SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Analyzing the Role of Bridges in Cross-Chain MEV Extraction

Danut Ilisei

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Analyzing the Role of Bridges in Cross-Chain MEV Extraction

Analyse der Rolle von Bridges bei Cross-Chain MEV Extraction

Author:	Danut Ilisei
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	M.Sc. Burak Öz
Submission Date:	15.05.2024

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.05.2024 Danut Ilisei

Acknowledgments

I want to express my deepest gratitude to my family for their unconditional support and encouragement throughout my studies. Their love has been a constant source of strength and motivation.

Special thanks to my advisor, Burak Öz, for accepting my thesis topic and for his invaluable guidance and meticulous attention to detail, ensuring that my research was conducted at the highest possible standard. I am equally grateful to Prof. Dr. Florian Matthes for allowing me to conduct my thesis under his chair.

Finally, I am grateful to my girlfriend, Ioana, for her continuous support throughout the thesis journey.

Abstract

Blockchain technologies have evolved into comprehensive platforms that support various applications across several blockchains. However, the development of multiple blockchains has divided resources and markets. To overcome the fragmentation, interoperability solutions such as blockchain bridges have been created to transfer assets and data between different networks. Concurrently, the proliferation of blockchain bridges exposed additional extractable value to the market participants. While the phenomenon of Maximal Extractable Value (MEV) has been extensively analyzed within the context of individual blockchains, its effects, and implications in an ecosystem where multiple blockchains are involved remain largely unexplored.

This thesis investigates MEV extraction across blockchains, notably between Ethereum and Polygon, through the Polygon bridge. Analyzing 4,488 instances, it uncovers profitable and loss-making cross-chain arbitrage patterns influenced by market volatility and bridge dynamics, with an average of 32 daily occurrences. Our study shows that the tokens involved in the bridge operation have low recognition and market capitalization. It also reveals that a small group of three actors dominates 95% of the activity, highlighting the complex strategies in MEV extraction. This research emphasizes the need for transparency in cross-chain transactions and contributes insights into the evolving landscape of blockchain MEV extraction, advocating for integrity and decentralization in blockchain ecosystems.

Contents

1. Introduction

In 2009, when Bitcoin as the first permissionless blockchain technology became known, a blockchain was primarily seen as a means of storing value, serving a limited purpose. Over time, blockchains have evolved and proved their utility and security. These platforms can now support various applications such as gaming, decentralized finance, governance, supply chain, etc [1]. While the innovations have generated a wide range of valuable applications, the side effect was segmentation in the infrastructure, leading to multiple blockchains with different states and consensus algorithms. While this was necessary since different blockchains serve other purposes, the proliferation of multiple blockchains has led to fragmented resources and markets. To enable interaction among blockchains, protocols known as interoperability solutions emerged. The purpose of these protocols is to enable the passing of messages from one blockchain to another in a secure and fast way.

Along this progress, Maximal Extractable Value, widely known as MEV, has emerged as an inevitable aspect of blockchain ecosystems [2]. This results from both the fundamental properties of decentralized networks and how transactions are ordered and validated. MEV characterizes the potential economic value that several actors, such as miners, validators, or searchers, can capture through strategically ordering transactions within blocks. Since these strategies had financial gains to offer, there was a high incentive for building communities that studied the phenomenon and profited from it. While MEV can have different properties depending on the blockchain it applies to, the most extensive research has been done in the Ethereum ecosystem. In the early stages of MEV occurrence, actors have greatly benefited from the unfamiliarity with the concept of the other market participants [3]. It has become an important topic in the blockchain community, with profits exceeding $600 million on the Ethereum blockchain alone [3]. Also, it has been shown that MEV can be a significant issue on public blockchains, especially those with high transaction volumes like Ethereum [2]. It can create severe vulnerabilities in the consensus layer, potentially jeopardizing the blockchain's existence and its users. Therefore, multiple initiatives have either already been implemented or are in progress and have had the goal of creating a healthy MEV supply chain in a trustless way.

MEV has been extensively studied and, consequently, evolved within the constraints of a single blockchain [2], [3], [4]. However, the same cannot be said of its advancement in a context where multiple blockchains are involved, and the possibilities are relatively unexplored. In this setting, users can interact with multiple blockchains through interoperability protocols. Cross-chain transactions enable the transfer of assets and data across different blockchain networks. These transactions are facilitated through bridges and infrastructure components that introduce another degree of complexity to the dynamics of MEV that has yet to be explored. Blockchain bridges are protocols that facilitate the transfer of assets or

information, enabling interoperability across separate blockchain networks. Multiple bridge protocols and design details are to be accounted for when it comes to their impact on cross-chain MEV. It is worth noting that other cross-chain protocols besides bridges can influence it.

1.1. Motivation

The growing number of cross-chain transactions [5] emphasizes the importance of studying the role of bridges. The success of these bridges is closely related to the operational success of decentralized applications, the security of asset transactions, and the general stability of blockchain ecosystems. Moreover, the success of bridges today also relies on their ability to efficiently manage MEV and safeguard their users from adverse side effects.

Exploring the phenomenon of MEV in a single blockchain context has often been described as navigating into a dark forest [4]. This metaphor comes from the difficulty of grasping the true impact and profits gained by different actors by utilizing various blockchain mechanisms. Thankfully, recent innovations such as creating mev-inspect-py [6], which is a tool used for inspecting MEV on Ethereum, have played an important role in illuminating the dark forest. Yet, the MEV phenomenon may extend beyond individual chains and into a more extensive, cross-domain network, although this remains largely unexplored.

1.2. Scope of the Thesis

The scope of the thesis will predominantly focus on understanding the mechanisms of MEV extraction across various blockchain networks facilitated through blockchain bridges. Although our significant attention is on MEV extraction through blockchain bridges, we will also touch upon related concepts, such as MEV extraction across blockchains without the infrastructure provided by blockchain bridges or value extraction across related domains, such as centralized exchanges. Through theoretical investigation and hands-on data analysis, where we will conduct a study on the use of blockchain bridges to extract MEV across different blockchains, we aim to show how blockchain bridges operate and how various players use them for financial benefit. We will also perform a speculative analysis to imagine possible developments and trends in MEV extraction. By understanding the techniques used by the participants, the intricacies of the systems, and market dynamics, we aim to forecast how MEV-related activities will grow over time, as well as the more significant implications for blockchain ecosystems.

1.3. Research Challenges

Cross-chain MEV can be challenging to narrow down due to its composition of various blockchain domains, each with its own set of smart contracts, token standards, and consensus rules. To evaluate the reliability of bridge mechanisms and understand how they

interact with blockchains, it is necessary to clearly specify what set of rules are applied. However, due to the large number of bridges and blockchains, it can take time to create a general-purpose analysis framework.

The area where miners and validators can take advantage of economic opportunities is known as this action space [7]. When it comes to multi-chain, one more layer of complexity is added: the bridge itself. The action space of the bridge also has to be accounted for. Due to the different types of implementations, the action space can be dense and complex. Defining trust models and understanding how protocols interact is essential. Having these relationships formally defined can help highlight the critical factors.

Additionally, as interconnections between various smart contract domains grow, new MEV extraction opportunities open up. Cross-domain MEV can take advantage of new forms of blockchain interactions, such as cross-domain financing [8] and voting. In addition, new opportunities for value extraction are introduced by the bridges, oracles, and governance mechanisms that enable communication across different blockchains.

As a result of the expanding use of cross-chain bridges in the blockchain community, understanding cross-chain MEV is essential. The complexity of interrelated domains, action spaces, and trust models must be considered when it comes to solutions to the MEV issues, which go beyond the bounds of a single network. To correctly solve the complex problems of cross-chain MEV and establish secure and fair connections, developing theoretically sound and practically usable models is essential.

1.4. Research Questions

With the aim of illuminating the complex character of the study topic, this thesis will thoroughly investigate the following research questions.

1. What are the existing interoperability solutions for connecting different blockchain networks?
    - What is the current status of interoperability solutions, with a particular focus on blockchain bridges?
    - Is there a formal classification of blockchain bridges that categorizes them according to their functionality, security features, and decentralization capabilities?
    - What specific functionalities of blockchain bridges have the potential to generate MEV?
2. What does the existing literature reveal about the current state of MEV in the context of cross-chain operations?
    - Is there any work conducted on the extraction of MEV utilizing the infrastructure of blockchain bridges?
3. How can we quantify cross-chain MEV extraction enabled by a selected blockchain bridge?

- Is it possible to identify historical cases of cross-chain MEV extraction, and if so, what methodologies or tools are available for such identification?

4. What are the risks of cross-domain MEV?
    - How can we explore further strategies to mitigate the negative side effects of MEV in the cross-chain domain?

2. Background

This chapter gives the readers a thorough grasp of blockchain technologies and how and why they have evolved. We will present them and chart their development to the present day in the order of their evolution. We shall focus on Ethereum and Polygon, some of the primary blockchain platforms leading the blockchain market. We are going to dive into these platforms since they provide potent features that are useful not only to comprehend the potential of blockchain technologies but also to enable the extraction of value between different domains. Blockchain interoperability and its significance within the blockchain ecosystem will also be examined. Interconnected blockchain systems have the potential to completely transform several industries, including finance. Also, we intend to contrast traditional alternatives with the financial opportunities made possible by blockchain technologies. This comparison will emphasize the distinct value proposition of blockchain technology in the financial sector. This exploration will provide insights into the possibilities and challenges of blockchain technologies and the capabilities offered to different actors by these systems.

2.1. Blockchain Technologies

Blockchain technologies have revolutionized multiple sectors, such as financial services and supply chains, by offering decentralized and immutable transaction methods. A blockchain is a decentralized, distributed ledger technology that records transactions within a network of computers. Furthermore, the data is tamper-resistant and immutable. Each blockchain block contains a cryptographic hash of the previous block, a timestamp, and transaction data, forming a chronological chain of blocks [9].

Blockchain networks rely on consensus methods to enable nodes to reach a consensus over the ledger's current state without needing a central authority. Consensus algorithms play a significant role in distributed ledger security and dependability. The consensus is necessary because of the Byzantine General's Problem. In the scenario, a group of Byzantine generals surrounded a city and had to coordinate their attack or withdrawal using messengers. However, some treacherous generals may transmit contradicting information to derail the operation. The aim is to devise a procedure that allows loyal generals to reach a unanimous decision in the face of disloyal generals and untrustworthy communication routes [10]. Blockchain systems rely heavily on consensus since they require many nodes (computers) that maintain a distributed ledger of transactions. Without an agreement, there is no way to ensure that all nodes agree on the legitimacy and sequence of transactions, resulting in inconsistencies and attack vulnerabilities.

Various blockchain systems employ different mechanisms for choosing the consensus par-

ticipants or the next block proposer. The two most frequent types are proof-of-work (PoW) and proof-of-stake (PoS). Bitcoin's PoW algorithm needs miners to solve complicated mathematical problems to validate transactions and generate new blocks. This provides network security through computational efforts. Conversely, PoS, promoted by platforms such as PoS-Ethereum and Polygon PoS, awards block validation privileges based on network stake. This encourages users to operate in the network's best interests while reducing the energy consumption problems associated with PoW. Both approaches effectively enable decentralization and trust in blockchain networks, but in different ways that serve various purposes within the blockchain ecosystem [11]. It is important to note that the consensus mechanisms used by the blockchains can influence other aspects of it.

2.1.1. Ethereum

Vitalik Buterin created Ethereum in 2013 and publicly released it in 2015 [12]. Unlike Bitcoin, which is essentially digital money, Ethereum is a decentralized platform that allows for the design and execution of software programs as decentralized applications. Ethereum's core blockchain technology supports a wide range of functions, including decentralized finance (DeFi), non-fungible tokens (NFTs), decentralized autonomous organizations (DAOs), and others. To better understand how Ethereum operates, we will explain the core concepts of the system. It is worth noting that the Ethereum blockchain evolves continuously. Ethereum has changed over time through several Ethereum Improvement Proposals (EIPs). These EIPs offer updates, additions, and modifications to the Ethereum network, including anything from feature-complete new functionality to parameter adjustments or technical advancements. Before being incorporated into the Ethereum protocol, EIPs must pass a rigorous testing, implementation, and discussion process led by developers and the community. For example, Ethereum now uses proof-of-stake, but this has only sometimes been the case. One of the most notable upgrades was "The Merge." This was accomplished by upgrading the original proof-of-work mechanism to proof-of-stake. Unless specified otherwise, when we refer to Ethereum in the following chapters, we refer to its version after the Shanghai upgrade.

Nodes

Ethereum nodes are individual computers that connect using a peer-to-peer network and establish the Ethereum network. They are responsible for keeping a copy of the blockchain's ledger and communicating with one another to validate and relay transactions. They do this by adhering to the rule of the established consensus, which is proof-of-stake. The Ethereum network can be joined by any software program that implements the Ethereum specifications, making Ethereum a permissionless network. It's also worth noting that there are multiple types of nodes in the network: full nodes, which hold blockchain data and participate in block validation; light nodes, which do not participate in block validation but can verify blockchain data by connecting to a full node and archive nodes, which are full nodes that, additionally, maintains storage of historical blockchain states. Furthermore, the Ethereum

nodes have different implementations using different programming languages. This is a sign of network maturity. Various node implementations contribute to the overall security and resilience of the network. Nodes are vital in maintaining the Ethereum network's integrity and security [13].

Gas

In the Ethereum network, the term gas refers to the amount of processing power needed to carry out particular tasks. Ethereum transactions need computing resources, which must be paid for to keep spam out of the system and keep Ethereum nodes from becoming trapped in an endless computation loop. The gas fee is used to pay the computation. The cost per unit of gas is multiplied by the volume of gas utilized to complete an operation to determine this price. Whether a transaction is successful or not, the gas fee must be paid. Users must use ether (ETH), the native currency of Ethereum, to cover gas expenses. The unit of measurement for gas prices is commonly gwei, an ETH denomination that stands for giga-wei, which is a billion wei. The network will process the transaction faster as more gas is paid for. To prevent spending too much on gas, monitoring gas prices is critical, mainly when network congestion is at its worst. You can specify how much gas you will pay for when you submit a transaction. This gas price is effectively a bid to include your transaction in the following blocks. Finding the ideal balance when determining the gas amount can be challenging. If you provide too little, validators might ignore your transaction, which could lead to delays or even failures. On the other hand, if you offer too much, ETH can be spent unnecessarily [14].

The number of transactions a blockchain can have over time and its degree of centralization must be balanced. This nails down to the amount of gas a block can have. Many validators couldn't afford the computational power needed to meet the requirement if the gas limit was set too high. Conversely, the chain's throughput will be constrained if the gas restriction is too low. Ethereum had a set gas limit for every block before EIP1559. Nowadays, the block gas limit is still limited to between 15 and 30 million. However, it is no longer set. The network congestion level has an impact on how much gas a validator will aim to fit into a block [15].

Accounts

Accounts are essential in the Ethereum ecosystem because they interact with the blockchain. There are two sorts of accounts: externally owned accounts (EOAs) and smart contracts. EOAs are governed by private keys and represent persons or entities that own Ether or other assets. A private key is part of the key pairs defined by asymmetric cryptography, often known as public-key cryptography. This is a prominent cryptographic method included in many security protocols. It makes use of key pairs: public keys, which may be freely shared, and private keys, which are only known by the owner. The creation of these key pairings is based on mathematical procedures. This approach is utilized in digital signatures, encryption to protect data transfer, and TLS/SSL to secure internet connections. Asymmetric

cryptography provides safe communication across insecure channels without requiring both parties to share a secret key beforehand. In this approach, material encrypted using a public key can only be decrypted using its matching private key and vice versa. An Ethereum EOA account is represented by its public-private key pair. Contract accounts, on the other hand, are managed by smart contracts that perform predetermined functions when triggered by transactions (initiated by EOAs). Accounts enable the movement of assets and the execution of smart contracts logic on the Ethereum network [16].

Transactions

Transactions on the Ethereum blockchain are signed messages delivered from an EOA to an Ethereum node that cause state changes. These modifications may involve moving Ether between accounts, deploying smart contracts, or executing logic inside existing smart contracts. Key information like the sender, destination addresses, Ether transfer amount, and optional data are required to send a transaction. The transaction cost is variable and depends on parameters such as the amount of computation required (gas) to execute the state change, how congested the network is, and how fast the sender wishes to record its transaction on the ledger [17].

Blocks

Blocks are the fundamental units of the Ethereum blockchain. They comprise batches of validated transactions and include a header with metadata such as a timestamp, a reference to the preceding block, and a cryptographic hash of the block's content. To maintain the integrity of the transaction history, the blocks in the blockchain are organized so that each new block references its parent (previous) block. Transactions inside blocks are likewise rigorously arranged in an immutable order. Typically, all network participants agree on the precise number and history of blocks and collaborate to combine active transaction requests into the next block. A randomly selected validator on the network creates a block, which is then shared with the rest of the network. All nodes then put this block to the end of their respective blockchains, and a new validator is picked to generate the next block. Ethereum's proof-of-stake protocol defines the exact block construction and consensus processes [18].

Ethereum Virtual Machine

Another essential component of the Ethereum blockchain is the Ethereum Virtual Machine (EVM). Because Ethereum can execute smart contracts, this particular feature sets Ethereum apart from the previously popular blockchain, Bitcoin. The execution and verification of these smart contracts are carried out via the global network of nodes using the engine provided by the EVM. Like a mathematical function, EVM produces a deterministic output for each input. Thus, it is possible to characterize Ethereum as having a state transition function. It has a set of instructions that utilize gas as a unit of computation. Gas is used to keep track of the amount of computing power a validator needs to spend to calculate the new state of

the ledger after executing a transaction and determine the new storage that will be added to the ledger [19].

Smart Contracts

Ethereum's smart contracts are programs executed on the EVM. They are created using high-level programming languages like Solidity or Vyper, compiled into EVM bytecode, and then deployed on the Ethereum network. A smart contract's code is immutable once it is on the blockchain. Every contract has a distinct address. The contract address can be known before deployment. However, the actual computation of the address differs depending on how the contract is deployed. The contracts can include functions that can be called by other contracts or EOAs, as well as reading from and writing to their storage. To execute these smart contracts, the sender must pay the processing cost in units of gas, which is the equivalent of ether. Furthermore, smart contracts have the ability to emit events that are readable from outside the blockchain and documented in transaction logs. Various blockchain clients, such as user interfaces of applications that operate on the blockchain, use the logs to monitor specific events and act according to their business logic. Deploying a smart contract is done by sending a transaction, which appends the bytecode resulting from the compilation process and its initialized storage to the blockchain's ledger. Smart contracts can only access data that is stored on the blockchain. They are unable to access data from outside sources. This was done on purpose because depending on outside data could negatively impact the blockchain network's decentralization and security. Smart contracts are self-contained and execute automatically based on predefined conditions and rules encoded within them. This ensures that the contract is executed exactly as intended, without any interference or manipulation from external sources [20].

2.1.2. Polygon

As mentioned in the previous chapter, Ethereum's block size is limited. The network can produce only one block approximately every 12 seconds. This indicates a constraint on the network throughput. The Ethereum network and other blockchains have frequently been criticized for their limited scalability [21]. The Polygon network is a solution designed to tackle Ethereum's scalability issues. This is effectively done by processing transactions on a different blockchain compatible with Ethereum and then returning them to the main Ethereum blockchain after processing. By reducing the load on Ethereum's network, this procedure enables Polygon to speed up transactions and cut transaction fees. Formerly called the Matic Network, Polygon provides an easy-to-use Ethereum platform for blockchain applications. The goal of Polygon is for users to not worry about network congestion while interacting with decentralized applications [22]. Polygon offers multiple solutions, including Polygon POS, Polygon zkEVM, and Polygon Miden [23]. For the scope of our thesis, we will refer to the Polygon POS, the EVM-compatible, proof-of-stake sidechain for Ethereum.

A blockchain originating from the main blockchain and operating concurrently with it is known as a sidechain. It is typically connected to the main blockchain via a two-way peg. As

a result, digital assets may be interchanged between the main blockchain and its sidechain. Transaction processing takes place on the Polygon sidechain. Unlike the Ethereum mainnet, which might encounter congestion and more expensive transaction costs, the sidechain offers a more affordable and scalable solution. The Polygon Network operates as follows: [22]

1. Initially, a user funds the Polygon contract on the parent chain, which is the Ethereum blockchain, by depositing an asset into it.
2. Once the tokens are deposited and confirmed, they will appear on the Polygon network using the Polygon deposit bridge.
3. After that, the user can transfer the tokens to anyone they want almost instantly, since the Polygon blockchain produces blocks that take approximately 2 seconds, and the fees are significantly cheaper than on Ethereum.
4. Finally, whenever the user wishes to, they can withdraw the tokens back to the main Ethereum chain by establishing proof of the remaining tokens on the Root contract, which is deployed on the Ethereum chain.

The Polygon network employs a dual strategy of proof-of-stake. It combines it for its checkpointing layer with block producers at the block production layer. Using header blocks, or checkpoints, the Ethereum root contract effectively guarantees solvency and finality [22]. Figure 2.1 illustrates how the consensus along both of the layers is achieved.

