The Elimination of Maximum
Extractible Value

An exploration of MEV elimination & mitigation techniques for the Ethereum Blockchain

John Sinclair
B.A.(Mod) Computer Science
Final Year Project, April 2022
Supervised by Prof. Donal O’Mahony

School of Computer Science and Statistics
O’Reilly Institute, Trinity College Dublin, Ireland
Acknowledgements

I’d like to thank my parents, who have always put an emphasis on my education and supported me throughout. Their constant encouragement has truly been a blessing.

I’d also like to thank Donal O’Mahony, this project's supervisor. Your support and invaluable feedback throughout the course of this project made this experience thoroughly enjoyable, it would not have been possible without your supervision.
Abstract

The Ethereum blockchain and its smart contract capabilities have enabled developers to design and deploy their code to create an entirely new ecosystem of applications. These applications span the full range of complexity, from simple to other more sophisticated systems, such as a full featured domain naming system or entirely on-chain automated financial market makers. Crucially, all of these applications depend on the Ethereum blockchains public and immutable ledger. Its openness and immutability serves as a foundation for their operation and allows users peace of mind when interacting with the variety of offerings available.

However, in its current form the Ethereum blockchain falls victim to a fatal flaw in which unwitting users of the network and its many smart contract enabled applications, can be targeted by highly efficient and increasingly sophisticated bots that lurk within the network’s so-called “dark forest”. These bots await an opportunity to deploy their strategies for financial gain and the capital earned by these bot strategies referenced above is commonly referred to as MEV. Commonly referred to as Miner Extractible Value, MEV involves the manipulation of the transaction orders within blocks of a blockchain. The author aims to examine both how transaction order is manipulated for financial gain, as well as what solutions are available.

This value extraction directly affects the chain’s users, acting as an invisible tax to those who are unaware of this thriving, on-going industry. However, more importantly, this extraction has been shown to pose a fundamental threat to the stability of Ethereum’s blockchain and its consensus mechanism. The implications of instability caused by these operations could threaten Ethereum and its ability to continue as its supposedly incorruptible ledger can be altered to benefit those select groups with the capability to do so.

For the purposes of this paper, the author first examined the fundamental characteristics of Ethereum that make this value extraction possible. They then identified the methods used to extract value by these bots, before concluding with an in depth view of what the author believes to be the two most viable solutions to Ethereum’s MEV issue.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>1</td>
</tr>
<tr>
<td>Abstract</td>
<td>2</td>
</tr>
<tr>
<td>Contents</td>
<td>3</td>
</tr>
<tr>
<td>List of Figures</td>
<td>5</td>
</tr>
<tr>
<td>Introduction</td>
<td>6</td>
</tr>
<tr>
<td>Motivation</td>
<td>6</td>
</tr>
<tr>
<td>Problem Statement</td>
<td>6</td>
</tr>
<tr>
<td>Thesis Objectives</td>
<td>7</td>
</tr>
<tr>
<td>Thesis Structure</td>
<td>7</td>
</tr>
<tr>
<td>State of the Art</td>
<td>7</td>
</tr>
<tr>
<td>MEV Elimination Research</td>
<td>8</td>
</tr>
<tr>
<td>Conclusion</td>
<td>8</td>
</tr>
<tr>
<td>State of the Art</td>
<td>9</td>
</tr>
<tr>
<td>Key Foundational Concepts</td>
<td>9</td>
</tr>
<tr>
<td>Cryptography</td>
<td>9</td>
</tr>
<tr>
<td>Hash Functions</td>
<td>9</td>
</tr>
<tr>
<td>Symmetric Key Encryption</td>
<td>9</td>
</tr>
<tr>
<td>Asymmetric Key Encryption</td>
<td>10</td>
</tr>
<tr>
<td>Network Topologies</td>
<td>12</td>
</tr>
<tr>
<td>Blockchains</td>
<td>13</td>
</tr>
<tr>
<td>Bitcoin</td>
<td>13</td>
</tr>
<tr>
<td>Ethereum</td>
<td>15</td>
</tr>
<tr>
<td>Introduction</td>
<td>15</td>
</tr>
<tr>
<td>Ether</td>
<td>15</td>
</tr>
<tr>
<td>Blocks</td>
<td>16</td>
</tr>
<tr>
<td>Accounts and Transactions</td>
<td>18</td>
</tr>
<tr>
<td>Smart Contracts</td>
<td>22</td>
</tr>
<tr>
<td>The EVM</td>
<td>23</td>
</tr>
<tr>
<td>Token Standards</td>
<td>23</td>
</tr>
<tr>
<td>Consensus Mechanisms</td>
<td>24</td>
</tr>
<tr>
<td>Proof of Work</td>
<td>24</td>
</tr>
<tr>
<td>Proof of Stake</td>
<td>25</td>
</tr>
<tr>
<td>Miners</td>
<td>26</td>
</tr>
<tr>
<td>EIP 1559</td>
<td>27</td>
</tr>
<tr>
<td>Mining pools</td>
<td>28</td>
</tr>
<tr>
<td>Decentralised Finance</td>
<td>29</td>
</tr>
</tbody>
</table>

3
Lending Protocols 29
Decentralised Exchanges 30
Priority Gas Auctions 31
MEV 32
Transaction Frontrunning 33
Sandwich Attacks 34
Arbitrage 34
Time Bandit Attacks 35
Uncle Bandit Attacks 37

MEV Elimination Research 38
Flashbots 39
Flashbots Auction Design 39
Design Goals 39
Technical Architecture 40
The Flashbots Bundle 40
Searchers 41
Relayers 42
Miners 43
Flashbots Protect RPC 44
Solution Evaluation 44
CowSwap 45
Gnosis Protocol 45
Balancer 46
CowSwap Design 49
Solution Evaluation 51

Conclusion 52
Bibliography 54
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Symmetric Key Encryption</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Asymmetric Key Encryption</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Network Topologies</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>Simplified Ethereum Block Structure</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>Ethereum Transaction Object Format Example</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>Ethereum Transaction Request Response Example</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>Ethereum Transaction Lifecycle</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>Top 25 Miners by Blocks</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>Two Bot Priority Gas Auction Example</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>Example Arbitrage Transaction</td>
<td>35</td>
</tr>
<tr>
<td>11</td>
<td>Private Transaction Lifecycle</td>
<td>38</td>
</tr>
<tr>
<td>12</td>
<td>Flashbots Bundle Format</td>
<td>41</td>
</tr>
<tr>
<td>13</td>
<td>Flashbots Auction Infrastructure - Relayer’s Role</td>
<td>42</td>
</tr>
<tr>
<td>14</td>
<td>Flashbots Auction Infrastructure - Miner’s Role</td>
<td>43</td>
</tr>
<tr>
<td>15</td>
<td>Gnosis Batch Auction Closing Price Calculation Visualisation</td>
<td>46</td>
</tr>
<tr>
<td>16</td>
<td>Balancer V1 vs Balancer V2 Centralised Liquidity (Vault) Depiction</td>
<td>47</td>
</tr>
<tr>
<td>17</td>
<td>The Balancer Vault Architecture Overview</td>
<td>48</td>
</tr>
</tbody>
</table>
Introduction

This section exists to convey the motivations behind this thesis and to provide an introduction to one of the largest issues facing Ethereum's blockchain, and its greater community today. Additionally, it will contain a detailed problem statement, the author's research objectives and an overview of the thesis's structure.

Motivation

Typically referred to as MEV, Maximum or formerly Miner Extractible Value, MEV is a problem widely regarded to be amongst the largest fundamental flaws possessed by the Ethereum blockchain that is yet to be dealt with. MEV is a term used to reference the maximum monetary value that is extractible during the block construction process. Miners within the network modify the composition of block’s in order to earn more than the standard reward for their duties. MEV is an issue that is persistent throughout all programmable blockchains and is a result of their Turing completeness and of design inefficiencies at a fundamental level of the protocol’s implementation.

To date the capital extracted from the Ethereum blockchain has amounted to over $600 million, with nearly $10 million extracted in the trailing thirty days alone. While this extraction is certainly a clever usage of Ethereum’s mechanics, it has significant negative effects on the network and its users. It is undoubtedly hugely harmful to both users but also the ethereum consensus mechanism itself. Without proper management by the community, these activities of extraction will inevitably become an increasing risk to Ethereum's economic viability, its immutability and its general potential to be adopted further.

Problem Statement

Developers have devised numerous ways to extract value from Etheruem’s blockchain. While it was previously referred to as Miner Extractible Value, it is often not the miners themselves that directly capture the revenue. Instead, sophisticated bots are developed to engage in the activity of finding profitable opportunities in order to execute them which in turn, allows the miners profit from the increased fees that come as a result. Front running, a strategy that is illegal in traditional financial markets, is a large subset of the value extracted. The front running bots observe the pending transactions in the Ethereum network, searching for transactions that are worth targeting. There are many forms of vulnerable transactions, even other profitable MEV extraction transactions are targeted and front-run in
order to capture the opportunity themselves. Once discovered, the extractor sends their own transaction, paying a higher fee, in an effort to get included in the coming block ahead of the target transaction. However, their options are not limited to simply paying a higher fee, some operate huge infrastructure networks spanning the globe, ensuring proximity and vast reach allowing for wide scale observation. Others operate their own mining nodes or have an established network to pay miners via back channels.

Ethereum’s decentralised exchanges, or more accurately their users are a frequent target for frontrunning attacks. These attacks make users suffer far worse prices and often higher gas prices while the attackers themselves generate risk-free profits. These risk-free profits are known as pure revenue opportunities, and MEV extraction generally is an activity that contributes nothing positive to the system.

**Thesis Objectives**

For this project the author had a number of research goals. Primarily, they wished to determine what methods are currently being employed to reduce the impact maximum extractable value is having on the Ethereum blockchain. In order to do this, the author broke the research into three subsections.

Firstly, the author set out to determine and clarify exactly what MEV is. For many the term is referenced without any true context and so this can lead to a degree of uncertainty. This step required research into many of Ethereum’s characteristics at the protocol level in order to fully understand the extent of the issue and so a number of foundational topics were covered. Secondly, the author wished to understand what methods are currently being used to extract value from Ethereum and other programmable blockchains today. This process is purposely opaque as those extracting do not wish to expose their methods for fear of additional competition. Finally, the author will explore in detail the solutions in development today that are attempting to tackle the MEV issue.

