SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

# Detection and Analysis of Cross-Chain Arbitrages Between Ethereum and Polygon

**Murad Muradli**

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

# Detection and Analysis of Cross-Chain Arbitrages Between Ethereum and Polygon

# Erkennung und Analyse von Cross-Chain-Arbitragen zwischen Ethereum und Polygon

| | |
|---|---|
| Author: | Murad Muradli |
| Supervisor: | Prof. Dr. Florian Matthes |
| Advisor: | Burak Öz, M.Sc. |
| Submission Date: | 03.12.2024 |

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

Munich, 03.12.2024 Murad Muradli

# AI Assistant Usage Disclosure

## Introduction

Performing work or conducting research at the Chair of Software Engineering for Business Information Systems (sebis) at TUM often entails dynamic and multi-faceted tasks. At sebis, we promote the responsible use of *AI Assistants* in the effective and efficient completion of such work. However, in the spirit of ethical and transparent research, we require all student researchers working with sebis to disclose their usage of such assistants.

For examples of correct and incorrect AI Assistant usage, please refer to the original, unabridged version of this form, located at this link.

## Use of *AI Assistants* for Research Purposes

**I have used AI Assistant(s) for the purposes of my research as part of this thesis.**

Yes ◉ No ○

**Explanation:** In the course of conducting research and performing tasks assigned to me, I have made use of AI tools, including Grammarly and ChatGPT, to refine sentence structure and ensure that my ideas are articulated clearly. However, all written content, including the ideas, and code presented in this work, are entirely my own. The use of AI tools was limited to improving clarity and coherence and did not contribute to the originality or intellectual ownership of the content.

I confirm in signing below, that I have reported all usage of AI Assistants for my research, and that the report is truthful and complete.

Munich, 03.12.2024 Murad Muradli

# Acknowledgments

I would like to thank my thesis advisor, Burak Öz, for his guidance and support. His organization of twice-weekly meetings and his patience in answering my questions greatly helped me stay on track and make consistent progress.

I would also like to thank Filip Rezabek, who was my external advisor, for his help in setting up and running the pipeline, which was essential for my work. His assistance made the technical aspects of my thesis much more manageable.

My thanks go to Danut Ilisei, who explained the codebase to me at the start of my thesis and helped me understand how everything was interconnected. He also directed me to useful resources and provided support during the initial stages of my research.

Finally, I am grateful to my family for their support throughout this process, especially my brother, Turqut Muradlı, for his constant mental support and encouragement.

# Abstract

Cross-chain arbitrage has gained increasing attention within DeFi due to the growing number of Ethereum Virtual Machine (EVM)-compatible blockchains and the opportunities they present for profit extraction. While Maximum Extractable Value (MEV) strategies on single chains have been well studied, cross-chain arbitrage remains underexplored. This thesis seeks to address this gap by investigating cross-chain arbitrage between the Ethereum and Polygon networks, focusing on the unique challenges and strategies involved in these transactions.

Through a year-long study, we develop a methodology to systematically identify and analyze historical cross-chain arbitrages between the two networks. We manage to detect 23,404 instances of successful arbitrage which constitute 0.89% of the examined blocks.

Our findings highlight how network protocols, such as Polygon's checkpointing mechanism, influence the efficiency of arbitrage transactions. Specifically, we observe that Ethereum to Polygon transactions occur more frequently, primarily due to shorter bridging times, compared to the reverse direction. Additionally, we explore the roles of different transaction types, noting that while bridge transactions are nearly evenly split between public and private, swap transactions are predominantly private, suggesting strategies aimed at mitigating risks such as front-running. We also examine the key players in the cross-chain arbitrage space, identifying a small group of searchers responsible for the majority of arbitrage activity and provide a detailed analysis of the intermediary tokens used in arbitrages.

This research contributes to a deeper understanding of cross-chain arbitrage mechanics and the emerging challenges associated with it.

# Contents

# 1. Introduction

The rise of blockchain technology has revolutionized the financial sector by introducing open, decentralized networks that operate independently of traditional banks and financial institutions. Public blockchains like Bitcoin and Ethereum enable secure, peer-to-peer transactions without the need for intermediaries, leveraging cryptographic protocols and consensus mechanisms to ensure trust and transparency in every exchange [1]. A key application of blockchain technology is Decentralized Finance (DeFi), an ecosystem that provides a wide array of financial services—including lending, borrowing, and trading—directly on the blockchain. By eliminating many of the constraints and costs of traditional banking, DeFi makes financial services more accessible to a global audience [2].

A critical concept within DeFi is Maximal Extractable Value (MEV), which represents the potential profits that network participants—such as miners, validators, or algorithmic bots—can capture by reordering, inserting, or censoring transactions within a block before it is confirmed on the blockchain [3]. MEV is often extracted through techniques like front-running, where an actor places their transaction just before a large pending trade to benefit from anticipated price movements, or back-running, where trades are executed immediately after another transaction to capture profits from resulting price changes [4].

MEV opportunities arise largely due to the the transparency of public mempools, where pending transactions are visible to all participants prior to confirmation. This transparency enables certain actors to monitor the mempool, anticipate others' trades, and strategically adjust their own transactions to exploit predicted price shifts or other market dynamics. While MEV can generate profits for some, it also poses risks to market fairness and efficiency, as it can lead to increased transaction costs and volatile market conditions [4].

One of the most lucrative MEV extraction strategies is arbitrage, which seeks to profit from price discrepancies across different markets or platforms. The concept of arbitrage is simple: it involves simultaneously buying and selling an asset on different exchanges to capitalize on price differences. For instance, if a particular token is priced lower on one Decentralized Exchange (DEX) than on another, an arbitrageur can purchase the token on the cheaper exchange and sell it on the more expensive one, securing a profit from the price differential. This process not only generates profits for the arbitrageur but also contributes to price alignment across markets, contributing to the overall market efficiency [5].

In recent years, the emergence of more blockchain networks offering DeFi platforms, particularly those compatible with the Ethereum Virtual Machine (EVM), has introduced both new complexities and opportunities within the DeFi space. These networks include sidechains like Polygon and rollups like Arbitrum and Optimism. Polygon, a Layer 2 scaling solution for Ethereum, addresses Ethereum's scalability challenges through its PoS (Proof-of-Stake) chain. This high-speed, low-cost sidechain is fully compatible with Ethereum, enabling the

development of more sophisticated arbitrage strategies [6].

While single-chain arbitrage—or atomic arbitrage—occurs entirely within one blockchain and can be resolved in a single transaction, cross-chain arbitrage spans multiple blockchains, such as Ethereum and Polygon. Cross-chain arbitrage typically involves non-atomic operations, requiring separate transactions for bridging assets and executing swaps across chains [5]. These transactions introduce additional challenges, including delays and increased risk of failure. However, they also open up opportunities to exploit price discrepancies across chains, providing a new dimension to MEV extraction.

## 1.1. Motivation and the Scope of Our Thesis

MEV has been widely studied within the context of a single blockchain. However, cross-chain arbitrage, which involves leveraging price differences across multiple blockchain networks, remains a relatively understudied and evolving field of research. This thesis aims to contribute to this growing body of research by exploring executed cross-chain, non-atomic arbitrages between Ethereum and Polygon. To address these challenges, our study will analyze historical data from both blockchain networks to assess key factors influencing cross-chain arbitrage. These factors include transaction costs, execution times, and the frequency of arbitrage events between the platforms. Additionally, we will evaluate the profitability of each identified arbitrage, considering network fees, bridge costs, and potential risks associated with MEV, such as frontrunning or sandwich attacks by block producers. This thesis aims to shed light on the economic feasibility and practical challenges of cross-chain arbitrage, contributing to future research and the improvement of strategies in DeFi.

## Research Questions

The research questions addressed in our thesis are as follows:

- **RQ1:** *What is the state-of-the-art literature on cross-domain profiting strategies in the context of blockchains?*
  This question surveys existing research on multi-chain arbitrages and profit-making strategies, identifying current methodologies, trends, and knowledge gaps.
- **RQ2:** *How can we develop a methodology to detect cyclic arbitrages between Ethereum and Polygon PoS using the Polygon bridge?*
  - **RQ2.1:** *How can we devise heuristics to identify both successful and unsuccessful arbitrages?*
    This sub-question explores strategies to distinguish between profitable arbitrages and failed attempts by analyzing transaction patterns, gas fees, and bridging delays.
- **RQ3:** *Who executes cross-chain arbitrages between Ethereum and Polygon, and how frequent and profitable are these arbitrages?*

This question investigates the searchers involved in cross-chain arbitrages, their strategies, and the characteristics of these arbitrages. Key aspects include:

  - **Searchers:**
    - **RQ3.1:** *Are the searchers using externally owned accounts (EOAs) or smart contracts?*
    - **RQ3.2:** *How do searchers compare in terms of the frequency of arbitrages and profitability?*
  - **Arbitrages:**
    - **RQ3.3:** *How often do arbitrages occur between Ethereum and Polygon, and how profitable are they?*
    - **RQ3.4:** *Which tokens are most commonly used for profiting?*
    - **RQ3.5:** *How long are the bridging times for these arbitrages?*
    - **RQ3.6:** *Is the Ethereum leg of the arbitrage submitted to the public mempool?*

- **RQ4:** *How do our obtained results compare to other profit-making strategies observed on blockchains?*
  This question examines cross-chain non-atomic arbitrages within the broader landscape of blockchain strategies, comparing their occurrence and profitability to methods like atomic arbitrages, CEX-DEX arbitrages, and sandwich attacks.

## 1.2. Thesis Structure

We begin with an introduction to the key concepts and terminology in the background section (Chapter 2). Chapter 3 provides a comprehensive overview of recent and relevant literature, followed by a detailed discussion of our methodology in Chapter 4. Our findings are presented in Chapter 5, with our thesis concluding with a discussion in Chapter 6 and final remarks in Chapter 7.

# 2. Background

## 2.1. Blockchains

A blockchain is a distributed ledger shared across a network of computers. The participants in this network are referred to as nodes. It consists of a chain of immutable blocks, where each block contains a cryptographic hash of the previous block, a timestamp, and a list of verified transactions. This configuration makes the blockchain tamper-resistant, as trying to alter a block would also require changing all subsequent blocks as well [1].

As additional blocks are appended, a blockchain employs consensus mechanisms to ensure that all participating nodes agree on the current state of the chain. The two most widely used consensus mechanisms used in blockchains are Proof-of-Work and Proof-of-Stake. The Proof-of-Work algorithm requires miners to solve complex cryptographic puzzles, to gain the right to mine or create a new block. The miner who successfully mines a new block is rewarded with cryptocurrency [7]. In contrast, Proof-of-Stake offers a more energy-efficient alternative to Proof-of-Work, allowing nodes to stake a portion of their cryptocurrency to participate in the network as validators. The probability of a node being selected as a validator is proportional to the amount of cryptocurrency it is willing to stake [8].

There are different types of blockchains, each catering to specific use cases and privacy requirements. Public blockchains (e.g., Bitcoin and Ethereum) are accessible to anyone, allowing any participant to join the network and access the ledger. Such networks typically have a high degree of decentralization due to the large number of participating nodes. In contrast, private blockchains are restricted to a limited group of participants and are often used by private enterprises for internal record-keeping. In a private blockchain, all network participants are pre-selected and authorized. As such, private blockchains offer much higher transaction speeds and a greater degree of privacy [9].

## 2.2. Ethereum

### Overview

Launched in 2015 by Vitalik Buterin, Ethereum is a public, decentralized blockchain for building and deploying smart contracts and decentralized applications (DApps). Ethereum allows developers to write and execute code in the form of smart contracts. Network participants use Ether (ETH), Ethereum's native cryptocurrency, to pay for transaction fees and computational services on the network [10].

### The Ethereum Virtual Machine (EVM)

Ethereum Virtual Machine (EVM) is the most critical component of the Ethereum infrastucture. It is a runtime environment for executing smart contracts and decentralized applications (dApps). It allows developers to write applications in high-level languages such as Solidity and deploy them on the blockchain. For this, developers compile their smart contracts into bytecode which is then executed by the EVM. One important feature of EVM is that it ensures deterministic execution. This means that for the same input and initial state, nodes will generate the same output when processing transactions and smart contracts. This is important for ensuring consensus on the state of the blockchain. Another key feature is Turing completeness, allowing the EVM to execute any computation that can be defined algorithmically. This distinguishes Ethereum significantly from simpler blockchains, which lack such computational versatility [11].

### Mempool

The mempool in Ethereum, short for "memory pool," stores pending (unconfirmed) transactions before they are included in a block on the blockchain. When a user submits a transaction, it is first broadcast to the network and temporarily held in the mempool of each node. Here, transactions wait to be picked up and processed by validators. When selecting transactions to be included in a block, validators usually prioritize them by their gas prices. Validators then select these transactions, add them to a block and propagate the block throughout the network. When a transaction is included in a block, it is removed from the mempool to become part of the permanent blockchain [12].

### Smart Contracts

A smart contract, which is the cornerstone of Ethereum's ecosystem, is a program that runs on the Ethereum blockchain. Smart contract deployment is permissionless, which means any network participant can use simple languages such as Viper or Solidity to create smart contracts, which can be deployed given that the user has sufficient amount of funds. After smart contracts are deployed, they are immutable. Smart contracts have a wide array of use cases, ranging from Decentralized Finance (DeFi) and Insurance to Gaming and NFTs. Each smart contract has an address. When a node or another smart contract wants to call a method defined in the smart contract, they send a message to this address [13].

### Gas fees

Gas fees on Ethereum are the cost needed to issue transactions and execute smart contracts. These fees are denominated in 'gwei,' a unit of Ether (ETH) equal to one billionth of an ETH (1 gwei = $10^{-9}$ ETH), and are paid by users to incentivize validators to include their transactions in a block. The amount of gas required for a transaction depends on the complexity of the operation being performed. By making users pay a certain amount of gas for developing and running smart contracts, the system incentivizes them to create smart contracts that actually

serve their needs, rather than wasting resources on unnecessary or overly complex contracts [14].

Before EIP-1559 [15], Ethereum worked like a basic auction where users would choose and submit a gas price (bid) for their transactions. Miners would then prioritize transactions with the highest bids. This setup had many issues such as users not bidding honestly, overbidding, and resulting in unpredictable gas price fluctuations which made it difficult for users to predict transaction fees accurately. EIP-1559 was introduced to fix these problems by changing how gas fees are handled. One of its key features is a base fee, which is automatically adjusted by the network based on current demand and congestion. This means the base fee rises and falls depending on how busy the network is, helping to stabilize gas prices [14].

### Accounts

In Ethereum, accounts are entities that manage ETH and facilitate interactions within the blockchain. Two primary types of accounts are: Externally Owned Accounts (EOAs) and Contract Accounts. Although both types of accounts can hold, send and receive Ether, as well as interact with smart contracts, they serve different purposes. The main difference is that EOAs don't store state or data, and are controlled by users through their private keys. Smart Contract Accounts (SCA), on the other hand, represent the smart contracts that have been deployed on the network. As such SCAs have associated code, and can be trigerred by messages and transactions. SCAs don't have private keys, and cannot issue transactions, but they can be triggered by incoming transactions to call upon other smart contracts [16].

### Transactions

Ethereum transactions are actions facilitated by EOAs, i.e., user-controlled accounts. They can be described as cryptographically signed messages that contains the instructions of a user. Users can transfer Ethereum-based assets from one account to another by making use of transactions. Once transactions are issued, and executed on the EVM, they are broadcast throughout the network, where they are picked up by validators. In Ethereum, transaction propagation occurs through two mechanisms: transactions can either be submitted to the public mempool, where they are broadcast to every node on the network, or they can be privately submitted directly to validators (or to builders in MEV-Boost [17], an implementation of Proposer-Builder Separation (PBS) [18]). This bypasses the public mempool, enhancing privacy and reducing risks like front-running or MEV exploitation. Validators then verify transaction details, and if they are correct, they include the transactions in a block, which is then added to the blockchain, ensuring they become part of a permanent, immutable ledger. Users also pay gas fees for each transaction, which is meant to compensate validators for the computational effort required to process and verify the transactions [19].

### Blocks

In Ethereum, blocks are collections of transactions linked in a chain, each referencing the previous one through a cryptographic hash. This structure ensures the blockchain's integrity, making it impossible to alter any block without changing all subsequent blocks. Transactions within blocks are ordered and added sequentially, ensuring that all participants agree on the transaction history. Blocks are created by randomly selected validators in Ethereum's PoS system, which executes transactions and updates the global state. These blocks are then verified by other validators before being added to their copies of the blockchain. Block creation and validation maintain a decentralized, secure network, with new blocks proposed every 12 seconds. Block sizes are indirectly limited by the gas limit of 30 million, which constrains the computational resources needed to process transactions, helping to maintain network performance and prevent centralization [20].

