Elements of DeFi

https://web3.princeton.edu/elements-of-defi/

Professor Pramod Viswanath

Princeton University

Lecture 8

MEV Deep dive

Last lecture: Improving CFMMs

- Make CFMMs more capital efficient and fair
- Concentrated liquidity – move liquidity around
  - Better deal for both LPs and traders
- Batching + Routing – avoid sandwiching and arbitrage MEV
- Private CFMMs – to avoid MEV

This Lecture: MEV deep dive

- What is MEV?
- Centralizing effects of MEV
- Solutions on Ethereum: Flashbots
- Flashbots Auctions
- Proposer-Builder Separation
- SUAVE

MEV

- MEV: Miner extractable value
  - Maximum extractable value
- Action by Miner to maximize individual benefit
  - Block creation process
  - Ordering of transactions not specified by protocol
- Picking the right ordering of transactions
  - High transaction fees
- Inserting new transactions
  - Frontrunning in markets

Recall : CFMMs

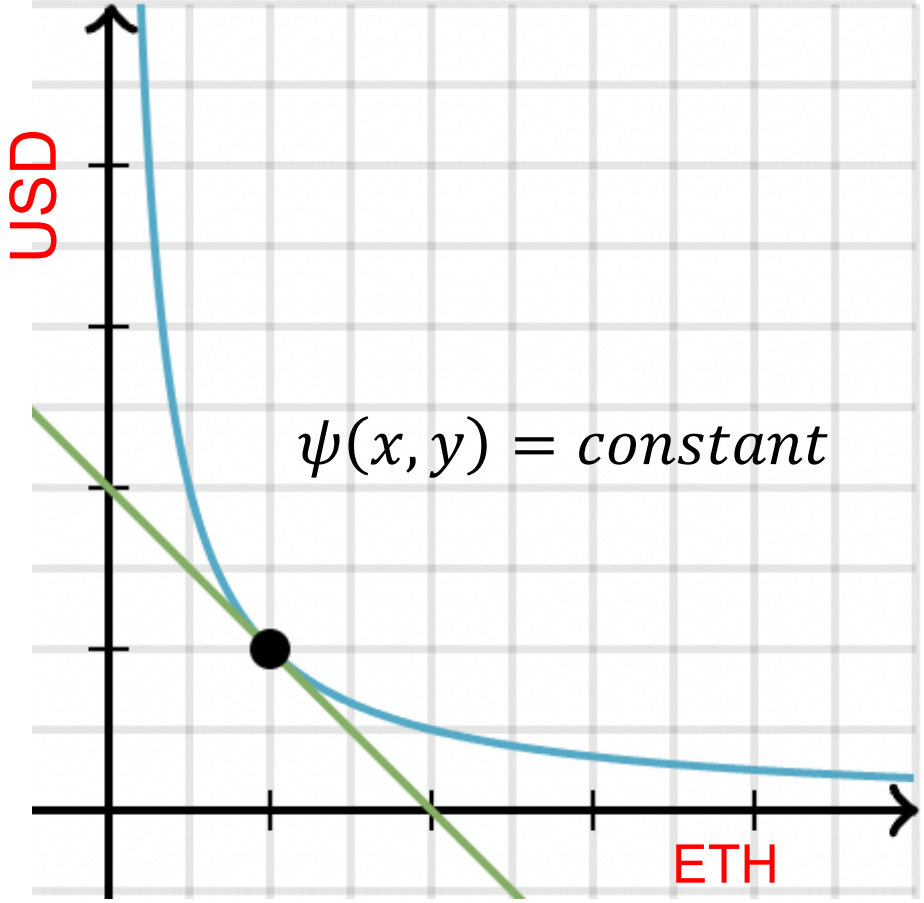
- CFMM: Constant Function Market Makers
- Use Bonding Curves to constrain reserves

