not fully visible to us, because we lack the methods to track off-chain or non-atomic MEV.

Because our data collection methods are limited, the numbers below represent only a subset of the true extracted value on each chain, the so-called Quantified Extracted Value

. Below we provide a lower bound for the MEV extracted on selected chains where relevant data is available for 2021 and 2022. [16]

These figures might seem relatively small at first glance. They might even make one question the mental effort that is spent on engineering solutions for MEV. We will see later in this article how the incentives to compete in this MEV extraction game can, in the worst case, result in a centralized, monopolistic ecosystem with very high barriers to entry.

This is why MEV is regarded as an existential issue to blockchains. We see this resulting from two main negative externalities:

1. 1.

Centralization

: MEV is a threat to decentralization. Due to the nature/structure of the MEV marketplace, if left unchecked, it leads to centralization at the builder/validator level. This might result in censorship, geographic centralization (& exposure to regulatory capture), and other dystopian futures that make a truly decentralized and permissionless blockchain impossible.

1. 2.

Worsened user experience

: Some MEV strategies, like sandwiching, result in worse execution for users (high slippage) or unprofitability for liquidity providers (CEX-DEX arbitrage). On certain blockchains, dominant strategy for searchers is to simply spam transactions to extract value, which increases congestion and transaction fees.

So, the real cost of MEV, which is much harder to quantify, is an opportunity cost. The value that is lost by not having a truly decentralized ecosystem and lower protocol adoption due to subpar user experience compared to centralized alternatives. [15] We must address MEV in a way that aligns the incentives of the different participants in the market in a way that ensures decentralization and democratizes access to the benefits.

The importance of addressing MEV is also validated by the continued interest of both founders and investors in building solutions addressing the negative externalities of MEV. Fundraises from companies like bloXroute, Flashbots, Eden Network, Skip & Jito among others support this.

Competition between solutions is fierce and will drive margins lower than they are today. Despite this, we believe there is significant value that can be captured. Highly technical teams can build a moat by compounding their early-mover advantage and building solutions that grow the pie of total on-chain value. Even if their sliver of the pie is smaller, a larger pie means that these teams will be able to capture higher absolute revenues.

The Current MEV Supply Chain

In an ideal world, we would not have to bear these negative externalities from MEV. We need to build solutions that align the incentives of the different stakeholders involved in value extraction with those of end-users. Who are these entities exactly, and what incentivizes them to participate in the MEV supply chain?

We briefly mentioned the main stakeholders in our earlier illustration of the transaction lifecycle on Ethereum, but let’s recap them here. The main parties involved in the current MEV supply chain are:

1. 1.

Users

: have an intention to interact with a blockchain to achieve some goal.

1. 2.

Wallet

: cryptographically signs the calldata, created by a dApp, which achieves the user’s goals and creates a transaction or signed message from them and sends it to an RPC node.

1. 3.

Searchers

: compete against each other to extract MEV with bundles of transactions or operations.

1. 4.

Block builders

: try to build the most profitable block using the searchers’ bundles and other transactions available in the mempool.

1. 5.

Relays

: a communication layer between block builders and proposers.

1. 6.

Block proposers

: validators who run MEV-boost, select and add the most profitable block to the chain.

The process of extracting MEV proceeds in the same order as the list above. Each step represents an opportunity for some sort of settlement regarding the distribution of the MEV in the original transaction. This is illustrated in the following figure:

The most important participants for the extraction of MEV (on Ethereum) are the searchers, block builders and proposers. In the following section, we do a deep dive on each of these participants.

There are also other entities that are involved in the MEV supply chain, like relays, solvers in intent-based systems, and order-flow auctions. For a deeper analysis of relays, check out: "Optimistic relays and where to find them" by Frontier Research and for the latter two, check out our post on order-flow auctions (“OFAs”).

4.1 Searchers

As discussed above, searchers scan the pending transactions visible in the mempool and compete against each other for opportunities to execute their MEV extraction strategies (section 6). This raises a few questions:

- Who are these searchers?
- Why do they participate in the extraction game?
- How exactly do they make a profit?
- How do they ensure that their strategies get executed?
- What makes a searcher competitive?

After a searcher has identified an MEV opportunity, it needs to construct a sequence of transactions, a bundle

, that exploits the opportunity and then submit the bundle to a block builder. There are most likely other searchers competing for the same opportunity. Thus, the searcher has to “bribe” the block builder with a promise of a specific amount being paid to the validator, which is larger than the base gas fee (per unit of gas) that the transaction would pay otherwise. In addition to this, the builder receiving the searcher's bundle has to win against the other builders in the builder auction (next section). Because searchers have little control over this (it relies on the rest of the transactions that the builder has access to), the best strategy for them is to send their bundles to as many builders as they trust.

A major reason why MEV is so competitive is the fact that a large share of the opportunities (specifically, EV_ordering) can be exploited risk-free. This is enabled by things like bundle reversion if a searcher fails to win an opportunity and flash loans.[18] This stands as a contrast to many other sophisticated, and potentially profitable trading strategies, where the profit is only statistically guaranteed; in atomic MEV, if you notice and win an opportunity, you are deterministically guaranteed to make money.

Theoretically, anyone with a computer, an ability to code and a willingness to find opportunities can start working on MEV. Practically, this is not what happens anymore, at least not on most competitive opportunities, which require teams of sophisticated traders/engineers. Opportunities like arbitrage, liquidations, frontrunning and sandwiching are among this “short-tail

” of MEV strategies.

Searchers that want to compete on the most profitable / in-demand forms of EV_ordering, have to be great at optimizing smart contract code. The bribe for inclusion to the validator is paid out of the MEV profit that is available from the transaction. Remember that a searcher can pay more than any other searcher as a bribe to the validator, then they will win the opportunity. Let’s say there is a MEV opportunity of $100 available. If searcher A’s strategy costs $10 in gas, and searcher B’s strategy costs $5, then searcher B will be more competitive in the auction. The lower the searcher can make the gas costs, the more headroom they will have to bribe the validator to win (in this case $95 vs. $90).

This means that specific strategies often become a winner-take-all marketplace. A searcher finds an optimization to a specific type of MEV strategy and controls the market until someone else finds an optimization that allows them to bid more than the current incumbent.

If a searcher wants to be competitive on EV_signal, they need to be great at optimizing exchange connectivity (to be the quickest to react to price changes), fee tiers, inventory risk management and potentially relationship building with block builders to guarantee inclusion (more on this in the next subsection).

As EV_ordering opportunities have become more competitive and the profit margins have been compressed, some searchers have shifted from trying to extract atomic opportunities to competing on the same strategies (like sandwiching) in a non-atomic, EV_signal, way where profit is not necessarily guaranteed.[19] When successful, risk-taking searchers are able to extract at least as much, and generally more value from the same opportunity as those that refuse to take risk. This does not necessarily mean that such searchers will be more profitable, especially if the risk they take actualizes.

The more niche (capacity/awareness) the opportunity, the more likely a solo searcher will be able to be competitive. This is because the profit opportunity is not large enough to attract sophisticated teams. As an example, some searchers will focus on alternative L1s/L2s or new on-chain applications. These opportunities, especially after launch and before MEV distributing/democratizing solutions are released, can promise relatively large profits (for a sole searcher) in a less competitive market. These include Avalanche [20] and most recently Base with friend.tech[21]. These opportunities are often referred to as long-tail MEV.

Depending on the competitiveness of the MEV opportunity, searchers are forced to pay up to 99% of the available value as a bribe to the validator.[22] Opportunities that are simple to notice, simulate, and extract on your own are crowded and the searchers rationally bid up to the point where they will still make a profit over time (considering opportunity and infrastructure costs). This is also because the current MEV auction infrastructure is built in a way that ensures that the searcher either gets the MEV profit from their bundle or the transaction reverts, with the searcher getting back their bid. This was not necessarily the case before off-chain MEV auctions were introduced, as searchers had to take into account the potential cost of losing an opportunity and still having to pay a potentially significant gas fee (priority gas auctions - see section 6.1).

Therefore, we can divide MEV into both competitive

and non-competitive

opportunities, based on whether the searcher has to bid away a significant portion of their profits to the block proposer in exchange for inclusion.

- There are two ways an MEV opportunity can be non-competitive. Either there is only one searcher that has seen the MEV creating transaction, or the transaction is visible to everyone, but there is only one searcher that has a strategy to extract MEV out of it (long-tail MEV).
- A transaction would be seen by only one searcher if they somehow have an exclusive orderflow agreement[23] with the originator of that transaction. It might even be that the MEV-creating transaction is created by the searchers themselves.
- For non-competitive MEV opportunities, the searcher only needs to pay a nominal amount, just enough to get included in the block by the validator. This can be even less than 0.01% of the MEV opportunity on offer.

There are a plethora of MEV opportunities for searchers to compete on, but how do they actually extract them?

- First the searcher has to notice an opportunity (or, search) from the pending transactions in the mempool. They run algorithms that scan the pending transactions for opportunities that they are interested in exploiting.
- Note that there is no single mempool. Each node has its own constantly updating pool of pending transactions. Nodes also propagate the valid transactions they receive to other nodes. Many searchers therefore run their own Ethereum full nodes to have access to transactions in the mempool. Others pay for mempool services (e.g., bloXroute or Blocknative) to receive mempool data at a minimal latency, or to not run their own nodes. Latency optimization can be important for searchers because it allows them to compete for opportunities that arrive just before the end of the 12-second slot.
- After noticing an interesting transaction, the searcher bot uses a simulation engine

to check whether an opportunity is profitable for them. These tools, usually run off-chain, take the targeted transaction, simulate its effect on the blockchain and determine whether they can create a bundle that extracts profit from the transaction. As with mempool connectivity, optimizing the latency of the simulation engine is key if the searcher wants to be able to react to transactions arriving near the end of a slot.