**Thesis Structure**

**State of the Art**

The state of the art aims to bring the reader up to speed on background terms and concepts primarily relating to Ethereum and its protocol implementation. This foundational information is required to fully understand the implications of MEV as well as how current solutions and mitigations are being implemented.
MEV Elimination Research

This section examines what the author believes are the two of the most viable solutions or mitigations for the maximum extractable value issue today, the Flashbots Auction infrastructure and the decentralised exchange CowSwap underpinned by both the Gnosis and Balancer protocols.

Conclusion

To conclude the report, the author provides a reflection on the research carried out and comments on the impact of the studied solutions.
State of the Art

Key Foundational Concepts

The Key Foundational Concepts section will initially give an introduction to basic cryptography. Then, it will introduce the various topologies a network can have, followed by how the two relate to blockchains. These concepts will then be applied to give an introduction to Bitcoin.

Cryptography

Modern cryptography involves the study of mathematical techniques for securing digital information, systems, and distributed computations against adversarial attacks\(^1\). Cryptography is the basis for all blockchains operation as it allows users to interact with each other without directly sacrificing their own security or anonymity, to a degree.

Hash Functions

A core pillar of cryptography, hash functions are single direction functions that take any arbitrary data input, perform the required processing before finally producing a now hashed output of a predefined size. Hash functions are capable of producing vastly varied outputs even when fed inputs with only the slightest differences, this is known as the avalanche effect. This complete non-correlation of input and output data is a crucial and very desirable feature of hash functions. It is important to note that the hashed output value produced by a hashing function cannot be processed in reverse order in an attempt to obtain the original input data. It is this characteristic that allows for asymmetric key pairs to exist. Generally speaking, any size data set can be hashed to produce a representation of a predetermined size, but as mentioned previously, this does not mean the actual data content persists, the hash is merely a representation of the input and cannot be reproduced.

Symmetric Key Encryption

Symmetric key encryption is the simplest form of encryption where implementations make use of linear scramble functions. This means the same key is used to both encrypt and decrypt a piece of data. It is, however, limited due to its capabilities. Primarily, symmetric key encryption is used for the private encryption of data and to share data in a secret or protected manner. Private encryption of data is useful

\(^1\) Katz, J. (2019). Introduction To Modern Cryptography.
for when a user wishes to encrypt their confidential data with a private key that remains a secret to themselves, however issues arise when it comes to the distribution of this encrypted data as communicating the encryption / decryption key safely is not always feasible. Due to the soundness of the encryption process itself, attacks are often made via side-channels that don’t directly attack the cipher but rather its implementations. It is these exploited vulnerabilities that make symmetric key encryption limited in its applications.

Figure 1: Symmetric Key Encryption

In the above figure we can visually see the symmetric key encrypted data sharing process. A common encryption key is used for both the encryption and decryption process. User A simply encrypts the desired data for transmission using the shared key and shares the encrypted message. The receiver, user B utilises the shared key to decrypt the data and reveal the ciphered message, both parties can be sure even if the transmission was intercepted by a third party, it cannot be read provided sufficient precautions were taken during the initial key distribution process.

Asymmetric Key Encryption

Unlike symmetric key encryption, asymmetric key encryption is far more applicable for scenarios where the safe physical distribution of an encryption key is unfeasible. Developed by Whitfield Diffie and
Martin E. Hellman, asymmetric key encryption or public key encryption arose from the need for a reliable method of secure key distribution.

Asymmetric key encryption makes use of key pairs, where each pair consists of a public and private key. These two keys are different, yet mathematically related, as the public key is a derivation of the private key. Crucially, the private key can not be derived in reverse order using the public key. In a public key cryptosystem enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible. The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. Each user of the network can, therefore, place his enciphering key in a public directory².

Figure 2: Asymmetric Key Encryption

In the above figure you can see user A utilises user B’s public key to encrypt the data they wish to transmit. Doing so, ensures that only user B, or someone in possession of their private key will be capable of decrypting the transmitted encrypted message.

Asymmetric key or public key encryption methods are one of the operational foundations for blockchains such as Bitcoin and Ethereum. They have found their primary role as signature verification systems that

allow miners or validators in the network to confirm whether the sender of a transaction request has the required authorisation to do so. Ethereum makes use of the Elliptic Curve Digital Signature Algorithm (ECDSA) to ensure the validity of the transaction request. For Ethereum, each account has a corresponding key pair, where the public key is essentially a hash of the private key; this process is detailed in the Accounts and Transactions section of this paper.

The ECDSA is a variant of the Digital Signature algorithm. The user uses their private key to sign a message or piece of data. This produces a signature which can then be used by anyone who receives it to recover the public key and to confirm the validity of the message sent. If the signature or message hash is altered in any way the resulting public key recovered will be different to that of the original author and so can be invalidated.

Network Topologies

![Network Topologies](image)

Figure 3: Network Topologies

A centralised network possesses one central node to which all peripheral nodes connect to directly, the resulting topology forms a star shape. The centralised network is obviously vulnerable as the destruction of a single central node destroys communications between the end stations. The decentralised network is a marked improvement upon the centralised network as it is far better equipped to deal with security

---

concerns. It is composed of multiple star networks where each star’s central node is itself connected to a number of other central nodes. This allows for communication to continue between unaffected nodes, in the event one or more central nodes goes offline. This network architecture represents the then-current structure of the U.S. telephone network. This is a notable improvement on its fully centralised counterpart yet the decentralised network is still vulnerable in the event a number of the central nodes are attacked. In order to provide true network resilience, each node in the system must be functionally the same to all other nodes as well as maintain multiple links to many other nodes. The distributed architecture allows for multiple nodes to become unoperational or go offline yet still allow the network to communicate and its overall health not too affected.

Blockchains

In essence, a blockchain is a public database that is updated and shared by many computers known as nodes in a network. Blockchains get their name from their composition as chunks of data or state are stored in consecutive, strictly ordered batches known as blocks. These blocks are linked or “chained” together cryptographically where each block references its predecessor or parent block.

As the blockchain is distributed amongst a number of nodes, there needs to be a method for establishing agreement on the state of the blockchain. This agreement is made via what is known as a consensus mechanism. Each node within the network must agree upon each new block and by enforcing this, the blockchain’s state is agreed upon by the entire network.

Currently, Ethereum employs what is known as a proof-of-work consensus mechanism. For proof-of-work blockchains there exists specialised nodes known as miners that use their own computational power to brute force solve a cryptographic puzzle through trial and error. Once successful, they verify and assemble transactions into blocks and add them to the end of their chain. This new chain state is then broadcast to their peers who each verify the validity of this new block. For engaging in this activity, the protocol grants miners the chain's native token as a reward. This acts as an incentive for miners to verify and secure the network.

Bitcoin

Bitcoin is undoubtedly the most recognisable cryptocurrency. During 2008, the world was reeling from a global financial crisis, caused by the collapse of the subprime mortgage market in the United States. It

---

4 Rogers, J.D. (1998). Internetworking and the Politics of Science
undoubtedly led to an increased lack of trust in the financial institutions and banks that are in control of our economies. On October 31st an anonymous programmer or group of programmers known as Satoshi Nakamoto distributed a paper entitled “Bitcoin: A Peer-to-Peer Electronic Cash System” to a cryptography mailing list. Satoshi proposed a way to create a permissionless form of electronic transaction processing and while it was not the world's first cryptocurrency, it was the first to operate without the requirement of a trusted third party.

Satoshi’s vision proposed the new method for nodes to agree upon the state of the network known as proof-of-work. Now used by many other cryptocurrencies, it was this new method for establishing consensus among a network of distributed nodes that allowed the Bitcoin network to prevent various issues and attacks. More on consensus mechanisms in further chapters.
Ethereum

Introduction

Created by the Canadian / Russian programmer Vitalik Buterin, Ethereum differs from Bitcoin substantially. Bitcoin utilised its underlying technology, the blockchain, to create a new form of value transfer, a single application of its potential use case. Ethereum on the other hand aimed to become a worldwide, decentralised computer.

Where Bitcoin uses the blockchain to store transactions and maintain a public payment network, Ethereum makes it possible to store and execute code snippets that are stored on the blockchain. This development enabled an ecosystem of developers to create decentralised and trustworthy applications, composed of smart contracts, stored permanently on the Ethereum blockchain. This decentralised computer enabled by the Ethereum blockchain is known as the Ethereum Virtual Machine or the EVM. A purpose built programming language known as solidity was developed with an emphasis on smart contracts and their functional requirements. Solidity and others such as vyper enable developers to build smart contracts that act as a backbone for the exciting new decentralised applications operating via Ethereum. The decentralised applications are commonly referred to as DApps.

Ether

As Ethereum’s native cryptocurrency, ether is the only acceptable form of payment for users to pay for transaction fees. If a user wishes to make a transaction they must pay a sufficient amount of ether to have their transaction included in the next block, these payments are known as gas fees. Ethers supply is not fixed, however the underlying protocol is the only entity capable of creating or minting new ether, and does so by granting a miner a reward for assembling a new block.

When a user wishes to send a transaction they specify two elements, the gas price and the gas limit. Ethereum or more specifically the EVM charges users for each computational unit used whilst performing or executing a transaction, and so the user must specify the number of units they are willing to perform before the miner should abandon the transaction. This number is known as the gas limit and for context, a standard ethereum transfer transaction, where user A sends an amount of ether to user B, uses 21,000 units of gas. The other component, the gas price, refers to the price in gwei (a subunit of Ether) the user is willing to pay per unit of computation. Ether has many denominations or sub-units but wei and gwei are
the most useful, with wei being the base unit for calculations made in many technical implementations and gwei being the common way to price gas.

