---

title: Maximal extractable value (MEV) description: An introduction to maximal extractable value (MEV) lang: en

---

Maximal extractable value (MEV) refers to the maximum value that can be extracted from block production in excess of the standard block reward and gas fees by including, excluding, and changing the order of transactions in a block.

Miner extractable value {#miner-extractable-value}

Maximal extractable value was first applied in the context ofproof-of-work, and initially referred to as "miner extractable value". This is because in proof-of-work, miners control transaction inclusion, exclusion, and ordering. However, since the transition to proof-of-stake via The Merge validators have been responsible for these roles, and mining is no longer part of the Ethereum protocol. The value extraction methods still exist, though, so the term "Maximal extractable value" is now used instead.

Prerequisites {#prerequisites}

Make sure you're familiar withtransactions, blocks, proof-of-stake and gas. Familiarity with dapps and DeFi is helpful as well.

MEV extraction {#mev-extraction}

In theory MEV accrues entirely to validators because they are the only party that can guarantee the execution of a profitable MEV opportunity. In practice, however, a large portion of MEV is extracted by independent network participants referred to as "searchers." Searchers run complex algorithms on blockchain data to detect profitable MEV opportunities and have bots to automatically submit those profitable transactions to the network.

Validators do get a portion of the full MEV amount anyway because searchers are willing to pay high gas fees (which go to the validator) in exchange for higher likelihood of inclusion of their profitable transactions in a block. Assuming searchers are economically rational, the gas fee that a searcher is willing to pay will be an amount up to 100% of the searcher's MEV (because if the gas fee was higher, the searcher would lose money).

With that, for some highly competitive MEV opportunities, such as DEX arbitrage, searchers may have to pay 90% or even more of their total MEV revenue in gas fees to the validator because so many people want to run the same profitable arbitrage trade. This is because the only way to guarantee that their arbitrage transaction runs is if they submit the transaction with the highest gas price.

Gas golfing {#mev-extraction-gas-golfing}

This dynamic has made being good at "gas golfing" — programming transactions so that they use the least amount of gas — a competitive advantage, because it allows searchers to set a higher gas price while keeping their total gas fees constant (since gas fees = gas price * gas used).

A few well-known gas golf techniques include: using addresses that start with a long string of zeroes (e.g. 0x0000000000C521824EaFf97Eac7B73B084ef9306) since they take less space (and hence gas) to store; and leaving small ERC-20 token balances in contracts, since it costs more gas to initialize a storage slot (the case if the balance is 0) than to update a storage slot. Finding more techniques to reduce gas usage is an active area of research among searchers.

Generalized frontrunners {#mev-extraction-generalized-frontrunners}

Rather than programming complex algorithms to detect profitable MEV opportunities, some searchers run generalized frontrunners. Generalized frontrunners are bots that watch the mempool to detect profitable transactions. The frontrunner will copy the potentially profitable transaction's code, replace addresses with the frontrunner's address, and run the transaction locally to double-check that the modified transaction results in a profit to the frontrunner's address. If the transaction is indeed profitable, the frontrunner will submit the modified transaction with the replaced address and a higher gas price, "frontrunning" the original transaction and getting the original searcher's MEV.

Flashbots {#mev-extraction-flashbots}

Flashbots is an independent project which extends execution clients with a service that allows searchers to submit MEV transactions to validators without revealing them to the public mempool. This prevents transactions from being frontrun by generalized frontrunners.

MEV examples {#mev-examples}

MEV emerges on the blockchain in a few ways.

DEX arbitrage {#mev-examples-dex-arbitrage}

Decentralized exchange (DEX) arbitrage is the simplest and most well-known MEV opportunity. As a result, it is also the most competitive.

It works like this: if two DEXes are offering a token at two different prices, someone can buy the token on the lower-priced DEX and sell it on the higher-priced DEX in a single, atomic transaction. Thanks to the mechanics of the blockchain, this is true, riskless arbitrage.

Here's an example of a profitable arbitrage transaction where a searcher turned 1,000 ETH into 1,045 ETH by taking advantage of different pricing of the ETH/DAI pair on Uniswap vs. Sushiswap.