- After a searcher has simulated an opportunity, they build the bundles of transactions to be submitted to block builders with some associated bid they assume to be enough to win the opportunity.

- The on-chain games start at this point. The searchers have deployed smart contracts on Ethereum, which contain the actual code that extracts the MEV (like executing a swap on a DEX). A searcher's bundle contains a call to one of these smart contracts (say buying a token on Uniswap before a retail user) and then the transaction that the searcher wants to extract MEV from.

After this, the searcher gets included on-chain and receives the difference between the MEV generated by the opportunity and how much they ended up bidding to the validator.

4.2 Block Builders

The role of a block builder was introduced after the Merge. It is part of proposer-builder separation, which was first formalized by Vitalik in 2021. [24]

PBS was introduced as a solution to mitigate centralization risks at the consensus level after the Merge. We'll dive deeper into why it was necessary to externalize full block construction in proof-of-stake Ethereum in section 6.

As covered in the previous section, block builders collect transactions and searcher bundles to build a block that they submit to a relay, which forwards it to the block proposer in that slot.

To build the most valuable block possible, builders use different algorithms to select and order bundles/transactions. After a builder has finished building a block, it uses a relay to send the proposed block to the validators (on ETH; on other chains might communicate directly). Since transactions are constantly arriving within a single 12 second slot (the blocktime on Ethereum), builders will build and submit a block to a relay multiple times in a slot, as new order flow enables them to create more valuable blocks.

MEV-boost will select the block that pays most to the validator. A natural question that might arise from this is what makes one builder more competitive than another builder, i.e., able to pay more in their block?

- Like searching, building is also a highly specialized and competitive marketplace.

- Orderflow represents the biggest advantage that builders have over each other. If a builder has access to bundles that other builders do not and if those bundles contain more MEV than what competing builders have access to, a builder will be able to submit higher value blocks to the validators and land on chain more often.* Some builders have been able to secure exclusivity agreements with certain wallets and applications to route their transactions only to them. In exchange for this exclusive order flow, the builders pay compensation to the transaction originators.

- Searchers do not submit their bundles to every available builder, which differentiates order flow between builders. While it may seem that the optimal strategy for searchers is to send their bundles to all builders (to maximize the chance of being included in the next block), this does not happen in practice. Every searcher does not trust every builder (due to the potential for MEV stealing and other adverse execution) and some builders charge a fee for their services, which might affect the profitability of specific strategies such that they prefer not to send bundles to such builders.

- This means that many searchers will send their bundles to the block builders who are historically most successful at landing their blocks on chain.[25] These searchers do not send bundles to less successful builders unless they can be certain that their bundles will not get stolen or censored. This trust is a very defensible moat for the incumbents. New builders can only really break into the market by initially operating at a loss through block subsidies (i.e., paying above the value of the block to win the PBS auction) or by paying for exclusive order flow.

- Note that different sources of orderflow will have a varying amount of value in different blocks. This explains why the builders' respective market shares fluctuate in the short term. In the long term, builder sophistication and ability to secure relationships with order flow originators explain the gradual changes in market share.

- Some builders have been able to secure exclusivity agreements with certain wallets and applications to route their transactions only to them. In exchange for this exclusive order flow, the builders pay compensation to the transaction originators.

- Searchers do not submit their bundles to every available builder, which differentiates order flow between builders.

While it may seem that the optimal strategy for searchers is to send their bundles to all builders (to maximize the chance of being included in the next block), this does not happen in practice. Every searcher does not trust every builder (due to the potential for MEV stealing and other adverse execution) and some builders charge a fee for their services, which might affect the profitability of specific strategies such that they prefer not to send bundles to such builders.

- This means that many searchers will send their bundles to the block builders who are historically most successful at landing their blocks on chain.[25] These searchers do not send bundles to less successful builders unless they can be certain that their bundles will not get stolen or censored. This trust is a very defensible moat for the incumbents. New builders can only really break into the market by initially operating at a loss through block subsidies (i.e., paying above the value of the block to win the PBS auction) or by paying for exclusive order flow.
- Note that different sources of orderflow will have a varying amount of value in different blocks. This explains why the builders' respective market shares fluctuate in the short term. In the long term, builder sophistication and ability to secure relationships with order flow originators explain the gradual changes in market share.
- Another contributor to differentiated order flow is the fact that some builders run their own searchers. Such searcher-builders

might be willing to pay a larger share of the MEV they extract, because even if they do not make a profit with their searcher, the value of landing the next block might be enough to compensate.

- This is best exemplified by a few specific builders (like, beaverbuilder & rsync) that compete on CEX-DEX arbitrages (section 5.2) with their own integrated searchers. Research has shown [26] that when prices on CEXes change significantly within the span of a single block, a builder-searcher's chances of winning the next block increase drastically, specifically due to their ability to subsidize their blocks with their CEX-DEX arbitrage profits. Data analysis in the article showed that if the price of a liquid token changed over 2% within the 12-second block time, the chance that a CEX-DEX extracting builder-searcher won was over 96%.
- Builders also make different decisions on which bundles to include in the blocks that they submit. Some relays censor blocks that aren't compliant with US sanction regulation (OFAC), while others run different block building algorithms and are able to construct more profitable blocks that way. Some block proposers are only connected to such relays, and therefore when it is their turn, only builders that also build "regulated" blocks can win.
- Some builders have been able to secure exclusivity agreements with certain wallets and applications to route their transactions only to them. In exchange for this exclusive order flow, the builders pay compensation to the transaction originators.
- Searchers do not submit their bundles to every available builder, which differentiates order flow between builders.

While it may seem that the optimal strategy for searchers is to send their bundles to all builders (to maximize the chance of being included in the next block), this does not happen in practice. Every searcher does not trust every builder (due to the potential for MEV stealing and other adverse execution) and some builders charge a fee for their services, which might affect the profitability of specific strategies such that they prefer not to send bundles to such builders.

- This means that many searchers will send their bundles to the block builders who are historically most successful at landing their blocks on chain.[25] These searchers do not send bundles to less successful builders unless they can be certain that their bundles will not get stolen or censored. This trust is a very defensible moat for the incumbents. New builders can only really break into the market by initially operating at a loss through block subsidies (i.e., paying above the value of the block to win the PBS auction) or by paying for exclusive order flow.
- Note that different sources of orderflow will have a varying amount of value in different blocks. This explains why the builders' respective market shares fluctuate in the short term. In the long term, builder sophistication and ability to secure relationships with order flow originators explain the gradual changes in market share.

Running a block builder is not free since it comes with both significant infrastructure[27] and development costs. How do block builders generate a profit?

- Some builders operate their services as a public good, to ensure censorship resistance, that there is a neutral alternative that searchers and users can send their bundles through.

- Other builders offer the service as sort of a loss-leader as a part of a full-stack of MEV solutions that they offer to searchers, some of which would not work without a dedicated builder. Examples include transaction simulation, fast access to mempool data or co-location with infrastructure (to lower latency).

- Lastly, the searcher-builders extract MEV using their internal searchers and by capturing MEV that is left on the table by other searchers, by, for example, merging bundles and conducting bottom-of-the-block arbitrage.

- Some builders have also charged fees from searchers but this seems to have fallen out of fashion. A builder could charge a fee for its services if it produces blocks that are more valuable than the competition by at least the fee amount.

- This can be possible if the builder has access to exclusive order flow or runs their own searcher that increases their block value. As MEV opportunities can disappear quickly, some searchers are also willing to submit bundles to such builders, if this increases their chances of being included in the next block.

It is also important to consider the trustworthiness of block builders.

- It might be that a block builder can be trusted to in 99% of cases, when the MEV extracted is relatively small. In cases where the opportunity is large enough that it is equivalent to months or years of block building revenue, a builder might be tempted to steal the bundle. This is even though this would most likely end their business, as other searchers would not trust them after this.

- Even if builders do not outright steal bundles from searchers, they can still benefit from their position. This includes frontrunning transactions, outbidding competing searchers with their internal searcher (knowing the exact amount they bid), or using the information from the transactions off-chain.

- This explains why many searchers only submit to so-called “neutral builders”

, builders that do not run an integrated searcher that could create a conflict of interest with other searchers. Builders' actions are opaque, especially if they exploit their informational advantage on off-chain venues. Verifying that a “neutral” builder is truly neutral is difficult, and trust that can really only be built over time and by building relationships.

- A natural question to ask is: how can it be guaranteed that a builder loses its business after stealing a bundle? How would other searchers even find out?* The searcher whose bundle was stolen by the builder would be able to see that a different address from theirs executed the bundle that they originally submitted. After this, the searcher whose MEV was stolen would likely stop using that builder.

- If this builder were to continue stealing bundles, eventually enough searchers would stop using it such that the builder’s orderflow would be significantly worse than the other builders. After this, it would no longer be able to produce valuable enough blocks to send to the validator for inclusion on-chain. This is because it would be building blocks from a much smaller selection of bundles than other builders, which limits the likelihood that it constructs the most valuable block available.

- Consider that this is the dynamic even when no searcher leaks the information that their MEV was stolen by a builder. Since most searchers communicate with each other on channels like Discord & Twitter, the information about stealing would likely leak eventually and thus cause an even quicker collapse in the bundles submitted to the MEV stealing builder, nailing the final nail in their coffin.

- The searcher whose bundle was stolen by the builder would be able to see that a different address from theirs executed the bundle that they originally submitted. After this, the searcher whose MEV was stolen would likely stop using that builder.

- If this builder were to continue stealing bundles, eventually enough searchers would stop using it such that the builder’s orderflow would be significantly worse than the other builders. After this, it would no longer be able to produce valuable enough blocks to send to the validator for inclusion on-chain. This is because it would be building blocks from a much smaller selection of bundles than other builders, which limits the likelihood that it constructs the most valuable

block available.

