Becoming Immutable: How Ethereum is Made

Andrea Canidio ~~CoW Protocol~~
joint work with Vabuk Pahari (Max-Planck-Institute for Software Systems)

Decentralized Finance & Crypto Workshop @ Scuola Normale Superiore
26 -28 Jan 2026

Is Blockchain, permissionless, deterministic and immutable?

- Once a transaction in on-chain: yes

Is Blockchain, permissionless, deterministic and immutable?

- Once a transaction in on-chain: yes
- **But**: several layers of intermediaries determine if/how a transaction is included on-chain

Is Blockchain, permissionless, deterministic and immutable?

- Once a transaction in on-chain: yes
- **But**: several layers of intermediaries determine if/how a transaction is included on-chain
- On Ethereum, the process is called PBS (for Proposer-Builder separation),

Is Blockchain, permissionless, deterministic and immutable?

- Once a transaction in on-chain: yes
- **But**: several layers of intermediaries determine if/how a transaction is included on-chain
- On Ethereum, the process is called PBS (for Proposer-Builder separation),
  - For each Ethereum block added, thousands are proposed and then discarded.

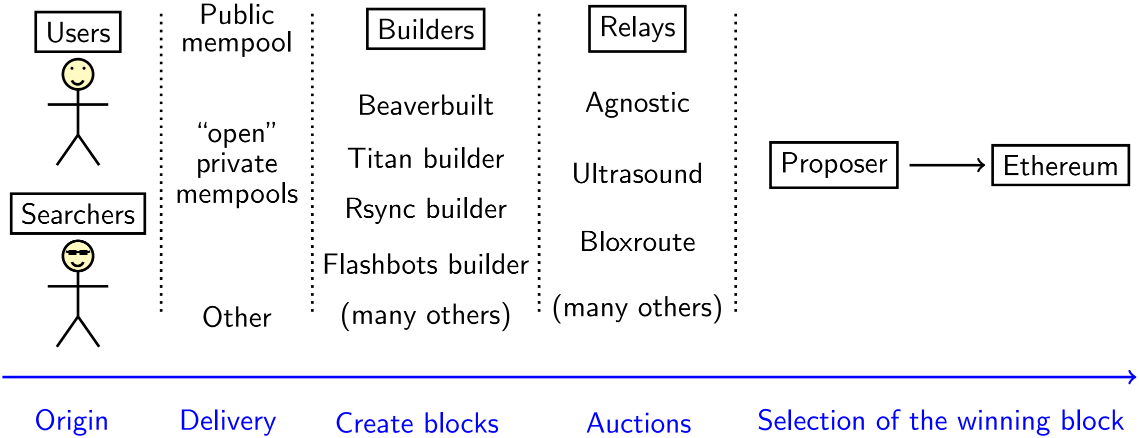
Is Blockchain, permissionless, deterministic and immutable?

- Once a transaction in on-chain: yes
- **But**: several layers of intermediaries determine if/how a transaction is included on-chain
- On Ethereum, the process is called PBS (for Proposer-Builder separation),
  - For each Ethereum block added, thousands are proposed and then discarded.
  - The block added to the chain is chosen via a **noisy** and **opaque** auction.

Is Blockchain, permissionless, deterministic and immutable?

- Once a transaction in on-chain: yes
- **But**: several layers of intermediaries determine if/how a transaction is included on-chain
- On Ethereum, the process is called PBS (for Proposer-Builder separation),
  - For each Ethereum block added, thousands are proposed and then discarded.
  - The block added to the chain is chosen via a **noisy** and **opaque** auction.
- In what way these intermediaries affect/distort Ethereum's fundamental promise?

Background: The Journey of an Ethereum Transactions



Users

Searchers

Public mempool

“open” private mempools

Other

Builders

Beaverbuilt

Titan builder

Rsync builder

Flashbots builder

(many others)

Relays

Agnostic

Ultrasound

Bloxroute

(many others)

Proposer → Ethereum

Origin

Delivery

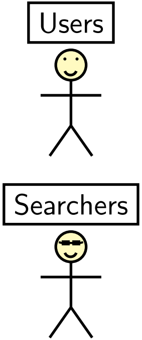
Create blocks

Auctions

Selection of the winning block

(Repeat approximately every 12 seconds)

Origin of transactions

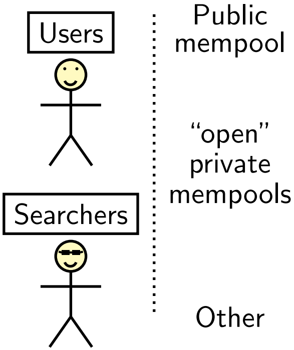


Users

Searchers

- Searchers: arbitrage bots, sandwich bots, liquidation bots, ...
- We classify a tx as a ‘searcher’s tx” if: (1) it is a swap, (2) generated using a smart contract, (3) whose code is not disclosed