#### Ethereum Tokens and ERC-20

Tokens are digital assets built on top of existing blockchains, typically created to support specific projects. Unlike coins, such as Bitcoin (BTC) or Ether (ETH), which function as native currencies on their own blockchains, tokens are often created for particular purposes, such as representing ownership or access rights. Tokens are often distributed during events like ICOs to raise funds from investors. They can be traded, used to access platform services, or participate in governance. Ethereum provides an infrastructure to create and manage tokens through smart contracts, which define the rules and functions of each token [21].

Ethereum employs token standards to promote interoperability across different tokens. One of the most widely used standards is ERC-20. Proposed by Fabian Vogelsteller in 2015 as an Ethereum Request for Comment (ERC), it was later approved and adopted in 2017 as an Ethereum Improvement Proposal (EIP-20) [22]. Token developers who wish for their tokens to be recognized as ERC-20 compatible must ensure their smart contract adheres to this standard. By following ERC-20, developers can guarantee seamless integration of their tokens within the broader Ethereum ecosystem without needing to revise existing projects for each new token release. Well-known ERC-20 compatible tokens on Ethereum include Wrapper Ether (WETH), Tether USD (USDT), USD Coin (USDC), and DAI Stablecoin (DAI) [22].

#### Stablecoins

Stablecoins are a type of cryptocurrency designed to maintain a stable or consistent value by being pegged to assets like the U.S. dollar, other fiat currencies, or commodities such as gold. The need for stablecoins arose from the highly volatile nature of cryptocurrency exchange rates. For example, a trader who accepts cryptocurrency as payment risks a loss if its price suddenly drops. For a non-legal tender to function as a medium of exchange, it must stay relatively stable, ensuring short-term purchasing power[23].

Stablecoins can be categorized by the types of assets used to back their value. These are as follows:

- **Fiat-Collateralized Stablecoins** - These stablecoins are backed by reserves of fiat currency, like the U.S. dollar, held in bank accounts. For every stablecoin issued, an equivalent amount of fiat currency is kept in reserve, creating a 1:1 peg. Examples include USD Coin (USDC) and Tether (USDT), the latter holding the largest market capitalization at over $112 billion [23].
- **Commodity-Backed Stablecoins** - These stablecoins are pegged to the value of physical assets, like gold, oil, or other commodities. An example is Tether Gold (XAUt), which is backed by gold reserves [23].
- **Crypto-Collateralized Stablecoins** - These stablecoins are backed by cryptocurrency reserves, rather than fiat currency or physical assets. These stablecoins are typically over-collateralized, with the cryptocurrency reserves exceeding the total value of the issued stablecoins. This excess collateral is meant to serve as a buffer against the high volatility that cryptocurrencies can experience. One well-known example is MakerDAO's Dai (DAI) stablecoin, which maintains a peg to the U.S. dollar but is supported by reserves of Ethereum (ETH) and other cryptocurrencies, valued at around 155% of the total DAI supply in circulation [23].

### Memecoins

Memecoins are cryptocurrencies that are most often created as jokes or for fun, often inspired by internet memes or viral trends. Unlike more established cryptocurrencies, memecoins usually have little to no underlying technology or use case, and their value is largely driven by community engagement.

Despite their humorous origins, some memecoins have managed to gain popularity. An example of a memecoin is Dogecoin, which originated as a parody of Bitcoin, based on the popular "Doge" meme featuring a Shiba Inu dog. As of September 1st, 2024, it had a market cap of $14.42 billion [24].

## 2.3. Layer 2

As Ethereum gained popularity, it has faced significant scalability challenges due to the large volume of transactions processed on the network. For example, in 2021, around 1.3 million transactions were processed on Ethereum everyday resulting in 35 percent annual increase [6]. The sheer volume of transactions has led to network congestion, slower transaction processing times and higher fees. One major issue, apart from the increasing number of transactions, that contributes to these problems, is the way that most Layer 1 blockchains are designed. For reference, Bitcoin can process 4.6 transactions per second (TPS), and Ethereum around 14.3 TPS, whereas Visa, one of the largest electronic payment platforms, can process up to 1,736 TPS (and has been recorded to reach up to 47,000 TPS) [25].

Layer 2 scaling solutions have emerged as a way to address these limitations by offloading the majority of transaction processing from the main blockchain (Layer 1) to a secondary layer,

or Layer 2. These solutions work by processing transactions off-chain or in a separate non-L1 chain, and only periodically settling the results back onto the main chain. This approach allows for a much higher volume of transactions to be processed more quickly and at a lower cost, while still benefiting from the security and decentralization of the Layer 1 blockchain. For example, sidechains like Polygon, rollups such as Optimistic Rollups and zk-Rollups, and state channels like the Lightning Network are all types of Layer 2 solutions that enable more efficient transaction processing [6]. By reducing the load on the main blockchain, Layer 2 scaling also improves the user experience by making transactions faster and cheaper [25].

### Polygon PoS

One example of a L2 scaling solution is Polygon PoS. Polygon PoS is a sidechain that operates alongside a main blockchain (often called the "mainchain") and is fully interoperable with it. This means that assets or tokens can be transferred between the mainchain and the sidechain. Although sidechains originate from the mainchain and operate as independent blockchains with their own tokens and consensus mechanisms, they differ from rollups in that sidechains have their own validation process, whereas rollups execute transactions off-chain but rely on the main Ethereum chain for security and finality, submitting aggregated data back to the mainchain for validation [26]. One of the key features of Polygon PoS is its compatibility with the EVM, which means that any smart contract or dApp that runs on Ethereum can be easily deployed on Polygon without the need for significant code modifications. Polygon has its own cryptocurrency, known as MATIC. This token is used to pay transaction fees on the Polygon network, participate in staking, and engage in governance, allowing MATIC holders to vote on changes to the Polygon protocol [27].

Polygon PoS uses a Proof-of-Stake (PoS)-based consensus mechanism where validators stake MATIC tokens to participate in the network. Validators, chosen based on their stake, handle block production and transaction validation. They must lock up some MATIC, as collateral to assure honest behavior, and they risk losing this stake if they act maliciously. The network uses a Byzantine Fault Tolerant algorithm, called Proof-of-Stake Checkpointing, where chosen validators propose and vote on checkpoints added to Ethereum. Checkpoints refer to specific points in the blockchain's history that are periodically recorded and anchored onto the Ethereum mainnet [28].

## 2.4. Blockchain Interoperability

As blockchain technologies gain traction across various industries, the demand for systems that can exchange and verify information across separate blockchains is growing. For example, a logistics company on one blockchain might need to confirm transactions with a financial institution on another—a task that is impossible without interoperability between these networks. Blockchain interoperability in this context refers to the "the ability of blockchain networks to communicate with each other, sending and receiving messages, data, and tokens" [29].

There are many different solutions for cross-chain interoperability. One of the most common approaches to achieving this is through cross-chain bridges, which act as connectors for token and data transfers between two distinct blockchains. There are two main types of bridges: trusted and trustless. A trusted bridge relies on a central authority or a specific group of validators to oversee and confirm transactions. This centralized control can make transactions faster and provide a smoother user experience. However, it introduces a potential vulnerability because the security of the system depends on the integrity and proper functioning of the central authority. If the authority overseeing the bridge is compromised, the assets being transferred may be at risk. This is referred to as custodial risk. Trusted bridges are typically more efficient but come with the trade-off of centralization and trust dependency. A trustless bridge, on the other hand, relies on smart contracts and decentralized consensus mechanisms to operate without a central authority. Transactions are verified automatically by code, and the security of the bridge is rooted in the underlying blockchain networks' consensus algorithms. However, trustless bridges may be slower or more complex to use, as they rely on decentralized verification processes [29].

#### Polygon PoS Bridge

Within the context of L2 solutions, a bridge connecting Ethereum to Polygon serves as a crucial component for enabling interoperability between the two networks. The bridging protocol responsible for this connection between Ethereum and Polygon is the Polygon PoS Bridge. The Polygon PoS Bridge is a "trusted" bridge, meaning it relies on external validators and security mechanisms to enable secure transactions between Ethereum and Polygon [30]. When assets are transferred from Ethereum to Polygon, they are locked in an Ethereum smart contract, while corresponding tokens are minted on Polygon. For withdrawals, tokens on Polygon are burned, and the original tokens are released on Ethereum. In addition to the PoS Bridge, Polygon also provides the Plasma Bridge, which employs a different, more secure mechanism for specific token types but features a slower withdrawal process (with a seven-day waiting period) to provide additional security [30]. This bridge model ensures that only assets on Ethereum are secured by a fraud-proof mechanism, rather than relying solely on validator trust. The PoS Bridge design prioritizes speed and low fees, making it ideal for frequent transactions at a lower cost, though it carries some reliance on Polygon's validator network rather than a fully decentralized system [30].

## 2.5. Decentralized Finance

DeFi refers to financial services that operate on public blockchains. DeFi allows users to make use of these financial services and engage in lending, borrowing, trading and investment of their assets, without having to rely on a third-party. These financial services are smart contracts [2].

### Centralized Exchanges

Centralized Exchanges (often referred to as CEX) act as intermediaries between cryptocurrency buyers and sellers. In the blockchain world, centralized exchanges (CEXs) resemble traditional finance by acting as trusted intermediaries, similar to how banks and payment processors operate. CEXes are owned and managed by private companies and, thus, are bound by the laws and regulations of the jurisdictions in which the companies are based. For example, many CEXes require "Know-your-Customer/Anti-Money Laundering (KYC/AML) ID verification" before a user can trade. As such, users must register and create accounts before using the services provided by CEXes [31]. CEXes are often regarded as "entry points to the crypto ecosystem", since many of them enable payments with fiat currencies allowing non-crypto holders to purchase digital assets using USD, EUR, and other currencies. And the KYC requirements are helpful in reducing the number of malicious actors [32].

Centralized exchanges (CEXes) use order books to manage trades. An order book is a digital list that shows all current buy and sell orders for an asset, arranged by price. When a trader places an order on a CEX, it is added to the order book where it is to be matched with a corresponding order from another trader. The order book displays the highest price someone is willing to pay (bids) and the lowest price someone is willing to accept (asks). This setup helps match buyers and sellers, making it easy for traders to see prices and complete trades quickly [33].

### Decentralized Exchanges

A decentralized exchange (or DEX) is a peer-to-peer platform where users can trade cryptocurrencies directly with each other without the need for an intermediary. Instead of having a central authority to manage the trades, DEXs make use of smart contracts for automatic transaction handling. This means that trades occur directly between users on-chain, and the rules governing the exchange are enforced by the smart contracts instead of human operators [32]. Most DEXs focus on trading cryptocurrencies against each other. To trade using fiat currency, users usually need to first convert their fiat into cryptocurrency through a CEX [34]. A famous example for a DEX is UniSwap [35]. Uniswap (launched in November 2018), currently has a liquidity balance of over 3 billion USD in different cryptocurrencies and enables transactions worth over 700 million USD every day [36].

## 2.6. Maximum Extractable Value (MEV)

MEV is a concept that first entered the blockchain and cryptocurrency lexicon in 2019, through research conducted by Daian et al [3]. MEV describes the additional profit miners can gain by strategically reordering, including, or excluding transactions within the blocks they produce. This is made possible by the fact that block producers (e.g., miners or validators) have the power to choose which transactions to include and how to order them. This introduces a broad range of strategies for block creators to extract additional profit, often to the users' detriment. Common MEV strategies include front-running, where transactions are placed

ahead of others to exploit price movements, and sandwich attacks, where transactions are positioned before and after a target transaction to manipulate the outcome. In theory, miners and validators should benefit the most from MEV since they control transaction execution. However, in practice, it's mostly done by independent participants called 'searchers,' who use advanced algorithms to scan the blockchain for MEV opportunities and then deploy bots to automatically create profitable transactions [37]. While miners and validators have the power to execute transactions and order them within blocks, they are typically less incentivized to actively search for these opportunities, which is where searchers step in. For the most common types of MEV, searchers pay high gas fees to incentivize block producers to include their transactions in specific blocks. These gas fees act as incentives, giving validators a share of the MEV [37].

People who take advantage of MEV can make substantial profits, but this practice raises serious concerns about the fairness and integrity of blockchain networks. Regular users often bear the brunt, facing higher fees, or even failed transactions because of these tactics. As a result, the blockchain community has proposed several solutions to tackle the negative impacts of MEV extraction. These include fair ordering protocols, private transaction pools, decentralized sequencers and tools like Mev-Boost [17] provided by Flashbots to make MEV access more equal, and creating MEV auctions to sell transaction ordering rights transparently [38].

### Proposer-Builder Separation

Proposer-Builder Separation (PBS) addresses the issue of centralization in Ethereum caused by MEV extraction [39]. PBS separates the roles of block proposers and block builders to ensure that validators are not directly involved in constructing blocks optimized for MEV. If only a few validators possess the ability to build these MEV-maximizing blocks, it could lead to centralization of block production. To mitigate this, PBS introduces builders, who are responsible for constructing blocks. When it's a validator's turn to propose a block, they can choose to sell their block construction rights to a builder, receiving payment based on the MEV extracted from the block. This system helps prevent centralization by allowing any validator to participate in the MEV extraction process, rather than just those with the ability to construct blocks. Since this process requires a trustless and fair exchange between validators and builders, PBS must be integrated into Ethereum's protocol. Given the significant implications of this change, the community chose to implement it off-chain through MEV-Boost [17]. Since the Merge, MEV-Boost has been widely adopted, with approximately 90% of blocks being built through it. In other words, 9 out of 10 validators sell their block construction rights to builders [39][18].

### Private Order Flow

Private Order Flow (POF) in Ethereum refers to a mechanism where transactions bypass the public mempool and are sent directly to validators, block builders, or specialized platforms for execution. This process addresses challenges associated with the public mempool,

such as front-running, sandwich attacks, and other forms of MEV exploitation. By keeping transactions private until they are confirmed in a block, users can safeguard sensitive trading strategies, and prevent losses caused by malicious actors that exploit visible pending transactions [40].

Several tools and platforms facilitate private order flows. Private RPC endpoints, such as Flashbots Protect [41], offer confidentiality by sending transactions directly to validators instead of broadcasting them publicly. Order Flow Auctions (OFAs) allow users to directly monetize their transaction data by auctioning access to their order flow in a competitive marketplace. In this model, users submit their transactions to an auction platform, where block builders or MEV searchers bid for the right to include these transactions in a block. This way users can capture value that might otherwise be extracted by MEV bots operating in the public mempool [42]. MEV auction platforms, on the other hand, create a space where MEV searchers bid to determine the order of transactions in blocks. This setup helps block builders or validators make more profit by prioritizing transactions that generate higher returns. The focus of these platforms is to allow MEV searchers to extract value from transaction ordering [43]. While Order Flow Auctions (OFAs) let users earn money by selling their transaction data, MEV auctions aim to maximize profits for searchers.

### Searchers

Searchers are independent network participants who actively look for opportunities to extract MEV. They usually use MEV bots that or automation tools to monitor the Ethereum mempool for pending transactions. After spotting an opportunity, the searcher analyzes the trade, creates a bundle - a set of transactions grouped and ordered to achieve their MEV goal - and sends this bundle to one or multiple block builders. Searchers place bids for block inclusion by either offering higher gas prices or making direct ETH payments to the block builder's address. Once a searcher's transaction has been included in the block, this finalizes the MEV extraction process for that searcher [39][18].

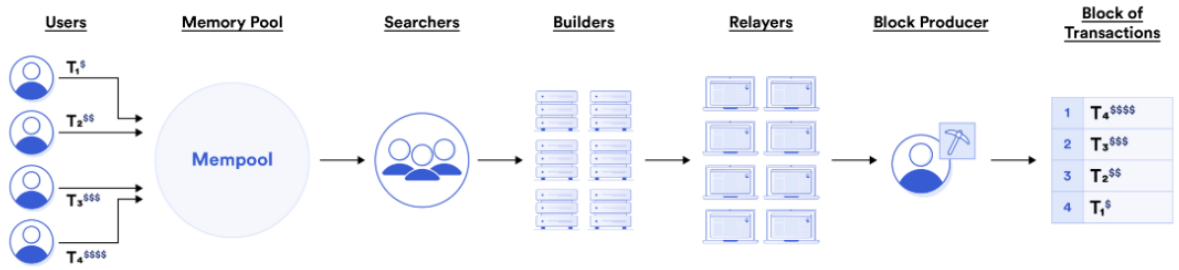


Figure 2.1.: Lifecycle of a Searcher Activity [44]

## Most Common MEV Strategies

### Sandwich Attacks

A sandwich attack is a malicious MEV strategy where an attacker leverages both frontrunning and backrunning techniques to profit at the expense of a victim's trade. The attacker begins by monitoring the mempool for a pending transaction that could impact an asset's price. Once such a transaction is identified, the attacker places two trades to "sandwich" the victim's transaction: a frontrunning trade and a backrunning trade [45].

In the first step (frontrunning), the attacker submits a buy transaction with a higher gas fee to ensure it is processed before the victim's transaction. This allows the attacker to manipulate the asset's price, typically driving it higher. Frontrunning enables the attacker to profit from price changes caused by others' trades and often results in higher costs and slippage for regular users [46].