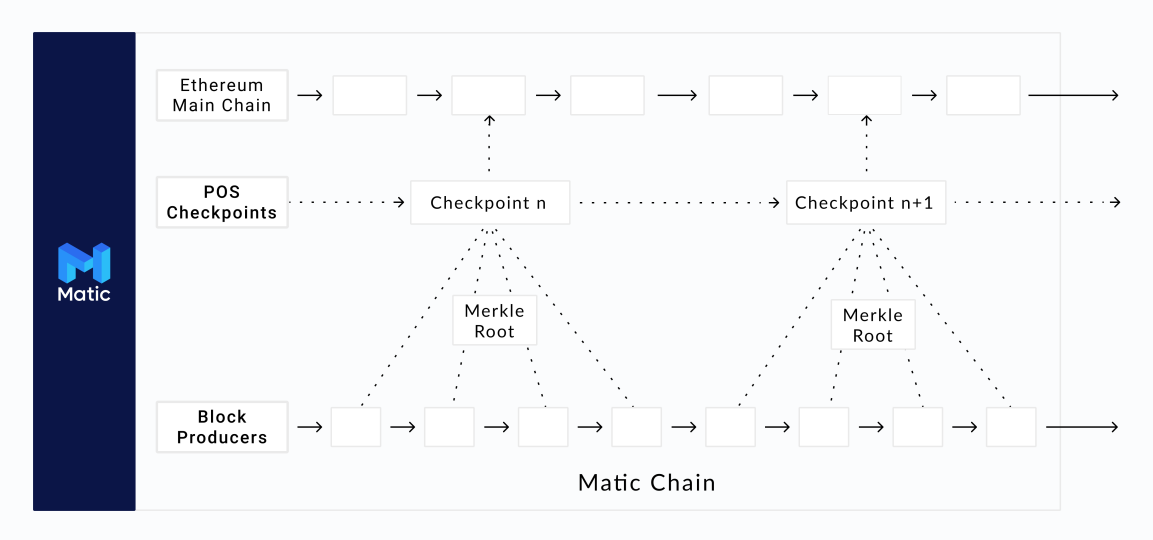


Figure 2.1.: Polygon network consensus (Source: [22])

Checkpointing Mechanism

The checkpointing layer is a series of smart contracts deployed on the Ethereum network. PoS stakers choose block producers from the base layer of the Polygon network's blockchain

to create the Polygon blocks. The Polygon network's checkpointing layer, which supports the network's PoS protocol, selects a proposer from the pool of stakeholders every several blocks at the network's block layer to propose a checkpoint on the Ethereum chain. This proposer is in charge of verifying all recent blocks in the Polygon network's block layer and building a Merkle tree from their hashes since the last checkpoint was set. The resulted Merkle root is then communicated to the stakers' network, allowing them to sign. Other stakeholders analyse this proof and, if it is found to be legitimate, sign off on the proposed proof. The system requires agreement from $\frac{2}{3}$ of stakeholders to submit a checkpoint to the root contract.

Once a checkpoint is submitted on the parent chain, every participant on the Ethereum network has the chance to challenge the header block within a set time frame. If there are no successful challenges before the end of this period, the checkpoint is formally recognised and included as a valid checkpoint on the parent chain. Checkpoints not only secure the parent chain's finality, but they also play an important part in user withdrawals by including the proof-of-burn for token withdrawals. This feature enables users to prove their remaining tokens in the root contract using Patricia Merkle and header block proofs. Withdrawals will incur the same Ethereum gas costs as regular transactions [22]. This method allows the Polygon network to deposit and withdraw assets *from* and *to* the Ethereum network. While the assets are on the Polygon network, the users are able to process and manoeuvre them at a lower cost. Therefore, the Polygon network achieves great transcation speed with a high degree of finality and decentralization.

Cross-chain transactions

After establishing the Polygon network and explaining how it facilitates deposits and withdrawals with its parent chain, Ethereum, we may now define the cross-chain transactions. A transaction that involves the movement of assets or data across two distinct blockchain networks is known as a cross-chain transaction. Each of these blockchains operates independently, regulated by its own protocols, and digital currencies. A cross-chain transaction always consists of more than one operation. It comprises a minimum of two transactions, with at least one conducted on each chain. Such transactions can be useful for moving assets or data across these separate networks for the following reasons:

- Interoperability: This allows for the easy communication and interaction of diverse blockchain networks by facilitating the exchange of assets or data.
- Asset Migration: This is the process of transferring assets from one blockchain to another due to differences in scalability, functionality, or governance.
- Decentralized Finance (DeFi): This enables the execution of transactions across many DeFi protocols or platforms built on various blockchains.

2.2. Financial Systems

In today's interconnected world, financial systems play an important role in facilitating economic activity by providing mechanisms for asset transfer. In this section, we try to pinpoint fundamental concepts of financial systems, emphasising their significance, components, and how assets are exchanged within these networks.

In [24], we understand a financial system is made up of institutions, markets, and intermediaries that allow funds to be transferred from those with surplus savings to those in need, as well as assets to be exchanged. It is a fundamental mechanism for the optimal allocation of economic resources. The key roles of financial systems are:

- Intermediation: Financial institutions act as middlemen, facilitating the movement of cash from savers to borrowers.
- Transaction Facilitation: Financial markets provide venues for the trading of a variety of financial assets, including stocks, bonds, and derivatives.
- Price Determination: Financial markets contribute to the determination of the value of financial assets by integrating information about the assets with wider economic factors.
- Risk Management: Financial systems offer tools and strategies for managing a variety of hazards, including credit, market, and operational risks.

Financial systems are made up of various interconnected entities, each critical for facilitating the exchange of assets [24]:

- Financial Institutions: This category includes banks, insurance firms, investment banks, and other entities that provide financial services, such as facilitating transactions between depositors and borrowers and providing investment products.
- Financial Markets: These are areas where financial assets are purchased and sold. They are divided into two types: primary markets for new securities and secondary markets for existing securities.
- Financial Instruments: These are assets having a monetary value, such as stocks, bonds, and derivatives, each with its own risk and return profile.
- Payment and Settlement Systems: These systems are critical for the execution of financial transactions because they guarantee the secure and timely transfer of money and securities, lowering the risks involved with trading.

Assets can be exchanged inside financial systems in a variety of ways, depending on the transaction and the assets involved. Direct transactions allow buyers and sellers to trade directly, whereas brokered transactions involve brokers who, for a charge, give access to a diverse selection of assets and promote a faster transaction settlement [24]. Electronic

trading platforms have changed the way trading is done by providing rapid, transparent, and efficient means to trade assets, hence increasing market liquidity. In the next two sections we will turn our attention to the last type of asset exchange, diving into how centralized and decentralized exchanges accomplish automatic trades on behalf of the users.

2.2.1. Centralized Finance

Centralised exchanges (CEXs) act as intermediaries in the trade of financial assets. These platforms let buyers and sellers complete transactions in a regulated and typically more user-friendly environment. They are managed by a centralised body and provide benefits like improved liquidity, faster transaction speeds, and customer assistance, but they have also been criticised for security flaws and regulatory compliance. In traditional finance, all firms operate as centralised exchanges, such as banks (Goldman Sachs), stock trading applications (Robinhood), and payment processors (Visa). All centralised exchanges are trustworthy intermediaries [25].

The CEX connects buyers and sellers by gathering their orders into a order book. The exchange serves as a trustworthy middleman between buyers and sellers. Users rely on the exchange throughout the transaction process, trusting that it will not take advantage of their privileged knowledge. The exchange also serves as a custodian for any cash or cryptocurrency held in user accounts, ideally providing a secure location for users to store their funds. Unfortunately, this has not been the case in many previous exchanges. The financial stability of the companies operating the exchanges is not always transparent. [25].

Order books are important components of centralized exchanges, acting as the means by which buy and sell orders are processed. In today's digital finance world, the limit order book (LOB) is a fundamental trading structure. It is where traders, also known as market makers, create trading opportunities by placing limit orders, a price at which they are willing to buy or sell a specific quantity of an asset. These limit orders are then compiled into a list and made available for public inspection. When other traders see a favourable offer on the list, they try to capitalise by placing marketable limit orders or market orders. When a market order comes in, it is combined with the existing limit orders in the book, resulting in a transaction at the agreed-upon price [26].

2.2.2. Decentralized Finance

The recent surge of interest in cryptocurrency and blockchain technology has significantly change market dynamics. Notably, numerous trading platforms that use smart contracts have emerged on several blockchains, such as Ethereum and Polygon, to facilitate transactions in a decentralized manner. These platforms are referred to as DEXs, and rely on a trustless record-keeping system maintained by a vast network of blockchain nodes, making them resistant to cyber attacks and eliminating a single point of failure [26]. Figure 2.2 shows that trading volume on DEXs has been exponentially growing in 2021, reaching a record high of $217B monthly volume in May 2021. Following that time, the growth ceased to be exponential. However, the market has remained stable, having achieved a volume of

$262B in March 2024. This represents 21.75% of the trading share for digital tokens compared to traditional centralized exchanges [27].

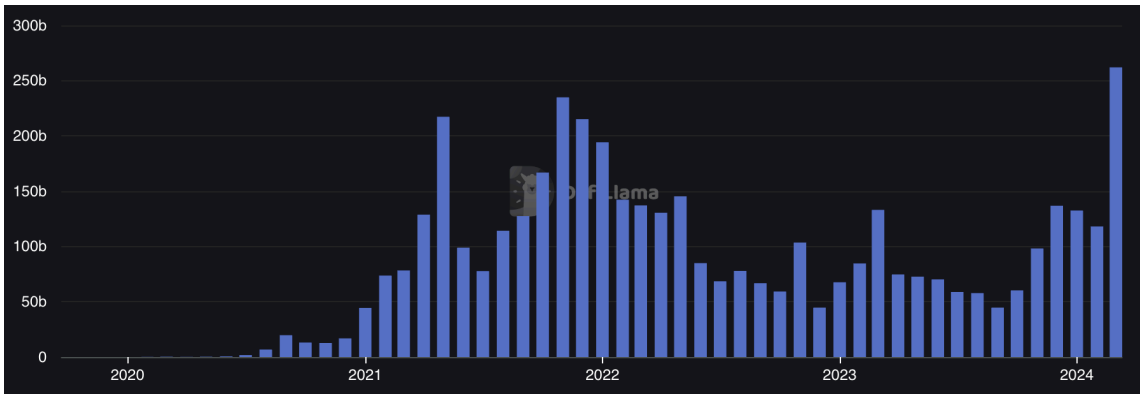


Figure 2.2.: DEXs volume, expressed in USD (Source: [27])

DEXs have introduced and adopted a novel pricing and matching system known as automated market makers (AMMs), which contributes significantly to the success of DEXs. An automated market maker keeps traded assets in liquidity pools and uses a single-function algorithm to set asset prices (or exchange rates) that take into account the pools' condition. Traders can gain access to liquidity by executing trades against these pools, which eliminates the need for active market makers or dealers. Traders obtain liquidity by trading against pools, eliminating the need for active market makers or dealers to execute pricing and orders in person. As a result, an AMM requires much less memory than a traditional order-book algorithm, allowing a large portion of trades to be conducted on the blockchain. Constant Product Market Makers (CPMMs) proposed by Uniswap and adopted by Sushiswap are the dominant market structure. This method can be summarised by the equation x*y=k, where x is the amount of token A in a liquidity pool, y is the amount of token B in a liquidity pool, and k is a constant. As the ratio of token A to token B fluctuates during trading, so does the exchange rate between the two assets [26].

Consider a liquidity pool containing tokens A and B, initially stocked with 10 tokens of type A and 1,000 tokens of type B. The CPMM equation (x*y=k) calculates the initial constant k, 10 * 1,000 = 10,000. This means that the product of the quantities of A and B in the pool must always equal 10,000. Suppose a trader wants to swap one A token for B tokens. As the trader removes one A token from the pool, the new quantity of A becomes 9, while the quantity of B increases to maintain the constant k. The amount of B tokens given to the trader depends on the current ratio of A to B in the pool, which adjusts dynamically with each trade. This phenomenon is known as slippage, where the actual exchange rate experienced by the trader may differ from the displayed rate due to changes in liquidity. Moreover, traders who use AMMs incur trading fees. These fees are a percentage of the transaction value and incentivize liquidity providers. Liquidity providers earn these fees and contribute to the earnings of liquidity providers who contribute assets to the pools.

2.3. Maximal Extractable Value

In previous chapters, we discussed how traditional finance, as represented by centralized finance, is based on trust and requires the participation of multiple parties to ensure a fair and transparent market. In contrast, to all appearances, decentralized exchanges appear well-designed. They provide efficient price discovery and fair trading without the issues common to centralized exchanges. Transactions are carried out using a series of smart contracts, which ensures that they are executed atomically and recorded on the Ethereum blockchain. This fosters a sense of transparency. Furthermore, smart contracts implement the exchange algorithm, manage and secure custody, making it impossible for exchange operators to misuse funds. Despite the benefits they provide, many DEXs have a major problem. Because each block on Ethereum is produced approximately every 12 seconds, trades made using on-chain smart contracts might be slower than those conducted on centralized exchanges. This may cause traders to inadvertently attempt to execute orders that have already been taken or cancelled but still look live because of delayed network updates. Unfortunately, this position may be abused by profit-seeking actors who watch orders and quickly make their own orders with higher costs, allowing them to be included first in the block and benefit at the expense of others. Maximal Extractable Value (MEV) is the maximum value that can be extracted from block production by including, excluding, and changing the order of transactions in a block. The concept was first introduced by Daian et al. in 2019 [2].

An event that signifies a turning point in Ethereum's history is the article [4], which has become famous owing to its depiction of Ethereum as a place where vulnerabilities are exploited without clear explanation or transparency, depicting images of a strange, enchanted forest. In 2020, Dan Robinson and Georgios Konstantopoulos wrote an essay titled "The Dark Forest" about a cooperative attempt to reclaim $12,000 in customer funds stuck in a DEX contract [4]. The article emphasized that the Ethereum mempool, where the nodes store candidate transactions before they are mined, is constantly monitored by individuals seeking chances to extract value. It establishes the foundation for a competitive scenario known as MEV, where rational economic participants are involved.

There are multiple types of MEV, including arbitrage, liquidation, sandwich (front/back run), long-tail, and generalized front running [28].

Arbitrage

Arbitrage possibilities make for the vast majority of all MEV collected. Arbitrage is relevant for DEXs to maintain competitive AMM pricing and align with off-chain oracle prices. A frequent arbitrage opportunity emerges when AMMs with similar token pairings, across several DEXs are misaligned (disparate value), allowing a buy or sell that might equalise prices in both AMMs [28].

Liquidations

Liquidations occur when the collateral backing an on-chain debt position is no longer sufficient. People who want to borrow liquidity from their on-chain holdings can do so by presenting these assets as collateral for a loan. If the loan's value falls below a certain ratio, the debt position is liquidated. Platforms such as MakerDAO hold auctions to liquidate collateral and repay loans, eliminating debt from the system. Liquidators are compensated for their work and receive discounts on the assets they acquire. Because liquidation activities are so profitable, actors seeking to capture this specific type of MEV face aggressive competition. Liquidations provide significant cash opportunities, making them one of the most appealing MEV choices for individuals who can take advantage of them [28].

Sandwiches

Sandwich attacks are generally recognized as a type of MEV abuse that directly affects users. This type of attack occurs when a user attempts to finish a swap transaction that allows for some slippage and sends it to the public mempool, where all actors can see them. Following that, a MEV bot operated by someone trying to profit from the scenario steps in. It inserts its own transaction ahead of the user's, obtaining a better deal for itself and then allows the user's transaction to proceed at the maximum slippage the user is prepared to tolerate. By the time the user executes their deal, the price has shifted against them. To capitalise on the considerably higher asking price, the bot places another transaction after the user's, a practice known as back running. As a result of the slippage, the user pays a greater price for their swap, resulting in a lower asset value than if the front/back running had not happened [28].

Long-tail

Long-tail Minor Extractable Value (LTMEV) are uncommon or less usually types of MEV prospects. These LTMEV activities can result in significant profits, generally by engaging with more atypical protocols, implementing strategies based on specific occurrences, or capitalising on unique characteristics of a system's architecture. For example, a MEV bot might do transactions ahead of a fraud prover by uploading a fraud proof to the blockchain and then earning the reward for uncovering the fraud [28].

Generalized front running

Generalized front-running is often seen as a potentially negative characteristic of MEV. In such instances, a bot developed for generalized front running scans the mempool for lucrative transactions. It then copies them, but replaces the sender address with its own and raises the gas price, to guarantee that it is included in a block before the original transaction, thereby overtaking the initial transaction originator. This method is frequently viewed as a means of abusing MEV. To circumvent this, searchers employ private mempools [28].

2.3.1. Cross-domain Maximal Extractable Value

Cross-domain MEV refers to extracting value by conducting transactions in a predetermined order across distinct domains, such as Layer 1 and Layer 2 blockchain networks, or centralized exchanges. For example, one possibility is to arbitrage from a CEX to a DEX, switching from off-chain to on-chain operations [28]. The analysis performed in this thesis will focus on value extracted between Ethereum and Polygon, using arbitrage techniques.

3. Literature Review

In this chapter, we will present an overview of the available literature related to the subjects covered in the thesis. The first section will look into the literature on detecting MEV extraction in the context of a single blockchain. Following that, we will review the research on cross-domain MEV, with an emphasis on larger settings in which one domain may include a centralized exchange rather than only Layer 1 blockchains. Finally, we will investigate cross-chain MEV extraction, which is the primary topic of this thesis, by reviewing previous research in this field.

3.1. Single-Chain MEV Detection

As previously discussed in our thesis, the phenomenon of MEV has significant ramifications for blockchain networks. Understanding these consequences requires detecting and analyzing transactions that cause such impacts. One of the first research investigates frontrunning on the Ethereum blockchain, identifying three distinct types: displacement, insertion, and suppression [29]. Displacement happens when frontrunners replace other transactions by providing higher gas fees for their own transactions. Insertion is the process of putting transactions in a way that takes advantage of arbitrage opportunities. Suppression, on the other side, seeks to keep specific transactions from being included in blocks. Identifying displacement entails examining transactions with lower gas prices and lower transaction indexes that have the same input as the successful transaction. The insertion detection algorithm uses event emissions from ERC-20 tokens, as well as transaction data (e.g., transaction index, gas price, etc.), to detect the attacks. For suppression, the approach collects transaction data from the same block, where the transaction shares the same receiver and excludes simple transfers. Then, it applies a threshold of the gas consumed, which should surpass 99% across transactions. The algorithm would consider it a valid suppression only if a sequence of blocks matches the above heuristic. While the study's approach makes certain assumptions, such as that attacks are carried only through specially deployed smart contracts, which may miss specific detections, it assures a low percentage of false positives. The investigation finds over 200,000 frontrunning attacks across more than 11 million blocks, with attackers profiting by $18.41 million in total. These findings illustrate the profitable and widespread nature of frontrunning, establishing it as a critical problem in the blockchain ecosystem [29].

While the previous work focused on a Layer 1 solution, Bagourd et al. looked into a detection solution for the upstream blockchain layers. Their research, like ours, makes use of the well-known mev-inspect-py [6], an Ethereum-based MEV inspector tool for detecting MEV activity [30]. For every given block, mev-inspect may detect miner payments (such as gas

and coinbase transfers), token transfers, profits, swaps, and arbitrages. Their article focuses on MEV on Layer 2 scaling solutions, evaluating three prominent blockchains: Polygon, Arbitrum, and Optimism. It provides insights into its implications on users, developers, and the larger ecosystem. The authors show that while Layer 2 scaling solutions are designed to reduce the congestion and high fees encountered on Layer 1 blockchains like Ethereum, they also present difficulties in accommodating MEV since it may impact the fairness and efficiency of the systems. The study evaluates the potential revenues from MEV methods such as transaction reordering, sandwich attacks, and frontrunning using a combination of empirical and theoretical research, as well as an examination of transaction data, block timestamps, and network factors. The findings indicate significant MEV activity on the concerned chains, notably Polygon, with prior estimates of $46 million being dramatically revised to $213 million owing to previously undetected transactions [30]. It's also important to note that the study provides a conservative estimate of the actual MEV extracted while acknowledging that the methodology has limitations and does not account for the full range of MEV opportunities, such as token sniping, cross-chain MEV, or other protocol vulnerabilities.

MEV is shaped by the consensus method employed by blockchain validators to determine transaction inclusion. For example, in Ethereum, validators order transactions depending on the fees given. We previously explored in [29] how attackers are able to displace transactions by increasing the fee, a direct consequence of the chosen consensus. Öz et al. look into a different sort of blockchain, in which transactions are prioritized on a First-Come-First-Served (FCFS) basis [31]. This implies that transactions are processed in the order in which they are received by block proposers. The subject of the research is Algorand, a blockchain that uses a latency-based FCFS technique to organize transactions under typical circumstances. However, during periods of network congestion, a fee-based selection procedure is used. The article performs an empirical investigation to examine patterns of MEV extraction in arbitrage executions, specifically how those seeking MEV benefit from transaction ordering mechanisms and possible latency improvement with block proposers. The primary goal of the research is to identify atomic arbitrage transactions on the Algorand blockchain. It achieves this by generating swap objects from the analyzed transactions. Then, they identify cyclic arbitrage opportunities that meet certain criteria: they must involve at least two swaps, create a cycle with the tokens (beginning and ending with the same token), and result in a profit (input less than output). Although this research does not contain comparison validation findings, the methodology is intended to prevent false positives. The research examined 401,679 blocks and discovered that 2.92% of the transactions were exploited arbitrages, with MEV searchers earning a total of $251,650.15. Interestingly, after evaluating arbitrage attributes, it was discovered that nearly 75% of arbitrages included three or fewer swaps and tokens, with the most complicated cases comprising up to nine swaps and eight tokens. ALGO was the most prevalent of the 26 distinct used tokens found, used in about 97% of cases, followed by the USDC stablecoin and the AlgoFi platform's AF-BANK-ALGO token. Surprisingly, the top two pools used in arbitrages were ALGO/COOP and ALGO/PEPE, yet the corresponding COOP and PEPE tokens were not used as profit tokens [31].

