HOCHSCHULE
MITTWEIDA
University of Applied Sciences

MASTER THESIS

Mr.
Sebastian Wunderlich, B.Sc.

Exploring Maximal Extractable Value in the Ethereum Ecosystem

Mittweida, July 2023

Faculty of **Applied Computer Sciences and Biosciences**

MASTER THESIS

Exploring Maximal Extractable Value in the Ethereum Ecosystem

Author:
Sebastian Wunderlich

Course of Study:
Blockchain & Distributed Ledger Technologies (DLT)

Seminar Group:
BC20w1-M

First Examiner:
Prof. Dr.-Ing. Andreas Ittner

Second Examiner:
Dipl.-Volkswirt Mario Oettler

Submission:
Mittweida, 01.07.2023

Defense/Evaluation:
Mittweida, 2023

Bibliographic Description

Wunderlich, Sebastian:
Exploring Maximal Extractable Value in the Ethereum Ecosystem. – 2023. – 64 S.
Mittweida, Hochschule Mittweida – University of Applied Sciences, Faculty of Applied Computer Sciences and Biosciences, Master Thesis, 2023.

Abstract

Over recent years, Maximal Extractable Value (MEV) has gained significant importance within the decentralized finance (DeFi) ecosystem. Remarkably, within just two years of its emergence, MEV has seen an extraction of approximately 600 million USD - a phenomenon that has sparked concerns regarding potential threats to blockchain stability.
With growing interest in the Ethereum network and the growing DeFi sector, research surrounding MEV has substantially increased. This work aims to offer a comprehensive understanding of MEV, carefully examining Ethereum's microstructure and introducing a Classification Framework to evaluate its effects.
Additionally, this research quantifies the largest types of MEV (Arbitrage, Sandwich and Liquidations) from March 2022 to March 2023. The data are then compared to other sources, revealing a general upward trend, with a particularly noticeable increase in Sandwich Attacks.

Contents

Contents I

List of Figures and Tables III

1 Introduction 1

2 Fundamentals (Ethereum Ecosystem) 2
2.1 Ethereum . . . 2
2.1.1 Proof-of-Stake . . . 2
2.1.2 Proposer-Builder Separation . . . 3
2.1.3 Private Transactions/Relayer . . . 5
2.2 Decentralized Finance . . . 6
2.2.1 Decentralized Exchanges/AMM . . . 7
2.2.2 Lending/Flash Loans . . . 8
2.2.3 Atomicity . . . 9
2.3 Layer 2 Solutions . . . 9

3 Research Objective 11

4 Maximal Extractable Value (Background) 12
4.1 Definition . . . 13
4.2 ETH Microstructure . . . 14
4.3 Types of MEV . . . 15
4.3.1 Frontrunning . . . 15
4.3.2 Sandwich Attacks . . . 17
4.3.3 Arbitrage . . . 19
4.3.4 Liquidations . . . 22
4.3.5 Oracle Attacks . . . 24

5 Quantifying MEV 26
5.1 Technical Approach . . . 26
5.1.1 Data Sources . . . 27
5.1.2 Archive Node . . . 28
5.2 Data Collection . . . 30
5.2.1 Arbitrage.py . . . 31
5.2.2 Sandwiches.py . . . 32
5.2.3 Liquidation.py . . . 33
5.3 Limitations . . . 34
5.3.1 Protocol Selection . . . 34
5.3.2 Methodology . . . 35
5.4 Overall MEV Results . . . 36
5.4.1 MEV Data Script . . . 37
5.4.2 MEV via Flashbots . . . 38
5.4.3 MEV Data Zeromev . . . 39
5.4.4 Comparison . . . 40

5.5 MEV by Protocol . . . 43
5.6 MEV on L2 . . . 44

6 MEV Classification 46
6.1 Efficiency vs. Fairness . . . 46
6.2 Beneficial Effects . . . 48
6.3 Detrimental Effects . . . 48
6.4 MEV Classification Framework . . . 49

7 Conclusion 51
7.1 Limitations . . . 52
7.2 Future Work . . . 52

A Example1 54

B Code and Data 55

Bibliography 58

Statutory Declaration in Lieu of an Oath 65

List of Figures and Tables

List of Figures

2.1 PBS Scheme . . . 4
2.2 Normal transaction vs. private transaction . . . 5
2.3 AMM Dex in DeFi . . . 7
2.4 Conservation function of AMMs . . . 7
4.1 ETH Microstructure . . . 15
4.2 Frontrunning Attack Types . . . 16
4.3 Example of Sandwich Attack on AMM DEXs [16] . . . 18
4.4 Simple Arbitrage Dex . . . 21
4.5 Cyclic Multi Dex Arbitrage . . . 21
4.6 Liquidation Example . . . 23
5.1 Setup MEV Measurement . . . 27
5.2 Node Types . . . 28
5.3 Methods . . . 30
5.4 Total monthly MEV Script . . . 37
5.5 Total monthly MEV via Flashbots . . . 38
5.6 Total monthly MEV Zeromev . . . 40
5.7 Arbitrage Script vs. Zeromev . . . 41
5.8 Sandwich Script vs. Zeromev . . . 42
5.9 Liquidations Script vs. Zeromev . . . 43
5.10 MEV by Protocol . . . 44
A.1 Sandwich Attack . . . 54

List of Tables

5.1 DeFi platforms used by MEV Script vs. Dex and Lending . . . 35
5.2 MEV Dataset Overview . . . 36
6.1 MEV Classification Framework . . . 49
B.1 MEV Data Zeromev . . . 56
B.2 MEV Data Script . . . 57

1 Introduction

Blockchain technology has revolutionized the way we store and transfer value, enabling the creation of decentralized systems that operate without the need for intermediaries. Unfortunately, with the rise of Decentralized Finance (DeFi) a phenomenon called Maximal Extractable Value (MEV) created millions in Revenue for Ethereums Stakeholders and losses for Users, potentially threatening the network.
MEV refers to the amount of value that can be extracted from a given blockchain system. It can be generated through various means, such as frontrunning, transaction ordering manipulation, and liquidity provision. As such, understanding MEV and its potential impacts on blockchain systems is crucial for ensuring the integrity and stability of these networks.
This thesis aims to explore the concept of MEV in depth and examine its effects on different blockchain systems. Specifically, it aims to evaluate the effectiveness of MEV quantification scripts in detecting and classifying MEV, as well as their capacity to shed light on the consequences MEV has on the Ethereum ecosystem. This thesis will not deal with mitigation strategies, an excellent overview can be found here [1].

To develop a deeper understanding of the impact of MEV, it is essential to explore the Ethereum ecosystem (Chapter 2.1) and various techniques in depth. This section starts with a brief introduction and then explains Ethereum's recent developments, such as the transition from Proof-of-Work (PoW) to Proof-of-Stake (PoS) and the introduction of Proposer-Builder Separation (PBS). Further, a deeper look at Relayer is made as they play a crucial role for MEV. Continuing the discussion, the next topic is decentralized finance (DeFi) with a specific emphasis on Decentralized Exchanges (DEXs) and their innovative Automated Market Maker (AMM) structure. We will also explain the concepts of Flash Loans and Atomicity, highlighting their significance. Lastly, attention is directed towards Rollups, which are regarded as the most promising Layer 2 solution for scaling blockchain systems. Despite their potential, the implications of Rollups for Maximal Extractable Value (MEV) have received relatively limited attention in current research. Next, the concept of Maximal Extractable Value (MEV) will be introduced (Chapter 4), aiming to provide a comprehensive and suitable definition for it. The concept ETH Microstructure will be introduced as a graphical approach to quickly grasp ethereums moving parts, which play a crucial role in the MEV landscape. Furthermore, the significant types of Maximal Extractable Value (MEV) are Arbitrage, Sandwich, and Liquidations. Additionally, a novel source of MEV is explored, referred to as Oracle Attacks. The script developed by Weintraub et al. [2], inspired by Flashbots mev-inspect, is used for the analysis (Chapter 5) of 2.5 million Ethereum Mainchain Blocks. The objective is to quantify Maximal Extractable Value (MEV) related to Arbitrage, Sandwich, and Liquidations within the timeframe of 03/2022 to 03/2023. The results are compared with scraped results from Zeromev [3].

Lastly, the economic implications of Maximal Extractable Value (MEV) are explored (Chapter 6) at both network and user levels. This analysis aims to contrast the beneficial aspects of MEV with its potential detrimental effects. This discourse results in the development of an MEV Classification Framework. Finally, we come to a conlusion (Chapter 7) with an outlook on future research.

2 Fundamentals (Ethereum Ecosystem)

In this chapter, the focus is on the complex dynamics of the Ethereum ecosystem, centering on essential elements such as Proof-of-Stake (PoS), Proposer-Builder Separation (PBS), and Private Transactions. These components play a significant role in the realization of Maximal Extractable Value (MEV). As Ethereum transitions from the traditional PoW mechanism to PoS, a clear understanding of these elements is vital to fully grasp the nuances of MEV. Additionally DeFi is briefly discussed, emphasizing Automated Market Makers (AMMs) in decentralized exchanges, the innovative concept of flash loans, and the critical aspect of Atomicity. Lastly, the emerging concept of Rollups is introduced as increasingly essential solutions for managing rising transaction loads.

2.1 Ethereum

Ethereum operates as a decentralized platform that empowers individuals to engage in transactions and utilize smart contracts, eschewing the need for a central governing body. The bedrock of this platform is distributed ledger technology, which is sustained by a global network of autonomous nodes. Every one of these nodes possesses an individual copy of the Ethereum blockchain, a universally distributed transaction record within the network.

The execution of transactions is handled at the network's peer-to-peer layer. Here, each transaction is received and authenticated by every node. Once a transaction successfully undergoes validation, it is assimilated into the blockchain and henceforth remains a permanent fixture in the ledger's historical chronicle. Originally, Ethereum employed a consensus mechanism called Proof-of-Work (PoW), which engaged miners in a competitive race to resolve complex mathematical equations, a prerequisite to appending new blocks to the blockchain. However, in September 2022, Ethereum transitioned to Proof-of-Stake (PoS). Under PoS, the voting power in consensus decisions is proportional with a participant's individual stake in the system. Ether, Ethereum's native digital currency, serves numerous functions, inclusive of covering transaction fees. On Ethereum, smart contracts represent self-fulfilling programs that permit all participants to confirm the execution of a computer program. This capability facilitates the execution of contractual agreements devoid of centralized trust requirements. Ethereum Virtual Machine (EVM) is responsible for executing these smart contracts [4].

Ethereum is under constant development. The focus will lie on recent changes and nuances which have an impact on how MEV is extracted.

2.1.1 Proof-of-Stake

On September 15th, at `Block:15537393`, a significant upgrade called `The Merge` took place in the Ethereum blockchain. This pivotal shift fundamentally modified the block generation mechanics, transitioning from the energy-intensive Proof-of-Work (PoW) mining protocol to the more sustainable Proof-of-Stake (PoS) method of block validation.

The transition to Proof-of-Stake presented a new system for block production in Ethereum. Under this new regime, users could stake their Ether to become validators and propose new blocks, rather than solving complex cryptographic puzzles. Instead of computational power restricting the creation of potentially harmful blocks, as in the PoW system, the PoS system limits users' consensus power to the amount of Ether they have staked. Misbehaving participants face slashing, where their staked Ether is destroyed, resulting in a substantial economic penalty.

The consensus for blocks now occurs on the Beacon-Chain, eliminating the reliance on hash puzzles. Yet, the basic operations of the Execution Layer, where transactions not related to consensus are evaluated and executed, have been preserved.

The Merge, as this shift is often called, primarily aimed at altering Ethereums consensus mechanism without introducing significant changes to the execution layer's mechanics, which impacts many users. The overall trends and patterns concerning gas usage, transaction count, and DeFi activity remained largely unchanged, despite the fundamental modifications to the underlying consensus mechanisms. However, two broad categories of changes were observed. First, Blocks became more predictable, resulting in denser distributions across metrics such as gas usage, gas fees, transaction counts, and others. Second, there is a reduction in Empty/Bad Blocks. The merge seems to have sidelined miners who were producing blocks with no transactions or failed transactions [5].

With Ethereum's transition, a new participant, called Block Builders, was introduced to the network. Previously, both block building and proposing were handled by the same actor under the PoW system. However, The Merge brought a significant influx of new validators who often lacked the technical skills for constructing optimized blocks. This led to the emergence of Block Builders, specialists who competitively construct blocks on behalf of validators. This division of roles is termed Proposer-Builder separation (PBS), which is currently implemented via a protocol adjunct called MEV-Boost which will be discussed in the next chapter [4].

2.1.2 Proposer-Builder Separation

Subsequent to Ethereum's shift from the PoW protocol to the PoS system, the Proposer-Builder Separation (PBS) mechanism has ascended to a leading role in the market for Ethereum block construction and will eventually be implemented into the protocol. The Proposer-Builder Separation (PBS) mechanism is a transformative system introduced in the Ethereum network. It diversifies the roles of participants and effectively diminishes the influence of PoS proposers in the selection of transactions for their proposed blocks. The PBS mechanism encompasses three distinct roles: searchers, builders, and relays, each with specific duties. The key goals are to evaluate the effectiveness of the PBS mechanism in decentralizing block validation and in preventing censorship within the Ethereum network [6].

Searchers in the PBS ecosystem are tasked with pinpointing MEV opportunities. They compile transaction bundles that leverage these opportunities for profit. These bundles are then transferred to builders, whose role is to construct blocks that maximize revenue generation.

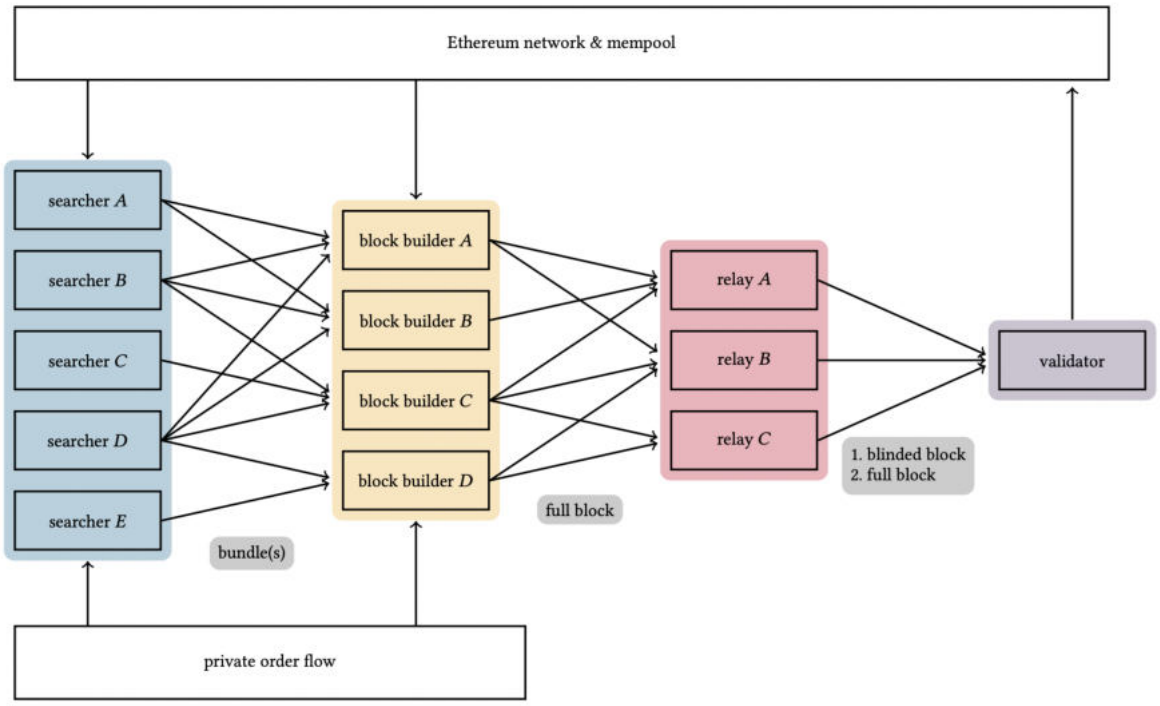


Figure 2.1: PBS Scheme

Simultaneously, relays evaluate these constructed blocks to identify the most profitable ones. These are then forwarded to the proposers (validators). Equipped with MEVBoost [7] Flashbots software, the designated block proposer can utilize externally constructed blocks to carry out its function effectively. This workflow is illustrated in Figure 2.1.

Previously known as miners, proposers (validators) now have a choice to make. They can either propose blocks that have been recommended by a relay or opt to propose blocks that they have constructed themselves. The latter may contain transactions sourced either from Ethereum's public peer-to-peer layer or directly transmitted to them by other entities. In this scenario, pending transactions are typically given due consideration [8].

In contrast to its great promises Heimbach et al. found, that despite initial high expectations, the research finds considerable centralization among builders and relays in the PBS system. The mechanisms ability to fulfill its intended objectives, such as enhancing block profitability for hobbyist validators and preventing censorship, is also investigated. Findings suggest that while PBS does provide all validators with equal opportunities to access optimized blocks, it could unintentionally stimulate censorship. Furthermore, the reliability of relays is brought into question, with observed inconsistencies in their performance and commitments. Particularly, instances have been noted where proposers fail to receive the full value promised, and the pledged censorship or filtering capabilities of the relays have considerable limitations [6].

Also, Waehrstaetter et al. highlights an inherent paradox in PBS: while proposers must commit to a block header without prior knowledge of its content, they need to trust block builders and relayers. This dynamic could appear to violate the decentralized spirit of Ethereum. There's an underlying risk that these actors might manipulate MEV rewards via generalized front-running strategies, thereby challenging the very rationale for PBS's existence [8].

2.1.3 Private Transactions/Relayer

Private transactions, are directly forwarded to miners, skipping the usual process of appearing in the mempool. Subsequently, miners prioritize these private transactions over regular ones when forming new blocks. However, this privilege of increased privacy and priority comes at an additional cost, as these transactions often require an extra fee paid directly during the transaction's execution [9]. The two different stages of mining a normal transaction in comparison to a private trasanction are shown in Figure 2.2.
Initially, miners introduced dedicated ports (private channels) enabling traders to submit transaction data directly. Centralized intermediaries, such as Flashbots, later managed sealed-bid auctions for transaction inclusion, which facilitated a rapid market penetration. Further private services were offered by bloXroute [8] [6].

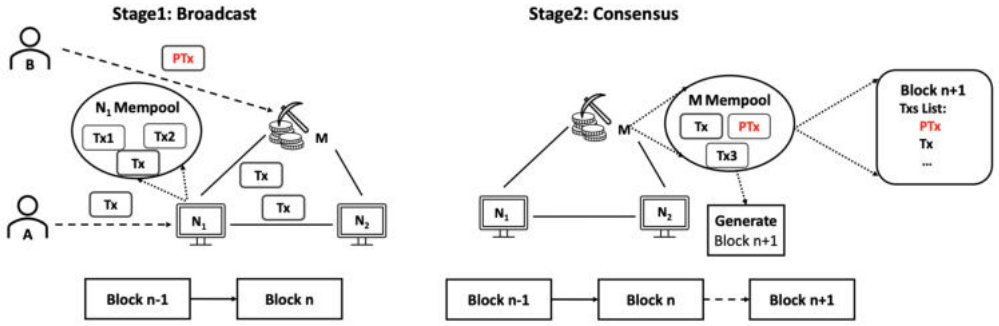


Fig. 1: The two stages of mining a normal transaction (Tx) vs. a private transaction (PTx) in Ethereum. User A sends a normal transaction (Tx) to a node (N_1), which is then put into the mempool of this node. Generally, in Stage 1 N_1 broadcasts Tx to its peers, N_2 and M (M is the miner). However, the Stage 1 is different for a private transaction (PTx) sent from User B, in which PTx is sent directly to M. In Stage 2, M solves the Proof-of-Work puzzle, and includes PTx and Tx in the newly mined block. In particular, PTx is put in front of Tx.

Figure 2.2: Normal transaction vs. private transaction

Relay systems play a pivotal role in the blockchain ecosystem by providing a private pathway for transactions, facilitating direct submissions from users to miners. They act as intermediaries between builders, who submit blocks, and proposers, also known as validators. A critical function of these relays is to maintain block privacy until the validator commits to its inclusion by signing the block's header. Essentially, relays hold blocks from builders in escrow for validators, transmit the header of the most profitable block to validators, and then forward the full block after receiving a signed header [6].

The relay market used to be dominated by MEV Boost and Flashbots, but over time, the market diversified [10] with more entities introducing their own relay systems. For example, Blocknative operates a relay system with its own implementation, called Dreamboat, while most other relays are based on MEV Boost, a product of Flashbots. These relay systems, despite their distinct implementations, adhere to Flashbots' relay API specification [6]

Research has explored the impact of Maximal Extractable Value (MEV) relays, often referred to as "dark venues," on blockchain networks. Findings from Capponi et al. suggest that these relays do not effectively diminish frontrunning risks nor do they lower transaction costs. Contrarily, MEV relays seem to enhance miners' payoff, with empirical evidence indicating an

increase in miner profits associated with MEV activity [11]. In agreement, Qin et al. observed that MEV relays failed to alleviate network congestion [12]. Lyu et al. shed light on an additional risk tied to private transactions. In these instances, miners can effortlessly discern and pick out the most rewarding private transactions, consequently discarding the less beneficial ones. They can then integrate these selected transactions into the new block. This practice amplifies the block rewards linked with private transactions, thereby possibly serving as a stimulant, spurring the escalation of undercutting attacks [9].

Moreover, an analysis of private transactions versus public mempool transactions provides insight into the use of MEV auction platforms for MEV activities [1]. The analysis showed a significant preference for MEV auction platforms over the public mempool due to the privacy and atomicity these platforms provide. More than 71% of arbitrage transactions and 83% of liquidation transactions were conducted via MEV auction platforms. For sandwich attacks (see 4.3.2), which require atomicity, a remarkable 94.04% utilized private transactions. These findings align with previous research, indicating a low rate of sandwich attacks conducted using the public mempool [2].