Liquidations {#mev-examples-liquidations}

Lending protocol liquidations present another well-known MEV opportunity.

Lending protocols like Maker and Aave require users to deposit some collateral (e.g. ETH). This deposited collateral is then used to then lend out to other users.

Users can then borrow assets and tokens from others depending on what they need (e.g. you might borrow MKR if you want to vote in a MakerDAO governance proposal) up to a certain percentage of their deposited collateral. For example, if the borrowing amount is a maximum of 30%, a user who deposits 100 DAI into the protocol can borrow up to 30 DAI worth of another asset. The protocol determines the exact borrowing power percentage.

As the value of a borrower's collateral fluctuates, so too does their borrowing power. If, due to market fluctuations, the value of borrowed assets exceeds say, 30% of the value of their collateral (again, the exact percentage is determined by the protocol), the protocol typically allows anyone to liquidate the collateral, instantly paying off the lenders (this is similar to how margin calls work in traditional finance). If liquidated, the borrower usually has to pay a hefty liquidation fee, some of which goes to the liquidator — which is where the MEV opportunity comes in.

Searchers compete to parse blockchain data as fast as possible to determine which borrowers can be liquidated and be the first to submit a liquidation transaction and collect the liquidation fee for themselves.

Sandwich trading {#mev-examples-sandwich-trading}

Sandwich trading is another common method of MEV extraction.

To sandwich, a searcher will watch the mempool for large DEX trades. For instance, suppose someone wants to buy 10,000 UNI with DAI on Uniswap. A trade of this magnitude will have a meaningful effect on the UNI/DAI pair, potentially significantly raising the price of UNI relative to DAI.

A searcher can calculate the approximate price effect of this large trade on the UNI/DAI pair and execute an optimal buy order immediately *before* the large trade, buying UNI cheaply, then execute a sell order immediately*after* the large trade, selling it for the higher price caused by the large order.

Sandwiching, however, is riskier as it isn't atomic (unlike DEX arbitrage, as described above) and is prone to a salmonella attack.

NFT MEV {#mev-examples-nfts}

MEV in the NFT space is an emergent phenomenon, and isn't necessarily profitable.

However, since NFT transactions happen on the same blockchain shared by all other Ethereum transactions, searchers can use similar techniques as those used in traditional MEV opportunities in the NFT market too.

For example, if there's a popular NFT drop and a searcher wants a certain NFT or set of NFTs, they can program a transaction such that they are the first in line to buy the NFT, or they can buy the entire set of NFTs in a single transaction. Or if an NFT is mistakenly listed at a low price, a searcher can frontrun other purchasers and snap it up for cheap.

One prominent example of NFT MEV occurred when a searcher spent $7 million to buy every single Cryptopunk at the price floor. A blockchain researcher explained on Twitter how the buyer worked with an MEV provider to keep their purchase secret.

The long tail {#mev-examples-long-tail}

DEX arbitrage, liquidations, and sandwich trading are all very well-known MEV opportunities and are unlikely to be profitable for new searchers. However, there is a long tail of lesser known MEV opportunities (NFT MEV is arguably one such opportunity).

Searchers who are just getting started may be able to find more success by searching for MEV in this longer tail. Flashbot's MEV job board lists some emerging opportunities.

Effects of MEV {#effects-of-mev}

MEV is not all bad — there are both positive and negative consequences to MEV on Ethereum.

The good {#effects-of-mev-the-good}

Many DeFi projects rely on economically rational actors to ensure the usefulness and stability of their protocols. For instance, DEX arbitrage ensures that users get the best, most correct prices for their tokens, and lending protocols rely on speedy liquidations when borrowers fall below collateralization ratios to ensure lenders get paid back.

Without rational searchers seeking and fixing economic inefficiencies and taking advantage of protocols' economic incentives, DeFi protocols and dapps in general may not be as robust as they are today.

The bad {#effects-of-mev-the-bad}

At the application layer, some forms of MEV, like sandwich trading, result in an unequivocally worse experience for users. Users who are sandwiched face increased slippage and worse execution on their trades.