$$\psi(x, y) = \psi(x + \Delta_x, y - \Delta_y)$$

OR

$$\psi(x, y) = constant$$

- Slope of the tangent = Price

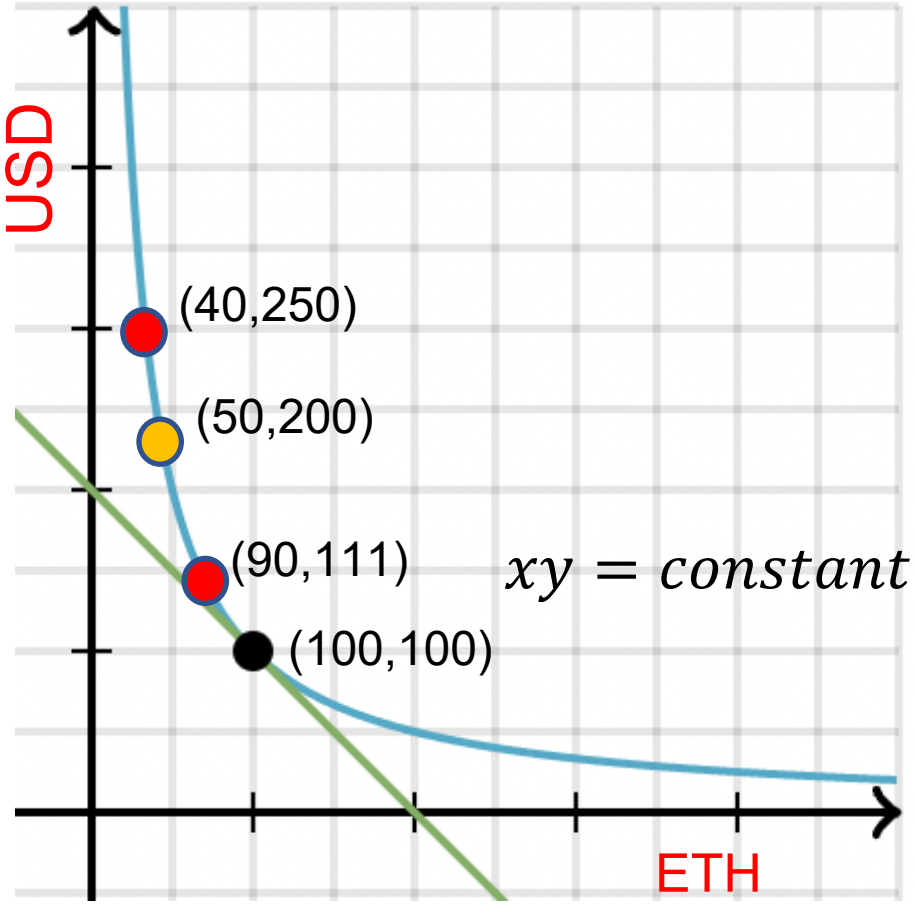


Recall: Front Running

MEV : Sandwich Attack

- User wants to do a normal trade :
  - Buy 50 ETH, (has to pay 100 USD normally)

- If miner sees a large buy txn,
  - Introduce a buy txn just before it : buy 10 ETH
  - Put the txn
  - Introduce a sell txn just after it : sell 10 ETH
- Miner gets profit with no risk : 39 USD
- User gets a worse price : 139 USD



Current Ethereum protocol

- Special nodes – validators – compose, verify, approve new blocks
- For each block, one of them is chosen to be proposer
- Proposer produces a block with valid txns, validators approve it
- Proposer has complete freedom to reorder, insert and censor txns
- The maximum total wealth that can be extracted this way - MEV

Examples of MEV

- Arbitrage:
  - Exploiting price differences to make riskless profit – usually benign
- Liquidations:
  - Like margin calls in TradFi – spot lending positions that are underwater and liquidate them to earn commission
- Frontrunning, Backrunning, Sandwiching:
  - Introducing transactions before/after user swaps to earn risk-free profit
  - Frontrunning is bad for the user and malicious
  - Backrunning is benign since it does not give anyone a worse price and stabilizes market

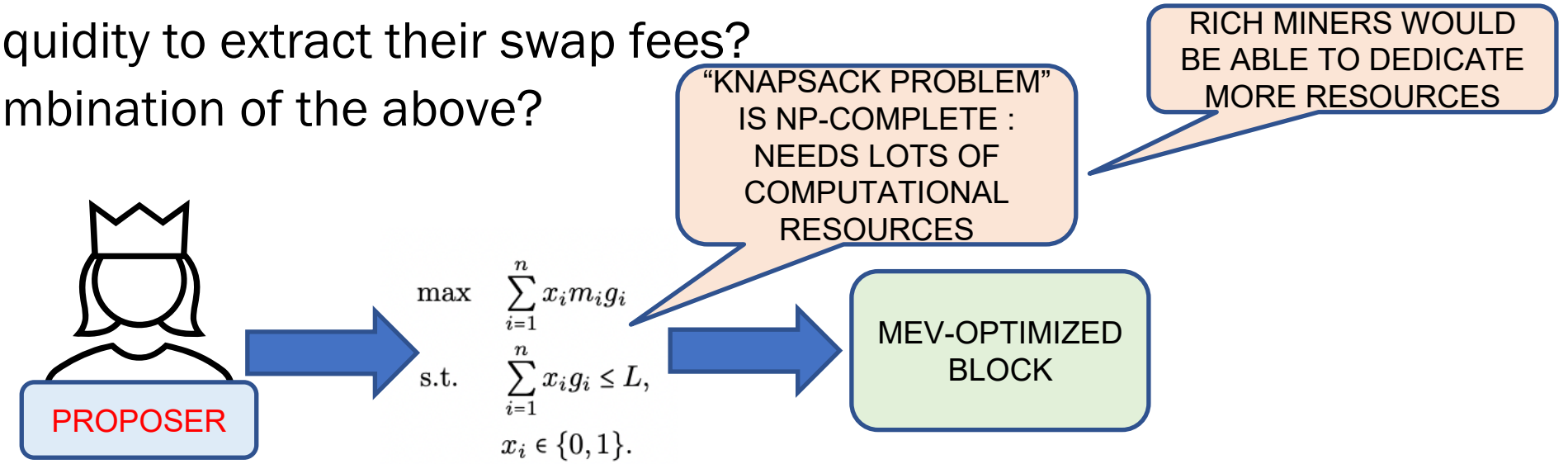
Examples of MEV

- Transaction manipulation:
  - Spot a profitable transaction and copy it, with the original transaction being censored or suffering worse execution
  - Similar attack can be done on a liquidation transaction