These findings collectively suggest that while MEV relays or "dark venues" and MEV auction platforms do not necessarily mitigate the risks or costs associated with blockchain transactions, they do offer significant privacy and atomicity advantages. As a result, they are heavily favored [13] [1] specific transaction types, notably arbitrage, liquidation, and sandwich attack transactions.

2.2 Decentralized Finance

This chapter provides an introductory overview of DeFi in the Ethereum ecosystem. By examining the functionalities of Decentralized Exchanges (DEXs), lending platforms, atomic swaps, and flash loans, understanding of MEV dynamics and their implications for stakeholders in the DeFi space is enhanced. Figure 2.3 shows key microstructure in the DeFi market. One of the reasons MEV became such a dominant topic is the introduction of financial applications on the Ethereum Ecosystem. With these decentralized applications, certain behaviors 4.3 have become interesting and profitable for Ethereum's stakeholders.

Ethereum is still dominant in DeFi, making it an ideal ecosystem for studying MEV. Our discussion begins with DEXs, specifically Automated Market Maker (AMM) DEXs, as this design is used by the leading Dex Uniswap and accounts for most of the volume (see 5.5). These inventions have revolutionized asset exchange through liquidity pools, eliminating traditional order books. AMM DEXs also serve as a significant source of MEV revenue.

Finally, atomic swaps, a trustless, instant mechanism for cross-chain asset transfers are discussed. These swaps facilitate the exchange of assets across blockchains, bypassing centralized entities. Flash loans, which allow users to borrow funds without collateral, repaying the loan within a single transaction will be briefly touched.

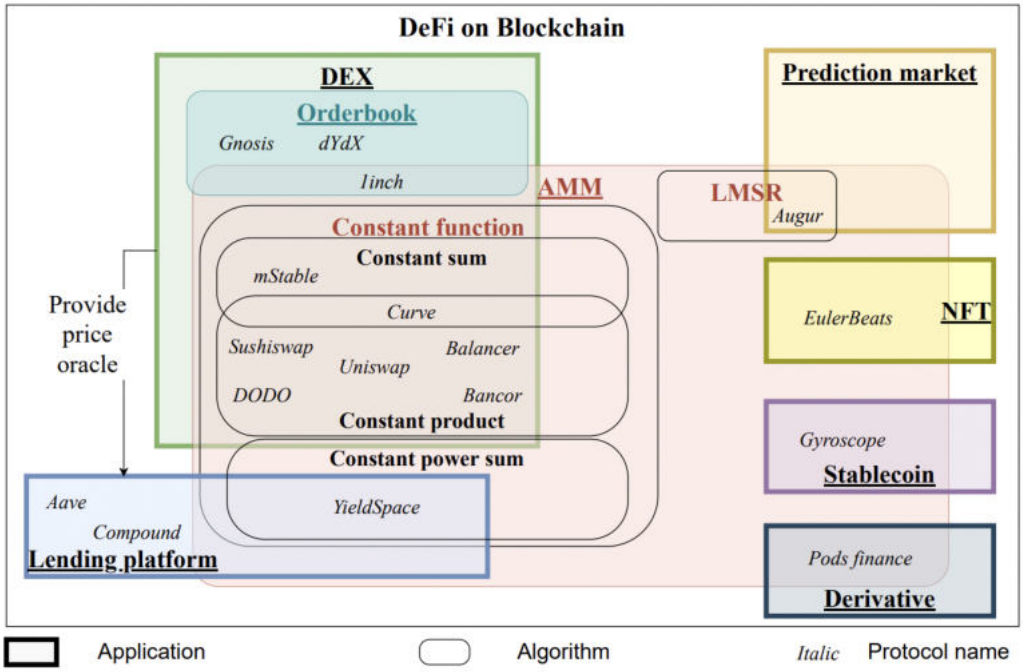


Figure 2.3: AMM Dex in DeFi

2.2.1 Decentralized Exchanges/AMM

Decentralized exchanges (DEXs), key to Decentralized Finance (DeFi), facilitate peer-to-peer digital asset trading, with all operations verifiable on-chain. Initially restricted to their native blockchain's assets, advancements such as wrapped tokens and cross-chain solutions have broadened their functionality. The DEXs employ various models for price discovery, among which Automated Market Makers (AMMs) have gained significant attention.
Unlike traditional market makers, AMMs use algorithms to provide liquidity. They offer multiple benefits including support for lesser-known assets, democratizing market making, and obviating the need for on-chain order books. In the AMM model, liquidity providers deposit assets into a smart contract and receive tokens representative of their stake. These reserves interact with trades, impacting one asset's reserves while affecting another's. A transaction fee is proportionally divided among the providers based on their contribution [14].

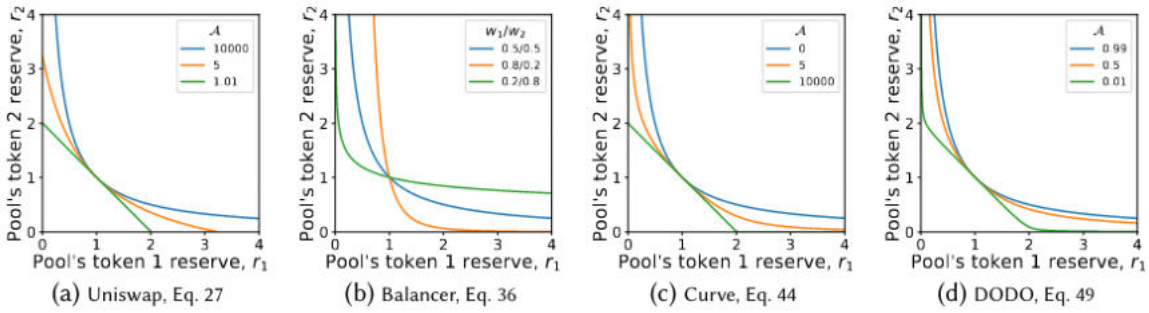


Figure 2.4: Conservation function of AMMs

AMM asset prices are algorithmically determined, making them susceptible to significant shifts due to trades, especially in illiquid pools. This volatility invites arbitrage opportunities, aligning AMM prices with the broader market. However, shifting liquidity may alter the ratio of tokens returned to providers upon withdrawal. Further technical analysis of AMM design is discussed in [15]. Uniswap leverages a constant product function to determine asset prices. The constant product formula used in automated market makers is given by $x \cdot y = k$, where x and y are the amounts of the two tokens in the liquidity pool, and k is the constant product. More general and following the notation of Figure 2.4 conservation function in automated market makers is described by $R_1 \cdot R_2 = K$, where R_1 and R_2 are the reserves of the two tokens, and K is the invariant product.

This is also the most prominent function. Xu et al. show and generalizes the product market function (conservation function) and provide a generalized formal proof. They show that this generalized framework can be used to present and discuss various AMM protocols [15].

As an effect the transaction exchange rate on Automated Market Makers (AMMs) is dictated by set algorithms and the reserves of market liquidity. Asset prices rise with purchase orders, while sell orders lead to a decrease in these prices [16].

2.2.2 Lending/Flash Loans

Protocols for Loanable Funds enable lending and borrowing of blockchain-based assets, creating a decentralized market for these crypto-assets. These protocol pool funds in a smart contract, which comprises the market's liquidity, equivalent to the total supply and borrowings of a token. An entity can borrow directly from these reserves, given sufficient liquidity in the token market. The borrowing cost is determined by the prevailing market interest rate.
Protocols for Loanable Funds loans are predominantly of two types: over-collateralized loans and flash loans. In an over-collateralized loan, borrowers pledge an asset value greater than the loan's value, ensuring loan repayment and risk mitigation for the lender, often a smart contract. If the collateral value dips below a specified liquidation point, "liquidators" or keepers have the opportunity to purchase the collateral at a reduced price and settle the borrower's debt. Contrastingly, flash loans are uncollateralized and exist within a single transaction span. These loans mandate repayment of the borrowed sum and interest by the transaction's end. Such loans leverage the blockchain's atomicity, failing the transaction if the loan is not repaid within it. Flash loans find utility in decentralized exchange arbitrage and collateral swapping, but they are also susceptible to misuse [14].

Flash loans in the Decentralized Finance (DeFi) sector take advantage of the principle of transaction atomicity, where if a borrower fails to repay the loan within the transaction's duration, the blockchain state remains as it was prior to the loan transaction. This feature guarantees lenders that borrowers can't default, despite the lack of collateral.

While flash loans are primarily employed in DeFi arbitrages and liquidations, minimizing the financial risk associated with holding upfront assets, they have also been associated with DeFi attacks. These attacks, facilitated by flash loans, led to losses exceeding $100 million

in 2020. Flash loans are not the direct vulnerabilities behind these attacks, but they offer adversaries instantaneous access to substantial capital with negligible initial cost, limited to the blockchain transaction fees. Such instantaneous loans have no equivalent in Centralized Finance (CeFi).

While flash loans are not the direct cause, they contribute to DeFi attacks by providing adversaries immediate access to vast amounts of capital. Essentially, these loans democratize capital access, lowering the entry barriers to a market typically exclusive to a select few in CeFi [17].

In addition, Qin et al. showed that atomictiy and flash loans increases the arbitrage revenue and how flash loans are manipulated as mechanisms for conducting financial attacks, leading to significant cryptocurrency outflows from DeFi systems. [18]

2.2.3 Atomicity

In the realm of decentralized finance (DeFi), the concept of atomicity plays a pivotal role, especially with respect to transaction execution and flash loans, increasing the arbitrage revenue [18]. Defined in the computer science context, atomicity stipulates that a transaction must either succeed completely, updating the state, or fail altogether, leaving the state unaltered, so as to avoid any invalid state scenarios [14]. Specifically, atomic transactions must be initiated and concluded within the same block.

This feature of atomicity is largely absent from Centralized Finance, whereas in the DeFi ecosystem, it's harnessed efficiently through smart contract technology to ensure consistent and reliable transactions. For instance, when a contract encounters an error, all the state changes are reverted, guaranteeing that transactions are atomic – they either fully complete or have no effect on the state and are entirely reverted. This assurance remains valid, irrespective of the number of contracts involved or their functions when called [17].

Apart from reliable transactions, atomicity also enables atomic swaps, or atomic cross chain trading. This technology employs smart contracts to facilitate the exchange of one cryptocurrency for another without necessitating a centralized exchange. The smart contract autonomously executes the trade when predetermined conditions, like receipt of the correct amount of cryptocurrency from the counterparty, are met.

2.3 Layer 2 Solutions

Layer 2 (L2) represents a comprehensive category encompassing distinct Ethereum scaling solutions. It denotes an autonomous blockchain that extends the functionalities of Ethereum while benefiting from the security assurances provided by the Ethereum network. The combination of Optimism and Arbitrums transactions surpassed Ethereum's weekly transactions in 2022 and is on the rise [19].

This paradigm addresses network congestion by transferring the bulk of computation and state storage off-chain, maintaining only necessary transaction data on-chain. The heart of this system is an on-chain smart contract which holds the state root. This root, the Merkle root of the Rollup state, is crucial in fraud detection procedures. The characteristic feature of Rollups is their method of processing transactions in batches.
Rollups can be categorized into two major types: Optimistic Rollups and Zero-Knowledge (ZK) Rollups. Optimistic Rollups follow a principle of presumed validity until confronted by an on-chain fraud proof. Following the publication of a batch, any participant can present a fraud proof during a fixed dispute period, indicating a faulty state transition within the batch. If the proof is found credible, the invalid batch and all ensuing batches are rolled back. It's important to note that a more extended dispute period creates a longer finality window. However, it can also function as a defense mechanism against potential censorship attacks. On the other hand, ZK Rollups include a validity proof, known as a ZK-SNARK, which is calculated off-chain. This proof confirms the accuracy of the new state. The inclusion of ZK-SNARKs allows for instantaneous batch validation, consequently leading to reduced finality time [20].

The sequencer plays a pivotal role in executing and storing Layer 2 transactions submitted by users and is also tasked with submitting the state root of the batch of corresponding state transitions on Layer 1. The selection of sequencers in Layer 2 can be approached in various ways. The most prevalent among these is the centralized sequencer, characterized by a single operator capable of submitting batches. Despite its operational efficiency, this model introduces a certain degree of centralization and thus potential censorship in Layer 2 solutions due to its reliance on a single entity for liveness. This centralization aspect could potentially be a point of concern in a technology that thrives on its decentralized nature [20] [21].

3 Research Objective

The principal research aim of this thesis revolves around assessing the effectiveness of MEV detection scripts, such as the one developed by Weintraub et al. [2] which in turn is inspired by Flashbots mev-inspect [22], in accurately detecting, classifying, and measuring the impact of diverse types of MEV on the Ethereum ecosystem. Thus, the key research question that this work seeks to answer is:

> *Research Question: "How effective are MEV quantification scripts in identifying and categorizing MEV, and in revealing its impact on the Ethereum ecosystem?"*

The scope of this thesis will primarily encompass three types of MEV: Arbitrage, Sandwich Attacks, and Liquidations. Arbitrage refers to the profit obtained from capitalizing on price discrepancies of an asset across various exchanges or markets. Sandwich Attacks occur when an adversary strategically places a transaction between two legitimate transactions to gain an unfair advantage. Liquidation includes the profits derived from liquidating undercollateralized loans on DeFi platforms. The quantification and in-depth analysis of these MEV types form the core of this research.

An essential aspect of this study involves the introduction of the term `ETH Microstructure`. This term will facilitate a granular understanding of the components that generate and influence MEV, serving as a cornerstone for interpreting the obtained data. This comprehension will later shed light on the general rise of MEV and the increased frequency of Sandwich Attacks on the Ethereum Mainchain.

Another significant objective of this work is to create a classification framework to categorize MEV types based on their ethical ramifications. This framework will help distinguish between neutral MEV (like legitimate arbitrage profits) and toxic MEV (like Sandwich Attacks that compromise the network's integrity). The identification and categorization of MEV types serve as stepping stones towards formulating strategies that can mitigate their adverse effects on the network.

In conclusion, this thesis aims to increase the understanding of MEV's effects on the Ethereum network, and its findings could potentially promote the development of strategies to reduce the detrimental impacts of MEV, thereby enhancing the overall security and fairness of the Ethereum network.

4 Maximal Extractable Value (Background)

Maximal extractable value (MEV) refers to the potential financial gain that can be obtained from exploiting the vulnerabilities or inefficiencies in a given system. It was first identified as a problem as early as in 2014 by a Reddit user [23].
The concept of Maximal Extractable Value (MEV) encompasses the additional value that can be obtained from block production, going beyond the standard block reward and gas fees. MEV extraction involves manipulating transactions within a block by including, excluding, or changing their order. Participants in this process, namely searchers, builders, and validators, have varying levels of control and dependency. Searchers rely on builders to include their MEV bundles, avoiding theft or omission, while builders depend on validators to incorporate these MEV bundles into blocks [6]. In the Ethereum ecosystem, MEV has garnered significant attention due to its potential impact on the overall security and stability of the network [24]. In addition, MEV offers miners an extra source of financial incentives that can be utilized for bribery [25] and undercutting attacks [26]. These attacks involve adversarial miners intentionally providing monetary rewards, such as extractable MEV and transaction fees, on a forked blockchain to attract mining power. The concentration of revenue objectives by MEV relayers further amplifies the potential value that miners can extract, thereby increasing the risks associated with consensus layer forks [12].

In response to frontrunning attacks, several recent studies [27] [28] [29] have focused on addressing the transaction reordering issue at the consensus layer. These efforts involve the development of new consensus protocols that possess fair ordering properties, aiming to mitigate the impact of such attacks. Additionally, alternative techniques seek to conceal transaction contents until they are sequentially ordered, thus providing a means to prevent certain forms of adversarial frontrunning. Fair ordering emerges as a central problem in these investigations and has been long discussed, with numerous authors attempting to propose solutions to this challenge [30].

Further, another fact is that there is a correlation between moments of crisis and substantial increases in MEV payments when compared to the baseline [8]. The most prevalent forms of MEV include sandwich attacks, arbitrage, and liquidations [12] [2].The impact of MEV on the Ethereum ecosystem, whether beneficial or detrimental, remains a subject of ongoing debate. Nonetheless, MEV currently represents a significant portion of block rewards [6].

In this chapter, the fundamental concepts behind MEV, including its definition and various types are explored. The chapter also examines the potential negative effects associated with MEV and analyzes the extent to which MEV can pose challenges. MEV exists on all smart contract blockchains where there is a party responsible for transaction ordering, including non-miner actors such as validators in ETH2.0 and rollup providers on Rollups. MEV extraction on Ethereum so far has been primarily conducted by non-mining DeFi traders and bot operators.

4.1 Definition

It is somewhat not a trivial task to define MEV and the definiton has evolved over time. A generic and thorough definition of extractable value is difficult to establish because permissionless cryptocurrencies are complex systems with many different actors and potential forms of extractable value.
Most of the analysis is done by analyzing historical Ethereum blocks and associated transactions in order to identify profitable trades on the blockchain [24] [12]. Additionally, there is currently a lack of consensus among researchers and practitioners on how to define [12] [31] [30] and measure extractable value in these systems. Therefore, developing a comprehensive and all-encompassing definition for extractable value in permissionless cryptocurrencies is challenging.
It is even argued that a quantification is not possible due to the very nature of permissionless blockchains [31]. Most common types of MEV and how they arise will be discussed in detail in Section 4.3.
The Term Miner Extractable Value was first introduced by Daian et al. who studied arbitrage bots and the behaviour of frontrunning in decentralized exchanges and showed that blockchains do not create fair and transparent trading systems. Miners can reorder transactions and add their own, which have the potential to be higher than just the block reward [24]. The first Definition ever states that:

> MEV refers to the total amount of Ether miners can extract from manipulation of transactions within a given timeframe, which may include multiple blocks' worth of transactions. In systems with high MEV, the profit available from optimizing for MEV extraction can subsidize forking attacks of two different forms [24, p.14].

It is common knowledge that block proposer$(\text{usually miners or validator in PoS})$ can strategically order, censor and include transactions in a block. This gives a rational proposer an incentive to gain more economic value [30].
However, network participants realized that the extraction was not only limited to block producers. The utilization of strategies such as spamming transactions or outbidding competitors by blockchain users and bots, commonly referred to as searchers, also allows for the extraction of miner-extractable value. Thus the concept of miner extractable value has been expanded to maximal extractable value acknowledging the fact that extraction is not limited to solely block proposers [32][22].
Babel et al. [30] introduced the Clockwork Finance Framework, a formal verification framework for reasoning about the economic security properties of decentralized finance (DeFi) smart contracts. They also introduce a new approach for MEV, namely extractable value (EV) as a new formal notion of economic security in composed DeFi contracts. They define EV as:

> maximum value, expressed in terms of the primary token, that can be extracted by a given player from a valid sequence of blocks that extends the current chain. Formally, for a state s, and a set $\mathcal{B}$ of valid block sequences of length k, the EV for

a player P with a set of accounts A_P is given by:

$$\mathrm{EV}(P, \mathcal{B}, s) = \max_{(B_1,\ldots,B_k)\in\mathcal{B}} \left\{ \sum_{a\in A_P} \begin{array}{c} \text{balance}_{\,k}(a)[0] \\ -\ \text{balance}_{\,0}(a)[0] \end{array} \right\}.$$

where $s_0 = s = (\ \text{balance}_{\,0}, \text{data}_{\,0})$, $s_i = \mathrm{action}\,(B_i)\,(s_{i-1})$, and $s_k = (\ \text{balance}_{\,k}, \text{data}_{\,k})$ [30, p.9-10].

Others argue that it is difficult to establish a precise definition for the various forms of extractable value in permissionless cryptocurrencies. This is due to the lack of a universally accepted framework for measuring extractable value, as well as the inherent probabilistic nature of these systems. Additionally, the extractable value of different actors, such as users, cannot be captured by a narrow definition of extractable value. Furthermore, the precise calculation of extractable value in permissionless cryptocurrencies is challenging due to the presence of imperfect information and the potential for cryptographically interlinked cryptocurrencies. Estimating the extractable value of any particular actor is difficult or even impossible in practice [31].
Even if the definition and measurement of MEV is a challenging task this work will follow the common used term in practice maximal extractable value (MEV). Therefore a definition is used that is rather broad and tries to capture any activity in a system:

> MEV is the general classification of techniques and methodologies to extract data from the blockchain that is (monetary) valuable.

This definition provides a broad understanding of MEV as it encompasses various techniques and methodologies to extract valuable data from the blockchain. It serves as a comprehensive classification that can summarize the different types of MEV.

4.2 ETH Microstructure

In traditional markets microstructure is a branch of finance concerned with the details of how exchange occurs in markets. Market microstructure economics focuses on how prices adjust to new information and how the trading mechanism affects asset prices [33]. Being inspired by traditional finance literature the term ETH Microstructure is introduced. A new branch of research that is concerned with the details of exchange that occurs on the ethereum blockchain. One could generalize this Term to Blockchain Microstructure. A similar concept, focusing on the chain of series of actions that facilitate users in converting their intentions into finalized state changes, considering the presence of MEV is called MEV Supply Chain [34]. However, this concepts focused more on the user data. Xu et al. also presented a graphical display of microstructure focusing on DeFi design (see: Section 2.2.1).

The ETH Microstructure approach helps determine where potential bottlenecks and vectors are occurring, and where MEV can be generated. It shows the value flow and graphically helps understand the most important key parts of the ever-changing MEV landscape, with a holistic approach in capturing not only blockchain-relevant actors but also potential threats from outside the system.

