MIT Digital Currency Initiative and the University of Brasilia presents

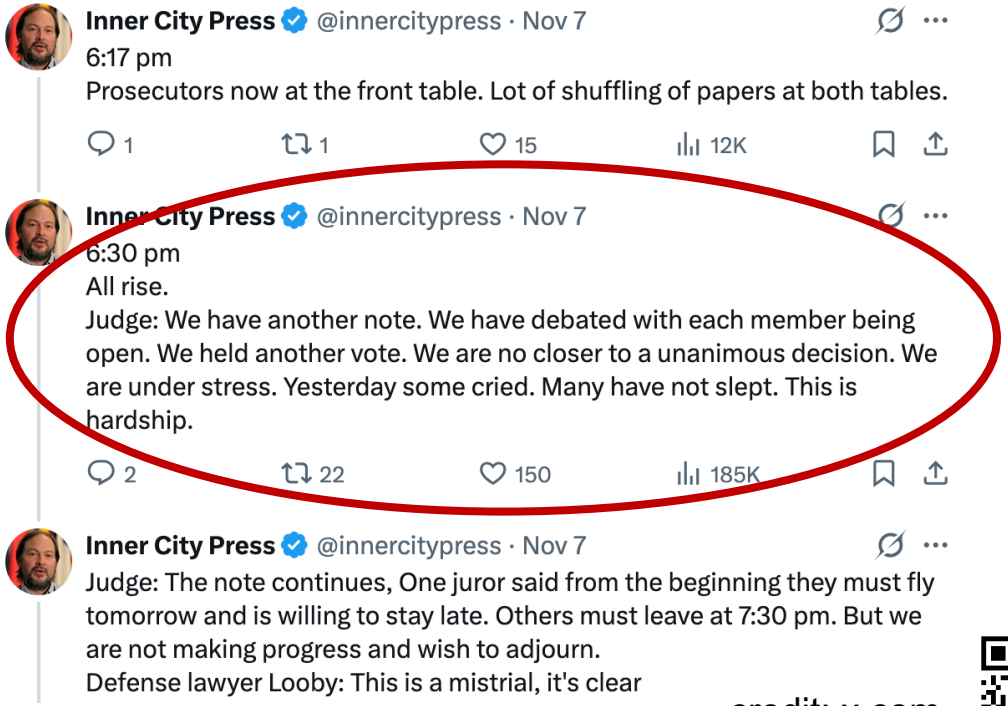
Cryptocurrency Design and Engineering

Lecture 20: Unfair Behaviors: Maximal Extractable Value (MEV)
Taught by: Jason Milionis (Columbia University & Category Labs)
Date: 12/02/2025
MAS.S62

MEV Makes People Cry

US v Peraire-Bueno

Mistrial declared



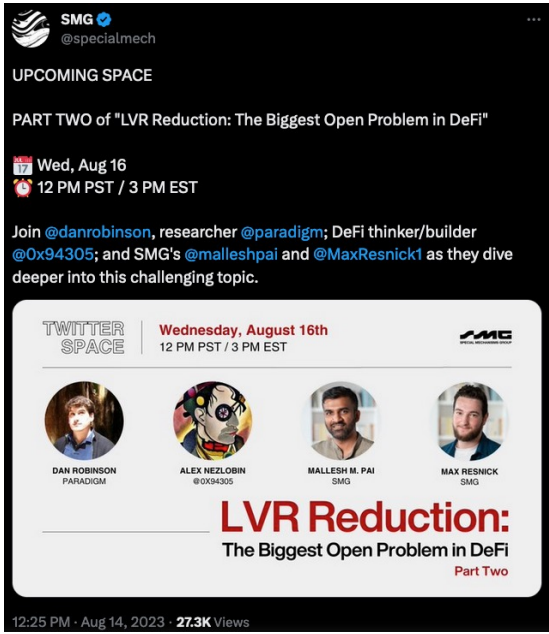
Inner City Press @innercitypress · Nov 7
6:17 pm
Prosecutors now at the front table. Lot of shuffling of papers at both tables.
1 1 15 12K

Inner City Press @innercitypress · Nov 7
6:30 pm
All rise.
Judge: We have another note. We have debated with each member being open. We held another vote. We are no closer to a unanimous decision. We are under stress. Yesterday some cried. Many have not slept. This is hardship.
2 22 150 185K

Inner City Press @innercitypress · Nov 7
Judge: The note continues, One juror said from the beginning they must fly tomorrow and is willing to stay late. Others must leave at 7:30 pm. But we are not making progress and wish to adjourn.
Defense lawyer Looby: This is a mistrial, it's clear

credit: x.com

MEV is the Top Issue in Modern App Design



credit: x.com

Today

Previous lectures: transaction fees & mempool inner workings, Ethereum's smart contract programming model