3.2. Cross-Domain MEV

Obadia et al. [32] begin by defining precisely what comprises a domain. They define it as an self-contained system with a globally shared state that may be modified by many parties via what are often referred to as "transactions" within the framework of a shared execution environment. Furthermore, the idea of a domain requires the presence of a sequencer, which is an entity that sets the order of actions inside a domain before they are performed. Each action changes the state of the domain. Domains encompass Layer 1 and Layer 2 networks, side chains, shards, and centralized exchanges. In the previous section, we discussed the influence of the domain where the actors are operating on MEV. This section investigates the consequences of interaction across domains. The phenomenon is widely known as cross-domain MEV. However, others [33] refer to it as non-atomic arbitrage. Although cross-domain MEV might include characteristics such as cross-domain liquidations, these cases are unusual, and the majority of research is focused on identifying arbitrage possibilities.

3.2.1. Formalization

Since we now exit the constraints of a single blockchain, the writers are expanding beyond the typical bounds by adding notions from conventional domains. For example, in [34], the authors highlight the distinctions between informed and uninformed traders in the traditional financial industry. While informed traders may cause losses for market makers (or liquidity providers), these losses are recovered from uninformed traders. Even after fifty years, the informed trader model remains reliable and frequently utilized in theoretical and practical contexts. The article builds upon this idea and introspects on how this concept applies to MEV. McMenamin also discusses the similarities between typical finance options and some forms of MEV [35]. He emphasises that the expected value of extracting profits from transaction or protocol states, which have the same intrinsic worth when they expire, rises as the time to expiry grows.

In the context of multiple domains, Chiplunkar et al. provide a new perspective of MEV, focusing on two key approaches for extracting value from transactions that reach the mempool: $EV_{ordering}$ and EV_{signal} [34]. $EV_{ordering}$ is the strategic placing of a transaction within a bundle of completed transactions to maximize value, also known as atomic arbitrage. This strategy optimizes transaction order by using the blockchain's transparency. Examples include DeFi atomic arbitrages, strategically organizing user transactions to capitalize on slippage disparities, and liquidating unstable loans. $EV_{ordering}$ is entirely dependent on information that is currently available within the blockchain ecosystem. On the other hand, EV_{signal} uses statistical arbitrage techniques similar to those used in traditional finance to extract value by incorporating external information into transaction data. Unlike $EV_{ordering}$, this technique makes use of insights that are not immediately visible from the blockchain state or mempool transactions. This can include arbitrage possibilities between DEXs and centralized platforms, order flow trading by aggregating orders from multiple private or public mempools, or copy trading. Those who apply $EV_{ordering}$ are known as searchers, while individuals utilizing EV_{signal} are referred to as informed searchers. The authors highlight an important point

regarding the difficulty of quantifying EV_{signal}, which arises from the challenge of identifying the particular triggers of the on-chain transactions that are extracting the MEV. Our work dives deeper into this issue, particularly in cross-chain circumstances in which the signal is visible on the other blockchain, but complex methodologies have to be employed in order to pair it up with the extracting transaction. In contrast, our previous chapter demonstrated the ability to reliably estimate $EV_{ordering}$ in cases limited to a single domain. Nevertheless, the authors estimate that in 2022 on Ethereum, \$133M was extracted via $EV_{ordering}$ (excluding sandwiching), whereas the lower bound for EV_{signal} was \$100M [34].

Another interesting topic for investigation is the area of cross-domain MEV, which extends beyond two domains and addresses scenarios involving numerous domains. In [32] the authors of the study investigate a specific situation involving three domains: Ethereum, Binance Smart Chain, and Polygon. Figure 3.1 shows an arbitrage opportunity detected across the three domains, resulting in a gain of \$3018.56. Unlike cases involving only two domains, this circumstance necessitates the transfer or bridging of assets across three domains, which adds complexity to the operation.

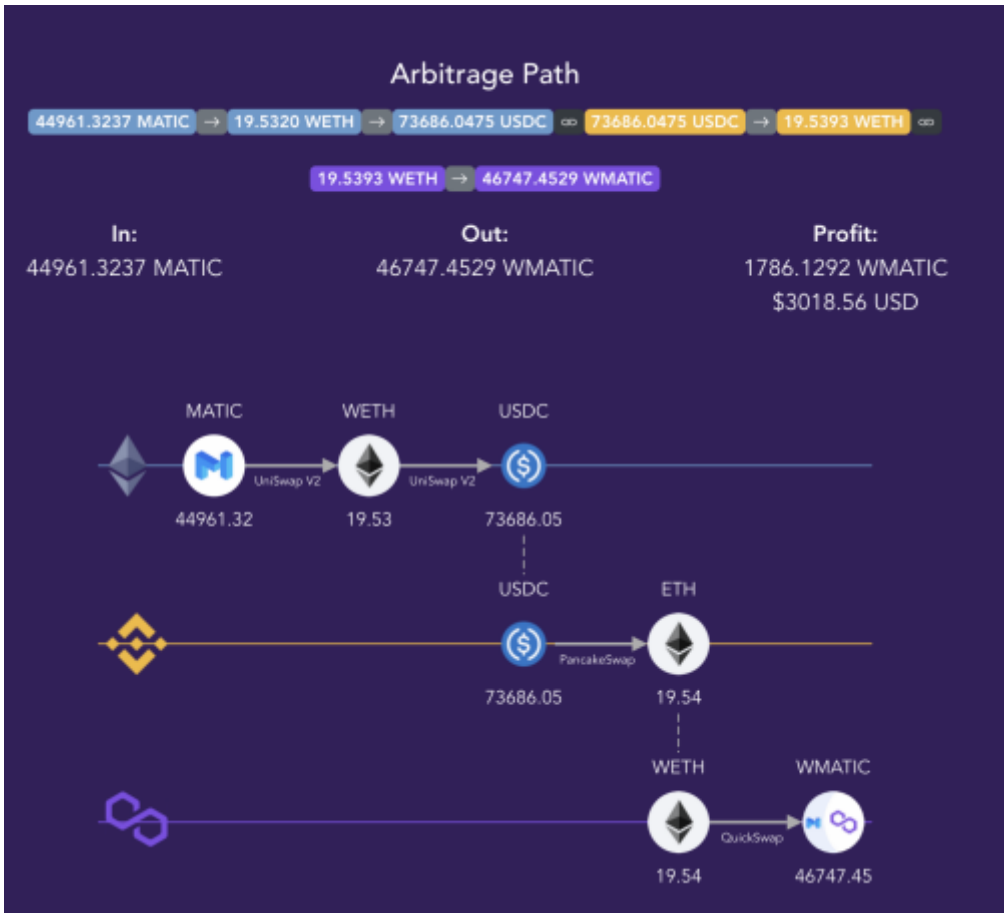


Figure 3.1.: Example of 3-domain arbitrage between Ethereum, Binance Smart Chain, and Polygon (Source: [32])

Obadia et al. also contributed to the topic by investigating the impact of cross-domain MEV on domain sequencers [32]. They begin their investigation by pointing the actual diversity in sequencers across domains. The goal was to see if, in a future where multiple domains coexist, extracting value from different domains would motivate sequencers from the distinct domains to cooperate or collude. This is true especially in scenarios where actions that span multiple domains are aiming to maximize profits. Their findings imply that when AMMs and other protocols with high MEV potential are deployed across domains, the benefits of extracting MEV from those domains frequently outweigh the disadvantages of collusion [32]. As a result, it may be concluded that cross-domain MEV is a centralization force. It it economically rational for a domain sequencer to operate in many domains, which may have an impact on the extent of decentralization within these domains.

In [35], the authors look into the complexity of cross-domain MEV, more precisely at how value is extracted and where it originates from. The emphasis is on the extractor's perspective, which is the actor with the most value to gain, as this is typically the entity which ends up winning the right the extract the value. Regarding MEV extraction, the author's views are quite similar to those given in the previous work [34]. The author's contribution to this viewpoint emphasises that earnings for extractors of $EV_{ordering}$ are almost nonexistent once the extraction rights are paid for. This is due to its low risk, atomicity and predictability, which result in essentially equal profit margins for all extractors. On the other hand, profit from signal extraction is highly subjective and influenced by risk tolerance, access to centralized exchanges private information, signal delay, and other variables. The author proposes a unique model to answer the question of where the value of a transaction originates. According to this model, the expected extractable value of a transaction can be divided into two mutually exclusive components: intrinsic-extractable value and time-extractable value. Intrinsic-extractable value represents the expected value to the extractor at the time when the blockchain state or transaction must be acted on. For instance, in the case of a Uniswap order, this value can be approximated by the maximum expected value of all front and back-running opportunities. On the other hand, for a Uniswap pool, it is the expected extractable value from moving the price up or down when orders are to be included in the blockchain. Intrinsic-extractable value can be realized when the time to act on the blockchain state or order is zero.

Time-extractable value is slightly more complicated. It can be derived in a similar way to an option with respect to a blockchain protocol or transaction state that the extractor can act on. The extractor has the time between confirmation times/blocks to decide whether or not to act on the blockchain state in question. The time-extractable value to the extractor of this optionality is the sum of all paths with a positive extractable value at expiration, times the probability of that path happening. For user transactions, the time-extractable value is also similar. The time value to an extractor for a transaction is the expected value that can be extracted from the transaction between seeing the transaction and including the transaction. This is again analogous to options pricing [35].

Another important point to make when considering non-atomic arbitrage are the risks taken by the extractor. As detailed in [36], inventory risk is one of the most significant risks

taken by the cross-domain extractor. It refers to the possibility of holding inventory for an extended period until it's hedged by the second leg of the trade. This can be a concern, especially for low-liquidity tokens that are known for their volatility. Moreover, CEX liquidity providers may alter their quotes based on DEX trades, complicating arbitrage opportunities. Inclusion risk arises when multiple traders compete for the same chance, which can lead to non-inclusion of on-chain legs. This risk is further compounded by chain reorganizations. Adverse selection happens when CeFi-DeFi arbitrageurs outbid their rivals, resulting in an overestimation of the opportunity size. On the other hand, atomic arbitrageurs are satisfied with riskless profits. The barriers to entry include the challenges of inventory management, particularly with low-liquidity tokens, and the need for inventory rebalancing across venues, resulting in additional operational costs. Fair value assessment requires optimizing latency throughout the trading process, from exchange to block validation. Additionally, CeFi-DeFi arbitrage necessitates substantial capital and low fees, whereas atomic arbitrage relies on efficient smart contracts, with trading capital frequently acquired through flash loans. Validators claim a significant portion of the expected extractable value given the risks and barriers associated with this type of trading. Due to its riskiness and substantial entry hurdles, CeFi-DeFi arbitrage presently sees 35-77% of the anticipated extractable value routed to the validator by successful seekers [36]. Another notable observation in the study is that CeFi-DeFi arbitrage is likely to gain a larger market share. This happens especially when the fair value of an asset changes, but the on-chain prices remain the same (for example, between two consecutive blocks). In such cases, only CeFi-DeFi arbitrage can take advantage of this opportunity. Additionally, if on-chain prices adjust due to a user's trade, the expected value of CeFi-DeFi arbitrage exceeds that of atomic arbitrage due to lower hedging expenses.

3.2.2. CEX-DEX Arbitrages

While the previously analyzed literature in this section explored more theoretical models, we will focus on empirical studies on MEV detection across different domains in the end. Two relevant works in this area are "Non-Atomic Arbitrage in Decentralized Finance" [37] and "searchbuilders.pics" [33]. Both studies use the Ethereum blockchain and employ a similar methodology. They analyze all transactions within blocks and identify simple swaps. Transactions that exhibit traits of MEV extraction, such as bribing the builder through a coinbase transfer or utilizing high gas prices, are categorized as non-atomic arbitrage. Moreover, in [37], the authours utilize more advanced techniques like verifying if transactions were submitted via a private mempool, determining if the swap is the first of its kind in a pool's direction, ensuring consistency in recipient addresses among preceding transactions, and checking if the exchange is done between established tokens traded on CEXs. One important discovery emphasized in the research is that not all non-atomic MEV transactions are related to CEX-DEX arbitrage. Some may involve cross-chain arbitrages. The study proposes that around 90% of these transactions are associated with CEX-DEX arbitrage. To determine the accurate cross-chain arbitrage, the authors needed to filter out non-CEX-DEX arbitrages. This was done by cross-referencing volatility in CEX prices. However, this method may produce numerous false positives. For a precise estimate, identifying cross-chain arbitrages

by matching them with the corresponding transaction on the other blockchain would yield better estimates.

Figure 3.2 emphasises the builder-searcher relationship. It appears that non-atomic MEV extractions by searchers total around 3.5 billion USD. What's especially important is the confirmation of a remark highlighted in Heimbach et al. research on integrated searchers [37]. In that study, the authors reveal the presence of integrated searchers who extract EV_{signal}. Notably, the searcher "Wintermute" has an impressive market share of 53%. Furthermore, the builder "rsync-builder" handles 99% of Wintermute's MEV extraction transactions, hinting to integrated searcher model. Given its substantial market share, this approach appears to be highly profitable. We can now see in this practical example how economic incentives result in collision between parties.

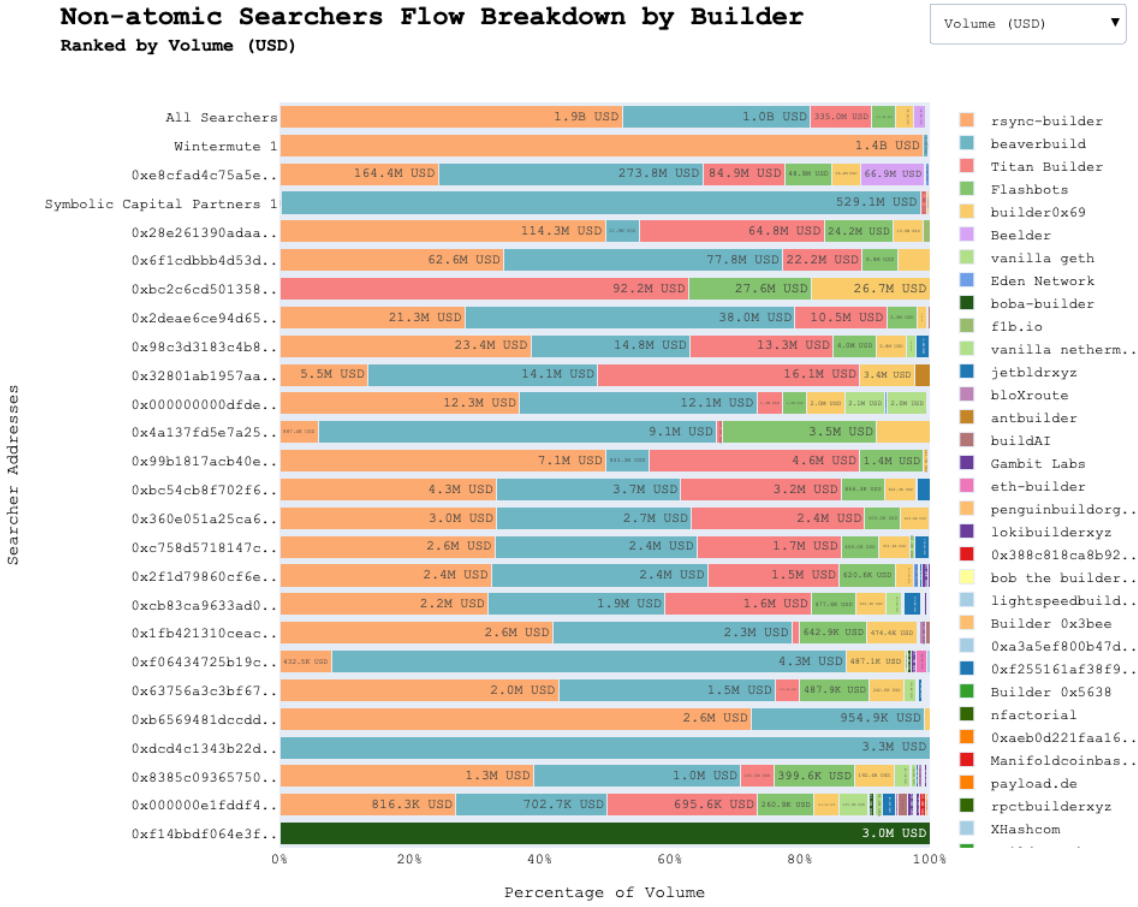


Figure 3.2.: Non-atomic Searcher Flow Breakdown between 2023-11-21 and 2023-12-05 (Source: [33])

Heimbach et al. [37] conducted a much more comprehensive analysis of Ethereum's block data from 15 September 2022 to 31 October 2023. The study revealed that about 25% of the trading volume on Ethereum's top five DEXs comes from non-atomic arbitrage. This highlights the importance of understanding and addressing this phenomenon. The study identified a small group of eleven traders responsible for over 80% of the non-atomic arbitrage volume. This raised questions about the market dynamics due to the concentrated activity. Moreover, the researchers found a correlation between block construction centralization and non-atomic arbitrage. Transactions linked to this activity represented more than 10% of Ethereum's total block value. The research emphasizes the necessity for further investigation into non-atomic arbitrage, its impact on market efficiency, and potential measures to improve security and fairness within decentralized finance.

The author presents two possible solutions for the centralization effect caused by non-atomic arbitrage. The first solution suggests separating top-of-block extractions from other transactions to mitigate the dominance of specialized arbitrage in high volatility periods. This implies separating the PBS auction to allow for specialized handling of top-of-block transactions by integrated searchers. However, concerns arise regarding the potential dominance of top-of-block transactions and security issues, such as time-bandit attacks, which are not directly addressed by this strategy. The second solution focuses on reducing block time intervals as an additional mitigation strategy against non-atomic arbitrage. This approach aims to diminish the profitability of arbitrage opportunities by making the time between blocks shorter. However, there may be challenges associated with altering block times, particularly concerning the consensus layer. To minimize these challenges, the author suggests migrating DEXs volume to Layer 2 solutions like Arbitrum and Optimism, which possess shorter block times and may circumvent some of the issues associated with block time reduction.

3.2.3. Cross-Chain MEV

In the previous chapter, we discussed the general cross-domain MEV. Now, we will focus on a specific subdomain called cross-chain MEV. This type of MEV occurs only across different blockchain domains. It is important to note that cross-chain MEV is different from cross-domain MEV, which does not involve only blockchains. The main difference between the two is that cross-chain MEV occurs only within transparent domains (blockchains), making it theoretically more accurate to detect. For example, If we attempt to identify a cross-domain arbitrage between a DEX and a CEX, we face a lack of transparency in terms of the actors and the environment operating on the CEX, which is not an issue on the DEX. Joules Barragan introduced an additional classification of cross-chain MEV in his article, which distinguishes between two types: cross-chain MEV facilitated through bridging and parallel cross-chain MEV [38]. With the increasing popularity of multi-chain frameworks in the evolving web3 environment, cross-chain MEV opportunities are expected to grow significantly. However, there are inherent challenges with atomicity in transactions as they must adhere to a binary outcome: an event either occurs entirely or not at all. The multi-chain ecosystem presents nuanced risks that are difficult to comprehend, quantify, and mitigate. Traditional cross-

chain transaction methods, such as bridging, have prolonged settlement times, making them vulnerable to opportunistic actors during the latency period. In response, parallel-chain MEV is emerging as a viable alternative. Transactions are executed concurrently across various chains to exploit arbitrage opportunities, resulting in faster settlement times. However, parallel-chain transactions have certain drawbacks, such as increased resource demands across involved chains, which may reinforce centralization tendencies. As a result, entities with greater resource allocations have a competitive advantage, which raises decentralization concerns within this operational framework [38].

We found two research works that focused on cross-chain MEV. Both studies used a similar method but with different scopes. Sjursen et al. focused on extracting swap events from Uniswap pools across four distinct domains, namely Ethereum and three of its Layer 2 solutions (Arbitrum, Optimism, and Polygon) [39]. The study emphasized arbitrage opportunities and collected data between 1 June and 7 July 2022, which totaled 3.7 million swap events. The findings showed that Polygon and Optimism had more swap events than Ethereum, while Arbitrum showed the lowest activity. The authors also noted that Ethereum exhibited more swap activities triggered by routers than the other chains, indicating a higher proportion of swaps conducted by users through web interfaces (previously referred to as uninformed actors) on Ethereum than on the other chains.

To investigate cross-domain MEV extraction, the authors started by merging the unique sender addresses from multiple networks. This led to the creation of groups of sender addresses participating in Uniswap pools across different networks. Later, the authors assessed these addresses individually using blockchain explorers to determine whether they had been involved in cross-domain MEV extraction activities. According to the research, no matching addresses exist for the pairs Ethereum/Optimism, Arbitrum/Polygon, and Optimism/Polygon. However, the researchers found 2 overlapping addresses for Ethereum/Arbitrum, 12 for Ethereum/Polygon, and 4 for Arbitrum/Optimism. The authors further explain that some overlaps likely involve the same searcher and may be linked to the same actor. Nonetheless, this quantification was conducted by manually exploring block explorers, making it challenging to verify the accuracy of the data. Ultimately, through this manual process, the researchers identified an address engaged in cross-chain MEV across multiple networks [39].