Delivery of transactions



Users

Searchers

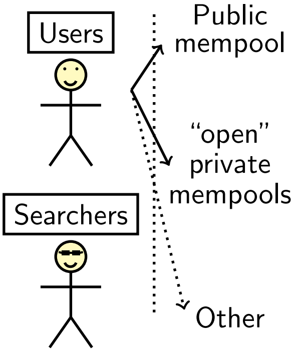
Public mempool

“open” private mempools

Other

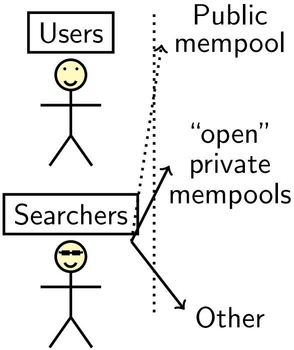
- “Open” private mempools: MEV Blocker, MEV Share (Flashbots), Blink, Merkle, Metamask Smart Transactions
- Other: direct communication with builders

Delivery of transactions



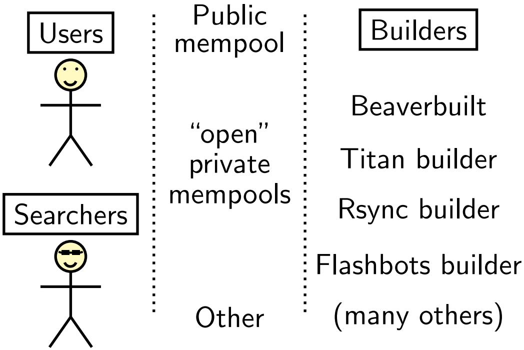
- "Open" private mempools: MEV Blocker, MEV Share (Flashbots), Blink, Merkle, Metamask Smart Transactions
- Other: direct communication with builders

Delivery of transactions



- “Open” private mempools: MEV Blocker, MEV Share (Flashbots), Blink, Merkle, Metamask Smart Transactions
- Other: direct communication with builders

Packaging Transactions into Blocks (Builders)



Users

Searchers

Public mempool

“open” private mempools

Other

Builders

Beaverbuilt

Titan builder

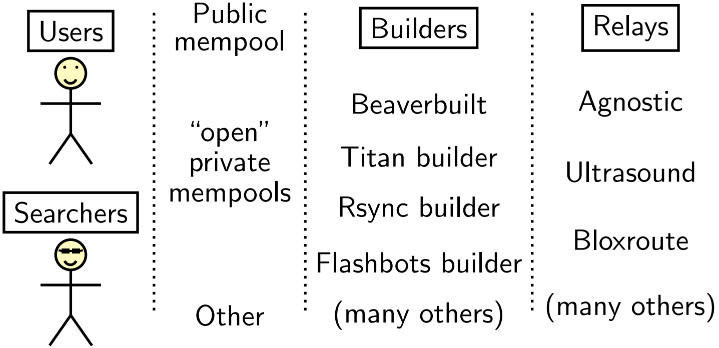
Rsync builder

Flashbots builder

(many others)

- Public transactions: appear in the public mempool and are available to all builders,
- Private transactions: not public but are included in blocks by multiple builders
- Exclusive transactions: appear only in blocks by a single builder.

Bidding (Relays)



Users

Searchers

Public mempool

"open" private mempools

Other

Builders

Beaverbuilt

Titan builder

Rsync builder

Flashbots builder

(many others)

Relays

Agnostic

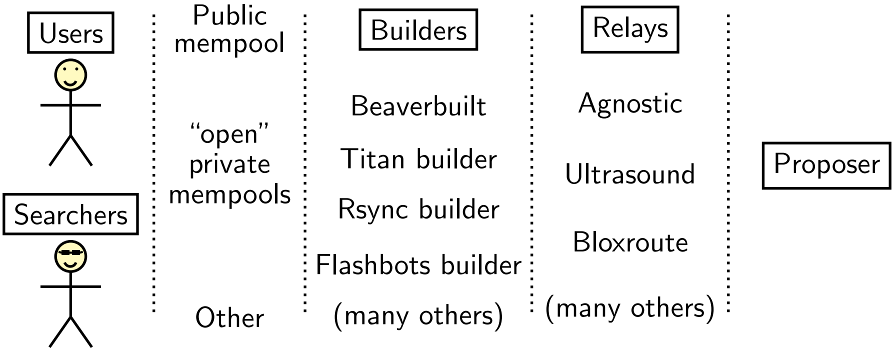
Ultrasound

Bloxroute

(many others)

Each relay runs an ascending price auction: block builders submit bids by proposing a block and a payment to the block proposer

The block proposer (or validator)



Users

Searchers

Public mempool

"open" private mempools

Other

Builders

Beaverbuilt

Titan builder

Rsync builder

Flashbots builder

(many others)