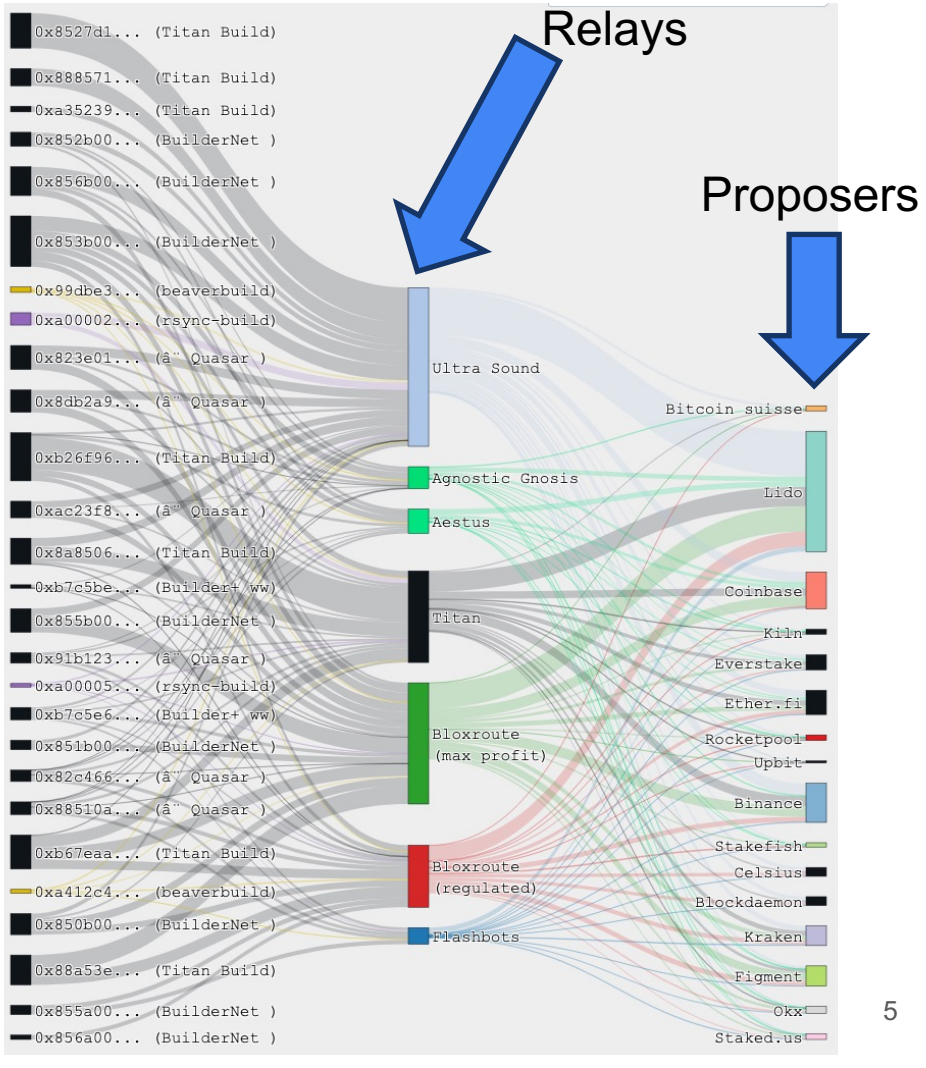
Today: combine → how they interact, value derived from the public mempool "trades"

1. MEV definition: originally called "~~Miner~~ Extractable Value," now "Maximal"
2. Deep dive into most important types of MEV in DEXs
3. Who gets the MEV?
4. How MEV affected the landscape of competitive block building / transaction validation (Proposer – Builder Separation, PBS)

MEV Supply Ecosystem

Proposer-Builder Separation on Ethereum

Builders



- Why so few relays?
- Why builders with same tag names?

Credit: screenshot from mevboost.pics

Motivation: the Blockchain Ecosystem

Applications on top of blockchains (**dApps**): trillions $ used

Enable functionality: implement traditional primitives in purely automated, transparent way

- This economic activity generates value in reordering / including transactions.

Decentralized Finance (DeFi)

- Decentralized Exchanges (**DEXs**)
- Example: trading via Automated Market Makers (**AMM** ∈ dApps) → most MEV

Maximal Extractable Value (MEV)

Definition: surplus derived from control over **ordering/insertion/exclusion** of transactions in a block

Arbitrage = (the practice of) taking advantage of some discrepancy/inefficiency (e.g., in price across venues), resulting in (risk-free = guaranteed) profit

Arbitrageurs

Flash Boys 2.0:
Frontrunning, Transaction Reordering, and
Consensus Instability in Decentralized Exchanges

Philip Daian, *Cornell Tech*, phil@cs.cornell.edu
Steven Goldfeder, *Cornell Tech*, goldfeder@cornell.edu
Tyler Kell, *Cornell Tech*, sk3259@cornell.edu
Yunqi Li, *UIUC*, yunqil3@illinois.edu
Xueyuan Zhao, *CMU*, xyzhao@cmu.edu
Iddo Bentov, *Cornell Tech*, ib327@cornell.edu
Lorenz Breidenbach, *ETH Zürich*, lorenz.breidenbach@inf.ethz.ch
Ari Juels, *Cornell Tech*, juels@cornell.edu

MEV Taxonomy (Moallemi, 2024)

Intrinsic

- Internal blockchain inconsistency (fully on-chain)
- “DEX-DEX” cyclic arbitrage
- “Sandwich” MEV on DEXs
- Liquidations (lending protocols)
- NFT mints (even, e.g., Bitcoin “inscriptions”)

Extrinsic

- Inconsistency between blockchain and the “outside world”
- “CEX-DEX arbitrage” (LVR)
- “Multi-block” MEV

Trading in Decentralized Exchanges (DEXs)

DEXs: On-chain contracts that anyone can transact to trade one asset for another. How?