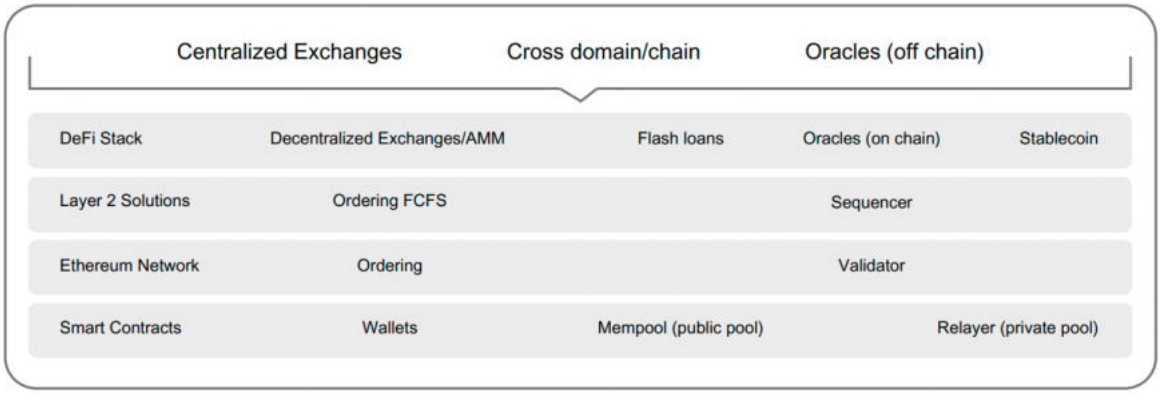


Figure 4.1: ETH Microstructure

Figure 4.1 shows the nuances of the Ethereum network, including the various layers and instances that influence MEV. The figure also highlights how external information contributes to the emergence of MEV opportunities. In any given blockchain network, the landscape is essential for MEV. Not only are the rules (e.g., ordering) of the underlying network important, but also networks on top of the networks (L2s) and the design of applications, most importantly Dexs. Additionally, the design of sources of prices (Oracles) plays a crucial role. Each of these components will be discussed in the following chapters.

It is essential to determine new ways of MEV by understanding all the little details of the interplay between different actors and instances in the ecosystem and beyond. The microstructure is ever-changing, and the fine details make the difference between an MEV opportunity or not. When applied to Ethereum, "ETH Microstructure" might refer to the detailed examination of how trades and transactions occur on the Ethereum blockchain, especially focusing on where Maximal Extractable Value (MEV) can arise. This would likely involve studying the process and impact of transaction ordering, block propagation, and consensus protocols, among other aspects.

4.3 Types of MEV

This chapter aims to provide a comprehensive overview of the different types of MEV that exist within the Ethereum ecosystem, specifically focusing on MEV Arbitrage, Liquidations, and Sandwich attack as they are the biggest form of MEV [12] [35]. Relevant techniques such as Frontrunning and Backrunning are also presented. In addition, an often overlooked form of MEV known as Oracle Attacks is explored, which may become more relevant in the future. While the broad definition of MEV encompasses any Extractable Value within a system, it is important to note that hacks are not traditionally classified as MEV and are extensively discussed in the security literature [36].

4.3.1 Frontrunning

In the Ethereum network, the sequential execution and independent operation of transactions on the Ethereum state make the ordering of transactions within a block crucial. This results in behavior, where users attempt to front-run profitable transactions. Given that all

information in Ethereum is public, front-running can be considered a form of rapid response to complex public information, similar to the behavior of high frequency traders in traditional financial markets [37].
Frontrunning refers to a situation where a miner or a non-miner deliberately places their own transaction ahead of a target victim's transaction within a block. This is achieved by the miner choosing to prioritize their own transaction over the victim's, or by a non-miner increasing the gas price of their transaction to make it more attractive for the miner to include in the block. On the other hand, backrunning involves an attacker wishing for their transaction to be ordered after the victim's. This can be accomplished through similar methods as in frontrunning, where the miner prioritizes their own transaction after the victim's, or the non-miner decreases the gas price of their transaction [2].
Frontrunning has been identified and proven by Daian et al. [24] and first measured with historical blockchain data by [38]. Other studies have quantified frontrunning and show a general increase over the years [12] [37].

Torres, Camino and State [38] define three specific types of frontrunning namely Displacement, Insertion, and Suppression building upon the work of Eskandari, Moosavi and Clark [39].

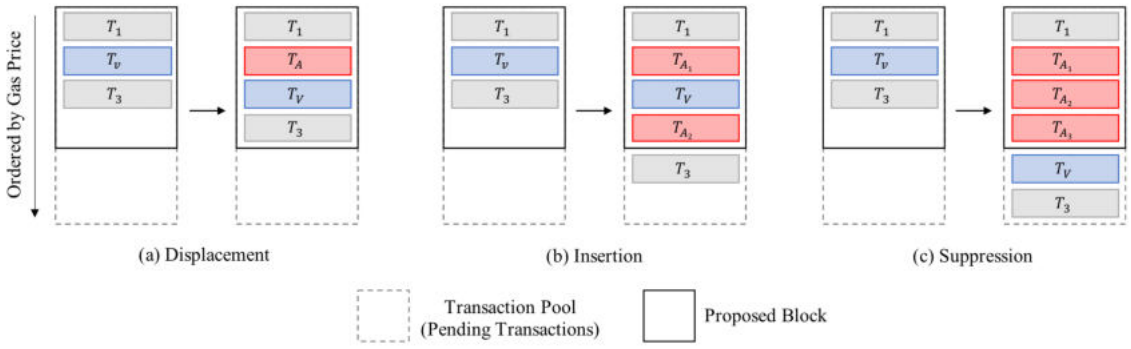


Figure 4.2: Frontrunning Attack Types

Displacement (a) refers to a situation where an attacker, designated as A, identifies a profitable transaction (TV) made by a victim (V) and subsequently broadcasts their own transaction (TA) to the network with a higher gas price. This is done with the intent of having their transaction (TA) prioritized by miners, who will include it before the victim's transaction (TV), effectively displacing the original transaction.

Insertion (Sandwich) (b) In the event that an attacker recognizes a transaction (TX) within the mempool that interacts with a known liquidity pool and is categorized as a swap, they engage in a front-running and back-running strategy. The attacker first front-runs the original transaction by broadcasting their own transaction with identical information but with a higher gas price. Subsequently, the attacker back-runs the original transaction by executing a reverse swap transaction with a lower gas price. This approach allows the attacker to benefit from the differential between the high gas price and the low gas price of the two transactions they broadcast. This has been shown in detail in Section 4.3.2.

Suppression (c) Attackers can interfere with an agent's transaction (TX) from being mined by overwhelming the network with their own transactions with a higher gas price, thus occupying the available block space. If the attacker's transactions all have a higher gas price and there is a sufficient amount of collective computational effort, the agent's transaction may not be processed until these conditions change. This behavior is also referred to as clogging [40].
These are the main types of frontrunning. Different categories and namings exist. Heimbach and Wattenhofer [40] use the term clogging and categorize them as fatal frontrunning, which is the same as Suppression. Also Qin, Zhou and Gervais [12] introduce a new algorithm (transaction replay) which can be seen under category of generalized frontrunning or just frontrunning. However, the original taxonomy by Eskandari, Moosavi and Clark hold and is able to describe frontrunning comprehensive.

4.3.2 Sandwich Attacks

The concept of sandwiching has been a well-established and widely recognized trading strategy in traditional financial markets. Sandwich attacks have gained widespread attention [38] [18] [12] [41] [2] in academia, within a short period of time after being first reported by Zhou et al. [42]. A recent study [16] provides the most comprehensive heurisitc for detection of sandwich attacks. Sandwich attacks happen mostly in automated market maker (AMM) decentralized exchanges (DEXs) [43].
During a sandwich attack, an attacker places two transactions around a victim's regular trade in order to manipulate asset prices and benefit from the victim's loss. The attacker scans the mempool for pending transactions which might be profitable. Due to the design of AMM exchanges (see 2.2.1) the attacker is aware that large transactions will change the price. He then executes a two-step transaction in which they first engage in front-running the large transaction. By doing so, they purchase or sell a quantity of the asset prior to the fluctuation in its price. In the second transaction, they engage in back-running, which allows them to either reacquire the original asset at a decreased price or sell the newly acquired asset for a higher price, thus yielding a profit due to the price difference [2].

However, executing this type of attack can be risky, as a misordering of the transactions can result in a loss for the attacker. Such attacks are commonly carried out through private pools, as those guarantee privacy and atomicity (see Section 2.1.3) [1].

Graphic 4.3 explains an attack with constant product function and no commission fee in great detail. The attacker first front-runs the victim's transaction by placing a purchase order of USDT. This results in a change of the reserves in the liquidity pool from an initial state of 100 ETH/300,000 USDT to 105 ETH/285,714 USDT. Subsequently, the victim transaction takes place, altering the reserves to 125 ETH/240,000 USDT. Finally, the attacker back-runs the victim transaction with a sell order, resulting in the reserves being changed to 118 ETH/254,286 USDT. For a detailed on chain example see Example A.

The victim intends to exchange 20 ETH for USDT at the market rate. In the absence of an attacker, the victim would receive 500,000 USDT. However, a sandwich attacker may submit front-running and back-running transactions to gain 2 ETH as revenue, resulting in the vic-

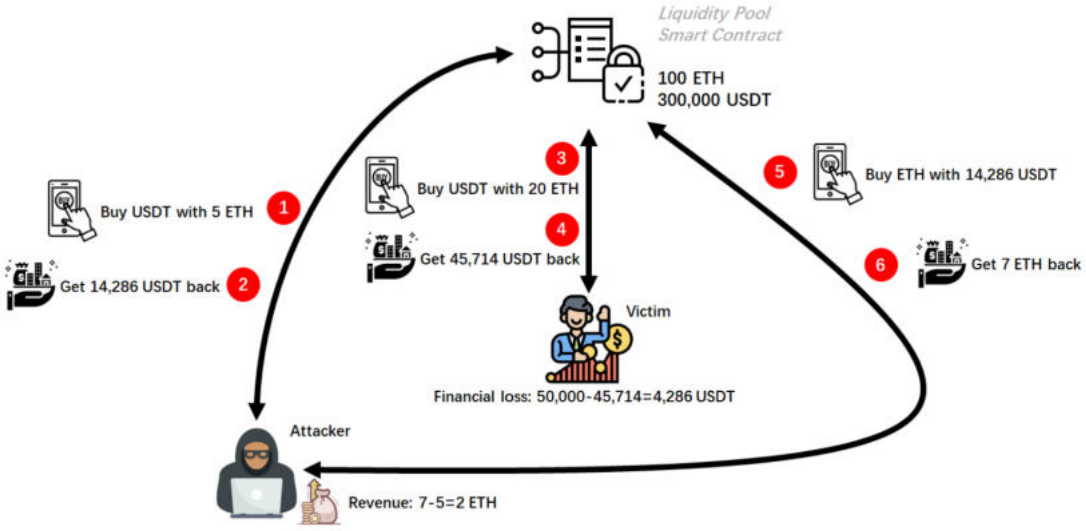


Figure 4.3: Example of Sandwich Attack on AMM DEXs [16]

tim receiving a less favorable exchange rate. The extent of price increase is limited by the chosen slippage rate, which represents the maximum difference between the ideal market price and the actual exchange price. If the market price increases excessively before the victim's transaction is executed, it will trigger slippage detection and fail. Typically, the larger the market price move generated by the victim's transaction, the greater the revenue that sandwich attackers can obtain. Therefore, most sandwich attacks aim to push the asset price close to the worst acceptable price for the victim, which is defined by the slippage rate[16].

Qin, Zhou and Gervais [12] and Torres, Camino and State [38] developed useful heuristics to detect sandwich attacks on chain. However, Wang et al. [16] point out some flaws. First both authors require a victim transaction. While Wang et al. [16] do not see these heuristic as necessay. This is because the execution of the victim transaction may be unsuccessful. One example of this scenario is when multiple front-runs occur, leading to a high degree of price slippage. Alternatively, the victim transaction may be recorded in a different block, or at a different position within the same block. In such instances, the blockchain would only display the two attack transactions and not the victim transaction. Transactions that fail to execute are also classified as sandwich attacks. This approach provides an insight into the proportion of failed sandwich attacks and the magnitude of the actual profit obtained from these attacks, rather than just the gain.

Torres, Camino and State [38] further require the liquidity of swaps to be provided by the same address and that the gas price of the victim transaction lies between the gas price of T_{A1} and T_{A2}. Wang et al. [16] point out that these heuristics do not make sense due to the fact that a lot of transactions are carried out in collaboration with miners. Qin, Zhou and Gervais [12] require that T_{A1} and T_{A2} are either signed by the same account or sent to the same smart contract. Also their study is limited to 71 tokens. Consequently, in the current environment (see Section: 2.1) the modified heuristics seem to have an advantage where T_{A1} is a frontrun and T_{A2} is a backrun tranaction:

1. T_{A1} and T_{A2} are included in the same block and in this order.
2. T_{A1} and T_{A2} have different transaction hashes.
3. T_{A1} and T_{A2} swap assets in the same liquidity pool, but in opposite directions. The input amount for the swap in T_{A2} is equal to the output amount of the swap in T_{A1}.
4. Every transaction T_{A2} is mapped to exactly one transaction T_{A1} [44, p.7]

Heuristic 3 is the primary way to detect sandwich attacks. Imperfect sandwiches are excluded from analysis with the knowledge that there are existing sandwich attacks that might have different outputs and are profitable. Heuristic 1 is the lower bound with the assumption that an attacker wants the trade to be included in one block due to the dangers of losing profits when swaps are done too often. Heuristic 2 catches transactions that are true in Heuristic 1 but do not represent a valid sandwich attack. Heuristic 4 is introduced to prevent double-counting revenues. Any cases where two identical back running transactions are found were excluded [16].

Last, Wang et al. [16] found that a significant number of individuals trade on decentralized exchanges (DEXs) without possessing a comprehensive understanding of their functioning. Additionally, DEXs often lack adequate information and guidance regarding the price risks associated with their usage. Traders commonly rely on the slippage rate recommended by the exchange or opt for a relatively high rate to ensure the successful execution of their transaction. However, they are often unaware that this choice may result in a worse price and an unanticipated financial loss due to the higher slippage tolerance.

4.3.3 Arbitrage

Arbitrage refers to the practice of buying and selling assets in different markets or locations in order to take advantage of price discrepancies. In the context of the Ethereum ecosystem, arbitrage opportunities arise due to differences in the prices of Ethereum-based assets on different decentralized exchanges or due to differences in the prices of the same asset in different jurisdictions.
The detection and understanding of arbitrage have evolved over time. Arbitrage has been first detected and described in detail from Daian et al. [24] who used Priority Gas Auctions (PGAs) transactions on DEXs to detect arbitrage bots. Later, Zhou et al. [41] focus on single- exchange arbitrage. Wang et al. [43] investigated cross exchange arbitrage using a machine learning approach. Jin et al. [45] used a feature fusion and positive unlabeled learning approach to detect arbitrage on single exchange and cross exchange. Further, Qin, Zhou and Gervais [12] develop heuristics to catch arbitrage:

1. All swap actions $S_{A1}, S_{A2}, ..., S_{An}$ of an arbitrage must be included in a single transaction T_A.
2. The transaction T_A must contain $\geq$ one swap actions S_{A1} and S_{A2}
3. The swap actions within transaction T_A must form a loop L_A such that the last action S_{An} links back to the first action S_{A1}.

4. For any swap action S_{Ai} in transaction T_A, its input amount must be $\leq$ to the output amount of the preceding action S_{Ai-1}.

Due to its popularity, it is also highly competitive. The process involves the identification of discrepancies in the prices of a particular token offered by two decentralized exchanges (DEXs). An individual can then engage in a risk-free arbitrage by simultaneously purchasing the token on the DEX offering a lower price and selling it on the DEX offering a higher price through a single atomic transaction (utilization of multiple actions within a single transaction and the execution of said actions in an all-or-nothing sequence). The deterministic nature of blockchains allows for such transactions to be executed in a trustless and risk-free manner [12].

Arbitrage is by far the biggest type of MEV [46] [47] in the ecosystem and is generally seen as net sum positive for the ecosystem. It helps to keep markets more effiecient while providing more liquidity [12].
According to Weintraub et al. [2] an MEV extractor can adopt either a passive or proactive strategy when performing MEV on arbitrage transactions. The passive strategy entails monitoring the current state of the blockchain and comparing the prices of various assets across multiple exchanges. An arbitrage transaction is only executed if the expected revenue from purchasing an asset on one exchange and selling it on another exceeds the anticipated transaction costs. On the other hand, the proactive strategy involves monitoring the mempool for pending arbitrage transactions or large pending trades. In the event of detecting a pending arbitrage transaction, the MEV extractor replicates the transaction and pays higher transaction fees to frontrun the existing transaction and claim the profits of the arbitrage transaction. In the case of identifying a large pending trade, the MEV extractor verifies if the large trade will result in a price difference across different exchanges before crafting an arbitrage transaction to backrun the large pending trade.

If an Automated Market Maker (AMM) protocol is utilized, it is possible for a bot to engage in arbitrage by exploiting discrepancies in prices between different pools. This allows the bot to acquire profit through a single transaction by purchasing assets at a lower price in one pool and simultaneously selling them at a higher price in another pool.

For example, a trader has turned 1,000 ETH into 1,045 ETH by taking advantage of the different pricing of the ETH/DAI pair on Uniswap and Sushiswap. This is a simple example of how DEX arbitrage can be used to generate profits, by exploiting the price difference between different DEXs.

Furthermore, if there exists a significant difference in prices between off-chain and on-chain trading pools, it is possible for a bot to extract risk-free profit by executing two separate transactions, one on an off-chain exchange and the other on an on-chain exchange. It is worth noting that this form of arbitrage is not atomic, as the profits are not guaranteed to be made within a single block on the blockchain, but it still allows for profit extraction from the protocol.

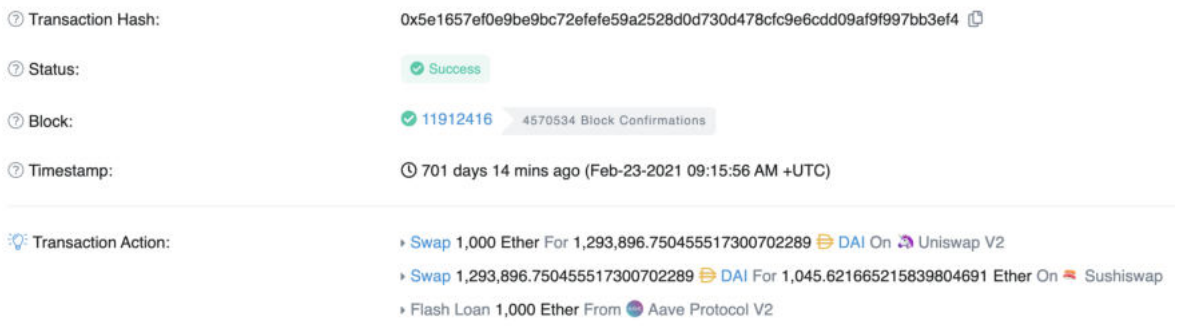


Figure 4.4: Simple Arbitrage Dex

Wang et al. [43] state that the range of cyclic arbitrage opportunities in decentralized exchanges (DEXs) is wider than in centralized exchanges (CEXs) as DEXs support a broader range of trading pairs and tokens. This is exemplified by Uniswap V2, which supports trades between 30,000 tokens while a leading CEX, Binance, only supports less than 400 tokens. Analysis of data shows that more than 2890 liquidity pools and 1143 tokens have been involved in arbitrage opportunities larger than 0.1 ETH, indicating a wider range of tokens in cyclic arbitrage in DEXs. Additionally, the market size in DEXs is larger than in CEXs. According to Makarov and Schoar [48], the potential arbitrage revenue between 34 CEXs is 2 billion USD over four months, while a single DEX like Uniswap V2 can generate a daily revenue of 24 million USD or even 240 million USD. Furthermore, the arbitrage index in CEXs is generally lower than 1.1, while in DEXs, the maximum revenue of a single arbitrage opportunity has been persistently larger than 1 ETH since July 2020, indicating that exploiting arbitrage opportunities in DEXs may be more efficient than in CEXs.

Other more sophisticated strategies are called triangular or cyclic arbitrage. Cyclic arbitrage is a trading strategy that involves taking advantage of price differences between three or more assets and have been analysed by Wang et al. [43] and are discussed in [37].

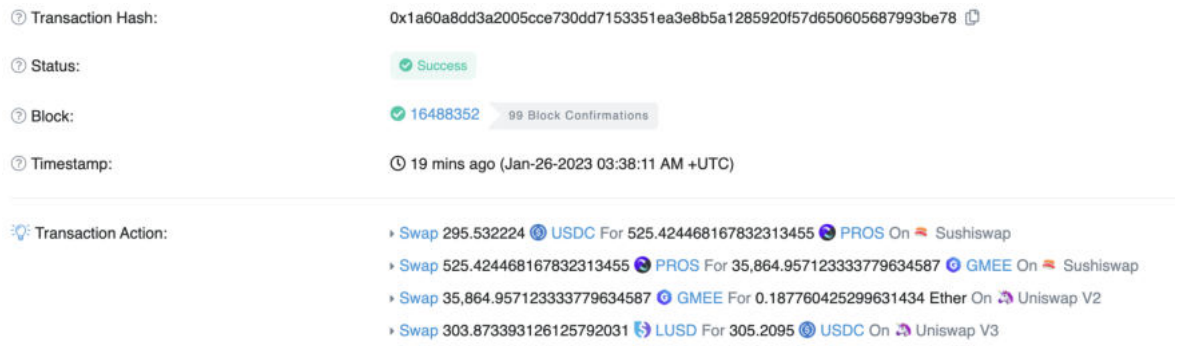


Figure 4.5: Cyclic Multi Dex Arbitrage

The following example (Figure 4.5) shows a more sophisticated trade. In this example a trader used several different cryptocurrencies and exchanges to realize a profit of $0.464196 while paying $9.21 in transactions fees. The trader swapped USDC for Pros on a first DEX then PROS for GMEE followed by GMEE to Ether on another DEX. Lastly he swapped LUSD back to USDC on a third DEX.