- Consider that this is the dynamic even when no searcher leaks the information that their MEV was stolen by a builder. Since most searchers communicate with each other on channels like Discord & Twitter, the information about stealing would likely leak eventually and thus cause an even quicker collapse in the bundles submitted to the MEV stealing builder, nailing the final nail in their coffin.
- The searcher whose bundle was stolen by the builder would be able to see that a different address from theirs executed the bundle that they originally submitted. After this, the searcher whose MEV was stolen would likely stop using that builder.
- If this builder were to continue stealing bundles, eventually enough searchers would stop using it such that the builder’s orderflow would be significantly worse than the other builders. After this, it would no longer be able to produce valuable enough blocks to send to the validator for inclusion on-chain. This is because it would be building blocks from a much smaller selection of bundles than other builders, which limits the likelihood that it constructs the most valuable block available.

- Consider that this is the dynamic even when no searcher leaks the information that their MEV was stolen by a builder. Since most searchers communicate with each other on channels like Discord & Twitter, the information about stealing would likely leak eventually and thus cause an even quicker collapse in the bundles submitted to the MEV stealing builder, nailing the final nail in their coffin.

4.3 Block Proposers

A block proposer is a validator who is responsible for submitting valid blocks to the Ethereum blockchain. On Ethereum, a validator can be chosen to be a block proposer if they have staked at least 32 ETH to the deposit contract. Validators get randomly selected by the Ethereum consensus mechanism to become block proposers. In return for this work and for committing their stake to be slashed if they misbehave, they receive a block reward.

The more economic stake a node operator has, the more likely they are to be selected as the block proposer. All nodes with at least 32 ETH staked have an equal chance of being picked, i.e., a single node does not get picked more often if they have deposited more stake. However, operators can collect many multiples of that amount to run more than one node and therefore get picked more often.[28]

These block proposers are the final step before the MEV-extracting blocks are submitted to the blockchain. Block proposers receive different proposals for blocks from block builders, via third parties called relays, with an associated bribe to the block proposer (which is paid out of the MEV extracted) to incentivize the proposer to select that block over the others submitted to it.

Actually, the proposer only receives so-called “execution payload headers” from the relay. These contain enough information to reliably evaluate the value of a block, but not to execute it (because it does not have the transaction contents). This means that the proposer cannot just steal the MEV by submitting the block without paying the searchers (and possibly the builder). The contents of the block are only revealed after the validator has committed to the payload header, after which it can be proposed as the next block.

Before MEV-boost, block proposers were also responsible for building blocks, but now they can focus on only proposing the blocks that they get sent through the relay. A small minority, 5-10%, of blocks built are not submitted through the MEV-boost client and instead built locally, usually with the default algorithm of the validator client.

Recall from the previous section that MEV-boost running proposers always select the blocks that give them the most economic benefit, i.e., in which the amount bribed to them is maximized. The benefit of participating in the MEV ecosystem is that validators can boost the yields that they get for the ETH that they have staked above the normal block reward, simply by running the MEV-boost sidecar in addition to the other validator software. The yield that validators get is a sum of the block reward, transaction fees and, if the validator is extracting MEV, the bribe paid to the validator out of the MEV available in that block.

MEV extraction also creates a competitive marketplace between different validator operators. Some validators receive their stake through delegation from external stakers, who seek a way to earn a yield for their tokens without running their own validator node (liquid staking/staking-as-a-service). A share of the additional yield that validators receive for participating in MEV extraction can be paid out to these delegated stakers as a bonus. The validator that can pay the most yield will start accumulating more delegated stake than other validators, which forces other validators to start paying a higher yield.

This eventually leads to a situation where, even though validators earn up to 99% of the value of certain MEV opportunities as a bribe, this bribe is in turn immediately paid out to the external stakers and thus redistributed to the wider ecosystem.

There is very little that validators can theoretically[29] do to get higher MEV yields than what is available through MEV-boost. This explains the relative dominance of blocks that are built using MEV-boost, and why there are no significant node operators running custom MEV strategies/clients.

While the dominant strategy might be to run MEV-boost, some block proposers have recently started to play so-called timing games within this system. Generally, proposers request blocks (built by the block builders) from the relays they are connected to immediately at the start of their slot. By delaying this request slightly, they are able to propose a higher value block (more time => more MEV opportunities => builders' blocks are higher value).[30]

Types of MEV Strategies

MEV is a billion-dollar market, where sophisticated parties compete in a complex repeated game to get a slice of the pie. The searchers are responsible for actually identifying and extracting value from opportunities. In the following, we analyze what strategies the searchers actually use to extract MEV.

There are three main "primitive" types of bundles that the searchers use to extract MEV opportunities:

1.Frontrunning

- The searcher sees a pending transaction in the mempool and aims to build a bundle where their transaction is included before the original transaction.

2.Backrunning

- A searcher aims to insert their own MEV extracting transaction immediately after another user's transaction
- Sandwiching
- A combination of both frontrunning and backrunning, where the searcher tries to insert a transaction before and after a pending transaction in the mempool.

In the following, we describe how searchers use these three primitives in different MEV strategies to extract value from transactions.

5.1 Arbitrage

When executing an arbitrage, a searcher takes advantage of the mispricing between two different trading venues. By buying the asset on the venue where the price is lower and selling on the venue where the price is higher (or selling then buying) the searcher profits the spread minus the gas he paid to get that execution.

Arbitrage is the most common form of MEV and the most competitive one. For the on-chain MEV opportunities tracked by EigenPhi [31], arbitrages made up over 47% of the profits, 68% of transactions and 82% of the bots. And this is without considering off-chain MEV, which. depending on market volatility, can be more than half of the entire MEV opportunity.

There are two main types of arbitrage DEX arbitrage

and CEX-DEX arbitrage.

DEX arbitrage:

Most decentralized exchanges today are AMMs, which means that the market price of a specific trading pair changes in a completely deterministic manner in reaction to customer orders. When these orders are large enough, they create a difference in prices between two or more liquidity pools.

Searchers need to constantly simulate whether incoming transactions in the mempool create large enough moves in asset prices on specific exchanges to create an arbitrage opportunity. If this is the case, a searcher will try to backrun

this transaction with their own buy and sell orders.

DEX arbitrages are atomic or EV_ordering, that means that there is only one on-chain leg of the trade, within a single bundle. The searcher does not hold inventory between buying and selling, since everything is executed within the same block.

Practically, searchers save the space of tokens that they want to conduct arbitrages on in some sort of data structure, where they constantly update the prices across different venues. In the above figure, pairs are saved as a multigraph where each node represents a token and each edge between tokens represents a trading venue, with the weight of that edge representing the exchange rate. The searchers' job is to find some closed cycle (say ETH->ETH) for which the product of the weights of the edges is larger than one, meaning that you get more at the end of the loop than what you put in.

As a concrete example in figure 2, the red cycle represents an arbitrage opportunity because when you convert 1 USD to EUR to CAD back to USD you get 1.5 cents (-0.29 + 0.31 + -0.005 = 0.015, exp(0.015) = 1.015, we take the exponent because prices are the logarithms of the real prices).

This is an example of a cyclic arbitrage[32], where one trades through one or more intermediary tokens (USD->EUR-CAD->USD) before trading back to the original token.

Another common type of arbitrage is a simple arb, where the same trading pair has a different price in two venues and the trade goes through only one token say (USDC->ETH->USDC).

There are multiple algorithms that exist for this job, but an added problem is the complexity of the search space in crypto.

There are thousands of liquidity pools across multiple exchanges. If you want to be the first to extract an arbitrage opportunity, you need to be able to optimize your search space (prune the graph of unnecessary pools) so that you are able to find the (best) arbitrage opportunities in time to exploit them.

Thus, the searcher needs to find the relevant transactions in the mempool (and optimize latency if they want to react to opportunities near the end of the slot), accurately simulate their effect on rates across multiple pairs, find the most profitable route (there can be more than one profitable) and submit a bundle to the next block.

We consider the following concrete example of an arbitrage opportunity:

Here, the searcher has noticed a positive cycle between three different pools: ETH/USDC, USDC/SIL and SIL/ETH, where at the end of the three hops the searcher has turned 16 ETH into 32 ETH. Usually arbitrage opportunities yield a much smaller return because price differences are arbitraged away almost immediately, which means they do not grow very large.

It is likely that in this case, a retail user sold a massive amount of SIL in one go on the USDC/SIL pool, shifting the price so much that it created a large arbitrage opportunity.

Another important facet of on-chain atomic arbitrage is the ability to use flash loans to supply capital for the trades. This is since the DEX arbitrage is atomic and flash loans require the loan to be returned within the same block. A searcher can take out a flash loan for a specific token, use it to supply the capital for the arbitrage and take home the arbitrage profit while returning the initial capital. This means that even the most profitable type of MEV is accessible with a low capital base.

CEX-DEX arbitrage

As the name suggests, CEX-DEX arbitrage is a strategy that profits off the price difference on a trading pair between a centralized exchange and a decentralized exchange. CEX-DEX arbitrages represent a significant portion of the arbitrage opportunity on Ethereum, with some estimates indicating that it made up 60% of arbitrage revenues in Q1 2023. [33]

Because the execution of CEX-DEX requires at least two legs that are non-atomic, an on-chain and an off-chain trade, it is classified as EV_ordering. A searcher is not guaranteed to be able to execute both legs of the arbitrage. They could buy a token on a centralized exchange but be outbid by someone trading on the decentralized exchange, or the block-builder(s) that they submit to might not win the block auction. Either way, the searcher bought without closing the position, potentially exposing them to a loss.