Most common paradigm: Automated Market Makers (AMMs)

Trading via an AMM [Lu and Köppelmann, 2017]

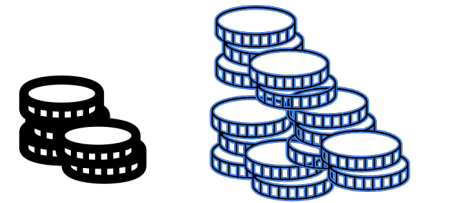
Liquidity providers (LPs) supply pools of assets, e.g., ETH and USDC

- Inspired by use of AMMs for prediction markets [Hanson, 2002]

AMM offers a bunch of allowable trades

- Anyone transacting/trading with an AMM must follow a rule/invariant.
- This (automated) formula is used to quote prices for trades
- Example: say I want ETH, will pay in USDC → what needs to happen?
- Anyone can select their favorite trade, and get/give the difference
- Price determined by current level of reserves

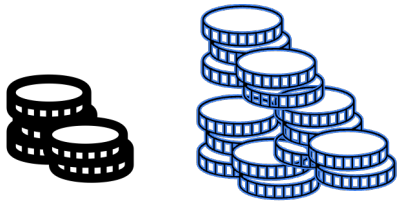
Trading via an AMM [Lu and Köppelmann, 2017]



state 1

Get 1 ETH

Give 6,000 USDC



state 2

+ trading fee

Example rule: **constant product market maker**, product of pool reserves = constant/invariant

- In this case, (marginal) price is determined by the ratio of pool reserves.

AMM Prices

AMM price is determined solely by current reserves

- Executed trades change reserves ⇒ AMM price changes

Spot (marginal) price of AMM before trade: 4,000 USDC/ETH

Average price paid for trade: 6,000 USDC/ETH

Spot (marginal) price of AMM after trade: 9,000 USDC/ETH

“LVR” (CEX-DEX) Arbitrage in AMMs

Problem: AMM does not react to information (external conditions change)

- What if average price someone gets for trade ≠ external price?
- E.g., previous example, if price on Coinbase is 9,000 USDC/ETH?
- Got the trade for 6,000 USDC/ETH. So, profit = (1 ETH) x (9,000 – 6,000)

Adverse selection: Trading with (better) informed traders

- Aka, “CEX-DEX” arbitrage: most well-known, profit ≥$150M/year

The cost of adverse selection in AMMs is called:

Loss-versus-Rebalancing (**LVR**; pronounced “lever”)

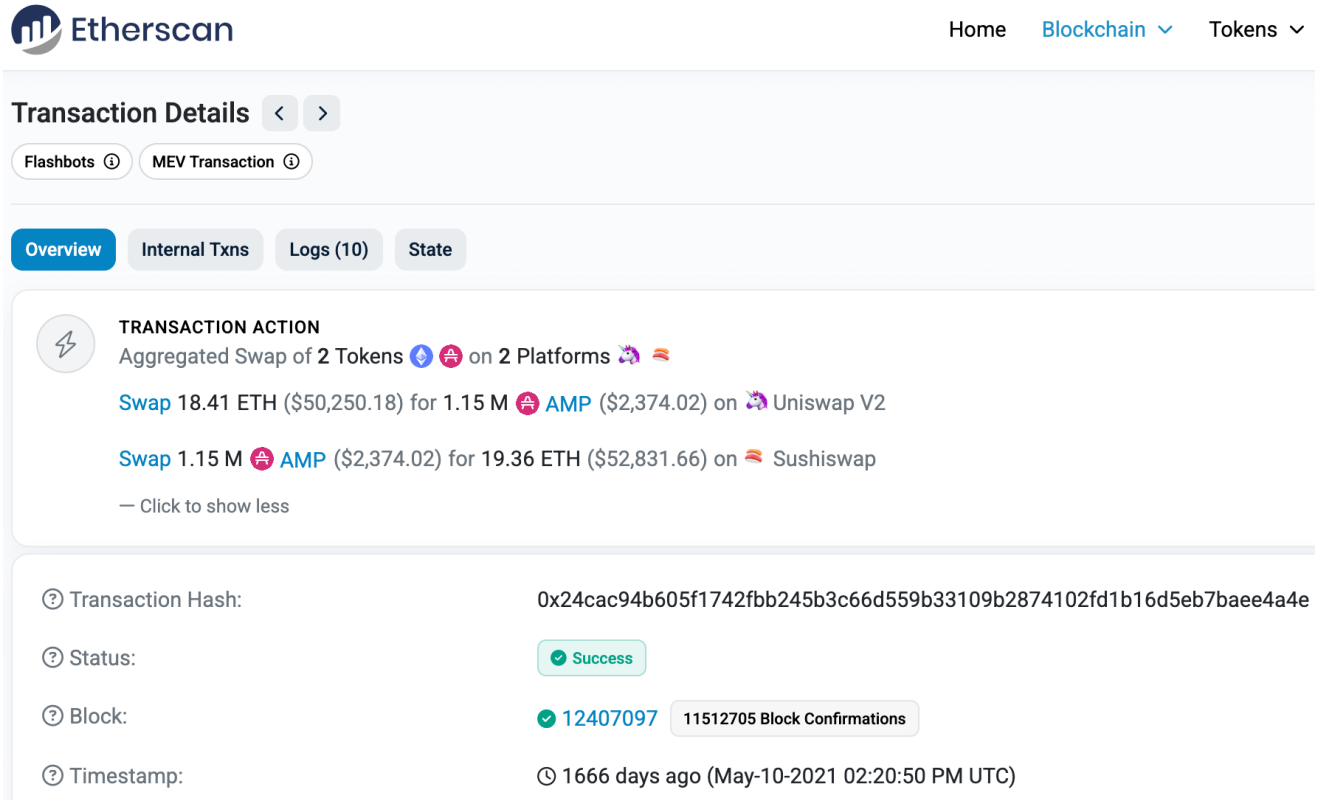
[Milionis, Moallemi, Roughgarden, Zhang, 2022]