Following the literature, arbitrage is mostly seen as negative. Jin et al. [45] point out that arbitrage has been found to result in significant economic losses for unsuspecting users, while also exacerbating network congestion. The negative effects of these activities on both individuals and the network as a whole necessitate the implementation of regulatory measures. Adding to this Daian et al. [24] and Wang [43] point out that Abitrageurs optimize network latency massively (spam) and conduct priority gas auctions to frontrun profitable trades. This in turn leads to network congestion (clogging) and results into higher transaction fees in the overall network. Furthermore, the potential for high miner-extractable value in the context of arbitrage activities on Ethereum, has led to the emergence of fee-based forking attacks and time-bandit attacks. These types of attacks create systemic vulnerabilities at the consensus layer and pose a significant threat to the overall stability and security of the network [31] [41] [24] [12].

Positive aspects of Arbitrageurs are that they play a crucial role in the price discovery process by exploiting price imbalances across different exchanges through cross-exchange arbitrage, as well as through triangular arbitrage, which involves taking advantage of relative exchange rates in multiple currencies. This helps to ensure that prices across different exchanges and currencies converge, providing a more accurate and efficient market [14]. The utilization of automated arbitrage techniques has been shown to lead to an improvement in price efficiency on blockchain networks [37] , similar to how algorithmic trading in traditional markets has been demonstrated to reduce the occurrence of arbitrage opportunities [49]. This highlights the potential for automation to enhance market efficiency and improve overall performance in decentralized financial systems. It is important to note that during periods of market turbulence and heightened volatility, blockchain markets experience reduced efficiency. Consequently, this reduction in efficiency increases the likelihood of cyclic arbitrage opportunities [50].

4.3.4 Liquidations

The decentralized nature of permissionless blockchains, characterized by the absence of trust in any single entity, has led to the implementation of overcollateralization as a crucial safeguard for decentralized finance (DeFi) protocols. However, the inherent volatility of cryptocurrency prices may compromise the effectiveness of this mechanism. To mitigate potential losses, protocols may implement procedures for liquidating undercollateralized positions as a preventive measure. In situations where the ratio of collateral to borrowed funds falls below a specified liquidation threshold, the borrower's position is considered to be in default, triggering a liquidation process. In this process, the collateral provided by the borrower is sold at a discounted price in order to recover the outstanding debt. The liquidation of undercollateralized positions in decentralized finance (DeFi) protocols is open to participation by any network participant. This is achieved by paying off the debt asset in order to acquire the underlying collateral at a discounted rate. As a result, individuals known as liquidators are incentivized to actively monitor the collateral-to-borrow ratios of other participants in order to identify potential opportunities for liquidation [51].

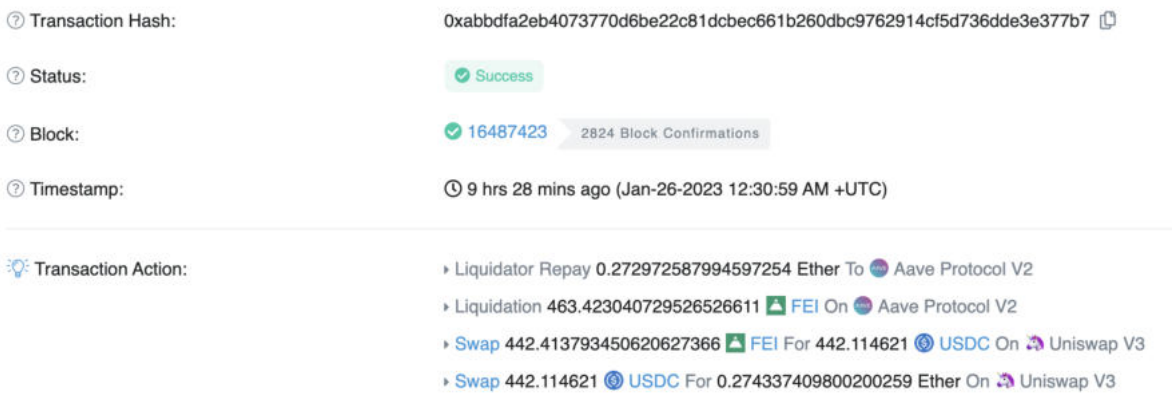


Figure 4.6: Liquidation Example

The following Liquidation 4.6 took place on Aave, one of the leading liquidity protocols. The liquidated Asset is FEI and the debt Asset Weth. The amount liquidated is $463.11 and the debt to cover is $439.92. The total cost of the trade are $24.52, realising a loss of $1.32.
Qin et al. [52] noted that in the current decentralized finance (DeFi) ecosystem, two liquidation mechanisms have gained widespread adoption. These are the fixed spread liquidation and the auction liquidation.The fixed spread liquidation mechanism, which is employed by platforms such as Aave, Compound, and dYdX, enables a liquidator to acquire collateral at a fixed discount when repaying debt. In contrast, the auction liquidation mechanism allows for a liquidator to initiate an auction that lasts for a pre-configured period. During this period, competing liquidators bid on the collateral at the lowest possible price.

In order to better understand what a liquidation is one needs to better understand the underlying dynamics in the lending and borrowing market, followed by an example from Aave [53] one of the leading lending platforms and an example how fixed spread liquidation works.
The Loan to Value (LTV) ratio is a metric used in the lending industry to determine the maximum amount of assets that can be borrowed using a specific collateral. It is expressed as a percentage, with a higher LTV indicating a higher level of borrowing.

$$LTV = \frac{\text{BORROW}}{\text{COLLATERAL}}$$

The Liquidation Threshold is another metric used in lending, which is the percentage at which a position is considered undercollateralized and may be liquidated. The difference between the LTV and the Liquidation Threshold serves as a safety mechanism for borrowers. The Liquidation Penalty is a fee imposed on the collateral assets when liquidators purchase them during the liquidation of a loan that has passed the Liquidation Threshold. The Liquidation Factor is a metric that directs a portion of the Liquidation Penalty to a collector contract from the ecosystem treasury.

$$LiquidationThreshold = \frac{\sum \text{Collateral}_i \text{ in ETH} \times \text{Liquidation Threshold}_i}{\text{Total Collateral}}$$

Additionally, the Health Factor is calculated using various risks parameters and is used to determine when a position may be liquidated to maintain solvency.

$$H_f = \frac{\sum \text{Collateral}_i \text{ in ETH} \times \text{Liquidation Threshold}_i}{\text{Total Borrows in ETH}}$$

When $H_f < 1$ the position may be liquidated to maintain solvency as described in the diagram below.

As a type of MEV, liquidation is a process that occurs when a borrower's collateral value does not properly cover their loan/debt value. In the event of a negative price fluctuation of the debt collateral (a move below the Liquidation Threshold), a position can be liquidated. In permissionless blockchains, anyone can repay the debt and claim the collateral.
A liquidator observes the blockchain for unhealthy positions (the health factor is below 1) to be liquidated. The objective of these searchers is to identify borrowers whose collateral value has fallen below the Liquidation Threshold and, by being the first to submit a liquidation transaction, claim the associated liquidation fee as a reward. This competition among searchers to quickly parse and analyze blockchain data contributes to the efficiency of the liquidation process in permissionless blockchain systems [52].

4.3.5 Oracle Attacks

Decentralized applications (dApps) rely heavily on oracles for accurate pricing of their supported assets, with fluctuations in these oracle-supplied data points directly influencing the state of the markets within these applications. Oracles play a crucial role in the DeFi infrastructure and attacks on them generate heavy losses.
A price oracle is a tool used to view price information about a given asset. Information is published on chain and thus incentives arise to manipulate the information.

Blockchain-based smart contracts often require access to price data for token valuation, a requirement crucial for automated loan providers when appraising collateral value. While Automated Market Makers (AMMs) are a readily available source of price data, they suffer from security vulnerabilities. For example, an attacker can manipulate an AMM's price oracle to illicitly acquire value. These manipulation involve a large token purchase on an AMM, driving the price significantly higher, which then can be used as overvalued collateral for a loan from an automated provider.

To combat such manipulative practices, the introduction of Time-Weighted Price oracles (TWAPs) [54] functioning similar to a sliding-window price average, has been instrumental. TWAPs necessitate that an attacker maintain the manipulated price for an extended period, thus incurring substantial capital expenditure making attacks more difficult.

However, McLaughlin, Kruegel and Vignas analysis indicates that arbitrage events often exceed a single block duration, even for substantial arbitrages. Their examination of large arbitrage campaigns reveals a mean duration of 6 blocks. This suggests that the actual cost

of executing a price-oracle manipulation attack might be substantially less than anticipated [55]. Mackinga, Nadahalli and Wattenhofer [56] also introduce a new form of attack which makes oracle manipulation a lot cheaper than previously described attacks.

5 Quantifying MEV

This work utilizes a modified version of the mev-inspect tool as introduced by Weintraub et al. [2] which will be referred to as MEV detection Script or just Script. The focus is on the Ethereum blocks from 14,444,725 (dated March 23, 2022) to 16,666,666 (dated March 23, 2023), a crucial period in Ethereum's transition from Proof-of-Work to Proof-of-Stake, known as The Merge.

The selected cut-off point builds on the research of Weintraub et al., which covers data from block 10,000,000 (dated May 4, 2020) to block 14,444,725 (dated March 23, 2022). By employing the script created by Weintraub et al., this work aims to contribute further to the Ethereum ecosystem's existing knowledge base.

Additional data was scraped from Zeromev, a platform renowned for being a leading source of MEV data. This data set was used to compare the performance and accuracy of the modified MEV detection tool in identifying MEV instances.

Furthermore, this work shines a light on general trends within the Layer 2 (L2) ecosystem and points towards first relevant research in this area. It represents a pioneering exploration of MEV within the scope of L2 solutions.

5.1 Technical Approach

Relevant data for research on Maximal Extractable Value (MEV) was gathered utilizing a modified version of the mev-inspect software developed by Weintraub et al. [2]. This software is specifically designed to analyze and quantify MEV (Arbitrage, Sandwich, Liquidation) on the Ethereum Mainchain. The following setup and hardware were used. A Docker image was adapted for different chip architecture (ARM64 to ARM64) and executed on Google Cloud using an N2 highcpu instance (Intel Cascade Lake) with 80 vCPUs and 80GB memory.

There are generally two methods to access the required data. The first method involves setting up and synchronizing your own Geth archive node, which allows to download and store the complete history of the Ethereum blockchain, including all transactions and smart contract data. The second method involves using an RPC provider, which provides a remote interface to interact with the Ethereum blockchain. However, the substantial storage requirements and lengthy synchronization time of an archive node (currently estimated at approximately 14 TB of data [57] made a more efficient approach desirable. Instead of using a Geth archive node, a connection was made to Remote Procedure Call (RPC) endpoints provided by reputable service providers such as Alchemy. This allowed for accessing the required data without the extensive resource burden associated with maintaining an archive node. Despite the initial plan to utilize RPC endpoints provided by service providers such as Alchemy for data access, the strategy needed adjustment due to the high volume of requests that exceeded Alchemy's capacity. As a solution, a fully synchronized Geth node, generously made available by the community, was employed for effective access to the necessary blockchain data.

Adopting this methodology made it possible to access and analyze the pertinent blockchain data while mitigating the resource demands of maintaining a local archive node.

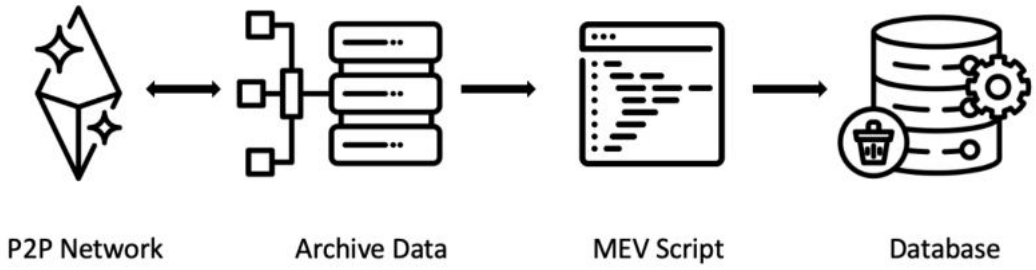


Figure 5.1: Setup MEV Measurement

Sandwiches [see Appendix B sandwiches.py] were evaluated by extracting token transfer events through a comprehensive crawl of archive node data. To detect sandwiching, the heuristics developed by Torres, Camino and State. [38] were applied. These heuristics are based on the assumption that attackers engage in buying and selling the same type of tokens as the victim, executing two separate transactions. It is noteworthy that the quantities of tokens bought and sold by the attacker are nearly identical, and the gas price of the attacker's initial transaction exceeds that of the victim's transaction.

The quantification of arbitrage MEV [see Appendix B arbitrage.py] was conducted by extracting token swap events through an exhaustive crawl of archive node data. To identify arbitrage opportunities, the heuristics proposed by Qin, Zhou and Gervais [12] were utilized. The assumption underpinning these heuristics is that an arbitrage scenario involves multiple swap events, and all these swap events are contained within a single transaction, forming a closed loop.

Quantifying liquidation MEV [see Appendix B liquidation.py] required a systematic crawl of archive node data, specifically targeting liquidation events across various lending platforms. By extracting relevant information from these events, such as liquidated debt and received collateral, it was possible to analyze and measure the impact of liquidations. The script implemented for this purpose was designed to detect liquidations on prominent lending platforms. The script specifically scans for events like Aave's LiquidationCall event and Compound's LiquidateBorrow event, which directly correspond to instances of liquidation [2].

5.1.1 Data Sources

A range of data sources were employed to conduct an comprehensive analysis of MEV on the Ethereum blockchain. These data sources proved crucial in providing the necessary information for the research.

Historical blockchain data was accessed through a Full Archive Node. This approach enabled access to detailed transaction traces, transaction receipts, and block metadata.

Cryptocurrency price data was included by integrating the Coingecko API [58] into the research workflow. The script was adjusted to adhere to the rate limits imposed by the API, ensuring compliance and reliability.

Further, the Flashbots' public API [59] has been used to access and examine bundles mined by participating miners within the Flashbots network. This publicly accessible dataset offered valuable information, including block numbers, miner addresses, miner rewards, and transaction details. This helped to identify Flashbots transactions in our analysis.

5.1.2 Archive Node

Nodes serve as the foundation for the blockchain network and can be categorized into various types, namely full nodes and archive nodes. This chapter intends to explain the concept and significance of Ethereum archive nodes, highlighting their key characteristics, differences from full nodes, and implications in blockchain applications. It also explores why it was necessary to use an archive node for the MEV detection script.

As of current Ethereum client statistics, Geth (Go Ethereum) is the most widely utilized, accounting for approximately 70 % of Ethereum nodes. Nethermind and Erigon follow with 13% and 9%, respectively. Notably, Erigon, although less used, is considered a more lightweight alternative to Geth with fewer overheads, thus offering easier setup [60].

Firstly, an Ethereum node is essentially a computer running the Ethereum protocol, storing the entire blockchain data, and maintaining the network's decentralization. There exist different types of nodes, each fulfilling unique roles and characterized by different data storage techniques. The two primary node categories include full nodes and archive nodes.

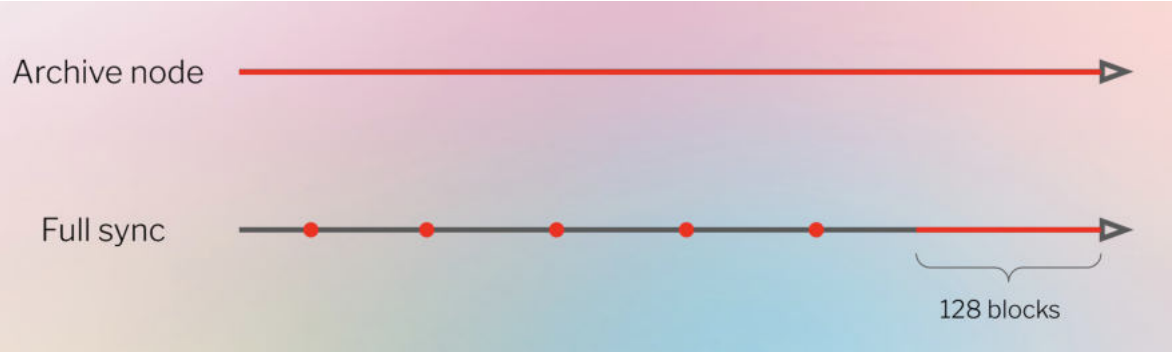


Figure 5.2: Node Types

A full node on Ethereum stores the entire blockchain, meaning it keeps all of the blocks and all the transactions in the chain. This information is enough to validate all transactions from the original block in the chain to the current state. However, full nodes only retain a snapshot of recent states, eliminating the necessity to keep the state of a chain. While a full node can track back through the blocks to compute data like user balance, it requires going through every block and the corresponding transactions. This process is computationally intensive and time-consuming, particularly for querying past data.
On the contrary, archive nodes cover a broader perspective. An archive node stores all of

the historical states of the chain, effectively retaining a snapshot of the entire Ethereum ecosystem at each block since its origin. Hence, unlike full nodes, they can query information from past states much faster and more efficiently. This robust data access makes archive nodes an excellent choice for applications requiring information from a chain's past state. Nonetheless, the vast amount of data stored in archive nodes makes them significantly larger in size compared to full nodes. As of this writing, Geth Ethereum archive nodes occupy approximately 14-15 terabytes of space, and a full sync and archive process can take over six months. This vast storage requirement and the accompanying computational demands make archive nodes substantially resource-intensive.
Further, the difference between an archive node and a full node lies in the purging mechanism. An archive node retains the state of every block, never purging any data. Conversely, a full node employs a mechanism that eliminates any nodes of the tree not needed by the most recent blocks. This difference directly impacts the capacity to run contracts on a smart chain at any given block. An archive node can execute this at any block, whereas a full node can generally only do so at the most recent blocks [61].

In conclusion, Archive nodes, although resource-intensive, offer the advantage of complete data access across the Ethereum chain's history, enabling in depth analytics that require extensive data access.

The interaction of software applications with the Ethereum blockchain, whether through reading blockchain data or transmitting transactions to the network, requires a connection to an Ethereum node. To facilitate this interaction, each Ethereum client incorporates a JSON-RPC specification. This specification provides a standardized set of methods that applications can depend on, independent of the specific node or client implementation. Consequently, this harmonized approach advances interoperability and coherence across varying software applications and Ethereum node types [62].

Within Ethereum, the method refers to a function executed by an Ethereum node. Each method can either retrieve data from the node, execute an EVM function while returning a response, or transmit data to the Ethereum network, such as sending a transaction. As of now, popular Ethereum clients like Geth, Parity, Besu, and Nethermind support around 65 methods. These methods are instrumental in interacting with the Ethereum blockchain and triggering various types of transactions or operations and help in doing analysis [63]. Figure 5.3 gives an example of used methods during the data collection.

When a transaction gets mined in Ethereum, two essential events happen. Firstly, the on-chain Ethereum state modifies as a consequence of contract interactions. Secondly, event logs are published for public inspection. The publication of these event logs enables the execution of JSON-RPC requests to confirm what changed, relative to the events that require attention, thereby enabling subsequent actions.

Solidity, the programming language for Ethereum smart contracts, introduces a concept known as "events" which serve as an abstraction layer over the Ethereum Virtual Machine's (EVM) logging functionality. Events can be subscribed to and monitored through the Remote Procedure Call (RPC) interface of an Ethereum client. As an inheritable attribute of contracts, events are triggered to encapsulate arguments into the transaction's log, a distinct data struc-

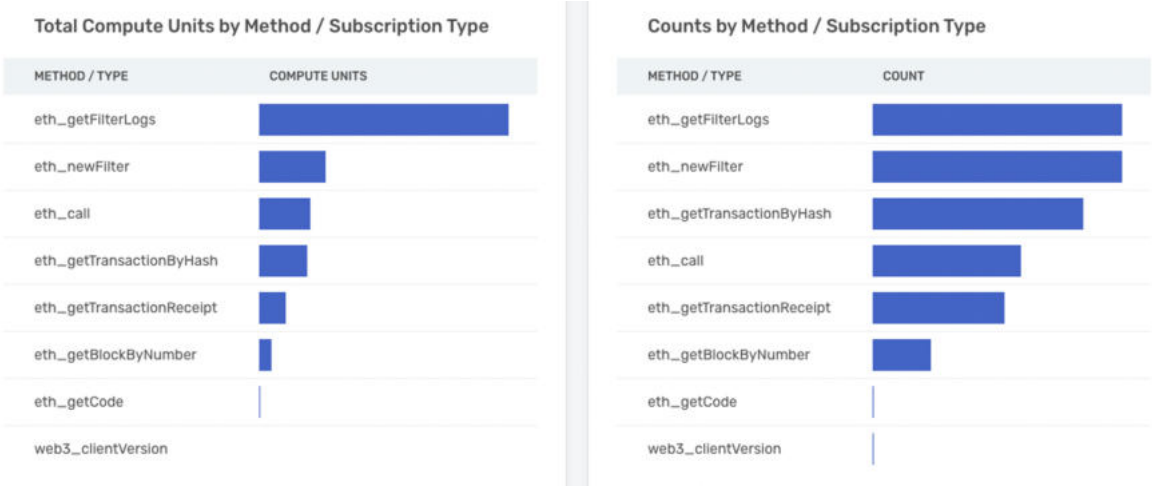


Figure 5.3: Methods

ture embedded in the blockchain. The logs, associated with the originating contract's address, are embedded into the blockchain and persist as long as the block remains accessible, creating a permanent record under the current model. The data stored in logs is inaccessible from within the contracts themselves, even the one that generated the log. Nonetheless, external entities can efficiently access log data due to a specialized indexing data structure. This structure, extending up to the block level, enables the effective implementation of events. A fraction of this log data is stored in bloom filters, which facilitates an efficient and cryptographically secure data search, making it accessible to network peers that operate as light clients and do not download the entire blockchain [64].

As the Ethereum blockchain's smart contracts possess the capacity to emit events during their execution. The resulting data from these events is catalogued in the Logs section of transaction receipts. Every event is indexed based on two aspects: its signature, which is a 256-bit hash, and the contract address that is responsible for emitting the event. Thus, events specific to DeFi were filtered for, emitted from the protocols under study.