Ether can also be effectively destroyed in a process known as burning. Burning ether is not a specific mechanic, the process simply entails sending ether to an address that is known to have no owner and can therefore never be recovered. This address is known as the zero address. This address was responsible for mining the genesis or first block in Ethereum’s chain. Due to the implementation of Ethereum Improvement Proposal 1559 or EIP-1559, this burning occurs during every transaction where what is known as the base gas fee is destroyed, this will be revisited in the Miners section. This dynamic burning of a portion of the gas fees can lead to scenarios where the quantity of ether burned in a block is greater than the quantity that is minted by the protocol. Therefore, Ethereum has a variable inflation or deflation rate. Since its implementation, over six million ether has been burned, equivalent to just under 7 billion US dollars at current market prices.

Blocks

Blocks are used to ensure all participants in the network maintain a synchronised state, this allows the all nodes to agree on the precise history of transactions made throughout the history of the chain. Each block is composed of a batch of transactions, bundled together by miners as they create blocks. Within the chain, blocks are strictly ordered, this means that each block directly references its predecessor or parent block. Additionally, the individual transactions within the blocks are also strictly ordered meaning a transaction can have a knock on effect to any subsequent transactions that follow. Blocks are constructed and propagated to the other nodes within the network where each peer node verifies the validity of it and adds it to their end of the chain.
The above figure is a simplified depiction of blocks but Ethereum blocks contain a number of additional values. Each Ethereum block consists of the following fields:

- **timestamp** - the timestamp refers to the time the block was mined. This is measured in seconds since the epoch.

- **blockNumber** - the blockNumber field, sometimes referred to as the block height, refers to the length of the blockchain in blocks. It acts as an incremental tally.

- **baseFeePerGas** - the baseFeePerGas refers to the minimum fee per gas that is required for a transaction to be included in a given block.

- **difficulty** - the block’s difficulty is an indication of the effort the successful miner performed to mine the block. This is an important field as the protocol is capable of modifying the difficulty of future blocks in order to maintain an approximate constant block time.

- **mixHash** - the mixHash is a unique identifier for that block. This is what will be referenced by the following block, linking itself to its parent.
**parentHash** - as stated previously in the mixHash, the parentHash field contains the previous block in the chains mixHash.

**transactions** - this section contains all the transactions that have been validated and executed within this block.

**stateRoot** - the stateRoot is a representation of the entire state of the system including account balances, contract storage, contract code and account nonces. Its inclusion in the blocks composition allows for what are known as light clients, clients that do not store the entire chain's history. It is a verified snapshot that allows low resources clients, such as those to be run on lower powered machines, to request confirmations or proofs of individual balances or the state of a smart contract, without having to download the full chain state or verify every individual transaction.

**nonce** - the nonce is simply a hash that when combined with the mixHash proves to other nodes that the block has successfully been produced via proof-of-work.

Accounts and Transactions

In the case of Ethereum, there are two forms of account that can exist within the system, externally owned or user accounts and contract accounts. They act the same for the majority of cases with both forms having the ability to send, receive and hold ether and other tokens. Crucially, both user accounts and contract accounts can send transactions to interact with the many deployed smart contracts stored on the blockchain. However, they do differ when it comes to transaction initiation. Only user accounts can send transactions as they please, smart contract accounts only have the ability to do so in response to receiving a transaction from an externally owned account. This means a user account must trigger a smart contract account in order to send any transactions transferring ether and tokens or to call other smart contracts to perform various other operations. All Ethereum account consist of the following fields:

**nonce** - the nonce acts as a tally used to indicate how many transactions the account has sent. By having this field as part of the account structure, it is ensured that transactions are only processed once. While ethereum transactions are irreversible provided they are yet to be added to the chain by a miner, they can be updated. By using the same nonce as a pending transaction, a user can replace the pending transaction by increasing the gas price. This means that when a transaction gets stuck in the pending state due to network congestion, users can bump up their gas price to have it processed sooner. Transactions can also be cancelled by replacing the unwanted transaction with another of the same nonce. By using a higher gas
price and a transfer of zero ether, the now cancelled transaction is essentially replaced with a pointless transaction.

**balance** - the balance is a straightforward field, it documents the balances of the account denominated in wei, a subunit of ether.

**codeHash** - the codeHash field really only applies to contract accounts as it refers to the code of an account on the EVM. smart contracts have code fragments programmed into them that are executed when triggered by a transaction. Importantly, this field is immutable, meaning it cannot be changed once the contract has been deployed. For user accounts, the codeHash is just the hash of an empty string.

**storageRoot** - the storageRoot, sometimes called the storage hash, is a 256-bit hash of the root node of a merkle patricia trie. This hash encodes the storage contents of the account, it is empty by default but the trie encodes the hash of the storage contents of the account.

Externally or user owned accounts consist of key pairs, a public key and a private key. In the case of Ethereum, the private key consists of sixty four hexadecimal characters. From this, the public key is derived using the elliptic curve digital signature algorithm. The derivation utilises the last twenty bytes of the keccak-256 hash of the private key and prepends it with a “0x”. The resulting value is a forty two character hexadecimal address. As mentioned previously, due to the nature of hash functions, the private key cannot be obtained using the public key. The private key is also used to verify ownership of a user account. Those in possession are the rightful owners of the account according to the protocol. It is used to sign transactions, preventing forgeries, but granting anyone in possession custody over the funds within the account.

When a smart contract is deployed from an externally owned account, a contract account is created. The newly created contract address is a derivation of the creator’s address and the creator’s account nonce.

Transaction can be initiated either directly by an externally owned account or indirectly by a smart contract account. They are actions that if successful, will cause a state change throughout the network. Any node with the Ethereum network can broadcast a request for a transaction to be executed by the Ethereum Virtual Machine. From there, a miner will select the transaction, execute it and then propagate the resulting state change throughout the network via its peers. Transactions only become valid once mined and included in a block.
Figure 5: Ethereum Transaction Object Format Example

The above figure shows an example of the transaction object format. The transaction object must be signed using the owner’s private key in order for the transaction to be successfully verified and included.
Figure 6: Ethereum Transaction Request Response Example

The above figure shows an example response received when sending a transaction request via an RPC endpoint. The raw section contains the signed transaction in Recursive Length Prefix (RLP) encoded form.

There are two types of Ethereum transactions. Standard transactions are simply from one account to another, these include the transfer of ether or token or the calling of smart contract functions, which don’t necessarily require an ether transfer. The other form of transaction occurs when deploying contracts. Contract creation transactions possess no “to” field and the data field is instead used for the code of the smart contract itself.

---

The purpose of recursive length prefix is to encode arbitrarily nested arrays of binary data (eth.wiki).
Upon sending of a transaction, a hash is generated unique to it. The transaction is then broadcast and added to the network mempool. The network’s mempool is essentially a list of pending transactions that are yet to be processed by a miner. From there a miner must select a transaction, verify it and include it in a block in order to consider it successful. Note, the selection process is not arbitrary and delays in transaction processing can occur if the specified gas price is insufficient. When successful, transactions receive confirmations, this number refers to the number of blocks that have been added to the chain since the block the transaction was included in.

The number of confirmations can be important as recent blocks can get re-organised, however the probability of a block re-organisation occurring diminishes with every subsequent block mined as the capital requirement to undertake such a process would become unreasonably large.

Smart Contracts

The Turing complete$^6$ EVM and its smart contracts capabilities enable developers to build and deploy arbitrarily complex decentralised applications and services ranging from games to marketplaces and sophisticated financial instruments.

Smart contracts can be described as “a set of promises, specified in digital form”.$^7$ In essence, smart contracts are reusable snippets of code, stored directly on the Ethereum blockchain. They are defined as “digital, computable contracts where the performance and enforcement of contractual conditions occur automatically, without the need for human intervention”.$^8$ In the case of Ethereum, smart contracts allow users to send transactions requesting the execution of these code snippets with varying parameters. They are a core innovation differentiating Ethereum from its other blockchain based predecessors such as Bitcoin.

Anyone with the technical knowledge to do so can write and deploy their own smart contract by paying a fee to the network. By deploying the contract it becomes visible to the network. Once deployed, accounts can request the execution of the contracts code via transactions. Note, while deployed smart contracts are visible to the entire network, some are designed in such a way as to limit usage to certain users.

---

$^6$ Turing complete, allows for encoding of arbitrary smart-contract functionality (Philip Dain et al, Flash Boys 2.0, 2019).


The EVM

The Ethereum Virtual Machine, known as the EVM is a “single, canonical computer whose state everyone on the Ethereum network agrees on”⁹. Within the network, each participating node keeps its own copy of the current EVM state. This means any participant or node can broadcast their own request for this global computer to perform arbitrary computation. Once selected for inclusion and mined, these requests cause a network wide state change that is propagated throughout the network.

Ethereum’s network state is stored as a record of the entirety of all transactions made throughout the history of the chain. The EVM’s present state gets stored on the blockchain meaning it is agreed upon by all nodes. The state is an enormous data structure known as a modified Merkle Patricia Trie. This trie keeps all accounts linked via hashes and crucially, is compressible to a single hash known as the single root hash, which is stored on chain.

There exist cryptographic mechanisms at the protocol level that attempt to ensure that once a transaction is verified as valid and added to the blockchain it is immutable. This immutability is a core foundational characteristic that Ethereum is built upon, but as the author will show, there are threats to this finality.

By utilising a gas price, denominated in ether, for transactions requests, the EVM has its own market where users can bid to use the finite resources provable by the network. The market structure has provided an economic incentive for the nodes and participants in the network to provide their computational resources to Ethereum to verify and execute transactions. As previously mentioned, each transaction request must specify a gas price. The use of these bounties ensures malicious or negligent participants cannot clog the network by requesting the execution of infinite computation as they pay for computational units.

Token Standards

Token standards are community designed and agreed upon formats for tokens to allow for applications to be interoperable. While there are numerous standards for the various forms of Ethereum tokens such as standard coins or NFTs which themselves are typically just either ERC-721 or ERC-1155 tokens, for the purposes of this paper and its scope only ERC-20 tokens will be defined.

The ERC-20 token standard is a widespread format for fungible tokens. An ERC-20 token acts fundamentally identical to ether, and no two tokens minted by the same contract will differ. The reason

ERC-20 tokens are of importance is because they are the most adopted token standard throughout Ethereum ecosystem, with majority of decentralised exchanges utilising them and a tokenized version of ether, wrapped ether, for their operations. By using wrapped ether or WETH, decentralised exchanges can reduce the size of their codebase or smart contract base effectively in half as they only have to facilitate operations in one token standard.