In the second part of the study, the researchers attempted to find equivalent token swaps across different chains. They started by filtering out swap events where the sender was a router address, which reduced the number of swap events by almost half. Then, the authors excluded swap events that involved non-round amounts (e.g., 1000, 10000) and added a temporal aspect to ensure that swap events were no more than an hour apart. Unfortunately, this approach did not produce any significant results [38]. However, traders use sophisticated software programs that calculate amounts based on market parameters, making it unlikely for arbitrage amounts to be exact round numbers. Lastly, setting a one-hour time limit may overlook arbitrage transactions facilitated by bridges, which could take longer than an hour.

Similar to the previous research, Mazor et al. [40] have introduced a model for the analysis of cross-chain MEV. They have studied data from DEX pools across two different networks. Specifically, they have analyzed data from PancakeSwap and QuickSwap over one month.

PancakeSwap operates on the BNB Chain, while QuickSwap on Polygon. The researchers have collected data on all the pools created up until 13 January 2023, and for each pool, they have extracted the states between 12 December 2022 and 13 January 2023 [40].

Öz et al. discussed the concept of cyclic arbitrage in a single chain environment, where there are multiple pools of a number of tokens [31]. This study extends this concept within a cross-chain framework [40]. To elaborate, let's consider a collection of m tokens represented as $\{t_i \mid i \in [1, m], m \geq 5\}$, where each pair of consecutive tokens shares a liquidity pool. There are two distinct chains, denoted as C_1 and C_2, and tokens t_1 and t_k (with $k \in [4, m-1]$) are deployed on both chains. This configuration is referred to as an m-cycle cross-chain setup. Below is an example of an m-cycle cross-chain arbitrage involving two chains in the context of Cross-Chain Swap, as illustrated by the research [40].

$$
\begin{gathered}
1 : n_1 \cdot t_1 \xrightarrow{C_1} n_2 \cdot t_2 \\
2 : n_2 \cdot t_2 \xrightarrow{C_1} n_3 \cdot t_3 \\
\ldots \\
k-1 : n_k - 1 \cdot t_k - 1 \xrightarrow{C_1} n_k \cdot t_k \\
k : n_k \cdot t_k \xrightarrow{C_2} n_{k+1} \cdot t_{k+1} \\
\ldots \\
m : n_m \cdot t_m \xrightarrow{C_2} n_{m+1} \cdot t_{m+1}
\end{gathered}
$$

The study denotes cyclic cross-chain arbitrage as cross-chain arbitrage. The profitability of this arbitrage is determined by finding the optimal input amount n_1 that maximizes the difference between the output amount n_{m+1} and n_1. Subsequently, n_1 is used to compute the revenue. To calculate the profit, transaction fees associated with the trade, which is referred to as the "transaction fee," need to be taken into account. Therefore, we can summarize that the formula of profit is $n_{m+1} - n_1 -$ transaction fees. The next focus of the study is to explore cross-chain arbitrage using a specific algorithm. To accomplish this, the algorithm uses the gathered liquidity data from both chains to create a graph that is a representation of their states. Afterward, it examines token pairs and the possible extraction paths between the two DEXs to identify potential arbitrage opportunities. These opportunities arise when there is a difference in token prices across them [40].

The analysis of the algorithm output explores the revenue dynamics and discrepancies observed in cross-chain arbitrage strategies implemented across PancakeSwap and QuickSwap. Tokens listed on PancakeSwap are denoted as $TOKENSYMBOL_P$, and on QuickSwap as $TOKENSYMBOL_Q$ Initially, the revenue ranged modestly between 50 to 80 USDC, which was mainly influenced by existing price discrepancies among token pairs. However, on 17 December, a notable turning point occurred, marked by a substantial surge in revenue exceeding 1,000 USDC and peaking at an impressive 10,000 USDC. This surge was mainly due to highly profitable arbitrage tactics leveraging tokens listed on pools such as $USDCS_P$

and $WBNB_Q$. For instance, one specific arbitrage pathway, $USDC_P \rightarrow BUSD_P \rightarrow WBNB_P \rightarrow WBNB_Q \rightarrow USDCS_Q$, yielded a remarkable revenue of 10,877 USDC, showcasing the potential for substantial gains.

Furthermore, subsequent spikes in revenue on Dec 19, 22, and 26 were linked to specific token discrepancies involving $GOTCHI_P$, ATP_P, and JST_P, respectively, each contributing to notable revenue peaks. The study also contrasts the revenue performance of cross-chain arbitrage using different DEXs. Revenue in PancakeSwap often exceeded QuickSwap due to arbitrage length limitations and resource constraints. This disparity underscores the strategic advantage of cross-chain arbitrage in optimizing profitability by capitalizing on market inefficiencies and price differentials across multiple DEXs [40].

The findings highlight the potential for cross-chain arbitrage strategies to enhance overall trading profitability and maximize returns, emphasizing the importance of leveraging varied DEX platforms to exploit lucrative opportunities in the decentralized finance landscape.

4. Blockchain Interoperability

So far, we have looked at various forms of MEV across different domains, including cross-chain MEV. Within cross-chain MEV, there are two types: parallel cross-chain MEV and cross-chain MEV facilitated through blockchain bridge infrastructure [38]. Since we are focused on the latter, we will now explore blockchain bridges. The purpose of this chapter is to explain the types of bridges that exist and how bridge technology fits within the interoperability landscape. Despite the complexity of this subject, which has seen significant growth in recent years, we aim to organize this knowledge in a systematic way, focusing on the essential aspects relevant to MEV extraction.

The blockchain trilemma, first introduced by one of Ethereum's founders, highlights a trade-off between security, scalability, and decentralization that blockchains face. Since without security the blockchain is compromised, than this priority is always prioritized. To ensure security, consensus algorithms, crypto-economics, formal modeling, and distributed systems research are used. As the number of nodes in a peer-to-peer network increases, the network becomes more resilient to attacks, but it also slows down consensus due to increased message exchanges and communication latency. Thus, decentralization and security are closely linked. However, the main challenge is to address scalability within this trilemma. The solution to scalability can be found in the research field of interoperability, which will become clearer as we delve deeper into this topic [41].

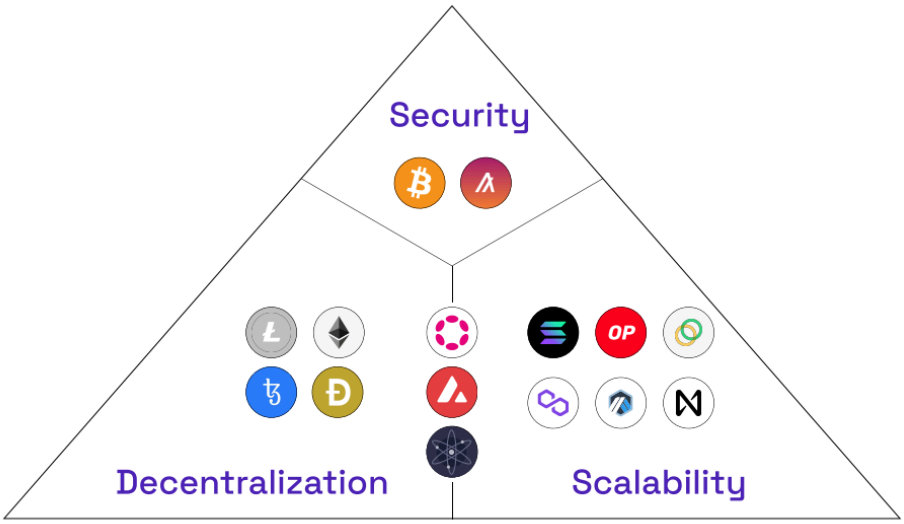


Figure 4.1.: Classification of blockchains according to the blockchain trilemma. (Source: [41])

Figure 4.1 depicts the classification of blockchain solutions based on the blockchain trilemma, as proposed by Belchior et al. [41]. They have divided these solutions into three categories. The blockchain ecosystems are arranged from left to right and top to bottom and include Bitcoin, Algorand, Litecoin, Ethereum, Polkadot, Solana, Optimism, Celo, Tezos, Dogecoin, Avalanche, Polygon, Arbitrum, NEAR, and Cosmos.

As the authors explain in [41], blockchain interoperability may be classified as either multi-chain or cross-chain. Multi-chain interoperability is the interaction of instances of a blockchain engine, commonly known as a "blockchain of blockchains." These instances interact using a trust anchor built into the protocol. Each instance, known as a mini-blockchain, has an interoperability protocol and data format that allows it to communicate with other mini-blockchains inside the same engine. Such engines include Cosmos, Polkadot, and Avalanche. For example, in Polkadot, parachains (mini-blockchains) connect using XCMP (Cross-Chain Message Passing), which establishes confidence via the relay chain. Similarly, in Cosmos, mini-blockchains known as zones connect using the Inter Blockchain Communication (IBC) protocol, which is supported by a light-client interoperability mechanism for validating cryptographic proofs. On the other hand, blockchain systems that prioritize scalability often use sharding, where each shard (a subset of transactions) is processed by a separate mini-blockchain and then aggregated across shards. However, achieving interoperability between different blockchain engines presents a challenge. For example, while Polkadot's parachains can communicate internally, they face hurdles in communicating with external blockchain engines like Cosmos due to differing protocols and global states, making them heterogeneous systems. This heterogeneity highlights the intrinsic boundaries of blockchain networks; without a unified cross-chain protocol, these systems remain distinct and heterogeneous. Cross-chain interoperability aims to bridge these boundaries by facilitating communication between diverse and heterogeneous blockchain chains. Conversely, multi-chain interoperability focuses on linking chains that share the same framework and are typically anchored within a common chain, thus forming a more homogeneous network. Therefore, the distinction lies in whether the interoperability targets chains within a similar framework (multi-chain) or extends to chains that are inherently diverse (cross-chain), thereby defining the scope and challenges of blockchain interoperability in heterogeneous environments [41].

Cross-chain communication is thoroughly examined by Zamyatin et al., defining it as the CCC (Correct Cross-Chain Communication) problem and outlining the phases of a generic CCC protocol [42]. The discussion begins by stressing the importance of communication among distributed processes, particularly in database systems, to ensure the atomicity of distributed transactions through solving the Atomic Commit problem (AC). Reference is made to the Non-Blocking Atomic Commit (NB-AC) problem, which tackles scenarios where processes must reach definitive outcomes despite potential failures.

A framework is introduced, marking two independent distributed systems (X and Y) with their respective ledgers (L_x and L_y) and operating under a closed system model. Through scenarios involving processes P and Q on systems X and Y, respectively, the impact of transactions on the state evolution of the systems is detailed. This highlights the importance

of aligning transaction descriptions (dP and dQ) with transaction validity across the two distributed ledgers.

The goal of cross-chain communication is to synchronize processes P and Q, ensuring that Q only commits transaction t_y to ledger L_y if P has already committed t_x to ledger L_x. This synchronization relies on the matching the true descriptions of t_x (desc(t_x)) with the ones held by Q (d_Q) and P (d_P). The concept behind this is that t_x and t_y are interdependent, meaning they must be either included or excluded from L_x and L_y simultaneously, as seen in an atomic asset exchange. To achieve this synchronization, P must prove to Q that it has initiated transaction t_x, which is already committed to the ledger L_x. Specifically, at a given time "t", Q must verify that t_x is included in the ledger state L_x at time t. An effective cross-chain communication protocol must demonstrate the following essential properties:

- Effectiveness. This property states that if both processes P and Q execute correctly and their respective transactions match the anticipated descriptions and are deemed valid, then t_x will be added to ledger L_x, and t_y will be added to ledger L_y. However, if the descriptions do not match as expected, or there is an error in either process's behavior, both transactions t_x and t_y will be excluded from their respective ledgers (L_x and L_y).
- Atomicity. There are no scenarios where P writes t_x to L_x without Q having written t_y to L_t.
- Timeliness. Eventually, a process that behaves correctly will write a valid transaction to its ledger.

Overall, these definitions establish the criteria for evaluating the correctness and robustness of cross-chain communication protocols, emphasizing both safety and performance considerations [42].

Vitalik Buterin has outlined three main technical approaches for blockchain interoperability [43]. The first approach is the centralized or multisig notary schemes. In this approach, a single party or a group of parties agree to execute an action on blockchain B after an event on blockchain A is triggered. The second approach is the sidechains/relays, which are systems within a blockchain that can verify and interpret events or states from other blockchains. Lastly, there is hash-locking, which is a method where operations are set up on both chain A and chain B. These operations activate only upon the revelation of a specific hash preimage.

Cross-chain operations have become easier to carry out thanks to notary mechanisms, which provide a simple technological solution. These mechanisms allow a trusted entity or group of entities to validate events taking place on one blockchain (chain A) and confirm specific claims related to another blockchain (chain B). The entities involved can either operate proactively by monitoring events and taking action automatically based on predefined criteria within a blockchain, or reactively, by responding with signed messages upon request. The approach provide a simple yet effective solution within its designated trust framework, which assumes that a specified portion of the selected notaries will not exhibit Byzantine behavior. This model allows for a flexible determination of notaries that can be customized for each cross-chain operation scenario. It envisions a negotiation process where

participants contribute their lists of trusted entities, and the final notary set is established as the intersection of these lists. Moreover, the concept of notaries can be integrated with other methodologies, potentially forming an inter-chain exchange protocol that strives for optimal security. In cases where the underlying blockchain lacks support for more decentralized relay mechanisms, the system gracefully falls back to notary-based schemes to ensure transaction security. This adaptable combination underscores the protocol's ability to adjust to varying levels of technological infrastructure and trust requirements [43].

We have analyzed various interoperability protocols that rely on notary schemes, also known as blockchain bridges. These protocols usually serve two primary purposes: transferring tokens across different blockchains or enabling the exchange of any message. Based on this analysis, we developed two blockchain bridge architectures that operate in this way.

Figure 4.2 shows the process of transferring tokens between blockchains using a blockchain bridge. The bridge's architecture includes on-chain actors, which are smart contracts deployed on both blockchains, and off-chain actors, namely the bridge validator and the bridge executor. To initiate a cross-chain operation, the user locks a specified amount of tokens in the bridge contract. Additionally, the user typically compensates the bridge validator for executing the cross-chain transfer. However, some bridges incentivize usage by charging zero fees. Subsequently, the bridge validator monitors and verifies the operation, ensuring its validity and finality on the source blockchain. Upon successful completion of the token lock operation, the validator instructs the bridge executor to mint an equivalent amount of tokens on the destination blockchain. Blockchain bridges, which are able to exchange arbitrary messages between blockchains, operate similarly (Figure 4.3). The difference is that now, the off-chain actors will need to send a custom message instead of a pre-defined one. This can be more complex because the delivered message can trigger a function costing an arbitrary amount of gas.

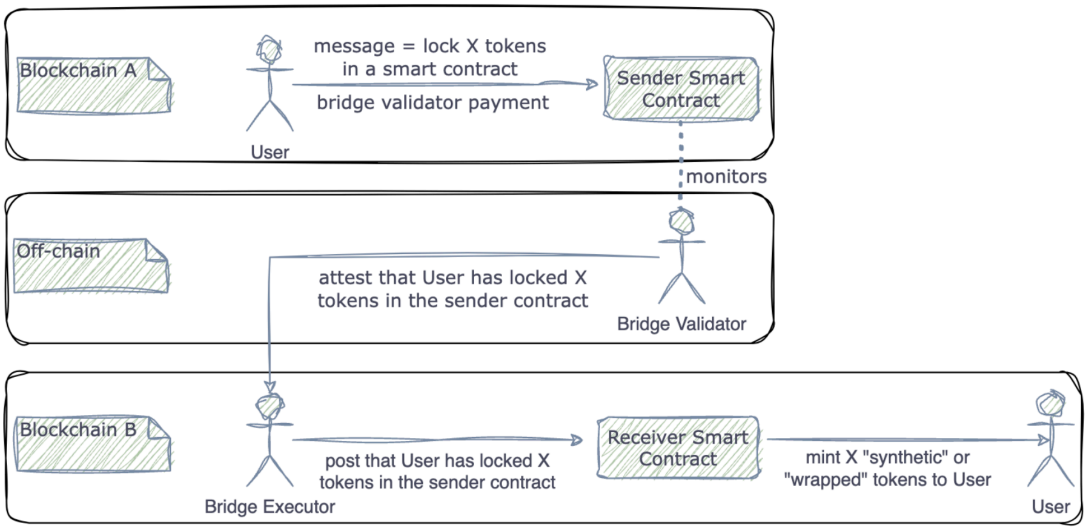


Figure 4.2.: Sending tokens between two blockchains using a bridge

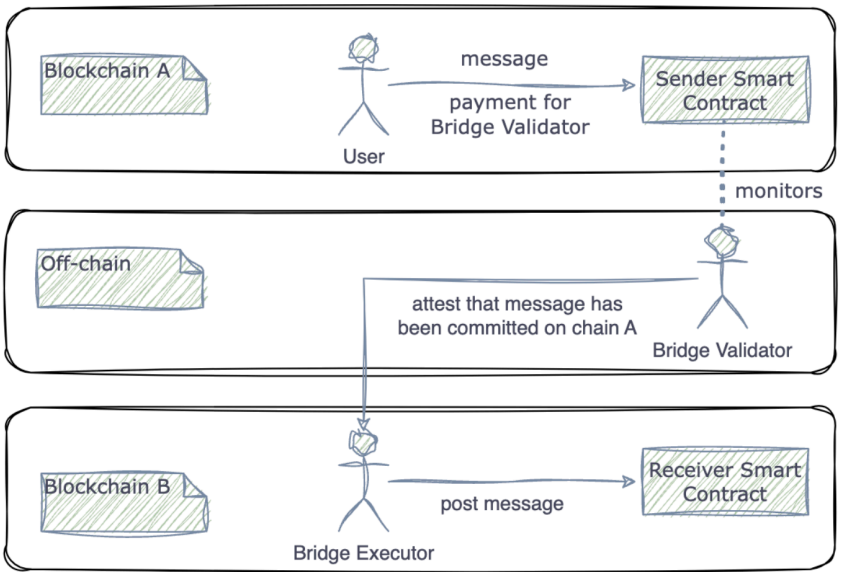


Figure 4.3.: Sending a message between two blockchains using a bridge

The integration of relay technology in blockchain systems is a significant advancement towards achieving interoperability between different blockchain networks, without the need for trusted third parties. Rather than relying on external entities to relay information between chains, relays enable chains to communicate and validate data across different networks independently. In the context of blockchain networks, relays function as a direct mechanism for enabling interoperability by allowing one chain (chain B) to verify specific events or state information from another chain (chain A). This process involves retrieving compact representations of blocks, called block headers, from chain A, which encapsulate cryptographic proofs of the block's validity and state. The block headers are then verified through standard consensus algorithms native to chain A, such as proof of work or Byzantine fault-tolerant consensus, ensuring the integrity and finality of the information. The concept of "light client verification" is crucial to the feasibility and efficiency of relay systems within resource-constrained blockchain environments. This method allows relays to validate transactions or state changes on a different blockchain without requiring full validation capabilities, which would be impractical due to computational limitations. Instead, relays validate specific segments of the blockchain's Merkle tree against the validated block headers, ensuring the authenticity of targeted transactions or data entries. Furthermore, the utilization of relay technology mitigates the inherent limitations of blockchain self-containment by facilitating on-demand data retrieval and validation across interconnected chains. This decentralized approach to blockchain interoperability promotes transparency and trust through cryptographic verification, allowing smart contracts within relays to autonomously verify and execute operations based on validated data from external chains. Relay systems represent a promising avenue for enhancing blockchain interoperability by enabling direct and verifiable communication between different networks [43].

The complex cryptographic procedures involved in relay operations can be abstracted and hidden from developers. Event verification can be enclosed within a smart contract acting as an event verification oracle, which can be invoked by other contracts. The act of reading events can be simplified into an asynchronous operation. A cross-chain smart contract programming language can incorporate a function such as *createEvent(destinationChain, params)*, which registers an event and assigns it a unique identifier. Additionally, a function *onReceiveEvent(senderChain, params)* could be designed to execute only upon successful validation of a cryptographic proof associated with the event. Once validated, this function would store a record preventing the same event from triggering it again. One of the first working relay solution was BTCRelay. It has been deployed on Ethereum blockchain. It enables interoperability between Bitcoin and Ethereum by allowing Ethereum to read the Bitcoin blockchain. However, it is important to note that this interoperability is unidirectional; Bitcoin cannot read the Ethereum chain due to limitations in its scripting language [43]. However, due to operation costs this has been shut down. This proved that decentralization has a price that the interoperability market decided it's not worth paying.

Hash-locking is a technique used to facilitate cross-chain atomic operations. When it comes to cross-chain digital asset exchange, hash-locking works in the following way:

1. Party A generates a random secret (s) and creates its hash (h = hash(s)). Then, they transmit h to party B. Both parties then commit their assets to a smart contract under specific conditions: A locks their asset first, and B locks theirs upon verifying that A's asset has been successfully secured.

2. The contractual rules stipulate that if A reveals the secret s within 2X timeframe, A's asset is transferred to B; otherwise, it reverts back to A. Similarly, on B's side, if the correct secret (i.e., the value producing hash h) is disclosed within X seconds, B's asset is transferred to A; otherwise, it reverts back to B.

3. Finally, A discloses the secret within X seconds, allowing A to claim B's asset from the contract. Simultaneously, B learns the secret during this disclosure, enabling B to claim A's asset from the contract. This synchronized process ensures the secure and simultaneous exchange of assets across blockchains using the principles of hash-locking.