5.2 Data Collection

The Script was deployed on Google Cloud, running for a total of 24 hours to investigate the three primary MEV types. Arbitrage made up the largest part of the investigation, requiring the longest duration for execution, producing a significant 5.2 GB of data. Sandwiches generated approximately 3.2 GB of data. Finally, liquidations produced a modest 175 MB. The retrieved data was then saved in a MongoDB database for further analysis. To advance the analysis process, Python and Jupyter Notebook were used as the primary tools.

To help understanding, the key procedures of the script are simplified into pseudocode. The goal is here to show the main elements of the code and strip away the less relvant code blocks. This pseudocode emphasizes the crucial elements necessary for data acquisition and follows the same heuristics outlined in previous chapters. The actual execution involves more comprehensive steps and additional safeguards to ensure the accurate functioning of the code and can be checked on Github B.

5.2.1 Arbitrage.py

The script operates by fetching and processing transaction and event data related to ERC-20 tokens. The script begins by initializing a connection to an Ethereum node using a provider URL, facilitated by the Web3 library. The successful connection to the Ethereum node is verified using the `isConnected()` method. This step ensures the reliability of the data retrieval and processing.

Algorithm 1 Arbitrage Detection

procedure DetectArbitrage($swaps$)
  for each tx_index in $swaps$ **do**
    if transaction has more than one $swap$ **then**
      if first token sold $=$ last token bought **or** both are ETH **or** WETH **then**
        Mark transaction as potentially valid
        Add the first $swap$ to $intermediary\_swaps$
        Create dictionary to track net gain or loss of each token
        for each $swap$ in transaction from the second one **do**
          Get $previous\_swap$ and $current\_swap$
          Add $current\_swap$ to $intermediary\_swaps$
          if token bought in $previous\_swap \neq$ token sold in $current\_swap$ **then**
            Mark transaction as invalid
            break
          end if
          if token bought in the last $swap =$ first token sold **and** transaction is valid **then**
            print "Arbitrage detected"
            Calculate and print details of each swap in $intermediary\_swaps$
            Clear the $intermediary\_swaps$ list
          end if
        end for
        if transaction is still valid **then**
          Get the block data for the current block number
          Get the price of ETH in USD at the time of the block
        end if
      end if
    end if
  end for
end procedure

Once the connection is established, the script fetches specific block information using the `getBlock()` function. This process includes the extraction of all transactions contained within the block.

```
w3 = Web3(PROVIDER)
```

```
if w3.isConnected():
block = w3.eth.getBlock(block_number)
```

Next, it gathers detailed information about each transaction within the block using the `getTransactionByHash()` and `getTransactionReceipt()` methods.

```
tx = w3.eth.getTransactionByHash(transaction["hash"])
tx_receipt = w3.eth.getTransactionReceipt(transaction["hash"])
```

The script interacts with ERC-20 smart contracts on the Ethereum network, forming a contract instance with the contract's ABI and address via `w3.eth.contract()`. This enables the extraction of pertinent token information such as decimals and symbols.

```
token_contract = w3.eth.contract(address=swap["in_token"], abi=[ABI])
in_token_decimals = token_contract.functions.decimals().call()
```

Afterwards, it uses `contract.functions.<function_name>().call()` to call constant functions on the smart contract, fetches the list of transactions in a block using the block transactions attribute, and parses transaction input data (the means by which function calls to smart contracts are executed) using `w3.codec.decode_function_input()`. In summary, this script connects to an Ethereum node, fetches block and transaction data, interacts with smart contracts to extract token information, parses transaction input data to understand the smart contract functions being called, and performs these operations within a designated range of blocks to detect potential arbitrage opportunities.

5.2.2 Sandwiches.py

The presented algorithm analyzes Ethereum blocks, inspecting all transactions within each block to identify potential sandwich attack patterns. Each transaction is investigated to see if it involves a token transfer. When a transaction with token transfer is found, it is stored in a dictionary named asset_transfers, indexed by a unique combination of the token address and receiver address. This forms the basis of our sandwich attack detection as it allows to track all instances of a specific token being transferred to a particular address within the same block.

As the algorithm navigates through the transactions, it seeks instances where the same token is transferred to the same address multiple times within the same block. This could be an indication of a sandwich attack. The algorithm then identifies the first (frontrun) and last (backrun) transactions involving the suspicious address/token pair.

After these potential frontrun and backrun transactions are identified, the algorithm inspects all transactions that took place between these two transactions in search of a large (victim's) transaction involving the same token. If such a transaction is found, it then confirms whether

this is indeed a sandwich attack by checking the ordering of transactions - the bot's buy (frontrun) transaction should precede the victim's transaction, and the bot's sell (backrun) transaction should follow the victim's transaction. Furthermore, the buying and selling prices should be different, indicating a price manipulation.

If these conditions are met, the algorithm returns True, indicating a sandwich attack. Otherwise, if no such pattern is identified after going through all transactions in the block, the algorithm returns False. This process is repeated for all blocks within the specified range, allowing us to collate a set of sandwich attacks for further analysis.

Algorithm 2 Sandwich Attack Detection

procedure AnalyzeBlock($block\_number$)
    $block \leftarrow$ fetch block data from Ethereum blockchain using $block\_number$
    $transactions \leftarrow$ get list of transactions in the block
    $asset\_transfers \leftarrow$ initialize empty dictionary
    for each $transaction$ in $transactions$ **do**
        if $transaction$ is a token transfer **then**
            $asset\_transfers[token\_address + receiver\_address] \leftarrow transaction$
            if $asset\_transfers$ has multiple entries for same token and receiver address **then**
                $frontrun\_transaction, backrun\_transaction \leftarrow$ get first and last transfers of same token to same address
                for each $transaction$ in $block$ **do**
                    if $transaction$ is a large transfer of the same token (victim_transaction) **and** it happened after $frontrun\_transaction$ **and** before $backrun\_transaction$ **then**
                        if $frontrun\_transaction$ happened before $victim\_transaction$ **and** $backrun\_transaction$ happened after $victim\_transaction$ **and** the buying and selling prices are different **then**
                            return True ▷ Sandwich attack detected
                        end if
                    end if
                end for
            end if
        end if
    end for
    return False ▷ No sandwich attack detected
end procedure

5.2.3 Liquidation.py

Firstly, the script iterates over a specified range of blocks in the Ethereum blockchain. For each block, it examines all transactions contained within that block. It retrieves the receipt of each transaction using the fetchTransactionReceipt(transaction) function. This receipt contains detailed information about the transaction, including event logs which can indicate specific activities such as liquidations.

If a transaction receipt indicates a liquidation event, the script extracts the details of this liquidation using the extractLiquidationDetails(transactionReceipt) function. These details are stored in a dictionary called liquidations, using the index of the transaction as the key.

Algorithm 3 Liquidation Detection

```
for each block in the given block range do
    for each transaction in the block do
        transactionReceipt ← fetchTransactionReceipt(transaction)
        if transactionReceipt indicates a liquidation event then
            liquidationDetails ← extractLiquidationDetails(transactionReceipt)
            liquidations[transactionIndex] ← liquidationDetails
        end if
    end for
end for
for each liquidation in liquidations do
    profitOrLoss ← computeProfitOrLoss(liquidation[debtAmount], liquidation[collateralAmount])
    liquidationDetails[profitOrLoss] ← profitOrLoss
    saveLiquidationDetailsToDatabase(liquidationDetails)
end for
```

After all blocks have been analyzed and all liquidations have been identified, the script proceeds to compute the profit or loss from each liquidation. This is done using the computeProfitOrLoss(debtAmount, collateralAmount) function, which takes as input the debt amount and collateral amount from the liquidation details.

Finally, the script stores the details of each liquidation, including the computed profit or loss, in a database using the saveLiquidationDetailsToDatabase(liquidationDetails) function.

5.3 Limitations

The work presented is not without its limitations, owing to the constraints imposed by the methodology employed and the inherent complexity of the domain being explored. The following discussion describes some key areas where these limitations manifest, focusing on limitations related to the Script.

Another limitation concerns the difficulty in dealing with overlapping MEV instances, such as mixed arbitrage and sandwiches, or overlapping sandwiches. At the present moment, there exists no MEV detection software that can fully resolve these scenarios. Lastly, certain types of MEV, despite being identifiable, are not quantified within this thesis.

5.3.1 Protocol Selection

For the purpose of this thesis, it was crucial to identify and select the specific protocols to be examined. The chosen protocols are primarily associated with three principal types of MEV. This chapter explains the rationale behind the selection of these protocols and the types of MEV they serve. The following table 5.1 presents the protocols supported by the MEV detection script, and compares them with the top decentralized exchanges (DEXs) and lending platforms, according to their Total Value Locked (TVL) as reported by DeFiLlama [65].

MEV Script	DexTVL	LendingTVL
Sandwiches:		
Bancor		
SushiSwap	Curve - $3.94b	Aave - $5.16b
Uniswap V1, V2, V3	Uniswap - $3.53b	JustLend - $3.64b
Arbitrage:	Balancer - $1.02b	Compound Finance - $1.89b
0x Protocol	SushiSwap - $307.4m	Venus - $934.24m
Balancer	Loopring - $109.6m	Morpho - $322.58m
Bancor	Bancor - $93.87m	Radiant - $229.68m
Curve	Frax Swap - $65.38m	Benqi Lending - $135.76m
SushiSwap	PancakeSwap - $33.39m	Tectonic - $125.33m
Uniswap V2, V3	ShibaSwap - $28.23m	Fraxlend - $119.29m
Liquidations:	SashimiSwap - $23.23m	Kava Lend - $92.75m
Aave V1, V2		
Compound		

Table 5.1: DeFi platforms used by MEV Script vs. Dex and Lending

Sandwich transactions have been identified within various protocols in the Ethereum DeFi ecosystem. For the purpose of this analysis, Bancor, SushiSwap, and Uniswap V1, V2 and V3 protocols have been chosen. These protocols were selected based on their popularity and the substantial volume of transactions they process, thereby providing a significant sample size for sandwich transaction examination.

The complexity and variety of arbitrage opportunities within the DeFi ecosystem made a strategic selection of protocols for this category necessary. The protocols selected for the analysis of arbitrage opportunities include 0x Protocol, Balancer, Bancor, Curve, SushiSwap, and Uniswap V2 and V3. These protocols are known for their significant arbitrage potential due to their substantial liquidity and diverse asset offerings.

For the liquidations category, the thesis focuses on Aave V1 and V2 and Compound. This selection allows for a comprehensive understanding of the mechanisms of liquidations, which occur when the value of collateral in a loan falls below a specified level.

This protocol selection offers a representative sample of the Ethereum DeFi ecosystem, highlighting various types of MEV, their occurrences, and their implications for different protocols. Please note that this selection is not exhaustive and does not cover every existing protocol or type of MEV. The script used various protocols, as highlighted in the table. These protocols have been selected based on their significance and widespread adoption within the ecosystem.

5.3.2 Methodology

The work from Weintraub et al. [2] build upon Qin, Zhou and Gervais [12] and Torres, Camino and State [38] and thus use the same heuristics to detect MEV. In case of Arbitrage, given the expansive nature of the Ethereum blockchain, which consists of more than 11 million blocks and surpasses a billion transactions, a balance between efficiency and comprehensiveness was essential. An instance of this compromise is the implementation of a scanning

window of 100 blocks for detecting arbitrage attacks. This methodology is potentially incapable of identifying arbitrage attacks in cases where transactions are separated by more than 100 blocks. Moreover, limitations emerge from the specific focus of our detection heuristics on bot-performed arbitrage attacks. Attackers can execute transactions directly with a susceptible contract, circumventing the use of bot contracts. However, differentiating these transactions from those of benign users presents a significant challenge. In an effort to minimize potential false positives, the focus was confined exclusively to bot contract operations. Therefore, while this might lead to some false negatives, the results should be viewed as providing a conservative estimate [38].

Additionally, the Sandwich detection mechanism operates under the assumption that both transactions of a single sandwich take place within the same block. This assumption facilitates the efficient processing of the vast blockchain history, but it is not entirely accurate. Situations may arise where the transactions of a profitable sandwich span across multiple blocks, which our current methodology would fail to detect. Therefore, these outlined limitations highlight the necessity for further refinement and enhancement of the current heuristics and methodologies [12].

A further constraint of our methodology is its exclusive focus on the most recognized and prevalent forms of Maxmimal Extractable Value (MEV): sandwiching, arbitrage, and liquidation. This specialized focus prevents the inclusion of other potential types of MEV. Were additional variants to exist, they would require the development and application of distinct detection techniques and subsequent analyses. This narrow scope, while enabling detailed examination of specific MEV forms, limits the breadth of MEV activity that can be accurately captured and assessed.

5.4 Overall MEV Results

The comprehensive analysis of Maximal Extractable Value (MEV) over the Ethereum blockchain offers significant insights into the dynamics of on-chain activity. A critical emphasis was placed on the three main categories of MEV, namely sandwiches, arbitrages, and liquidations. Table 5.2 presents an overview of the absolute Statistics. The chapter includes a discussion of the MEV results obtained from the MEV script used, which includes MEV via Flashbots and also data from Zeromev.

Data Source	Zeromev	Script	Flashbots
Sandwich	610,029	556,334	629,404
Arbitrage	641,289	1,258,474	799,774
Liquidation	23,880	54,803	37,572
Total	**1,275,198**	**1,869,611**	**1,466,750**

Table 5.2: MEV Dataset Overview

A key observation from the analysis was the apparent uptrend in the occurrence of MEV activities over time. This growing trend illustrates an evolving dynamic within the Ethereum ecosystem, revealing the increasing prominence of MEV as a factor in on-chain operations.

5.4.1 MEV Data Script

Particularly noteworthy was the surge in sandwich events, which came to light as a substantial contributor to the overall MEV activity. With 556,334 recorded instances, sandwiches emerged as a significant on-chain event in the analyzed period.

Arbitrages, characterized by the capitalization on price discrepancies across different exchanges, were the most prevalent type of MEV, totalling 1,258,479 instances. This indicates the vast extent of opportunity present on the Ethereum network for traders to exploit such disparities for profit.

Liquidations, albeit less frequent in occurrence compared to arbitrages and sandwiches, still presented a noteworthy count of 54,803 instances. Liquidations, defined by the compulsory closure of positions when collateral falls beneath the required level, exhibit an essential component of risk management in DeFi platforms.

The comprehensive results thus highlight a vibrant MEV landscape within the Ethereum ecosystem, characterized by a rising trend and considerable instances of sandwiches, arbitrages, and liquidations.

The presence of negative profit in certain MEV transactions may appear paradoxical initially. However, a closer inspection of block-level dynamics provides a plausible explanation. Ethereum validators exercise control over all transactions within a block. A single transaction, despite yielding a negative profit, may enable a larger, positive net profit when combined with other transactions within the same block. Hence, while examining MEV profitability, it's essential to focus on the cumulative profit across all transactions within a block, underlining the complex interplay of Ethereum transactions.

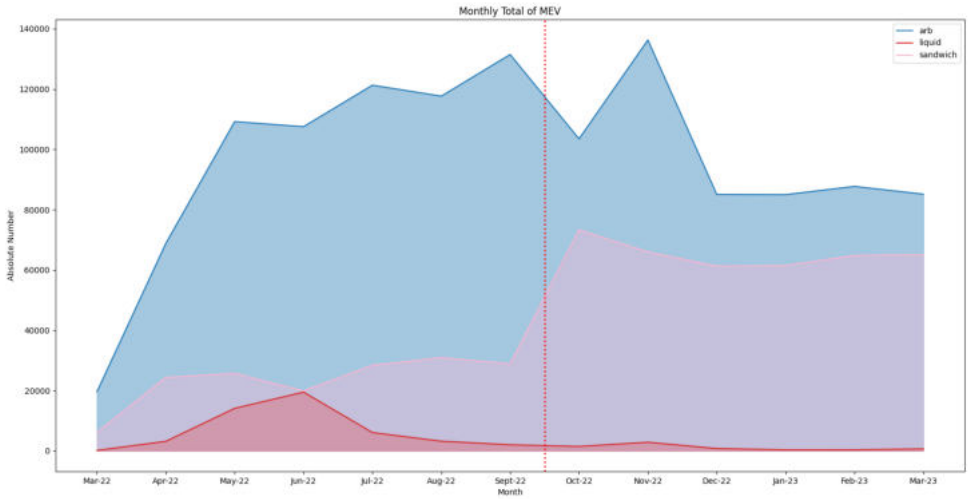


Figure 5.4: Total monthly MEV Script

A significant trend in the MEV landscape is the marked increase in the prevalence of sandwich attacks. These types of attacks have become especially appealing after PBS due to their risk-free nature. If a transaction within the bundle fails, the entire bundle remains unexecuted, thereby eliminating potential losses for the attacker.

Heimbach et al. support this, indicating a significant increase in these types of attacks. Their findings document a total of 1,208,707 sandwich attacks during their data collection period, with a stark contrast between the frequency of attacks in PBS and non-PBS blocks. In fact, their data suggests that nearly all sandwich attacks were taking place within PBS blocks [6]. This underscores the influence of PBS on the facilitation of sandwich attacks. Similarly, Wahrstaetter et al. provide additional perspective on the rise of sandwich attacks. Their research highlights an increase in the confirmation latency for Ethereum transactions following the platform's transition to Proof-of-Stake and Proposer-Builder Separation (PBS). Such delays in transaction confirmation are likely to exacerbate the risk of sandwich attacks [66].

Wahrstaetter et al. further note that the design of MEV-Boost, aimed at enhancing decentralization, inadvertently creates an environment favorable for risk-free sandwich attacks. This potential side effect, they argue, might warrant regulatory attention [8].

5.4.2 MEV via Flashbots

In the aftermath of the The Merge the rise of Flashbots sparked considerable interest and concern within the blockchain community. Leveraging their API, the volume of MEV that was being routed through Flashbots could be detected by comparing our results with their database. It was observed that Flashbots had gained an unexpectedly high market share, with figures nearing 70% at its peak.

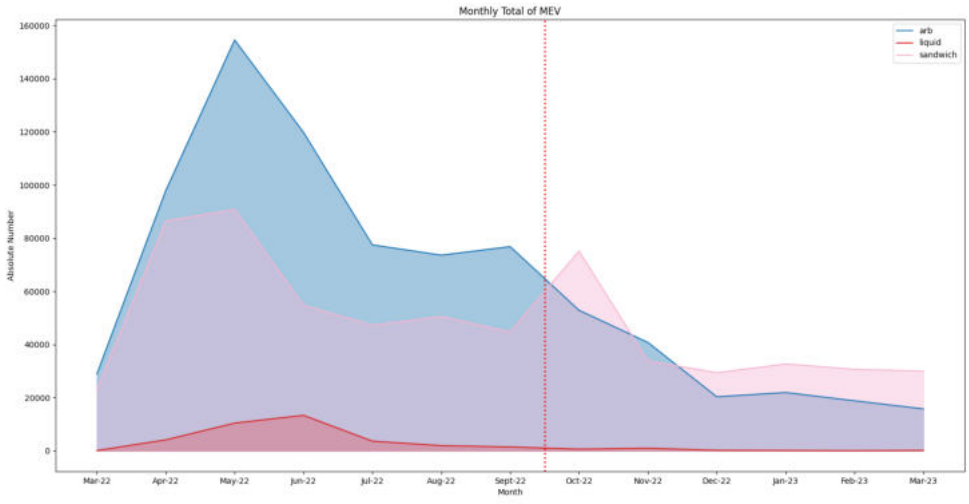


Figure 5.5: Total monthly MEV via Flashbots

This significant market share signaled a potential centralization within the MEV extraction process, triggering unease within the community. Centralization is largely at odds with the principles of blockchain, which prioritizes decentralization to ensure fairness, security, and transparency. The high degree of market control by Flashbots has therefore intensified the discourse on the potential risks and drawbacks of such centralization.

Moreover, Flashbots' approach towards compliance further raised concerns. In particular, their adherence to the guidelines set forth by the OFAC, specifically their practice of censoring blocks, amplified the centralization issue. The introduction of censorship mechanisms within the Flashbots operation has raised questions about the balance between regulatory compliance and maintaining the inherent ethos of a decentralized blockchain network.

Yet, the recent trends show a notable shift. There has been a visible downtrend in the volume of Flashbots blocks, attributable to a more competitive landscape among relayers [10]. The increase of these entities, responsible for relaying transactions to the blockchain, have begun to dilute Flashbots' previously dominant market share. This indicates a move towards a more robust and decentralized Ethereum network, echoing the core principles of blockchain technology.

This evolving dynamic between Flashbots and the broader ecosystem of relayers requires continued observation and analysis, particularly given the regulatory challenges and the community's concerns about centralization.

5.4.3 MEV Data Zeromev

The data utilized in this research was primarily obtained through API access provided by ZeroMEV. This open-source resource has dedicatedly compiled data on various MEV strategies, making it a valuable point of reference.

A custom script was developed to extract detailed block data. The primary goal of this script was to calculate the monthly frequency of different types of MEV: arbitrage, liquidation, and sandwich attacks. This allowed for an examination of trends over time, providing insights into the evolution and prevalence of different types of MEV.

The process of identifying and counting sandwich attacks required particular attention. Sandwich attacks consist of a front-run transaction, one or more victim transactions, and a back-run transaction. Initially, each victim transaction was counted as a separate sandwich attack. However, this approach overlooked the fact that multiple victim transactions can be part of the same sandwich attack within a single block. As such, the methodology was revised to count the front-run and back-run transactions within each block. This change in approach ensured that each sandwich attack, regardless of the number of victims, was counted only once, providing a more accurate picture of the frequency of sandwich attacks.

An examination of the extracted data reveals a discernible upward trend in the prevalence of MEV, particularly an increase in sandwich attacks. These attacks have become notably more frequent over time, illustrating a shift in the choice of MEV strategies adopted.