- JIT liquidity:
  - Insert liquidity provisioning transaction just before a large trade and pull out liquidity just after it
  - Earns most of the pool fees for the transaction and gives user better market depth
  - Bad for other (passive) LPs

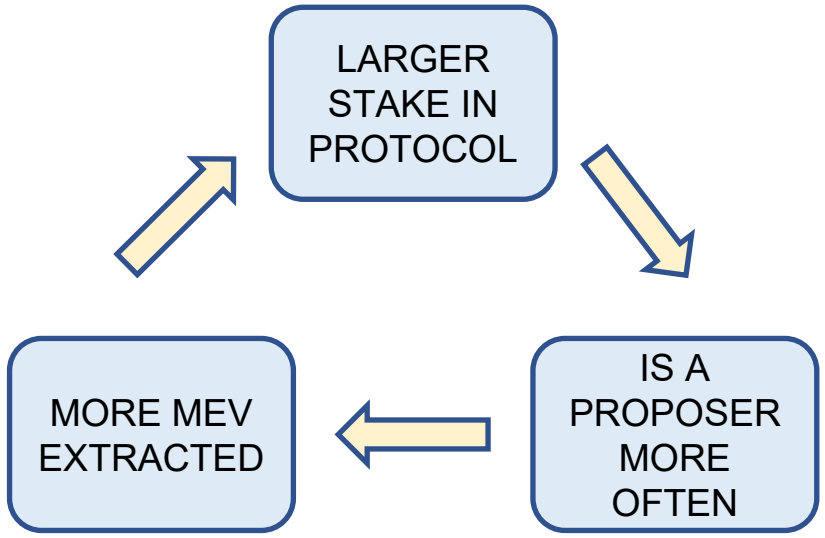
Optimizing MEV: a knapsack problem

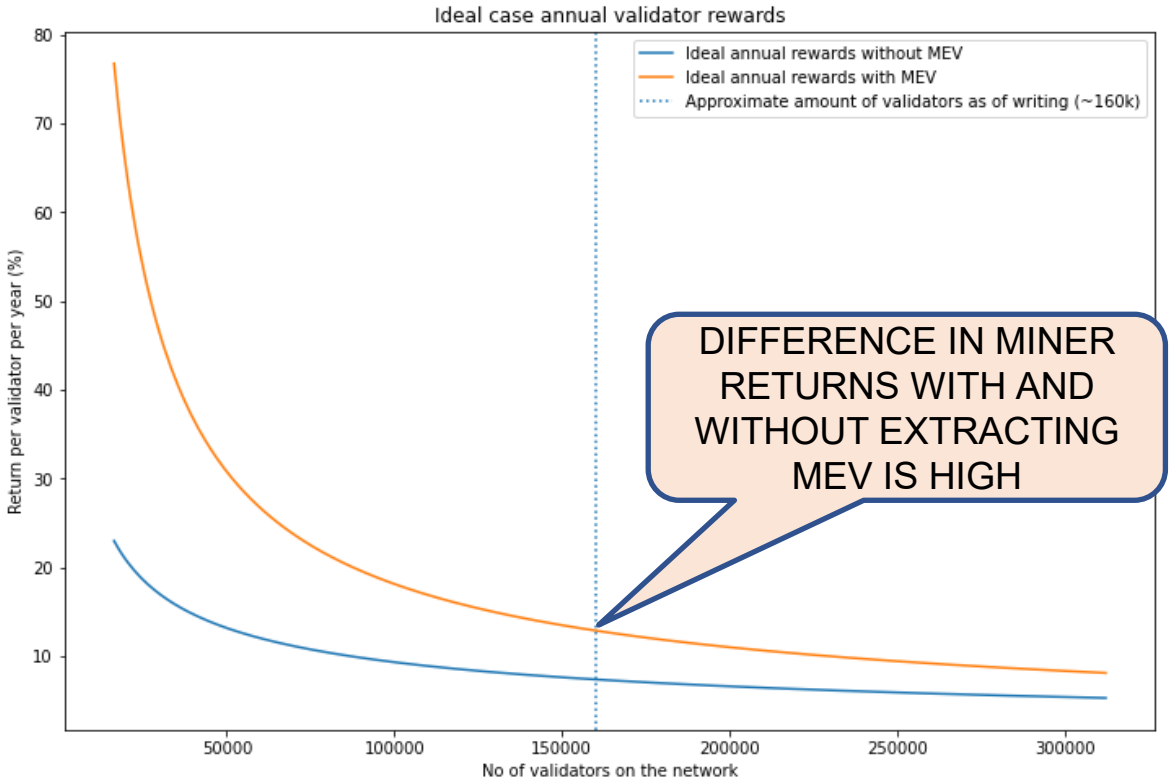
- Block space is limited, proposer's action space is potentially limitless
- e.g. Given a set of DEX swap transactions
  - Sandwich each of them?
  - Exploit arbitrage opportunities created after their execution?
  - Use JIT liquidity to extract their swap fees?
  - Some combination of the above?