This process is provably atomic. If party A reveals the secret s within X seconds, it provides a minimum of X seconds window for party B to claim their asset. If party A reveals the secret s too late, they risk losing the opportunity to recover their own asset. However, such missteps are easily avoidable and solely the responsibility of party A. If party A reveals s between X seconds and 2 times X seconds, they forfeit their asset while allowing party B to claim theirs. This consequence is attributable to party A's actions. If party A reveals s after 2 times X seconds or not at all, both parties regain possession of their respective assets. It is worth noting that if party A fails to lock their asset, party B refrains from locking theirs as well. Similarly, if party B fails to lock their asset or misjudges the deadline for s, party A can choose to withhold s indefinitely, thus reclaiming their asset [43].

4.1. Taxonomy of Blockchain Bridges

In the previous section, we discussed the topic of blockchain interoperability and attempted to understand its details. While some authors use the term "blockchain bridge" interchangeably with interoperability solutions, others restrict it to specific limitations. In this section, we will use "blockchain bridges" to refer to any solution that achieves interoperability between two systems outside of a "blockchain of blockchains" framework.

Currently, notary solutions are the most prevalent in the field of interoperability. This is because they are easy to implement and market. Moreover, the security and decentralization aspects of bridges can be challenging to understand and explain, allowing bridge developers to promote abstract solutions positively while ignoring crucial details. In most cases, the validators of the bridge serve as the primary assurance of a trustworthy solution. However, it is challenging to determine precisely how effective these deterrents are against misconduct for each validator. Furthermore, the avenues for recourse are slow, complicated, costly, and legally uncertain.

In this section, we will introduce the bridge taxonomy that we have developed and explain it in detail. Bridge protocols are complex systems that operate across different networks and environments. They consist of numerous components and require careful management of multiple security considerations, assumptions, and trade-offs. Additionally, they involve coordinating the actions of various actors who have different motivations and trust models. Therefore, evaluating bridge protocols necessitates a comprehensive analysis of a diverse range of factors. Therefore, our focus is on bridges that have gained popularity and display distinctive features which relate to cross-chain MEV. The focus lies primarily on the functionality and features of these bridges.

Lock and mint token bridges

Lock and mint token bridges are systems that allow for the transfer of blockchain native tokens from one blockchain network to another. Let's consider two blockchains: blockchain A will be referred to as the source blockchain, and blockchain B will be known as the destination blockchain. The process involves first locking the tokens on blockchain A in an escrow smart contract. To initiate the transfer, tokens are sent to a specific smart contract or address on blockchain A. Once the tokens are confirmed, the same token value is made available on blockchain B in the form of a synthetic asset, often called a wrapped token. This process allows for blockchain native assets to be transferred between different blockchains. However, it's worth noting that each bridge must create its own wrapper token. This often results in multiple representations of the same token on different blockchains, leading to user confusion and liquidity fragmentation. Usually, assets created through the native bridges of a blockchain (e.g. Polygon bridge) are regarded as the preferred canonical assets on a chain based on social consensus.

Type			Protocol	Validation
Arbitrary Messaging Bridges			LayerZero	External
			Wormhole	External
			Axelar	External
Token Bridges	Liquidity networks	Pool Based	Across	Optimistic
			Hop	Optimistic
			Connext	Optimistic
			cBridge	External
			Stargate	External
		Order Flow	deBridge	External
			UniswapX	External
	Burn and mint	Stablecoins	Circle CCTP	External
			Maker Teleport	External
		Wrappers	Connext xERC20	Optimistic
			LayerZero OFT	External
			Polygon Bridge	Native
	Lock and mint		wBTC	External

Table 4.1.: Taxonomy of blockchain bridges

Burn and mint token bridges

Burn and mint token bridges are another type of system used for transferring blockchain tokens across different networks. The process starts by securely and irreversibly destroying or "burning" tokens on Blockchain A. This is done by sending them to a designated burn address or smart contract that is designed to remove them from circulation on Blockchain A. Once the tokens are successfully burned and verified, an equivalent amount of tokens is generated on Blockchain B through the minting process. This process creates new tokens on Blockchain B that are equivalent in value to the tokens burned on Blockchain A. The newly minted tokens can then be used within Blockchain B's ecosystem.

There are two types of blockchain bridges that operate in this manner: stablecoin bridges and wrapper bridges. Stablecoin bridges have permission to directly burn tokens on the source contract, reducing its supply. On the other hand, wrapper bridges use a different version of the lock and mint bridges. They wrap the asset on the source chain to a different token, which the bridge contract is authorized to burn, allowing the burn and mint process to be executed.

Liquidity networks token bridges

A liquidity network bridge allows users to transfer and exchange assets between different blockchain networks with ease. This bridge simplifies the process by enabling users to perform bridging and swapping operations by initiating a single transaction on the source blockchain. Without such a bridge, users would have to initiate a transfer from the source

chain, wait for tokens to arrive on the destination chain, and then execute a separate exchange transaction to convert the tokens to their desired asset. Using a liquidity bridge, users can bridge and swap tokens in a single transaction, making the process more efficient and potentially reducing gas costs. However, this method comes with the risk of MEV generation, where bad actors could exploit users' intentions to bridge and swap tokens. Bridge validators, who are aware of these operations, are in the best position to do so. They have the responsability to perform the bridge and the swap in the name of the user, therefore, they have an opportunity to profit from MEV. There are two primary types of liquidity bridges:

1. Pool-based bridges: Liquidity providers send funds into pools across different chains, which are then used to facilitate token exchanges. Many existing liquidity networks operate on this model.

2. Order flow auctions: These protocols prioritize user experience and aim to achieve the best possible prices for users. They accomplish this by matching users' orders with the most competitive quotes from Market Makers, often determined through an auction mechanism

Arbitrary messaging bridges

Cross-chain communication is not limited to token transfers. Arbitrary Messaging Bridges aim to widen the range of communication between blockchains by enabling users to send any data from one blockchain to another. This type of bridges can facilitate cross-chain governance, token launches, contract calls, gaming experiences, and other functions. They serve as building blocks for decentralized applications that operate across blockchains, separating concerns for systems. For instance, decentralized application developers need not build their own bridge; they can use an Arbitrary Messaging Bridge by implementing specific smart contract interfaces provided by the bridge. However, the decentralized application's security and scalability rely on the bridge.

4.2. Case Study: Polygon Bridge

The Polygon bridge is the mechanism by which the Polygon blockchain reads data from the Ethereum blockchain. Additionally, the Polygon bridge allows data to be sent from the Polygon blockchain to the Ethereum blockchain. This mechanism is native from the perspective of Polygon, meaning that it relies on Polygon validators to contribute to the transfer of data between the two blockchains. There is no third-party entity involved in notarizing the cross-chain transfer. However, from the perspective of Ethereum, the Polygon bridge is an externally verified blockchain because the Polygon validators are third-party entities to the Ethereum validators. In this section, we will explain how the Polygon network reads state data from the Ethereum blockchain and vice versa.

When an actor begins a cross-chain transaction, the first step is to send tokens to a bridge contract on the source chain via a transaction. Once the source chain bridge contract veri-

fies receipt of the user's tokens, the data is transferred to the target chain bridge contract. After receiving the data from the source chain, the bridge contract on the target chain will perform a second transaction to transfer the assets to the address on the target chain [44]. This process involves deploying a sender and a receiver smart contract, mapping them, and exchanging data. Transferring data from Polygon to Ethereum differs from the opposite direction. Therefore, we will tackle the two cases separately.

From Ethereum to Polygon

When a user initiates a cross-chain transfer from Ethereum to Polygon, the transaction interacts with a sequence of smart contracts. The first contract called is *RootChainManagerProxy*[1], which forwards the call to the *RootChainManager*[2] contract. This contract then initiates a series of contract calls which ultimately lock the user's assets. The process is completed by emitting events such as *LockedEther()*, *LockerERC20()*, and *LockedMintableERC20()* depending on the type of assets being transferred.

The final step on Ethereum involves calling a contract named *State Syncer* to transmit state synchronization data to Polygon. This data is used to read Ethereum data on the Polygon EVM chain, which is known as *State Sync*. This facilitates the transfer of arbitrary data from the Ethereum chain to the Polygon chain. This process is enabled by Polygon validators who monitor for the specific event StateSynced from a Sender contract. Once this event is detected, the data contained in the event is recorded on the Receiver contract.

On the Polygon side, once the state synchronization data is updated, Polygon's null contract triggers the respective token contract to mint the desired amount of assets. Finally, the newly minted assets are sent to the user's address on Polygon.

The *State Sync* mechanism is an essential process that enables users and decentralized applications (dApps) on the Polygon Proof-of-Stake (PoS) chain to access the latest Ethereum blockchain data. This mechanism involves a collaborative effort between validators operating across different layers of the Polygon network, specifically the Heimdall and Bor layers [45]. The Heimdall oversees validators, the selection of block producers, spans, the *State Sync* mechanism between Ethereum and Matic, and other important properties of the system. The BoR validators' consensus model involves a set of block producers who engage in a voting process to designate new producers, taking turns to generate blocks. Applications which make use of the Polygon *State Sync* mechanism, employ the following workflow [45]:

1. Initiating the *State Sync*: The process begins when a specific function in the *StateSender* smart contract is called. This function triggers an event called *StateSynced*, characterized by its unique ID, the contract address, and the accompanying data in bytes.

2. Broadcasting the Event: The *StateSynced* event is then broadcast across all validators on the Heimdall chain. A validator, motivated by the potential to claim transaction fees, may carry out the transaction that pushes this event to Heimdall.

[1] 0xA0c68C638235ee32657e8f720a23ceC1bFc77C77
[2] 0x37D26DC2890b35924b40574BAc10552794771997

3. Heimdall Block Inclusion: The *State Sync* transaction, once validated, is included in a Heimdall block. This action places the transaction in a queue of pending *State Sync* events.

4. Data Retrieval by Bor: In the next phase, Bor nodes, during their operational cycle, fetch the list of pending *State Sync* events from Heimdall. This retrieval is facilitated through an API call.

5. Execution of custom logic in the receiver contract: The last step involves the receiver contract, which conforms to the *IStateReceiver* interface, employing its custom logic to interpret the data bytes from the state-sync events. The *onStateReceive* function within this contract is crucial for decoding the data bytes and executing corresponding actions based on the decoded information.

The shows the relationship between the Heimdall and Bor layers of the Polygon network, and they facilitate cross-chain transactions. It also explains the crucial role played by smart contracts in facilitating the State Sync process [45]. This sequence guarantees that users and the applications on the Polygon PoS chain can quickly access and use the most recent Ethereum blockchain data.

From Polygon to Ethereum

The process of transferring data from Polygon to Ethereum is different from transferring data from Ethereum to Polygon. To achieve this, validators of the Polygon blockchain create checkpoint transactions on the Ethereum blockchain. A transaction is initially created on Polygon, and it must emit an event and include the data we wish to transfer from Polygon to Ethereum. Within 10-30 minutes, the checkpoint transaction is submitted on the Ethereum chain by the validators. Once this is done, the hash of the transaction created on the Polygon chain can be submitted as proof on the *RootChainManager* contract on the Ethereum chain. This contract then verifies the transaction, ensures it is included in the checkpoint, and decodes the event logs from the transaction. Using the decoded event log data, the root contract deployed on the Ethereum chain can digest the message sent from the Polygon blockchain. This architecture guarantees that state changes on Ethereum only happen when the transaction on Polygon is validated and verified on the Ethereum chain by the *RootChainManager* contract [45].

5. Methodology

This chapter will introduce our approach to identify cross-chain MEV extraction using bridges between blockchains. In the first part, we will present a general methodology capable of detecting the extraction of MEV between Ethereum and Polygon as long as the bridge does not keep essential information private. In the second part, we will provide an example of how our methodology works by showcasing our implementation by choosing the Polygon bridge. Our methodology is centered on detecting cross-chain arbitrage. Our goal is to provide insights into the strategies and profits of the parties involved. To achieve this, we utilize definitions of cyclic arbitrage and, more specifically, cross-chain cyclic arbitrage that have been previously defined [31], [40].

5.1. Cross-chain MEV Detection

This section will discuss our methodology for identifying cross-chain arbitrage opportunities between Ethereum and Polygon using blockchain bridges. We have developed a set of generally applicable algorithms that use certain functions specific to particular bridges. One of the objectives of this methodology is to enable other researchers to use it for different bridges, or even blockchains, by implementing the relevant specific functionalities. We will walk through the illustrated algorithms to explain our heuristics. Furthermore, to complement the explanation of the algorithms, Figure 5.1 includes a depiction of the algorithms illustrated through diagrams.

The first algorithm describes the matching process for cross-chain MEV extractionz. The detection algorithm requires the input of Ethereum blocks where MEV actors may have included transactions that extract MEV across two blockchains. We create a list of empty instances of identified cross-chain MEV extraction instances. The algorithm iterates through every transaction of every block and analyzes if the transaction exhibits non-atomic arbitrage behavior, similar to previous works in detecting non-atomic arbitrage [33], [37]. Since this is a superset of cross-chain arbitrage, this means that, at this point, the transaction is a candidate for our detection. We also determine if the transaction interacts with the relevant bridge. If both of these properties apply, this transaction might be a cross-chain arbitrage, and we should try to detect the other extraction leg on the Polygon side.

To do this, we first need to see if this is the starting or ending leg of the extraction. The `getBridgeDirection` function provides us with this information. Furthermore, we need to extract the token address of the bridged token on the Polygon blockchain, the amount that is bridged, and the receiver or sender, depending on the direction of the cross-chain transaction. The `getPolygonBridgeTx` function aims to find the relevant transaction on the Polygon

blockchain that corresponds to the bridge segment of a cross-chain MEV extraction. This function requires several parameters, including the Polygon block number (polygonBlock), the direction of the bridge (bridgeDirection), the address of the bridged token (bridgedTokenAddress), the amount of the bridged token (bridgedTokenAmount), and the transactor involved in the bridged transaction (bridgedTokenTransactor).

After this, we need to determine the corresponding Polygon block number to the Ethereum block number where the initial transaction occurred to index the search on the Polygon blockchain. Additionally, the direction of the bridge is relevant because, depending on it, we might want to get the closest corresponding block with the upper or lower time limit. Based on all the obtained information, we can now call the `getPolygonBridgeTx` and `getPolygonSwapTx` functions, which provide us with the corresponding transactions for the Polygon leg. It's worth noting that the Polygon leg of extraction can be one or two transactions depending on whether the extractor performs the bridge operation with the swap operation in a single transaction.

At this point, we have the Ethereum leg transaction, which includes a swap transaction exhibiting MEV behavior and performing a bridge operation, as well as one or two transactions on the Polygon leg, corresponding to the bridge and swap transaction. We add all this information to the `detectedCrossChainMev` list and return it as the result of the detection.

Algorithm 1 Cross-chain MEV extraction matching algorithm

```
1: procedure detectCrossChainMEVExtraction
2:     Input: ethereumBlocks
3:     detectedCrossChainMev ← Empty List
4:     for all ethereumBlock ∈ ethereumBlocks do
5:         for all ethereumTx ∈ ethereumBlock.trasactions do
6:             if isNonAtomicArbitrage(ethereumTx) and isTouchingBridge(ethereumTx) then
7:                 bridgeDirection ← getBridgeDirection(ethereumTx)
8:                 bridgedTokenAddress ← getBridgedTokenAddress(ethereumTx)
9:                 bridgedTokenAmount ← getBridgedTokenAmount(ethereumTx)
10:                bridgedTokenTransactor ← getBridgedTokenTransactor(ethereumTx)
11:                polygonBlock ← getPolygonBlock(ethereumTx, bridgeDirection)
12:                polygonBridgeTx ← getPolygonBridgeTx(polygonBlock, bridgeDirection,
   bridgedTokenAddress, bridgedTokenAmount, bridgedTokenTransactor)
13:                polygonSwapTx ← getPolygonSwapTx(bridgeDirection, polygonBridgeTx,
   bridgedTokenAddress, bridgedTokenAmount)
14:                detectedCrossChainMev.append(ethereumTx, polygonBridgeTx, polygonSwapTx)
15:    return detectedCrossChainMEV
```

Next, we will walk through the second algorithm, which describes the function called `getPolygonBridgeTx`. The function has been designed to locate the specific transaction on the Polygon blockchain that corresponds to the bridge segment of a cross-chain MEV extraction. This function requires several parameters to function correctly, including the Polygon block number (`polygonBlock`), the direction of the bridge (bridgeDirection), the address of the bridged token (`bridgedTokenAddress`), the amount of the bridged token

(`bridgedTokenAmount`), and the transactor involved in the bridged transaction (`bridgedTokenTransactor`).

We define the `zeroAddress` variable, which represents the burn and mint address of the bridge, typically referred to as the zero address. The procedure then differentiates between two cases: when the bridge operates from Ethereum to Polygon and vice versa. This distinction serves two purposes. Firstly, it determines whether to search for the bridge transaction from this block onwards (in the case of the second leg of the extraction or coming from Ethereum) or to search backward (in the case of the first leg of the extraction or going to Ethereum). Secondly, it accounts for potential timing discrepancies in the bridge operation, which can affect the duration of the movement of assets between the blockchains.

Algorithm 2 Trace the Polygon bridge transaction algorithm

```
1:  procedure getPolygonBridgeTx
2:      Input: polygonBlock, bridgeDirection, bridgedTokenAddress, bridgedTokenAmount,
3:      bridgedTokenTransactor
4:      zeroAddress ← 0x...0
5:      if bridgeDirection == FromEthereumToPolygon then
6:          futurePolygonBlock ← getFuturePolygonBlock(polygonBlock)
7:          for all polygonTx = polygonBlock.transactions, ..., futurePolygonBlock.transactions do
8:              processedTransfers ← processTransactionTransfers(polygonTx)
9:              for all transfer ∈ processedTransfers do
10:                 if transfer.recipient == bridgedTokenTransactor then
11:                     if transfer.token == bridgedTokenAddress then
12:                         if transfer.amount == bridgedTokenAmount then
13:                             if transfer.sender == zeroAddress then
14:                                 return polygonTx
15:     else                                          ▷ direction is FromPolygonToEthereum.
16:         pastPolygonBlock ← getPastPolygonBlock(polygonBlock)
17:         for all polygonTx = pastPolygonBlock.transactions, ..., polygonBlock.transactions do
18:             processedTransfers ← processTransactionTransfers(polygonTx)
19:             for all transfer ∈ processedTransfers do
20:                 if transfer.recipient == bridgedTokenTransactor then
21:                     if transfer.token == bridgedTokenAddress then
22:                         if transfer.amount == bridgedTokenAmount then
23:                             if transfer.sender == bridgedTokenTransactor then
24:                                 return polygonTx
```

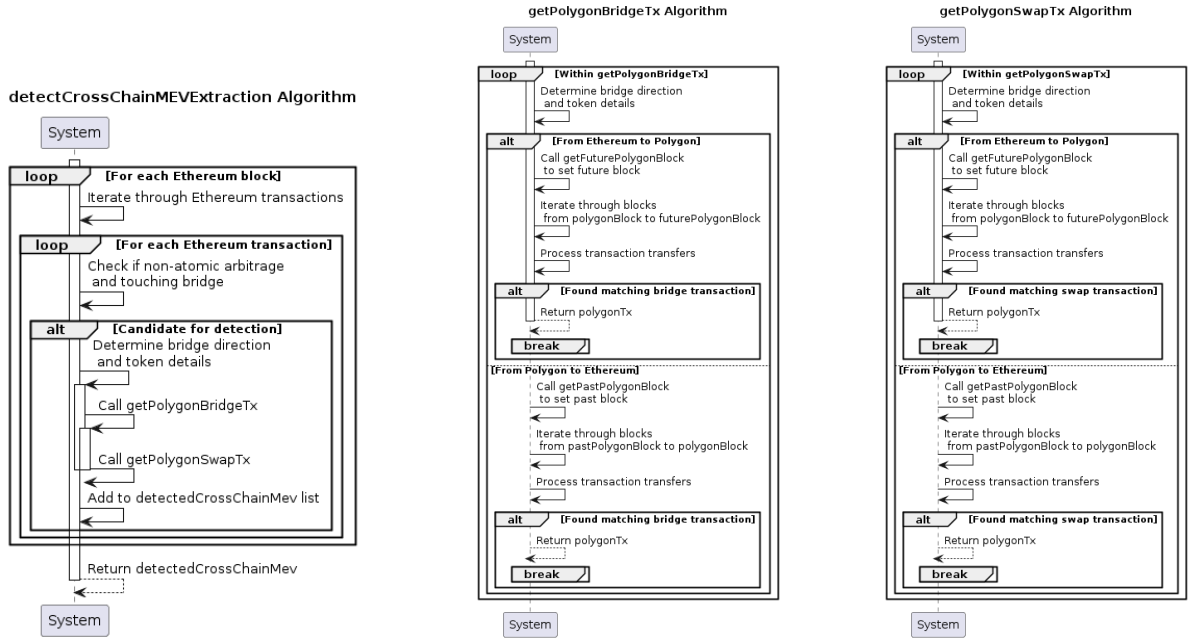
To find the bridge transaction, we need to specify an interval depending on the direction we are looking for. This interval is determined by the polygon block and the outputs of the functions `getFuturePolygonBlock` and `getPastPolygonBlock`, which set the boundaries of our search based on the maximum expected duration of the bridge operation. Next, we review each block's transaction and identify its transfers. Since we are only interested in transfers, we filter out other actions. Depending on the direction, we verify specific attributes of each transfer, such as the amount and token address. If the cross-chain extraction originates from Ethereum, the cross-chain transaction should mint the tokens, and the sender should

be the zero address. At the same time, the receiver should match the expected transactor. In the other case, the receiver should be the zero address (the burn address), and the sender should be the transactor. Once we have confirmed all these conditions, we can successfully identify and return the corresponding bridge transaction on the Polygon side.