What contributes to the significant size of the CEX-DEX arbitrage opportunity is the fact that prices on decentralized exchanges update only when blocks get added and trades are made in a pool. Base layer Ethereum finalizes blocks every 12 seconds, which means that trades happen at prices that are potentially 12 seconds behind the centralized exchange price, which updates up to millions of times a second. This means that prices are almost constantly divergent between the exchanges, creating many opportunities.

Searchers engaging in this sort of arbitrage have large amounts of capital on both a centralized exchange and a decentralized exchange. As mentioned previously, CEX-DEX arbitrageurs also need to have low latency to the centralized exchange and a block builder (most CEX-DEX arbitrage is done by block builders themselves), so that they can update their trade to reflect the price just before the block gets submitted to a proposer.

CEX-DEX arbitrage is very important to keep the on-chain prices in line with the centralized exchanges, where price discovery happens on the most liquid pairs. CEX-DEX arbitrageurs are a form of price-feed for the on-chain economy. In general, CEX-DEX arbitrages are most common for pairs that are highly liquid and where price discovery happens on centralized exchanges.

To compensate for the higher risk CEX-DEX arbitrageurs take, they do not bid their profits away as aggressively. This means that searchers can often bid significantly less than 99% of the MEV profit to the proposer (which might be required in atomic opportunities) and still win. Some estimates put these figures at 35-75% of the revenue.

While CEX-DEX arbitrage provides a valuable service in keeping on-chain prices in-line with the most liquid venues, it is extracted at a significant cost to the LPs. Whenever a CEX-DEX arbitrageur trades against a DEX, it means that LPs are either selling for too cheap or buying for too much. Either way, the LP makes a loss.

The current consensus is that LVR makes it mostly unprofitable for passive LPs to provide any liquidity on DEXes like Uniswap. The discussion around LVR and its connection to LP profitability has significantly increased in the last year. There are many DeFi protocols in development to address this issue. Uniswap’s newest version (v4) promises to feature tooling that could allow LPs to avoid or be compensated for some of their losses.

5.2 Sandwiching

The sandwich primitive takes its form as an MEV strategy most often through exploiting retail users trading on DEXes.

The searcher scans the mempool for transactions on AMMs that have a high slippage tolerance, that is trades that will still execute even if the price the user gets executed at differs significantly from the price that was displayed when the trade was submitted.

Let’s say a user wants to buy ETH with 100,000 USDC, with a slippage tolerance of 10%. Say this trade would move the price in the pool up by 6%. A searcher that sees this transaction in the mempool can build a bundle where they first buy ETH just before the user’s transaction. The searchers execute at a size that moves the price of the AMM pool up by some amount, but not enough to go over the user’s slippage tolerance and revert the transaction. Now that both the user and searcher have bought ETH, the price in the pool has moved by nearly the slippage tolerance, say 9.9%. The searcher includes a final sell transaction in their bundle, which sells at the new higher price and profits the difference between the buy and sell price minus the trading fees and the gas paid to guarantee execution.

DEXes are aware of this strategy and there are a few ways to mitigate it:

A user can set a low slippage tolerance, which limits the amount by which a price can change while a trade is being executed. This reduces sandwiching, because the searchers’ transactions mean the trader gets worse slippage, which could exceed the tolerance set by the trader. Searchers are still able to exploit opportunities, since they know the slippage tolerance in the trade and can size their own frontrunning transactions in a way that guarantees the tolerance will not be violated. Sometimes setting a low slippage tolerance is not even possible. If the pool has low liquidity and high volatility, users have to set a high slippage tolerance if they want their trade to go through. The boom in memecoin trading has created a significant opportunity for some sandwich bots, like “jaredfromsubway.eth”.[34]

Routing transactions through private RPCs like Flashbots Protect means that the trades never enter the public mempool, and thus cannot be seen/exploited by any searcher. This is an example of an exclusive orderflow that the Flashbots-run block builder has been able to secure. As these solutions get more popular, especially with the advent of OFAs, we can expect sandwich volumes to be reduced in the medium to long-term.

Sandwiching is seen as a form of bad MEV (Mafia EV

) that extracts value from retail users in the form of worse execution, which is why many design efforts have been made to mitigate it.

5.3 Liquidations

Liquidations are a type of backrunning strategy that searchers execute on on-chain lending protocols.

Lending protocols rely on external parties to manage the risk of their loans. To lend from a protocol, a user has to post collateral (usually more than what they borrowed) in some token. Borrowers are obliged to keep the collateral value above a certain ratio to the amount loaned. As an example, on Aave a borrower can borrow up to 80% of the value of the collateral in ETH.

Naturally, the value of this collateral fluctuates much like any cryptocurrencies. If the value of the collateral decreases so that the value of the borrowed assets crosses the liquidation threshold or if the value of the borrowed assets increases so that the liquidation threshold is hit, your collateral can be sold off.

Let’s say that an asset price suddenly started collapsing. Any loans that are collateralized by this asset are now potentially in danger of liquidation because the collateral is no longer enough to cover the assets that were borrowed. This presents a significant risk for the decentralized lending protocols. They need to be able to sell off the collateral in time to avoid credit losses.

The solution these protocols came up with was to allow anyone to liquidate a lending position that has crossed the liquidation threshold by calling a function. In exchange for submitting this call, the lending protocols pay either a flat fee for the service or even a share of the liquidated collateral.

Searchers compete for liquidation opportunities by tracking positions that are near the liquidation threshold. DeFi lending protocols often rely on an external oracle (like Chainlink) to provide them a price feed for their assets. These oracles update their on-chain prices by sending transactions to the chain. The searcher scans the mempool for oracle price updates that would push the price of the collateral below the liquidation threshold. As soon as the searcher notices such an opportunity, it creates a bundle where they insert a backrun transaction immediately after the update transaction.

This backrun is performed as follows (all in a single atomic

transaction): 1) the searcher either takes a flash loan (gas intensive) or already has enough of the loaned asset in their account. 2) The searcher then pays off the loan to liquidate the position and collects the remaining collateral (after the liquidation penalty is applied) + the liquidation bonus. This amount should be higher than the value of the loan that the searcher repays, otherwise there would be no incentive to liquidate. Finally, if the searcher took a flash loan, they repay it within the same transaction.

Liquidations are highly competitive as the strategy is simple to perform. This means that it is common to see liquidations paying up to 99.9% of their revenue to the validator.

5.4 Generalized Frontrunning

A simple generalized frontrunner is a bot that simulates the resulting state change of each pending transaction in the mempool. If a certain transaction (or sequence of) yields a net increase in the balance of the wallet initiating the transaction, it creates an opportunity for frontrunning.

In this case, a frontrunning bot will take the pending transaction and replace all the fields referring to the original transaction creator with the bots’ own addresses. After this, it will submit the new transaction to the chain (via mempool or otherwise) with a higher gas fee than the original transaction and potentially anyone else that is trying to frontrun the transaction, to guarantee its transaction is the one that gets executed.

As a toy example, consider the Ethereum version of the Bitcoin bounty by Peter Todd from our history section. Say there is some crypto influencer who has announced a puzzle on Twitter. The solution to this puzzle must be submitted to a smart contract as a transaction with some message. The smart contract will return the first person to send the solution 1 ETH. If this solution transaction is submitted through the public mempool, a generalized frontrunning bot would be able to simulate its effect and notice that it returns 1 ETH to the sender of the transaction. A frontrunning bot would then simply copy the transaction, submit it with a higher gas price, and without having any knowledge about the puzzle or its solution, be able to claim the prize. The solver could have avoided being frontrun, if they had submitted their transaction through a private RPC endpoint, like Flashbots Protect [35]

While the premise seems simple (simulate all the incoming transactions and frontrun those that generate a profit according to the simulation), in practice these bots need to run sophisticated checks to ensure they are not being exploited. There are attacks that are able to make a simple generalized frontrunning bot think that a frontrunning opportunity is available, when, in reality, the bot might lose most of the capital it deploys in the transaction.[36]

Sometimes hackers exploiting DeFi protocols fall victim to these generalized frontrunning bots, because they are not aware of their existence, or are otherwise careless. A recent example of this that also showcases how advanced these bots can become is a recent exploit of a protocol called Roe Finance.[37]

The exploiter created an exploit contract and set up a chain of transactions stretching 50 blocks that set up the protocol to be exploited. Then the exploiter submits the work() call, which fails, because a frontrunning bot recreated the exploit and executed a frontrun.

The generalized frontrunner bot was able to simulate the cumulative effect of the transaction chain above. With the exploiters work call still pending in the public mempool, it realized that the final work call resulted in the sender’s address increasing by $80k. Thus the bot copied every transaction from the exploiter (in the last 50 blocks, including the pending work call) and submitted them in the same order. The bot did this just before the exploiters last transaction, and walked away with the $80k instead. Because the exploit had already been executed, the final transaction from the exploiter fails.

While it is difficult to establish whether the bot creator expected it to frontrun an exploit, this frontrunning sequence of transactions could also be said to be an accidental exploit. The bot only evaluates whether a certain transaction or sequence of transactions generates a pure profit opportunity, not whether that sequence of transactions is legal or not. This adds another level of complexity for those wishing to run front-running bots. This example also showcases a lack of sophistication from the exploiter's end as they too could have used a private RPC to hide their transaction from the public mempool.

5.5 Cross-Domain MEV

Cross-domain MEV refers to the extraction of MEV between two or more different domains. Practically this can mean either between a blockchain and an off-chain venue (e.g., CEX-DEX arbitrage), or another blockchain (cross-chain MEV

).

The strategies themselves are not fundamentally different across chains but mainly relate to arbitrage at the moment. It might be that in the future, there will be other MEV strategies that could be exploited across chains.