Another Form of Arbitrage in AMMs: Cyclic arbitrage

- Aka “DEX-DEX” arbitrage
- Again, many different venues on-chain → arbitrageur correct their prices

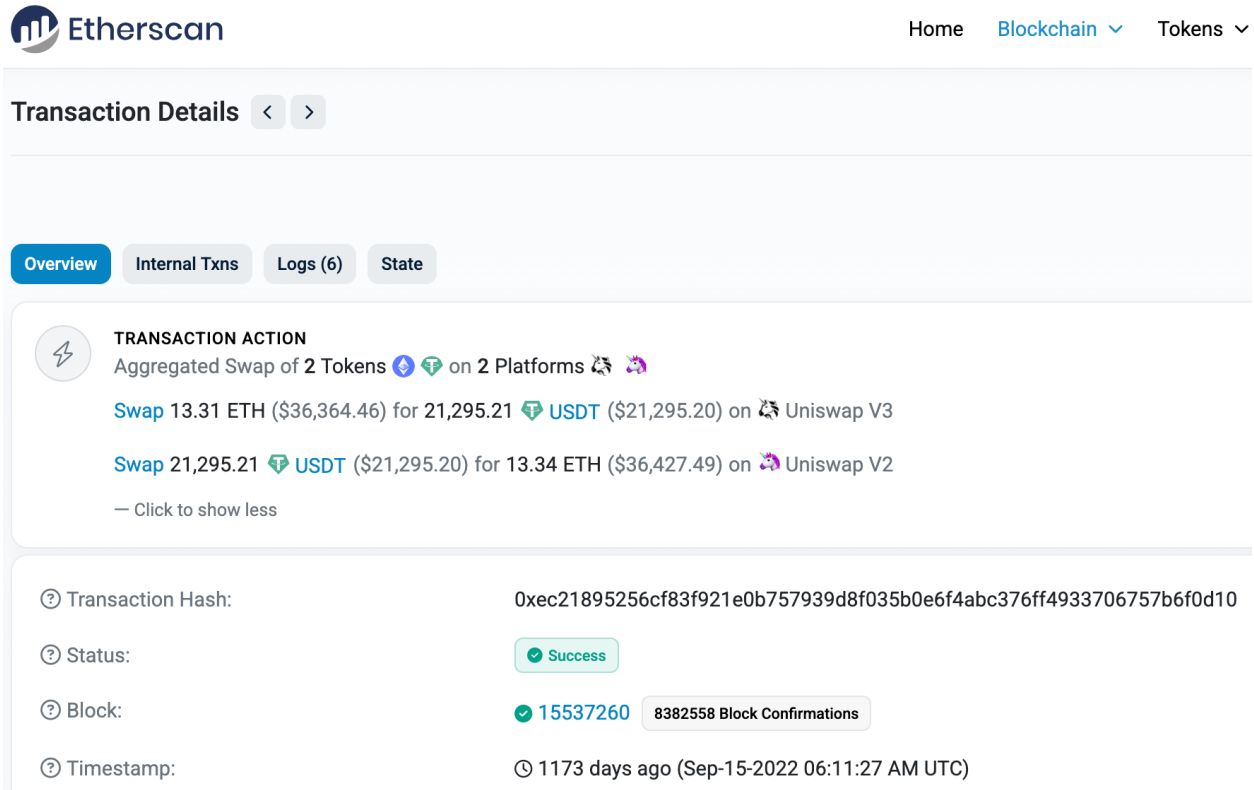
- Algorithms: “cycle detection” (find negative cycles in a graph)
  - Sometimes, takes a lot of time to compute → on-chain “MEV contracts” that get triggered with some external data by a bot
  - Why? Buys a few extra (milli)seconds, potentially more efficient (smaller)

Example 1



Credit: Etherscan

Example 2



Credit: Etherscan

“Sandwich” Arbitrage on AMMs

Large trades move the price a lot: “**slippage**”

Due to uncertainty; someone will take advantage, making the rate “worst-case”

- “Maximum slippage tolerance” must be set before the trade posts as a transaction

“**Sandwich**:” consists of a frontrun + a backrun transaction, the trade is in between