After the victim's transaction is executed—purchasing the asset at the inflated price—the attacker completes the sandwich by executing a sell transaction (backrunning) to offload the asset at the now higher price. Backrunning, in this case, allows the attacker to capture profits from the manipulated price after the victim's transaction has taken effect [47].

By combining these techniques, sandwich attacks harm the victim by increasing their trading costs (due to the inflated price) and undermine market fairness by exploiting their trade for personal gain [45].

### Arbitrage

Arbitrage is a low-risk trading strategy that takes advantage of price differences for the same cryptocurrency across different exchanges. Traders buy a cryptocurrency on one exchange where it's priced lower and sell it on another exchange where it's priced higher, profiting from the brief price gap. This strategy, also used in traditional markets, benefits from the 24/7 nature of cryptocurrency trading, offering more frequent opportunities. However, arbitrage requires constant monitoring of prices, quick execution, and sometimes specialized tools to succeed. The price differences arise due to factors such as varying supply and demand, liquidity levels, and exchange fees [48]. The 2 main types of arbitrage that we're interested in are atomic and non-atomic arbitrages.

- **Atomic Arbitrage:** Atomic arbitrage is a MEV strategy where arbitrageurs exploit price differences between decentralized exchanges (DEXes). The key feature of atomic arbitrage is that all parts of the trade are executed in a single transaction—either the entire trade, or none of it is completed, which makes the strategy risk-free with respect to holding any inventory. Traders simultaneously buy low on one DEX and sell high on another. Since atomic arbitrage has low barriers to entry and is riskless, it is highly competitive, with traders often competing to tip block builders for transaction inclusion. As more DEXes and tokens are involved, atomic arbitrage becomes more complex, but the core principle remains to profit from price dislocations between trading venues [49].

- **Non-Atomic Arbitrage:** Non-atomic arbitrage refers to a trading strategy where arbitrage opportunities are exploited through transactions that are not executed as a single, cohesive event. Unlike atomic arbitrages, which involve simultaneous execution of trades, non-atomic arbitrage involves trades that may span multiple transactions or involve trading pairs across different exchanges or blockchains. This type of arbitrage is more complex and less transparent because it often involves one side of the trade being executed on Ethereum, while the other side might occur on centralized exchanges (CEXes) or different blockchains, which makes it harder to trace and analyze [5].

# 3. Related Work

## 3.1. Cross-Domain MEV

A short yet informative introduction to cross-chain MEV has been provided by Joules Barragan [50]. The author starts by describing how the rise of multiple blockchain networks introduces unique arbitrage opportunities. For instance, Uniswap - one of the most popular DEXes - has a Total Value Locked (TVL) of $3.8 billion on Ethereum, yet the DEX also operates on other ERC20-compatible chains such as Arbitrum and Polygon. When put together the total TVL locked on Uniswap across all domains reaches $4 billion. The article discusses how traders exploit price differences across these networks where asset values may diverge momentarily. It also covers the technical challenges such as settlement time lags that arise from the fact that cross-chain operations are rarely atomic. This overview helps frame the complexities we consider in our Ethereum-Polygon analysis.

A paper by McMenamin [51] is a deeper dive into the topic of cross-domain MEV, with a particular emphasis on various cross-domain protocols. The paper focuses on the ability of each protocol in mitigating the negative impacts of cross-domain MEV extraction. It gives a brief overview of how MEV is extracted, namely the difference between ordering and signal. When MEV is extracted through ordering, it means that a sequence of transactions is arranged in such a way that the total value at the end of the transaction sequence is higher than at the start. With signal extraction, however, extractors act on off-chain information ("signals") to execute profitable on-chain transactions. A common example of signal extraction is CEX-DEX arbitrage, where a token's price on a centralized exchange drifts away from the price on a decentralized AMM.

The paper also introduces a new categorization which are *intrinsic-extractable value* and *time-extractable value*. The former refers to the expected value an extractor can obtain if they must act immediately, without any delay. Essentially, it's the immediate profit potential available to an extractor if they don't have the luxury of waiting. With time-extractable value the extractor has some flexibility, as they can choose the ideal moment within a time window to act. For instance, in the case of a user transaction, time-extractable value might involve watching a pending transaction in the mempool and determining whether it's profitable to front-run or back-run based on expected price movements.

## 3.2. CEX-DEX Arbitrage

Chen et al. [49] draw comparisons between atomic arbitrages and CEX-DEX, i.e., non-atomic arbitrages by introducing the notion of risk. The authors classify atomic arbitrage as riskless

since traders are not required to hold any inventory between the legs of the arbitrage. Moreover an atomic arbitrage is revertable; if it happens to fail, then the procedure can be reverted at no actual cost to the trader. This makes atomic arbitrage far more preferred by traders, making it highly competitive. On the other hand, CEX-DEX arbitrage, which is off-chain, is not risk-free. It introduces a variety of risks that can result in a net loss of profit for trader, such as the risk of inclusion or the maintainance of token inventory. However, the authors state that despite the risks, CEX-DEX arbitrages offer traders a higher revenue than atomic arbitrage. They observe that while traders who engage in atomic arbitrage pay 90-99% of their revenue to block validators, CEX-DEX arbitrageurs pay only 35-77%. By examining 2 instances or arbitrage, the authors show that a trader who makes $15.79 in profit from atomic arbitrage, may earn a $22.49 before fees in revenue for the same arbitrage in a CEX-DEX setting. And when we account for the percentage of revenue paid to validators between the 2 types of arbitrage, we see that non-atomic arbitrage offers a significantly higher profitability than it's atomic counterpart [49]. These results are relevant to our research in that we also look into non-atomic cross-chain arbitrage and it's profitability.

Heimbach et al. [5] examine non-atomic arbitrage in decentralized finance, focusing on opportunities that exploit price differences between DEXes and off-chain markets like CEXes. The paper starts by differentiating between atomic and non-atomic arbitrage, highlighting the challenges posed by the non-atomic nature of such arbitrages, including increased risks of front-running and failed attempts due to fluctuating prices and the time-sensitive nature of executing trades across separate domains.

The authors use the following heuristics for non-atomic arbitrage detection:

- **Simple Swap**: Executes a single swap on a DEX, is not involved in sandwich attacks, arbitrage, or liquidation, and uses no more than 400,000 gas.
- **Private Transaction**: Submitted directly to the block builder, bypassing the mempool.
- **Fee Requirement**: Includes a bribe to the builder or has a priority fee of at least 1 GWei.
- **First in Sequence**: The swap is the first in that direction in the pool or follows swaps with the same recipient.
- **Established Tokens**: Involves two widely traded tokens listed on CEXs.

The authors study Ethereum data from five leading DEXs, demonstrating that non-atomic arbitrage constitutes nearly 30% of the trading volume on these platforms, equating to approximately $132 billion. They find an average of 5,216 non-atomic arbitrages per day, which, compared to sandwich attacks (5,219 per day), is the second most prevalent type of MEV among the four, followed by cyclic arbitrages and liquidations. The study shows that, despite its complexity and risk, non-atomic arbitrage has become valuable in DeFi, particularly during times of high cryptocurrency volatility, and presents distinct security challenges for the ecosystem.

## 3.3. Cross-Chain DEX-DEX Arbitrage

When examining MEV on Layer 2 networks like Polygon, as in our thesis, a notable study is [52] by Bagourd and Francois. Although this paper focuses on intra-chain DEX arbitrage within Layer 2 networks, rather than cross-chain MEV, it provides a useful foundation for our investigation. By comparing our results with theirs, we can analyze the differences between cross-chain and single-chain arbitrage on Layer 2 chains. The paper finds that MEV extraction on Layer 2 is primarily driven by arbitrage and liquidations, with Polygon showing especially high MEV potential. The authors estimate Polygon's total MEV to exceed $213 million—a significant increase from earlier assessments—which can be attributed to Polygon's low transaction fees and flexible transaction ordering. Unlike Ethereum, Polygon's design supports "optimistic" or "probabilistic" arbitrage, where searchers can send numerous transactions that are allowed to revert if they don't yield a profit. This practice is made affordable by Polygon's low gas fees, which allow for a highly competitive environment in which the authors identified 7.7 million MEV-related transactions across all blocks analyzed between January 2021 and November 2022. This study is particularly relevant to our thesis as it highlights how L2 architecture impacts arbitrage strategies, with Polygon's low costs and transaction volume making it especially attractive for high-frequency arbitrage techniques not viable on other chains.

Mazor et al. [53] conduct an empirical study of cross-chain arbitrage in decentralized exchanges (DEXs), addressing a significant gap in blockchain and decentralized finance (DeFi) research that has primarily focused on single-chain arbitrage within platforms like Ethereum. They propose a framework to examine cross-chain arbitrage opportunities, specifically between PancakeSwap [54] and QuickSwap [55], which operate on separate blockchains (BNB Chain and Polygon, respectively). Building on earlier research on constant product market makers (CPMMs), their work analyzes how cross-chain price discrepancies can yield profitable arbitrage opportunities, despite challenges such as increased transaction fees and the necessity for synchronized pool data across chains. They employ graph-based algorithms to model token swap sequences and detect viable arbitrage paths, gathering data from blockchain explorers and APIs to monitor pools and liquidity states over a one-month period. Their findings show that cross-chain arbitrage diversifies strategies and boosts profitability beyond what's possible in single-chain settings. Unlike prior studies, which have concentrated on cyclic arbitrage within a single DEX ecosystem, Mazor and Rottenstreich's work quantifies the duration, frequency, and revenue potential of cross-chain arbitrage, indicating that these opportunities could be highly profitable under specific conditions.

A study that more directly addresses cross-chain arbitrage is Cross-Rollup MEV: Non-Atomic Arbitrage Across L2 Blockchains by Gogol et al [56]. Unlike our approach, this study does not analyze historical arbitrage transactions executed between Layer 2 networks and the Ethereum mainnet. Instead, it identifies arbitrage opportunities that went unexploited, focusing specifically on rollups rather than sidechains. The authors examined data from Arbitrum, Base, and Optimism from December 31, 2023, to April 30, 2024. Their analysis revealed over 500,000 unexploited arbitrage opportunities between these rollups over a four-month period. Comparatively, arbitrage on Ethereum typically yields around 0.15% of

trading volume in opportunities, while rollups showed distinct patterns: Arbitrum, Base, and Optimism pools offered arbitrage values between 0.03% and 0.05% of trade volume, with ZKsync reaching up to 0.25%. Rollups benefit from faster block times and lower gas fees, allowing for more frequent swaps, though often with smaller volumes. While our research focuses on arbitrage between Ethereum and its sidechain Polygon, this rollup-centered study provides valuable insights, showing how Layer 2 solutions can shape arbitrage opportunities by lowering execution costs and facilitating non-atomic arbitrage across chains.

A significant contribution to the study of cross-domain MEV is the work by Torres et al. [57], which examines MEV within Ethereum and rollups like Arbitrum, Optimism, and zkSync individually. Unlike the study by Gogol et al. [56], which focuses on unexplored MEV opportunities, Torres et al. analyze historical and concluded MEV activities. As such, the study claims to conduct "the first large-scale measurement of MEV practices across Arbitrum, Optimism, and zkSync", and draws a comparison of obtained results with Ethereum over a period of 3 years. Their study explores various aspects of MEV, including volume, profits, costs, competition, and response times across Ethereum and rollups.

The authors then outline their methodologies for detecting various forms of MEV, including arbitrage, sandwich attacks, liquidations, and more. They detect arbitrages by scanning past blocks for token swap events from DEXes like Uniswap V2, Uniswap V3, Balancer V1, Balancer V2, and Curve, extracting information such as token addresses, amounts involved, and the DEX addresses. They group swaps by transaction and link them together to identify arbitrages as cycles of swaps. The process involves checking that the token output from one swap matches the token input of the next swap and that the DEX addresses involved are different. To calculate the arbitrage profit, the authors track token balances and deduct transaction fees. In Ethereum, they also consider coinbase transfers linked to Flashbots bundles, which are used by MEV extractors to incentivize block producers. This allows them to identify arbitrage opportunities and compute the final profit after deducting costs and MEV-related payments.

The study finds that Ethereum consistently has higher arbitrage activity than rollups, though Arbitrum, Optimism, and zkSync show notable increases in arbitrages as they matured. For example, in April 2023, Arbitrum had 7.2× more arbitrages than Ethereum. While Ethereum remains the platform with the highest profits from arbitrage, Arbitrum shows considerably higher arbitrage profits compared to Optimism and zkSync. This is likely due to larger trade volumes on Arbitrum. The number of competitors for arbitrage opportunities is also higher on Ethereum, with up to 14 competitors targeting the same opportunity, compared to fewer competitors on rollups. Additionally, the use of flash loans for arbitrage is more common on rollups, with Arbitrum, Optimism, and zkSync showing higher usage rates compared to Ethereum.

A key reference for our research is the work by Ilisei [58]. This thesis explores cross-chain interoperability and presents a useful taxonomy of blockchain bridges. Alongside its theoretical contributions, it features algorithms designed to detect historical arbitrages between Ethereum and Polygon. In examining data over one million Ethereum blocks between November 4, 2023, and March 23, 2024, the research identifies 4,488 arbitrage events moving

in both directions between chains, revealing that only three searchers dominate 95% of this market activity. The thesis employs the following heuristics:

- The transaction should have interacted with the Polygon bridge.
- The transaction must involve a swap.
- The transaction's index should be within 10% of the total transaction indices in the block.

This approach is successful at identifying arbitrages where bridging and swapping are bundled within a single transaction but misses cases where searchers conduct bridging and swapping in separate transactions on the Ethereum leg of the arbitrage [58]. This is a drawback, as less sophisticated searchers may conduct swapping and bridging operations in separate transactions. Despite these limitations, Ilisei's work has been invaluable to our research, providing a strong methodological foundation on which our approach is based.

# 4. Methodology

In this chapter, we outline our methodology for detecting cross-chain arbitrages between Ethereum and Polygon. We begin with a brief explanation of successful and failed arbitrages. Next, we introduce the heuristics developed to identify Ethereum transactions that are potentially involved in arbitrage activities. Following this, we detail key implementation aspects and outline the process for calculating profits and revenues associated with the detected arbitrages. For a comprehensive explanation of the core algorithm, we refer the reader to [58].

## 4.1. Definitions

Before we explain our methodology, it's essential to define what we mean by successful and failed arbitrage.

### Successful Arbitrage

A successful arbitrage begins with a swap of Token A on the source chain, exchanging it for Token B. Token B is then bridged to the target chain, where it arrives with a corresponding bridge transaction. On the target chain, Token B is swapped back for Token A (or possibly a different profit token). If the arbitrage concludes with Token A, the same token with which it started, we call this a **cyclic arbitrage**. If it concludes with a different profit token, we refer to it as a **non-cyclic arbitrage**. Both cyclic and non-cyclic arbitrages are considered successful in our analysis. It's important to note that our notion of success does not necessarily imply profitability; the arbitrageur may still incur losses. However, we regard the arbitrage as successful as long as it was concluded on the target chain.

### Failed Arbitrage

A failed arbitrage is an arbitrage attempt that has not been successfully completed, typically because the expected price difference between markets disappeared before the transactions could be finalized. It begins similarly with Token A swapped for Token B on the source chain, and Token B is then bridged to the target chain. However, instead of the arbitrage being concluded with a final swap on the target chain, Token B is bridged back to the source chain. Occasionally, the arbitrageur may perform a final swap to convert Token B back to Token A on the source chain in order to revert their funds to the original token they started with. We regard these instances as failed, but *recovered cases of arbitrage*.

## 4.2. Pipeline

In this section, we provide an overview of our pipeline, detailing each stage and the relevant algorithms involved. A diagram of the pipeline is shown in Figure 4.1.

### Data Fetching

The first stage of our pipeline involves data retrieval, starting with collecting Ethereum block data within a specified range, including all transactions within these blocks. We then update this dataset by integrating MEV types for each transaction from Zeromev's API [59], which provides transaction-level details on MEV activities. This allows us to classify transactions, particularly those related to *SWAP* activities, which are key for identifying failed arbitrages.

Next, we process transaction traces within the block range. This serves two purposes: first, to detect if a searcher has included a coinbase transfer (a bribe to the block builder) by analyzing traces for Ether transfers to the block builder; and second, to identify interactions with the Polygon bridge, determining the direction of token transfers. Detection of coinbase transfers is important, as some searchers may offer substantial bribes to builders to accelerate transaction inclusion, which can significantly influence the success of an arbitrage.

It is important to note that we only store Ethereum transaction data locally and do not retain any Polygon blockchain data.

#### Applying Heuristics

Once all Ethereum transactions, along with their corresponding MEV types (if applicable) and coinbase transfer values, have been fetched and stored locally, our pipeline proceeds to the next stage. At this point, we must filter the transactions to identify those that are the most likely candidates for being part of an arbitrage. To achieve this, we apply the following heuristics:

1. **Heuristic 1:** The transaction must interact with the Polygon ERC20 Bridge, regardless of the direction of interaction. Such transactions are referred to as *bridge transactions*.

2. **Heuristic 2:** A bridge transaction must be linked to a swap, either executed within the same transaction or occurring in a separate transaction, even if it is in a different block.

