

*Maximal* (formerly *Miner*) *Extractable Value* ( $\text{MEV}$ ) is the value that can be extracted from a blockchain by any agent without special permissions. Considering this permissionless nature, any agent with transaction ordering rights will be in a privileged position to perform the extraction. In Proof of Work blockchains, it is miners who determine transaction ordering within a block, hence the former "*miner*" term. In practice, bot operators seek to extract  $\text{MEV}$  by either paying high fees to increase the likelihood that their transactions are mined, or by fine-tuning their gas price choices in order to "time" their transactions right, as is the case when backrunning an oracle update to perform a liquidation.

Despite much recent discussion about the topic and, in particular, its associated risks for the Ethereum protocol, we still lack a cohesive formal model for quantifying  $\text{MEV}$  extraction. At [Flashbots](#), we have released an [explorer](#) where we shed light on various aspects of this phenomenon. While we do elaborate on our metrics in the site, they still lack a formal definition. Here, we attempt to provide a unifying operational framework that consolidates  $\text{MEV}$  extraction metrics, focusing on the Ethereum network<sup>[1]</sup>.

## General Considerations

The first important point to make is that  $\text{MEV}$  is a theoretical quantity that we can only approach asymptotically. Unforeseen extraction methods can and will be devised (every new DeFi hack is an  $\text{MEV}$  extraction event). Hence, we will here focus instead on the *Realized Extractable Value*, notated  $\text{REV}$ , where  $\text{REV} \leq \text{MEV}$ . In other words,  $\text{REV}$  is the *actual value* extracted from the blockchain from  $\text{MEV}$  opportunities<sup>[2]</sup>.

We note that there's two classes of actors in the system, *searchers* and *miners*. We use the generic label *miner* for actors that bear the privileged role of transaction inclusion and ordering. *Searchers* are any non-privileged actors aiming to profit from these opportunities. Crucially, miners can also act as searchers.

Extracting  $\text{MEV}$  incurs externalities, like increasing gas costs for all users of the network, or bloating the chain. We won't dwell into the implications of these externalities here, but we will make them manifest in the model. For more on this and the " $\text{MEV}$  Crisis", be sure to check our [Flashbots introductory post](#).

We are now ready to begin building our framework, starting with a simple model that ignores direct miner payments and searcher competition and other externalities like opportunity checks. We will revisit the model to account for these at a later stage.

## Extraction Model

We begin with a model where a single searcher performs an  $\text{MEV}$  extraction, sharing proceeds with the block miner via *gas fees only*. We further assume that miners order transactions by descending gas price, which approximates well their optimal strategy and is the default in Ethereum node software.

We decompose the opportunity's  $\text{REV}$  in two terms:

$$\text{REV} = \text{REV}_S + \text{REV}_M, \tag{1}$$

where  $\text{REV}_S$  is the value that goes to the searcher, and  $\text{REV}_M$  is the value that goes to the miner. Note that, as we expand on below,  $\text{REV}$  already encompasses the opportunity's *extraction costs* (i.e. the actual  $\text{REV}$  of an opportunity depends on the gas price of the network at extraction time<sup>[3]</sup>).

Let's dive into the two different terms. The searcher  $\text{REV}$  is composed by:

$$\text{REV}_S = V_{\text{out}} - V_{\text{in}} - g_{\text{MEV}} \cdot s_{\text{MEV}}, \tag{2}$$

where  $V_{\text{out}}$  is the value that flows from the searcher to the blockchain in the transactions performing the extraction (excluding gas);  $V_{\text{in}}$  is the value flowing from the blockchain to the searcher;  $g_{\text{MEV}}$  is the gas price of the transactions; and  $s_{\text{MEV}}$  is their *size*, i.e. the total amount of gas they consumed.  $V_{\text{out}}$ ,  $V_{\text{in}}$ , and  $g_{\text{MEV}}$  are denominated in the base network currency (ETH), while  $s_{\text{MEV}}$  is in units of gas. Separating the gas term from  $V_{\text{out}}$  will be helpful in quantifying the extraction costs, and it is also how we compute  $\text{REV}_S$  in practice.

We use "the blockchain" loosely here to refer to any other addresses (corresponding to smart contracts or EOAs) that are not the EOA of the extracting transactions, or smart contracts controlled by the searcher. Note that identifying these is a heuristic-guided process based on known searcher patterns, and by all means fallible. Also, we stress that any ancillary

transactions related to the  $\text{MEV}$  extraction (like the "meat" in a sandwich attack) are *not* part of the set of transactions that contribute to the above variables.