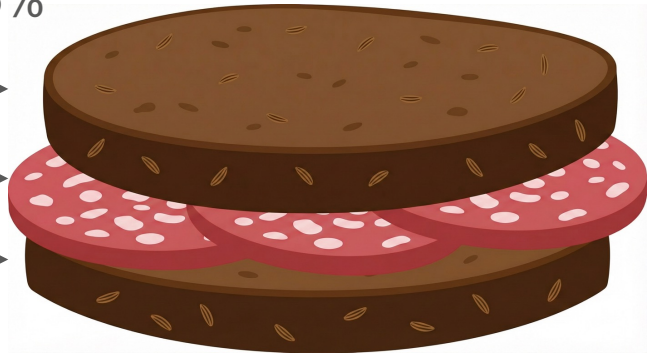
Example: “buy A, sell B”, max slippage tolerance 0.5%

Frontrun: “buy A, sell B” until tight →

Substance of sandwich (transaction) →

Backrun: “buy B, sell A” →

⇒ profit in tokens



A Notorious “Sandwich” (Ethereum Block 22029771)

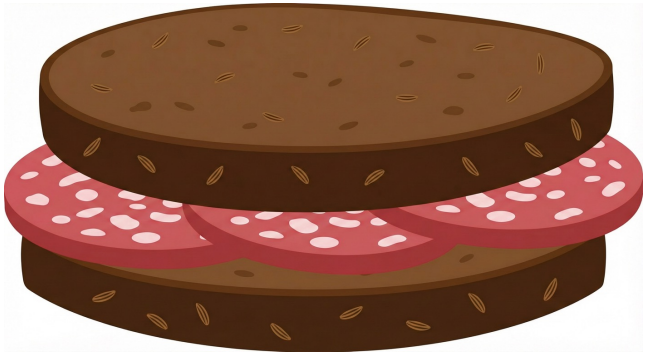
Backrun

Salami

Frontrun

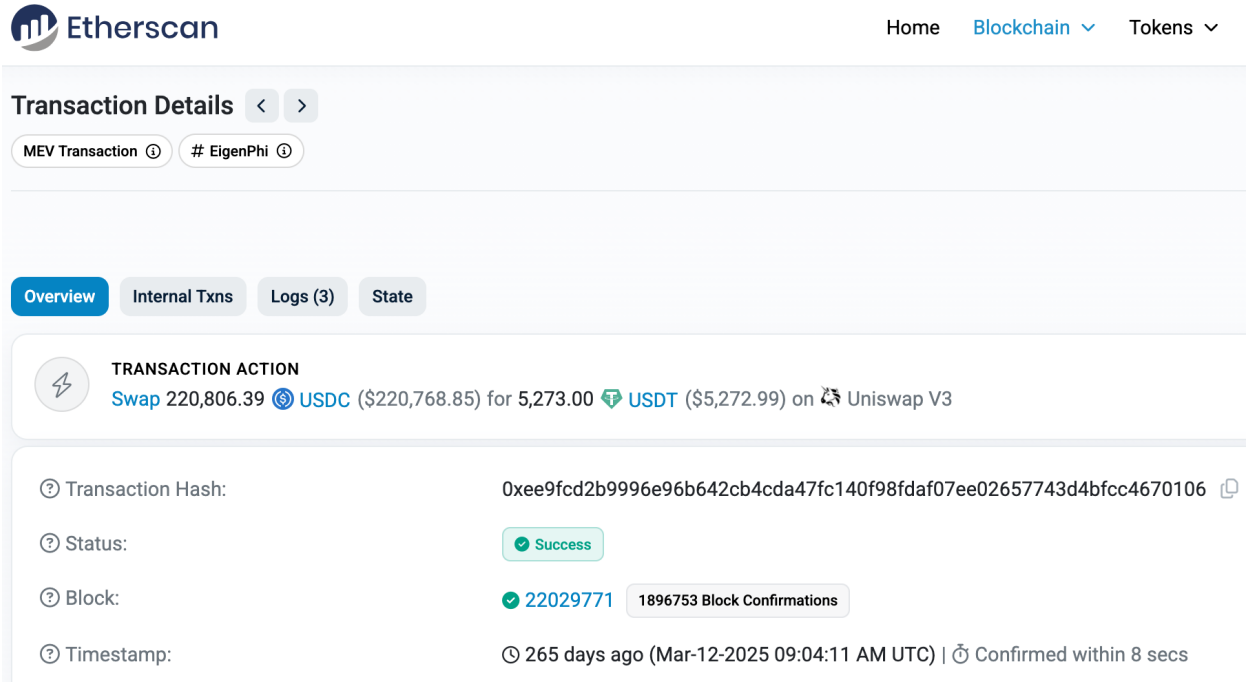
0x3e72f3ad09d...	0xa7b8e688	22029771	265 days ago	0x26cE7c19...1B08850aF	→	MEV Bot: 0x000...B...
0xee9fcd2b999...	Exact Input Si...	22029771	265 days ago	0x5A89D040...40906eC5D	→	Uniswap V3: Router
0xde5aa0c152...	0xd2925723	22029771	265 days ago	0x26cE7c19...1B08850aF	→	MEV Bot: 0x000...B...