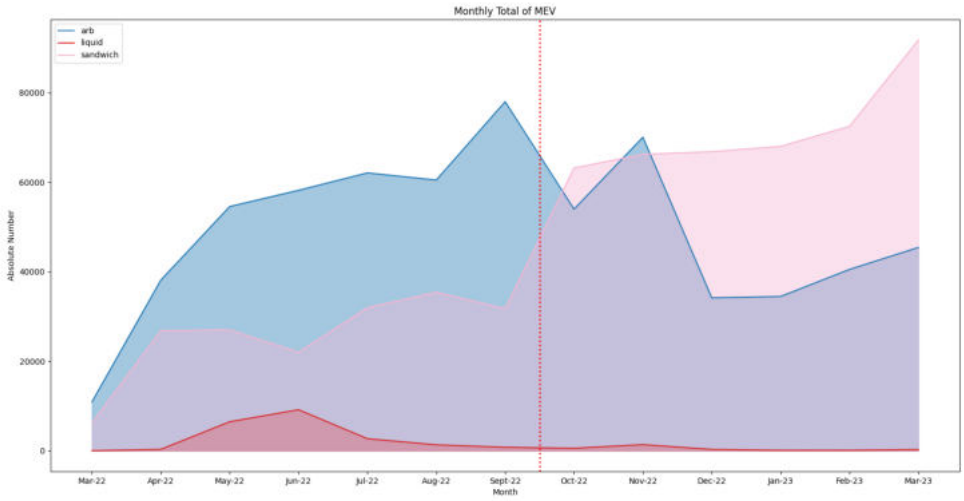


Figure 5.6: Total monthly MEV Zeromev

While liquidations are certainly present within the dataset, their relative frequency compared to the other MEV types is significantly lower. Despite their presence, liquidations do not appear to play a dominant role in the MEV landscape. This could suggest a trend towards strategies that offer a more predictable return, or possibly reflect the characteristics of the protocols and market conditions under analysis.

5.4.4 Comparison

This chapter provides a systematic comparison between two primary data sources: data fetched with a script provided by Weintraub et al. and data sourced from Zeromev. The comparison revolves around the three primary financial metrics: Arbitrage, Sandwich, and Liquidation.

Arbitrage data shows less absolute magnitude in Zeromev compared to the script-based method, yet follows a similar trend across both sources. Sandwich data, on the other hand, is higher in Zeromev than the script data, while sticking to an analogous trend. In the case of Liquidation data, the script reveals a greater magnitude than Zeromev, albeit exhibiting a consistent trend between both sources. Notably, during the final month of observation, both data points converge closely.

In every corresponding graph is the inclusion of a dotted line. This line represents a significant event, 'The Merge', that occurred in September, thus serving as a crucial point within the analyzed period.

A substantial discrepancy has been observed between the Script and Zeromev with regard to detected arbitrage instances. The Script consistently identifies a larger number of such instances than Zeromev. In this section, potential explanations for this observed discrepancy, focusing on differences in detection techniques, definitions, and the handling of various arbitrage scenarios are presented.

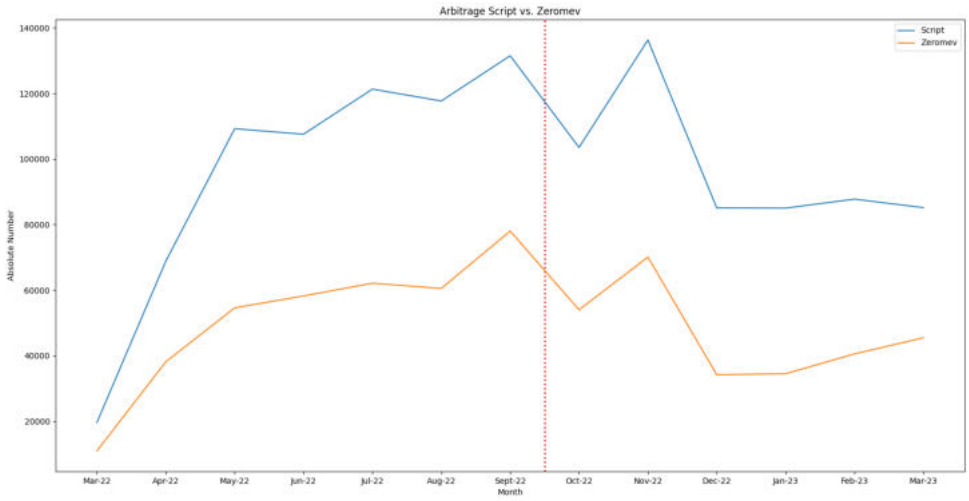


Figure 5.7: Arbitrage Script vs. Zeromev

A significant difference between the two lies in the handling of split arbitrages. A split arbitrage refers to a series of transactions where tokens are exchanged across more than two liquidity pools. This can be represented as a sequence: $Token1 \rightarrow PoolA \rightarrow Token2 \rightarrow PoolB_{(50\%)} \rightarrow Token1 \rightarrow PoolC_{(50\%)}$. While our Script appears capable of handling multi-step transactions and, therefore, detecting split arbitrages, it is unclear to what extent it can detect more complex split arbitrages. Conversely, Zeromev explicitly states that it does not support split arbitrages, contributing to a lower number of detected arbitrage instances [67].

A second area of difference between the two tools lies in the handling of overlapping MEV. Zeromev admits to difficulties in detecting overlapping MEV situations. These include cases where arbitrage opportunities coexist with other types of MEV, such as sandwich attacks. If the Script handles overlapping MEV more adeptly, this could account for its higher arbitrage detection rate.

Further contributing to these discrepancies is the respective definition of arbitrage employed by each tool. The Script classifies a transaction as arbitrage when there is a price disparity for a single currency between two exchanges, thus rendering the exchange profitable even after accounting for mining fees. Zeromev's precise definition of arbitrage is, however, unclear from the information available. Any deviation in these definitions might lead to different detection outcomes.

Additionally, the inclusion of frontrun arbitrage in the Script could contribute to the observed discrepancy. Frontrun arbitrage involves a participant identifying an arbitrage opportunity in the public mempool and executing the same transaction with a higher gas fee to preempt the original transaction. It is plausible that these instances are included as arbitrage by the Script, while Zeromev could categorize them as sandwich attacks or another MEV type, leading to fewer detected arbitrage instances.

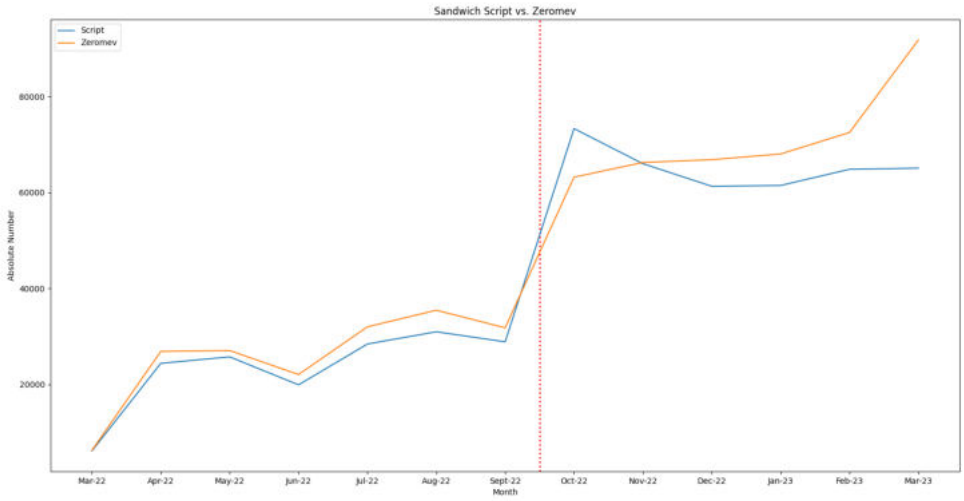


Figure 5.8: Sandwich Script vs. Zeromev

Upon a comparison of our detection Script and the Zeromev sandwich attack data, a number of differences surface that could potentially explain the higher detection rates of sandwich attacks reported by Zeromev. It is crucial to note that these are potential explanations based on the information available regarding Zeromev's methodology. To fully understand why Zeromev detects more sandwich attacks than the MEV Script by Weintraub et al., a closer comparison of both pieces of code is needed. However, since Zeromev's code and the specific rules it uses aren't openly available, this detailed comparison can not be done.

Zeromev has integrated a comprehensive mechanism for the identification of what is termed "position taking". This occurs in instances where the output of the attacker's frontrun does not equate to their input in the backrun, resulting in an imbalance. Such instances could falsely inflate the estimated profitability of the attack. The thorough adjustment for position taking in Zeromev's algorithm might thus contribute to the system's increased identification of sandwich attacks.
Additionally, Zeromev employs a distinct method to extend the parameters of Automated Market Maker (AMM) pools from the sandwich transactions. This approach allows Zeromev to recreate and analyze the attack with more accuracy. Zeromev also demonstrates its robustness by minimising potential errors attributable to differences in protocol mechanics and fee structures across the varied DeFi protocols in the Ethereum ecosystem. By factoring in these differences, Zeromev likely achieves a more precise detection of sandwich attacks.
Furthermore, Zeromev displays resilience in handling outliers, including but not limited to, Pool Imbalance Attacks, Low Liquidity/Malicious Tokens, Split Transactions, and Undetected Reverts. The specific handling of these outlier conditions could contribute to a more extensive detection of sandwich attacks [67].
Finally, Zeromev's broader analysis, which includes considerations for user losses, might add to the larger set of detected sandwich attacks.

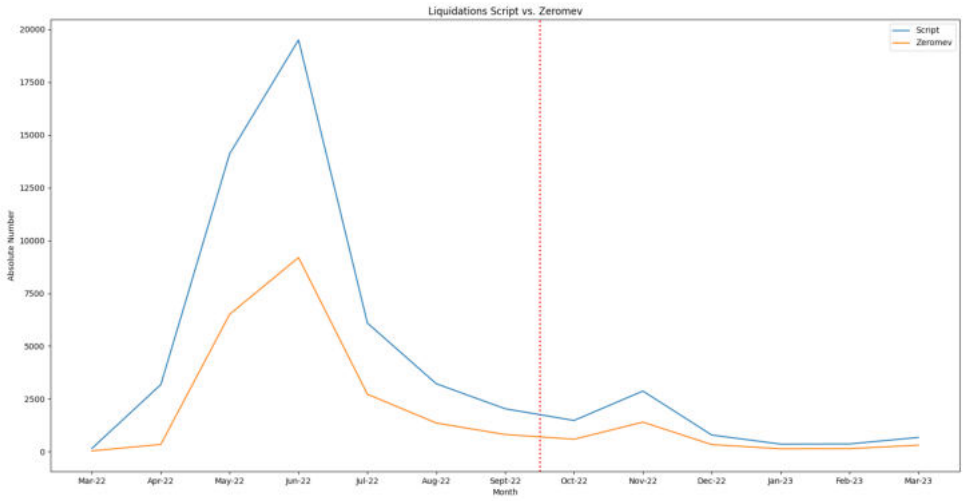


Figure 5.9: Liquidations Script vs. Zeromev

In the domain of liquidations, a distinct variance has been noted between the Script and Zeromev, particularly before the Merge. The Script, in these instances, consistently detected more liquidations than Zeromev. This could potentially be attributed to differences in the underlying heuristics, specific definitions of liquidations, or processing methods employed by each tool.

However, this divergence appears to lessen significantly following the implementation of the Merge. Post-Merge, the Script and Zeromev demonstrate an increasing convergence in detected liquidations. This increased alignment suggests that the heuristics or data processing techniques used by both tools are becoming more consistent with each other, or that the Merge has impacted the on-chain conditions and mechanisms associated with liquidations, leading to a closer correlation between the two tools.

5.5 MEV by Protocol

This chapter dives into the distribution of MEV across various popular DeFi protocols. The distribution across popular protocols relies on data sourced from Zeromev. Three main types of MEV transactions are analyzed and counted from 03/2022 until 03/2023 per protocol.

The protocols analyzed include Uniswap V2, Uniswap V3, Curve, Balancer1, 0x, Bancor, Aave, and CompoundV2. The data reveals an uneaven distribution of MEV, with some protocols capturing a significant percentage of the total value.

Uniswap V2 emerges as the dominant platform, capturing a massive 85.4% of the total MEV. Following Uniswap V2, Uniswap V3 is the second most significant platform with a MEV of 14.2% of the total value. The dominance of Uniswap platforms demonstrates their central role in the Ethereum and broader blockchain landscape.

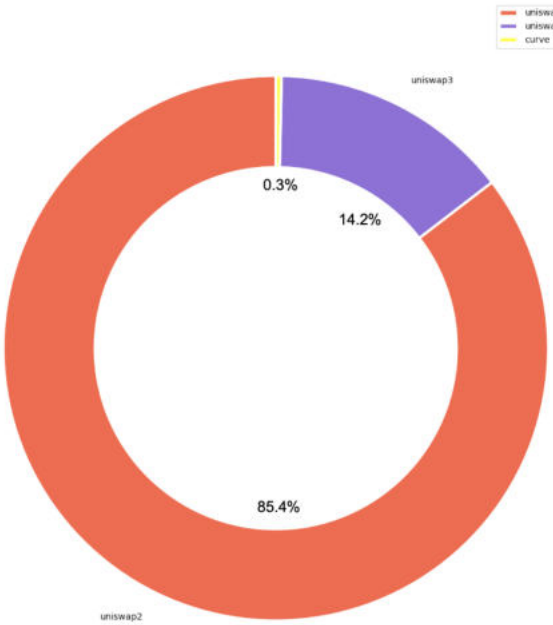


Figure 5.10: MEV by Protocol

Lower in the ranking, Curve manages to obtain 0.3% of the total MEV. Following closely, Balancer and 0x protocol capture 0.1% of the MEV. Bancor, Aave and Compound V2 account for the remaining, but small portions of the total MEV. Due to their comparatively low volumes, these protocols have been omitted from the primary visual representations.

In summary, the analysis provides a clear view of how often MEV occurs via different protocols, with Uniswap V2 and V3 taking the lead. These insights are helpful to understand the dynamics of MEV extraction and highlight the central role of exchanges, particular Uniswap in shaping the landscape in the Ethereum ecosystem.

5.6 MEV on L2

Maximal Extractable Value (MEV) on Layer 2 (L2) solutions forms a novel and largely unexplored field. This stems from several factors, the most notable being the diverse approaches implemented by various L2s, each possessing unique technical and economic incentives. However, first attempts have been done here [19] and here [68].

One important issue is the implications of latency competition within Layer 2 (L2) solutions, specifically, within rollup protocols in the Ethereum network. A significant element to consider in this context is the sequencer, a component that collects transactions and publishes them in a sequence that is essential for the execution stage.

In the rollup context, latency competition becomes a prevalent issue. This competition emerges when parties invest substantial resources to minimize latency, aiming to deliver their transactions to the sequencer before their competitors [69]. Unfortunately, this competition tends to favor parties with more resources, who can strategically target specific data centers to reduce latency. This phenomenon often undermines fairness and efficiency since the transaction's value does not necessarily correlate with the resources the party can use.

The policy that dictates the transaction ordering within the sequencer significantly impacts the sequence's final arrangement. The most common policy in use is the First-Come, First-Serve (FCFS) rule, largely due to its simplicity and perceived fairness. This in turn makes Sandwich Attacks not possible. It also contributes to minimizing latency, as it allows the immediate appending of transactions to the sequence upon arrival However, the FCFS rule also fuels latency competition, thereby creating a significant disadvantage. This competition, well-recognized in traditional finance, has led to the development of a whole industry around it [69].

6 MEV Classification

In previous chapters, various types of Maximal Extractable Value (MEV) have been explored and the Ethereum mainchain from block 14,444,725 through 16,950,602 has been analyzed. One central question emerges from this exploration: Does MEV enhance the welfare of distributed systems, rendering it a feature, or does it compromise these systems, making it a bug? This chapter aims to shed light on this question by examining diverse perspectives present in scholarly literature and among market participants.

Following this, an exploration into the interplay between efficiency and fairness in the context of MEV by discussing the topic on a general level will be made. Categorization and discussion of MEV will involve identifying and examining aspects that can be considered either 'beneficial' or 'detrimental' in nature. Lastly, the classification framework will be presented in detail, offering a robust approach to understanding and analyzing MEV.

We propose a classification framework to categorize different types of MEV. Our objective is to amalgamate these varied viewpoints into a single framework, which could serve as a foundation for further formal development and provide guidance for understanding MEV. This framework offers a comprehensive overview of the most commonly observed types of MEV and reflects the sentiment prevalent in current literature.

6.1 Efficiency vs. Fairness

The overall impact of MEV is not clear. While seen as normal or even necessary by practitioners, it is fastly seen by academics as a bad thing or even theft. Mostly noting that the network level is in danger, creating high gas fees and network spam. Others [70] challenge the prevalent perspective of Maximal Extractable Value (MEV) extraction as analogous to theft within cryptocurrency frameworks. The discourse unpacks the intrinsic role of MEV in systems secured by economic incentives, extending this paradigm even to centralized structures. MEV is seen as a vital metric for assessing network security, thus establishing its foundational significance in these systems.

In order to understand the discussion better, it helps to see the discussion as an economic and political motivated problem. In economic theory, efficiency relates to the ideal distribution of resources, a scenario in which the well-being of a given individual cannot be improved without negatively affecting another individual. This concept is known as Pareto efficiency. An economy operating with optimal efficiency will maximize benefit as a whole. However, this does not inherently dictate how said benefit will be allocated among users. Contrastingly, the principle of fairness addresses the fair allocation of resources, striving to ensure a fair "share" for all. The precise interpretation of what constitutes a "fair share," however, is subject to extensive debate and can vary significantly between individuals.

The discourse on Maximal Extractable Value (MEV) within blockchain systems has yielded two prominent schools of thought: one that portrays MEV as a mechanism contributing to efficient market price discovery, and the other that views MEV as a potentially exploitative element.

This dichotomy, in essence, mirrors the broader economic debate over efficiency versus fairness. Critics of MEV often highlight practices such as frontrunning and sandwich attacks, which impose costs on users and might be seen as unfair or even exploitative. However, proponents of MEV point to its role in enabling arbitrage that helps maintain pricing alignment across decentralized exchanges (DEXs). This, in their view, contributes to market efficiency and price discovery.

When engaging with the literature on MEV, it's critical to understand the perspectives of the authors and organizations behind the research. For example, Flashbots is a significant entity in the MEV landscape, and it largely advocates for MEV. Their resources offer valuable insights into the potential benefits and efficiency improvements associated with MEV. On the other side of the debate, organizations such as ZeroMEV.org [67] adopt a largely critical stance towards MEV. Their resources provide an invaluable counterpoint, highlighting potential fairness concerns and exploitative practices associated with MEV.

Poux, de Filippi and Bruno Deffains categorize the concept of "MEV" into two dimensions. The first dimension pertains to the effect of MEV on the efficiency of the blockchain network, which can either increase or decrease the efficiency. The second dimension considers the "fairness" of MEV [71].

Barczentewicz [72] expands on the nuances of market efficiency, hypothesizing that even when a participant experiences a loss within a singular transaction, they may still garner overall indirect advantages from the presence of MEV extraction, which could potentially compensate for such losses. Alternatively, the paper presents the possibility that even those arbitrages or liquidations deemed as non-toxic - ones that do not negatively impact any individual transaction - could nonetheless precipitate substantial adverse repercussions on the aggregate market. Noteworthy examples of such implications include spam and network congestion, particularly in circumstances where entities in pursuit of MEV are obliged to participate in priority gas auctions.

Drawing from the work of Wang et al. [16], attitudes towards sandwich attacks in the Decentralized Finance (DeFi) ecosystem vary considerably, making it difficult to categorically label these attacks as either malicious or regular market behaviors. The diversity in users' backgrounds, knowledge, experiences, and motivations significantly influence their perception of such attacks. The blockchain systems establish equitable markets for all DeFi participants, providing them with the opportunity to either capitalize on profitable situations or safeguard themselves. Quantitative evidence from the study indicates that the influence and execution of sandwich attacks have evolved in tandem with the progression of DeFi markets, insinuating that the ethical interpretation of such attacks may also shift over time. They, however, also come to the conlusion that from an ethical standpoint, the deployment of sandwich attacks within the Decentralized Finance (DeFi) sector has raised significant concerns. The gravity of these concerns is accentuated by a pervasive lack of awareness among market participants, which disproportionately impacts novices in the DeFi landscape who possess limited familiarity with these practices. Such conditions could potentially obstruct the wider adoption of DeFi technology, emphasizing the importance of democratizing MEV and enhancing user protection through the creation and distribution of suitable tools.

6.2 Beneficial Effects

The two types mainly seen as beneficial are Arbitrage and Liquidation. As highlighted by Qin, Zhou and Gervais [12], the function of arbitrage is crucial for enhancing market efficiency and is generally regarded as a beneficial practice. Barczentewicz [72] also underscores the importance of arbitrage strategies in the context of MEV, maintaining the alignment of prices across disparate decentralized markets. In this scenario, controlling the order of transactions within a block to determine who can capitalize on a particular arbitrage opportunity becomes of significant value.

Parallel to arbitrage, liquidation procedures within lending protocols contribute positively to the network's overall robustness. Protocols like Aave or dYdX necessitate borrowers to provide collateral to underpin their operations. If this collateral value dips below a certain limit, the position can be liquidated at a fixed price, generally less than the market price, allowing lenders to recuperate their loan [71]. This mechanism provides the liquidator with a profit margin and simultaneously prevents the position from becoming unsecured [52].

In summary, the mechanisms of arbitrage and liquidation play indispensable roles in ensuring the efficiency and stability of decentralized financial systems. These processes extend beyond mere profit-maximizing strategies, they facilitate the alignment of prices across decentralized markets and shield lending systems from non-performing loans.

6.3 Detrimental Effects

MEV accumulation can inadvertently promote centralization. The specific expertise and infrastructure necessary for MEV extraction increases the possibility of centralization, especially when an entity or consortium controls multiple sequential blocks. This is of particular concern in instances of oracle manipulation [72]. Even though MEV might initially seem harmless at the application layer, its impact on transaction overhead and the degradation of the blockchain incentive system makes it more of a design defect than a desirable attribute [12].