These heuristics enable us to construct the Ethereum leg of an arbitrage. However, identifying a swap-bridge transaction pair does not automatically confirm the presence of an arbitrage. For example, a searcher might simply be swapping tokens and bridging them to Polygon to maintain an inventory there, rather than executing an arbitrage. It is also particularly important to account for cases where bridging and swapping occur in separate transactions, as less sophisticated arbitrageurs often execute these actions independently rather than within a single transaction.

To detect such cases on the Ethereum leg of the arbitrage, we have implemented an algorithm called *individualSwapIdentifier*. The goal of this algorithm is to identify instances

where Ethereum bridging and swapping occur in two separate transactions, often across different blocks.

**Algorithm 1:** Identify Individual Swap Transactions on Ethereum

**Input:** *blockStart*, *blockEnd*
**Output:** *swapBridgePairs*, *bridgesWithoutSwaps*

```
const eth_blocks_in_one_hour ← 300;
transactions ← getTransactions(blockStart, blockEnd);
bridgeTransactions ← {bridgeTx | bridgeTx ∈
  transactions and bridgeTx.bridgeInteraction ≠ NONE and bridgeTx.mevType ≠
  SWAP};
swapBridgePairs, bridgesWithoutSwaps ← [], [];
foreach bridgeTx ∈ bridgeTransactions do
    token, sender, amount ← getBridgeTransactionInfo(bridgeTx.txHash);
    if bridgeTx.bridgeInteraction == EthToPol then
        swapTx ← matchEthereumTxs(blockNumber −
          eth_blocks_in_one_hour, blockNumber, token, amount, sender);
        if swapTx.amountOut == amount and swapTx.tokenOut == token then
            swapBridgePairs.append(bridgeTx.txHash, swapTx.txHash);
    else if bridgeTx.bridgeInteraction == PolToEth then
        swapTx ← matchEthereumTxs(blockNumber +
          eth_blocks_in_one_hour, blockNumber, token, amount, sender);
        if swapTx.amountIn == amount and swapTx.tokenIn == token then
            swapBridgePairs.append(bridgeTx.txHash, swapTx.txHash);
    else
        bridgesWithoutSwaps.append(bridgeTx);
return swapBridgePairs, bridgesWithoutSwaps;
```

The algorithm begins by filtering all bridge transactions to exclude those that involve a swap. This leaves us with pure bridge transactions—either sending or locking tokens into the contract, or receiving or withdrawing tokens from it. Next, based on the direction of the bridge interaction, we examine a one-hour window around the bridge transaction (or the preceding and following blocks) to find the corresponding swap. For example, if the bridge interaction is from Ethereum to Polygon, we take the Ethereum bridge transaction and search the preceding 300 blocks (the average number of blocks mined on Ethereum in an hour) for a transaction that matches the token, output amount, and sender of the bridge input. If such a transaction is found, we identify a case where the swap and bridge occur separately and keep track of this instance. Additionally, we keep track of bridge transactions that do not have a corresponding swap within the one-hour window. This information is later used to detect failed arbitrages, where tokens are bridged into Ethereum and then back to Polygon without a subsequent swap occurring on Ethereum.

After identifying the separate bridge-swap transaction pairs, the next step is to detect cases

where bridging and swapping are bundled together in the same transaction. Algorithm 2 addresses this requirement by verifying that a transaction has interacted with the Polygon bridge and also includes a swap. Once these bundled transactions are detected, we combine them with the previously identified independent bridge-swap pairs and pass the resulting set to the next stage of the pipeline.

**Algorithm 2:** Identify Ethereum Transactions with Bridge and Swap Bundled Together

**Input:** *transaction*
**Output:** `True` if the transaction is non-atomic MEV, `False` otherwise

```
if transaction.mevType ≠ SWAP then
    return False;
if transaction.polygonBridgeInteraction ≠ NONE then
    return True;
return False;
```

### Arbitrage Detection

After narrowing down all Ethereum transactions to those that are potential candidates for arbitrage, we proceed to the final stage of our pipeline, where the actual matching of Ethereum transactions to their Polygon counterparts occurs. This stage produces two lists: one containing all cross-chain MEV extractions, which are essentially the transactions from the previous stage that we successfully matched, and the other comprising failed arbitrages that originate and conclude on the same chain.

**Successful arbitrages**: The process for detecting successful arbitrages is relatively straightforward. When an Ethereum transaction is bridging tokens to Polygon, we locate the corresponding bridging transaction on Polygon that receives the tokens sent by the searcher. To do this, we examine the timestamp of the original Ethereum transaction's block and search for Polygon blocks published immediately after this timestamp within a 5-hour window. This 5-hour timeframe is based on prior research [58], which determined that checkpoint submissions on the Polygon network typically occur within this duration. By examining block explorers, the previous study established that a 5-hour window provides a reliable range to locate the corresponding bridge transaction on the Polygon blockchain. If we identify a transaction that receives the same amount of tokens as those sent by the Ethereum bridging transaction, we designate this Polygon transaction as bridge transaction of the Polygon leg. Subsequently, we look within a one-hour window to find the matching swap transaction that converts the tokens received from the bridge into the target token, thus concluding the arbitrage. This algorithm works similarly for the opposite direction. Our approach implements the same algorithm in [58].

**Failed arbitrages**: Failed arbitrages, like their successful counterparts, can occur in both directions. Specifically, a trader may initiate an arbitrage on Ethereum, bridge their tokens to Polygon, and then return to Ethereum without any further swaps, or vice versa. We will examine both scenarios here.

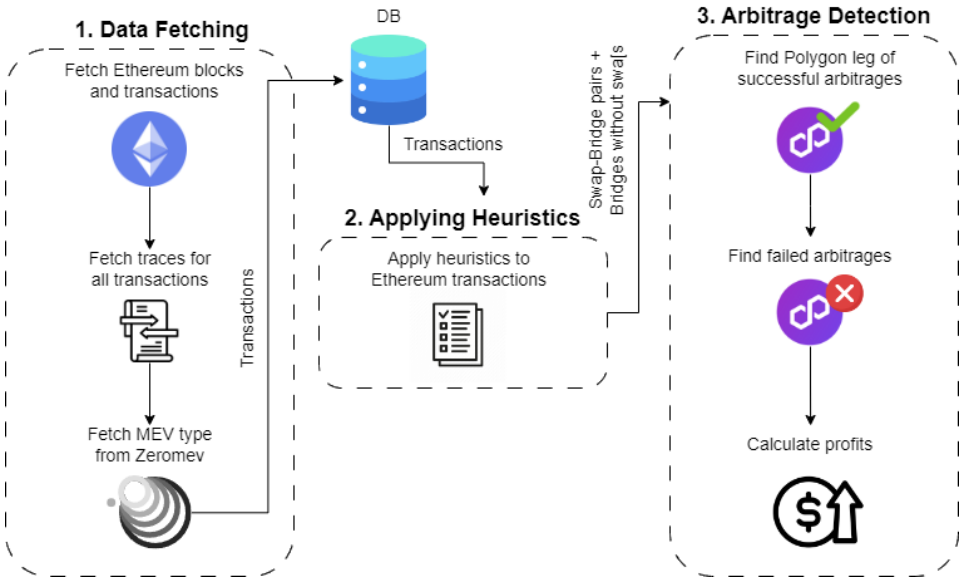


Figure 4.1.: The diagram of the pipeline. First stage represents data fetching, second stage is the heuristic application, and third stage is where the Polygon leg is matched to the Ethereum leg of the arbitrage

*From Ethereum to Polygon*: When an arbitrage begins on Ethereum and the searcher bridges their tokens after a swap, they must also receive these tokens on Polygon. The logic for detecting the bridge on the Polygon side is the same as that for successful arbitrages discussed above. If we cannot find a subsequent swap transaction on Polygon, this raises the possibility of a failed arbitrage. However, the absence of a follow-up swap transaction does not necessarily indicate a failed arbitrage, as a searcher may have simply bridged their tokens to Polygon for future use. To classify this as a failed arbitrage, the initial bridge transaction into Polygon must be immediately followed by a bridge transaction transferring the same amount of tokens back to Ethereum. This clearly indicates that the trader was unable to complete their arbitrage, possibly because the opportunity had expired by the time the tokens were transferred. Additionally, a searcher who has failed to finalize an arbitrage may choose to convert their tokens back to their original tokens after bridging them back to the source chain. Our algorithm is designed to detect this as well.

*From Polygon to Ethereum*: To identify failed arbitrages that originated on Polygon, we need access to Ethereum bridge transactions receiving tokens from Polygon that lack a subsequent swap transaction (either within the bridge transaction itself or separate). We have kept track of these transactions in the previous stage (Algorithm 1 returns a list of *bridgesWithoutSwaps*). At this stage, we analyze these Ethereum bridge transactions and perform two tasks. The algorithm is described in Algorithm 3 below:

In this algorithm, we begin by accepting the bridges without swaps as a parameter and, in line 5, filter them down to those that are withdrawing tokens into Ethereum. Between lines 6 and 8, we identify the corresponding bridge back transaction on Ethereum, which

occurs after the initial bridge, transferring the tokens back to Polygon. This step ensures that we are detecting a failed arbitrage attempt rather than a simple asset transfer to Ethereum by the searcher. Next, between lines 7 and 12, we locate the preceding Polygon swap and bridge transactions to confirm that the Polygon leg of the arbitrage is complete. This ensures that the searcher has first conducted a swap and bridge on Polygon before proceeding to Ethereum. After constructing a failed arbitrage attempt, we also check between lines 14 to 19, if the searcher has conducted an optional swap on Polygon to revert their assets back to their original token.

**Algorithm 3:** Process Polygon -> Ethereum Failed Arbitrages

**Input:** $bridgesWithoutSwaps$
**Output:** $optionalSwapTxnHash, optionalSwap$

```
1  const eth_blocks_in_one_hour ← 300;
2  const pol_blocks_in_five_hours ← 6000;
3  const pol_blocks_in_one_hour ← 1800;
4  foreach tx ∈ bridgesWithoutSwaps do
5      if tx.bridgeInteraction = PolToEth then
6          token, sender, amount ← getBridgeTransactionInfo(tx.hash);
7          blockNumber ← tx.blockNumber;
8          ethereumBridgeBackTransaction ←
             matchEthereumTransactions(blockNumber, blockNumber +
             eth_blocks_in_one_hour, token, amount, PolToEth, sender);
9          blockTimestamp ← getBlockTimestamp(blockNumber);
10         ethSearcherEoa, ethSearcherContract ← getTransactionFromAndTo(tx.hash);
11         polygonLeg ←
             matchToEthereum(tx, ethSearcherEoa, ethSearcherContract, blockTimestamp);
12         polBridgeBackTransaction ←
             matchPolygonTransactions(polBridgeBlockNumber, polBridgeBlockNumber +
             pol_blocks_in_five_hours, polToken, amount, sender);
13         swap ← processTransaction(polBridgeBackTransaction);
14         if swap ≠ null then
15             optionalSwapTxnHash ← polBridgeBackTransaction;
16             optionalSwap ← swap;
17         else
18             optionalSwapTx ←
                 matchPolygonTransactions(bridgeBackTxBlockNumber, bridgeBackTxBlockNumber +
                 pol_blocks_in_one_hour, polToken, amount, sender);
19             optionalSwap ← processTransaction(optionalSwapTx);
20 return optionalSwapTxnHash, optionalSwap;
```

## Detecting Swaps

Our algorithm uses a `SwapProcessor` utility class which is a tool to analyze Ethereum and Polygon blockchain transactions and extract swap-related data. It processes a transaction's logs to identify swap events across several decentralized exchanges (DEXes) and protocols, including Uniswap V2 and V3, Balancer V2, Curve Finance and MeshSwap. Upon receiving a transaction hash, the class retrieves its receipt and decodes the logs for each supported DEX using specific contract ABIs. Detected swap events are converted into `Swap` objects, sorted by their event index to preserve the transaction's sequence. If no swaps are found, the method returns `None`; otherwise, it outputs a list of swaps for further analysis.

This approach, based on [58], involves two significant enhancements. First, it expands the range of supported DEXes beyond Uniswap V2 and V3 to include Balancer V2, MeshSwap, and Curve Finance, enabling the tracking of swaps across a broader array of platforms. Second, it addresses the challenge of transactions containing multiple unrelated swaps, which were previously omitted. Now, such cases are handled by analyzing bridged token amounts. For example, in Ethereum-to-Polygon arbitrages, if multiple swaps are detected on the Polygon leg, the swaps are sorted by their event indices. The algorithm identifies the first swap whose input amount is within 2% of the bridged amount, provided it uses the same token. It then continues analyzing subsequent swaps with larger indices to detect any additional related swaps.

| **Protocol** | **Swap Event Structure** |
|---|---|
| Uniswap V2 | Swap(address sender, uint128 amount0In, uint128 amount1In, uint128 amount0Out, uint128 amount1Out, address to) |
| Uniswap V3 | Swap(address sender, address recipient, int256 amount0, int256 amount1, uint160 sqrtPriceX96, uint128 liquidity, int24 tick) |
| Balancer V2 | Swap(bytes32 poolId, address tokenIn, address tokenOut, uint256 amountIn, uint256 amountOut) |
| Curve Pool | TokenExchange(address buyer, int128 sold_id, uint256 tokens_sold, int128 bought_id, uint256 tokens_bought) |
| MeshSwap | ExchangePos(address token0, uint256 amount0, address token1, uint256 amount1) |

Table 4.1.: Swap Event Structures for Various DeFi Protocols

## Arbitrage Profit Calculation

To compute arbitrage profits, we first calculate transaction fees on Ethereum, including gas costs for bridge and swap transactions. Gas fees are adjusted based on the transaction type

(legacy or EIP-1559) and converted to USD using the historical ETH-to-USD rate at the transaction time.

Next, the Polygon leg is processed similarly. Gas fees for the bridge and swap transactions are calculated in MATIC and subsequently converted to USD using historical MATIC-to-USD rates. If the bridge and swap transactions are distinct, separate gas fees are calculated for each. The total gas fees for both networks are then summed up.

Following the calculation of fees, the input and output amounts for the arbitrage are determined based on the tokens exchanged in the swaps. These amounts are also converted to USD using historical token exchange rates. The profit before fees is then calculated as the difference between the output and input amounts in USD. Finally, the total profit is determined by subtracting the total gas fees from the revenue. These values are saved in the extraction object for further analysis.

By calculating the profits and fees associated with individual arbitrages, we gain a clear understanding of the financial outcome of each arbitrage attempt. Additionally, this information enables the analysis of searcher strategies, profit margins, and the profitability of various tokens in comparison to one another.

### Revenue and Profit Calculations

Revenue is defined as the difference between the output of the last swap on the target chain and the input of the first swap on the source chain. This represents the net amount received after completing the arbitrage transaction, calculated as:

$$\text{Revenue} = \text{Output amount of the last swap} - \text{Input amount of the first swap}$$

Profit is determined by subtracting the total costs associated with the arbitrage transaction from the revenue. These costs include the total gas fees on both chains as well as any coinbase transfer amounts. Profit is effectively the remaining amount after accounting for all transaction-related expenses:

$$\text{Profit} = \text{Revenue} - (\text{Total Gas Fees} + \text{Coinbase Transfer Amounts})$$

The formula used to calculate profit margins in an arbitrage is:

$$\text{Profit Margin} = \frac{\text{Total Profit}}{\text{Revenue}} \times 100$$

## 4.3. Transaction Transparency

Ethereum transactions are typically broadcast to the public mempool, where all validators and block builders can see and include them in a block. However, some transactions are privately submitted to a block builder to avoid revealing sensitive information, like the details of a high-value arbitrage or liquidation opportunity, before it is executed. This reduces the risk of front-running, where others could exploit this information to execute their own transactions ahead of the original [60].

After detecting an arbitrage transaction, we analyze the Ethereum swap and bridge transactions to determine whether it was submitted to the public mempool or directly to the block builder via a private channel. To achieve this, we use the historical mempool transaction archive provided by Flashbots [61]. We query this dataset using DuneAnalytics. By submitting a list of transaction hashes, we check whether a transaction appears in the mempool records. If found, we classify it as public; if not, we designate it as private. The DuneAnalytics query we have used is given in the appendix [A.2].

Accessing transaction transparency is vital for understanding whether arbitrage transactions are privately sent to block builders, potentially indicating order flow agreements between searchers and builders. This helps in analyzing transaction strategies and the role of private transactions in MEV.

## 4.4. Case Studies: Successful and Failed Arbitrage

### Successful Arbitrage

In this section, we will review case studies for both successful and failed arbitrage. A successful arbitrage can occur in two directions: either from Ethereum to Polygon or from Polygon to Ethereum. Since the process is symmetric, we will provide an example for only the former direction.

Below is an example of a successful arbitrage that starts on Ethereum as the source chain and ends on Polygon as the target chain. The operations and their corresponding transaction hashes are provided:

- The arbitrageur swaps 2 WETH for 177,670 on Uniswap V3 [1].
- He then bridges these FOAM tokens to Polygon using the Polygon ERC20 Bridge [2].
- The FOAM tokens are received on Polygon [3].
- Finally, 177,670 FOAM are swapped back for 2.089 WETH on Paraswap V5 [4].