Relays

Agnostic

Ultrasound

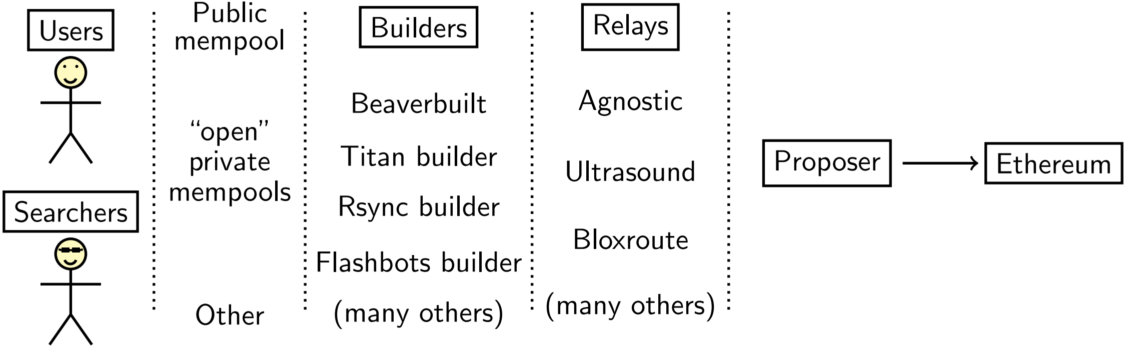
Bloxroute

(many others)

Proposer

- The block proposer is known in advance and changes with every slot
- The block proposer queries each relay it is connected with for its highest-paying block
- It chooses the best-paying overall and adds it to the blockchain

The block proposer (or validator)



Users

Searchers

Public mempool

“open” private mempools

Other

Builders

Beaverbuilt

Titan builder

Rsync builder

Flashbots builder

(many others)

Relays

Agnostic

Ultrasound

Bloxroute

(many others)

Proposer ⟶ Ethereum

- The block proposer is known in advance and changes with every slot
- The block proposer queries each relay it is connected with for its highest-paying block
- It chooses the best-paying overall and adds it to the blockchain

Background: the journey of transactions (aka BPS)

- Payments: from users/searchers to Builders, from Builders to Proposer, from Builders to private mempools, ...

Background: the journey of transactions (aka BPS)

- Payments: from users/searchers to Builders, from Builders to Proposer, from Builders to private mempools, ...
- Almost continuous process

Background: the journey of transactions (aka BPS)

- Payments: from users/searchers to Builders, from Builders to Proposer, from Builders to private mempools, ...
- Almost continuous process
- Non-winning blocks are discarded → we never observe the vast majority of this activity.

Background: the journey of transactions (aka BPS)

- Payments: from users/searchers to Builders, from Builders to Proposer, from Builders to private mempools, ...
- Almost continuous process
- Non-winning blocks are discarded → we never observe the vast majority of this activity.
  - For each block received, each relay discloses its hash, the name of the builder, a timestamp, and the associated bid, but not its content.

Background: the journey of transactions (aka BPS)

- Payments: from users/searchers to Builders, from Builders to Proposer, from Builders to private mempools, ...
- Almost continuous process
- Non-winning blocks are discarded → we never observe the vast majority of this activity.
  - For each block received, each relay discloses its hash, the name of the builder, a timestamp, and the associated bid, but not its content.
- **Private mempools / RPC endpoints** are becoming increasingly important:
  - MEV Blocker, MEV Share, Blink, Merkle share users' transactions with multiple builders;
  - no public documentation on how Metamask Smart Transactions work.

The “micro” implications of PBS

- A user’s transaction may appear in some proposed blocks but not in others, leading to **delayed on-chain inclusion**.

The “micro” implications of PBS

- A user’s transaction may appear in some proposed blocks but not in others, leading to **delayed on-chain inclusion**.
- A user’s transaction (i.e., a swap) may be included in differently in different blocks, leading to different execution prices in different blocks.

The “micro” implications of PBS

- A user’s transaction may appear in some proposed blocks but not in others, leading to **delayed on-chain inclusion**.
- A user’s transaction (i.e., a swap) may be included in differently in different blocks, leading to different execution prices in different blocks.

Open research question: it requires to compare the content of non-winning blocks with that of the winning one.