Consensus Mechanisms

Consensus mechanisms allow distributed networks to work together and agree upon a network's state, in a secure way. The definition of consensus is when a majority of nodes in a network agree on the global state. So when 51% of nodes agree upon the network's state, the nodes have reached consensus.

Over time various processes have been used to establish consensus among database nodes and other enterprise infrastructure, but the advent of new consensus mechanisms such as proof-of-work have allowed a new generation of distributed economic systems such as Bitcoin and Ethereum to agree upon the state of the network. Without these mechanisms or algorithms, economics attacks are viable methods to alter the state of the chain. Currently, Ethereum employs a proof-of-work consensus mechanism however it will transition to a new mechanism known as proof-of-stake after an event referred to as the merge, however this is out of the scope of this paper.

Proof of Work

Mining nodes for the Ethereum network must spend a substantial, but variable, amount of energy and computational power to produce what is known as a certificate of legitimacy. This certificate proves the required work was performed and gives the producer the right to propose the forthcoming block in the chain and the expense of production prevents a myriad of attacks. Once produced, the miner propagates their newly constructed block to the network via its peers, when other miners hear about any new block with a valid certificate they must accept it as the canonical next block in the chain.

There is a protocol in place for dealing with the situation where two or more miners produce the next block simultaneously, however as the exact time required to produce the certificate of legitimacy is determined by a random variable with a high variance, this event is highly unlikely. As stated previously, the block time refers to the time required to mine a new block, the current average is between 12 and 14 seconds. This average is maintained by the protocol, which evaluates the time after each block. The protocol compares the latest block time to the protocol’s expected block time, which is a set constant at the protocol level. This constant is used to protect the security of the network against the inevitable
increase of computational power possessed by miners, however it also protects against the possibility there is a decrease in the computational mining power. The protocol's evaluation process entails comparing the average block time to its constant and if the average is higher than the expected value, the expected difficulty is decreased, conversely, if the average is less than what is expected the difficulty is increased in the block header.

Due to variable network demands, Ethereum’s blocks vary in size however they do have an upper boundary. The protocol targets a block size of fifteen million gas but it does allow for expansion up to thirty million during times of high usage. The total gas expended by all transactions in a block must be less than this block gas limit. This limit ensures blocks cannot be of an arbitrarily large size, as if that were the case, the hardware requirements to run a node would increase which would in turn raise the barrier for entry to operate and reduce decentralisation.

Proof of Stake

Due to the concerns surrounding many proof-of-work blockchain’s enormous environmental footprint, more environmentally friendly methods of consensus are in constant development. Proof-of-stake is a marked improvement, dramatically reducing energy requirements for validating nodes. By removing the hardware and energy intensive process of mining, one of the barriers for entry for prospective nodes is dramatically lower. This should in theory increase the network's decentralisation as more casual users and enthusiasts will be capable of running their own nodes to maintain the network, however there are some caveats such as the large financial requirements due to the protocols minimum stake requirements.

While proof-of-work consensus mechanisms utilised the energy and computation power requirements to ensure malicious behaviour was discouraged, proof-of-stake systems need to develop a new method of ensuring good behaviour. Known as validators, nodes in the proof-of-stake network do not engage in the activity of mining. Instead they deposit or “stake” a quantity of funds, predetermined by the protocol, that acts as insurance ensuring their good behaviour. While miners utilise their computational power to produce a certificate of legitimacy for a new block proposal, validators are instead chosen at random. If not chosen, validators are responsible for validating and confirming new blocks. These duties are known as attestations and validators earn protocol rewards for attesting as well as proposing new blocks. In an effort to increase decentralisation, the protocol design decision to cap the effective balance of a validator at 32 ether was chosen. The effective balance means that earned rewards will be the same for validators with more than 32 ether and those with just 32 ether.
Finality or immutability is an important feature for a distributed network as this ensures transactions that have been processed cannot change. For Ethereum’s migration to its new consensus mechanism, Casper, a finality protocol was developed. Casper uses checkpoints to ensure validators agree on the state of the chain. When two thirds of the network's validators agree on the state of the chain at a checkpoint the block is finalised. If any malicious validator attempts to revert this finality they lose their entire staked balance.

Miners

In proof-of-work systems miners perform an integral role. They are nodes that engage in the process of collating transactions into blocks to be added to the blockchain. By running specific softwares such as the go-ethereum (geth) or erigon clients they use their time and computational power to process transactions and produce blocks.

For fundamental operation decentralised networks require consensus, it is imperative all nodes agree or at least a majority agree on the network’s current state. If this is not the case users can potentially spend funds they should not have access to. When large subsections of the mining nodes disagree with one another, a fork in the network can occur, such as the fork witnessed in the aftermath of the DAO hack. By securing the network via computationally difficult puzzles, it ensures it is prohibitively expensive to alter the history of the chain maliciously.

The mining process is as follows: when a new transaction request is broadcast to the network it first awaits inclusion in the mempool. The mempool is a dynamic list of all transactions that are yet to be included in a block. A mining node will then select and aggregate a number of transactions into a potential block. This selection for inclusion when done correctly is done in a way that maximises their reward via received transaction fees, by selecting those with the highest gas prices. Note, the transactions must all remain under the block gas limit. However, miners are incentivised to maximise their reward. The miner then verifies the validity of each request, executes the required code and alters their local copy of the blockchain assigning the transaction or gas fee to themselves. Once a miner is successful in producing a valid certificate of legitimacy the miner broadcasts it along with their completed block and adds a checksum of their new altered EVM state. When other miners hear about this newly created, valid block they must accept it as the next in the chain. They then execute all the transactions within the block and verify that the state of their local EVM matches the checksum provided. If the provided checksum is a match, the miner appends the new block and removes all the transactions from the block from their local mempool.
Figure 7: Ethereum Transaction Lifecycle

The above figure graphically shows the transaction lifecycle, outlining the path a transaction takes from initial request all the way to block inclusion.

New nodes joining the network must catch up with the other nodes and the current network state. They do so by first initialising their own, blank copy of the EVM before repeating the process above for each block in the chain’s history. By executing every transaction in each block and verifying checksums that can be assured they have a correct snapshot of the network’s state.

EIP 1559

EIP-1559 or Ethereum Improvement Proposal 1559 introduced a new form of pricing mechanism for transactions. Before this proposal was implemented Ethereum utilised a simpler gas price mechanism where users would only specify a gas price and limit. This led to a number of issues such as, when users specified their maximum gas price, if selected that was the user was required to pay the bid amount. This led to users typically overpaying for transactions.

EIP-1559 introduced a base fee whereby a portion of the fees for each transaction is burned rather than awarded to miners. The proposal introduced the idea of a base fee per gas for the protocol’s current block. This base fee is dynamic, meaning it can increase or decrease with each block. This allows the protocol to deal with temporary on-chain congestion by increasing the fee when the block's ideal gas limit is exceeded, or reduced when the ideal gas limit is not reached.

Due to this proposal, transactions now specify a priority fee, a fee to be paid to miners, and a max fee per gas, which refers to the amount they are willing to pay in total for their transaction. The sum of the protocol base fee and the users priority fee must be less than the user specified max fee per gas. If valid, once a transaction is included, the priority fee is awarded to the miner and the base fee is burned.
This proposal has undoubtedly led to more consistent gas prices with reduced fluctuations, but it has also led to contempt within the mining community. Since its introduction, EIP-1559 has drastically reduced the profitability of mining leading to miners pursuing other avenues to make up for their lost revenue.

Mining pools

Due to the protocol’s design, there exists a degree of luck when attempting to successfully produce the certificate of legitimacy required by the network to propose a new block. This has given rise to what are known as mining pools.

Mining pools are large groups of miners who have organised themselves such to enable their members to pool their resources and effectively mine as a group. By mining as a collective, each additional member of the pool increases the chances that a pool miner is successful in producing the next block. Block rewards earned by members of the pool are distributed amongst the members proportional to the hashing power provided by each member. Note, the pool itself as organiser typically takes a percent of the rewards earned. Unfortunately, this mass organisation has its drawbacks and has led to a clear centralisation issue for the Ethereum ecosystem.

Figure 8: Top 25 Miners by Blocks, from etherscan.io
As you can see in the above figure from Etherscan, huge percentages of the network's total hashrate is controlled by these mining pools. So much so that during the trailing seven days, over half of the network's mined blocks have been produced by just three mining pools, a concerning metric for many. This degree of centralisation of the network's hashing power is concerning as the possibility for censorship or other malicious activity is far greater with so few authorities in control.

Taking Ethereum’s largest mining pool Ethermine as an example of such issues, they who mine just under 30% of Ethereum’s blocks recently decided to prohibit decentralised exchange front running\(^{10}\). The intricacies of such activities will be detailed later in this paper but for many this was seen as the crypto equivalent of opening Pandora’s box. While this act of censorship is positive for Ethereum’s users, it has set a precedent for mining pools to use their power to actively censor any transactions they desire.

Decentralised Finance

Decentralised finance or DeFi is a subset of financial applications powered by Ethereum’s smart contracts. They operate independently, free from direct regulation and at the time of writing, have a total value of nearly 80 billion US dollars\(^{11}\) locked or in use throughout their protocols. The two most common forms of decentralised finance application are lending protocols and decentralised exchanges.

Lending Protocols

Lending protocols are a colossal subset of DeFi accounting for nearly half of the total value locked (TVL) in protocols. These protocols offer loans backed by collateral similar to those issued by traditional banks but as they operate without knowledge of their customer they must implement over-collateralized loans to protect their funds. While this seems counter-intuitive, many do not wish to sell their crypto assets and risk losing exposure to their price movements and so some use them as collateral to borrow other coins, typically stablecoins which are coins that are tied usually to the price of the US dollar. Others can use these loans as a way to leverage their position in a particular asset. Users typically lock up more than 120% of the value that the borrower wishes to lend out. This collateral acts as an insurance in the event the loanee fails to return the borrowed assets. If the value of said collateral decreases such that the loan’s collateral ratio falls below the required collateralization ratio, the collateral is freed up for liquidation. This allows those with the capacity to do so, to purchase the collateral at a discount in order to repay the debt. The top three lending platforms on Ethereum today are Maker, Aave and Compound, with Maker accounting for 20% of the TVL alone.

