EC'24 Tutorial on Transaction Fee Mechanism Design

Organizers:
Hao Chung, Matheus V. X. Ferreira, Yotam Gafni, and Aviv Yaish

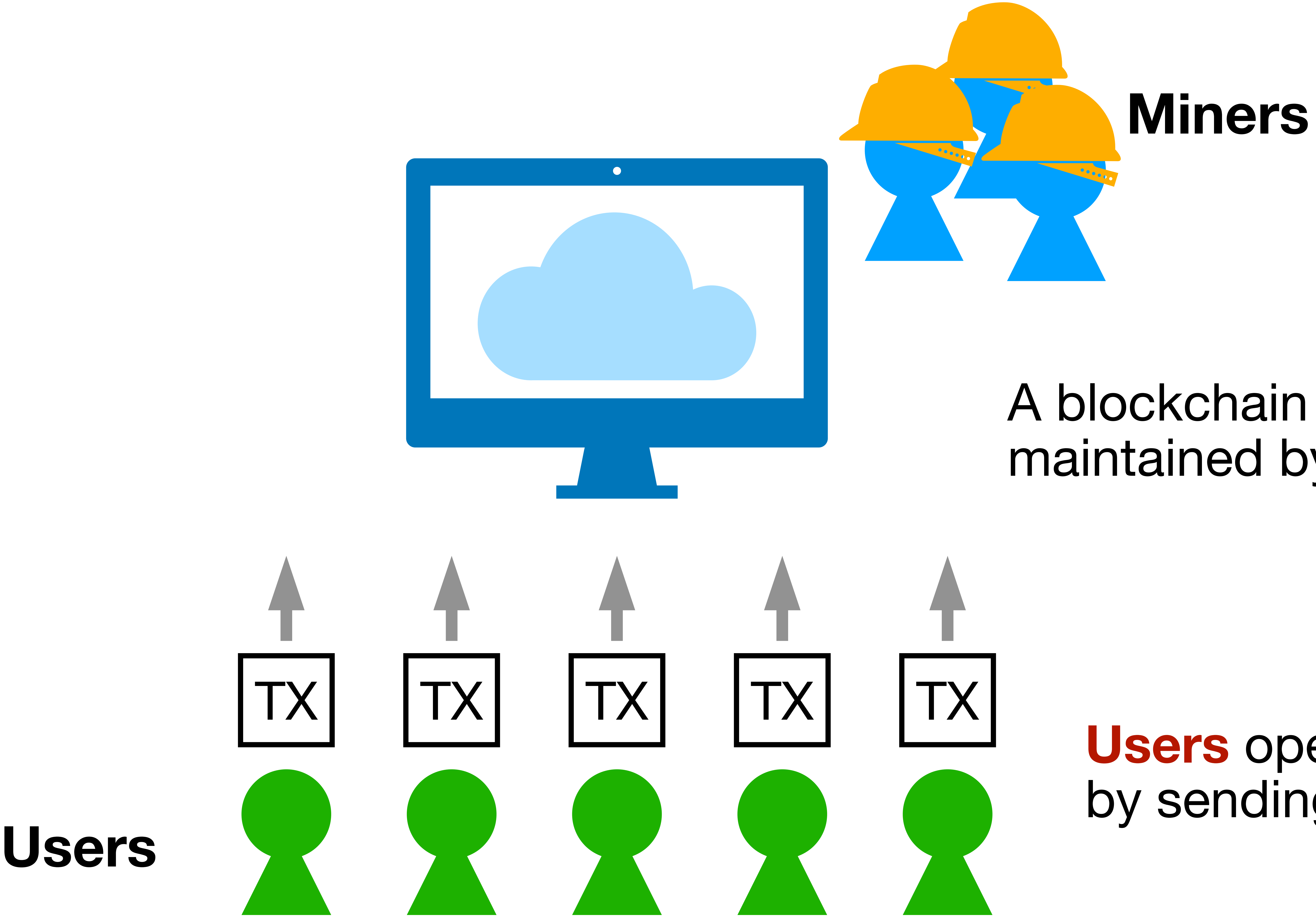
Agenda

- Lecture 1 (20 mins): TFMs for a single block
- Lecture 2 (20 mins): Dynamics TFMs
    - Break (30 mins)
- Lecture 3 (20 mins): Extensions to the TFM frameworks
- Panel discussion (30 mins):
    - Mallesh M. Pai (Rice University and Consensys)
    - Tim Roughgarden (Columbia University and a16z crypto)
    - Noam Nisan (Hebrew University of Jerusalem and Starkware)

Lecture 1: TFMs for a Single Block

- What are blockchains and TFMs?
- TFM’s desiderata
- Limitations in the single-block setting
- What can cryptography do for TFM design?