In this example, the arbitrageur makes a profit of 0.089 WETH, which, adjusted for historical exchange rates, amounts to $343.69. After deducting gas fees paid on both legs of the arbitrage, totaling $80.03, the net profit is $263.66.

### Failed Arbitrage

Failed arbitrages generally result in financial losses; however, this is not always the case, as we have detected instances where arbitrageurs unintentionally profited. Below is an example of a failed arbitrages resulting in a net loss, which starts on Ethereum as the source

[1] 0x2c71ab38a89aeae6a0d62f581aa7dea16c89d8902afe0d1d5e41bd0c1cf506be
[2] 0x809cee2c07fa778dcb89620f74d1e4a8bf0580518a5a73b73b89feac6a86cf6a
[3] 0x0c0540c8675c8607d4d9feae76a12577a05f17785880e6d634f791e17e4bcf7f
[4] 0x8f5feee033d42988444004b1bcf439de5dd8537a5dffd8f58d324166328d1204

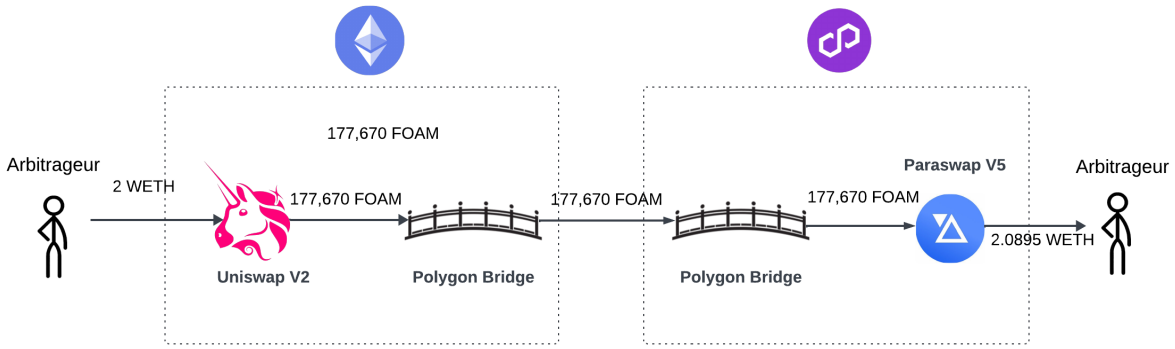


Figure 4.2.: Successful Arbitrage Example

chain and, instead of concluding on Polygon, reverts back to Ethereum. The operations and corresponding transaction hashes are provided:

- The arbitrageur swaps 0.383 ETH for 6,270 SWAP tokens [5] on Uniswap V2.
- The SWAP tokens are then bridged to Polygon using the Polygon ERC20 Bridge [6].
- The SWAP tokens are received on Polygon [7].
- Instead of performing a swap on Polygon, the tokens are bridged back to Ethereum [8].
- The tokens are received on Ethereum through a bridge transaction [9].
- Finally, the arbitrageur swaps the SWAP tokens back into 0.369 ETH on Ethereum [10].

In this case, the arbitrageur ends up losing $23.50. When we deduct the total gas fees paid on both legs of the arbitrage (which amount to $15.41), the total loss comes to $38.91.

## 4.5. Limitations

Our algorithm relies on events emitted directly by decentralized exchanges (DEXs), such as Swap events, to detect swaps. However, this approach is less effective when swaps are executed through routers like MeshRouter or Aggregation Router v4/v5, which often omit standard Swap events. Instead, these routers may emit custom or generalized events without token in/out details, complicating the tracking of swaps.

For example, in transaction *0x87213638c25c205a244f7e1615b843432ab751eeed5dd1b484b0160634afdf3d*, a trader swapped 200 USDC for 961.55 VAI using the 1inch Aggregation Router V5. Since no

[5] 0x9056eef70bb40ddc97a6dc9c595958a595b5e4d53ddea1fc99766f87885919a6
[6] 0x3ca2994e91d640b96441180160e5c2dbe4c611c50ea2e28975a508c23a5f6962
[7] 0xf0945f056522a285bacb828aa67afed23fb48c61686f4708ceb09ca2e2462676
[8] 0xfe1dc3a5a7082201dac6e1e9311e1f122722c3488950500866bd83679ac14d20
[9] 0xb755724522761073de27f0b13f920242bac5ffc14600fd5522cf4d328a0aab82
[10] 0xb755724522761073de27f0b13f920242bac5ffc14600fd5522cf4d328a0aab82

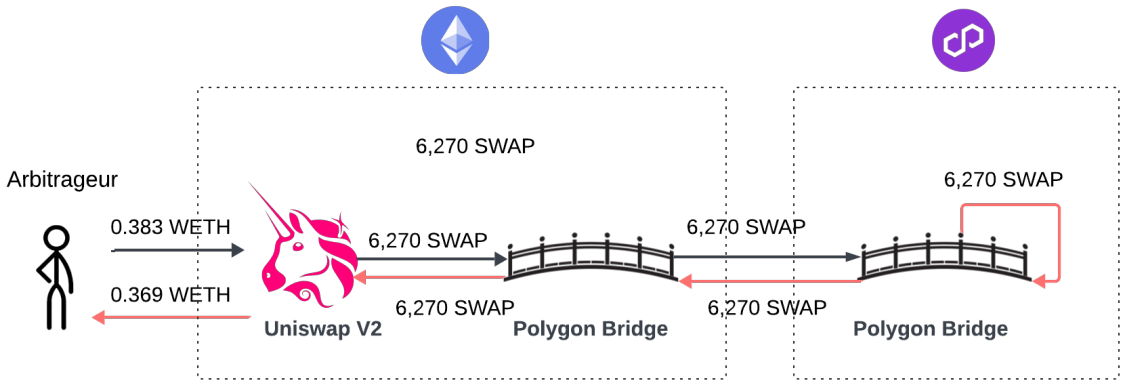


Figure 4.3.: Failed Arbitrage Example

standard Swap event was emitted, our current approach failed to capture the swap details. Consequently, certain swaps executed via such routers may go undetected.

Another limitation is the speed of data retrieval. Since we need to access a full year's worth of Ethereum data (approximately 2.4 million blocks) and the traces for each transaction within those blocks, the data-fetching process can be quite slow. Running our own node would be too time-intensive, so we opted to use a public RPC endpoint. While convenient, public RPC endpoints have usage limits, including request and sometimes daily limits, which impact retrieval speed. A more efficient approach would be to rely on events emitted by the bridge rather than examining transaction traces individually to determine interactions with the Polygon ERC20 bridge and the direction of those interactions. For example, by tracking all ERC20Locked events for a specific bridge contract within a given block range, we could quickly identify all transactions that locked tokens into the bridge. This method would greatly reduce latency, as we would no longer need to check each transaction's traces.

In order to pinpoint coinbase transfers (i.e., bribes to the block builders), we look at transaction traces. As discussed, this requires individual trace processing and is time-consuming. The coinbase transfer values can be easily imported using DuneAnalytics. An example DuneAnalytics query to fetch coinbase transfer values can be accessed in A.3

Additionally, we rely on a heuristics-based approach for detecting arbitrage transactions. This approach involves making educated guesses about searcher behavior based on observed patterns rather than deterministic methods. While heuristics can offer faster insights, they also introduce the risk of inaccuracies or missed detections, as they depend on assumptions that may not always hold true in every case.

## 4.6. Data Sources

### RPC Services

In blockchain applications, Remote Procedure Call (RPC) endpoints are services that allow programs to interact directly with the blockchain by submitting transactions, querying data, and accessing blockchain state. Running an RPC node independently for networks like Ethereum and Polygon can be extremely resource-intensive due to the large storage and bandwidth requirements. As an alternative, Ankr [62] offers a reliable RPC service for accessing Ethereum and Polygon networks. Besides these chains, the service also supports 50+ blockchains, has a relatively higher transaction rate of 30 requests / second, and no daily request limits. The endpoints also support tracing, which is important, as we use transaction traces to determine if a transaction has interacted with the Polygon bridge, as well as the direction of this interaction. However, as outlined in the limitations section above, this information could also be determined from the bridge contract events, in which case, any public RPC (including the ones that don't support tracing) will suffice.

### Historical Token Exchange Rates

To fetch historical exchange rates of tokens on different chains, we used the following API endpoint offered by Covalent [63]:

```
https://api.covalenthq.com/v1/pricing/historical_by_addresses_v2/
```

### Token Mapping

Token mapping between Ethereum and Polygon is essential because tokens often exist on both networks but have different contract addresses on each. When a token is bridged from Ethereum to Polygon, a mapped version of that token is created on Polygon, and the contract addresses for these versions are distinct. Access to the mapped token addresses on Polygon is necessary for applications to identify and handle the correct token representations, especially in cross-chain transactions or when verifying token identities. To obtain the mapped token addresses on Polygon, we use the following API [64]:

```
https://api-polygon-tokens.polygon.technology/tokenlists/mapped.tokenlist.json
```

### Zeromev API

Zeromev is an organization that offers a set of tools and APIs designed to understand and mitigate MEV by analyzing blockchain transaction data [59]. Its API provides detailed transaction-level MEV summaries, categorizing the types of MEV detected within transactions and blocks. We rely on the *Swap* type to update our transactions, identifying whether or not they involve a swap.

### Polygonscan API

We use the Polygonscan API to identify Polygon blocks relative to a specific timestamp [65]. This functionality is important for analyzing cross-chain interactions involving Ethereum and Polygon bridge transactions. The API's `getblocknobytime` endpoint helps pinpoint the Polygon block number that aligns with a given timestamp.

### Usage

- **Bridge Transaction Timestamp:** The timestamp from the Ethereum side bridge transaction's block serves as the reference point.
- **Determining Polygon Blocks:**
  - If the Ethereum bridge transaction **withdraws tokens** (from Ethereum to Polygon), the goal is to identify what occurred **before** this timestamp on the Polygon chain.
  - If the Ethereum bridge transaction **deposits tokens** (from Polygon to Ethereum), the focus shifts to identifying events **after** this timestamp on the Polygon chain.

### API Request Example

```
https://api.polygonscan.com/api?module=block&action=getblocknobytime&
    timestamp={timestamp}&closest=before&apikey={PolygonAPIKey
```

- `timestamp`: The Unix timestamp of the Ethereum bridge transaction's block.
- `closest`: The parameter is set to `before` to find the block immediately preceding the timestamp. Changing this to `after` would retrieve the first block after the timestamp.
- `PolygonAPIKey`: Your unique API key for accessing the Polygonscan API.

### Dune Analytics

Dune Analytics [66] is a powerful blockchain analytics platform that allows users to query, visualize, and share data from multiple blockchain networks using SQL. Flashbots [67] is a research and development organization which focuses on solving the issue of Maximal Extractable Value (MEV) on Ethereum and other blockchains. Flashbots offers a free mempool transaction archive called Mempool Dumspter. For our analysis, we use Dune Analytics to query data from this Flashbots mempool in order to obtain the transparency information of arbitrage transactions. If a transaction appears in the mempool, it was publicly submitted and visible to anyone monitoring the mempool. If it does not appear, the transaction was likely submitted privately to the block builder. We have included the query in the appendix A.2.

# 5. Results

## 5.1. Overview

We collected and analyzed 2,614,000 Ethereum blocks between September 1, 2023, and August 31, 2024. Within these blocks, we identified 23,404 transactions involved in arbitrage between Ethereum and Polygon, corresponding to 0.89% of the examined blocks. Out of these 23,404 detected cross-chain arbitrages, 16,812 are cyclic (arbitrages which have started and concluded with the same token), and 6,592 are non-cyclic. Additionally, we detected 1,296 failed arbitrages, 1,092 of which were recovered. In these cases, the searchers, after bridging their tokens back to the source chain, executed an additional swap to revert their tokens to their original state. We found that 58% of arbitrage transactions move from Ethereum to Polygon, while 42% go the other way. This difference may be due to shorter bridging times for Ethereum-to-Polygon transfers, averaging about 19.76 minutes, compared to 48.45 minutes for the reverse. This delay for transfers from Polygon back to Ethereum on the Polygon PoS Bridge stems from the checkpointing process, which is part of Polygon's security model. Specifically, when moving assets from Polygon to Ethereum, the protocol requires a checkpoint that records the transaction on the Ethereum network. This checkpointing process usually takes between 45 minutes to 5 hours, as the validators need to batch and verify multiple transactions before submitting them as a single checkpoint on Ethereum. In contrast, transfers from Ethereum to Polygon are generally faster, typically taking around 15 minutes on average [68].

| **Aspect** | **Results from [58]** | **Our Detection** |
|---|---|---|
| **Timeframe** | Nov-04-2023 to Mar-23-2024 | Nov-04-2023 to Mar-23-2024 |
| **Successful Arbitrages** | 4,488 | 10,684 |
| **Failed Arbitrages** | 158 | 554 |
| **ETH $\rightarrow$ POL** | 58% | 41.21% |
| **POL $\rightarrow$ ETH** | 42% | 58.79% |

Table 5.1.: Comparison of results with [58]. The differences in directionality of arbitrage transactions (ETH $\rightarrow$ POL vs. POL $\rightarrow$ ETH) stem from a methodological limitation in [58], where on the Ethereum leg, only transactions involving both the bridge and swap are detected.

These results show an improvement over the findings of [58]. In contrast to that study, which limited the Ethereum leg of arbitrages to cases where swapping and bridge interactions occur within the same transaction, our approach also identifies arbitrages where the Ethereum

leg spans multiple transactions. This broader detection method enables us to capture a wider range of arbitrages. Some notable differences are summarized in Table 5.1.

## 5.2. Transparency

In this section, we evaluate the transparency of Ethereum swap and bridge transactions. Figures 5.1 and 5.2 present the distributions of bridge and swap transactions based on their transparency. One notable difference is that for the bridge transactions on the Ethereum leg, 45.35% of them are private, while 54.65% are public. However, for the swap transactions, a very different trend is observed: 70.23% are private, and only 29.77% are public. This indicates that swap transactions tend to be more private compared to bridge transactions. Swaps are more sensitive, as they are often part of strategies that searchers want to keep confidential to avoid leaking their arbitrage opportunities. This increased privacy is likely a result of searchers' efforts to prevent competitors from detecting their strategies and capitalizing on them before they can be executed.

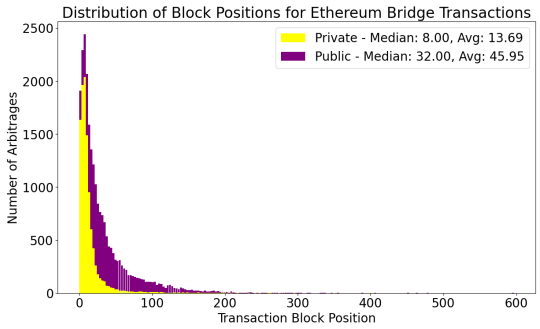


Figure 5.1.: Distribution of private and public bridge transactions

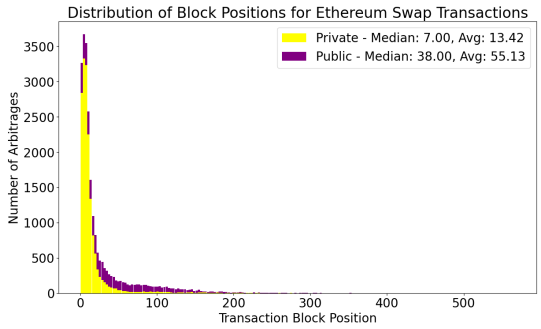


Figure 5.2.: Distribution of private and public swap transactions

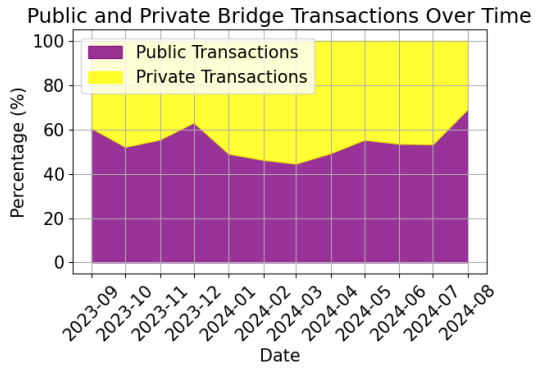


Figure 5.3.: Stacked plot for private and public bridge transactions

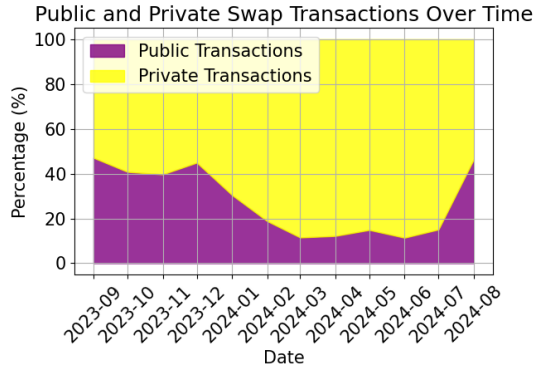


Figure 5.4.: Stacked plot for private and public swap transactions

For both bridges and swaps, we observe that private transactions are more commonly clustered towards the top block positions. This is likely because participants aim to execute their transactions first, ensuring their arbitrage strategies are not affected by state changes caused by others' transactions. In bridge transactions, the average block position for private transactions is 13.69, compared to 45.95 for public transactions. Similarly, in swap transactions, private transactions occur at an average block position of 13.42, while public transactions are positioned at an average of 55.13. The higher concentration of private transactions in smaller block positions indicates that participants prioritize executing early to prevent competitors from reacting to their arbitrage opportunities.