---

11 defipulse.com. (n.d.). DeFi Pulse - The Decentralized Finance Leaderboard
An important innovation facilitated by Ethereum’s lending protocols is the flash loan. A flash loan is an entirely uncollateralized loan that is provided by the lending protocols significant capital. This capital is obtained by the tokens and ether deposited by the users of the lending protocol. The protocol can implement these uncollateralized loans risk free by using codified guarantees that make it impossible for a borrower to take a loan and not repay it. This guarantee is possible as the repayment must be repaid within the same transaction that took it out, with an additional small fee. Ethereum allows for multiple operations to occur within one transaction and so users wishing to take out a flash loan must make the request and repayment within the same transaction. If the transaction does not include these two steps the whole transaction is reverted.

Decentralised Exchanges

The other large subset of decentralised finance is decentralised exchanges or DEXs. DEXs account for approximately 25% of the decentralised finance total value locked. The most dominant iterations of decentralised exchanges employ what is called an automated market maker design. Early iterations of decentralised exchanges and traditional exchanges maintained their orderbook of bids and asks off-chain. Automated market makers differ here by using smart contracts to manage a pool of liquidity, typically consisting of but not limited to two assets, that allow users to trade assets via predefined rules and functions. These predefined rules specified within the smart contracts outline the conditions for when a user can buy from or sell to the liquidity pool. “At first glance, decentralised exchanges seem to be ideally designed. They appear to provide effective price discovery and fair trading, while doing away with the drawbacks of centralised exchanges”

Due to their design, automated market makers price their trades depending on the available liquidity within the smart contracts pool of equity. This pool typically consists of two unique tokens and so when one is removed in favour of another due to a user executing a trade, the purchased token quantity decreases proportionally to the sold token’s increase in quantity. This ratio of tokens within the pool is used with a constant product formula to determine the price for swaps.

The entirely on-chain nature of automated market makers has huge advantages when it comes to decentralisation and asset custody, or lack thereof, but also leads to large disadvantages as they are fundamentally slow to execute orders. Additionally, this on-chain execution of orders leads to a problematic level of order transparency, exposing profitable pure revenue trades by publicly broadcasting

---

them to the Ethereum mempool. The top three decentralised exchanges in the space are Curve Finance, Uniswap and Balancer, but there are numerous others.

Priority Gas Auctions

Due to the fact that decentralised exchange order details are made public before they are executed, the competition between bots executing pure revenue opportunities has led to a phenomenon referred to as priority gas auctions. Miners select pending transactions to be placed in the next block based upon transaction price, favouring highest first and remaining in descending order. This means within the block the transaction with the highest gas price is executed first and each following transaction has either the same or decreasing gas price. Priority gas auctions or PGAs emerge when bots, typically arbitrage, compete with each other over a pure revenue opportunity, bidding increasing transaction fees in the effort to have their transaction win and get included in the forthcoming block by the miners. A priority gas auction process involves two or more bots issuing numerous transactions with the same nonce but with each transaction expressing an incrementally higher gas fee. This eventuality resembles an auction as the bots are bidding to be the auction winner who has their transaction included.

Bot operators are encouraged to bid up gas prices to be the auction winner due to the mechanics of Ethereum’s protocol, miners are incentivised to select the transactions from the mempool with the highest gas prices. Additionally, a portion of valuable block space is taken up by failed transactions, those who engage in the PGA but do not win still pay a fee to have their attempt included, but as they did not win their transaction will fail. This artificially raises the prices of any blocks where PGAs are underway. “In game theoretic terms, each auction represents a variant of all pay auction, where instead of paying their full bid, the loser is forced to pay a percentage to the miner”13. PGAs are a variant of an all pay auction, an auction where every bidder must pay regardless of whether they win. Hence PGAs are all pay auctions due to the fact that all bidders engaging pay a fee, even if not the winner the auction loser has their failed transaction included in the block by the miner due to the inflated gas price they paid.

---

Figure 9: Two Bot Priority Gas Auction Example, from Flash Boys 2.0, Philip Dain et al, 2019

The figure above depicts a priority gas auction between two bot operators each represented by their own colour, blue or orange. Their transactions are represented by the orange triangles and blue circles. Each point on the graph indicates a new transaction request. The gas price bid is denominated on the y-axis and the time elapsed since the auction began on the x-axis. These two competing bot operators issue multiple orders per second, a total of nearly 90 bids between them are placed in the space of 14 seconds. The transactions marked with the green star and red square are the transactions from each bot operator that are included in the next block. The subsequent transactions submitted after the winning bids were selected are due to the latency between transaction submission and the block being mined.

MEV

MEV is a term used to reference the maximum monetary value that is extractible during the block construction process. By modifying the block’s composition to suit particular strategies, the capital earned can be greater than the standard block rewards and earned transaction fees. Miners do so by including, excluding, replacing and re-ordering the transactions within the block. These miner specific profits are known as ordering optimization fees but they fall under the category of MEV.

MEV, formerly referred to as Miner Extractible Value, is now known as Maximal Extractable Value due to a number of reasons. Firstly, miners are being phased out of Ethereum during its transition from proof-of-work to proof-of-stake, when the term was initially coined, miners had the most power when it came to transaction selection and ordering within block and so it was termed miner extractible value. Secondly the majority of value extracted is typically not actually captured by the miners but by bot operators, the miners simply profit from the inflated gas prices, from data obtained from the Flashbots MEV-explore dashboard we know searchers or bots capture about 65% of the gross profit earned.
There exists a time difference between when a user broadcasts a transaction request to the network and when it is actually finalised and included in a block. During this time difference the transaction requests are exposed to the public as they await inclusion. During this delta arbitrageurs, frontrunners and miners monitor the mempool in search for pure revenue opportunities. If the frontrunner also happens to be a miner themselves, they can alter the order or censor specific transactions within the block.

Unfortunately, MEV directly harms Ethereum’s users, it acts as an invisible tax on unsuspecting users that are unaware of the booming economy surrounding the exploitation of Ethereum’s mechanics and those who don’t have the knowledge of how to reduce their exposure. Not only this, but as the financial reward for engaging in such activities can be greater than that of the protocol’s block reward, there exists an incentive for miners to destabilise Ethereum’s consensus mechanism by reorganising and censoring transactions within blocks to suit their agenda. This reorganisation is known to occur to approximately 1.3% of Ethereum’s blocks\(^\text{14}\).

Taking a look at the MEV-explore dashboard provided and maintained by Flashbots gives an insight into how large an issue this is. Since the research groups monitoring efforts inception, they’ve recorded approximately 600 million US dollars worth of MEV extracted from the Ethereum chain, with the majority of this profit being captured by arbitrage bots targeting various automated market makers. Note, this is also a lower bound estimate as there exists limitations when it comes to the available data and so the group has limited their calculations to single transaction opportunities and entirely on chain activities.

It is important to note that not all MEV is bad, some activities actually benefit the crypto economy. For example, arbitrage bots actively balance automated market makers prices, providing fairer, more balanced decentralised exchange prices across the ecosystem.

**Transaction Frontrunning**

Frontrunning is likely one of the most straightforward methods to extract value from Ethereum. Frontrunning takes advantage of the time a transaction request remains in the public mempool. When a transaction request is made it remains in the mempool until selected by a miner. It is important to note that an Ethereum block contains many different transactions that are executed in sequential order. This order is supposed to be determined by gas price but isn’t always the case. This transaction order can be manipulated for risk-free profit. When a bot surveying the mempool observes a transaction that they determine to be profitable, the bot can submit an identical transaction with a higher gas price, capturing the profitable opportunity.

\(^\text{14}\) Calculated by dividing the number of reorganised blocks by total number of blocks mined, data from etherscan.io
Sandwich Attacks

Another form of attack that utilises both the front and back running of transactions is known as a sandwich attack. Sandwich attacks take advantage of a user's slippage tolerance when attempting to swap tokens using an automated market maker. Slippage refers to the difference between the expected price of a trade and the price at which the trade is executed\textsuperscript{15}. In essence, slippage details the minimum number of tokens the user is willing to accept in receipt for their trade. For example, given a slippage tolerance of 2\%, a user could submit a swap transaction request to buy 1 ETH at a price of 2,900 USDC, and still get their order filled at a price of 2,958 USDC per ETH. A sufficient slippage tolerance is required for on chain market markers for without, the frequent price movements would mean few swap transaction requests would be filled due to the delta between the time the transaction request is sent and when it is included in a block.

A sandwich attacker exploits this mechanism by placing a transaction of their own before and after a transaction with a vulnerable slippage tolerance. The purpose of these two sandwiching transactions is to manipulate the price of the token. The attacker first sends a transaction with higher gas buying a large quantity of a particular token. This large purchase, that often utilises a flash loan, causes a spike in the price of the token. The attacker then sends another transaction, this time pricing the gas sufficiently low to ensure it gets included after the original transaction. This second attack transaction sells the entire balance of the token purchased in the first transaction to the unsuspecting original user allowing the attacker to pocket the small percent difference in price. Taking the example used above, in theory the attacker would profit 58 USDC minus the gas fees paid to the miners for the pair of transactions used to sandwich the original transaction. In reality the gas prices would likely eliminate the profit generated from this attack but scaled up to say 10 or 100 ETH the profit opportunity is evident. While this method may not be the most profitable strategy on a once off basis, sandwich attackers often repeat this process hundreds, if not thousands of times per day.

Arbitrage

With over 99\%\textsuperscript{16} of MEV extraction transactions being arbitrage related, arbitrage strategies are by far the most popular methods employed to extract value from the chain. In their simplest form, arbitrage strategies involve the execution of simultaneous transactions to both buy an asset from one market and to sell that same asset in another market. The strategy takes advantage of pricing differences for the same asset across different markets and decentralised exchanges. By purchasing a token at one market and

\textsuperscript{15} investopedia (2019). Sharper Insight. Smarter Investing.
\textsuperscript{16} explore.flashbots.net. (n.d.). MEV Explore.
selling at another for a higher price simultaneously, the bot pockets a risk free profit in the form of the difference in price.