Strategies such as sandwiching and most other front-running, are commonly classified as harmful MEV, as they can impair transaction execution. The negatively impacted transaction either experiences unfavorable conditions or does not execute at all. This can be due to a front-running transaction seizing a limited opportunity, such as an NFT minting. Interestingly, strategies solely involving back-running— the act of placing a transaction only after a targeted transaction — can similarly lead to worsened execution conditions. This typically occurs when potential MEV extractors postpone the execution of a transaction until the most lucrative time within a block [72].

Sandwich attacks, a particular type of MEV strategy, can have detrimental consequences on decentralized finance (DeFi) systems [12] [42] [24]. First, they can contribute to a surge in on-chain transactions. These additional transactions occupy block space, sidelining standard trades, and resulting in escalated gas prices within the DeFi ecosystem. Secondly, the profits

yielded from these sandwich attacks inflate the MEV in blockchain systems. This growth in MEV can give rise to systemic vulnerabilities at the consensus layer, leading to scenarios such as fee-based forking attacks and time-bandit attacks [24].

Another negative impact of MEV is the potential consensus layer threat shown by many scholars [41] [24]. [12]. The heavy utilization of MEV can inadvertently cause congestion on the peer-to-peer (P2P) network. This congestion could degrade communication efficiency and latency, which in turn, increases the occurrence of stale blocks, thereby compromising the consensus mechanism's security [41]. Daian et al. state that in regards to application-layer security, decentralized exchanges (DEXs) present a serious security risk to the blockchain systems on which they operate at the consensus layer [24]. Further, the existence of MEV grants miners with extra monetary resources, enabling them to initiate bribery and undercutting attacks. In these scenarios, hostile miners purposely propose financial rewards on a forked chain to attract more mining power. The focus on revenue maximization by a MEV relayer increases the potential earnings for miners, consequently escalating the threat of forks at the consensus layer [12].

While it's not common to categorize hacking activities within the context of MEV, there's an argument to be made about the significant value extraction they can engender in a network. Yet, it's critical to note that hacking, with its vast literature and different operational methods operates in its unique spectrum [36] [73].

6.4 MEV Classification Framework

Our investigation primarily concentrated on prominent types of MEV such as arbitrage, sandwich transactions, and liquidations, yet our framework has accommodated more diverse forms and effects of MEV to present a comprehensive representation. The framework also probes the implications of MEV for different stakeholders in the network, such as users, validators, and the overall network health. A model, however, can only approximate reality and in this case, the complexity is intensified by the multifaceted nature of the discussions surrounding MEV, which occur across technical, economic, and political levels. Consequently, there is a multitude of prominent perspectives on this matter.

Level	Beneficial Effects	Detrimental Effects
User	-	Frontrunning, Displacement, Sandwich [38] [12] [42] [24]
Network	Arbitrage [12] [72] [71] , Liquidation [52] [71]	Suppression (Clogging) [40], Consensus Attacks [25][26]
Overall	Arbitrage, Liquidation	Governance Attack [74], Hacks[36], Oracle Attack[56]

Table 6.1: MEV Classification Framework

One essential aspect to highlight when assessing the impact of MEV is the issue of trust. If market participants perceive MEV as unfairly tilted against them, it could erode their trust in the system, which might subsequently lead to reduced network usage. Although this trust deficit is a significant loss, it is inherently difficult to quantify in monetary terms.

The model, however, has its limitations. It may not capture all possible types of extractable value, and it simplifies the complexity of the MEV ecosystem by focusing on three dimensions. Furthermore, it doesn't comprehensively capture the multitude of actors in the MEV supply chain like searchers, builders, and wallets, among others. Despite these constraints, the framework's primary objective is to provide a broad overview of the MEV landscape.

Over time, there has been a growing consensus in the market that MEV is generally negative, leading to the emergence of various tools and techniques to mitigate MEV. For instance, Shutter Network and solutions from CowSwap and Gnosis aim to address and reduce the impact of MEV. A recent solution is called MEV Blocker, an exclusive RPC endpoint that protects your trades from MEV risks. It utilizes a network of searchers to scan for backrunning opportunities without engaging in frontrunning or sandwiching trades [75]. This trend indicates the ongoing evolution and development of the blockchain ecosystem as it grapples with the realities of MEV [1].

7 Conclusion

This research has undertaken a comprehensive exploration of Maximum Extractable Value (MEV) in the Ethereum ecosystem, offering a deep dive into the MEV landscape and selected types of MEV and their effects on network stability, user experiences, and overall transactional fairness.
The quantitative investigation focused on the major types of MEV, namely Arbitrage, Sandwich Attacks, and Liquidations, using the MEV detection script developed by Weintraub et al. [2] and comparing its results with data from Zeromev [67]. By inspecting approximately 2.5 million blocks, this research not only contributes to the understanding of MEV but also enriches the research community by providing an additional dataset. This work revealed a general rise in MEV and a significant surge in Sandwich Attacks, a toxic type of MEV. The Relayer ecosystem is witnessing diversification, evident from a decrease in Flashbots related block activities. The research findings provide insights into the effectiveness of MEV quantification scripts in identifying and categorizing MEV, as well as revealing its impact on the Ethereum ecosystem.

Categorizing MEV requires a deeper understanding of the economic and political dynamics within the entire system. The classification of MEV types involves complex considerations that extend beyond technical implementation, as it is an economic and political discussion that necessitates a holistic understanding. While MEV quantification scripts play a valuable role in providing initial insights, a broader and interdisciplinary perspective is crucial for a comprehensive understanding of MEV's impact and the development of robust solutions.

This work reveals that MEV poses a significant impact on the Ethereum network, manifesting both beneficial and detrimental effects. On one hand, MEV provides opportunities for profit through mechanisms like Arbitrage, contributing to the financial dynamism of the network. On the other hand, the rise of harmful MEV types, such as Sandwich Attacks, threaten the integrity of the network and create a potential barrier to Ethereum's promise of a decentralized and fair financial system.

To shed light on the interplay of various factors contributing to MEV, the concept Ethereum Microstructure is introduced. This model allows us to examine the complex dynamics of the Ethereum ecosystem and its components, paving the way for a more nuanced understanding of MEV. Additionally, a MEV Classification Framework is presented, an attempt to categorize different types of MEV based on their ethical implications. This framework serves as a stepping stone towards developing strategies that can mitigate the adverse impacts of harmful MEV types whilst reinforcing beneficial ones, contributing to the network's overall health.

This work has highlighted the urgency to address harmful MEV types, hinting towards an ongoing challenge that the Ethereum community needs to address. This challenge presents a vast area for future research, specifically focusing on strategies to mitigate harmful MEV impacts and enhancing transactional fairness in the Ethereum network. This line of research

holds significant promise, with potential to yield rewarding outcomes and innovative solutions to substantial challenges faced within the field potentially not only fixing problems in the decentralized ledger world but also in traditional finance [40].

7.1 Limitations

The research also unveiled the complexities of quantifying MEV. These complications arose from several factors, the most significant of which was the scope of the investigation. This research was primarily centered around decentralized exchange DEX - DEX based MEV, whereas CEX to DEX interactions, which are harder to quantify but undeniably present, were not thoroughly investigated. Furthermore, the ever-evolving landscape of protocols and their adoption presented additional challenges in the accurate measurement of MEV.

Another limitation of this research lies in the types of MEV explored. This work focused on known and major types of MEV, leaving room for unidentified types and those possibly known only to specific entities.

The issue of MEV detection heuristics also complicates the process, as different heuristics yield different results, and not all researchers provide open access to their methodologies and code. Thus, comparing different results is difficult. This is in line with Judmayer et al. [31], who already came to the conclusion that a quantification of MEV is inherently difficult due to the continuously evolving network environment and the variety of value-extraction mechanisms. The adoption of the transparent approach championed by Weintraub et al. [2], who publicly disclosed their results and code, and Hansson [37], with his detailed appendix outlining the employed heuristics, would be a beneficial practice for all researchers in this field.

Although this work offers an in-depth exploration of MEV on the Ethereum mainnet, it does contain some limitations. Layer-2 protocols were not thoroughly investigated. While early data from these protocols was examined to gain a quick understanding, a comprehensive analysis involving the setup of dedicated nodes was not pursued. This is attributed to the distinct characteristics of each L2 protocol, the vast volume of blocks they produce which requires substantial computational capacity, and the significant resource investment for node setup. Therefore, while L2 protocols were briefly considered, a full-scale examination was beyond the scope of this work.

7.2 Future Work

The concept of Maximal Extractable Value (MEV), with its multifaceted nature and extensive scope, sets the stage for plentiful future research prospects. The significance of MEV is broad and far-reaching, influencing an array of disciplines. Several potential areas of study are discussed in the following.

Cross-chain MEV amplifies the complexity of MEV in the context of numerous blockchain networks. As highlighted by Judmayer et al. there may be scenarios where miners, motivated by potential gains, resort to reordering or excluding transactions based on the occurrence of

cross-chain payments on other chains [76]. Consequently, comprehensive exploration of this cross-chain MEV landscape presents a unique and fruitful opportunity for future research. Such a study could delve into understanding the impact of MEV on the security and stability of various blockchains, and may even pave the way for the identification of potential methods to deminish its negative consequences.

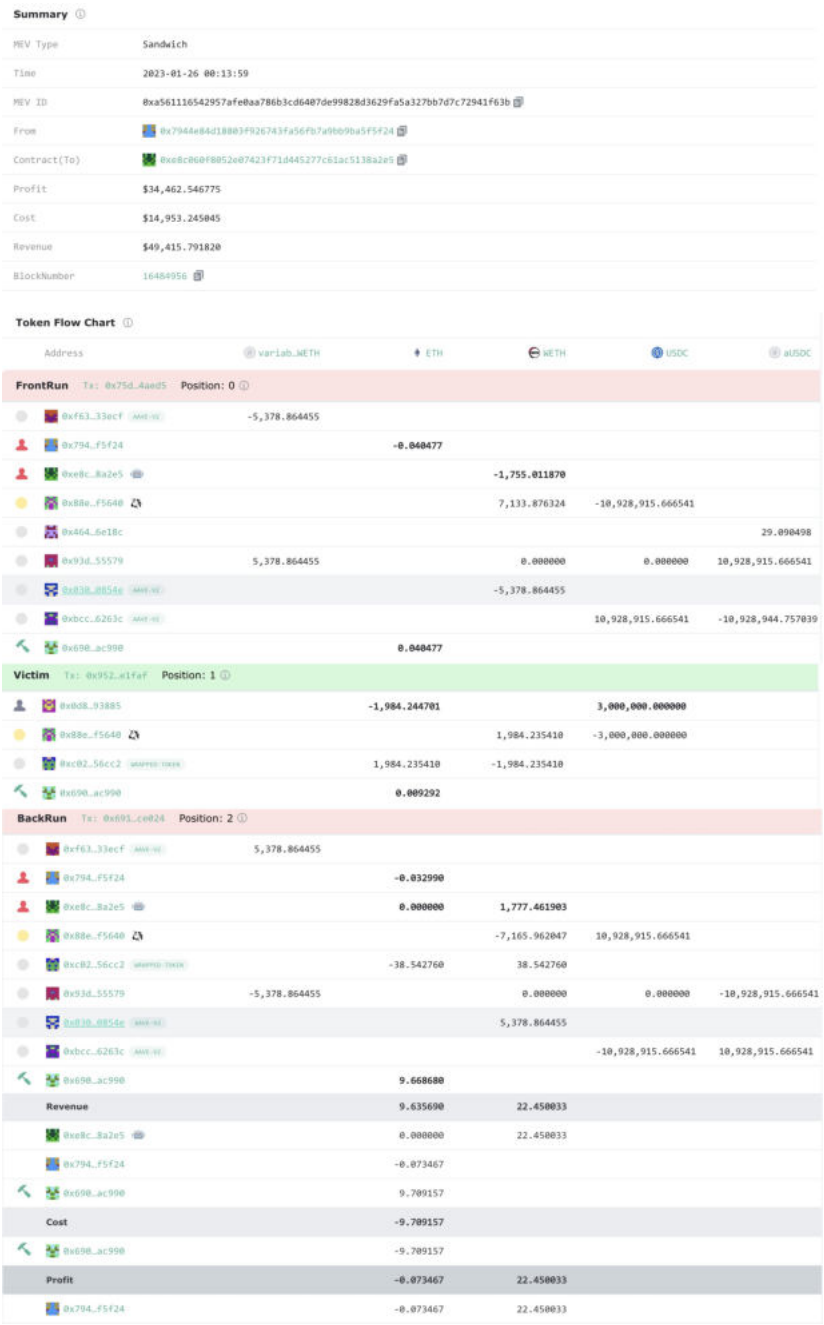
Moreover, the investigation of cross-domain MEV stands as another path for further inquiry. This research's prime goal would be to understand the relationship between MEV and multiple application classes, along with identifying the possible areas of overlap or influence [77]. The findings could offer valuable insights that may inform strategies for optimizing MEV across domains.

The transition to PoS presents a novel dimension for MEV, referred to as multi-block MEV. The deterministic nature of block proposal in PoS systems, where the upcoming block proposers within an epoch can be known in advance, enables the possibility of exploiting MEV over multiple blocks. Heimbach et al. underscore the security implications of multi-block MEV in PoS, and advocate for a greater degree of decentralization to address these risks [6]. While there have been preliminary attempts [78] to understand the full impact, the field may benefit from more in-depth and extensive research.

Additionally, an exciting avenue for future research is a more thorough exploration of Layer-2 protocols. This works early findings suggest a promising potential for uncovering more about the MEV landscape within these protocols. Given the growing volume of transactions happening on L2s, a comprehensive analysis, including setting up dedicated nodes for extensive data validation, could significantly enhance understanding.

In conclusion, these prospective research directions aim to explore further into the diverse nature of MEV. This exploration could provide a more comprehensive understanding of MEV's role and implications in the ever-evolving landscape of blockchain technology. It could also contribute towards the development of more robust systems and applications with a greater awareness of MEV. Additionally, these research efforts might also pave the way for improved quantification methods and foster a clearer understanding of the overall picture of MEV in the blockchain environment.

A Example1



Summary

MEV Type	Sandwich
Time	2023-01-26 00:13:59
MEV ID	0xa561116542957afe0aa786b3cd6407de99828d3629fa5a327bb7d7c72941f63b
From	0x7944e84d18803f926743fa56fb7a9bb9ba5f5f24
Contract(To)	0xe8c060f8052e07423f71d445277c61ac5138a2e5
Profit	$34,462.546775
Cost	$14,953.245045
Revenue	$49,415.791820
BlockNumber	16484956

Token Flow Chart

Address	variab..WETH	ETH	WETH	USDC	aUSDC
FrontRun Tx: 0x75d..4aed5 Position: 0					
0xf63..33ecf	-5,378.864455				
0x794..f5f24		-0.040477			
0xe8c..8a2e5			-1,755.011870		
0x88e..f5640			7,133.876324	-10,928,915.666541	
0x464..6e18c					29.098498
0x93d..55579	5,378.864455		0.000000	0.000000	10,928,915.666541
0x030..0854e			-5,378.864455		
0xbcc..6263c				10,928,915.666541	-10,928,944.757039
0x690..ac990		0.040477			
Victim Tx: 0x952..e1faf Position: 1					
0x0d8..93885		-1,984.244701		3,000,000.000000	
0x88e..f5640			1,984.235410	-3,000,000.000000	
0xc02..56cc2		1,984.235410	-1,984.235410		
0x690..ac990		0.009292			
BackRun Tx: 0x691..ce024 Position: 2					
0xf63..33ecf	5,378.864455				
0x794..f5f24		-0.032990			
0xe8c..8a2e5		0.000000	1,777.461903		
0x88e..f5640			-7,165.962047	10,928,915.666541	
0xc02..56cc2		-38.542760	38.542760		
0x93d..55579	-5,378.864455		0.000000	0.000000	-10,928,915.666541
0x030..0854e			5,378.864455		
0xbcc..6263c				-10,928,915.666541	10,928,915.666541
0x690..ac990		9.668680			
Revenue		9.635690	22.450033		
0xe8c..8a2e5		0.000000	22.450033		
0x794..f5f24		-0.073467			
0x690..ac990		9.709157			
Cost		-9.709157			
0x690..ac990		-9.709157			
Profit		-0.073467	22.450033		
0x794..f5f24		-0.073467	22.450033		

Figure A.1: Sandwich Attack

B Code and Data

The supporting code and datasets for this research are publicly available and can be accessed at the following URL: https://github.com/MaMwWun/MAThesis.

Code

The repository contains:

1. A script adapted from Weintraub et al. with adjustments to the sleep parameter at Coingecko prices and fixes for Python utility errors. Most relevant for this thesis were arbitrage.py, sandwiches.py and liquidation.py.
2. Code for scraping the ZeroMev API and visualizing the data using Python and Pandas. This script generates monthly MEV type absolute values, USD values, and detects outlier blocks.
3. A simple script for determining the start and end of a month, and the corresponding block range.

Datasets

The datasets included in this repository are:

1. Data from Script (Weintraub et al.) for Blocks 14.444.725 - 16.700.000
   - `sandwich_results.json`
   - `arbitrage_results.json`
   - `liquidation_results.json`
   - `prices.json`
2. Data from Zeromev scraped for Blocks 14.444.725 - 16.700.000
   - `ZeromevBlock 14.444.725:16.700.000`
3. A table detailing the months and dates corresponding to the Zeromev data.

API Links:
(https://blocks.flashbots.net/)
(https://data.zeromev.org/docs/#/public/mevBlock)

Table B.1: MEV Data Zeromev

Month	MEV Type	Absolute Number	USD Value
Mar-22	arb	10 958	3 208 443.47
	sandwich	6244	3 653 137.66
	liquid	37	1557.55
Apr-22	arb	38 151	8 754 916.76
	sandwich	26 867	16 297 688.92
	liquid	343	270 690.64
May-22	arb	54 588	10 583 167.02
	sandwich	27 043	13 830 327.83
	liquid	6508	3 098 351.12
Jun-22	arb	58 216	23 824 268.62
	sandwich	22 043	9 028 776.70
	liquid	9193	2 975 346.16
Jul-22	arb	62 104	6 156 662.36
	sandwich	31 991	5 760 589.67
	liquid	2720	92 616.94
Aug-22	arb	60 511	16 572 695.14
	sandwich	35 451	7 293 085.23
	liquid	1356	424 039.54
Sept-22	arb	78 014	2 303 461.92
	sandwich	31 789	5 062 043.10
	liquid	813	45 851.79
Oct-22	arb	54 003	2 232 617.02
	sandwich	63 193	8 001 698.65
	liquid	590	83 161.12
Nov-22	arb	70 072	5 877 358.16
	sandwich	66 253	11 078 402.69
	liquid	1399	66 738.43
Dec-22	arb	34 191	1 632 262.52
	sandwich	66 835	5 342 034.09
	liquid	335	28 957.06
Jan-23	arb	34 495	1 900 113.63
	sandwich	68 026	7 229 864.86
	liquid	137	12 143.63
Feb-23	arb	40 542	875 151.14
	sandwich	72 504	10 814 778.87
	liquid	143	35 415.71
Mar-23	arb	45 444	4 657 255.33
	sandwich	91 790	19 082 608.22
	liquid	306	494 644.43

Table B.2: MEV Data Script

Month	MEV Type	Absolute Number	USD Value
Mar-22	arb	19 632	0
	sandwich	6128	0
	liquid	154	0
Apr-22	arb	68 818	0
	sandwich	24 381	0
	liquid	3175	0
May-22	arb	109 222	0
	sandwich	25 725	0
	liquid	14 106	0
Jun-22	arb	107 545	0
	sandwich	19 905	0
	liquid	19 495	0
Jul-22	arb	121 307	0
	sandwich	28 406	0
	liquid	6091	0
Aug-22	arb	117 678	0
	sandwich	30 955	0
	liquid	3216	0
Sept-22	arb	131 482	0
	sandwich	28 865	0
	liquid	2032	0
Oct-22	arb	103 511	0
	sandwich	73 309	0
	liquid	1477	0
Nov-22	arb	136 294	0
	sandwich	66 003	0
	liquid	2870	0
Dec-22	arb	85 077	0
	sandwich	61 294	0
	liquid	788	0
Jan-23	arb	85 011	0
	sandwich	61 460	0
	liquid	358	0
Feb-23	arb	87 739	0
	sandwich	64 833	0
	liquid	368	0
Mar-23	arb	85 158	0
	sandwich	65 070	0
	liquid	673	0

Bibliography

[1] Sen Yang, Fan Zhang, Ken Huang, Xi Chen, Youwei Yang, and Feng Zhu, "SoK: MEV countermeasures: Theory and practice", pp. 8–10, Dec. 9, 2022. arXiv: 2212.05111[cs]. [Online]. Available: http://arxiv.org/abs/2212.05111 (visited on 01/15/2023).

[2] Ben Weintraub, Christof Ferreira Torres, Cristina Nita-Rotaru, and Radu State, "A flash(bot) in the pan: Measuring maximal extractable value in private pools", in *Proceedings of the 22nd ACM Internet Measurement Conference*, Oct. 25, 2022, pp. 3–5. doi: 10.1145/3517745.3561448. [Online]. Available: http://arxiv.org/abs/2206.04185 (visited on 02/15/2023).

[3] pmcgoohan. "Data sources & limitations", zeromev. (2023), [Online]. Available: http://info.zeromev.org/sources.html (visited on 06/27/2023).

[4] Ethereum Foundation. "Proof-of-stake (PoS)", ethereum.org. (2023), [Online]. Available: https://ethereum.org (visited on 06/13/2023).

[5] Mipasa. "Ethereum 2.0 - PoS post-merge analysis". (2022), [Online]. Available: https://mipasa.unbounded.network/publications/notebooks/Ethereum-2.0-PoS-Post-Merge-Analysis (visited on 06/13/2023).