The third algorithm is used to detect the swap operation, which marks the extraction's conclusion or initiation depending on the transaction's direction. This algorithm is similar to the previous one in that it requires defining the zero address, setting the bounded interval for the search, and processing the transfers. However, a key difference is that a precise match of the amount is not required; instead, a variance of up to 1% in value is allowed. This adjustment accommodates strategies used by the MEV extractor that may not precisely swap the exact value. We chose this approach based on observed data. To distinguish this from a burn or mint transfer, we ensure that the address exchanging the transactor's tokens differs from the zero address. Finally, the identified swap transaction is returned.

Algorithm 3 Trace the Polygon swap transaction algorithm

```
1:  procedure getPolygonSwapTx
2:      Input: bridgeDirection, polygonBridgeTx, bridgedTokenAddress, bridgedTokenAmount
3:      zeroAddress ← 0x...0
4:      polygonBlock ← getPolygonBlock(polygonBridgeTx)
5:      tokenTransactor ← getTokenTransactor(polygonBridgeTx)
6:      tokenAmountLowerInterval ← 99% of bridgedTokenAmoun
7:      tokenAmountUpperInterval ← 101% of bridgedTokenAmoun
8:      if bridgeDirection == FromEthereumToPolygon then
9:          futurePolygonBlock ← getFuturePolygonBlock(polygonBlock)
10:         for all polygonTx = polygonBlock.transactions, ..., futurePolygonBlock.transactions do
11:             processedTransfers ← processTransactionTransfers(polygonTx)
12:             for all transfer ∈ processedTransfers do
13:                 if transfer.token == bridgedTokenAddress then
14:                     if transfer.amount < tokenAmountUpperInterval then
15:                         if transfer.amount > tokenAmountLowerInterval then
16:                             if transfer.recipient ≠ zeroAddress then
17:                                 if transfer.sender == bridgedTokenTransactor then
18:                                     return polygonTx
19:     else                                                ▷ direction is FromPolygonToEthereum.
20:         pastPolygonBlock ← getPastPolygonBlock(polygonBlock)
21:         for all polygonTx = pastPolygonBlock.transactions, ..., polygonBlock.transactions do
22:             processedTransfers ← processTransactionTransfers(polygonTx)
23:             for all transfer ∈ processedTransfers do
24:                 if transfer.token == bridgedTokenAddress then
25:                     if transfer.amount < tokenAmountUpperInterval then
26:                         if transfer.amount > tokenAmountLowerInterval then
27:                             if transfer.recipient == bridgedTokenTransactor then
28:                                 if transfer.sender ≠ zeroAddress then
29:                                     return polygonTx
```



(a) (b) (c)

Figure 5.1.: Algorithmic methodology illustrated with diagrams

5.2. Data Collection

Blockchain data

One method to collect data from the Ethereum blockchain is by syncing a node with the network. However, this process can be time-consuming and can cause a bottleneck. Moreover, the process becomes more complicated since we plan to collect data from two blockchains. To address this, we designed the data collection process in a blockchain-agnostic way, which means that the module responsible for collecting blockchain data from Polygon or Ethereum is shared. All that is required is a valid RPC connection plugged into the module to enable access to a blockchain-synced node. In the case of Ethereum, the node must have tracing enabled, which allows for fetching transaction traces. A trace is a detailed record of all the steps taken by the EVM during execution, including all the operations performed and the changes made to the blockchain state.

For RPC connections, we used https://eth.llamarpc.com for Ethereum and for Polygon https://polygon-mainnet.infura.io/v3. However, these connections can be replaced with any other RPC connections that meet the abovementioned requirements, and the data collection process should still work. We collected and analyzed data from 1,000,000 blocks, from block 18,500,000 to block 19,500,000, approximately equivalent to 140 days of data, from Nov-04-2023 to Mar-23-2024.

Zeromev

Zeromev is an organization that aims to protect Ethereum users from frontrunning and censorship [46]. The Zeromev API provides users with access to transaction-level MEV summary data from the Ethereum blockchain. We fetch information from this API, and then we label all the previously collected transactions with the given label. This REST API provides information on various types of events, including:

1. Arb: Refers to arbitrage transactions that aim to gain profits from price differences across exchanges.
2. Frontrun: Indicates transactions that initiate an attack by front-running in a sandwich, causing prices to move against the victim.
3. Sandwich: Refers to transactions of the victims in a sandwich attack, where there can be one or more victims.
4. Backrun: Describes the closing transaction in a sandwich attack, which allows the attacker to profitably close their position.
5. Liquid: Indicates events involving liquidation in a DeFi lending protocol.
6. Swap: Includes swap transactions to provide volume data for non-MEV transactions.

FindBlock

FindBlock offers a public API designed to streamline typical operations when interacting with EVM chains [47]. Our utilization of the API involves locating Polygon blocks that closely match a specified timestamp.

Polygon token mapping

Token mapping is a crucial process for enabling the easy transfer of tokens between Ethereum and Polygon. Before depositing any token from Ethereum to Polygon, it must undergo mapping. This entails registering the relationship in the *RootChainManager*, which is the entry and exit point contract deployed on Ethereum. The *RootChainManager* specifies that Contract A on Ethereum is mapped to Contract B on Polygon. For each mapped token, there is an associated predicate that interacts with the token during deposit or withdrawal. Each predicate represents the token's class type, and changes in token type may require corresponding modifications to the methods implemented by the tokens, which in turn necessitate adjustments to the interacting predicate. Polygon offers an API that provides addresses of all mapped tokens, which we use to determine the corresponding Polygon token address for an Ethereum token address [48].

5.3. Implementation

During the previous discussion, we provided a conceptual overview of our methodology, which did not include the details required to implement such a detection algorithm. This section will explain our approach to implementing these functionalities using the Polygon bridge as a study case. We also integrated several external libraries and technologies together with the data obtained from the collection process, which we will highlight in this section.

Python

Python is a popular programming language that is known for its ease of use and readability. It is an excellent choice for data collection and analysis in a blockchain environment, especially for research projects. The language's vast libraries, easy-to-understand syntax, and strong ecosystem make it ideal for running algorithms, managing large datasets, and seamlessly interacting with blockchain networks.

Web3.py

In the world of blockchain technology, powerful tools that can efficiently interact with blockchain networks are often needed to conduct comprehensive data collection and analysis. One such tool is web3.py, a Python library designed specifically for Ethereum, but it can also be extended to other EVM-compatible blockchain networks. The library bridges Python applications and Ethereum nodes, making it easy to interact with Ethereum networks. This simplifies tasks such as querying blockchain data, sending transactions, and deploying smart contracts, all from within Python scripts. With web3.py, retrieving blockchain data such as transaction histories, smart contract states, and token balances is straightforward, making comprehensive data gathering for analytical purposes possible. Moreover, web3.py facilitates interactions with deployed contracts, allowing function calls, state readings, and event monitoring. While initially built for Ethereum, web3.py can be adapted to interface with other blockchain networks that support the JSON-RPC protocol, making it more versatile.

Non atomic arbitrage detection

In contrast to atomic MEV, where identifying transactions as part of an MEV action is highly confident, non-atomic MEV introduces more uncertainty. Specifically, non-atomic behavior is determined based on specific transaction characteristics rather than pinpointing the entire batch of transactions extracting MEV. Our literature review has shown that previous research on this topic has taken a stricter approach to identifying these transaction characteristics [37], [33]. Our approach is modeled after Xiao et al. [33], but with relaxing some of the criteria. Although this broader approach may result in more false positives, it is not a concern, as our primary objective is to track these transactions on the Polygon network and assess their profitability. Therefore, if we cannot trace a transaction, it is likely a false positive that we can eliminate in the final analysis.

Function selector	Function signature	Direction
0x4faa8a26	depositEtherFor(address)	From Ethereum to Polygon
0xe3dec8fb	depositFor(address, address, bytes)	From Ethereum to Polygon
0x3805550f	exit(bytes)	From Polygon to Ethereum

Table 5.1.: Polygon bridge direction classification

Our algorithm uses data from the Zeromev API, specifically choosing transactions labeled as *Swap*. Within these labeled transactions, we classify it as a potential non-atomic arbitrage if it satisfies one of the following conditions: it involves a bribe to the validator (a coinbase transfer) or if the transaction is among the top 10% by block position. It is important to note that the second condition generates most false positives.

Polygon Bridge interaction

We aim to determine whether a transaction involves sending or receiving tokens via the Polygon bridge. However, the Polygon bridge lacks documentation on this process, therefore, we came up with two options. One way is to identify specific events emitted during a transaction interacting with the Polygon bridge. Alternatively, we can analyze the transaction traces. In the former case, as outlined in the Polygon bridge case study, we discovered events emitted by tokens. However, these events could originate from tokens not necessarily interacting with the Polygon bridge, which could lead to false positives. On the other hand, analyzing the transaction traces provides a more effective solution involving a detailed analysis of every step and internal calls within a transaction.

We determined that analyzing the transaction traces was more suitable for our needs. Therefore, we obtained the transaction traces from the blockchain node and systematically reviewed each trace. This process revealed every smart contract call executed during the transaction. If a smart contract call was made to the *RootChainManagerProxy* at address *0xA0c68C638235ee32657e8f720a23ceC1bFc77C77*, it indicates potential token activity via the Polygon bridge. Furthermore, we wanted to discern whether the transaction involved sending or receiving tokens. To achieve this, we used function selectors to examine the type of call made at that address. Table 5.1 shows our classification based on the mentioned information.

Polygon Bridge transaction processing

Our aim is to retrieve specific details such as the token amount, the sender or recipient (transactor), and the token's address on the Polygon blockchain when a user initiates a cross-chain transfer via the Polygon bridge. To capture the amount and transactor, we make use of the *Transfer(address from, address to, uint256 value)* log event that provides the necessary information about the transferred amount. In cases where transfers occur from Polygon to Ethereum, the recipient is typically the same as the sender. Therefore, we identify the transactor on the Polygon side by extracting the "to" information from this event. On the

other hand, the *Locked* event, which varies based on the type of token being transferred, contains information about the recipient, which is crucial for our application's needs. To identify the mapped token, we utilize the Polygon token mapping API, which provides us with the token's address on the Polygon blockchain.

Polygon bridge transfer duration

The functions `getPastPolygonBlock` and `getFuturePolygonBlock`, employed both in the second and third algorithms, are used to retrieve the search boundaries for a Polygon leg. The former is used in cross-chain transfers from Polygon to Ethereum to find blocks that come before the token reception event on the Ethereum blockchain. To determine the typical duration for a checkpoint submission, we have examined block explorers and settled on a 5-hour window. This means that we will search through 5 hours of Polygon blockchain blocks to locate the bridge transaction.

On the other hand, the latter function is used to locate the second leg of a cross-chain transfer from Ethereum to Polygon. Here, the goal is to explore blocks that come after the token has been sent to the bridge contract on Ethereum. Similarly, based on our analysis of the block explorer, we have opted for a 1-hour interval of blocks.

Transaction transfers processing

The task of the `processTransactionTransfers` function is to handle transfer events that may occur within a transaction. This function is responsible for identifying the *Transfer(address from, address to, uint256 value)* events, which can provide us with important details such as the sender's address, recipient's address, and the amount transferred. Additionally, by examining the address where the event was triggered, we can determine the token involved in the transaction.

Swap processing

Once we have identified all transactions involved in cross-chain arbitrage, our goal is to analyze the events that occur during these transactions. Specifically, we are interested in understanding the types of tokens and the amounts involved in the arbitrage. To achieve this, we focus on processing Swap events that occur within the exchange transactions of each arbitrage leg. It is worth noting that different DEX pools may use different event signatures to indicate token exchanges. Our investigation is limited to two primary types of DEX pools: Uniswap V2 and Uniswap V3. Table 5.2 demonstrates the relationship between events and the extracted information for our analysis. Using this approach, we have successfully identified all transactions involved in cross-chain arbitrage, including the exchanged amounts between different token types and the entities involved in the process.

DEX type	Swap event signature
Uniswap V2	event Swap(address, uint128, uint128, uint128, uint128, address);
Uniswap V3	event Swap(address, address, int256, int256, uint160, uint128, int24);

Table 5.2.: Swap events signatures

5.4. Alternative Strategy

During our testing of the methodology, we observed an intriguing occurrence during the extractions. Specifically, we found that certain arbitrage transactions from Ethereum to Polygon were not completed, despite our methodology detecting transactions up to the Polygon bridge one. Upon further investigation, we discovered that some searchers occasionally opted to return the funds without finalizing the arbitrage. This behavior is likely because other actors had already resolved the price discrepancies in the tokens, making it advantageous to return the funds and exchange them for the original token in their possession.

To address this issue, we refined our methodology by incorporating an algorithm to identify these instances. The idea is to detect situations where no swap transaction occurs before or after the bridge operation, indicating an incomplete arbitrage. In such cases, similar to how we identify the Polygon bridge transaction, we reverse the process to locate another bridge transaction on Polygon that performs another bridge operation in the reverse direction. Suppose an actor executes this action using token A and token B, where token B is susceptible to arbitrage. The following sequence of operations would describe the process:

1. Swap tokens A to B on the source blockchain
2. Bridge tokens B from source blockchain
3. Bridge tokens B to destination blockchain
4. Bridge tokens B from destination blockchain
5. Bridge tokens B to source blockchain
6. Swap tokens B to A on the source blockchain

We recognize that depending on the arbitrage leg, certain operations could potentially be merged into a single transaction. For instance, our approach can identify arbitrages that involve a swap and bridge operation within a single transaction at either the beginning or end of the extraction. However, our method is unable to detect cases where the process begins on Polygon, as the destination chain (Ethereum) involves only bridge transactions. Thus, with this additional algorithm, we can specifically identify operations originating from Ethereum. Given that an actor may combine the bridge and swap operations at the start or end of the process, our method can pinpoint either the initial or final operation within the described sequence.

6. Results

In this chapter, we will be sharing the results and analysis of our study on cross-chain MEV identification between Ethereum and Polygon utilizing the Polygon bridge. We will begin by discussing the frequency and duration patterns we observed. Then, we will delve into the tokens involved in arbitrage. Lastly, we will examine the searchers who conducted the operations, their revenues, and the corresponding gas fees they incurred.

6.1. Frequency and Duration

We have successfully identified 4,488 instances of cross-chain MEV extraction using our described methodology. Figure 6.1 illustrates the difference in the number of extractions that originated from Ethereum and concluded on Polygon (blue) versus the number of extractions that originated from Polygon and concluded on Ethereum (green). We have noticed that there is a difference in the frequency of arbitrage extraction based on direction. Although this difference is not significant, we should consider why extractors prefer certain directions for arbitrage.

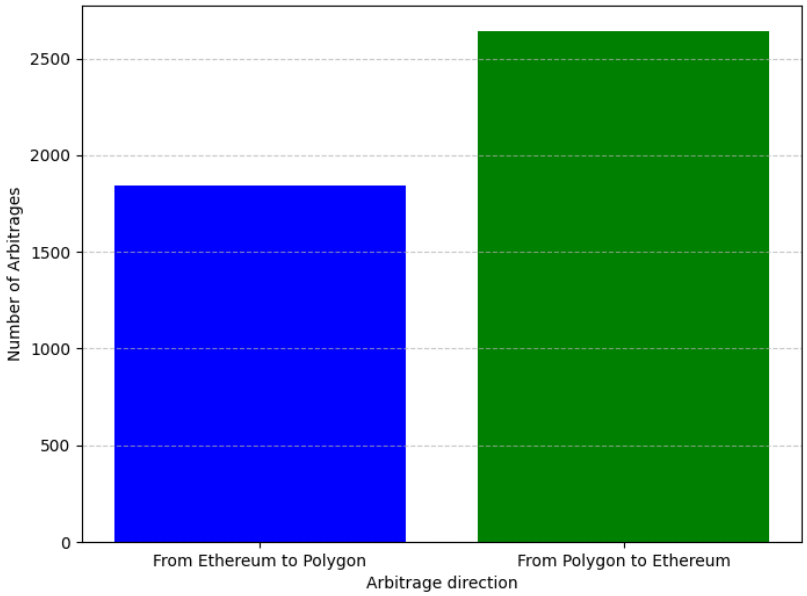


Figure 6.1.: Distribution of executed arbitrages

One possible explanation is that our analysis approach has limitations. We identify arbi-

trage opportunities by analyzing Ethereum transactions that involve non-atomic arbitrage and interactions with the Polygon bridge, in the same transaction. However, a different approach could involve conducting a swap on Ethereum (which is also a non-atomic arbitrage) followed by a bridge transaction later. Our current methodology would overlook this scenario, which might be more profitable for those initiating transactions from Ethereum to Polygon. Another factor that could affect arbitrage is the duration of the process. The longer it takes to complete an arbitrage, the less likely it is to be executed profitably before others.

In addition, we have refined our analysis by applying the definition of cross-chain cyclic arbitrage. It is worth noting that we found 3,901 of these extractions to be cyclic arbitrages. It's important to understand that even if an arbitrage is not cyclic, it remains valid. However, calculating profit becomes more complex as we must consider historical prices between the tokens involved in the arbitrage. For revenue analysis, our primary focus is on the cyclic arbitrage instances. Due to the non-atomic nature of cross-chain MEV, our theoretical framework has highlighted the risks associated with this behavior. Therefore, our findings have revealed that among the cyclic arbitrages, some resulted in losses. In particular, 590 transactions incurred losses while 3,311 transactions yielded a positive revenue.

Figure 6.2 illustrates the daily frequency of executed arbitrages. It indicates a range from 17 to 52 arbitrages per day, with an average of 32. Arbitrage opportunities typically arise during market volatility. Thus, we anticipate that days with a high number of executed arbitrages correspond to days of significant market volatility.

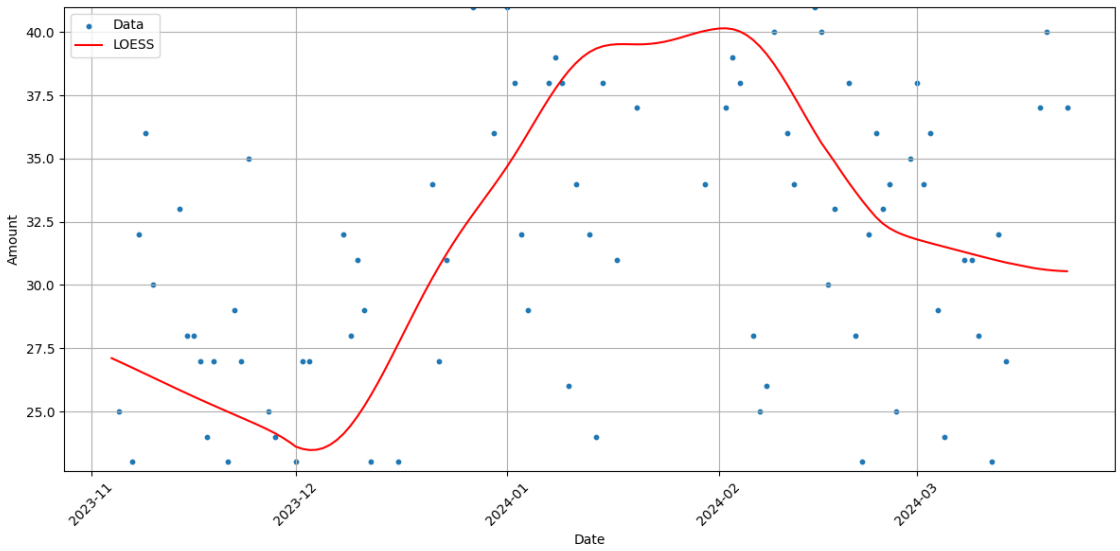


Figure 6.2.: Scatter plot with the number of arbitrages per day along with a LOESS line

In Figure 6.3, we can see the duration of arbitrages using a rolling technique with a one day window. As expected, there is a noticeable difference in the time it takes for transfers in different directions. This is because the Polygon bridge uses different mechanisms for token transfers in each direction. As a result, it takes longer to complete an extraction from

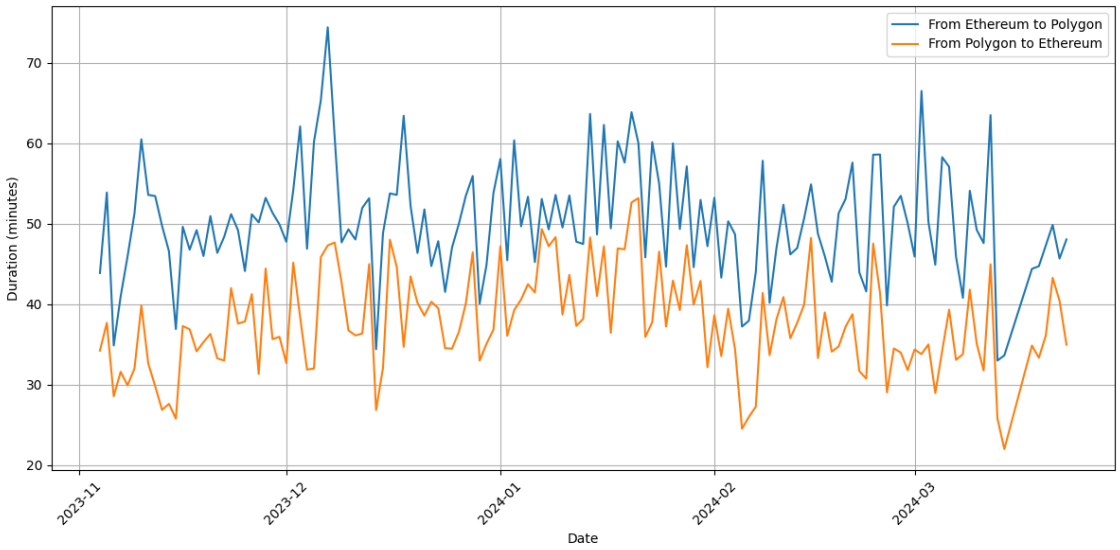


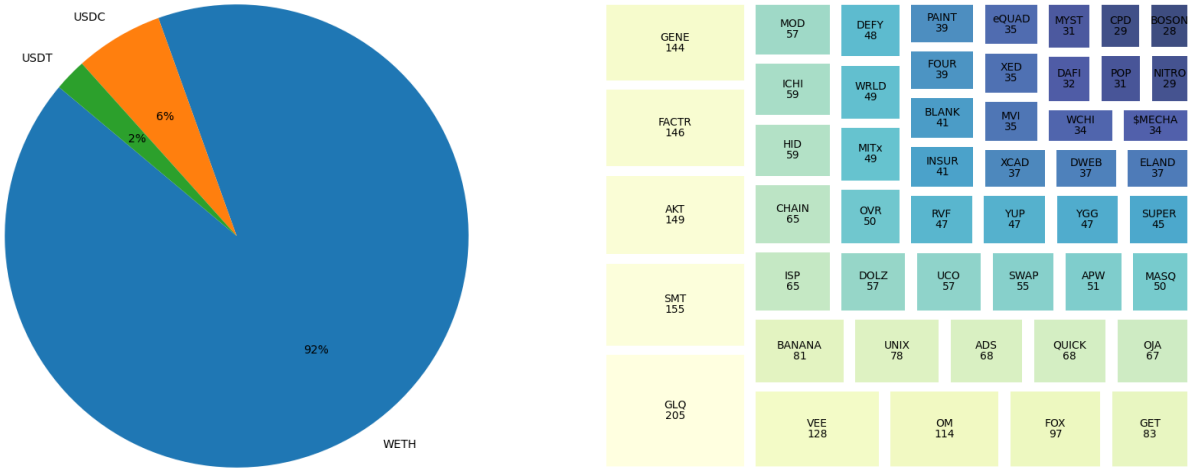
Figure 6.3.: Visualizing arbitrage durations over time: The time series plot captures the durations of arbitrages using a rolling one-day window

Polygon to Ethereum compared to the reverse direction. We conducted a correlation analysis and obtained a result of 0.67. This indicates that there is a fairly strong positive association between the durations of arbitrage in both directions. As one duration increases, the other tends to increase too, and vice versa. This correlation is probably due to the operational intricacies of the Polygon bridge. However, there is a lack of clear documentation on this aspect. On average, extracting MEV from Ethereum to Polygon takes 20 minutes and 8 seconds, while the other direction takes an average of 50 minutes and 17 seconds.