Figures 5.3 and 5.4 shows the trends of public and private transactions over time. A key difference is that for bridge transactions, the proportion of public transactions does not drastically differ from the private ones. In contrast, for swap transactions, the majority are overwhelmingly private. This is, perhaps due to swaps involving higher-value trades, where participants may prefer the confidentiality provided by private transactions to minimize front-running or other market risks.

The increase in private arbitrages observed after November 2023 in Figure 5.4 can be attributed to searchers who favor private transactions conducting more arbitrages. Meanwhile, the volume of searchers engaging in public arbitrages remains stable, as can be seen in Figure 5.11 and Table 5.2.

## 5.3. Arbitrage Durations

In this subsection, we analyze the time differences between various stages of arbitrage transactions. Figure 5.5 illustrates the distribution of time differences between Ethereum bridge and swap transactions. The majority of these transactions occur within a minute of each other, indicating that many arbitrageurs either perform the swap and bridge transactions back-to-back or even in the same transaction. The delay between these stages could be due to changes in position profitability, with arbitrageurs waiting for a more favorable opportunity before executing the next step.

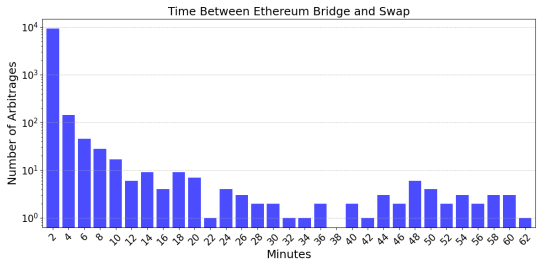


Figure 5.5.: Time Between Ethereum Bridge and Swap

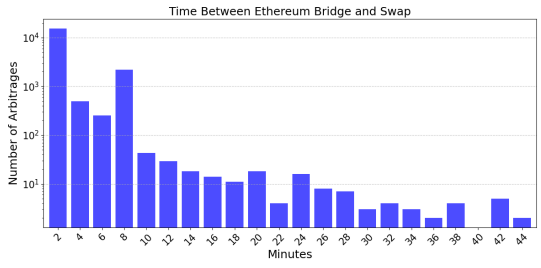


Figure 5.6.: Time Between Polygon Bridge and Swap

In a similar manner, Figure 5.6 illustrates the distribution of time differences for Polygon bridge and swap transactions. The pattern closely mirrors that of Ethereum, with the majority

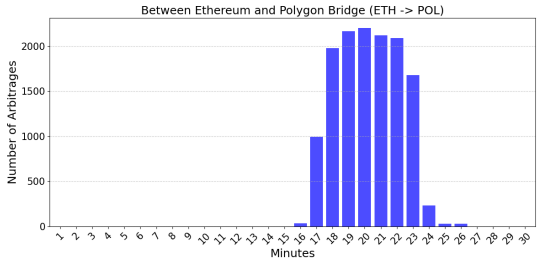


Figure 5.7.: Time Between Ethereum Bridge and Polygon Bridge (ETH → POL)

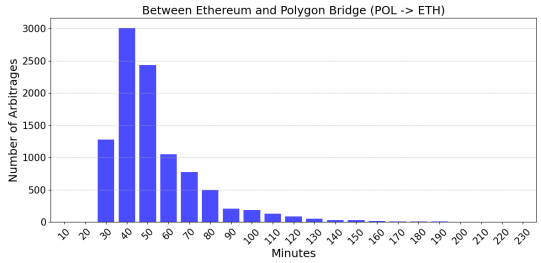


Figure 5.8.: Time Between Ethereum Bridge and Polygon Bridge (POL → ETH)

of transactions occurring within a relatively short time frame, and an average time difference of 57.39 seconds. However, a few transactions show longer delays, suggesting that while the timing dynamics across both networks are generally consistent, there are occasional inefficiencies.

Figures 5.7 and 5.8 display the time differences between bridge transactions between Ethereum and Polygon, categorized by the direction of arbitrage. As previously discussed, the average bridging time for Ethereum to Polygon (ETH → POL) arbitrages is 19.76 minutes, which is significantly shorter compared to the reverse direction (POL → ETH), where the average time is 48.45 minutes. This discrepancy can likely be attributed to the more time-consuming checkpointing process involved in transfers from Polygon to Ethereum.

We also analyzed the impact of the time difference between bridge and swap transactions on Ethereum on the profitability of arbitrage. The correlation was extremely weak, indicating no meaningful relationship between the time difference and profitability. Therefore, we couldn't find any significant connection between the timing of these transactions and the resulting profit.

## 5.4. Searchers

To understand the competition within the arbitrage space, we analyze the addresses involved in the identified arbitrages. These addresses, referred to as searchers, can be either externally owned accounts (EOAs) or smart contracts (SCs). Our analysis has revealed that a significant 85.68% of all transactions can be attributed to the top six searchers, highlighting the concentration of market activity among a few dominant players in the Ethereum ecosystem.

To accurately assess the contributions of each searcher, we obtain a list of searcher addresses from the searcherbuilder.pics dashboard[1]. In this process, we verify whether the searcher contract addresses associated with our detected arbitrages are classified as non-MEV smart contracts based on their presence in the dashboard's list. If an address is found on the list, we use the corresponding externally owned account (EOA) behind the arbitrage; if it is not

[1] `https://github.com/winnsterx/searcherbuilder.pics/blob/main/labels/non_mev_contracts.py`

present, we retain the searcher smart contract address. The results of our analysis can be seein in Table 5.2

**Arbitrage Statistics by Top 6 Searchers**

| Searcher | Type | Total Arbs | Profit ($) | Gas Fees ($) | Profit Margin (%) | Private (%) | Avg. Profit/Arb ($) |
|---|---|---|---|---|---|---|---|
| 0x826A | SC | 6,002 | 114,921.36 | 197,786.36 | 36.75 | 77.52 | 19.15 |
| 0x882d | SC | 5,992 | 224,665.12 | 136,861.14 | 62.14 | 50.22 | 37.49 |
| 0x0a6c | EOA | 4,159 | 52,865.96 | 81,113.01 | 39.46 | 6.42 | 12.71 |
| 0xcA74 | EOA | 1,759 | 98,902.89 | 36,404.21 | 73.10 | 1.42 | 56.23 |
| 0x3fA6 | SC | 1,229 | 42,498.92 | 34,130.95 | 55.46 | 99.43 | 34.58 |
| 0x84e7 | SC | 910 | 1,176.56 | 1,699.58 | 40.91 | 0.00 | 1.29 |
| Others | - | 3,350 | 198,365.91 | 59,378.15 | 76.99 | - | - |
| **Total** | - | 23,401 | **733,396.72** | - | - | - | - |

Table 5.2.: *Searcher* denotes the first 4 characters of the searcher address (full addresses can be accessed in the appendix), *No. of Arbs* is the total number of arbitrages conducted by this searcher, *% of Total Arbs* is the percentage of the total arbitrages performed by this searcher relative to all searchers, *Private Arbs %* is the percentage of arbitrage transactions executed privately by the searcher, *Total Profit* is total profit earned from arbitrages, and *Avg. Profit/Arb* is the average profit per arbitrage, all in USD. Full addresses of top searchers are available in [A.1]

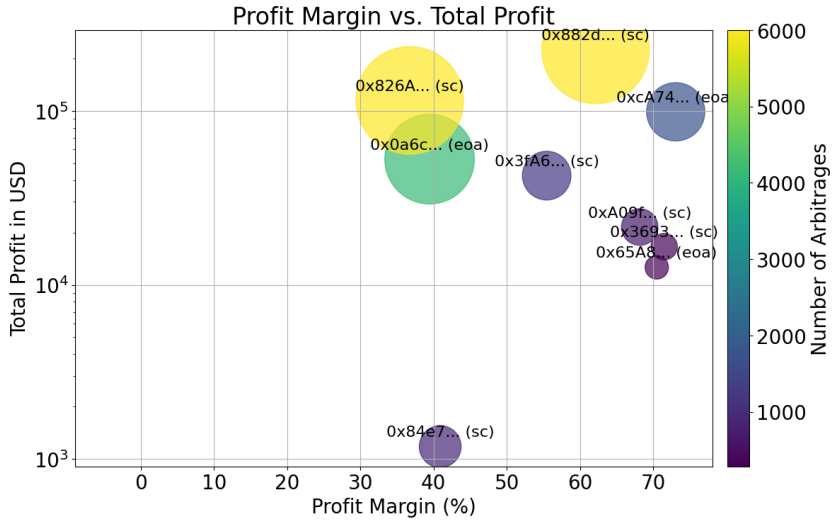


Figure 5.9.: The x-axis shows the profit margin in percentage terms, while the y-axis shows the total profit (the cumulative profit achieved by a searcher) in a logarithmic scale. Each bubble represents an individual searcher, with the bubble size proportional to the number of arbitrages the searcher has conducted.

The Figure 5.9 illustrates the relationship between profit margin and total profit for all searchers in our results. We observe that the largest bubbles generally occupy areas of higher

profit, which suggests that the most active searchers tend to be more profitable overall.

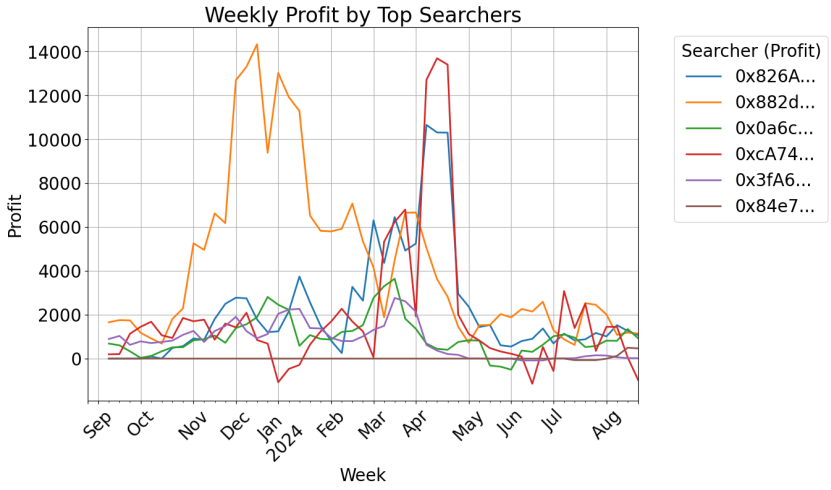


Figure 5.10.: Profit distribution of the top 6 searchers based on the total profit generated through arbitrages between Ethereum and Polygon. The figure highlights the differences in profit for each searcher.

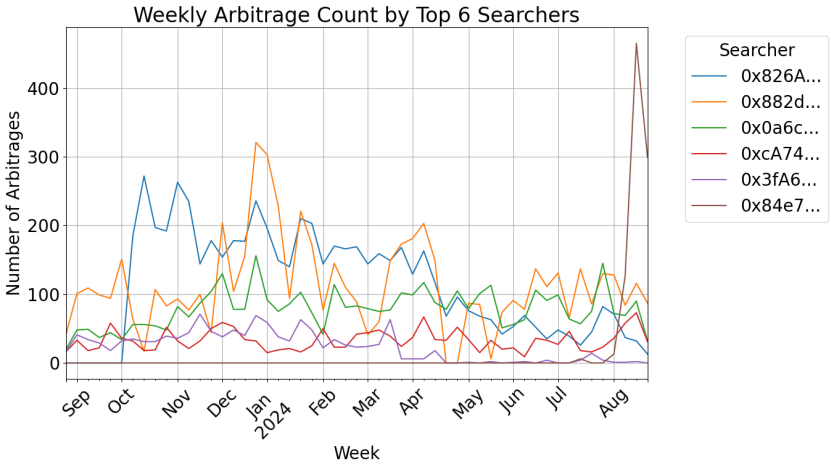


Figure 5.11.: Arbitrage count for the top 6 searchers, showing the total number of arbitrages conducted between Ethereum and Polygon. This figure illustrates the volume of activity of each searcher.

Figure 5.10 presents the number of arbitrages over time. For example, we can see that the searcher 0x826A has been especially active between October and April. However, despite this high trading volume, their profits, as can be seen on Figure 5.11, remain relatively modest. This suggests a trading strategy that prioritizes smaller transactions rather than larger, more lucrative opportunities. This example shows that executing a larger number of arbitrages does not necessarily correlate to higher profitability.

Looking at Figure 5.11, we can see how different searchers performed in terms of profitability. 0xcA74, for example, shows significant fluctuations in their profit over time. There are a few periods—early January, mid-to-late June, and late August—when their profits briefly dipped into the negative. However, there was also a notable surge in profits between March and May, indicating a successful period for this searcher. On the other hand, 0x882d's performance is much more consistent, with profits steadily increasing throughout the year and reaching a peak in mid-December. This consistent increase in profits suggests that 0x882d's trading strategy is more effective, managing to consistently capitalize on opportunities over the course of the year.

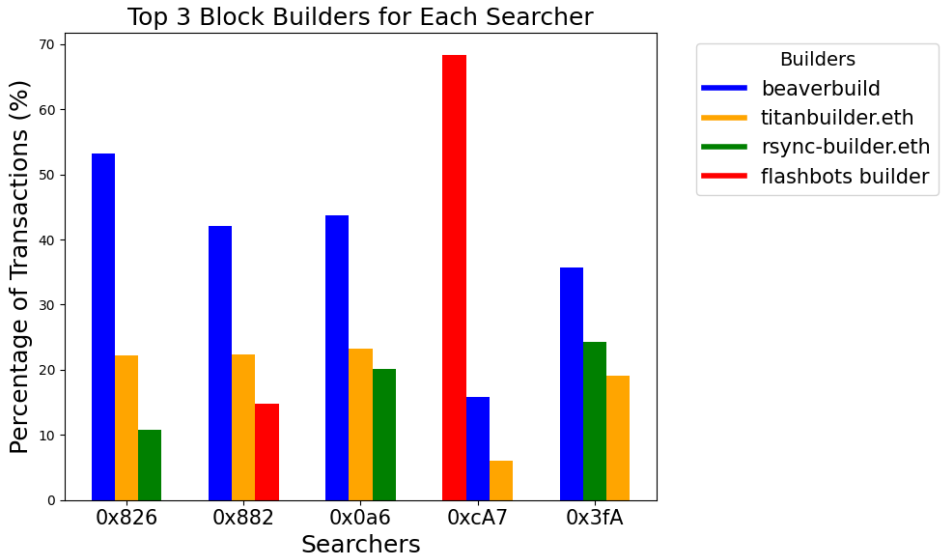


Figure 5.12.: Distribution of block builders per top 5 searchers. Only the private swap transactions have been taken into account.

We used transaction transparency data to investigate potential private order flow deals between searchers and block builders, focusing on the distribution of private swap transactions from the top five searchers to block builders. The results, shown in Figure 5.12, align with our expectation, as the identified top block builders already handle a significant portion of Ethereum's block production. Notably, the majority of private swap transactions from searcher 0xcA7 were processed by Flashbots, suggesting the use of the private RPC endpoint provided by Flashbots Protect [41]. However, it must be noted that searcher 0xcA7, as shown in Table 5.2, engages in very few private transactions, with only 1.42% of their transactions being private and a fewer number of 1,759 arbitrages.

While we cannot conclude the presence of private deals—an area beyond the scope of this thesis—as the space becomes more profitable, we may observe an increase in private order flow agreements [69].

## 5.5. Tokens

In this section, we analyze the tokens that were involved in cross-chain arbitrages. We have found that stablecoins and highly liquid tokens, which are popular in cyclic or CEX arbitrages, are rarely used in cross-chain non-atomic arbitrages. This is because such tokens are frequently traded on both CEXs and DEXs, where fees are lower, and price discrepancies are quickly corrected due to their high liquidity and widespread availability. Instead, cross-chain arbitrages often focus on less liquid tokens that are primarily traded on DEXs. For such tokens, cross-chain arbitrage may represent the only viable option to exploit price differences, as they are not listed on major centralized exchanges. Additionally, the extended bridging times required for cross-chain transactions often make stablecoins less appealing, as their prices are more stable and rapidly adjusted across chains.