(this is showing reverse order in the block)



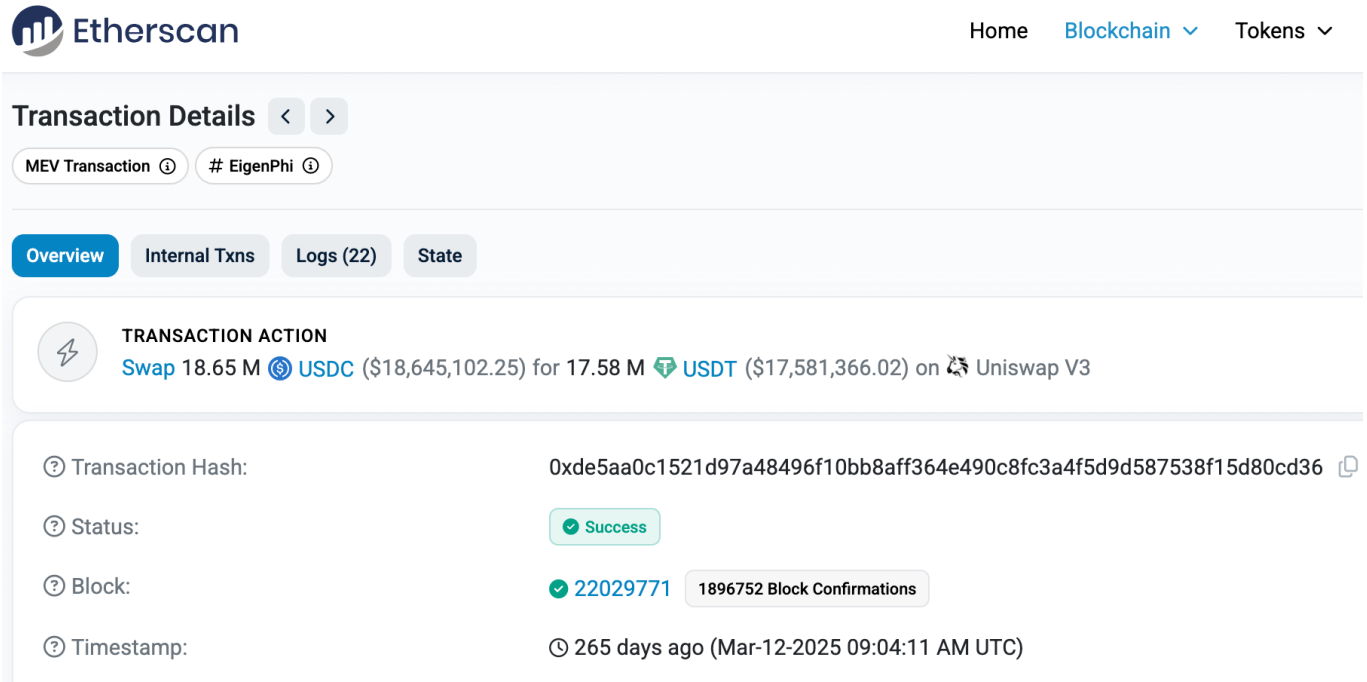
Credit: Etherscan

A Notorious “Sandwich:” “The Trade”



Credit: Etherscan

A Notorious “Sandwich:” “The Frontrun”



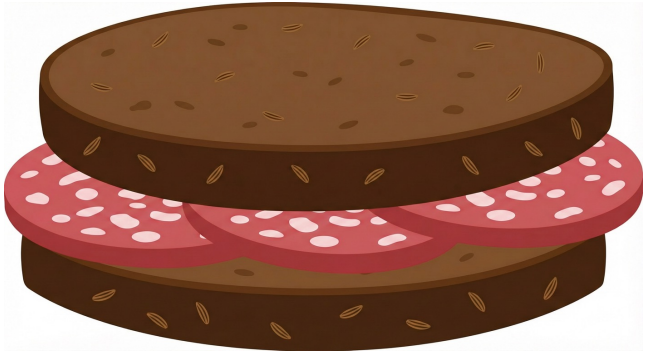
(the backrun is a bit harder to see)

Credit: Etherscan

“Sandwich” Arbitrage on AMMs

Risk: needs to be run atomically + individualized to the “sandwiched” trade

- Okay if trade can be seen on “public mempool”
- Arbitrageur submits the “whole sandwich” (needs to be tight) for inclusion in the block
- How? See next (PBS).



Who Gets this Value? MEV Affects Transaction Fees

MEV requires ordering/insertion/exclusion control

- Validators are the only ones who can guarantee execution of a profitable transaction

But validators are* ordering transactions according to their priority fee (per gas)

⇒ MEV bots keep bidding up the transaction fees to get prioritized execution (higher in the block, before the other MEV bots → opportunity vanishes if later)

⇒ MEV flows (i.e., profits are outsourced) to Validators

Side-effect 1: The “failed” (“reverted” in the language of Ethereum) MEV bots’ transactions are still posted on-chain → inefficiency, blockchain is “spammed”

*normally

Who Gets this Value? MEV Affects Transaction Fees

Side-effect 2: MEV transactions are auto-generated via bots → if these are in the public mempool, what if other bots copy these transactions + replace the "profiting" address with their own? "frontrunning the MEV bot" + obfuscation

- Endgame: what if the validator (monopoly power over block production) does that?