6.2. Tokens

In this section, we will discuss the tokens used in cross-chain arbitrage. Figure 6.4a provides a visual representation of the profit token used in cyclic cross-chain arbitrages. The profit token is defined as the token that is used by the arbitrager to start and end the cross-chain extraction process. It is observed that wrapped Ether is the most commonly used token for arbitrage, followed by variants of USD stablecoins in the second and third positions.

Next, we examined the tokens most frequently used for bridging. In cases where cross-chain MEV extraction involves just one swap per blockchain leg, this specific token becomes the only target for exploiting price disparities between blockchains. If there are multiple swaps on any leg, it's still likely that the primary arbitrage profits stem from price variations in this token. However, there's a chance the extractor could benefit from multiple price discrepancies across different tokens involved in simultaneous swaps. In Figure 6.4b, we identified the symbols of the most bridged tokens in our study. This analysis segment reveals a significant number of tokens in play. It suggests that extractors likely monitor a



(a) Profit token

(b) Bridged tokens

Figure 6.4.: Tokens used in arbitrage operations

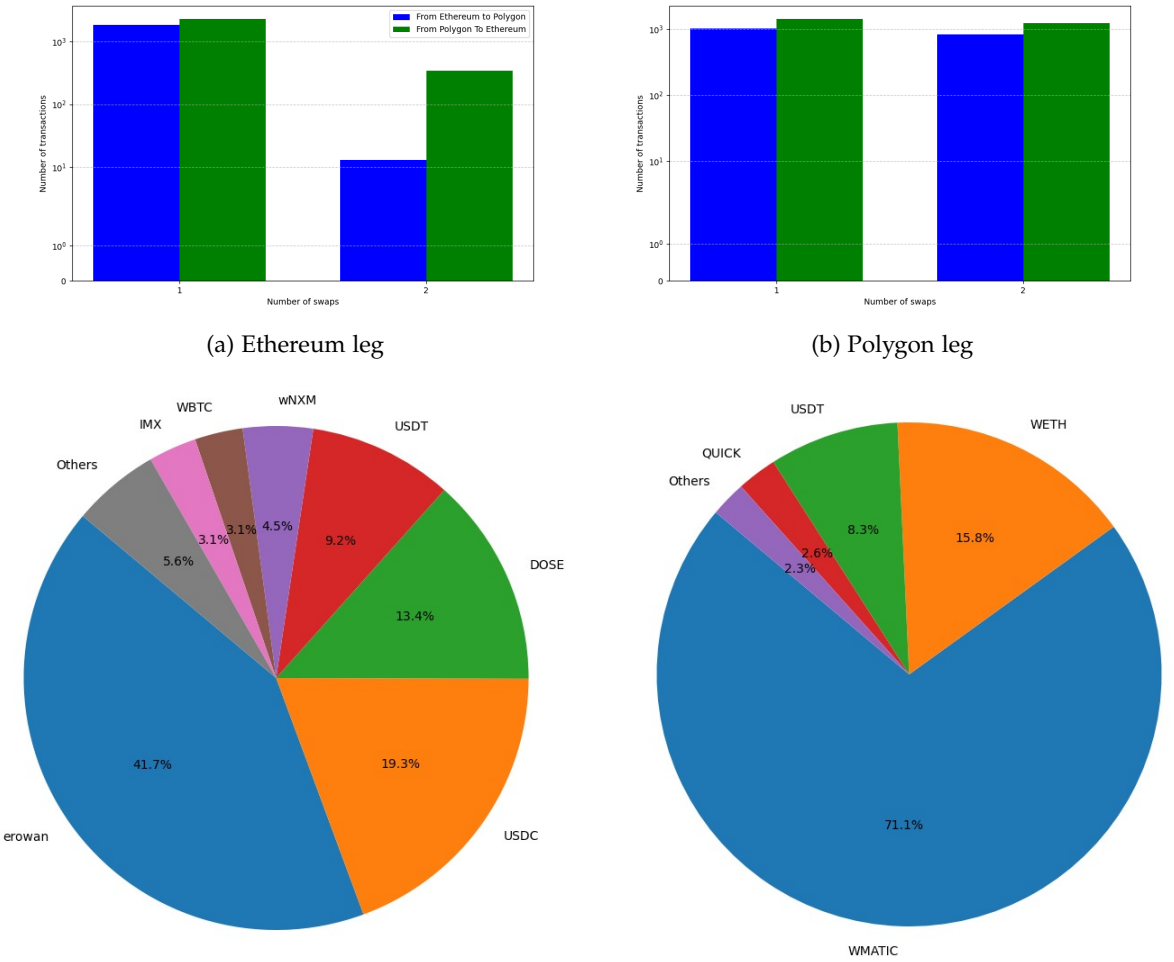
broad selection of tokens available via the Polygon bridge, employing extraction strategies abstracted to the token level rather than focusing on a limited set.

Furthermore, it's notable that the tokens involved in the bridge operation are not widely recognized tokens with large market caps or significant traction. In an earlier part of the thesis, we explored arbitrage strategies for tokens listed across multiple blockchains and found that executing parallel MEV between blockchains appears to be a more effective approach. This involves holding an asset that is being arbitrated across multiple blockchains, and when a price difference emerges, buying the asset where it's cheaper and selling where it's more expensive simultaneously.

Another strategy under consideration currently is to possess a profit token (such as a stablecoin or a wrapped version of a native blockchain token) and then conduct swap operations and bridge transactions. Based on this analysis, it appears that miners are unable to exploit price discrepancies in well-known tokens across different chains using bridges. This could be attributed to the fact that these tokens are already being arbitraged through parallel MEV, and bridging a token to another blockchain would take too much time.

Conversely, this strategy seems to be successful for low-liquidity tokens, which are typically less popular. Managing an inventory of a large number of tokens on each blockchain and then executing parallel MEV might be too costly, with the opportunity cost being too significant.

In each step of the process, at least one token swap is required, but multiple swaps are also possible. Figures 6.5a and 6.5b shows how many swaps are typically performed on each leg during extraction. Additionally, we distinguish between the direction of arbitrage. There is a noticeable difference in the number of swap counts between Ethereum and Polygon. It seems that searchers prefer to conduct two swaps on the Polygon leg rather than on the Ethereum leg. Performing more swaps consumes more gas. This trend is likely due to the



(a) Ethereum leg

(b) Polygon leg

(c) Ethereum leg

(d) Polygon leg

Figure 6.5.: Swap operations

fact that gas costs are higher on Ethereum than on Polygon, making it more advantageous for arbitrageurs to limit swaps on the Ethereum leg to just one.

Furthermore, Figures 6.5c and 6.5d illustrate the tokens engaged in the swap operation when multiple swaps are conducted. Notably, the usage of popular tokens with high market caps, including stablecoins, is evident. This observation suggests that extractors who employ a strategy involving multiple swaps per operation may not necessarily be exploiting price variances across multiple tokens simultaneously, as these widely recognized tokens are likely already arbitraged by specialized traders. Instead, this approach may be adopted due to the absence of a direct token pair between the profit token utilized and the bridged token.

6.3. Searchers

As part of our investigation, we have uncovered information about the entities involved in cross-chain MEV extractions. The goal is to identify the addresses of these entities and the characteristics of their profit-making activities. These entities usually operate through specialized smart contracts designed for this purpose. In previous research, they have been identified by the addresses of these contracts. We will use a similar approach. Each searcher uncovered by our detection has deployed two smart contracts, one on Ethereum and one on Polygon. We will refer to the Ethereum-deployed smart contracts throughout our analysis and the presented data.

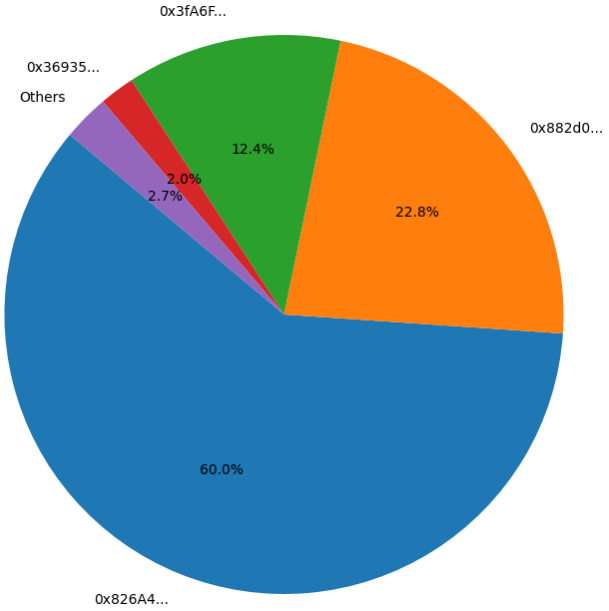
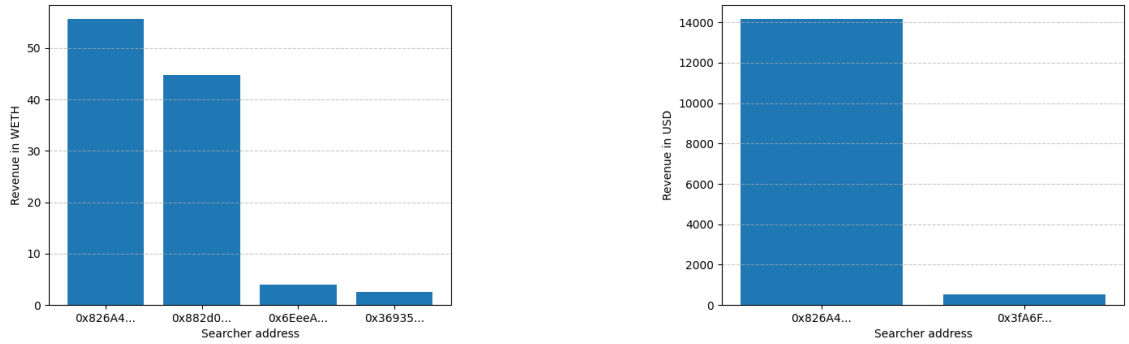


Figure 6.6.: Searcher domination by number of arbitrages

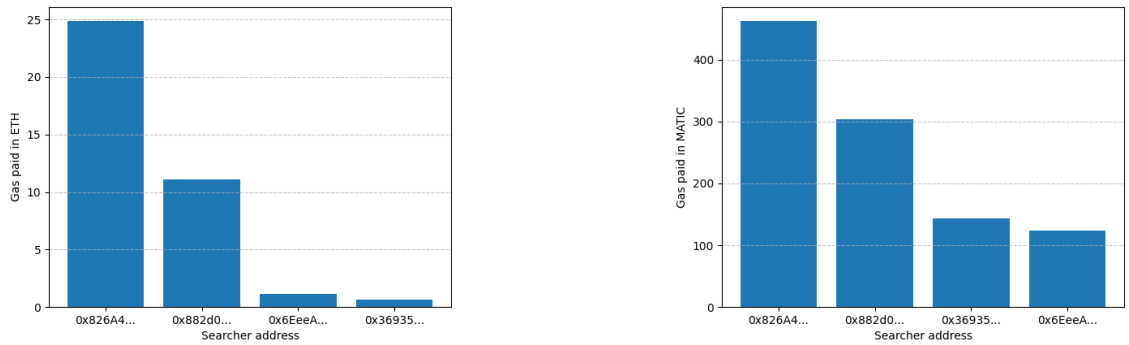
We conducted an analysis of the dominance of searchers based on their arbitrage activity. Figure 6.6 displays their dominance based on the number of arbitrages performed. This shows that only a limited number of searchers are primarily responsible for this activity, with just three accounting for 95% of the arbitrages. This observation is in line with concerns raised in the literature about the centralization forces of MEV. In particular, if a few searchers consistently succeed in extracting MEV, they can collect value to improve their algorithms and infrastructure, thereby consolidating their dominance in the space. As they amass more resources, this trend could lead to centralization within the cross-chain domain.

Subsequently, as shown in Figure 6.7, we conducted a similar analysis focusing on revenue generated through cross-chain cyclic arbitrages. This time, we examined arbitrages involving WETH as the profit token and those involving a USD stablecoin variant. We also included arbitrages resulting in losses in our charts. Notably, despite one actor dominating with 60% of the arbitrages, their WETH revenue is not so far from that of the second-place actor, who held a 22.8% share. This information is not complete since the analysis does not take into



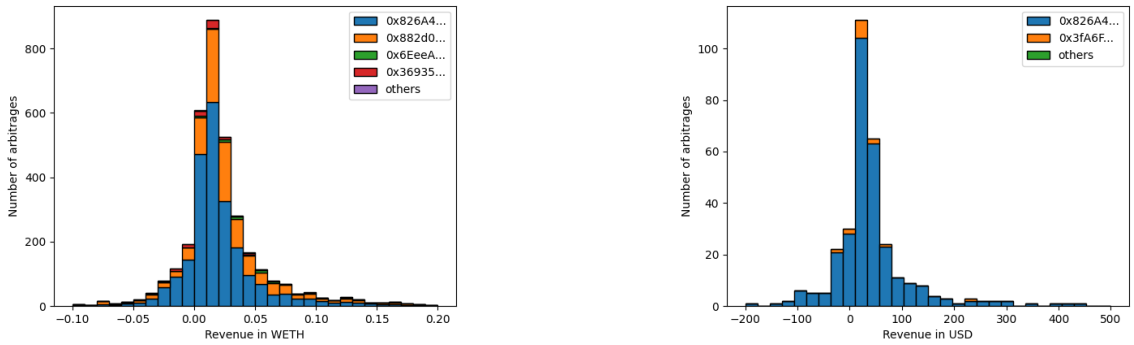
(a) WETH (b) USD

Figure 6.7.: Searcher domination by revenue



(a) Ethereum (b) Polygon

Figure 6.8.: Searcher gas paid



(a) WETH (b) USD

Figure 6.9.: Distribution of revenue per arbitrage

consideration the fees paid, but this can be an indication that the searchers use different algorithms and strategies.

Next, in Figure 6.8, we analyze the fees paid by the dominant searchers. Although we need more information on direct bribes that searchers might pay to validators through coinbase transfers within the same or different transactions, this still offers insight into the potential net profit for these actors. Additionally, as noted in the literature, integrated searchers acting also as validators may not necessarily need to pay high fees, given their dual role.

Moreover, each arbitrage involves Ethereum and Polygon blockchain fees, necessitating the conversion of these fees using historical prices to calculate net profit. Examining the top two searchers, we observe a stronger correlation between fees paid and the number of arbitrages conducted compared to previous charts. Considering these factors, we can infer that while the first searcher earns a higher revenue, the second searcher likely maintains a better profit margin.

Finally, Figure 6.9 illustrates the distribution of revenues generated from each arbitrage operation. Out of 3,311 cyclic arbitrages, 590 resulted in a loss. The average revenue for cyclic arbitrages using WETH was 0.02, whereas for those involving a USD stablecoin variant, it was 45. The highest revenues achieved using different tokens were 1.83 WETH and 708 USD, while the maximum losses were 0.61 WETH and 443 USD.

6.4. Alternative Strategy

Our enhanced detection system successfully identified 158 occurrences where actors opted to revert token bridging instead of completing the swap process. Among these, our algorithm flagged 148 instances of the final two steps (reverting back to Ethereum and swapping to the original token), and 9 occurrences of the initial two steps (swapping and bridging from Ethereum). We will showcase one such example:

1. Ethereum: [1] Swap 0.449 WETH to 20,017 GoBlank
2. Ethereum: [2] Bridge out 20,017 GoBlank from Ethereum
3. Polygon: [3] Bridge in 20,017 GoBlank to Polygon
4. Polygon: [4] Bridge out 20,017 GoBlank from Polygon
5. Ethereum: [5] Bridge in 20,017 GoBlank to Ethereum and Swap 20,017 GoBlank to 0.452 WETH

Our non atomic arbitrage detection heuristic detected step 5 of the arbitrage. Interestingly, the actor manages to achieve a small revenue of 0.003 WETH. However, in practical terms, factoring in all gas fees and expenses, this is more likely to result in a net loss.

[1] 0x72028385e005728895504558c341387f2ec107c051a1ddaeeaec725417a4d82e
[2] 0xc107bcc5bc111855e9af5e76a5e0a245f6fa2ebaa02ef11cd561a538d06e5dc1
[3] 0x8670138151e0d5ee8eca5c6977d2239ac90c787f7fd07cff56280e4e9c9f4ee2
[4] 0x0d6eb8fa1d6ee18f8d425884050406592f923751affe9752f844f9da18d901a7
[5] 0x70c74828317a2544c4981bb9b2d95742fb78767063f38ac3000eb7c47dcfe4db

7. Discussion

7.1. The Generation of MEV By The Bridges

Our investigation focused mainly on the Polygon bridge, which acts as a token bridge that allows tokens to be transferred between different blockchains. We also observed the bridge's role in enabling value extraction across various chains by using different liquidity pools on interconnected blockchains. Although the Polygon bridge does not perform the extraction itself, it plays a crucial part in the process. Without it, there would be no means of extraction, and the tokens would not be created initially, as they are multi-chain tokens created with the Polygon sync state as an interface.

Moreover, the validators of the Polygon bridge have the power to decide when tokens are bridged and what is included in these bridge operations. In the context of cross-chain MEV, bridge validators wield a similar level of authority to single-chain MEV's blockchain validators. They are the ultimate actors in the bridge supply chain, much like how blockchain validators are the ultimate actors in the transaction inclusion supply chain.

Furthermore, there exist more complex bridges than the Polygon one, capable of performing more sophisticated operations. Despite the largely unexplored nature of this domain, we believe that the influential position held by bridge technologies can be even increased by the more complex features within blockchain bridges.

7.2. Limitations

In our research, we were unable to address every extraction scenario. Essentially, this means that although our methodology can identify a significant portion of cross-chain MEV extraction via the Polygon bridge, it is not exhaustive, and we did not achieve full coverage of all cases. These limitations highlight areas where our methodology could be further improved. Furthermore, we will outline the limitations that exist.

- Our detection heuristic begins by identifying non-atomic arbitrage transactions that also involve a bridge operation. During our research, we noticed that different users have different methods of handling swap and bridge operations - some choose to combine them into a single transaction, while others perform them in separate transactions. While our methodology covers both scenarios on the Polygon side, we only consider cases where the operations are merged on the Ethereum side. Therefore, if a user performs these operations in separate Ethereum transactions, our detection algorithm will not be able to detect it.