Figure 5.13 shows the distribution of intermediary tokens used in arbitrages between Ethereum and Polygon, highlighting those involved in more than 1% of detected arbitrages. Intermediary tokens refer to those that were bridged from one chain to another, distinct from the initial tokens involved in the first and last swaps on the source and target chains. The analysis reveals a diverse range of tokens, with no single token dominating. GLQ leads with only 7.9% of the total arbitrages, closely followed by SMT and FACTR. This lack of dominance underscores the diversity of assets used by arbitrageurs, who seem to prioritize tokens with limited liquidity or niche appeal over well-known tokens.

Figure 5.14 shows the usage of intermediate tokens over a one-year timeline. We see that the usage of certain tokens, such as SmartMesh Token (SMT) and FACTR, remains relatively consistent throughout the year, which indicates a steady interest. On the other hand, tokens like YUP, OM, and AKT display peak usage between November 2023 and February 2024, followed by a significant decline in activity. Some tokens, like DG and UNIX, even reach a point of zero usage. These latter tokens are strong candidates for long-tail tokens, as their popularity seems to be short-lived and highly concentrated in specific time frames. Tokens like DG and UNIX seem to have quickly lost interest, followed by rapid decline in usage, suggesting a lack of long-term utility. Additionally, BANANA, although a memecoin, shows consistent usage throughout the year, indicating that its appeal goes beyond short-term trends and may have sustained interest in certain communities.

| **Profit Token** | **Count** | **Percentage** |
|---|---|---|
| WETH | 15318 | 93.1% |
| USDT | 322 | 1.9% |
| USDC | 808 | 4.9% |

Table 5.3.: Profit Tokens and their Percentages in Cyclic Arbitrages

Table 5.3 presents a summary of profit tokens used in cyclic arbitrages. The dominant profit token, WETH, accounts for 93.12% of the total cyclic arbitrages.

Tables 5.4 show token involvement in non-cyclic arbitrages, focusing on tokens involved in over 1% of non-cyclic arbitrages. Compared to cyclic arbitrages, WETH remains the

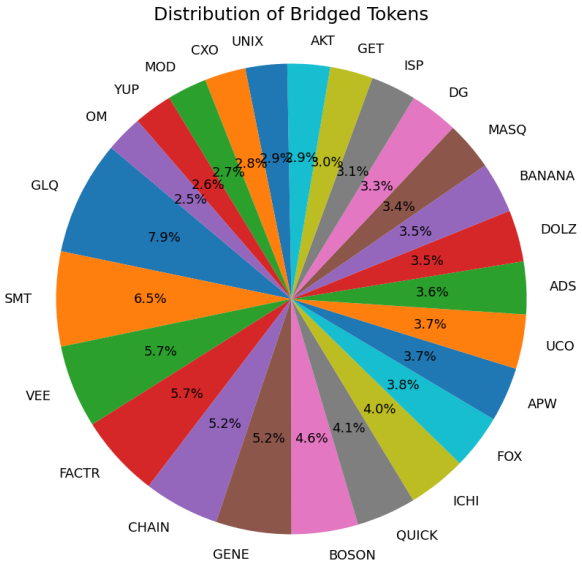


Figure 5.13.: Distribution of intermediary tokens bridged between Ethereum and Polygon, focusing on those involved in over 1% of arbitrages.

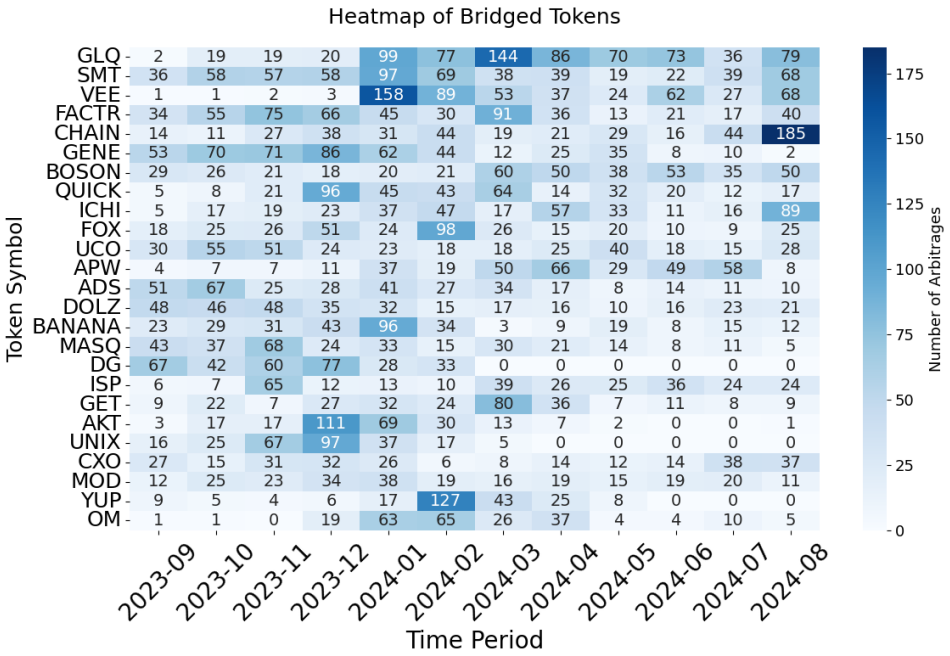


Figure 5.14.: Heatmap showing the usage of intermediate tokens over a one-year period.

predominant token in both, but non-cyclic arbitrages involve a greater diversity of tokens. Notably, USDC, WMATIC, and USDT also play significant roles, with USDC and WMATIC being more prominent in the end token positions of non-cyclic arbitrages.

| Token | Start Count | Start Percentage | End Count | End Percentage |
|---|---|---|---|---|
| WETH | 3875 | 56.35% | 1998 | 29.20% |
| USDC | 1430 | 20.08% | 2039 | 29.79% |
| WMATIC | 1011 | 14.70% | 2352 | 34.37% |
| USDT | 357 | 5.19% | 344 | 5.02% |

Table 5.4.: Non-Cyclic Arbitrage Start and End Tokens with Counts and Percentages

## 5.6. Comparison to Other MEV Strategies

In this section we draw a comparison between our results and some other forms of MEV.

| Category | Data Source | Count | Profit |
|---|---|---|---|
| Cross-Chain Arbitrages | Our Results | 23,400 | $743,368.86 |
| Atomic Arbitrages (Ethereum) | Dune, EigenPhi | > 3.6 million | > $300 million |
| Atomic Arbitrages (Polygon) | [52] | 7.7 million | > $213 million |
| Sandwich Attacks | Dune, EigenPhi | > 1.5 million | > $48 million |

Table 5.5.: Comparison of cross-chain arbitrages between Ethereum and Polygon, single-chain atomic arbitrages on Ethereum (September 2023 to August 2024) and Polygon (January 2021 to November 2022), and sandwich attacks on Ethereum, by count and profit.

For this, we have used two primary sources. One of them is Dune Analytics, which we use to access insights into atomic arbitrages [70] and sandwich attacks on Ethereum [71]. The other one is EigenPhi which we use to access general data on various MEV strategies [72]. Additionally, we have also made use of results in [52], where we find information regarding atomic arbitrages that have taken place on Polygon, as well as their profitability. It is worth noting that the 23,400 cross-chain arbitrages we detected represent only those occurring between Ethereum and Polygon, whereas Ethereum engages in cross-chain arbitrages with numerous other blockchains. In contrast, atomic arbitrages account for all decentralized exchanges (DEXes) on Ethereum, making this comparison inherently imbalanced. Additionally, the value of $300 million associated with atomic arbitrages on Ethereum (in Table 5.4) does not reflect actual profit but rather the input volume of these transactions.

The Table 5.4 presents a comparison of the frequency and profitability of results obtained from the aforementioned data sources. In atomic arbitrages on Ethereum, stablecoins make up the majority of the extracted value, with WETH, USDe, wstETH, and DAI accounting for over 80% of the all-time extracted volume [70]. This prevalence of stablecoins in in atomic arbitrages is expected because atomic arbitrages are designed to be revertible. On-chain

price discrepancies for highly liquid and well-known tokens persist long enough for these arbitrages to be executed. Additionally, searchers can revert unsuccessful arbitrages without incurring losses, making such tokens preferable for trading in these scenarios. In contrast, as observed in our token analysis, cross-chain non-atomic arbitrages often involve lesser-known tokens. Tokens with high usage rarely retain price discrepancies by the time a searcher completes the asset transfer to the target chain. Additionally, failed non-atomic arbitrages incur costs, such as fees for bridging and coinbase transfers, which cannot be recovered. These risks are further compounded by the challenges of holding inventory on multiple networks, as searchers must maintain sufficient liquidity across chains to execute trades. The bridging process itself introduces timing risks due to the potential for delays, which can lead to missed opportunities or unexpected price changes, increasing the overall risk of holding these assets. These financial risks drive cross-chain arbitrageurs to adopt less aggressive strategies, preferring opportunities with higher certainty of profit. This distinction also explains the stark contrast in both the volume and profitability between atomic and non-atomic arbitrages. However, one notable similarity is the block positioning: both atomic and non-atomic arbitrage transactions tend to cluster toward the top block positions [70]. That said, non-atomic arbitrages are more heavily concentrated in the very top block positions, while atomic arbitrages exhibit a slightly more even distribution across lower indices. This difference arises because non-atomic arbitrages must secure priority in block inclusion to minimize risks associated with price changes during cross-chain transfers. Conversely, atomic arbitrages, being executed entirely within a single transaction, are less susceptible to time constraints, allowing for more flexibility in block positioning.

# 6. Discussion

In our research, we focus on developing an effective strategy to detect and analyze cross-chain arbitrages that have occured between Ethereum and Polygon using the native Polygon PoS bridge between the networks. Over the course of a year, we identified 23,404 successful instances of arbitrage and analyzed their outcomes. Our approach builds upon prior methodologies, such as those outlined in [58], by capturing a broader range of arbitrage scenarios, including complex cases where bridging and swapping operations on the Ethereum leg span multiple transactions.

One notable finding is the imbalance in arbitrage flow, with a majority (58%) of transactions moving from Ethereum to Polygon. This trend can be explained by the shorter bridging times in this direction, which are critical for capturing arbitrage opportunities that depend on rapid execution. In contrast, the longer delays for transfers from Polygon to Ethereum, caused by Polygon's checkpointing mechanism, create obstacles for arbitrageurs aiming to act quickly. These insights could guide future improvements in cross-chain protocol design, with a focus on minimizing bridging times to boost arbitrage efficiency.

We analyzed the searchers conducting cross-chain arbitrages and found that the top six searchers account for 85.68% of the total arbitrages. Our analysis also revealed differing preferences for private transactions among searchers. For example, searcher 0x3fA predominantly uses private transactions, while searcher 0xcA74 favors public arbitrages and relies on Flashbots Protect [41] to defend against MEV attacks, such as frontrunning.

Token usage shows that no single token dominates cross-chain arbitrages. Even so, the prevalence of WETH in cyclic arbitrages emphasizes its role as a staple asset in the Ethereum ecosystem. The shifts in token popularity over time, especially among speculative assets, reveal the fast-paced nature of cross-chain activity. Some tokens see brief spikes in usage due to hype or temporary arbitrage opportunities, while others, like WETH and USDC, maintain steady use because of their stability and strong market presence.

Transparency analysis shows that bridge transactions on Ethereum exhibit a nearly balanced distribution between public (54.65%) and private (45.35%) categories. However, swap transactions are predominantly private, with 70.23% being conducted outside of the mempool. This stark contrast suggests that participants engaging in swaps prioritize confidentiality, possibly to avoid risks of front-running or market manipulation. Additionally, private transactions are more commonly found at the top of block positions.

### Limitations

While our research represents an improvement of prior strategies, such as those which can be found in [58], particularly in widening the range of arbitrage, we have not been able to address

all edge cases. As such, certain limitations should be acknowledged. Recognizing these limitations, shown below, is essential for contextualizing our results and guiding subsequent work in this area.

- One limitation of our algorithm lies in how it identifies swaps. Currently, the algorithm relies on specific events, such as Swap events, that decentralized exchanges (DEXs) emit to signal token exchanges. This works well for direct DEX interactions but becomes challenging when swaps are performed through complex intermediary contracts, such as MeshRouter or Aggregation Router v4/v5. These routers often do not emit the standard Swap events that explicitly show which tokens were input or output. Instead, they generate more generalized or custom events, which lack the detailed information necessary for straightforward swap detection. Our algorithm does not currently incorporate the level of internal transaction tracing, needed to catch such cases, which limits its ability to detect swaps executed through such intermediaries.
- To analyze a full year's worth of Ethereum data, which spans approximately 2.4 million blocks, we need to fetch traces for every transaction within these blocks. This process is inherently time-consuming. A more efficient alternative would be to shift from examining individual transaction traces to examining events emitted directly by the bridge contracts. For instance, by monitoring all ERC20Locked [73] events within a specific block range for a targeted bridge contract, we could easily pinpoint transactions that locked tokens into the bridge. This event-based approach would bypass the need to analyze traces for each transaction, and significantly reduce the latency of data collection.
- In similar fashion, one does not need to rely on transaction traces to get coinbase transfers to block builders, as these can be easily fetched from DuneAnalytics.

**Future Work**

Future research can begin by addressing the limitations identified in this study. Additionally, there is growing evidence that searchers may engage in arbitrage across multiple chains simultaneously, suggesting the need for algorithms capable of tracking such interactions across a broader set of blockchain networks, particularly EVM-compatible sidechains and rollups, such as Arbitrum, and Optimism. Furthermore, researchers could explore arbitrage opportunities that do not rely on bridges, and examine individual searcher strategies to gain deeper insights into their decision-making processes. It would also be valuable to study whether private mempool-like solutions, currently employed in Ethereum, are also being utilized on non-Ethereum networks, and how these may impact arbitrage opportunities and strategies.

# 7. Conclusion

We conducted a comprehensive study on cross-chain arbitrages between the Ethereum and Polygon networks, beginning with a review of the relevant literature to contextualize our work within existing research. Building on this foundation, we developed a methodology to systematically identify and analyze historic cross-chain arbitrages between Ethereum and Polygon.

Over a one-year period, our methodology uncovered 23,404 instances of successful arbitrages, with public arbitrages slightly outpacing private ones. The analysis highlights how protocol design influences transaction dynamics, as demonstrated by the quicker bridging times from Ethereum to Polygon. These shorter confirmation times facilitate a higher volume of arbitrages in this direction, underscoring the importance of network-specific characteristics in shaping arbitrage opportunities.

Our analysis uncovered significant market concentration, with a small group of searchers dominating cross-chain arbitrage activity. Interestingly, the presence of smart contracts seems to favor private arbitrages, while externally owned accounts (EOAs) are more likely to engage in public transactions.

In terms of token usage, we observed a diverse array of intermediary tokens employed in arbitrages, reflecting a strategic avoidance of reliance on widely traded assets. This is likely because price discrepancies in popular tokens tend to resolve quickly, often before a searcher can bridge funds to the target chain. Instead, arbitrageurs appear to favor lesser-known tokens, leveraging their longer-lasting price discrepancies to capture profits.

We also compared cross-chain arbitrages with other MEV strategies, such as atomic arbitrages and sandwich attacks. While cross-chain arbitrages offer unique profit opportunities, they are generally less profitable than atomic arbitrages executed within a single chain. However, they exhibit distinct characteristics, such as increased reliance on bridging mechanisms and greater vulnerability to failed transactions.

Overall, this study advances the relatively underexplored field of non-atomic cross-chain arbitrage by providing new insights into its mechanics, dynamics, and challenges. Future work could focus on refining detection methodologies, and extending the analysis to additional EVM-compatible sidechains and rollups, such as Arbitrum and Optimism, to capture a broader spectrum of cross-chain MEV activities.

# A. Appendix

## A.1. Full addresses of top 6 searchers:

The following are the top 6 searchers sorted by the number of arbitrages they have conducted between Ethereum and Polygon:

0x826A4f4DA02588737d3c27325B14F39b5151CA3C (**SC**)
0x882d04C3D8410dDF2061B3cBA2c3522854316FEB (**SC**)
0x0a6c69327d517568E6308F1E1CD2fD2B2b3cd4BF (**EOA**)
0xcA74F404E0C7bfA35B13B511097df966D5a65597 (**EOA**)
0x3fA6Fff7212D3fA4317cF1955fa690993d8ceD70 (**SC**)
0x84e717025Caa84016B8998A12D8Da9CBf8A8D6E8 (**SC**)

*Note:* "SC" refers to a Smart Contract, while "EOA" refers to an Externally Owned Account.