Side-effect 3 (consensus instability): generally theoretical concern (parallel to argument of "transaction fees ≫ block reward"), if at a block MEV ≫ block reward, then future validators have incentive to fork (to frontrun the past MEV)

- Especially in PoW like Bitcoin; less applicable in PoS due to finality guarantees / no effective reallocation of effort

Market Response: Proposer – Builder Separation, PBS

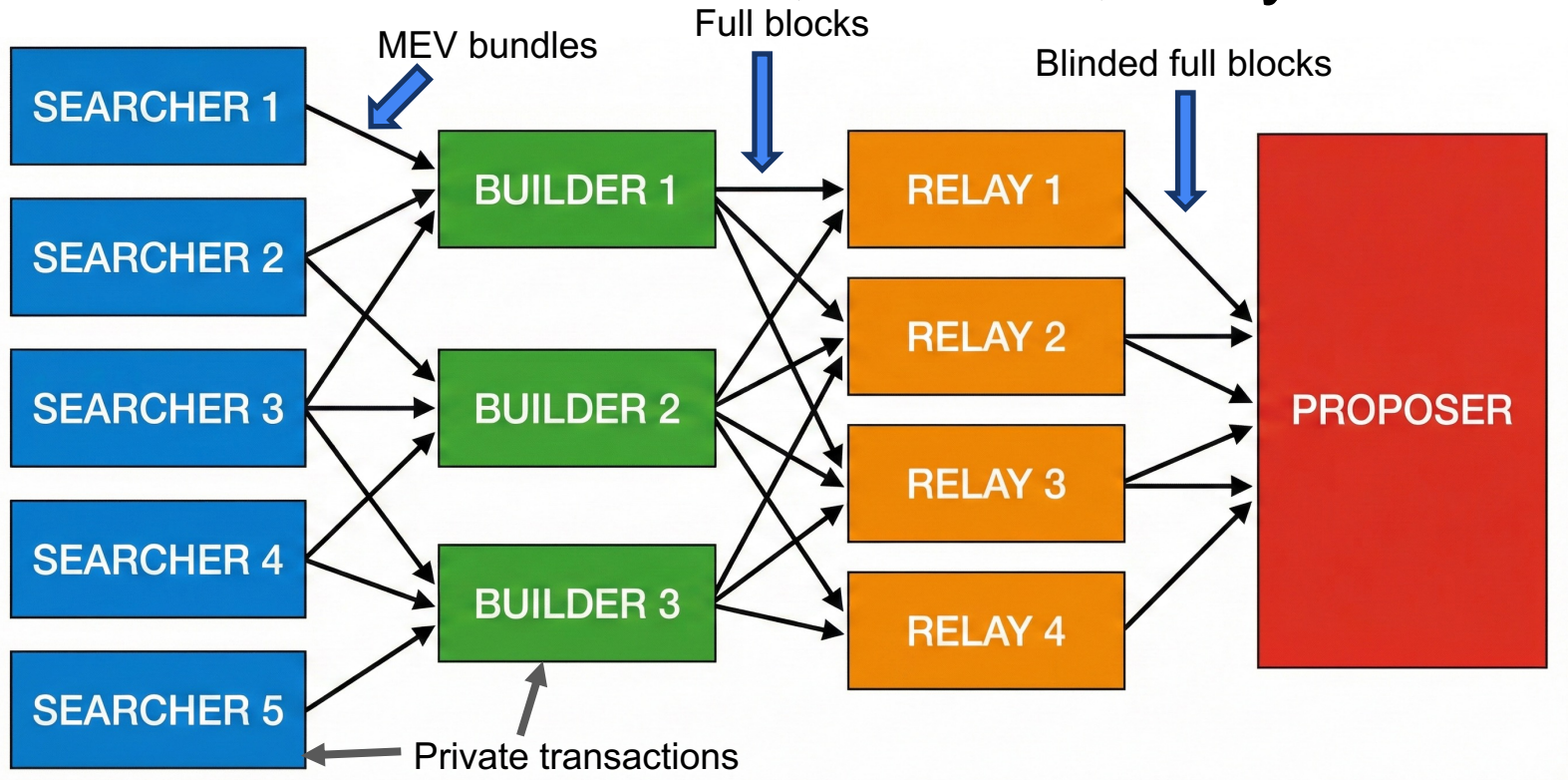
Outsource “block” production to external parties (“**builders**”) in exchange for $ payment to the validator (“proposer” in proof-of-stake) via middleman entity “**relay**”

- Proposed by Flashbots (building the relay system), originally called “MEV-Boost”
- Relays play dual role: 1. accept private blocks (from “builders”) containing MEV transactions, 2. keep proposer accountable to actually sign the block they committed to

\+ Prevent reverting transactions (“spam”) from being posted on-chain → keeps gas fees low for normal users, no “gas bidding wars”