There are some complications that arise from trying to extract MEV in a non-atomic way across different domains. As with any non-atomic MEV extraction, there is no guarantee that both legs of the MEV extraction land on-chain. In addition to this, the searcher must either have inventory across multiple chains or bridge funds over while executing the strategy, which both increase the risk surface.

Below is an example of cross-domain MEV extraction. A searcher noticed that by swapping from USDT to WETH on Polygon, and then bridging the funds over to ETH, it was possible to extract a profit of 289 USD. Note that if the searcher has ETH on the main Ethereum chain, they can run this strategy without bridging funds over.

The debate around cross-chain

MEV has been a hot topic for some time. The negative externalities from its extraction were even mentioned as a motivating factor for Flashbots pursuing the creation of SUAVE (more on that in the next section).

It was feared that the extraction of cross-chain MEV would require running validators across multiple chains, so that a proposer could guarantee atomic extraction (ensuring they include both legs of the strategy in the blocks they propose). This would cause centralization pressure on the validator set, as only specific entities would be sophisticated enough to run these cross-chain strategies or even have strong enough hardware to run validators on multiple chains.

However, recent criticism has questioned whether it truly exists as described and poses a realistic threat. While MEV might be possible to extract between two chains, it seems more likely that any sort of cross-chain arbitrages will just be settled via a CEX for both chains, as the CEX price will likely always update quicker.

Flashbots & MEV on Ethereum

Many ecosystem researchers thought that, if left unaddressed, the way MEV was extracted in the early days (2017-2020), by large mining pools collaborating opaquely with specific searchers, would have made Ethereum more centralized, less democratic, and resulted in a much worse user experience.

There was a clear need for a solution that could fix issues around MEV and align the incentives of MEV extraction with the greater good of the Ethereum ecosystem. As mentioned in the history section, Flashbots was founded in 2020 by Stephane Gosselin and Phil Daian to build MEV solutions and increase awareness around the negative externalities caused by extraction. They outlined their main goals as:

1.Illuminate the state of MEV in Ethereum

- MEV was a dark forest before Flashbots started releasing research on the size of the market and the negative impacts of MEV extraction on users.
- Democratize the extraction of MEV.
- Flashbots did not want to wait for the (purposefully) slow-moving Ethereum Foundation to start addressing the issue at the protocol level.
- Instead, they chose to develop an off-chain infrastructure that reduced the barriers to entry and mitigated centralization pressures, which increased competition in the space.
- Distribute the benefits.
- Flashbots recognized MEV as a potential source of income for the ecosystem, protocols, and developers.
- They wanted to create mechanisms to redistribute some of the MEV extraction back to the community to ensure the economic sustainability of the ecosystem.

In the following sections, we explore the past, present, and future of Flashbots and MEV extraction on Ethereum.

6.1 Priority Gas Auctions - MEV On Ethereum Before Flashbots

Before proof-of-stake, MEV was called miner

extractable value because miners were in control of block construction and therefore of any extraction of value from transactions.

By default, miners would select transactions from the Ethereum mempool based on how much gas they paid. In addition to a block reward, miners received the gas fees paid by transactions in the block, which incentivized them to choose the highest-paying ones. Therefore, for a searcher to win an MEV opportunity, they had to ensure their transaction paid a higher gas fee than others competing for the same opportunity.

The previous graph illustrates how MEV extraction happened before Flashbots. Two searchers competing for an opportunity would iteratively bid up the gas paid by their transactions to ensure that they were offering marginally more than the other searcher. Searchers were willing to bid up the gas all the way to the profit on offer from the MEV opportunity. This was essentially an open-bid first-price auction, which is why this form of competition was called a priority-gas auction (“PGA

").

This on-chain free-for-all had a few undesired consequences:

- Because the auction happened on-chain, the loser of the auction also lost all the gas that they bid for the opportunity. Over time this meant that only well-capitalized searchers could participate in auctions.
- The fact that the loser is also liable to pay the bid from their failed transaction meant that searchers systematically underbid on opportunities to avoid execution risk. In turn, this resulted in lower revenues for miners, which made mining less economically viable.
- Latency optimization is very important when competing in PGAs. The lower a searcher's latency to the mempool was, the quicker they could react to the competing searcher's bids. This made it more likely that they would be able to submit the final, highest bid before the block was finalized. This gave an advantage to high-frequency trading firms with already existing expertise in latency optimization, furthering centralization.
- Advantages from vertical integration (e.g., latency, censorship of competition) with miners added further difficulty for small-scale searchers to compete on the market. Not everyone could source the required relationships with the mining pools and gain their trust to negotiate a deal to extract MEV for them.
- PGAs limited the searchers' ability to express specific ordering preferences. To be able to extract value, a searcher would need to either pay more (to frontrun), less gas (to backrun), or both, (to sandwich) than their targeted transaction. This limited their opportunities, as there are strategies where the bundle consists of more than just two transactions immediately following each other in a block.
- PGAs also significantly increased the demand for blockspace. Blocks were filled and reverted transactions from the searchers that did not win the priority gas auction. This contributed to an increase in gas prices for all ecosystem participants.

6.2 Flashbots Before Proof of Stake

To fix the existing issues with priority gas auctions, Flashbots created a first-price sealed-bid auction for blockspace that enabled the off-chain settlement of the rights to extract MEV from transactions in the mempool. This fixed many issues with priority gas auctions, especially the execution risk, latency advantages, wasted blockspace and lack of ordering control.

Miners would run a fork of the main Ethereum client called MEV-geth, which connected the miners to a relay hosted by Flashbots. This relay acted as a trusted interface between the searchers and the miners, allowing searchers to submit bundles to a network of miners through one endpoint.

The first version of MEV-geth submitted only the most profitable bundle, relative to its gas fees, to the miners. The miner placed the bundle at the top of the block and filled the rest from transactions it had access to locally. However, most of the time, there was more than one non-overlapping MEV bundle available, which is why Flashbots introduced bundle merging. Bundle merging would allow the miner to request as many non-overlapping bundles as they wanted in the block they were mining.

This bundle merging was still done by the miner in the MEV-geth software, which meant that miners would usually only request 2-3 bundles. This was because each additional bundle increased computational complexity exponentially as they would have to be compared against previously added bundles for overlaps.

To allow for the maximum number of bundles to be added in a block, the computational load for each miner had to be reduced. The solution was to make the relay do the hard work of merging bundles for the miner, which allowed it to build so-called mega bundles. The miners would still fill the remaining part of the block with pending transactions in the mempool, but the relay did a significant portion of their work for them. This was the first step towards the proposer-builder separation model introduced after the Merge (more on that next).

One could ask whether the relay is needed here – couldn't the miners run the auctions themselves?

- The relay spares miners from denial-of-service attacks. If a miner had to filter through all the bundles that were submitted to it, others could make the miner unable to build blocks by sending it a massive number of worthless bundles. A miner could have executed this attack in order to get rid of its competition.

- Another benefit is not having to worry about simulating the effect of the bundle before including it in the block. This means miners can focus on mining and still reap the benefits of MEV extraction.
- In this system, the miners are also unable to steal MEV bundles from searchers without being noticed and put on the Flashbots blacklist. The relay is able to compare the bundles it submits to the miners with the blocks that end up on-chain and to identify if a miner has submitted a bundle from a searcher without paying.
- In the past, searchers would have to build trust with individual mining pools, which further increased centralization both on the searcher and the miner front (with only specific entities being trusted and therefore being able to extract MEV).

By submitting the bundles through the relay, the MEV opportunities are extracted without the searchers' transactions ever hitting the mempool. This way searchers are safe from being sandwiched or frontrun by others.

In addition to this, the edge in MEV shifts from minimizing latency to the mempool (so that you can be the last one to update your transactions gas paid) to optimizing gas in transactions (so that you can bribe the miner more for inclusion). This helps new searchers compete against the incumbents, as long as they are smart enough, which democratizes access to MEV extraction.

MEV-geth bundles also gave searchers significantly more control over the ordering of the transactions. They could include as many transactions within their bundle as they wanted, in whichever order they wanted, without worrying about gas prices. Bundles introduced atomicity to multi-leg strategies like sandwiching – either the whole bundle with all of the searcher’s transactions would be included in the block, or the bundle would not be included.

The more competitive the market is, the more searchers will have to pay to the ecosystem for the right to extract MEV. Democratization and removal of capital-based barriers to entry creates more value for the system. Flashbots acted as a neutral intermediary that could be trusted to create a fair marketplace that would allow searchers to find miners and extract value in a trust minimized manner (at least compared to pre MEV-geth).

6.3 Proposer-Builder Separation - MEV in Proof of Stake

The existence of MEV created a concern that after the Merge, there would be significant centralization pressures on the validator set. Due to the complexity of MEV extraction, it was feared that only large entities that could run MEV strategies (or collaborate with searching teams) would be able to be effective validators.

This was because the best MEV extractors would be able to use their MEV profits to capture a larger share of the stake and get chosen as block proposers more often, which would allow them to capture more MEV. They could also use these increased profits to invest into developing better MEV strategies, furthering their advantage in an iterative process that would end up with only a few node operators controlling Ethereum.

It was clear that for Ethereum to stay decentralized in the future, it was necessary to minimize the overhead that would be required to be an effective validator. The solution was to separate the role of building the most profitable block (difficult) from the role of proposing it in the network (trivial).

Proposer-builder separation creates an open market for block proposers to source blocks from block builders, who compete against each other to build the block that can pay the highest fee to the validator.

- By ensuring that all validators have access to the most efficiently built blocks, PBS ensures that even solo validators staking just 32 ETH can earn the same yield as huge validator operators. By minimizing the threshold to get the highest yield available, PBS promotes decentralization at the validator level.
- From Econ 101, we know that the specialization of actors in a marketplace tends to also result in better outcomes. The belief is that by separating the roles of validator and block builder, builders can focus on trying to create the most profitable blocks without having to worry about running a validator. This in turn leads to higher yields and thus more value produced for the entire ecosystem.