Blockchain is a public computer



A blockchain is a public computer maintained by multiple **miners**

Users operate the computer by sending **transactions**

A transaction can be as simple as coin transfer

Block: 20120974 1 Block Confirmation

Timestamp: 17 secs ago (Jun-18-2024 08:17:47 PM +UTC) | Confirmed within 30 secs

Transaction Action: Transfer 4.19705488 ETH To 0x0f967c884545d1b295aEf0281eE49688CA7255a4

Sponsored:

From: 0xFd90a4bF5892dA15F863e8C385A789e583F2117D

To: 0x0f967c884545d1b295aEf0281eE49688CA7255a4

A transaction can also be a complex program

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;
/// @title Voting with delegation.
contract Ballot {
    // This declares a new complex type which will
    // be used for variables later.
    // It will represent a single voter.
    struct Voter {
        uint weight; // weight is accumulated by delegation
        bool voted;  // if true, that person already voted
        address delegate; // person delegated to
        uint vote;   // index of the voted proposal
    }

    // This is a type for a single proposal.
    struct Proposal {
        bytes32 name;   // short name (up to 32 bytes)
        uint voteCount; // number of accumulated votes
    }
```

A transaction can also be a complex program

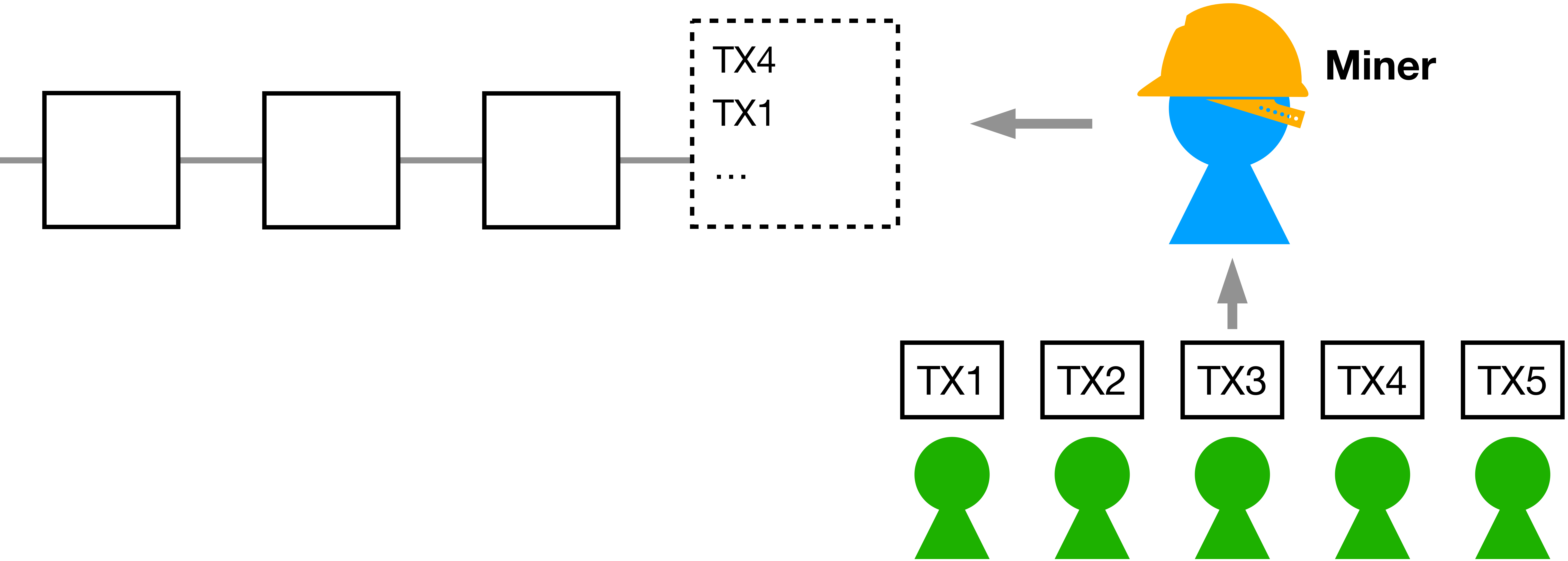
```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.7.0 <0.9.0;
/// @title Voting with delegation.
contract Ballot {
    // This declares a new complex type which will
    // be used for variables later.
    // It will represent a single voter.
    struct Voter {
        uint weight; // weight is accumulated by delegation
        bool voted;  // if true, that person already voted
        address delegate; // person delegated to
        uint vote;   // index of the voted proposal
    }