At the network layer, generalized frontrunners and the gas-price auctions they often engage in (when two or more frontrunners compete for their transaction to be included in the next block by progressively raising their own transactions' gas price) result in network congestion and high gas prices for everyone else trying to run regular transactions.

Beyond what's happening *within* blocks, MEV can have deleterious effects *between* blocks. If the MEV available in a block significantly exceeds the standard block reward, validators may be incentivized to reorg blocks and capture the MEV for themselves, causing blockchain re-organization and consensus instability.

This possibility of blockchain re-organization has been previously explored on the Bitcoin blockchain. As Bitcoin's block reward halves and transaction fees make up a greater and greater portion of the block reward, situations arise where it becomes economically rational for miners to give up the next block's reward and instead remine past blocks with higher fees. With the growth of MEV, the same sort of situation could occur in Ethereum, threatening the integrity of the blockchain.

State of MEV {#state-of-mev}

MEV extraction ballooned in early 2021, resulting in extremely high gas prices in the first few months of the year. The emergence of Flashbots's MEV relay has reduced the effectiveness of generalized frontrunners and has taken gas price auctions off-chain, lowering gas prices for ordinary users.

While many searchers are still making good money from MEV, as opportunities become more well-known and more and more searchers compete for the same opportunity, validators will capture more and more total MEV revenue (because the same sort of gas auctions as originally described above also occur in Flashbots, albeit privately, and validators will capture the resulting gas revenue). MEV is also not unique to Ethereum, and as opportunities become more competitive on Ethereum, searchers are moving to alternate blockchains like Binance Smart Chain, where similar MEV opportunities as those on Ethereum exist with less competition.

On the other hand, the transition from proof-of-work to proof-of-stake and the ongoing effort to scale Ethereum using rollups all change the MEV landscape in ways that are still somewhat unclear. It is not yet well known how having guaranteed block-proposers known slightly in advance changes the dynamics of MEV extraction compared to the probabilistic model in proof-of-work or how this will be disrupted when single secret leader election and distributed validator technology get implemented. Similarly, it remains to be seen what MEV opportunities exist when most user activity is ported away from Ethereum and onto its layer 2 rollups and shards.

MEV in Ethereum Proof-of-Stake (PoS) {#mev-in-ethereum-proof-of-stake}

As explained, MEV has negative implications for overall user experience and consensus-layer security. But Ethereum's transition to a proof-of-stake consensus (dubbed "The Merge") potentially introduces new MEV-related risks:

Validator centralization {#validator-centralization}

In post-Merge Ethereum, validators (having made security deposits of 32 ETH) come to consensus on the validity of blocks added to the Beacon Chain. Since 32 ETH may be out of the reach of many, joining a staking pool may be a more feasible option. Nevertheless, a healthy distribution of solo stakers is ideal, as it mitigates the centralization of validators and improves Ethereum's security.

However, MEV extraction is believed to be capable of accelerating validator centralization. This is partly because, as validators earn less for proposing blocks than miners currently do, MEV extraction may greatlyinfluence validator earnings after The Merge.

Larger staking pools will likely have more resources to invest in necessary optimizations to capture MEV opportunities. The more MEV these pools extract, the more resources they have to improve their MEV-extraction capabilities (and increase overall revenue), essentially creating economies of scale.

With fewer resources at their disposal, solo stakers may be unable to profit from MEV opportunities. This may increase the pressure on independent validators to join powerful staking pools to boost their earnings, reducing decentralization in Ethereum.

Permissioned mempools {#permissioned-mempools}

In response to sandwiching and frontrunning attacks, traders may start conducting off-chain deals with validators for transaction privacy. Instead of sending a potential MEV transaction to the public mempool, the trader sends it directly to the validator, who includes it in a block and splits profits with the trader.

"Dark pools" are a larger version of this arrangement and function as permissioned, access-only mempools open to users willing to pay certain fees. This trend would diminish Ethereum's permissionlessness and trustlessness and potentially transform the blockchain into a "pay-to-play" mechanism that favors the highest bidder.