Turning to the miner side, we have:

$$\text{REV}_M = s_{\text{MEV}} \cdot (g_{\text{MEV}} - g_{\text{eff}}), \tag{3}$$

where  $g_{\text{eff}}$  is the *effective gas price* of the transactions that *would have been included* in the block had the opportunity not been taken.  $\text{REV}_M$  is thus defined to include the *opportunity cost* the miner incurs by including the  $\text{MEV}$ -extracting transactions.

As transactions in the mempool are ephemeral, it is impossible to measure  $g_{\text{eff}}$  *a posteriori* from only blockchain data and logs. We resort to an approximation that also serves as a lower bound for the value realized by the miner:

$$\text{REV}_M \approx s_{\text{MEV}} \cdot (g_{\text{MEV}} - g_{\text{tail}}), \tag{4}$$

where  $g_{\text{tail}}$  is the gas price of the last transaction in the block<sup>[4]</sup>.

Plugging equations (2) and (4) we get for the total realized value:

$$\text{REV} \approx V_{\text{in}} - V_{\text{out}} - s_{\text{MEV}} \cdot g_{\text{tail}}. \tag{5}$$

In this rendition the miner and searcher roles are blurred, but we can clearly identify the *extraction costs* of the opportunity, given by the subtrahend  $s_{\text{MEV}} \cdot g_{\text{tail}}$ .

Finally, note that the value split between searcher and miner at this stage is entirely decided by the choice of  $g_{\text{MEV}}$ , which is in turn affected by the nature of the opportunity and the presence of other searchers trying to exploit it.

## Including Externalities

The scenario presented so far ignores the fact that while single-searcher successful value extractions exist, they constitute only a small fraction of all  $\text{MEV}$  activity. In practice, each time an opportunity arises in the network, several searchers will try to take it by engaging in Priority Gas Auctions or submitting multiple competing transactions to backrun a target state change. Moreover, sometimes searchers perform onchain calculations to validate the existence of an opportunity ("preflights"). When these checks fail, there is no value extracted, but miners will still benefit from the collected gas fees.

Note, however, that all of this activity is zero-sum: all costs incurred by searchers by competing with each other, validating state, etc., directly becomes miner profit. From the miner's perspective, the fee paid by a searcher's preflight is no different from, say, the one coming from an unsuspecting token transfer. This goes to say that none of this value amounts to a contribution to the  $\text{REV}$ . However, it is of interest of us to quantify it, since it is precisely what generates trouble for regular users of the network not involved in this commerce.

We will then need to take a broader view of  $\text{MEV}$ , and define a new term, the *Extractable Value Cost* ( $\text{EVC}$ ), which summarizes all the extra value *induced* by  $\text{MEV}$  extraction activity, that is not part of the extracted value itself. With this definition, we have:

$$\text{Total MEV Activity} = \text{REV}_S + \text{REV}_M + \text{EVC}. \tag{6}$$

In words, the *total MEV activity* is composed of the extracted  $\text{MEV}$  ( $\text{REV}$ ), divided among searchers and miners), and the global costs of extractable value, going directly to miners and quantified by  $\text{EVC}$ . Note that  $\text{EVC}$  only encompasses *on-chain* activity, ignoring off-chain costs like network bloating due same-nonce transaction replacement.

We can compute the  $\text{EVC}$  as follows:

$$\text{EVC} = \sum_{tx \in X} s_{tx} \cdot g_{tx}, \quad X = \{\text{preflights}, \text{failures}, \text{cancellations}\}. \tag{7}$$

While altogether eliminating  $\text{MEV}$  is likely impossible<sup>[5]</sup>, it is Flashbots' [core mission](#) to reduce the negative externalities of  $\text{MEV}$ -related activities by improving the overall efficiency of the extraction process. The present framework allows us to state this mission concisely as that of reducing  $\text{EVC}$ .

## Flashbots Bundles

Flashbots optimizes  $\text{MEV}$  extraction by allowing searchers to submit bundles of ordered transactions, effectively

transferring part of the ordering privileges from the miners to the searchers. These bundles are included by the miners provided they get more out of them than by mining regular transactions. While searchers can still set a gas fee, the preferred way to pay the miners for inclusion is via a direct coinbase transfer included in the bundle<sup>[6]</sup>.