- The detection of non-atomic arbitrage serves as a constraint in our system. It is employed to identify potential candidates that would be matched with transactions on the alternate leg of cross-chain arbitrage. Nevertheless, transactions not conforming to the prescribed non-atomic arbitrage heuristics might still participate in cross-chain arbitrage. For instance, a searcher could utilize advanced strategies to circumvent the detection of coinbase transfers and manipulate the positioning of the transaction.
- As explained in the alternative strategy sections of our study (5.4 and 6.4), we aim to identify searchers who use the bridging tokens back strategy. For the same reasons outlined earlier, our approach explicitly addresses situations where a bridge and a swap operation are combined during the Ethereum leg. Our method cannot detect cases where the process starts on Polygon, as the destination chain (Ethereum) only involves bridge transactions.
- Extractors can use more complex techniques to conceal a direct link between bridge and swap operations. For instance, a searcher can bridge 100 tokens but choose to swap only 50 of them. Because our search criteria focus on swap transactions that exchange between 99% and 101% of the token amount, we would not be able to detect this strategy.
- We address scenarios in which searchers begin with a single profit token. Nonetheless, it's also possible to start with multiple tokens, execute multiple swaps, and ultimately consolidate them into a single token for bridging.

7.3. Further Work

At first, it would be useful to investigate any limitations in the approach as groundwork for future research. Additionally, a deeper understanding of the incentives between searchers, bridge validators, and blockchain validators can be gained by conducting further investigations into their dynamics. This could done by examining the strategies used by searchers to gain an advantage over others. Are they employing bribery with blockchain validators or bridge validators and, if so, what percentage of profits are they sharing with them? Are they using private relays? Answering these questions would significantly contribute to understanding the phenomenon.

8. Conclusion

We conducted a study, focused on Ethereum and Polygon, on the use of blockchain bridges to extract MEV across different blockchains. Our approach involved understanding the bridges and cross-domain MEV, clarifying the connections between actors, and outlining our research objectives. To achieve this, we created a methodology to identify cross-chain MEV extraction across different chains, which we applied to the Polygon bridge as a case study.

We implemented the methodology which led us to identify 4,488 extraction instances. Out of these, 3,901 were determined to be cyclic. Although some extractions led to losses, most (3,311 cases) were profitable. Additionally, we looked into an interesting scenario where participants chose to bridge back funds instead of executing the arbitrage, potentially resulting in a loss. We also observed that arbitrage activities occur daily, averaging around 32 per day. The time required to complete an arbitrage varies depending on the direction and is influenced by the operational mechanics of the Polygon bridge.

We have analyzed the tokens used in arbitrage scenarios. Our findings indicate that profit tokens typically consist of well-known tokens, often the native blockchain token and stablecoins. Conversely, the tokens involved in arbitrage due to price discrepancies are normally lesser-known tokens with low market capitalization, and arbitrageurs engage in transactions across a broad range of these tokens. We believe this is due to excessive bridging times, which make arbitraging high-volume tokens impossible, as prices do not remain stable during the bridging process. Moreover, searchers may use multiple tokens in a single swap operation. In addition, we have examined the individuals who engage in these arbitrage activities. We have identified several of them and assessed their profits and the gas fees paid to validators. Our analysis revealed that these arbitrageurs employ diverse strategies, but their numbers are relatively limited, with only three individuals accounting for 95% of all arbitrage transactions.

To summarize, our research has made significant progress in detecting cross-chain MEV extraction through bridge analysis. Our study has successfully demonstrated how various actors make profits using the Polygon Bridge through different strategies. This field is relatively unexplored, and our work lays the groundwork for further research and helps anticipate potential outcomes. Moreover, our analysis concludes that this phenomenon may hinder decentralization efforts if not properly understood and conducted transparently.

A. Appendix

Listing A.1: Searchers Ethereum smart contract address to their externally owned accounts

```
0x1231DEB6f5749EF6cE6943a275A1D3E7486F4EaE:
    {0x0902fBd41516bb64CFb93D514D7E40A0C4E47919},
0x1646A4761aA54f23d7F4d5deB5D393F67D318B80:
    {0x34afdddDE003bB2d5A45151Cfa037a32ED804173,
    0xa9cE598b9286ACECF2E495D663FaA611256910F1,
    0xc24440ee887dc6Dc2b8adDd66A4E8d1FcB18E018,
    0xd835A69Fed3f06DCE244c2cECA11AAaBD831892c},
0x369350fE4421F16dBcfE28CB3943e7421BA48561:
    {0xcEd2A0e1C3d5B96D8cbeAd377b897f050b980574},
0x3fA6Fff7212D3fA4317cF1955fa690993d8ceD70:
    {0x0a6c69327d517568E6308F1E1CD2fD2B2b3cd4BF},
0x6EeeAFa18f7e764234AfFBe2A29c24Be76184F46:
    {0x121EDAD46B92B9388f721a9B29aFBB4721C3116c,
    0x30F0e667Dd87ac2Bf83CB86110FBc20709387048,
    0x3E97184d6cA0F505033aE11c0aB6a3ee1b43C9bb,
    0x5C988D4A5a7A3eA96B2F8cb8eF86D302b100DA1D,
    0x6eA16D099D1c0C2A56B0CF2b687d48F413988F39,
    0x7e5d96B086B3A1e171CFBbb38d645a3C41a195e4,
    0xE1C58Fcb072dE6b083bf5a40C2d87c00735Eca0e,
    0xFe2FeA957f033ccb46aA3f45E61Bd87749CEa7dD,
    0xd124Ee6A81A41741Fc3bb71dE1d353cde007D41B},
0x826A4f4DA02588737d3c27325B14F39b5151CA3C:
    {0x31e1a9079549FF110F5E96cdaD5100474612995D,
    0x83ECfd72514133D21f91DDcbAb70C6175E8EFA88,
    0x8518c521eba4e725F227a930014a997Bf613D5C5,
    0xE83F75907Fb4c575414FA6F5cfe8cef24Dc5870C,
    0xb8dC07c0E8d60Ade09994E9cBb8CE8Ac46e35c96},
0x882d04C3D8410dDF2061B3cBA2c3522854316FEB:
    {0x08D9894495320f38561b35B31c9EcE9a4dC5712A,
    0x38d07d33B3F10C4C807C154EfA320536EBC8C505,
    0x467B79AAfD7977F6d1E772e0b121047AC655C389,
    0x65D5e6D30B00c3Dbf49B426afc81B8bB7dbaE8AB,
    0x695590E97200DD6dfD298eE64738437222E1BCc2,
    0x9996150493Fd4E640219304998ED781a7b744785,
    0xDCF092459F2c558B1873C5C245f07F25b62ADa69,
    0xE7abfd127CFbEfA31ba6b37904B70D59037108f0,
    0xf9Ac535B6192D33DD5BCB4d088e1C11DeA9B7c3B},
```

```
0xDC00E700B1709a9B7a0032f3c84373C2DE00e620:
    {0x67C3A09BeD3389072ca85aa64A9b805897a940e1,
    0x761590734251aC5084a06e61bF5AF6c9eCD88C0f}
```

Listing A.2: Searchers Polygon smart contract address to their externally owned accounts

```
0x6131B5fae19EA4f9D964eAc0408E4408b66337b5:
    {0xd835A69Fed3f06DCE244c2cECA11AAaBD831892c},
0x643770E279d5D0733F21d6DC03A8efbABf3255B4:
    {0xa9cE598b9286ACECF2E495D663FaA611256910F1},
0x6EeeAFa18f7e764234AfFBe2A29c24Be76184F46:
    {0x121EDAD46B92B9388f721a9B29aFBB4721C3116c,
    0x2a0489d7b60097e6fAc5C2b91ec946D39562A12c,
    0x30F0e667Dd87ac2Bf83CB86110FBc20709387048,
    0x3E97184d6cA0F505033aE11c0aB6a3ee1b43C9bb,
    0x6eA16D099D1c0C2A56B0CF2b687d48F413988F39,
    0x7e5d96B086B3A1e171CFBbb38d645a3C41a195e4,
    0xE1C58Fcb072dE6b083bf5a40C2d87c00735Eca0e,
    0xd124Ee6A81A41741Fc3bb71dE1d353cde007D41B},
0x73F0764Af97a09A7d6A6f903325859d9598C646D:
    {0xcEd2A0e1C3d5B96D8cbeAd377b897f050b980574},
0x826A4f4DA02588737d3c27325B14F39b5151CA3C:
    {0x072dD4690457b28B967A459cc40Cb1d1a8e685D8,
    0x09Fa26545950b02CB083D941d1E189980504A5CF,
    0x2C61d22Af7B615D7D41def680b6EdAC29076709d,
    0x31e1a9079549FF110F5E96cdaD5100474612995D,
    0x41C2410c1b0D3511f8568927049c6f0bD9F55AC9,
    0x51c7872E970a94e4C6092B8D13Be7782CEdDcde5,
    0x63c51Ad3bD59Fed45b02007C41FdC661dd57c627,
    0x780520102FE534fb813850FAc10903834EA9fDd7,
    0x83ECfd72514133D21f91DDcbAb70C6175E8EFA88,
    0x8518c521eba4e725F227a930014a997Bf613D5C5,
    0x8Ad23DA652fB15c0c805f6825173D533Ca94828b,
    0x8DB91A1571B098e0cEdBA3E0148AE4E5FEF3622d,
    0xE83F75907Fb4c575414FA6F5cfe8cef24Dc5870C,
    0xb8dC07c0E8d60Ade09994E9cBb8CE8Ac46e35c96},
0x882d04C3D8410dDF2061B3cBA2c3522854316FEB:
    {0x08D9894495320f38561b35B31c9EcE9a4dC5712A,
    0x38d07d33B3F10C4C807C154EfA320536EBC8C505,
    0x467B79AAfD7977F6d1E772e0b121047AC655C389,
    0x65D5e6D30B00c3Dbf49B426afc81B8bB7dbaE8AB,
    0x695590E97200DD6dfD298eE64738437222E1BCc2,
    0x83Cb959A5BdE92A2dcA3313C3A5F36ED165Ff62B,
    0x9996150493Fd4E640219304998ED781a7b744785,
    0xDCF092459F2c558B1873C5C245f07F25b62ADa69,
    0xE7abfd127CFbEfA31ba6b37904B70D59037108f0,
    0xf9Ac535B6192D33DD5BCB4d088e1C11DeA9B7c3B},
0x8B6C4614b1F929360C64C05a2a1eAaE8b58e390d:
```

```
    {0xcEd2A0e1C3d5B96D8cbeAd377b897f050b980574},
0xDC00E700B1709a9B7a0032f3c84373C2DE00e620:
    {0x67C3A09BeD3389072ca85aa64A9b805897a940e1,
    0x761590734251aC5084a06e61bF5AF6c9eCD88C0f},
0xDef1C0ded9bec7F1a1670819833240f027b25EfF:
    {0x0902fBd41516bb64CFb93D514D7E40A0C4E47919},
0xad3b67BCA8935Cb510C8D18bD45F0b94F54A968f:
    {0x49BCbc58848AC642B48f3D824D8b728a1045D08b},
0xcaAF6e0977f3d47843eE4e289170D029B9f42341:
    {0x0a6c69327d517568E6308F1E1CD2fD2B2b3cd4BF},
0xec7BE89e9d109e7e3Fec59c222CF297125FEFda2:
    {0x34afdddDE003bB2d5A45151Cfa037a32ED804173}
```

Listing A.3: Searchers Ethereum smart contract address to their Polygon smart contract addresses

```
0x1231DEB6f5749EF6cE6943a275A1D3E7486F4EaE:
    {0xDef1C0ded9bec7F1a1670819833240f027b25EfF},
0x1646A4761aA54f23d7F4d5deB5D393F67D318B80:
    {0x6131B5fae19EA4f9D964eAc0408E4408b66337b5,
    0x643770E279d5D0733F21d6DC03A8efbABf3255B4,
    0xad3b67BCA8935Cb510C8D18bD45F0b94F54A968f,
    0xec7BE89e9d109e7e3Fec59c222CF297125FEFda2},
0x369350fE4421F16dBcfE28CB3943e7421BA48561:
    {0x73F0764Af97a09A7d6A6f903325859d9598C646D,
    0x8B6C4614b1F929360C64C05a2a1eAaE8b58e390d},
0x3fA6Fff7212D3fA4317cF1955fa690993d8ceD70:
    {0xcaAF6e0977f3d47843eE4e289170D029B9f42341},
0x6EeeAFa18f7e764234AfFBe2A29c24Be76184F46:
    {0x6EeeAFa18f7e764234AfFBe2A29c24Be76184F46},
0x826A4f4DA02588737d3c27325B14F39b5151CA3C:
    {0x826A4f4DA02588737d3c27325B14F39b5151CA3C},
0x882d04C3D8410dDF2061B3cBA2c3522854316FEB:
    {0x882d04C3D8410dDF2061B3cBA2c3522854316FEB},
0xDC00E700B1709a9B7a0032f3c84373C2DE00e620:
    {0xDC00E700B1709a9B7a0032f3c84373C2DE00e620}
```

List of Figures

List of Tables

Bibliography

[1] V. Gramlich, M. Principato, B. Schellinger, J. Sedlmeir, J. Amend, J. Stramm, T. Zwede, J. Strüker, and N. Urbach. "Decentralized Finance (DeFi) Foundations, Applications, Potentials, and Challenges". In: *SSRN Electronic Journal* (July 2022). DOI: 10.2139/ssrn.4535868.

[2] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels. "Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges". In: (2019). eprint: arXiv:1904.05234.

[3] K. Qin, L. Zhou, and A. Gervais. "Quantifying Blockchain Extractable Value: How dark is the forest?" In: May 2022, pp. 198–214. DOI: 10.1109/SP46214.2022.9833734.

[4] D. Robinson and G. Konstantopoulos. *Ethereum is a Dark Forest*. Aug. 2020. URL: https://www.paradigm.xyz/2020/08/ethereum-is-a-dark-forest.

[5] *The Power Surge of Cross-Chain Interoperability in Blockchain's Future*. URL: https://learn.scallopx.com/guides/crypto/intermediate/the-power-surge-of-cross-chain-interoperability-in-blockchains-future.

[6] Flashbots. *Maximal extractable value inspector for Ethereum, to illuminate the dark forest*. https://github.com/flashbots/mev-inspect-py.

[7] *On Client-Extractable Value*. URL: https://awmacpherson.com/posts/mev-def/.

[8] *What Is Cross-Chain DeFi?* URL: https://chain.link/education-hub/cross-chain-defi#:~:text=Cross%2Dchain%20lending%20allows%20users,within%20that%20on%2Dchain%20environment.

[9] S. Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: (May 2009). URL: http://www.bitcoin.org/bitcoin.pdf.

[10] L. Lamport, R. Shostak, and M. Pease. "The Byzantine generals problem". In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 4.3 (1982), pp. 382–401.

[11] J. Yu, R. Liu, X. Jiang, and J. Ren. "A Survey on Consensus Mechanisms and Mining Algorithms for Blockchain Networks". In: *IEEE Access* (2019). DOI: 10.1109/ACCESS.2019.2922669.

[12] V. Buterin. "Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform". In: (2013). URL: https://github.com/ethereum/wiki/wiki/White-Paper.

[13] E. Foundation. *NODES AND CLIENTS*. URL: https://ethereum.org/en/developers/docs/nodes-and-clients/.

[14] E. Foundation. *GAS AND FEES*. URL: https://ethereum.org/en/developers/docs/gas/.

[15] T. Wahrstätter. *On Block Sizes, Gas Limits and Scalability*. URL: https://ethresear.ch/t/on-block-sizes-gas-limits-and-scalability/18444.

[16] E. Foundation. *ACCOUNTS*. URL: https://ethereum.org/en/developers/docs/accounts/.

[17] E. Foundation. *TRANSACTIONS*. URL: https://ethereum.org/en/developers/docs/transactions/.

[18] E. Foundation. *BLOCKS*. URL: https://ethereum.org/en/developers/docs/blocks/.

[19] E. Foundation. *ETHEREUM VIRTUAL MACHINE (EVM)*. URL: https://ethereum.org/en/developers/docs/evm/.

[20] E. Foundation. *INTRODUCTION TO SMART CONTRACTS*. URL: https://ethereum.org/en/developers/docs/smart-contracts/.

[21] J. Xie, F. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu. "A Survey on the Scalability of Blockchain Systems". In: *IEEE Network* PP (Sept. 2019), pp. 1–8. DOI: 10.1109/MNET.001.1800290.

[22] A. A. Jaynti Kanani Sandeep Nailwal. "Matic Whitepaper". In: (2018). URL: https://github.com/maticnetwork/whitepaper.

[23] *Polygon Solutions*. URL: https://polygon.technology/blog-category/polygon-solutions.

[24] F. S. Mishkin and S. G. Eakins. *Financial Markets and Institutions*. 7th. The Prentice Hall Series in Finance. Pearson, 2012.

[25] Bitcoin.com. *What is a CEX?* URL: https://www.bitcoin.com/get-started/what-is-a-cex/.

[26] J. Aoyagi and Y. Ito. "Coexisting Exchange Platforms: Limit Order Books and Automated Market Makers". In: (Mar. 2021). DOI: 10.2139/ssrn.3808755. URL: https://ssrn.com/abstract=3808755.

[27] DeFiLlama. *Dexs volume*. URL: https://defillama.com/dexs.

[28] G. Damalas and P. Ambrus. *An introduction to maximal extractable value on Ethereum*. Mar. 2023. URL: https://assets.ey.com/content/dam/ey-sites/ey-com/en_us/topics/financial-services/ey-an-introduction-to-maximal-extractable-value-on-ethereum.pdf.

[29] C. Torres, R. Camino, and R. State. "Frontrunner Jones and the Raiders of the Dark Forest: An Empirical Study of Frontrunning on the Ethereum Blockchain". In: (Feb. 2021).

[30] A. Bagourd and L. Francois. "Quantifying MEV On Layer 2 Networks". In: (Aug. 2023).

[31] B. Öz, F. Rezabek, J. Gebele, F. Hoops, and F. Matthes. *A Study of MEV Extraction Techniques on a First-Come-First-Served Blockchain*. 2024. arXiv: 2308.06513 [cs.CR].

[32] A. Obadia, A. Salles, L. Sankar, T. Chitra, V. Chellani, and P. Daian. *Unity is Strength: A Formalization of Cross-Domain Maximal Extractable Value*. 2021. arXiv: 2112.01472 [cs.CR].

[33] W. Xiao. *searcherbuilder.pics*. URL: https://www.searcherbuilder.pics/.

[34] A. Chiplunkar and S. Gosselin. *A new game in town*. Feb. 2023. URL: https://frontier.tech/a-new-game-in-town.

[35] C. McMenamin. *SoK: Cross-Domain MEV*. 2023. arXiv: 2308.04159 [cs.CR].

[36] E. Chen, A. Toberoff, S. Srinivasan, and A. Chiplunkar. *A Tale of Two Arbitrages*. June 2023. URL: https://frontier.tech/a-tale-of-two-arbitrages.

[37] L. Heimbach, V. Pahari, and E. Schertenleib. *Non-Atomic Arbitrage in Decentralized Finance*. 2024. arXiv: 2401.01622 [cs.CE].

[38] J. Barragan. *The Fundamentals of Cross-Chain MEV*. URL: https://www.blocknative.com/blog/fundamentals-of-cross-chain-mev.

[39] J. H. Sjursen, W. Meng, and W.-Y. Chiu. "Towards Quantifying Cross-Domain Maximal Extractable Value for Blockchain Decentralisation". In: *Information and Communications Security*. Ed. by D. Wang, M. Yung, Z. Liu, and X. Chen. Singapore: Springer Nature Singapore, 2023, pp. 627–644. ISBN: 978-981-99-7356-9.

[40] O. Mazor and O. Rottenstreich. *An Empirical Study of Cross-chain Arbitrage in Decentralized Exchanges*. Cryptology ePrint Archive, Paper 2023/1898. https://eprint.iacr.org/2023/1898. 2023. URL: https://eprint.iacr.org/2023/1898.

[41] R. Belchior, J. SüSSenguth, Q. Feng, T. Hardjono, A. Vasconcelos, and M. Correia. "A Brief History of Blockchain Interoperability". In: (June 2023). DOI: 10.36227/techrxiv.23418677.

[42] A. Zamyatin, M. Al-Bassam, D. Zindros, E. Kokoris-Kogias, P. Moreno-Sanchez, A. Kiayias, and W. Knottenbelt. "SoK: Communication Across Distributed Ledgers". In: Oct. 2021, pp. 3–36. ISBN: 978-3-662-64330-3. DOI: 10.1007/978-3-662-64331-0_1.

[43] V. Buterin. *Chain Interoperability*. Sept. 2016. URL: https://allquantor.at/blockchainbib/pdf/vitalik2016chain.pdf.

[44] H. Pandey. *Polygon: An Overview of the Leading Layer 2 Scaling Solution on Ethereum*. URL: https://medium.com/coinmonks/polygon-an-overview-of-the-leading-layer-2-scaling-solution-on-ethereum-9b81d3c088.

[45] P. team. *Polygon Knowledge Layer - Polygon PoS*. URL: https://docs.polygon.technology/pos/.

[46] Pmcgoohan. *Zeromev API*. URL: https://data.zeromev.org/docs/.

[47] *Findblock*. URL: https://www.findblock.xyz/.

[48] *Mapped Tokens*. URL: https://api-polygon-tokens.polygon.technology/tokenlists/polygon.tokenlist.json.