Permissioned mempools would also accelerate the centralization risks described in the previous section. Large pools running multiple validators will likely benefit from offering transaction privacy to traders and users, increasing their MEV revenues.

Combating these MEV-related problems in post-Merge Ethereum is a core area of research. To date, two solutions proposed to reduce the negative impact of MEV on Ethereum's decentralization and security after The Merge are **Proposer-Builder Separation (PBS)** and the **Builder API**.

Proposer-Builder Separation {#proposer-builder-separation}

In both proof-of-work and proof-of-stake, a node that builds a block proposes it for addition to the chain to other nodes participating in consensus. A new block becomes part of the canonical chain after another miner builds on top of it (in PoW) or it receives attestations from the majority of validators (in Pos).

The combination of block producer and block proposer roles is what introduces most of the MEV-related problems described previously. For example, consensus nodes are incentivized to trigger chain reorganizations in time-bandit attacks to maximize MEV earnings.

Proposer-builder separation (PBS) is designed to mitigate the impact of MEV, especially at the consensus layer. PBS' major feature is the separation of block producer and block proposer rules. Validators are still responsible for proposing and voting on blocks, but a new class of specialized entities, called **block builders**, are tasked with ordering transactions and building blocks.

Under PBS, a block builder creates a transaction bundle and places a bid for its inclusion in a Beacon Chain block (as the "execution payload"). The validator selected to propose the next block then checks the different bids and chooses the bundle with the highest fee. PBS essentially creates an auction market, where builders negotiate with validators selling blockspace.

Current PBS designs use a commit-reveal scheme in which builders only publish a cryptographic commitment to a block's contents (block header) along with their bids. After accepting the winning bid, the proposer creates a signed block proposal that includes the block header. The block builder is expected to publish the full block body after seeing the signed block proposal, and it must also receive enough attestations from validators before it is finalized.

How does proposer-builder separation mitigate MEV's impact? {#how-does-pbs-curb-mev-impact}

In-protocol proposer-builder separation reduces MEV's effect on consensus by removing MEV extraction from the purview of validators. Instead, block builders running specialized hardware will capture MEV opportunities going forward.

This doesn't exclude validators totally from MEV-related income, though, as builders must bid high to get their blocks accepted by validators. Nevertheless, with validators no longer directly focused on optimizing MEV income, the threat of time-bandit attacks reduces.

Proposer-builder separation also reduces MEV's centralization risks. For instance, the use of a commit-reveal scheme removes the need for builders to trust validators not to steal the MEV opportunity or expose it to other builders. This lowers the barrier for solo stakers to benefit from MEV, otherwise, builders would trend towards favoring large pools with off-chain reputation and conducting off-chain deals with them.

Similarly, validators don't have to trust builders not to withhold block bodies or publish invalid blocks because payment is unconditional. The validator's fee still processes even if the proposed block is unavailable or declared invalid by other validators. In the latter case, the block is simply discarded, forcing the block builder to lose all transaction fees and MEV revenue.

Builder API {#builder-api}

While proposer-builder separation promises to reduce the effects of MEV extraction, implementing it requires changes to the consensus protocol. Specifically, the fork choice rule on the Beacon Chain would need to be updated. TheBuilder API is a temporary solution aimed at providing a working implementation of proposer-builder separation, albeit with higher trust assumptions.

The Builder API is a modified version of the Engine API used by consensus layer clients to request execution payloads from execution layer clients. As outlined in the honest validator specification, validators selected for block proposing duties request a transaction bundle from a connected execution client, which they include in the proposed Beacon Chain block.

The Builder API also acts as a middleware between validators and execution-layer clients; but it is different because it allows validators on the Beacon Chain to source blocks from external entities (instead of building a block locally using an execution client).