MEV: a centralizing force

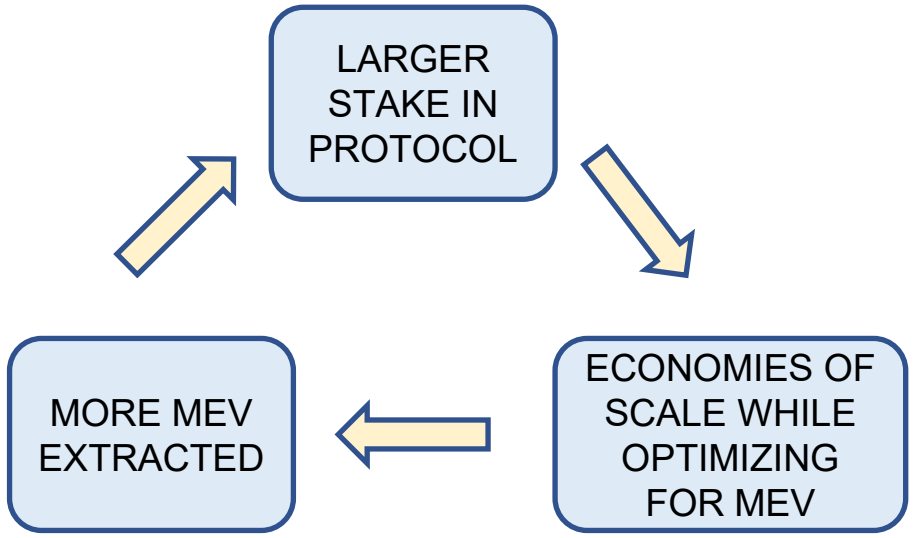
- After Ethereum became proof-of-stake, block rewards reduced substantially
- Validators incentivized to extract MEV

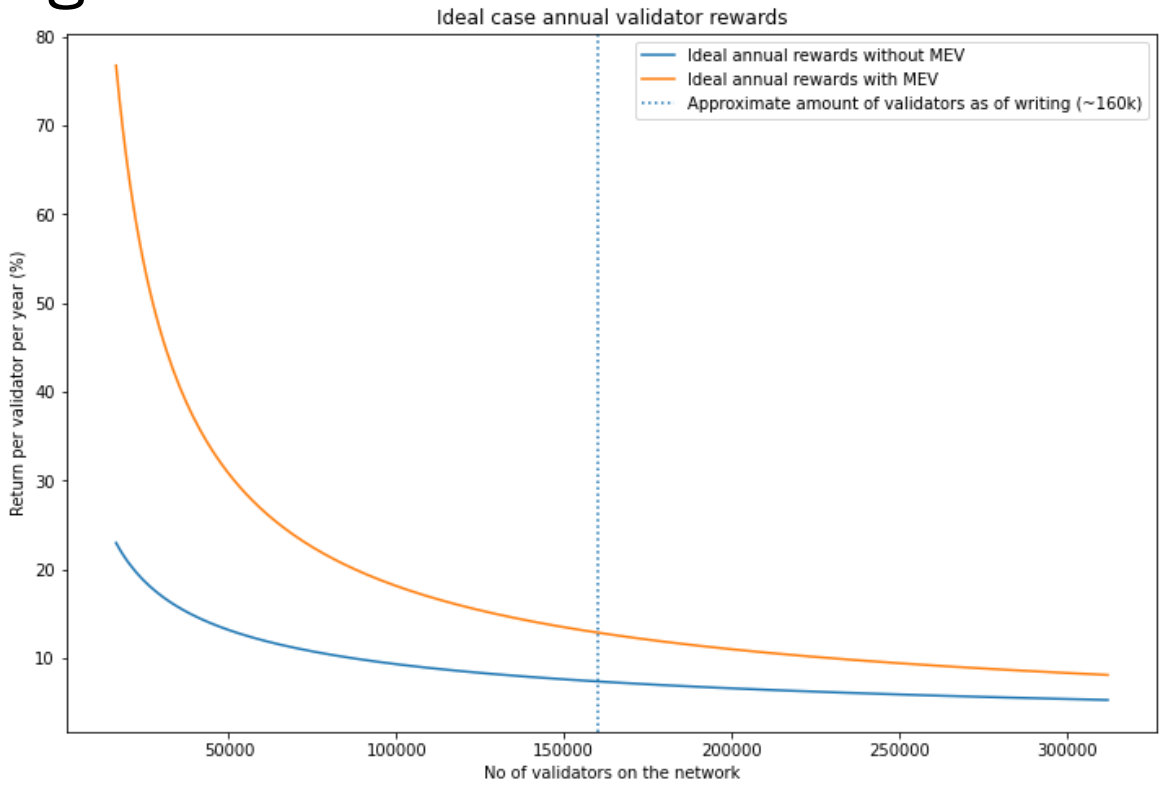




MEV: a centralizing force

- Also, because optimizing for MEV is hard, richer validators are able to dedicate more compute to it
- Larger validators keep getting larger - centralization





Need for protocol-level solution

- Batched-Auctions
  - All swaps in a block receive a common clearing price, prevents front running
  - But solution is application specific

- Do not address the dangers to consensus because of centralization

- As newer applications become decentralized, there would be other ways of extracting MEV from them, and from combinations of apps