[6] Lioba Heimbach, Lucianna Kiffer, Christof Ferreira Torres, and Roger Wattenhofer, "Ethereum's proposer-builder separation: Promises and realities", arXiv, May 30, 2023, pp. 5–8. arXiv: 2305.19037[cs]. [Online]. Available: http://arxiv.org/abs/2305.19037 (visited on 06/02/2023).

[7] Flashbots. "MEV-boost in a nutshell", MEV-Boost in a Nutshell. (2023), [Online]. Available: https://boost.flashbots.net/ (visited on 06/13/2023).

[8] Anton Wahrstätter, Liyi Zhou, Kaihua Qin, Davor Svetinovic, and Arthur Gervais, "Time to bribe: Measuring block construction market", arXiv, May 25, 2023, pp. 1–3, 12–14. arXiv: 2305.16468[cs]. [Online]. Available: http://arxiv.org/abs/2305.16468 (visited on 06/01/2023).

[9] Xingyu Lyu, Mengya Zhang, Xiaokuan Zhang, Jianyu Niu, Yinqian Zhang, and Zhiqiang Lin, "An empirical study on ethereum private transactions and the security implications", pp. 3–5, 10–13, Aug. 4, 2022. doi: 10.48550/arXiv.2208.02858. arXiv: 2208.02858[cs]. [Online]. Available: http://arxiv.org/abs/2208.02858 (visited on 01/20/2023).

[10] Toni Wahrstätter. "MEV-boost", mevboost.pics. (2023), [Online]. Available: https://mevboost.pics/mevboost.pics (visited on 06/13/2023).

[11] Agostino Capponi, Ruizhe Jia, and Ye Wang, "The evolution of blockchain: From lit to dark", pp. 25–34, Feb. 11, 2022. doi: 10.48550/arXiv.2202.05779. arXiv: 2202.05779[q-fin]. [Online]. Available: http://arxiv.org/abs/2202.05779 (visited on 02/12/2023).

[12] Kaihua Qin, Liyi Zhou, and Arthur Gervais, "Quantifying blockchain extractable value: How dark is the forest?", presented at the 2022 IEEE Symposium on Security and Privacy (SP), ISSN: 2375-1207, Dec. 2021, pp. 198–214. doi: 10.1109/SP46214.2022.9833734.

[13] Julien Piet, Jaiden Fairoze, and Nicholas Weaver, "Extracting godl [sic] from the salt mines: Ethereum miners extracting value", p. 20, 2022.

[14] Sam M. Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William J. Knottenbelt, "SoK: Decentralized finance (DeFi)", arXiv, Sep. 15, 2022, pp. 7–10. doi: 10.48550/arXiv.2101.08778. arXiv: 2101.08778[cs,econ,q-fin]. [Online]. Available: http://arxiv.org/abs/2101.08778 (visited on 01/04/2023).

[15] Jiahua Xu, Krzysztof Paruch, Simon Cousaert, and Yebo Feng, "SoK: Decentralized exchanges (DEX) with automated market maker (AMM) protocols", *ACM Computing Surveys*, vol. 55, no. 11, pp. 6–12, Nov. 30, 2023, issn: 0360-0300, 1557-7341. doi: 10.1145/3570639. arXiv: 2103.12732[cs,q-fin]. [Online]. Available: http://arxiv.org/abs/2103.12732 (visited on 06/14/2023).

[16] Ye Wang, Patrick Zuest, Yaxing Yao, Zhicong Lu, and Roger Wattenhofer, "Impact and user perception of sandwich attacks in the DeFi ecosystem", in *CHI Conference on Human Factors in Computing Systems*, New Orleans LA USA: ACM, Apr. 29, 2022, pp. 1–15, isbn: 978-1-4503-9157-3. doi: 10.1145/3491102.3517585. [Online]. Available: https://dl.acm.org/doi/10.1145/3491102.3517585 (visited on 01/27/2023).

[17] Kaihua Qin, Liyi Zhou, Yaroslav Afonin, Ludovico Lazzaretti, and Arthur Gervais, "CeFi vs. DeFi – comparing centralized to decentralized finance", arXiv, Jun. 16, 2021, p. 8. doi: 10.48550/arXiv.2106.08157. arXiv: 2106.08157[cs,q-fin]. [Online]. Available: http://arxiv.org/abs/2106.08157 (visited on 01/11/2023).

[18] Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais, "Attacking the DeFi ecosystem with flash loans for fun and profit", arXiv, Mar. 20, 2021, pp. 13–15. doi: 10.48550/arXiv.2003.03810. arXiv: 2003.03810[cs]. [Online]. Available: http://arxiv.org/abs/2003.03810 (visited on 01/24/2023).

[19] Dune. "Ethereum and its l2s: Growth comparison". (Jan. 24, 2023), [Online]. Available: https://dune.com/jhackworth/i-loves-l2s (visited on 06/16/2023).

[20] Vitalik Buterin. "An incomplete guide to rollups". (2021), [Online]. Available: https://vitalik.ca/general/2021/01/05/rollup.html (visited on 06/16/2023).

[21] Huy Ha, Vasiliki Vlachou, Quintus Kilbourn, and Cesare De Michellis, *MEV on I2*, Jan. 12, 2021. [Online]. Available: https://timroughgarden.github.io/fob21/reports/r11.pdf (visited on 10/01/2023).

[22] Flashbots. "Flashbots Transparency Report — October 2022 - The Flashbots Ship", The Flashbots Collective. (Nov. 9, 2022), [Online]. Available: https://collective.flashbots.net/t/flashbots-transparency-report-october-2022/721 (visited on 01/29/2023).

[23] pmcgoohan. "Miners frontrunning", r/ethereum. (Aug. 11, 2014), [Online]. Available: www.reddit.com/r/ethereum/comments/2d84yv/miners_frontrunning/ (visited on 01/23/2023).

[24] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels, "Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges", p. 14, Apr. 10, 2019. doi: 10.48550/arXiv.1904.05234. arXiv: 1904.05234[cs]. [Online]. Available: http://arxiv.org/abs/1904.05234 (visited on 01/24/2023).

[25] Joseph Bonneau, "Why buy when you can rent?", in *Financial Cryptography and Data Security*, Jeremy Clark, Sarah Meiklejohn, Peter Y.A. Ryan, Dan Wallach, Michael Brenner, and Kurt Rohloff, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2016, pp. 19–26, isbn: 978-3-662-53357-4. doi: 10.1007/978-3-662-53357-4_2.

[26] Miles Carlsten, Harry Kalodner, S. Matthew Weinberg, and Arvind Narayanan, "On the instability of bitcoin without the block reward", in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16, New York, NY, USA: Association for Computing Machinery, 2016, pp. 154–167, isbn: 978-1-4503-4139-4. doi: 10.1145/2976749.2978408. [Online]. Available: https://doi.org/10.1145/2976749.2978408 (visited on 01/02/2023).

[27] Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels, *Order-fairness for byzantine consensus*, Report Number: 269, 2020. [Online]. Available: https://eprint.iacr.org/2020/269 (visited on 01/01/2023).

[28] Mahimna Kelkar, Soubhik Deb, and Sreeram Kannan, "Order-fair consensus in the permissionless setting", in *Proceedings of the 9th ACM on ASIA Public-Key Cryptography Workshop*, ser. APKC '22, New York, NY, USA: Association for Computing Machinery, 2022, pp. 3–14, isbn: 978-1-4503-9174-0. doi: 10.1145/3494105.3526239. [Online]. Available: https://doi.org/10.1145/3494105.3526239 (visited on 01/24/2023).

[29] Klaus Kursawe, *Wendy, the good little fairness widget*, Jul. 16, 2020. doi: 10.48550/arXiv.2007.08303. arXiv: 2007.08303[cs]. [Online]. Available: http://arxiv.org/abs/2007.08303 (visited on 01/01/2023).

[30] Kushal Babel, Philip Daian, Mahimna Kelkar, and Ari Juels, "Clockwork finance: Automated analysis of economic security in smart contracts", Sep. 9, 2021. doi: 10.48550/arXiv.2109.04347. arXiv: 2109.04347[cs]. [Online]. Available: http://arxiv.org/abs/2109.04347 (visited on 01/23/2023).

[31] Aljosha Judmayer, Nicholas Stifter, Philipp Schindler, and Edgar Weippl, "Estimating (miner) extractable value is hard, let's go shopping!", pp. 2–7, 2021. [Online]. Available: https://eprint.iacr.org/2021/1231 (visited on 01/01/2023).

[32] Bruno Mazorra, Michael Reynolds, and Vanesa Daza, "Price of MEV: Towards a game theoretical approach to MEV", pp. 1–7, Aug. 29, 2022. doi: 10.48550/arXiv.2208.13464. arXiv: 2208.13464[cs]. [Online]. Available: http://arxiv.org/abs/2208.13464 (visited on 01/01/2023).

[33] Jeffrey R. Russell and Robert F. Engle, "CHAPTER 7 - analysis of high-frequency data", in *Handbook of Financial Econometrics: Tools and Techniques*, ser. Handbooks in Finance, YACINE Aït-sahalia and LARS PETER Hansen, Eds., vol. 1, San Diego: North-Holland, Jan. 1, 2010, pp. 383–426. doi: 10.1016/B978-0-444-50897-3.50010-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444508973500109 (visited on 06/09/2023).

[34] Stephane Gosselin. "Introducing the MEV supply chain". (2022), [Online]. Available: https://flashbots.mirror.xyz/bqCakwfQZkMsq63b50vib-nibo5eKai0QuK7m-Dsxpo (visited on 06/09/2023).

[35] Flashbots. "Flashbots transparency dashboard". (2023), [Online]. Available: https://transparency.flashbots.net/ (visited on 06/25/2023).

[36] Liyi Zhou, Xihan Xiong, Jens Ernstberger, Stefanos Chaliasos, Zhipeng Wang, Ye Wang, Kaihua Qin, Roger Wattenhofer, Dawn Song, and Arthur Gervais, "SoK: Decentralized finance (DeFi) attacks", arXiv, Sep. 20, 2022. arXiv: 2208.13035[cs]. [Online]. Available: http://arxiv.org/abs/2208.13035 (visited on 01/23/2023).

[37] Magnus Hansson, "Arbitrage in crypto markets: An analysis of primary ethereum blockchain data", Rochester, NY, Nov. 16, 2022, pp. 9–14. doi: 10.2139/ssrn.4278272. [Online]. Available: https://papers.ssrn.com/abstract=4278272 (visited on 01/29/2023).

[38] Christof Ferreira Torres, Ramiro Camino, and Radu State, "Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain", p. 18, 2021.

[39] Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark, "SoK: Transparent dishonesty: Front-running attacks on blockchain", arXiv, Apr. 9, 2019, pp. 4–13. arXiv: 1902.05164[cs]. [Online]. Available: http://arxiv.org/abs/1902.05164 (visited on 01/25/2023).

[40] Lioba Heimbach and Roger Wattenhofer, "SoK: Preventing transaction reordering manipulations in decentralized finance", Sep. 21, 2022, pp. 2–4. doi: 10.1145/3558535.3559784. arXiv: 2203.11520[cs]. [Online]. Available: http://arxiv.org/abs/2203.11520 (visited on 01/28/2023).

[41] Liyi Zhou, Kaihua Qin, Antoine Cully, Benjamin Livshits, and Arthur Gervais, "On the just-in-time discovery of profit-generating transactions in DeFi protocols", arXiv, Mar. 3, 2021, pp. 1–4. arXiv: 2103.02228[cs]. [Online]. Available: http://arxiv.org/abs/2103.02228 (visited on 01/24/2023).

[42] Liyi Zhou, Kaihua Qin, Christof Ferreira Torres, Duc V. Le, and Arthur Gervais, "High-frequency trading on decentralized on-chain exchanges", arXiv, Sep. 29, 2020, pp. 1–13. arXiv: 2009.14021[cs]. [Online]. Available: http://arxiv.org/abs/2009.14021 (visited on 01/23/2023).

[43] Ye Wang, Yan Chen, Haotian Wu, Liyi Zhou, Shuiguang Deng, and Roger Wattenhofer, "Cyclic arbitrage in decentralized exchanges", arXiv, Jan. 14, 2022, pp. 1–9. arXiv: 2105.02784[cs,q-fin]. [Online]. Available: http://arxiv.org/abs/2105.02784 (visited on 01/25/2023).

[44] Patrick Züst, "Analyzing and preventing sandwich attacks in ethereum", *Bachelor Thesis*, pp. 7–12, 2021. [Online]. Available: https://pub.tik.ee.ethz.ch/students/2021-FS/BA-2021-07.pdf.

[45] Hai Jin, Chenchen Li, Jiang Xiao, Teng Zhang, Xiaohai Dai, and Bo Li, "Detecting arbitrage on ethereum through feature fusion and positive-unlabeled learning", *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3660–3671, Dec. 2022, Conference Name: IEEE Journal on Selected Areas in Communications, issn: 1558-0008. doi: 10.1109/JSAC.2022.3213335.

[46] EigenPhi. "MEV data", EigenPhi. (2023), [Online]. Available: https://eigenphi.io/ (visited on 01/25/2023).

[47] Flashbots. "MEV explore". (2023), [Online]. Available: https://explore.flashbots.net/ (visited on 01/25/2023).

[48] Igor Makarov and Antoinette Schoar, "Trading and arbitrage in cryptocurrency markets", *Journal of Financial Economics*, vol. 135, no. 2, pp. 293–319, Feb. 1, 2020, issn: 0304-405X. doi: 10.1016/j.jfineco.2019.07.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304405X19301746 (visited on 01/10/2023).

[49] Alain P. Chaboud, Benjamin Chiquoine, Erik Hjalmarsson, and Clara Vega, "Rise of the machines: Algorithmic trading in the foreign exchange market", *The Journal of Finance*, vol. 69, no. 5, pp. 2045–2084, 2014, issn: 1540-6261. doi: 10.1111/jofi.12186. (visited on 01/25/2023).

[50] Jan Arvid Berg, Robin Fritsch, Lioba Heimbach, and Roger Wattenhofer, "An empirical study of market inefficiencies in uniswap and SushiSwap", arXiv, May 20, 2022, pp. 6–8. doi: 10.48550/arXiv.2203.07774. arXiv: 2203.07774[cs,q-fin]. [Online]. Available: http://arxiv.org/abs/2203.07774 (visited on 01/25/2023).

[51] Daniel Perez, Sam M. Werner, Jiahua Xu, and Benjamin Livshits, "Liquidations: DeFi on a knife-edge", in vol. 12675, 2021, pp. 1–5. doi: 10.1007/978-3-662-64331-0_24. arXiv: 2009.13235[q-fin]. [Online]. Available: http://arxiv.org/abs/2009.13235 (visited on 01/26/2023).

[52] Kaihua Qin, Liyi Zhou, Pablo Gamito, Philipp Jovanovic, and Arthur Gervais, "An empirical study of DeFi liquidations: Incentives, risks, and instabilities", in *Proceedings of the 21st ACM Internet Measurement Conference*, Nov. 2, 2021, pp. 336–350. doi: 10.1145/3487552.3487811. arXiv: 2106.06389[cs,q-fin]. [Online]. Available: http://arxiv.org/abs/2106.06389 (visited on 01/26/2023).

[53] Aave. "Liquidations". (2023), [Online]. Available: https://docs.aave.com/faq/liquidations (visited on 01/27/2023).

[54] Austin Adams, Xin Wan, and Noah Zinsmeister. "Uniswap v3 TWAP oracles in proof of stake", Uniswap Protocol. (Oct. 27, 2022), [Online]. Available: https://uniswap.org/uniswap-v3-oracles (visited on 06/09/2023).

[55] Robert McLaughlin, Christopher Kruegel, and Giovanni Vigna, "A large scale study of the ethereum arbitrage ecosystem", p.14, 2023.

[56] Torgin Mackinga, Tejaswi Nadahalli, and Roger Wattenhofer, "TWAP oracle attacks: Easier done than said?", Report Number: 445, 2022, p.2. [Online]. Available: https://eprint.iacr.org/2022/445 (visited on 06/09/2023).

[57] Etherscan. "Ethereum full node sync (archive) chart | etherscan", Ethereum (ETH) Blockchain Explorer. (2023), [Online]. Available: http://etherscan.io/chartsync/chainarchive (visited on 06/18/2023).

[58] Coingecko. "Crypto API documentation", CoinGecko. (2023), [Online]. Available: https://www.coingecko.com/en/api/documentation (visited on 06/21/2023).

[59] Flashbots. "Flashbots blocks API". (2023), [Online]. Available: https://blocks.flashbots.net/ (visited on 06/21/2023).

[60] Ethernodes. "Clients - the ethereum network & node explorer". (2023), [Online]. Available: https://www.ethernodes.org/ (visited on 06/19/2023).

[61] Ethereum Foundation. "Ethereum archive node", ethereum.org. (2023), [Online]. Available: https://ethereum.org (visited on 06/19/2023).

[62] Ethereum Foundation. "JSON-RPC API", ethereum.org. (2023), [Online]. Available: https://ethereum.org (visited on 06/19/2023).

[63] Infura. "Ethereum RPCs, methods and calls", Infura Blog | Tutorials, Case Studies, News, Feature Announcements. (Mar. 12, 2020), [Online]. Available: https://blog.infura.io/post/ethereum-rpcs-methods#using-eth-get-logs (visited on 06/19/2023).

[64] Ethereum Foundation. "Events, logs — solidity documentation". (2023), [Online]. Available: https://docs.soliditylang.org/en/latest/contracts.html#events (visited on 06/19/2023).

[65] DefiLlama. "DefiLlama TVL randkings", DefiLlama. (2023), [Online]. Available: https://defillama.com/protocols (visited on 06/28/2023).

[66] Anton Wahrstätter, Jens Ernstberger, Aviv Yaish, Liyi Zhou, Kaihua Qin, Taro Tsuchiya, Sebastian Steinhorst, Davor Svetinovic, Nicolas Christin, Mikolaj Barczentewicz, and Arthur Gervais, "Blockchain censorship", arXiv, Jun. 2, 2023, p. 10. arXiv: 2305.18545[cs]. [Online]. Available: http://arxiv.org/abs/2305.18545 (visited on 06/13/2023).

[67] Pmcgoohan. "Zeromev". (2023), [Online]. Available: https://zeromev.org/ (visited on 02/03/2023).

[68] Flashbots. "FRP-24: Quantifying MEV on L2s", The Flashbots Collective. Section: FRPs - Flashbots Research Proposals. (Oct. 5, 2022), [Online]. Available: https://collective.flashbots.net/t/frp-24-quantifying-mev-on-l2s/450 (visited on 06/21/2023).

[69] Akaki Mamageishvili, Mahimna Kelkar, Jan Christoph Schlegel, and Edward W. Felten, "Buying time: Latency racing vs. bidding in fair transaction ordering", arXiv, Jun. 3, 2023, pp. 1–3. doi: 10.48550/arXiv.2306.02179. arXiv: 2306.02179[cs,econ]. [Online]. Available: http://arxiv.org/abs/2306.02179 (visited on 06/08/2023).

[70] Philip Daian. "Mev wat do?" (2021), [Online]. Available: https://pdaian.com/blog/mev-wat-do/ (visited on 07/01/2023).

[71] Philémon Poux, Primavera De Filippi, and Bruno Deffains, "Maximal extractable value and the blockchain commons", pp. 7–11, Aug. 23, 2022. doi: 10.2139/ssrn.4198139. [Online]. Available: https://papers.ssrn.com/abstract=4198139 (visited on 01/26/2023).

[72] Mikolaj Barczentewicz, "MEV on ethereum: A policy analysis", Rochester, NY, Jan. 23, 2023, pp. 10–20. [Online]. Available: https://papers.ssrn.com/abstract=4332703 (visited on 01/26/2023).

[73] Gurdip Kaur, Arash Habibi Lashkari, Iman Sharafaldin, and Ziba Habibi Lashkari, "Smart contracts and DeFi security and threats", in *Understanding Cybersecurity Management in Decentralized Finance*. Cham: Springer International Publishing, 2023, pp. 91–111, Series Title: Financial Innovation and Technology, isbn: 978-3-031-23339-5 978-3-031-23340-1. doi: 10.1007/978-3-031-23340-1_5. [Online]. Available: https://link.springer.com/10.1007/978-3-031-23340-1_5 (visited on 01/12/2023).

[74] Lewis Gudgeon, Daniel Perez, Dominik Harz, Benjamin Livshits, and Arthur Gervais, "The decentralized financial crisis", in *2020 Crypto Valley Conference on Blockchain Technology (CVCBT)*, Jun. 2020, pp. 1–15. doi: 10.1109/CVCBT50464.2020.00005.

[75] MEV Blocker. “MEV blocker”. (2023), [Online]. Available: https://mevblocker.io (visited on 06/29/2023).

[76] Aljosha Judmayer, Nicholas Stifter, Alexei Zamyatin, Itay Tsabary, Ittay Eyal, Peter Gazi, Sarah Meiklejohn, and Edgar Weippl, *Pay to win: Cheap, crowdfundable, cross-chain algorithmic incentive manipulation attacks on PoW cryptocurrencies*, Report Number: 775, 2019. [Online]. Available: https://eprint.iacr.org/2019/775 (visited on 06/23/2023).

[77] Alexandre Obadia, Alejo Salles, Lakshman Sankar, Tarun Chitra, Vaibhav Chellani, and Philip Daian, “Unity is strength: A formalization of cross-domain maximal extractable value”, Dec. 5, 2021, pp. 1–8. doi: 10.48550/arXiv.2112.01472. arXiv: 2112.01472[cs]. [Online]. Available: http://arxiv.org/abs/2112.01472 (visited on 01/23/2023).

[78] Johannes Rude Jensen, Victor von Wachter, and Omri Ross, *Multi-block MEV*, Jun. 12, 2023. doi: 10.48550/arXiv.2303.04430. arXiv: 2303.04430[cs]. [Online]. Available: http://arxiv.org/abs/2303.04430 (visited on 06/23/2023).