\+ Prevent others from copying MEV opportunities

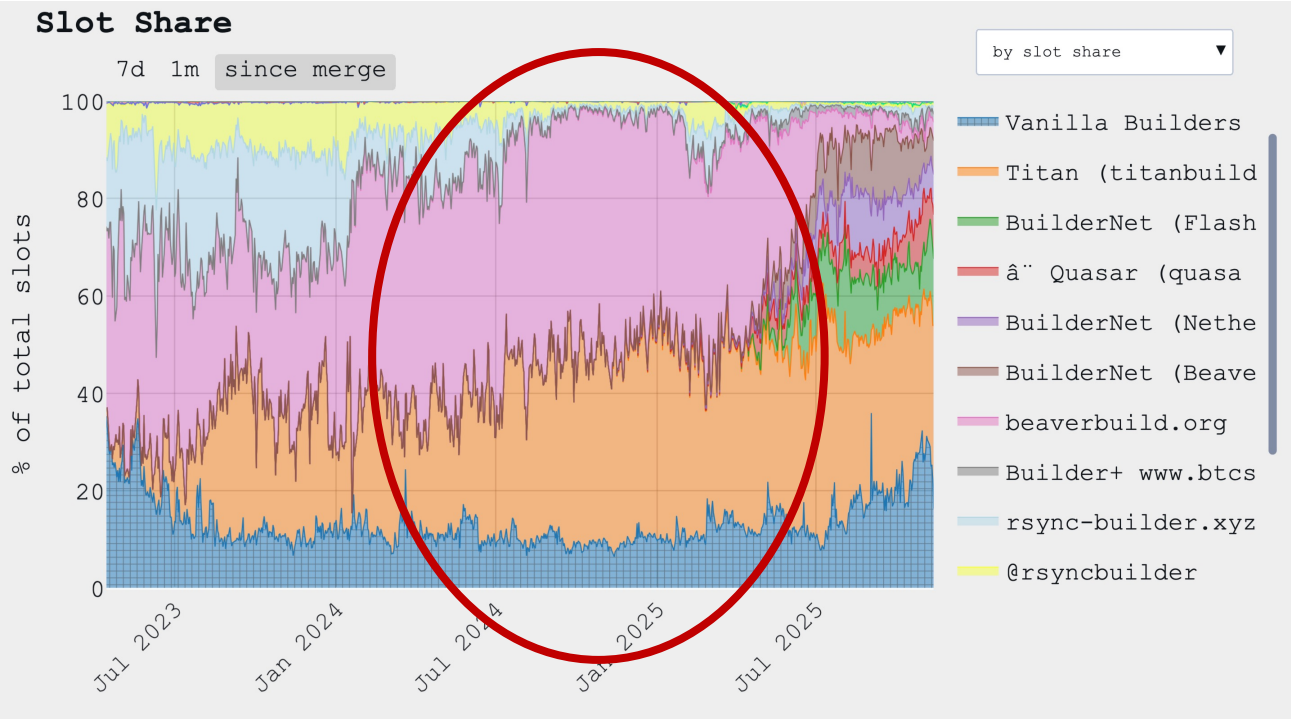
PBS in a Picture: searchers, builders, relays 🥲



Market Response: Proposer – Builder Separation, PBS

+ Solves consensus instability by introducing **competition**, but moves centralization one layer down (to “builders” / “searchers”)
- Who guarantees that the relay will act correctly? Nothing, trust assumption.
  - Competitive landscape of many relays helps. (MEV-Boost allows multiple)
  - What if the relay early-releases a (supposedly blinded) block which is then deemed invalid by system (consensus validation) rules / other proposers?
- Modern evolution: private order flow (transaction submitters cut deals with builders, who provide discounts, but benefit themselves paying less to proposers)

Centralization in Block Building due to MEV



Centralization in Block Building due to MEV

By virtue of becoming good at extracting MEV, end up controlling the production of the entire block.

1. MEV-capturing (necessary optimizations) creates **economies of scale** ⇒ larger builders becoming better at it can capture more extracted value ⇒ have more resources to invest even more + *exclude* other participants' transactions
2. Private order flow increases centralization even more

Future of Block Building on Ethereum

1. Enshrined PBS (ePBS): remove the trusted relays, change the consensus algorithm to separate out the phases of building the block + validating it
2. BuilderNet: proposal for open block-building, eliminates “private order flow” deals
    - Remote-attestation Trusted Execution Environments (TEEs) guarantee that “builder code” will build the optimal block from bundles and pay all participants “correctly”
    - Now, can **share** the private order flow in an encrypted fashion with a TEE key (it cannot be frontrun, due to the “guaranteed code execution”)

MEV Solutions

1. Faster blockchains: reduce some types of MEV (e.g., CEX-DEX arbitrage profits [Milionis, Moallemi, Roughgarden, 2023])
2. Better dApp designs to “not leak” MEV
3. MEV capture + refunds (but sometimes might be hard to attribute, e.g., due to composability)
    - E.g., idea: “priority taxes” [Robinson and White, 2024]: if you compete in gas fees, have to pay amount proportional to the priority fee to protocol.

In general, **introduce competition**.

Further topics

- Gas fee competition means that MEV bots attempt to minimize their transaction size
- Integrated searcher-builder advantage: auction dynamics in PBS auctions

Ethics of MEV/PBS

+ Correcting inefficiencies (but if it's easy to do, like CEX-DEX arbitrage, then competition should induce little $ payment for this service)

- Friction for users that increases their costs, lowering welfare

Where should it go?

- Redirecting to validators "improves security," but there should be alternative way to pay for it (modeled, e.g., via transaction fees).
- Validators are not "doing more work" for the MEV, e.g., in more volatile periods.
- Are behaviors "allowable by code" in the blockchain stack acceptable?