Figure 10: Example Arbitrage Transaction, transaction details from etherscan

In the above figure we can see an example of an arbitrage transaction. This arbitrageur executed three actions within the one transaction, profiting 45 Ether. The bot first flash loaned 1,000 Ether from Aave, it then swaps this ether for 1.3 million DAI using the decentralised exchange uniswap. DAI is an algorithmically managed US dollar stablecoin. Next, the bot swapped the DAI obtained from uniswap for 1,045 Ether on sushiswap. Once the flash loan was repaid with the required interest, the bot operator profited a risk free 45 Ether or approximately 140,000 US dollars at current market rate.

Arbitrage activities are actually an important part of the greater crypto-economy as without these bots constantly managing liquidity pools, large price differences would develop across the space. However, there still exists a number of negative externalities, such as the reduction of valuable blockspace due to failed transactions and inflated gas prices due to arbitrage bots competing for the same opportunity.

Time Bandit Attacks

Time bandit attacks are a strategy that involves the modification or reorganisation of past blocks, a clear threat to Ethereum’s immutability and finality. There exists an incentive for miners to rewrite the chain’s history in order to capture profitable opportunities if the MEV or fees earned through optimising the order of transactions within the block becomes larger than that of the protocol’s block reward.

These forms of attacks lead to instability in the blockchain and can theoretically lead to forks of the chain itself where those who possess large enough mining computational power, or those with the ability to rent it, can rewrite the chain’s supposedly immutable block history, funded by the profit obtained from both

---

executing the pure revenue opportunities themselves and by claiming the gas fees paid by the competitors failed transactions. If the reward for mining a block correctly is small enough when compared to the MEV opportunities that are executed within, miners could in theory decide to destabilise the protocol’s consensus and rewrite the blockchain’s history. It is unclear how frequent these attacks occur due to the fact not all block reorganisations are a result of time bandit attacks. However, 118 blocks have been reorganised in the last 24 hours.\(^{18}\)

Once transitioned to proof-of-stake, Ethereum will possess a number of structural protections against time bandit attacks. The primary method for ensuring good validator behaviour will be the protocol penalising or slashing a validators stake if misbehaving or engaging in malicious activity. The protocol administers their penalties automatically and they vary in severity, for example in the event a validating node goes offline, the stake is penalised a minimal amount to encourage its prompt upkeep. A “slashable” offence is far more severe, this included activities such as proposing invalid blocks or double signing blocks. If slashed, a validator is prohibited for engaging in validation activities forever. Additionally, if the staked balance of a validator falls below 16 ether, due to incurred penalties, they are forcefully removed for the network. The introduction of penalties such as this would make block reorganisations far more costly to perform however, the MEV opportunities could still be greater than any incurred penalty or slashing. Additionally, large scale block reorganisations are impossible as proof-of-stake blockchains periodically finalise blocks meaning they can never be changed.

A number of concerning developments surrounding time bandit attacks have emerged. Once such development is a modified version of the Flashbots mev-geth client that would facilitate payment for miners to reorganise past blocks. Additionally, a smart contract was developed by the twitter user “0xbunnygirl”\(^{19}\) that enabled users to ask a miner to reorganise the chain from a specified block. A user could send a transaction to this contract, calling its “request” function and specifying a past block number while attaching an ether reward for the operation that is locked in the contract. This transaction fills fields in a request object or struct contained within the contract that is initially set to unclaimed. A willing miner would then begin remining the specified block ensuring to include the initial request transaction. The miner can then initiate their own transaction calling the “reorg” function passing the address they wish to claim to reward to, this sets the request struct’s claimant field to their address and so once the expiry block has occurred, the miner can call the contract once more, passing the the claimant address in order to unlock the reward. Interestingly, a slashing mechanism was implemented whereby if the requester

\(^{18}\) Block reorganisation data from etherscan.io
determines any violation of trust, they can slash the miner reward before the expiry block, effectively locking their reward in the contract for no one to claim.

Uncle Bandit Attacks

Uncle bandit attacks target transactions that are made visible in what are known as uncles' blocks. As mentioned previously, there exists a chance two blocks are mined at the same time. These clashes or collisions lead to uncle blocks in the network. These uncle blocks remain within the network as they are cryptographically linked to the base genesis block, the miner even receives a small reward for their effort but the block is not a member of the longer, primary chain. Due to Ethereum’s shorter block time, these collisions happen relatively often, with twelve occurring in the last twenty four hours.

As uncles exist within the public network, anyone can see the transaction contents within, including any pure revenue opportunities, this is problematic for transactions that were submitted via private channels where privacy is required before block inclusion. Robert Miller, a core team member of the Flashbots group, was alerted to a sandwich attacker whose bundle was split up. Bundles like this should never be split up, this will be revisited in the Flashbots section. Upon investigation, Miller determined that the attacker's bundle was mined but included in an uncle block. This meant the sandwich attacker's pair of transactions were revealed and another attacker had spotted an opportunity. This second attacker targeted the initial sandwich attacker by replacing the back running transaction with their own, capturing the profit earned by arbitraging the pool following the sandwich attackers large purchase. This indicates that bot operators aren’t just surveying the public mempool but also the network’s uncle blocks.
MEV Elimination Research

Throughout the research process, two distinct systems were identified by the author as viable solutions to Ethereum’s MEV issue, the Flashbots auction infrastructure and its surrounding products as well as CowSwap, a new variation of decentralised exchange offering MEV protection.

MEV is a fundamental issue that many groups are attempting to solve or mitigate in a variety of ways, and it appears to be working. Ethereum has seen record low MEV extraction figures in recent times according to the Flashbots dashboard. Throughout 2021, it was common for MEV extraction to exceed a million dollars on a daily basis, yet now typical extraction amounts to about a third of that daily. This reduction is likely due to a combination of factors such as Ethermines prohibition of DEX front running as well as the efforts of MEV resistant alternatives to traditional decentralised exchanges such as CowSwap. This lends itself to the belief that MEV cannot be totally eliminated, but can be mitigated and its negative externalities drastically reduced.

One option to avoid your transaction getting attacked by the bots surveying the vulnerable public mempool is to pay a fee to have it directly included in an upcoming block by a miner. This is known as a private transaction and it bypasses the network’s mempool.

![Figure 11: Private Transaction Lifecycle](image)

As you can see in the above figure, the user submits their transaction request and instead of being added to the standard public mempool, it is selected directly by a miner for inclusion, this requires a degree of trust as the miner has no obligation to do so. While this is a highly effective method to avoid exposure to the network’s mempool it is a luxury that is not afforded to standard users of Ethereum as many have no method for communicating directly with miners.
Flashbots

The Flashbots research group developed a new process for block construction in Ethereum. Their tools provide a method for bots referred to as searchers to efficiently extract MEV whilst minimising the negative externalities associated with traditional extraction techniques. They reduce the impact of wasteful gas bidder wars or priority gas auctions by proposing a new system that utilises a private channel of communication between searchers and miners. By using a sealed-bid auction mechanism known as the flashbots auction, block producers are able to outsource the work involved in finding optimal block construction trustlessly, whilst also reducing the raised gas prices inflated by priority gas auctions.

The group also believes in the importance of eliminating information asymmetries. Before their exposition of the MEV issue, only a select few were aware of these activities. They hope that by making people aware of these activities, they can take necessary precautions to avoid being targeted themselves. They provide a dashboard, MEV explore, where anyone can see their data aggregation of MEV transactions on chain. This data aggregation requires significant analysis of the available on chain data as often, searchers attempt to conceal their activities in a variety of ways. By creating the MEV explore dashboard, they have given the community a way to understand the scale and significance of these activities.

Flashbots Auction Design

The Flashbots auction infrastructure aims to solve some of Ethereum's existing shortcomings by changing its fundamental topography. It employs a first-price sealed-bid auction process\(^\text{20}^\) to allow its users to privately communicate their bid and transaction order preference. By doing so, the risk of failed bids is eliminated, miner payoffs are maximised and an efficient market for pricing of MEV opportunities is provided. Additionally, the front-running vulnerabilities of the standard mempool are eliminated as transactions are submitted via their private channel, however there are caveats.

Design Goals

During development, the Flashbots team set out a number of design goals in order to fulfil their requirements, the first of which was pre-trade privacy. It is clear issues arise when trade requests wait for periods of time in the mempool and so a central goal was to have transactions only become publicly

\(^{20}\) First-price sealed-bid auction (FPSBA), also known as a blind auction, where all bidders simultaneously submit sealed bids so that no bidder knows the bid of any other participant. The highest bidder wins and pays the price that was submitted (wikipedia.org)
visible after they have already been included in a block. Another design goal was to ensure no unsuccessful transaction requests were included in any blocks. This would eliminate the artificial scarcity of valuable blockspace as well as reduce the unnecessary network load and chain congestion. Other goals such as the ability to merge incoming bundles, provide finality protection and to enable all this with complete privacy in a permissionless way were outlined. Complete privacy in a permissionless way is truly the group's end game achievement. The auction infrastructure has a number of intermediaries that as of now are trusted and can in theory censor transactions as they have access to the contents of the transaction bundles before they are added to the chain.

Technical Architecture

To reach these design goals, the Flashbots research group proposed a new network operating alongside Ethereum with three distinct parties, a searcher, a relayer and a miner. Each of these parties perform a specific role for the auction infrastructures communication channel.

The Flashbots Bundle

The Flashbots bundle is used for fundamental operation within the auction infrastructure. It is a standardised format for communication within the private channel. Searchers compile transactions into these bundles and submit them to relayers to be forwarded to miners for inclusion.
The above figure shows the format of a Flashbots bundle. Within the bundles parameters the “txs” field will provide an array or list of signed transactions that are to be executed together or not at all. The “blockNumber” field refers to the block number for which this bundle is proposed for, this is formatted in hexadecimal. These are the only required fields with the rest remaining optional, the min and max timestamp fields specify a timeframe for which the bundle is valid for. Ethereum transactions are reverted when they cannot be fulfilled. This can happen for a number of reasons but by including them on-chain, valuable block space is wasted with these useless transactions and so flashbots do not include any failed transactions on-chain by default. However, if required, an additional array of transactions that are allowed to revert can also be included in the “revertingTxHashes” field.