We collect and analyze **non-winning Ethereum blocks**

- 8 mins of Ethereum history: on Dec 3rd, 2024, from 02:37:35 PM to 02:45:25 PM UTC

We collect and analyze **non-winning Ethereum blocks**

- 8 mins of Ethereum history: on Dec 3rd, 2024, from 02:37:35 PM to 02:45:25 PM UTC
- 38 winning blocks (21,322,622 to 21,322,660)

We collect and analyze **non-winning Ethereum blocks**

- 8 mins of Ethereum history: on Dec 3rd, 2024, from 02:37:35 PM to 02:45:25 PM UTC
- 38 winning blocks (21,322,622 to 21,322,660)
- 15,097 proposed blocks, or 28.3% of all blocks proposed during the study period (all those submitted via Agnostic relay)

We collect and analyze **non-winning Ethereum blocks**

- 8 mins of Ethereum history: on Dec 3rd, 2024, from 02:37:35 PM to 02:45:25 PM UTC
- 38 winning blocks (21,322,622 to 21,322,660)
- 15,097 proposed blocks, or 28.3% of all blocks proposed during the study period (all those submitted via Agnostic relay)
- 10,793 individual transactions and 2.380.014 individual transactions/proposed blocks combinations

We collect and analyze **non-winning Ethereum blocks**

- 8 mins of Ethereum history: on Dec 3rd, 2024, from 02:37:35 PM to 02:45:25 PM UTC
- 38 winning blocks (21,322,622 to 21,322,660)
- 15,097 proposed blocks, or 28.3% of all blocks proposed during the study period (all those submitted via Agnostic relay)
- 10,793 individual transactions and 2.380.014 individual transactions/proposed blocks combinations

- Novel dataset but unbalanced:
  - Missing 11 auction cycles
  - Very few blocks from Beaverbuilt (9.4% in our data vs 40.2% total)

Preliminary: The set of transactions in *winning* blocks is very different from those in *submitted* blocks

- 1/2 of txs in our dataset are from users (the rest from searchers) vs 87.8% among txs included in winning blocks,

Preliminary: The set of transactions in *winning* blocks is very different from those in *submitted* blocks

- 1/2 of txs in our dataset are from users (the rest from searchers) vs 87.8% among txs included in winning blocks,
- 1/2 of txs in our dataset are swaps vs 14.5% among txs included in winning blocks,

Preliminary: The set of transactions in *winning* blocks is very different from those in *submitted* blocks

- 1/2 of txs in our dataset are from users (the rest from searchers) vs 87.8% among txs included in winning blocks,
- 1/2 of txs in our dataset are swaps vs 14.5% among txs included in winning blocks,
- 35% of txs in our dataset originate from just two addresses.

Preliminary: The set of transactions in *winning* blocks is very different from those in *submitted* blocks

- 1/2 of txs in our dataset are from users (the rest from searchers) vs 87.8% among txs included in winning blocks,
- 1/2 of txs in our dataset are swaps vs 14.5% among txs included in winning blocks,
- 35% of txs in our dataset originate from just two addresses.
    - We identify them as builder-integrated searchers performing CEX-DEX arbitrage, and call them "Rsync-bot" and "Titan-bot".

Preliminary: The set of transactions in *winning* blocks is very different from those in *submitted* blocks

- 1/2 of txs in our dataset are from users (the rest from searchers) vs 87.8% among txs included in winning blocks,
- 1/2 of txs in our dataset are swaps vs 14.5% among txs included in winning blocks,
- 35% of txs in our dataset originate from just two addresses.
  - We identify them as builder-integrated searchers performing CEX-DEX arbitrage, and call them “Rsync-bot” and “Titan-bot”.
- We find that 85% of the value of winning blocks is from exclusive transactions (10% from private but not exclusive transactions, the rest from transactions that go through the mempool).

Result 1: 22% of users' transactions are delayed

- 35% of delayed users' transactions initially appear as **exclusive to a single (losing) builder**

Result 1: 22% of users' transactions are delayed

- 35% of delayed users' transactions initially appear as **exclusive to a single (losing) builder**
- 12 users' transactions remain exclusive to one builder for 2–3 slots **before eventually appearing in the public mempool** (one of those transactions is sandwiched)

Result 1: 22% of users' transactions are delayed

- 35% of delayed users' transactions initially appear as **exclusive to a single (losing) builder**
- 12 users' transactions remain exclusive to one builder for 2–3 slots **before eventually appearing in the public mempool** (one of those transactions is sandwiched)
- In some cases, after one of these transactions appears in the public mempool, it is **included on chain by the same builder that held the transaction exclusively from the start.**

Result 1: 22% of users' transactions are delayed

- 35% of delayed users' transactions initially appear as **exclusive to a single (losing) builder**
- 12 users' transactions remain exclusive to one builder for 2–3 slots **before eventually appearing in the public mempool** (one of those transactions is sandwiched)
- In some cases, after one of these transactions appears in the public mempool, it is **included on chain by the same builder that held the transaction exclusively from the start.**
- Users' transactions are first delayed and then exposed to attacks.

Result 1: 22% of users' transactions are delayed

- 35% of delayed users' transactions initially appear as **exclusive to a single (losing) builder**
- 12 users' transactions remain exclusive to one builder for 2–3 slots **before eventually appearing in the public mempool** (one of those transactions is sandwiched)
- In some cases, after one of these transactions appears in the public mempool, it is **included on chain by the same builder that held the transaction exclusively from the start.**
- Users' transactions are first delayed and then exposed to attacks.