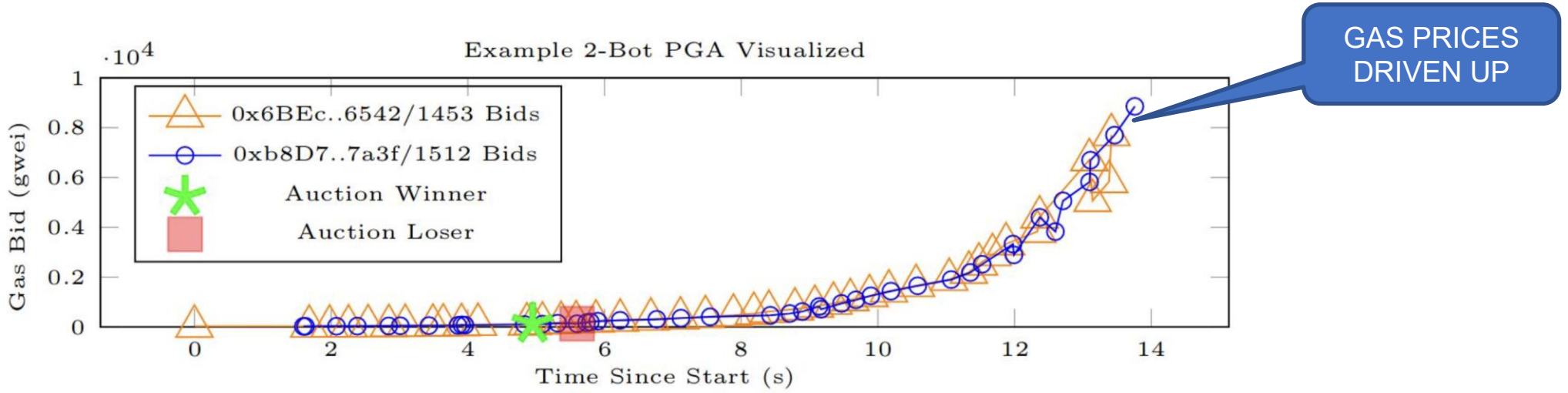
- Cross-chain MEV even more sophisticated – also a centralizing force

Need for protocol-level solution

- Solutions such as batching are application specific
- Do not address the dangers to consensus because of centralization
- As newer applications become decentralized, new ways of extracting MEV from those apps would appear
- Even more MEV may be extracted via combinations of apps on one chain and across chains
- Cross-chain MEV requires even more sophistication – is also a centralizing force

Design 0: Priority gas auctions

- Since position of txn in block matters, bots would bid for MEV opportunities – priority gas auctions (PGAs)
- e.g. Bots bidding on opportunity to do an arbitrage txn with a DEX



- Raises gas prices for others
- Unsuccessful MEV txns also included – wastes block space

Design 0: Priority gas auctions

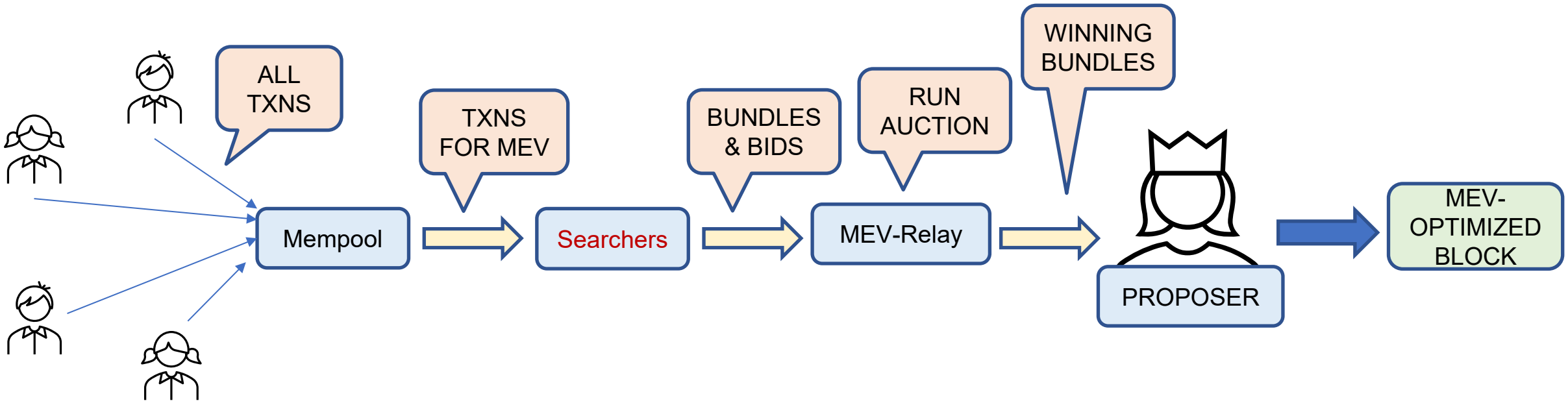
- Users seeking to backrun txns had to set gas just below the target txn's gas
- Led to spamming mempool with the same txn in a hope to get included after a large swap
- Every node affected because of the gossip protocol

- Main reason behind problem?
  - No way to bid for specific position in the block
  - This led to bidding wars for block space in general, causing worse gas fees for other users