Proposer-builder separation is able to ensure that MEV is not a contributor to validator centralization. This comes at the cost of pushing centralization to the block builder level, which is believed to be a preferred outcome.

- Centralization happening at the consensus level (validators) poses a more severe problem than centralization at the block-building level. In a world where block building is centralized, validators can still source blocks through the mempool and circumvent the block builders. If the consensus layer was centralized, there would be no recourse against censoring validators.
- Additionally, block building is more accessible from a resource intensity point of view. While block building requires significant technical sophistication and capital investment, entering the market is more accessible. Without PBS, new entrants would have to both source the immense capital required to be picked as a block proposer often and develop the necessary block-building algorithms.

While the importance of PBS was acknowledged going into the Merge, there are some clear theoretical and practical challenges with trying to implement it at the protocol level (enshrined PBS). These include ensuring censorship resistance and decentralizing the relayer role. The decision was made to not delay the merge to solve these issues, and therefore Flashbots took up the challenge to provide a solution for the interim.

6.4 Flashbots After The Merge - MEV-Boost As Proto-PBS

Going into the Merge, Flashbots had to adapt the MEV extraction infrastructure that it had built before. Flashbots built MEV-boost

as a solution for the period after the Merge but before the enshrinement of PBS at the protocol level.

MEV-boost is often called proto-PBS because the high-level architecture is very similar to how PBS is intended to be implemented in-protocol in the future. MEV-boost provides a market where even unsophisticated validators can have access to the most profitable blocks built in a slot.

The high-level infrastructure is pretty similar to MEV-geth, block producers received suggestions from a relay that received bundles from searchers. Only in this case, there is an additional entity between the relay and the searcher responsible for assembling the bundles.

In the next figure, we see the architecture of MEV-boost (note the addition of the builder role between searchers and relays). In Proof of Work, the relay would simply propagate valid bundles to miners who played both the role of the block proposer and the block builder.

Validators gain access to the block building market by running MEV-boost as a “sidecar module” with their normal validator clients.

Before the Merge, Flashbots could ensure that searchers’ bundles would not get stolen by having a whitelist of allowed mining pools that could receive bundles. These mining pools were significantly disincentivized from stealing MEV, because doing so would get them blacklisted from the Flashbots relay. This would mean losing access to a very lucrative source of income.

In PoS, this does not work, because all an MEV stealing validator would need to do to get off the blacklist is to restake 32 ETH and use a different address. Therefore, we needed to make MEV stealing unprofitable through a new architecture.

Relays play a similar role as in MEV-geth, acting as a verifying middleman and reducing the validator’s computational cost. They ensure the validity of the blocks (making sure that the proposer does not get slashed for signing an invalid block), calculate the value of the blocks and ensure proposers get paid the bribe they are promised. Relays then send headers of the blocks that they are building (not the full blocks) to the validator responsible for submitting the next block. If this validator is running MEV-boost, it automatically commits to validating the block that has the highest value paid to it.

This commitment consists of signing the block and propagating it in the network. The proposer sees the contents of the block only after it has committed to signing it. This means that if the proposer wanted to steal the MEV, it would have to submit a second block (double signing) where it would be the one extracting the MEV. This would incur slashing penalties, which are usually significant enough to deter it.

In some cases, the value of available MEV could be higher than the slashing penalty. Even then, a proposer is not guaranteed to be able to steal bundles by submitting a second block to the network. This is due to execution risk caused by how relays propagate blocks in MEV-boost.

- After the proposer signs the block header, the relay starts propagating the signed block to other nodes in the network. Crucially, it starts propagating this block even before the MEV-stealing proposer can submit a second block.
- If enough validators in the network see (and attest to) the original block before the altered block, where the proposer steals the MEV, the original will be added to the network. Essentially, to be successful, the stealing proposer would have to outrace the relay's original block.
- The validator would not only be unable to steal the MEV but would also suffer a slashing penalty. This further deters MEV stealing by validators. [38]

MEV-boost is now run on the majority of blocks validated on Ethereum (see the following figure). Validators not running MEV-boost forego the added yield from MEV. When they are proposing a block, they simply select transactions from the mempool based on fees paid. If a validator not using MEV-boost is responsible for validating a block, searchers can try to extract MEV through priority gas auctions.

MEV-boost has been able to reduce the centralization pressures at the validator level. The problem is that it is an out-of-protocol solution that is critical to the functioning of Ethereum. It also relies on third-party relays that are run essentially as a public good costing 20-100k [39] a year to run.

Relays are being run mainly by the same parties that run the builders. Builders are incentivized to run their own relays to ensure that their blocks make it to the proposer. Builders do not want to leave their revenues in the hands of unnecessary third parties. If they did not run their own relays, any downtime on the third-party relay would mean losing revenue due to something outside of their control.

There are only a few relay operators that supply the vast majority of blocks on Ethereum. Problems surrounding relay centralization have already created issues with censorship. Some relays censor all blocks containing US-sanctioned addresses because there are validators who want to avoid any potential legal ramifications of adding blocks with sanctioned transactions. Ensuring neutrality with regard to regulation is very important for Ethereum, which means that removing this vector of censorship is key.[40]

In fact, the original PBS specification did not include an external relay role. The current plan by the Ethereum Foundation, which is to enshrine PBS in the protocol, involves the creation of a neutral in-protocol relay for Ethereum. Because relays are currently the brokers of trust between builders and validators, an in-protocol solution will need to assure both parties that 1) builders will not have their MEV stolen by the proposer and 2) proposers will get the payment that was promised in the block header.

The current implementation of PBS relies on the Flashbots-built infrastructure, which means that installing the MEV-boost sidecar is still optional. This will not be the case when Ethereum moves to protocol-level proposer-builder separation, where validators automatically source blocks from block builders. Practical details on how the move away from the MEV-boost is still being discussed and there are many potential designs [41] that achieve in-protocol PBS. We will analyze these more deeply in a forthcoming report on PBS.

If we can mitigate centralization at the validator level, why not block building too? This is the era that SUAVE is trying to usher in, but more on that in the next section.

6.5 SUAVE - The Future of MEV?

In the current model of proposer-builder separation, even though validating is decentralized, centralization risks still exist at the block-builder level. While validators no longer have censorship capabilities if they run MEV-boost, block builders still retain autonomy over which transactions to choose in their blocks and how to order them.

Although anyone can run a block builder, the competitive dynamics that we explored in the previous sections result in a concentrated market. We recap these centralizing dynamics here:

- Exclusive order flow means that some builders have access to transactions that other builders do not, making them able to land blocks on the chain more often. The more transactions and bundles you have to choose from, the more likely you are to find the most profitable combination available. This effect can compound, because it is easier for builders that have a high win rate to secure exclusive order flow agreements with applications/wallets/searchers. These parties want their transactions to be included on-chain as quickly as possible, which means sending bundles to the builder that wins most often. These exclusivity agreements would include a payment from the builder to the originator, which explains why the wallets would be willing to submit to only one builder.

- Even if all builders had access to the same transaction flow, building the most profitable block out of those transactions is a difficult engineering challenge. The best-capitalized teams will be able to spend more to research and optimize their block-building algorithms, leading to a situation where only a few builders have the necessary capital to compete in the market.
- Flashbots feared that cross-domain MEV would mean that block builders with resources to extract MEV across multiple chains/domains would be more competitive than builders that only build for one chain. This is validated by the fact that one of the largest contributors to block builder centralization is integrated searcher-builders having a significant competitive advantage on CEX-DEX arbitrage (a form of cross-domain MEV).

That a small group of stakeholders, in a market with high barriers to entry, can control almost the entire transaction flow to an entire blockchain is a serious issue. Flashbots wants to solve it by decentralizing the block builder role completely.

The following figure illustrates the situation in the block-building space on Ethereum. The top four builders' control over 75% of all the blocks that are added to the blockchain.

The solution Flashbots created to solve block builder centralization is called the Single Unifying Auction for Value Expression, or SUAVE

. The idea is to create a completely new Layer-1 blockchain that acts as a mempool and the block building layer for multiple different blockchains that connect to it.

The information available on the technical implementation of SUAVE is sparse, but Flashbots has outlined the main design framework and goals of the system.[42],[43],[44]

There are three main components to the SUAVE chain: a universal preference environment, an execution market, and the decentralized block-building layer.

Universal Preference Environment

The concept of a user preference

is crucial to understanding what SUAVE tries to achieve. Essentially, execution is abstracted away from the user. They do not have to reason about which protocols to use, in what order, or how much gas to pay. Instead, it submits a signed offer (a preference) for a new stakeholder, called the executor

, to come up with the optimal way to achieve their goal, subject to specific conditions. If they achieve what the user wants given the conditions, a smart contract unlocks a payment from the user to the executor.

A concrete example of a user preference would be converting ETH on Ethereum to ARB tokens on Arbitrum, with the user conditioning the payment on the transaction being executed within a certain maximum slippage. User preferences can be more complex and the conditions more abstract than this. SUAVE creates an environment where users can communicate these preferences and have sophisticated parties compete for the right to execute them.

This stands in contrast to the current paradigm where users sign transactions that establish a specific computational path that instructs the EVM to execute a state transition achieving the user's goals. There is little leeway for a third party to optimize execution, which is not the case in the preference environment. Preferences can be thought of as the SUAVE equivalent for intents

, which have recently become a hot buzzword.

The universal preference environment can be thought of as a new type of mempool that allows users to express these preferences and propagates them to executors. Preferences can be thought of as the native transaction type on SUAVE.

Everything is built to enable the expression, execution, and settlement of these user preferences in a decentralized manner.