Result 1: 22% of users' transactions are delayed

- 35% of delayed users' transactions initially appear as **exclusive to a single (losing) builder**
- 12 users' transactions remain exclusive to one builder for 2–3 slots **before eventually appearing in the public mempool** (one of those transactions is sandwiched)
- In some cases, after one of these transactions appears in the public mempool, it is **included on chain by the same builder that held the transaction exclusively from the start.**
- Users' transactions are first delayed and then exposed to attacks.

Sharing policy of a private RPC operator?

A possible explanation: a private RPC shares transactions exclusively with one builder and falls back to the public mempool only after that builder loses several slots in a row.

Result 2: Execution Quality is Heterogeneus across Candidate Blocks

The execution of the same swap transaction (fail/succeed and price received if successful) varies across proposed blocks, and the distribution is very skewed:

- 1.7% swaps fail in at least half of the blocks in which they appear, and 1.1% fail in more than 80% of such blocks

Result 2: Execution Quality is Heterogeneus across Candidate Blocks

The execution of the same swap transaction (fail/succeed and price received if successful) varies across proposed blocks, and the distribution is very skewed:

- 1.7% swaps fail in at least half of the blocks in which they appear, and 1.1% fail in more than 80% of such blocks
- For 50% of swaps, execution prices are quite concentrated: at least two-thirds of realizations deviate by less than 4.3 basis points from the swap's average execution price across blocks.

Result 2: Execution Quality is Heterogeneus across Candidate Blocks

The execution of the same swap transaction (fail/succeed and price received if successful) varies across proposed blocks, and the distribution is very skewed:

- 1.7% swaps fail in at least half of the blocks in which they appear, and 1.1% fail in more than 80% of such blocks
- For 50% of swaps, execution prices are quite concentrated: at least two-thirds of realizations deviate by less than 4.3 basis points from the swap's average execution price across blocks.
- For 7% of swaps, execution prices are highly dispersed: at least one-third of realizations deviating by more than 100 basis points from the swap's average.

Result 2: Execution Quality is Heterogeneus across Candidate Blocks

The execution of the same swap transaction (fail/succeed and price received if successful) varies across proposed blocks, and the distribution is very skewed:

- 1.7% swaps fail in at least half of the blocks in which they appear, and 1.1% fail in more than 80% of such blocks
- For 50% of swaps, execution prices are quite concentrated: at least two-thirds of realizations deviate by less than 4.3 basis points from the swap's average execution price across blocks.
- For 7% of swaps, execution prices are highly dispersed: at least one-third of realizations deviating by more than 100 basis points from the swap's average.
- We perform regression analysis to decompose this variance

Result 2: Execution Quality is Heterogeneus across Candidate Blocks

The execution of the same swap transaction (fail/succeed and price received if successful) varies across proposed blocks, and the distribution is very skewed:

- 1.7% swaps fail in at least half of the blocks in which they appear, and 1.1% fail in more than 80% of such blocks
- For 50% of swaps, execution prices are quite concentrated: at least two-thirds of realizations deviate by less than 4.3 basis points from the swap's average execution price across blocks.
- For 7% of swaps, execution prices are highly dispersed: at least one-third of realizations deviating by more than 100 basis points from the swap's average.
- We perform regression analysis to decompose this variance

blockchain is far from deterministic when we include the transaction inclusion process

Result 3: novel insights about competition between arbitrageurs (i.e., Rsync-bot and Titan-bot):

- Two bots that account for 35% of transaction; on average 60% of the value of block in which they appear.

Result 3: novel insights about competition between arbitrageurs (i.e., Rsync-bot and Titan-bot):

- Two bots that account for 35% of transaction; on average 60% of the value of block in which they appear.
- They appear "integrated" with two builders: we call them *Titan-bot* and *Rsync-bot*.

Result 3: novel insights about competition between arbitrageurs (i.e., Rsync-bot and Titan-bot):

- Two bots that account for 35% of transaction; on average 60% of the value of block in which they appear.
- They appear “integrated” with two builders: we call them *Titan-bot* and *Rsync-bot*.
- We can derive an implied CEX price and compare it to Binance.

Result 3: novel insights about competition between arbitrageurs (i.e., Rsync-bot and Titan-bot):

- Two bots that account for 35% of transaction; on average 60% of the value of block in which they appear.
- They appear "integrated" with two builders: we call them *Titan-bot* and *Rsync-bot*.
- We can derive an implied CEX price and compare it to Binance.
- On the WETH/USDC and WETH/USDT markets, the bots trade at 2.8 bp *better* than the contemporaneous Binance price.