Design 1: Flashbots auctions

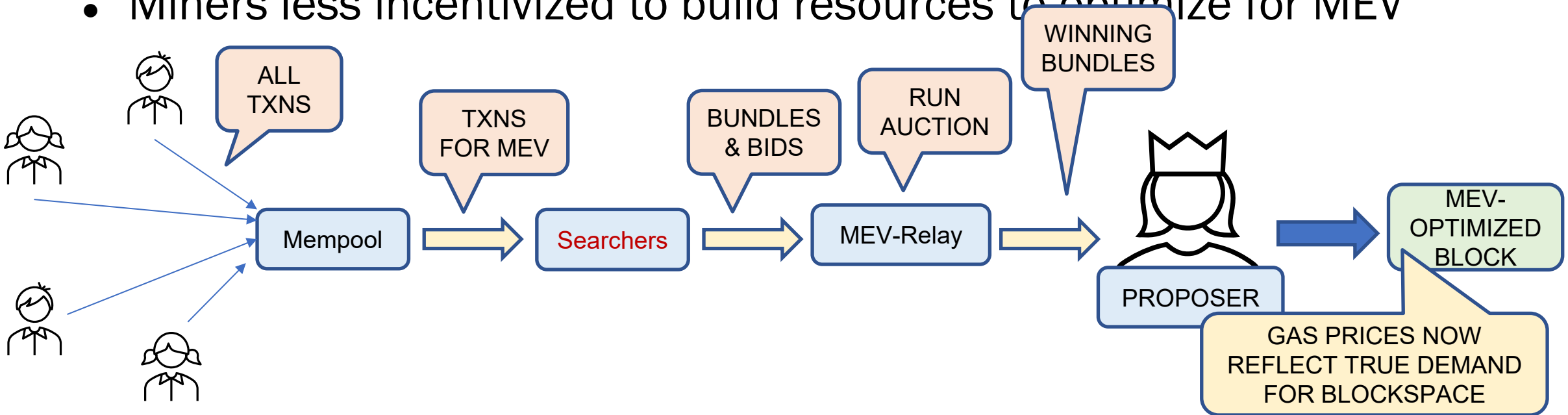
- New agent introduced: searchers
- They search for MEV opportunities, submit bundles of txns
- e.g. Searcher sees large buy trade on a DEX – bundle consists of sandwich (3 trades)

RELAY PROMISES TO KEEP BUNDLES PRIVATE UNTIL CONFIRMED ON-CHAIN



Design 1: Flashbots auctions

- Txns that lose bids do not appear in block – reduces wasted blockspace
- Separate auctions make it possible to bid on specific positions in block
- Miners less incentivized to build resources to optimize for MEV

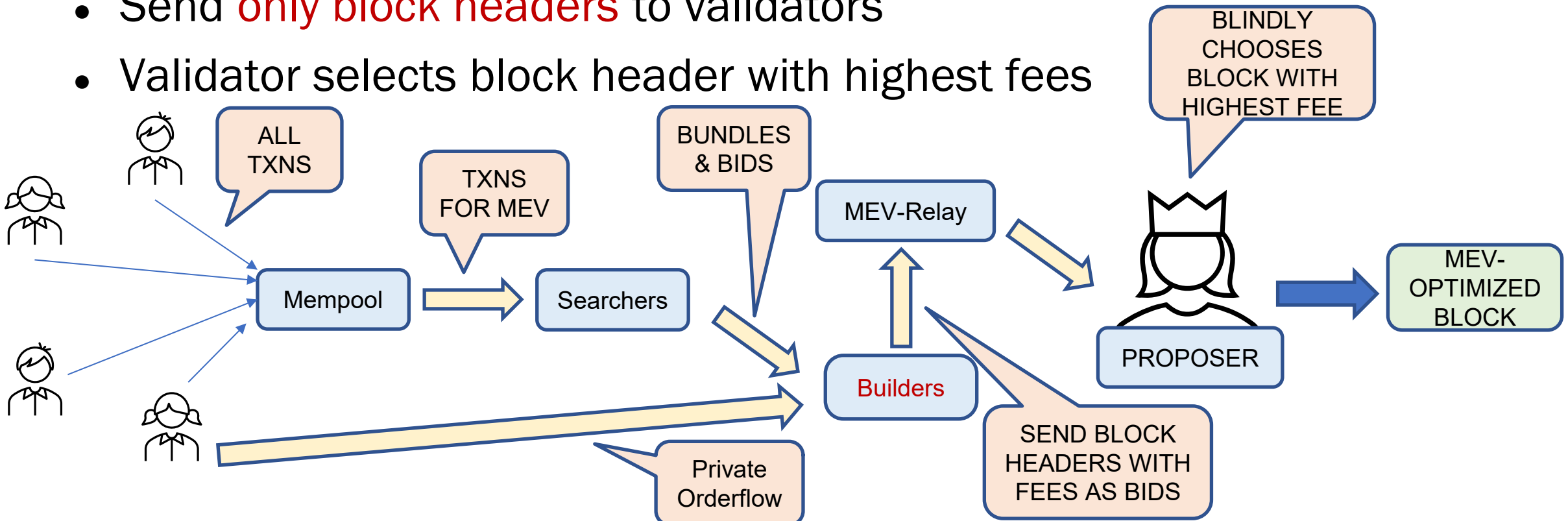


Flashbot Auctions: Problems

- Only a temporary solution – replaces the danger of centralization by another central entity (Flashbots)
- Relay operators and miners could misbehave by censoring bundles and then putting their own bundles which are copies of the censored bundles
- Relay operators do not misbehave only because that will cause loss of social reputation for Flashbots
- Miners do not misbehave because Flashbots would then stop sending them bundles
- Need to decentralize the Flashbots auction system