It is also important to note that it is not just retail users that can express their preferences on SUAVE. The idea is to allow searchers to express preferences about MEV opportunities. As an example, there could be a cross-chain arbitrage opportunity and a searcher could express a preference for it to be closed. Naturally, this would create another competitive market because other searchers would also be aware of these opportunities. Searchers would compete on who is willing to pay the executor the most to fulfill their preference.

After the user submits their preferences to the universal preference environment, the actor who gets to execute this preference is chosen in the optimal execution market.

Optimal Execution Market

While the preference environment allows to express and share a new type of transactions, the optimal execution market is an auction where specialized parties called the executors listen to the SUAVE mempool for incoming preferences and compete in an auction to execute these preferences in the best possible way. They are incentivized to do so by the associated reward promised by the user.

The executor is a new type of stakeholder in the lifetime of a blockchain transaction. Much like a searcher, executors run specialized algorithms that come up with ways to fulfil the users' requests and bid for the right to execute according to the reward offered by the user and the potential MEV available.

In fact, the executors in SUAVE are also the searchers. If a user preference creates an MEV opportunity, executors will compete to extract it and pay back as much of it back to the user. This means that the competitive landscape on SUAVE will mimic the one we see with searching on main-chain Ethereum today. Executors are likely to specialize in specific forms of preference execution and run different strategies, both risk-free EV_ordering and more speculative EV_signal to extract MEV out of the SUAVE mempool, while still fulfilling user execution needs. Returning MEV would mean that users could potentially transact with no gas fees or have the slippage their transaction generated returned to them – a feature that we see withorder flow auctions in the market today. The optimal execution market and universal preference environment classify SUAVE as a type of order flow auction. More on those in our dedicated report "Everything you need to know about OFAs".

The execution market will allow the executors to merge different user preferences for more efficient execution. A simple example of this is one user wanting to sell ETH for USDC and another user wanting to buy USDC for ETH. An executor could see these preferences, match the trades together and source any amount that was not fulfilled by the peer-to-peer trade on-chain, which would reduce slippage significantly.

Redistributing MEV back to the users will mean that the rest of the supply chain has to accept a smaller share of the pie. This is especially true for validators, who currently receive up to 99% of the value in competitive MEV from the searchers' bribes.

There is not much that these participants can do to prevent this shift in MEV share from happening. Validators could potentially ignore SUAVE as a source for bundles and either create all their blocks through the remaining transactions in the original mempool or use non-SUAVE block builders. The problem here is that since users are incentivized to route their MEV-creating transactions through SUAVE, there might not be a lot of MEV available in the chain's original mempool. This would mean that while paying much less to validators than what they currently receive, SUAVE-built blocks would still be more profitable than the alternative.

The fact that validators would still be free to choose blocks from non-SUAVE builders, or build the blocks themselves, ensures that SUAVE still must pay some share of the MEV back to the validator. This creates a balancing act between paying the user as much of the MEV as possible, while still being competitive enough to win in the block auction.

Where this equilibrium will end up is hard to predict and will highly depend on the share of the total transaction flow that will get routed through SUAVE. Already released order flow auctions have encoded this share to be anywhere from 50/50 to 90/10 between the user and the validator.

After an executor of a specific preference has been selected, the executor submits a bundle of transactions that achieve the user's goals to the decentralized block-building network.

Decentralized Block Building

The third main component is a decentralized network that allows block builders to collaborate in building blocks. The builders collect the preferences as bundles from the executors, after which they share information with each other on the block that they are trying to create. By collaborating, the builders can share their order flow and leverage the strengths of each other's building algorithms to build the most profitable block possible, which creates more value for the entire ecosystem.

There are a few benefits to collaborative block-building:

- It addresses the main reason SUAVE was created in the first place, builder centralization. Because blocks are built collaboratively, the threat from a few parties controlling the entire transaction flow to a chain is removed.
- Having exclusive order flow drives no edge when building on SUAVE because this information is shared (with the contents encrypted) with other builders participating in the decentralized block building. Users and wallets also have no incentive to submit their bundles to only one specific builder to ensure their transactions are private and cannot be frontrun.
- Because the collaborating block builders can specialize to build blocks for different chains, extraction of cross-domain MEV becomes a lot easier, with fewer barriers to entry. Builders can agree that a certain cross-domain MEV opportunity, like an arbitrage, should be extracted in the next block. Using SUAVE, they can now ensure that the next block SUAVE submits on both relevant chains contain the necessary transactions to close the arbitrage opportunity. This can be achieved without requiring every builder to be integrated on each chain, which lowers barrier to entry.

Flashbots wants to ensure that block builders do not leak the user preferences when building the blocks in a decentralized manner. Not having the transactions encrypted could be a problem because a builder could potentially front-run users, if they knew certain transactions would be submitted to a specific chain in the next block by SUAVE.

A simple example could be a sandwich attack, where a builder that knows a user has a preference to buy a token on Ethereum could submit a transaction to buy that same token before the user and sell the token after the user's transaction is finalized. Even if you could trust certain builders to not do this, the aim of the system is to be completely permissionless & trustless and to allow anyone to participate in the building. Therefore, user preferences need to stay private.

At the moment, the only existing technical solution to this is forcing the builders to run their block building algorithms in a "secure enclave", which is a dedicated part of the builder's computer that is able to run code in a fully private manner without leaking information outside of that system. This means that while the builder's block building algorithm can see the contents of the bundles that it receives, the builder itself cannot do anything with this information, because it would require running unauthorized software inside of this secure enclave, which SUAVE could detect.

The following figure recaps the currently outlined architecture of SUAVE:

Implications and the Future of SUAVE

Why does Flashbots expect other chains to want to outsource their block building and mempool to a completely new chain?

- The connecting chains benefit from maximally decentralized block building. This means that they are more protected from censorship (either due to regulation, like sanctions, or malicious actors). Ensuring that the transaction flow of an entire chain is not in the hands of a small group of stakeholders is the only way these chains can claim to be truly decentralized.
- Users of SUAVE integrated chains benefit from the fact that their transactions are guaranteed privacy and therefore cannot be frontrun. This means that the chains or other actors in that chain's ecosystem do not have to invest in researching and bootstrapping a custom solution to achieve the same result for their chain, which would also likely take more time.
- Aggregating and matching users' preferences in a single auction environment is a more efficient way to coordinate users' needs than having a dedicated auction for each chain individually. It unlocks many exciting applications, such as bridging or low-slippage peer-to-peer trading.

What are the implications of SUAVE on the current MEV supply chain?

- Decentralized block building has the potential to increase the average value of blocks, because builders share all of their order flow. Ceteris paribus, the more order flow a builder has access to, the higher the likelihood that it can construct the most valuable block in a slot. The main problem with this argument is the existence of CEX-DEX arbitrage. It remains to be seen whether a decentralized block builder, with a significant latency disadvantage (communication between builders + running the algorithm in an enclave) could outcompete the existing searcher-builders here.
- Builders/searchers: democratized and open access to user and searcher transactions, which decreases barriers to entry. A universal expression environment allows searchers to express more complex goals than what is possible today with bundles. Cross-chain coordination becomes much easier.
- Users benefit from private transactions that cannot be frontrun with minimal fees and get paid for the MEV that they create.

While SUAVE claims to solve block builder centralization and the suggested components do seem to achieve this goal, they also clearly achieve much more than that. By creating a marketplace for users to express their preferences, and for external parties to compete to offer the best execution of those preferences, SUAVE unlocks a new way for users to interact with blockchains. It also acts as a powerful tool that distributed applications can leverage to achieve the goals of their users, like bridging funds or trading assets.

Flashbots has defined a roadmap that takes the SUAVE ecosystem from a centralized version bootstrapped by Flashbots to a trust-minimized system in which they participate in the development of the marketplace, but do not participate in running any software that enables its functionality. Many of the outlined ideas sound quite abstract, with the technical implementation also being an open research problem in many cases.

There are still many open questions as to how SUAVE will achieve the goals it has identified. These include:

- How exactly will the different chains plug into the SUAVE chain? How will users interact with SUAVE? There are even suggestions to build the SUAVE chain as a roll-up instead of an independent Layer-1 blockchain.
- Will there be MEV on SUAVE? What will be its settlement mechanism?
- How are block builders incentivized in decentralized block building? If one builder’s algorithm contributes more to increasing a specific block's value, how do they get compensated for this?
- What is the exact mechanism that redistributes the MEV back to the users?
- How will the secure enclaves work? What are the potential attack vectors in this solution?
- How does Flashbots find a path to revenue if all the projects they are currently building are meant to be public goods?

Acknowledgements

Thanks to Danning Sui, 0xTaker, Greg Vardy, Eduardo Carvalho, Alex Watts, Barry Plunkett, Maghnus Mareneck, Barnabé Monnot, Justin Drake, 0xPrincess, for your review & valuable feedback.

Footnotes

1. 1.

Note that there is no global mempool. It is local to each validator and always evolving. Ethereum nodes become aware of transactions at different times, because transactions are gossiped in the peer-to-peer network, not sent to everyone at the same time. For further exploration of the mempool, check out ethernow.xyz by Blocknative.

1. 2.

Block builders are responsible for the construction of the entire block, so transactions that are not included in a searchers bundle (because there is no MEV to extract from them, or no one is willing to extract it) are also included because the block builder picks them from the mempool.

1. 3.

Term coined by Ethereum Robust Incentives Group. See also: Infinite Games, Frontier Research.

1. 4.

REWARD offered for hash collisions for SHA1, SHA256, RIPEMD160 and other, Todd, 2013

1. 5.

With the possible caveat of ordinal NFTs.

See also: On the Instability of Bitcoin Without the Block Reward, 2016, Carlsten et al. Describes how Bitcoin miners could steal transaction fees from previous blocks through intentional reorg attacks.