## A.2. DuneAnalytics query to retreive transparency of transactions

```
WITH tx_hashes AS (
    SELECT hash
    FROM UNNEST(split('{{tx_hashes}}', ',')) AS t(hash)
)
SELECT
    tx_hashes.hash AS "TX HASH"
FROM
    tx_hashes
WHERE EXISTS (
    SELECT 1
    FROM dune.flashbots.dataset_mempool_dumpster d
    WHERE tx_hashes.hash = d.hash
);
```

## A.3. DuneAnalytics query to retrieve coinbase transfer values

```
WITH tx_list AS (
  SELECT FROM_HEX(hash) AS tx_hash
```

```
  FROM UNNEST(split('{{tx_hashes}}', ',')) AS t(hash)
),
block_builders AS (
  -- Get the miner (block builder) address for each block
  SELECT
    b.number AS block_number,
    b.miner AS block_builder_address
  FROM ethereum.blocks b
),
traces_to_builder AS (
  -- Get the traces where the transaction sends value to the block builder
  SELECT
    t.tx_hash,
    t.to,
    t.value,
    t.block_number
  FROM ethereum.traces t
  JOIN block_builders bb ON t.block_number = bb.block_number
  WHERE t.to = bb.block_builder_address
)

-- Select transactions filtered by the provided transaction hashes
SELECT
  traces_to_builder.tx_hash,
  traces_to_builder.to AS block_builder,
  SUM(traces_to_builder.value) AS total_value_sent_to_builder,
  traces_to_builder.block_number
FROM traces_to_builder
JOIN tx_list ON traces_to_builder.tx_hash = tx_list.tx_hash
GROUP BY
  traces_to_builder.tx_hash,
  traces_to_builder.to,
  traces_to_builder.block_number;
```

# List of Figures

# List of Tables

# Bibliography


[1] D. Rodeck. *What Is Blockchain?* Ed. by M. Adams. Accessed: 2024-08-26. 2023. URL: https://www.forbes.com/advisor/investing/cryptocurrency/what-is-blockchain/.

[2] R. Auer, B. Haslhofer, S. Kitzler, P. Saggese, and F. Victor. "The technology of decentralized finance (DeFi)". In: *Digital Finance* 6 (Aug. 2023), pp. 1–41. DOI: 10.1007/s42521-023-00088-8.

[3] 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*. 2019. arXiv: 1904.05234 [cs.CR]. URL: https://arxiv.org/abs/1904.05234.

[4] O. Oluwatosin. *What Is MEV (Maximal Extractable Value) in Crypto?* Edited by Vera Lim, Updated August 19, 2024, Accessed September 3, 2024. 2024. URL: https://www.coingecko.com/learn/what-is-mev-maximal-extractable-value-crypto.

[5] L. Heimbach, V. Pahari, and E. Schertenleib. *Non-Atomic Arbitrage in Decentralized Finance*. 2024. arXiv: 2401.01622 [cs.CE]. URL: https://arxiv.org/abs/2401.01622.

[6] A. Cahill and S. Deshpande. *Layer-2 Scaling Solutions: A Framework for Comparison*. Commissioned by Polygon. The Block Research, May 2022. URL: https://www.tbstat.com/wp/uploads/2022/05/20220505_Layer2ScalingSolutions_TheBlockResearch.pdf.

[7] B. Lashkari and P. Musilek. "A Comprehensive Review of Blockchain Consensus Mechanisms". In: *IEEE Access* PP (Mar. 2021), pp. 1–1. DOI: 10.1109/ACCESS.2021.3065880.

[8] The Investopedia Team. *What Does Proof-of-Stake (PoS) Mean in Crypto?* Ed. by D. Clemon. Reviewed by Doretha Clemon, Fact checked by Suzanne Kvilhaug. Accessed: 2024-08-26. 2024. URL: https://www.investopedia.com/terms/p/proof-stake-pos.asp.

[9] K. E. Wegrzyn and E. Wang. *Types of Blockchain: Public, Private, or Something in Between*. en-US. Aug. 2021. URL: https://www.foley.com/insights/publications/2021/08/types-of-blockchain-public-private-between/ (visited on 08/26/2024).

[10] Cryptopedia Staff. *Ethereum Explained: A Guide to the World Supercomputer*. Updated June 28, 2022 • 5 min read. Accessed: 2024-08-26. 2022. URL: https://www.gemini.com/cryptopedia/ethereum.

[11] O. Network. *What is Ethereum Virtual Machine (EVM) in Blockchain?* Accessed: 2024-08-26. 2024. URL: https://medium.com/@orderlynetwork/what-is-ethereum-virtual-machine-evm-in-blockchain-e7db57167bea.

[12] S. Sen. *How to Access Ethereum Mempool*. Last updated on Dec 11, 2023. 2023. URL: https://www.quicknode.com/guides/ethereum-development/transactions/how-to-access-ethereum-mempool.

[13] Wackerow. *Introduction to Smart Contracts*. Last edited: April 22, 2024, Accessed: 2024-08-26. 2024. URL: https://ethereum.org/en/developers/docs/smart-contracts/.

[14] D. Koutmos. "Network Activity and Ethereum Gas Prices". In: *Journal of Risk and Financial Management* 16.10 (2023). ISSN: 1911-8074. DOI: 10.3390/jrfm16100431. URL: https://www.mdpi.com/1911-8074/16/10/431.

[15] V. Buterin, E. Conner, R. Dudley, M. Slipper, I. Norden, and A. Bakhta. *EIP-1559: Fee market change for ETH 1.0 chain*. https://eips.ethereum.org/EIPS/eip-1559. Accessed: 2024-12-02. 2019.

[16] Q. Huang. "Ethereum: Introduction, Expectation, and Implementation". In: *Highlights in Science, Engineering and Technology* 41 (Mar. 2023), pp. 175–182. DOI: 10.54097/hset.v41i.6804.

[17] Flashbots. *What is MEV-Boost?* Accessed: 2024-12-02. 2024. URL: https://docs.flashbots.net/flashbots-mev-boost/introduction.

[18] L. Heimbach, L. Kiffer, C. F. Torres, and R. Wattenhofer. *Ethereum's Proposer-Builder Separation: Promises and Realities*. 2023. arXiv: 2305.19037 [cs.DC]. URL: https://arxiv.org/abs/2305.19037.

[19] UNOFFICIALbgd. *Transactions*. https://ethereum.org/en/developers/docs/transactions/. Last edit: @UNOFFICIALbgd, July 16, 2024. 2024. URL: https://ethereum.org/en/developers/docs/transactions/ (visited on 08/27/2024).

[20] J. Sharples. *Blocks*. https://ethereum.org/en/developers/docs/blocks/. Last edit: May 26, 2024. 2024. URL: https://ethereum.org/en/developers/docs/blocks/.

[21] T. I. Team. *What Are Crypto Tokens, and How Do They Work?* Ed. by J. Mansa. Accessed: 2024-11-09. 2024. URL: https://www.investopedia.com/terms/c/crypto-token.asp.

[22] N. Reiff. *What Are ERC-20 Tokens on the Ethereum Network?* Accessed: 2024-11-09. 2024. URL: https://www.investopedia.com/news/what-erc20-and-what-does-it-mean-ethereum/.

[23] A. Hayes. *Stablecoins: Definition, How They Work, and Types*. Accessed: 2024-11-10. Reviewed by Doretha Clemon, Fact checked by Skylar Clarine. June 2024. URL: https://www.investopedia.com/terms/s/stablecoin.asp.

[24] E. Rosenberg. *Meme Coins: What They Are, Examples, Pros and Cons*. Accessed: 2024-11-10. Reviewed by Erika Rasure, Fact checked by Vikki Velasquez. Sept. 2024. URL: https://www.investopedia.com/meme-coin-6750312.

[25] C. Sguanci, R. Spatafora, and A. M. Vergani. *Layer 2 Blockchain Scaling: a Survey*. 2021. arXiv: 2107.10881 [cs.DC]. URL: https://arxiv.org/abs/2107.10881.

[26] Crypto.com. *What Are Sidechains? Scaling Blockchain on the Side*. Accessed: 2024-08-31. Feb. 2021. URL: https://crypto.com/university/what-are-sidechains-scaling-blockchain.

[27] Coinbase. *What is Polygon (MATIC)?* Accessed: 2024-08-31. 2024. URL: https://www.coinbase.com/en-de/learn/crypto-basics/what-is-polygon.

[28] J. Ganor. *What Is Polygon (Matic)? An In-depth Guide*. Accessed: 2024-08-31. July 2023. URL: https://www.chainport.io/knowledge-base/what-is-polygon-matic-an-in-depth-guide.

[29] E. B. Observatory and Forum. *The current state of interoperability between blockchain networks*. https://blockchain-observatory.ec.europa.eu/publications/current-state-interoperability-between-blockchain-networks_en. Accessed: 2024-11-09. 2023.

[30] T. Surve. *How to use the Polygon Bridge*. Accessed: 2024-11-09. Dec. 2023. URL: https://cointelegraph.com/news/how-to-use-the-polygon-bridge.

[31] Bitcoin.com. *What Is a CEX?* https://www.bitcoin.com/get-started/what-is-a-cex/. Accessed: 2024-08-30. 2024.

[32] S. Piech. *Centralized and Decentralized Exchanges - What's the Difference?* Nov. 2022. URL: https://research.binance.com/static/pdf/Centralized_and_Decentralized_Exchanges_Comparison.pdf.

[33] A. Jobst, D. Atzberger, R. Henker, J. O. Vollmer, W. Scheibel, and J. Döllner. "Examining Liquidity of Exchanges and Assets and the Impact of External Events in Centralized Crypto Markets: A 2022 Study". In: *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 2023, pp. 1–6. DOI: 10.1109/ICBC56567.2023.10174905.

[34] C. Staff. *What Is a DEX (Decentralized Exchange)?* https://www.gemini.com/cryptopedia/decentralized-exchange-crypto-dex. Updated October 3, 2023. 2023.

[35] Uniswap. *Uniswap*. Accessed: 2024-12-02. 2024. URL: %5Curl%7Bhttps://app.uniswap.org%7D.

[36] A. Lehar and C. A. Parlour. "Decentralized Exchanges". In: *Investments eJournal* (2021). URL: https://api.semanticscholar.org/CorpusID:237189812.

[37] Wackerow. *Maximal Extractable Value (MEV)*. Last edited by Wackerow, July 3, 2024, Accessed September 3, 2024. 2024. URL: https://ethereum.org/en/developers/docs/mev/.

[38] C. Guerini. *Understanding Maximum Extractable Value (MEV) in Crypto: Implications, Challenges, and Opportunities*. Published May 29, 2024. 2024. URL: https://www.linkedin.com/pulse/understanding-maximum-extractable-value-mev-crypto-claudio-dyk0f/.

[39] S. Stankovich. *What Is MEV? Ethereum's Invisible Tax Explained*. Accessed: 2024-11-08. 2021. URL: https://cryptobriefing.com/what-is-mev-ethereums-invisible-tax-explained/.

[40] A. O'Donnell. *Private transactions now dominate Ethereum order flow: Report*. Accessed: 2024-12-01. Aug. 20, 2024. URL: https://cointelegraph.com/news/private-transactions-dominate-ethereum-order-flow-report.

[41] *Flashbots Protect*. Accessed: 2024-12-01. 2024. URL: https://protect.flashbots.net.

[42] *Understanding Order Flow Auctions*. Accessed: 2024-12-01. 2024. URL: https://cow.fi/learn/understanding-order-flow-auctions.

[43] Karl. *MEV Auction: Auctioning transaction ordering rights as a solution to Miner Extractable Value*. https://ethresear.ch/t/mev-auction-auctioning-transaction-ordering-rights-as-a-solution-to-miner-extractable-value/6788. Accessed: 2024-12-01. 2020.

[44] Chainlink. *Maximal Extractable Value (MEV)*. Accessed: 2024-11-08. 2024. URL: https://chain.link/education-hub/maximal-extractable-value-mev.

[45] A. Sergeenkov. "What Are Sandwich Attacks in DeFi — and How Can You Avoid Them?" In: *CoinMarketCap Academy* (2022). Accessed: 2024-09-15. URL: https://coinmarketcap.com/academy/article/what-are-sandwich-attacks-in-defi-and-how-can-you-avoid-them.

[46] CowFi. *What is Frontrunning*. Accessed September 3, 2024. 2024. URL: https://cow.fi/learn/what-is-frontrunning.

[47] CowFi. *What is Backrunning? — MEV Attacks Explained*. Accessed September 3, 2024. 2024. URL: https://cow.fi/learn/what-is-backrunning-mev-attacks-explained.

[48] H. M. Krishnanunni. "Crypto Arbitrage Trading Explained". In: *Mudrex Investment Guide* (2024). Accessed: 2024-09-15. URL: https://mudrex.com/learn/crypto-arbitrage-explained/#:~:text=Definition%20and%20Mechanism%3A%20Crypto%20arbitrage,%2C%20and%20spot%2Dfutures%20arbitrage.

[49] E. Chen, A. Toberoff, S. Srinivasan, and A. Chiplunkar. "A Tale of Two Arbitrages". In: *Frontier Research* (June 2023). Accessed: 2024-09-15. URL: https://frontier.tech/a-tale-of-two-arbitrages.

[50] J. Barragan. *The Fundamentals of Cross-Chain MEV*. Accessed: 2024-10-30. Dec. 2022. URL: https://www.blocknative.com/blog/fundamentals-of-cross-chain-mev.

[51] C. McMenamin. *SoK: Cross-Domain MEV*. 2023. arXiv: 2308.04159 [cs.CR]. URL: https://arxiv.org/abs/2308.04159.

[52] A. Bagourd and L. G. Francois. *Quantifying MEV On Layer 2 Networks*. 2023. arXiv: 2309.00629 [q-fin.GN]. URL: https://arxiv.org/abs/2309.00629.

[53] O. Mazor and O. Rottenstreich. "An Empirical Study of Cross-Chain Arbitrage in Decentralized Exchanges". In: (Jan. 2024). DOI: 10.1109/COMSNETS59351.2024.10426894.

[54] *PancakeSwap*. Accessed: 2024-12-02. URL: %5Curl%7Bhttps://pancakeswap.finance%7D.

[55] *QuickSwap*. Accessed: 2024-12-02. URL: %5Curl%7Bhttps://quickswap.exchange/#/%7D.

[56] K. Gogol, J. Messias, D. Miori, C. Tessone, and B. Livshits. *Cross-Rollup MEV: Non-Atomic Arbitrage Across L2 Blockchains*. Presented at a conference. Oct. 2024.

[57] C. F. Torres, A. Mamuti, B. Weintraub, C. Nita-Rotaru, and S. Shinde. *Rolling in the Shadows: Analyzing the Extraction of MEV Across Layer-2 Rollups*. 2024. arXiv: 2405.00138 [cs.CR]. URL: https://arxiv.org/abs/2405.00138.

[58] D. Ilisei. "Analyzing the Role of Bridges in Cross-Chain MEV Extraction". Master's Thesis. Technical University of Munich, May 2024.

[59] Zeromev. *Zeromev Explorer*. https://info.zeromev.org/explorer.html. Accessed: 2024-11-17. 2024.

[60] Coindesk. *Inside the 'Private Mempools' Where Ethereum Traders Hide From Front-Running Bots*. Accessed: 2024-10-17. Jan. 31, 2024. URL: https://www.tradingview.com/news/coindesk:e9eee8f36094b:0-inside-the-private-mempools-where-ethereum-traders-hide-from-front-running-bots/.

[61] Metachris. *Mempool Dumpster: A Free Mempool Transaction Archive*. https://collective.flashbots.net/t/mempool-dumpster-a-free-mempool-transaction-archive/2401. Accessed: 2024-10-17. Sept. 2023.

[62] Ankr. *Ankr Documentation*. Accessed: 2024-11-12. 2024. URL: https://www.ankr.com/docs/.

[63] Mintlify. *Get Historical Token Prices - API Documentation*. https://goldrush.mintlify.app/docs/api/utility/get-historical-token-prices. Accessed: 2024-10-17. 2024.

[64] Polygon. *Polygon Token List*. Accessed: 2024-11-12. 2024. URL: https://github.com/maticnetwork/polygon-token-list.

[65] Polygonscan. *Polygonscan API - Blocks*. https://docs.polygonscan.com/api-endpoints/blocks. Accessed: 2024-11-17. 2024.

[66] D. Analytics. *Dune Analytics Documentation*. Accessed: 2024-11-12. 2024. URL: https://docs.dune.com/home.

[67] Flashbots. *Flashbots Documentation*. Accessed: 2024-11-12. 2024. URL: https://docs.flashbots.net.

[68] B. Academy. *How to Use the Polygon Bridge?* Online. Published Jan 24, 2022; Updated Nov 11, 2022. Jan. 2022. URL: https://academy.binance.com/en/articles/how-to-use-the-polygon-bridge.

[69] B. Öz, D. Sui, T. Thiery, and F. Matthes. *Who Wins Ethereum Block Building Auctions and Why?* 2024. arXiv: 2407.13931 [cs.CE]. URL: https://arxiv.org/abs/2407.13931.

[70] Hildobby. *Atomic MEV*. https://dune.com/hildobby/atomic-mev. Accessed: 2024-11-21. 2024.

[71] Hildobby. *Sandwiches*. https://dune.com/hildobby/sandwiches. Accessed: 2024-11-21. 2024.

[72] EigenPhi. *EigenPhi Platform*. https://eigenphi.io/. Accessed: 2024-11-21. 2024.

[73] E. Foundation. *ERC-1132: Extending ERC20 with token locking capability*. Accessed: 2024-12-02. 2021. URL: https://eips.ethereum.org/EIPS/eip-1132.