Searchers

Searchers are the first step in the Flashbots auction infrastructure. They are the users who monitor the state of the chain and send bundles to the relayers. Typically, searchers are either normal users looking for MEV protection, are bot operators themselves or are decentralised applications. The bot operators are not dissimilar to their standard Ethereum counterparts, searching for profitable opportunities. Decentralised applications also make use of the infrastructure when certain use cases are required. For example, mistX
is a decentralised exchange aggregator that utilises the auction infrastructure to offer its users front running protection. They do so by submitting their users' orders directly to flashbots relayers, bypassing the network's mempool.

Due to the fact that the searchers submit their transaction bundles directly to the relayers, they are able to obtain valuable pre-trade privacy. Similar to a gas price, searchers must submit a bid for their bundle to be included by the miners, this can be done by either specifying a price or by using a direct transfer of ether to the coinbase address. The coinbase transfer functionality is implemented via a smart contract which transfers the ether from the contract to the coinbase address of the miner who mines the block and it allows for the searchers to make payments conditional on their successful inclusion. This ether transaction cannot be isolated by the miner either, due to the way flashbots bundles are designed, they must be included in their entirety or not at all.

Relayers

The next step in the auction infrastructure is performed by the relayers. The relayers receive bundles from searchers and forward them to miners. This extra step was added as relayers are tasked with validating any incoming bundles thus preventing any invalid bundles reaching the miners. Invalid bundles have the potential to cause issues as searchers no longer have to pay for failed transactions as they are not included in the next block, therefore there was a risk malicious searchers would spam the Flashbots miners with invalid bundles.
Miners

Miners are the last step in the Auction system, their job is to produce blocks using the bundles forwarded to them by the relayers. These miners differ from their traditional counterparts by running a modified version of the go-ethereum or geth client known as mev-geth. Mev-geth is maintained by the Flashbots research group.

Figure 14: Flashbots Auction Infrastructure - Miner’s Role, figure from Flashbots documentation

Where traditional miners select transactions from the mempool based on their gas price, miners running the mev-geth client pick the most profitable bundles by selecting the bundles with highest bids forwarded to them and place them at the top of their constructed block. It is this ability to clearly express a bounty for inclusion that enables an efficient auction for blockspace, thus avoiding the artificial blockspace scarcity issues. Once ready, mev-geth miners compare their newly constructed block to the block generated by the standard geth client. The miner then begins mining their mev-geth block provided the standardly generated geth block is less profitable.

21 docs.flashbots.net. (n.d.). overview | Flashbots Docs.
22 docs.flashbots.net. (n.d.). overview | Flashbots Docs.
Flashbots Protect RPC

The Flashbots group also provides a way for Ethereum’s less technically adept users and developers to make use of their auction infrastructure. They do so by providing an RPC endpoint which offers the front running protection by letting users avail of their pre-trade privacy. To make use of flashbots protect users simply have to add an alternative url to their metamask, by doing so their transaction isn’t sent to a standard node. Flashbots protect’s users will also experience no failed transactions, saving them ether in gas fees, as only transactions are only mined successfully if they contain no reverts.

Solution Evaluation

It is clear the current implementation of the flashbot auction infrastructure has many benefits and succeeds in reducing a number of the negative externalities associated with traditional MEV extraction techniques however there are still issues with its current version. The system contains a number of technical limitations which prevent the network from operating in a fully trustless manner. Both relayers and miners are trusted parties in the system who have full access to bundle content and so can reorder, censor or steal any bundles or transactions within bundles sent to them.

Their implementation, while still in very active development, is currently devoid of finality protection, complete privacy and general permissionless-ness. For their auction system to become truly decentralised and in turn, the leading standard for the Ethereum network, the properties of complete privacy and permissionlessness are mandatory. The system is in constant development, currently on version 0.6, and their roadmap aims to replace the current guarantees of trust with a combination of cryptographic and crypto economic guarantees of full privacy.
CowSwap

CowSwap is the product of two existing protocols, Gnosis and Balancer. Together they developed the CoW Protocol, the underlying protocol utilised by the MEV resistant decentralised exchange, CowSwap. CowSwap was designed to be a decentralised exchange that offered the best prices and MEV protection available on Ethereum. By using Gnosis Protocol’s batch auction mechanism to provide MEV resistant batches of trades in addition to off-chain order placement and access to any on-chain liquidity when required. This on-chain liquidity is primarily provided by Balancer and its new Vault architecture. Improving upon their initial iteration, Gnosis Protocol’s version two optimises for coincidence of wants. A coincidence of wants refers to the economic phrase for an occasion “when two parties each hold goods that the other party wishes to have”23. As of August 2021, CowSwap has successfully upgraded to audited smart contracts that include a tight integration with Balancer’s vault architecture. This means CowSwap is now a stable, fully audited decentralised application and is no longer in alpha stage development.

Gnosis Protocol

Gnosis Protocol’s Auction is a platform for “conducting fair, transparent, and decentralised token price discovery”24, in essence it is a permissionless decentralised exchange focusing on gas and fee efficiency whilst maximising token liquidity. By removing the use of a central operator, replacing it with a decentralised third party, Gnosis is capable of settling large batches of trades providing the best execution for its users. A critical feature of the Gnosis Auction is that all bidders engaging in an auction receive the auctioned tokens at the same price. This price is determined when the auction concludes. This uniform clearing price is an essential feature as it makes the protocol MEV resistant. By not allowing front and back runners to take advantage of users by manipulating prices, Gnosis Protocols’s batch auction mechanism negates sandwich attacks unlike standard automated market maker solutions. It also protects its users from front runners as the uniform clearing price means there is no way to manipulate or inflate the price of the token for a bidder.

Batch auctions are initiated by the auctioneer defining a sell amount or quantity of tokens to be sold along with a minimum limit price per token. Once the auction has begun, bidders can express bids via limit orders, specifying an amount and maximum limit price. Upon conclusion of the auction the final clearing price is calculated. This is done by determining the price where the sell supply line and the demand curve intersect. You can see this calculation depicted in the figure below. This means the Gnosis batch auction is

a variation of a dutch auction. Gnosis employed this form of auction in order to provide the critical uniform clearing price for each bidder as without, the auction would no longer be MEV resistant.

![Dutch Auction Diagram](image)

**Figure 15:** Gnosis Batch Auction Closing Price Calculation Visualisation, from the Gnosis Auction documentation

All bidders with a sufficiently high limit price receive the auctioned tokens at the clearing price. The bidders that specified a limit lower than the clearing price do not receive auctioned tokens, instead they have the ability to withdraw the tokens used for bidding. In the event the final clearing price determined by the auction is below the limit price set by the auctioneer, no tokens are distributed and all bidders can withdraw their bidding tokens.

**Balancer**

Balancer set out to be the primary source of decentralised finance’s liquidity with its version two implementation, prioritising security, flexibility as well as capital and gas efficiency. It is an automated market maker similar to others that provides the liquidity for users to trade tokens. However it differs

---

25 A dutch auction is a market structure where the price of an item is lowered until all the units within the lot are sold
26 Gnosis. (2022). Gnosis Auction. [https://gnosis-auction.eth.link/#/docs/batch-auctions](https://gnosis-auction.eth.link/#/docs/batch-auctions)
from traditionally automated market makers by utilising a centralised liquidity pool known as the “vault”. Balancer V2 implemented the vault as the core of its upgrade.

The Vault is a smart contact that holds and manages all the assets for each Balancer pool. A pool is simply a smart contract containing the user deposited funds or tokens. These liquidity pools allow users to trade between the assets within the pool.

Traditionally, automated market maker designs have separate smart contracts for each liquidity pool or trading pair, with the pool's logic specified within. This logic provides the accounting mechanisms that dictate the pool's deposits, exits and trades, for example under what conditions a user can buy or sell a token. These individual pool or trading pair smart contracts were produced by smart contract factories. Balancer however, separated the smart contract in charge of token management and accounting from the individual pool contracts. By doing so, the individual pool contracts were simplified as they no longer needed the functional requirements to actively manage the assets that are contained within each pair’s contract, instead they only needed to calculate amounts for accounting operations. Additionally, as the pool logic is external to the vault managing the tokens, Balancer V2 pools can implement any desired arbitrary logic, meaning their pools are entirely customisable.

Figure 16: Balancer V1 vs Balancer V2 Centralised Liquidity (Vault) Depiction from Balancer documentation

---

An interesting development of this centralised vault structure means swaps are now notably cheaper. When executing swaps that require multiple hops across many different AMM pools, gas prices become cumbersome due to the need to send tokens at each stage. With Balancer’s new system, since all tokens are contained within one smart contract, swaps that would previously require trades and transfers across various AMM pools now require the same amount of gas as a single hop swap on a traditional AMM such as Uniswap.

Additionally, the vault has the ability to maintain token balances for individuals via ethereum addresses allowing users to have an internal balance within the vault. These internal token balances can be used to execute trades or swaps without the requirement for any token transfers saving hugely on gas requirements for trading activity. A significant amount of gas is retained by the protocol’s vault as trades can be executed in batches against any number of pools with only the final net token amounts needing to be transferred to and from the vault.

As the pool logic is implemented independently to that of the vault's accounting logic, the vault was designed to keep token balances separate and strictly independent. By doing so, Balancer has protected from any maliciously or negligently designed pools or tokens, that would have been capable of stealing funds by draining pools other than their own.
Figure 17: The Balancer Vault Architecture Overview, from Balancer documentation²⁸

The large scale of the liquidity within Balancer’s vault has allowed them to offer flash loans by utilising this combined token liquidity. Additionally, if price discrepancies occur across different pools within the vault, arbitrageurs can rebalance the offending pools by notifying the vault which in turn executes the required swaps and awards the arbitrageur the profit.

Another improvement made by Balancer was the inclusion of sandwich attack resistant oracles. These oracles, pioneered by uniswap, utilise accumulators. This means other decentralised applications reliant on Balancer pricing data will be able to obtain said data without having to poll an endpoint and store past accumulator states, reducing gas costs. Balancer offers two forms of pricing oracle, one which provides a more up to date price but is vulnerable to manipulation and one whose pricing data is less up to date but is far more resilient to manipulation. Attackers have in the past targeted pricing oracles as a way to obtain tokens at a heavy discount, by artificially inflating the price of a token they can trade this inflated token for another before its value is normalised. In order to make their oracles more resilient to manipulation, the price is taken as a rolling average rather than its immediate value. By doing so attackers would have to manipulate the price for extended periods of time which is an expensive endeavour.