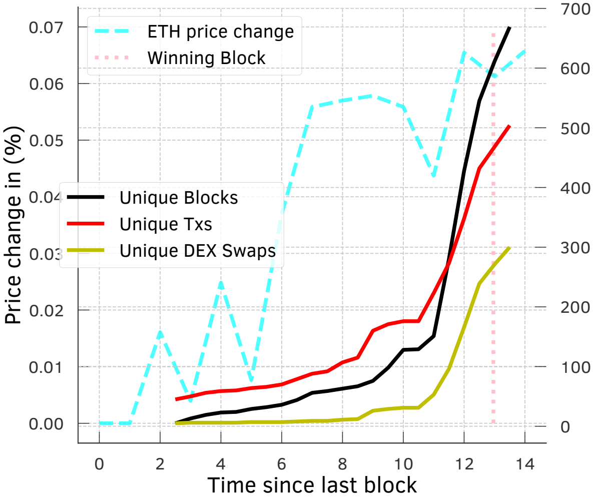
Outline

1. Auction cycle for Block 21322649 (nice graphs)
2. All data (regressions)
3. Relevant literature & Conclusions

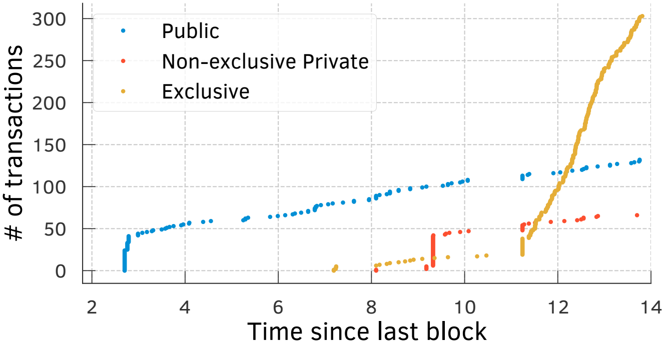
The auction cycle for block 21322649

Why this auction cycle?

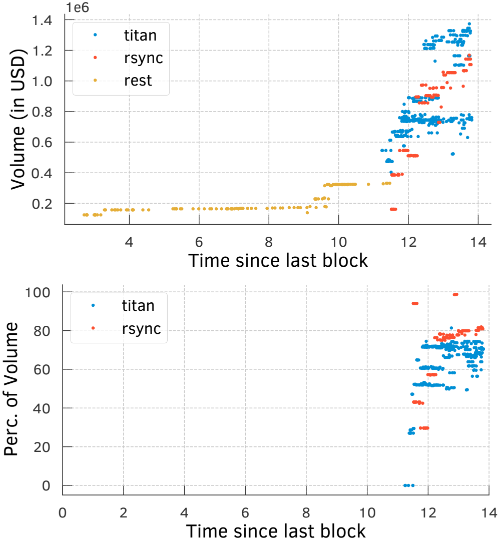
- Intense competition between Rsync and Titan builders *across all relays*.
    - Rsync wins in the end.
- Rsync-bot and Titan-bot compete on the same DEX pools and are responsible for 94%–97% of the total value of the blocks built by their respective builders.



Number of Unique transactions, blocks and swaps vs ETH price change on Binance, during the auction cycle for Block 21322649

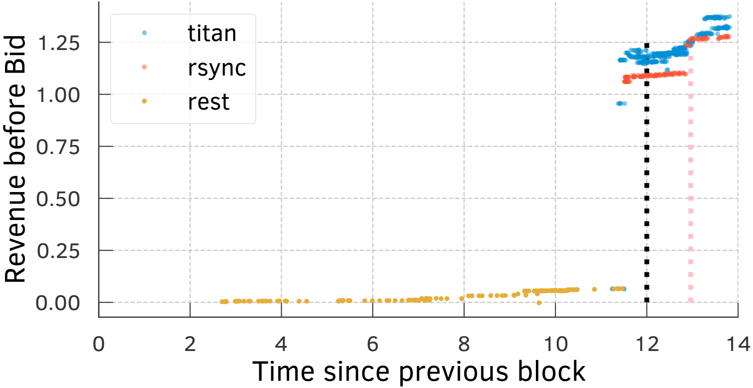


Number of Unique Public, Private, and Exclusive transactions



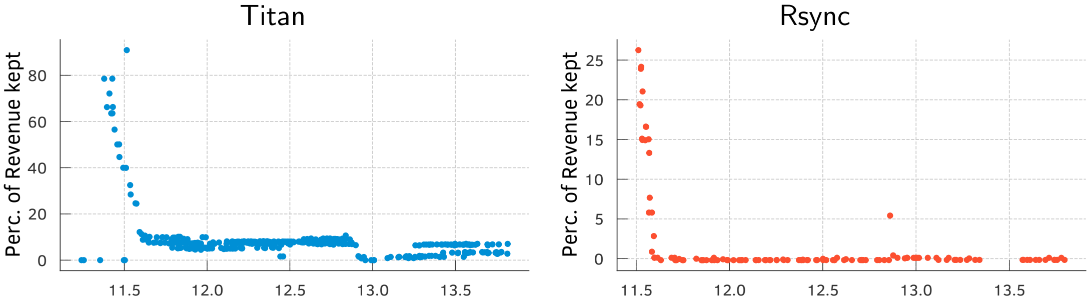
Swap volume per block and percentage attributable to Rsync-bot and Titan-bot