We thus need to account for this extra term in our miner revenue formula (4):

$$M^{\text{Flashbots}}(\text{REV}) = \text{MEV}_{\text{tail}} + V_{\text{direct}} + V_{\text{out}}$$

where  $V_{\text{direct}}$  is the total value transferred to the coinbase address. Note that this new term doesn't affect the total  $\text{REV}$  as it is accounted for by a corresponding increase of the  $V_{\text{out}}$  term.

Apart from the extra term, we now expand the set of transactions that contribute to the terms in the formula to include all transactions included in the bundle. In particular, if a bundle includes the "meat" in an arbitrage, its size and gas price count towards the extracted value. This ensures we account for the opportunity cost incurred by miners when including bundles instead of regular transactions.

Finally, expression (2) for searcher value stays the same, but also requires expanding the implied transaction set to all transactions in the bundle. The rest of the model remains unchanged by the utilization of Flashbots' bundles. With these changes, the model also encompasses alternative relay services like private mempools.

It is worth noting that Flashbots reducing  $\text{EVC}$  does not imply a decrease in miners' revenue, since their share of extracted value might increase as a result of the extraction optimization. Flashbots executions are deterministic and therefore searchers are likely to pay the miners more than if they risked paying for reverts or cancellations. Finding the revenue sharing equilibrium is ultimately a game-theoretic problem which deserves independent treatment.

## Final Thoughts

By unifying various quantities involved in  $\text{MEV}$  extraction, our framework enables us to pose clear-cut questions like "how big are the costs of  $\text{MEV}$  extraction activities for the network?", or "to what extent has Flashbots helped in reducing these costs?". These and other interesting perspectives on  $\text{MEV}$  extraction based on this framework will be included in coming versions of  $\text{MEV}$ -Explore.

Throughout this work, we have swept under the rug the complications involved in actually measuring the "atomic terms" entering the definitions, like  $V_{\text{out}}$ , or the actual set of  $\text{MEV}$ -extracting transactions. This task involves *ad hoc* heuristics for an ever-growing number of  $\text{MEV}$  extraction strategies, and while we strive for establishing a useful classification, there will always be unforeseen opportunities that will elude detection<sup>[7]</sup>.

We pass no judgment as of which types of  $\text{MEV}$  extraction are "good" and which are "bad", although there's certainly ways in which this question could be meaningfully approached ("has an unrelated third party incurred loss due to the extraction?"). We leave this subtle question for future work, but we restate our belief that  $\text{EVC}$  is undesirable and we shall strive to minimize it.

## Join our Efforts!

At Flashbots, we perform research and build systems for mitigating the upcoming  $\text{MEV}$  crisis, and we would love your help. We are a distributed organization with the principles of a [pirate hacker collective](#), and have several [open positions](#). We also issue grants to external researchers doing work aligned with ours, please find out more in our [Research repository](#), or join our [Discord](#)!

Special thanks to Scott Bigelow, Phil Daian, Stephane Gosselin, Alex Obadia, Tina Zhen, and Jason Paryani for welcoming me onboard and all Flashbots' crew for the work and discussions leading to this post.

<sup>[1]</sup>: Note however that most of what is presented here should apply with minor variations to other blockchains that have actors with transaction-ordering privileges. [Work is underway](#) to flesh out these variations.

<sup>[2]</sup>: The definitions here are consistent with a theoretical approach to  $\text{MEV}$  to be published in an upcoming paper (sneak peek [here](#)).

<sup>[3]</sup>: It's up for discussion whether the (non-measurable)  $\text{MEV}$  should also depend on this.

<sup>[4]</sup>: One would need to look into adjacent block gas prices if the entire block was taken up by  $\text{MEV}$ -extracting transactions.

<sup>[5]</sup>: Note however that it can likely be reduced to some extent with smart protocol design, there's some [ongoing efforts](#) to this avail.

<sup>[6]</sup>: A direct coinbase transfer could also be present in a standard, non-bundled extraction, but it would make little sense for the searcher to do it since miners typically only look at gas prices for transaction inclusion and ordering.

<sup>[7]</sup>: See [this paper](#) for a complementary, protocol agnostic approach to  $\text{MEV}$  detection based on flagging certain

types of "transaction dynamics" (displacement, insertion, and suppression).