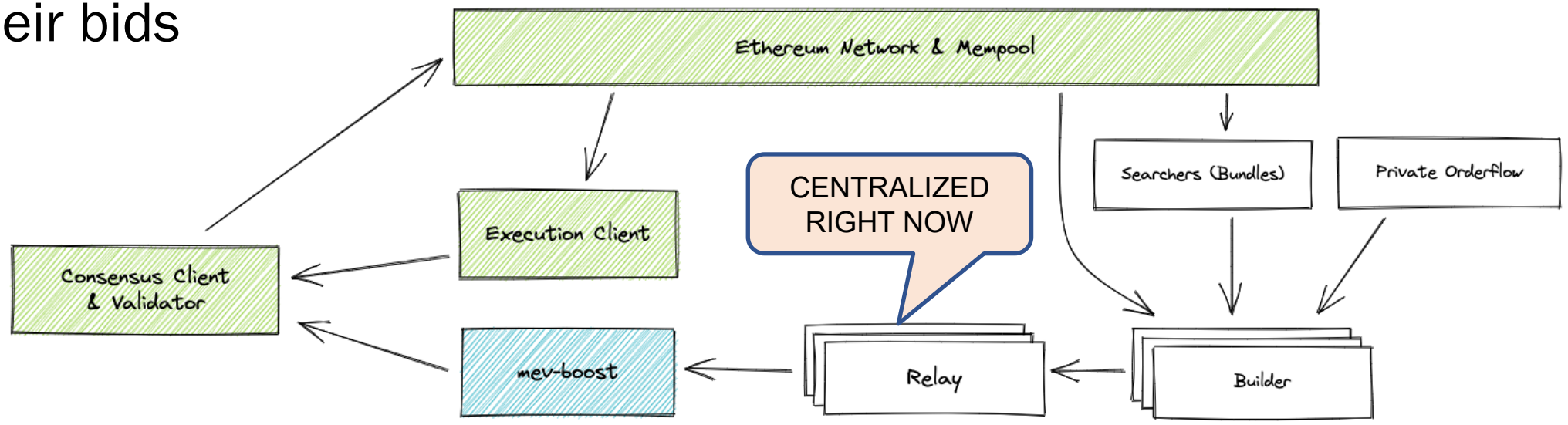
Design 2: Proposer-Builder separation

- New agent introduced: builders
- Builders collect bundles and solve the knapsack problem for MEV
- Send only block headers to validators
- Validator selects block header with highest fees



Design 2: Proposer-Builder separation

- Validators no longer need to optimize for anything except total fees – reduces centralization
- To be implemented as part of Ethereum protocol – “the Splurge”
- Currently implemented as mev-boost client by Flashbots
- Validators use client to connect to relays where block builders send their bids



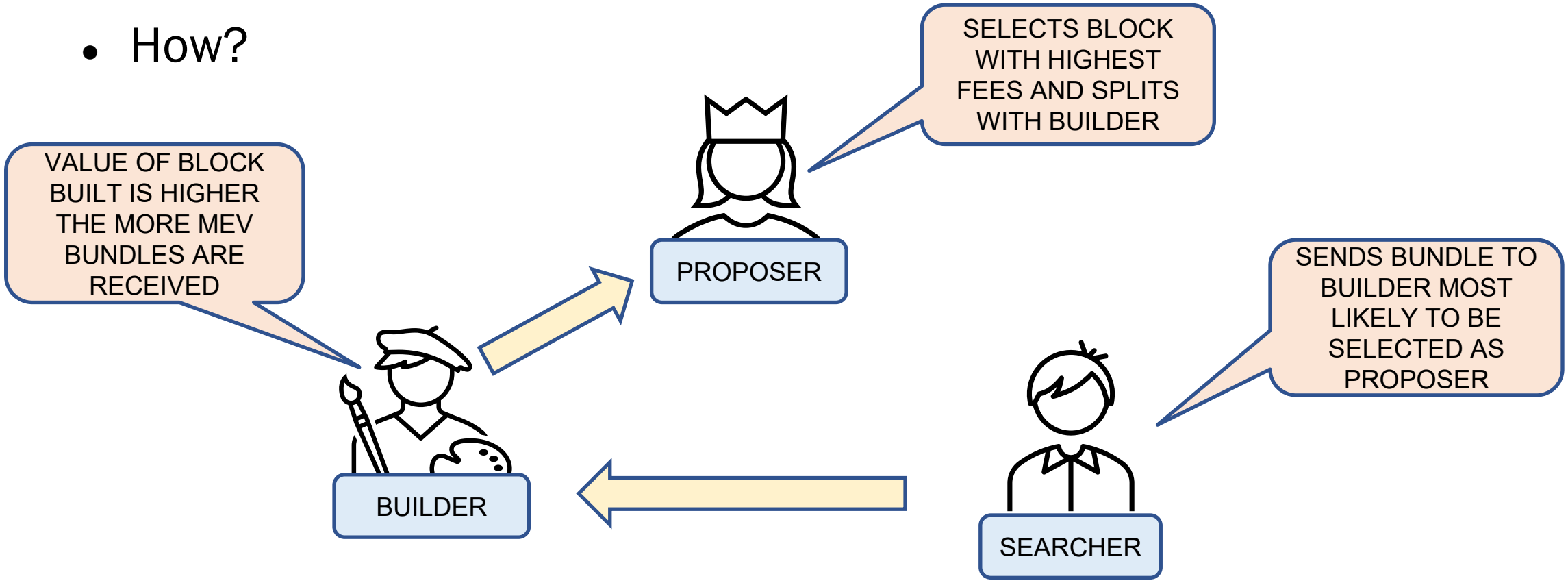
Design 2: Proposer-Builder separation

Incentives of the players:

- Validator – Wants to choose block with highest fees. No compute needed – no centralization pressure
- Builder – Gets payout from MEV only if block is chosen by validator (i.e. when fees included are highest)
  - Interested in collecting as much *exclusive* order flow as possible
  - Builders can get exclusive flow by promising no frontrunning – Flashbots Protect
- Searcher – Wants to choose builder whose block is most likely to be chosen

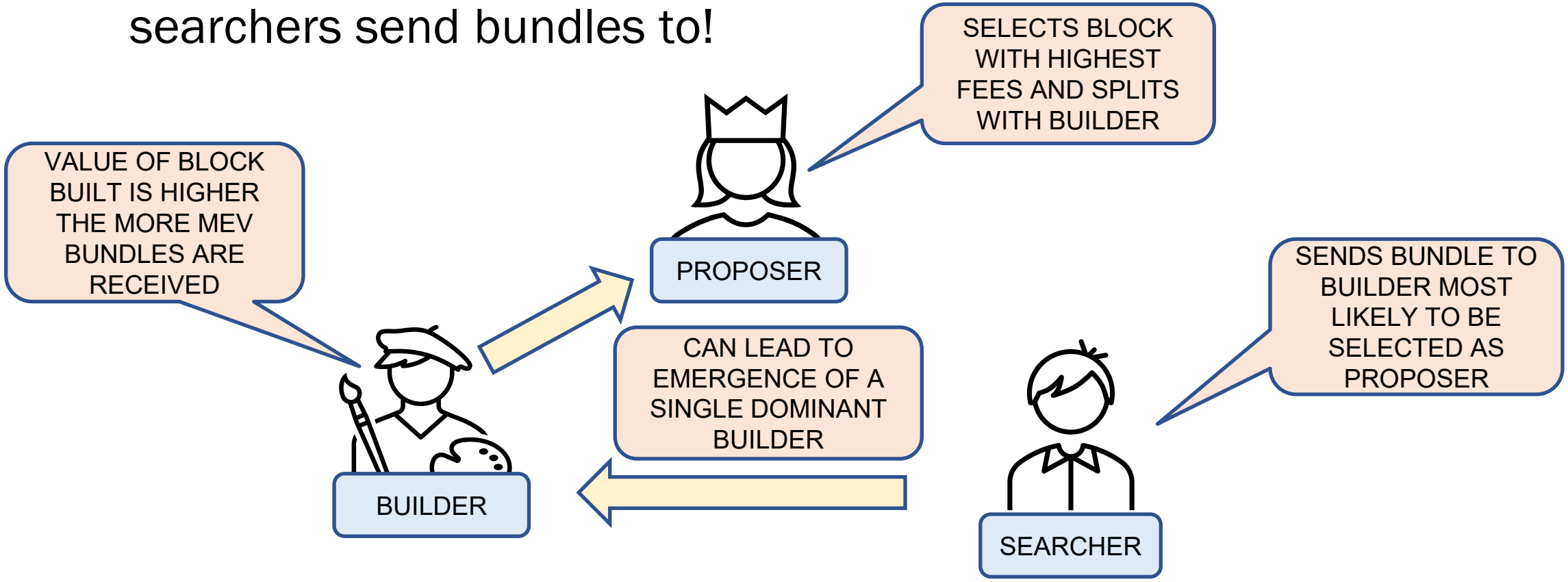
Proposer-Builder separation: Problems

- Builder – Searcher dynamic creates a centralizing force
- How?



Proposer-Builder separation: Problems

- Builder most likely to be selected – is also the builder most searchers send bundles to!



Proposer-Builder separation: Problems

- Once there are dominant builders who are receiving exclusive order flow, they need only bid just above other smaller builders
- Thus, most of MEV value stays with them and not much goes to validator – add to centralizing pressure

- This is indeed what has happened – Flashbots builder is dominant, and trusted
- But centralization is successfully isolated away from validation

Design 3: Decentralized Builders

- Released in Nov 2022 by Flashbots
  - Called SUAVE (**Single Unifying Auction for Value Expression**)
- Key idea : Use a separate blockchain for builders to build blocks in a decentralized way

- Users express preferences on the SUAVE chain – through bundles
- Executors find best execution for those preferences
- Builders collectively make a block to maximize MEV

- Detailed implementation not public yet

Open problems

- Rewards for incentivizing agents on SUAVE?
- MEV oracles? – estimate MEV extracted in block
- MEV redistribution? – giving parts of extracted MEV back to users
- Multi-block MEV? – Turns out MEV from consecutive blocks much more than sum of individual block MEV, is it possible to prevent proposers getting elected for many blocks in a row?

LECTURE ENDS