Value of blocks in our data (i.e., sum of all the payments to the builder)



Rsync and Titan build all the high-value blocks (in our sample)

Percentage of block value retained (vs bid)



The fraction of the value retained rapidly converges to 0% for both builders
→ Rsync is more aggressive

Competition between Rsync-bot and Titan-bot

Rsync-bot

- 42 unique transactions (5 included on chain)
- five DEX pools: USDC/WETH, USDT/WETH, MOGCOIN/WETH (2 pools), and WBTC/cbBTC

Titan-bot

- 181 unique transactions (none included on chain)
- five DEX pools: USDC/WETH, USDT/WETH, MOGCOIN/WETH (2 pools), and MATIC/WETH

Competition between Rsync-bot and Titan-bot

Rsync-bot

- 42 unique transactions (5 included on chain)
- five DEX pools: USDC/WETH, USDT/WETH, MOGCOIN/WETH (2 pools), and WBTC/cbBTC

Titan-bot

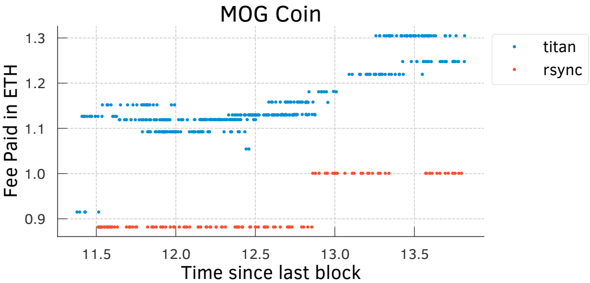
- 181 unique transactions (none included on chain)
- five DEX pools: USDC/WETH, USDT/WETH, MOGCOIN/WETH (2 pools), and MATIC/WETH

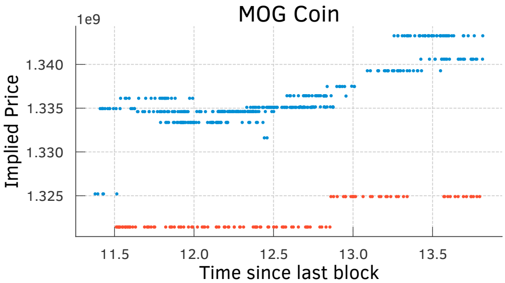
Competition between bots: Both sell MOGCoin for WETH and sell WETH for USDC/USDT

Risk-adjusted implied CEX prices

- We observe the amount swapped on the DEX v; the price of execution on the DEX p_{DEX}; and the fee attached to each transaction $=$ willingness to pay for inclusion $=$ **risk adjusted** profits π
- Depending on the direction of the trade, we have $\pi = v \cdot (p_{\text{CEX}} - p_{\text{DEX}})$ or $\pi = v \cdot (p_{\text{DEX}} - p_{\text{CEX}})$
- We can then back up a risk-adjusted implied CEX price and compare it to Binance (if it exists)

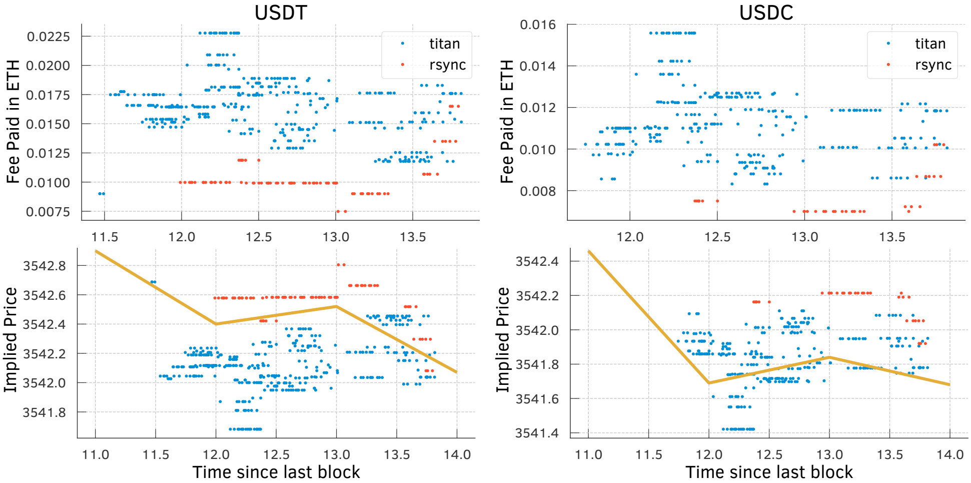
Competition on the MOGCoin/WETH Uni V3 pool.