1. 6.

Miners Frontrunning, pmcgoohan, 2014

1. 7.

The Cost of Decentralization in 0x and EtherDelta, Bentov et al., 2017

1. 8.

Miners aren't your friends, Prestwich, 2018

1. 9.

Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges, Daian et al., 2019

1. 10.

Time bandit attacks, MEV Wiki

1. 11.

Flashbots: Frontrunning the MEV crisis, Gosselin, 2020

1. 12.

A new game in town, Frontier Research, 2023

1. 13.

We include multiple blocks here because in some forms of multi-block MEV, extraction happens across multiple blocks, but by pure reordering, like in sandwiching.

1. 14.

This is MEV, Xinyuan Sun, 2022

1. 15.

Something being a valuable, even necessary service does not mean that the current way in which it is provided is unambiguously “good”. Designs might make users overpay for a service, meaning that the MEV / cost of service is higher than it would be in a system with better mechanism design.

1. 16.

Sources: MEV in 2021: A Year In Review, Flashbots

2022: Ethereum & BNB, Cosmos, Polygon

2023: Solana

1. 17.

MEV Supply Chain, Stephane Gosselin, 2022

1. 18.

Further reading: What are Flash Loans?

1. 19.

As an example, the MEV bot jaredfromsubway.eth takes inventory risk by holding memecoins while sandwiching users trades: "How Jaredfromsubway.eth Dominates Sandwich Bots and Rules the Meme Crypto Market", 0xFF

1. 20.

All is Fair in Arb and MEV on Avalanche C-Chain, Daniel D. McKinnon

1. 21.

Friend.tech MEV: https://twitter.com/tomwanhh/status/1694387751012188186

1. 22.

Example where the searcher paid 99.99% of the value of the sandwich to the validator. Source EigenPhi.

1. 23.

Note that this is different from an OFA, because opportunities are still available for competition in an OFA.

1. 24.

Proposer/block builder separation-friendly fee market designs, Vitalik Buterin, 2021

1. 25.

This data is publicly available on sites like https://relayscan.io or https://mevboost.pics

1. 26.

The Centralizing Effects of Private Order Flow on Proposer-Builder Separation, Gupta, M. Pai, Resnick, 2023

1. 27.

From discussions with a block builder: the cost of running a builder is over 30k/month + salaries for multiple developers.

1. 28.

This is how decentralized staking pools like LIDO, Rocket Pool or Ankr on Ethereum are able to generate yield for much more than just 32 ETH.

1. 29.

One example is multi-block MEV, where a node operator that knows they are proposing two blocks in a row, can hold off on extracting MEV in the first block, to maximize the amount that can be extracted in the second (i.e., allowing prices to diverge, to maximize arbitrages). Practically, there have been very few examples and in the absence of such opportunities, operators would be better off running MEV-boost.

1. 30.

Further reading: Unlock 10% more block rewards with MEV Maximizer feature, P2P.org

1. 31.

Thriving Amidst Turmoil: the Key Takeaways for MEV Annual Performance in 2022, EigenPhi, 2022

1. 32.

Cyclic arbitrage in decentralized exchanges, Wang et al., 2022

1. 33.

A Tale of Two Arbitrages, Chen et al., 2023

1. 34.

Robert Miller from Flashbots on jaredfromsubway:https://twitter.com/bertcmiller/status/1656392876438462464

1. 35.

Flashbots Protect documentation

1. 36.

Example of such an attack, where a bot bought 200k worth of a token, but was unable to sell the whole position afterwards:

https://twitter.com/SiegeRhino2/status/1381035640989626369?s=20

A developer can program some safeguards to check for such attacks, but it is difficult to cover all potential attack vectors and they make the bot less gas efficient.

1. 37.

Further reading: https://twitter.com/bertcmiller/status/1613257826654392320

1. 38.

Even this was not enough to fully protect from MEV stealing, it just made it extremely rare. A vulnerability in the Flashbots relay made it possible for a block proposer to submit an invalid block header, which allowed it to submit steal $20M from a searcher without having to “outrace” the relay. This has been fixed:

https://collective.flashbots.net/t/post-mortem-april-3rd-2023-mev-boost-relay-incident-and-related-timing-issue/

1. 39.

Ideas for incentivizing relays, Daniel Marzec, 2022

1. 40.

Dashboard visualizing the current situation relating to censorship of OFAC-sanctioned transactions:https://censorship.pics

1. 41.

Why enshrine Proposer-Builder Separation? A viable path to ePBS, Neuder & Drake, 2023

1. 42.

The Future of MEV is SUAVE, Flashbots, 2022

1. 43.

The MEVM, SUAVE Centauri, and Beyond, Robert Miller, 2023

1. 44.

M.O.S.S. - A SUAVE Block Building Proposal, Quintus Kilbourn, 2023

Note that there is no global mempool. It is local to each validator and always evolving. Ethereum nodes become aware of transactions at different times, because transactions are gossiped in the peer-to-peer network, not sent to everyone at the same time. For further exploration of the mempool, check out ethernow.xyz by Blocknative.

Block builders are responsible for the construction of the entire block, so transactions that are not included in a searchers bundle (because there is no MEV to extract from them, or no one is willing to extract it) are also included because the block builder picks them from the mempool.

Term coined by Ethereum Robust Incentives Group. See also: Infinite Games, Frontier Research.

REWARD offered for hash collisions for SHA1, SHA256, RIPEMD160 and other, Todd, 2013

With the possible caveat of ordinal NFTs.

See also: On the Instability of Bitcoin Without the Block Reward, 2016, Carlsten et al. Describes how Bitcoin miners could steal transaction fees from previous blocks through intentional reorg attacks.

Miners Frontrunning, pmcgoohan, 2014

The Cost of Decentralization in 0x and EtherDelta, Bentov et al., 2017

Miners aren't your friends, Prestwich, 2018

Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges, Daian et al., 2019

Time bandit attacks, MEV Wiki

Flashbots: Frontrunning the MEV crisis, Gosselin, 2020

A new game in town, Frontier Research, 2023

We include multiple blocks here because in some forms of multi-block MEV, extraction happens across multiple blocks, but by pure reordering, like in sandwiching.

This is MEV, Xinyuan Sun, 2022

Something being a valuable, even necessary service does not mean that the current way in which it is provided is unambiguously “good”. Designs might make users overpay for a service, meaning that the MEV / cost of service is higher than it would be in a system with better mechanism design.

Sources: MEV in 2021: A Year In Review, Flashbots

2022: Ethereum & BNB, Cosmos, Polygon

2023: Solana

MEV Supply Chain, Stephane Gosselin, 2022

Further reading: What are Flash Loans?

As an example, the MEV bot jaredfromsubway.eth takes inventory risk by holding memecoins while sandwiching users trades: "How Jaredfromsubway.eth Dominates Sandwich Bots and Rules the Meme Crypto Market", 0xFF

All is Fair in Arb and MEV on Avalanche C-Chain, Daniel D. McKinnon

Friend.tech MEV: https://twitter.com/tomwanhh/status/1694387751012188186

Example where the searcher paid 99.99% of the value of the sandwich to the validator. Source EigenPhi.

Note that this is different from an OFA, because opportunities are still available for competition in an OFA.

Proposer/block builder separation-friendly fee market designs, Vitalik Buterin, 2021

This data is publicly available on sites like https://relayscan.io or https://mevboost.pics

The Centralizing Effects of Private Order Flow on Proposer-Builder Separation, Gupta, M. Pai, Resnick, 2023

From discussions with a block builder: the cost of running a builder is over 30k/month + salaries for multiple developers.

This is how decentralized staking pools like LIDO, Rocket Pool or Ankr on Ethereum are able to generate yield for much more than just 32 ETH.

One example is multi-block MEV, where a node operator that knows they are proposing two blocks in a row, can hold off on extracting MEV in the first block, to maximize the amount that can be extracted in the second (i.e., allowing prices to diverge, to maximize arbitrages). Practically, there have been very few examples and in the absence of such opportunities, operators would be better off running MEV-boost.

Further reading: Unlock 10% more block rewards with MEV Maximizer feature, P2P.org

Thriving Amidst Turmoil: the Key Takeaways for MEV Annual Performance in 2022, EigenPhi, 2022

Cyclic arbitrage in decentralized exchanges, Wang et al., 2022

A Tale of Two Arbitrages, Chen et al., 2023

Robert Miller from Flashbots on jaredfromsubway: https://twitter.com/bertcmiller/status/1656392876438462464

Flashbots Protect documentation

Example of such an attack, where a bot bought 200k worth of a token, but was unable to sell the whole position afterwards:

https://twitter.com/SiegeRhino2/status/1381035640989626369?s=20

A developer can program some safeguards to check for such attacks, but it is difficult to cover all potential attack vectors and they make the bot less gas efficient.

Further reading: https://twitter.com/bertcmiller/status/1613257826654392320

Even this was not enough to fully protect from MEV stealing, it just made it extremely rare. A vulnerability in the Flashbots relay made it possible for a block proposer to submit an invalid block header, which allowed it to submit steal $20M from a searcher without having to “outrace” the relay. This has been fixed:

https://collective.flashbots.net/t/post-mortem-april-3rd-2023-mev-boost-relay-incident-and-related-timing-issue/

Ideas for incentivizing relays, Daniel Marzec, 2022

Dashboard visualizing the current situation relating to censorship of OFAC-sanctioned transactions: https://censorship.pics

Why enshrine Proposer-Builder Separation? A viable path to ePBS, Neuder & Drake, 2023

The Future of MEV is SUAVE, Flashbots, 2022

The MEVM, SUAVE Centauri, and Beyond, Robert Miller, 2023

M.O.S.S. - A SUAVE Block Building Proposal, Quintus Kilbourn, 2023