CowSwap Design

Attempting to protect its users from searchers, CowSwap has devised a new form of decentralised exchange that minimises the on-chain trading of tokens via AMMs. Its functionality in its simplest form is as follows, when two prospective traders each possess a token the other wants, the protocol matches these users and settles the trade directly between the two off-chain as there is no need for an automated market maker, the liquidity required exists within the trade. Following a successful off-chain settlement only the final token balances are transferred to each user and included on-chain. In the event there is an excess amount of a particular token following a successful off-chain settlement, the excess is settled using automated market makers, effectively acting as a decentralised exchange aggregator. By doing so CowSwap becomes a barter economy where traders can engage somewhat directly with each other. This implementation leads to improved prices for users as it tightens the spreads experienced by the trader. The so-called meta decentralised exchange aggregator utilises Gnosis protocol’s batch auction mechanism to settle any suitable orders to be batched and processed with the same clearing price, meaning that every order executed by the protocol within a given batch will receive assets priced equally to others.

---

Computational power is required to determine the optimal way to settle incoming CowSwap trades as a batch. To solve this issue, the protocol introduces a party to their system known as solvers. Solvers are tasked with providing valid settlement solutions to the batch auctions. These optimal batch settlement solutions will optimise for coincidences of wants as well as settle any excess trade amounts using the best prices available. By utilising this third party, CowSwap incentivises its solvers to determine the optimal batch settlement solution using incentives and can therefore operate without the use of a central operator or a constant function market maker. These solvers play a similar role to miners in the flashbots architecture who compete to have their block chosen as the optimal block, solvers compete with each other to submit the optimal batch. If successful, solvers earn tokens from the protocol as a reward for their efforts. The protocol, therefore, has removed and effectively outsourced their requirement to determine the most optimised batch auction solutions and eliminated any issues surrounding centralised authorities.

Those wishing to become a solver for the protocol must fulfil a number of requirements. The first step for a prospective solver is to deposit a bond of tokens, effectively acting like a stake or insurance in order to ensure good behaviour. This stake can be penalised or slashed by the CowSwap DAO if a solver does not possess the technical ability to assemble the appropriate batch settlement solution or is engaging in malicious behaviour as a result of its access to said batches. Once the required amount of tokens has been deposited, the DAO votes on whether to accept the new solver’s application. If the vote is passed successfully, the depositors address is added to a whitelist of other solvers.

As the optimal batch settlement solution may be accomplished through varying permutations, solvers are encouraged to compete against each other to deliver the best iteration. The protocol hopes that through decentralised and permissionless competition, CowSwap can deliver the best prices in DeFi. In the event a coincidence of wants is not or is only partially found, CowSwap has the ability to tap into the on-chain liquidity pool that offers the best price for the auction settlement. GPv2 solvers also implement very tight slippage tolerances, making the transaction much harder for predatory bots to victimise as only authenticated solvers can submit batch settlement solutions.

In essence, the underlying CoW Protocol, which utilises systems from both Gnosis and Balancer, aggregates trades placed by users on the CowSwap exchange. These aggregated trades are then bundled into batches where solvers, a decentralised third party, compete to provide the best execution for the traders, with an emphasis on providing for coincidences of wants. By settling these coincidences off-chain, and sourcing the best execution prices for on-chain execution settlement in batches, CowSwap effectively limited the potential for searchers to attack or abuse its traders.
Solution Evaluation

CowSwap manages to effectively offer protection from front and back-runners in all trade eventualities. In the event a coincidence of wants is found, there is no need to settle trades using other vulnerable on-chain automated market makers or liquidity providers as we know the liquidity for the trade must exist within the batch of trades. The potential for searchers to attack is therefore completely eliminated as the transaction is never made vulnerable to strategies abusing transaction ordering. Additionally, due to Gnosis’s batch auction uniform clearing prices, the result of batched trades do not rely on relative ordering within the batch meaning sandwiching and front running bots have no viable opportunities.

Additionally, their smart contracts are fully audited and have no admin key, they are entirely trustless and decentralised. The other fully decentralised element of their system is their frontend user interface, which is hosted on IPFS, the interplanetary file system. This means their interface can be used by anyone without the possibility of blocking or censorship.

However, there are issues that exist with the current implementation of the CowSwap decentralised exchange. Firstly, the barrier for entry to become a solver is quite high. Prospective solvers must be accepted by the GnosisDAO as well as have the technical knowledge to successfully implement the required technologies to perform a complex role. This difficulty is represented in the number of solvers as at present, there are only 34 unique solver addresses. This means the protocol's intended competitive market for supplying valid batch settlement solutions is somewhat diminished. Further, when aggregated, there are just 10 larger entities that run a large proportion of these solvers. An entity meaning a single party that runs multiple solvers at different addresses. These entities have, to date, accounted for 150 thousand of the 170 thousand historic batches solved which is about 88%. Some notable entities are 1inch and 0x, both decentralised exchanges or decentralised exchange aggregators, who combined, account for about 30% of the entity batches solved to date29. This is clearly a higher level of centralisation than is desired but there is large room for improvement as this is a relatively young protocol.

It is worth noting, the protocol plans to move to a fully permissionless model in the future. This would mean that solvers would no longer require DAO approval to be added to the whitelist of authenticated solvers. Provided a sufficient stake has been deposited to ensure good behaviour, anyone can become a solver for the protocol.

Another criticism of the protocol is its current use of a centralised off-chain order book management system. Solvers in the system communicate what is known as the “objective value” they have computed

---

for the current batch auction and using a centrally managed channel, the solvers reach consensus on who has earned the right to settle the current batch based upon this value. In order for solvers to compute this value the state of the auction must be also agreed upon. This agreement is determined via their off-chain orderbook API. The protocol plans to decentralise this orderbook element by moving it to a peer-to-peer client network. A method of consensus has not yet been determined but options such as native client implementation or utilising cheaper side chains for data availability are being explored.

Conclusion

Miner extractible value is undoubtedly a cause for concern for Ethereum at present. However, it is the author's belief that minimising miner extractible value’s negative effects on the overall network by leveraging infrastructures such as flashbot’s auction along with careful decentralised application design, such as that of CowSwap, Ethereum’s network health will drastically improve. As mentioned previously, MEV extraction techniques inflict a number of negative effects on the Ethereum network such as inflated gas prices, artificial blockspace scarcity as well as poor decentralised exchange trade execution. These negative externalities have, in the author's belief, been mitigated or resolved through the efforts of Flashbots and CowSwap.

By providing a private communications channel that has been widely adopted by miners, the Flashbots auction infrastructure has provided a way for searchers to compete with each other and express bids for their strategies without causing periodic spikes in gas prices. Additionally, by performing this communication off-chain, precious block space is saved by the removal of failed MEV extraction transactions, thus partially alleviating the pressure experienced by blocks in recent times. This leads to an improved user experience for Ethereum’s users who no longer experience such high variance in gas prices. If the Flashbots research and development group can fulfil their system design goals of replacing the current trust guarantees placed in the hands of relayers and miners within their infrastructure. By implementing codified guarantees of good behaviour, Flashbots auction will truly be an efficient mechanism for eliminating a large degree of MEV’s on-chain negative externalities.

In relation to general user experience, it is clear that with proper consideration and careful design, users can be effectively protected from Ethereum’s so-called invisible tax on users. The CowSwap meta decentralised exchange aggregator and the underlying CoW Protocol have paved the way for a new generation of carefully designed decentralised applications with a focus on reducing users MEV exposure. By batching users transactions and outsourcing the role of finding the best execution prices, CowSwap
has eliminated the ability for searchers to manipulate the transaction order to benefit themselves at the users expense. The typical avenues of attack for front runners or sandwich attackers are closed allowing users peace of mind when engaging in decentralised trading. While there are still improvements to be made in regard to the currently centralised methods of consensus relating to CowSwaps orderbook, the protocol has already defined a route of decentralisation for the future. With increased adoption and a more competitive solver economy, CowSwap will be an effectively designed exchange for users to trade without fear of being targeted.

There does however still exist threats to Ethereum's overall stability and finality. The potential earnings available to miners by rewriting past transactions and blocks is often already greater than the earned rewards and transaction fees. As concerning as this incentive is, improvements implemented at the protocol level have proved effective and further improvements made via proposals could provide sufficient dissuasions to these activities. Protocol improvement proposals such as EIP-1559 have undoubtedly led to more consistent gas prices and improved the network’s congestion handling mechanisms. While this does not directly serve as a solution to MEV, it does show that improvements made at the protocol level can improve the overall experience of Ethereum’s users. While its options are currently limited, once transitioned to the proof-of-stake consensus mechanism, Ethereum will have methods for avoiding such attacks. Protocol inflicted penalties and slashings may suffice to eliminate the possibility of time bandit attacks and block re-organisations. This combined with a strong finality of past blocks provided by Casper, will in theory make the modification of Ethereum’s immutable ledger financially infeasible, as attackers will not be capable of large rewrites, as large MEV extraction opportunities would need to be available within the past few blocks. This is why intelligent and conscious decentralised application design is imperative. By reducing MEV opportunities for validators to capture themselves, the financial implications of slashing should dissuade any malicious activity.

It is clear Ethereum’s miner extractible value problem has no one “silver bullet” solution. However, through a combination of conscious decentralised application design, effective means of communication between searchers and miners and protocol implemented penalties, Ethereum has the capability to effectively deal with miner extractible value.
Bibliography


docs.flashbots.net. (n.d.). overview | Flashbots Docs. [online] Available at: https://docs.flashbots.net/flashbots-auction/overview [Accessed 2 Mar. 2022].

docs.flashbots.net. (n.d.). welcome to flashbots | Flashbots Docs. [online] Available at: https://docs.flashbots.net/ [Accessed 2 Mar. 2022].


55

explore.flashbots.net. (n.d.). MEV Explore. [online] Available at: https://explore.flashbots.net/.