Large expected profits, Titan seem more efficient in the DEX leg of the trade

Competition on the WETH/USDC and WETH/USDT Uni v3 pools.



Fee for inclusion and implied DEX price **compared to Binance price**

The auction cycle for block 21322649: summary of results

Determinants of who wins the block:

- Titan-bot seems more efficient (i.e., trades at better prices on the CEX), especially on the most valuable arbitrage opportunity.
- But Rsync builders bids more aggressively (and wins)

All data

Probability of successful execution (only swaps, with transactions FE)

	All Swaps	Same Dir. as Bots	Opp. Dir. as Bots
Time Since Last Block	-0.0038***	-0.0217***	-0.0012***
	(0.0001)	(0.001)	(0.000)
Tx Index	0.0001***	0.0006***	5.74e-06
	(0.0000)	(3.18e-05)	(9.23e-06)
Is Titan Builder	-0.0078***	-0.1341***	-0.0144***
	(0.0008)	(0.005)	(0.001)
Is Rsync Builder	-0.0148***	-0.1592***	-0.0021
	(0.0006)	(0.006)	(0.002)
Has Titan-bot tx	0.0024***	-0.0423***	0.0286***
	(0.0005)	(0.004)	(0.001)
Has Rsync-bot tx	0.0075***	-0.0301***	0.0095***
	(0.0005)	(0.004)	(0.001)
Observations	318742	17224	26705
R-squared	0.744	0.556	0.493
Tx fixed effect	yes	yes	yes

Execution quality for swaps (higher values are better for the user)

	All Swaps	Same Dir. as Bots	Opp. Dir. as Bots
Time since Last Block	-0.0022***	-0.0430***	0.0163***
	(0.0005)	(0.002)	(0.002)
Tx Index	0.0003***	0.0001***	0.0007***
	(0.0000)	(4.01e-05)	(4.28e-05)
Is titan builder	-0.0053**	-0.1206***	0.0923***
	(0.0024)	(0.006)	(0.007)
Is rsync builder	0.0198***	-0.0924***	0.2758***
	(0.0031)	(0.008)	(0.009)
Has titan-bot tx	0.0048**	0.0068*	-0.0640***
	(0.0019)	(0.005)	(0.006)
Has rsync-bot tx	-0.0235***	-0.0038	-0.0402***
	(0.0019)	(0.005)	(0.006)
Observations	342528	15404	25718
R-squared	0.002	0.142	0.073
Tx fixed effect	yes	yes	yes

Implied CEX price: correlation with Binance (only WETH/USTD and WETH/USDC)

	Coefficient	Standard Error
Binance Price (0 sec.)	0.7945^{***}	0.108
Binance Price (+1 sec.)	-0.1491	0.129
Binance Price (-1 sec.)	0.2513^{**}	0.108
Binance Price (-2 sec.)	0.0972	0.063
Constant	21.5424	23.070
Observations	89	
R-squared	0.998	

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Price improvement over contemporaneous binance price

Time since Last Block	-0.009	-0.001
	0.001	0.001
Is Titan Bot	-0.0069**	-0.007**
	0.003	0.003
Volume × (Volume lower than 50)		-0.0007
		0.004
Volume × (Volume between 50 and 100)		0.0003
		0.004
Volume × (Volume between 100 and 150)		0.0066
		0.004
Volume × (Volume greater than 150)		0.0231***
		0.008
Constant	0.028	0.0287**
	0.018	0.013
Observations	89	89
R-squared	0.080	0.195

Relevant literature & Conclusions

Relevant literature

- Value of blocks vs bids: Yang, Nayak, and Zhang (2024),
- Competition between arbitrageurs pre-PBS: Capponi, Jia, and Yu (2024) (see also Capponi, Jia, and Wang, 2023)
- Quantifying arbitrage volume and arbitrage profits: Heimbach, Pahari, and Schertenleib (2024), Canidio and Fritch (2023), Fritch and Canidio (2024)
- Use "failed" transactions to identify HFT races and profits: Aquilina, Budish, O'Neill (2022)
- Builder-searcher integration: Pai and Resnick (2024),
- Builders' competition: many

Check out our other paper (with the same data): "How Exclusive are Ethereum Transactions? Evidence from non-winning blocks"

Conclusions

- The set of transactions included on-chain looks very different from the set of transactions submitted for inclusion,
- A surprisingly large number of users' transactions are delayed because they are exclusive (or not shared with all builders)
- Some indication that an OFA provider shares users' transactions exclusively with a single builder, and then releases them in the public mempool after 1 or 2 auction cycles if this builder does not win
- Novel insights relative to searchers' competition, in particular to the CEX leg of CEX-DEX arbitrage