Below is an overview of how the Builder API works:

1. The Builder API connects the validator to a network of block builders running execution layer clients. Like in PBS, builders are specialized parties that invest in resource-intensive block-building and use different strategies to maximize revenue earned from MEV + priority tips.
2. A validator (running a consensus layer client) requests execution payloads along with bids from the network of builders. Bids from builders will contain the execution payload header—a cryptographic commitment to the payload's contents—and a fee to be paid to the validator.
3. The validator reviews the incoming bids and picks the execution payload with the highest fee. Using the Builder API, the validator creates a "blinded" Beacon block proposal that includes only their signature and the execution payload header and sends it to the builder.
4. The builder running the Builder API is expected to respond with the full execution payload upon seeing the blinded block proposal. This allows the validator to create a "signed" Beacon block, which they propagate throughout the network.
5. A validator using the Builder API is still expected to build a block locally in case the block builder fails to respond promptly, so they don't miss out on block proposal rewards. However, validator cannot create another block using either the now-revealed transactions or another set, as it would amount to *equivocation* (signing two blocks within the same slot), which is a slashable offense.

An example implementation of the Builder API is MEV Boost, an improvement on the Flashbots auction mechanism designed to curb the negative externalities of MEV on Ethereum. Flashbots auction allows miners in proof-of-work to outsource the work of building profitable blocks to specialized parties called **searchers**.

Searchers look for lucrative MEV opportunities and send transaction bundles to miners along with a sealed-price bid for inclusion in the block. The miner running mev-geth, a forked version of the go-ethereum (Geth) client only has to choose the bundle with the most profit and mine it as part of the new block. To protect miners from spam and invalid transactions, transaction bundles pass through **relayers** for validation before getting to miners.

MEV Boost retains the same workings of the original Flashbots auction, albeit with new features designed for Ethereum's switch to proof-of-stake. Searchers still find profitable MEV transactions for inclusion in blocks, but a new class of specialized parties, called **builders**, are responsible for aggregating transactions and bundles into blocks. A builder accepts sealed-price bids from searchers and runs optimizations to find the most profitable ordering.

The relayer is still responsible for validating transaction bundles before passing them to the proposer. However, MEV Boost introduces **escrows** responsible for providing data availability by storing block bodies sent by builders and block headers sent by validators. Here, a validator connected to a relay asks for available execution payloads and uses MEV Boost's ordering algorithm to select the payload header with the highest bid + MEV tips.

How does the Builder API mitigate MEV's impact? {#how-does-builder-api-curb-mev-impact}

The core benefit of the Builder API is its potential to democratize access to MEV opportunities. Using commit-reveal schemes eliminates trust assumptions and reduces entry barriers for validators seeking to benefit from MEV. This should reduce the pressure on solo stakers to integrate with large staking pools in order to boost MEV profits.

Widespread implementation of the Builder API will encourage greater competition among block builders, which increases censorship resistance. As validators review bids from multiple builders, a builder intent on censoring one or more user transactions must outbid all other non-censoring builders to be successful. This dramatically increases the cost of censoring users and discourages the practice.

Some projects, such as MEV Boost, use the Builder API as part of an overall structure designed to provide transaction privacy to certain parties, such as traders trying to avoid frontrunning/sandwiching attacks. This is achieved by providing a private communication channel between users and block builders. Unlike the permissioned mempools described earlier, this approach is beneficial for the following reasons:

1. The existence of multiple builders on the market makes censoring impractical, which benefits users. In contrast, the existence of centralized and trust-based dark pools would concentrate power in the hands of a few block builders and

increase the possibility of censoring.

2. The Builder API software is open-source, which allows anyone to offer block-builder services. This means users aren't forced into using any particular block builder and improves Ethereum's neutrality and permissionlessness. Moreover, MEV-seeking traders won't inadvertently contribute to centralization by using private transaction channels.

Related resources {#related-resources}

- Flashbots docs
- Flashbots GitHub
- MEV-Explore - *Dashboard and live transaction explorer for MEV transactions*
- mevboost.org - *Tracker with real-time stats for MEV-Boost relays and block builders*

Further reading {#further-reading}

- What Is Miner-Extractable Value (MEV)?
- MEV and Me
- Ethereum is a Dark Forest
- Escaping the Dark Forest
- Flashbots: Frontrunning the MEV Crisis
- @bertcmiller's MEV Threads
- MEV-Boost: Merge ready Flashbots Architecture
- What Is MEV Boost
- Why run mev-boost?
- The Hitchhikers Guide To Ethereum