    // This is a type for a single proposal.
    struct Proposal {
        bytes32 name;   // short name (up to 32 bytes)
        uint voteCount; // number of accumulated votes
    }
```

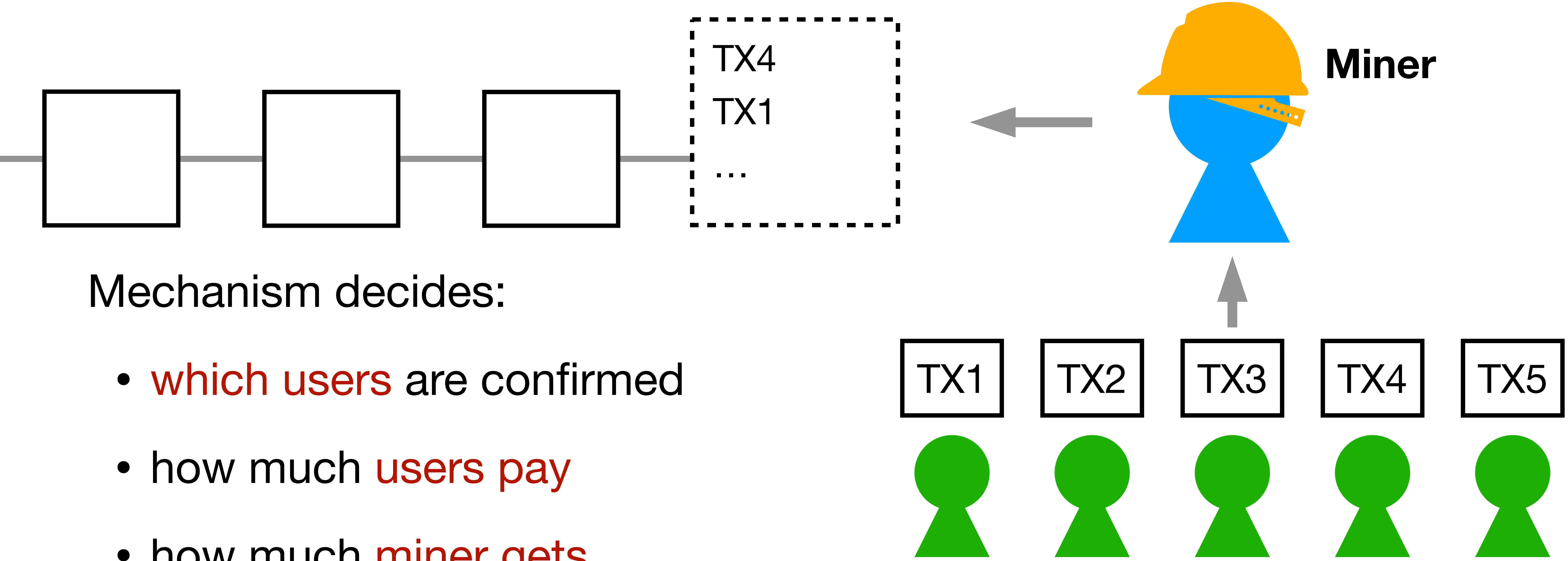
programs on blockchains are known as “smart contracts”

This computer is updated block-by-block

The miner packs a bunch of transactions into a **block**



Transaction fee mechanism is like an auction



Mechanism decides:

- which users are confirmed
- how much users pay
- how much miner gets

What counts as good TFMs?

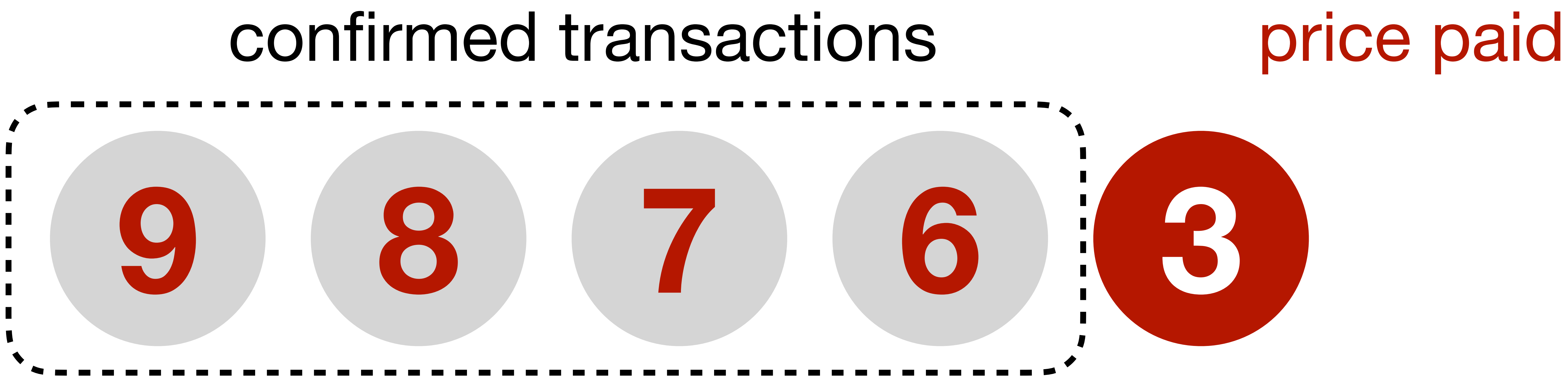
Bitcoin: first price auction

1. Top k bids are confirmed
2. Pay your own bid
3. All payment goes to the miner

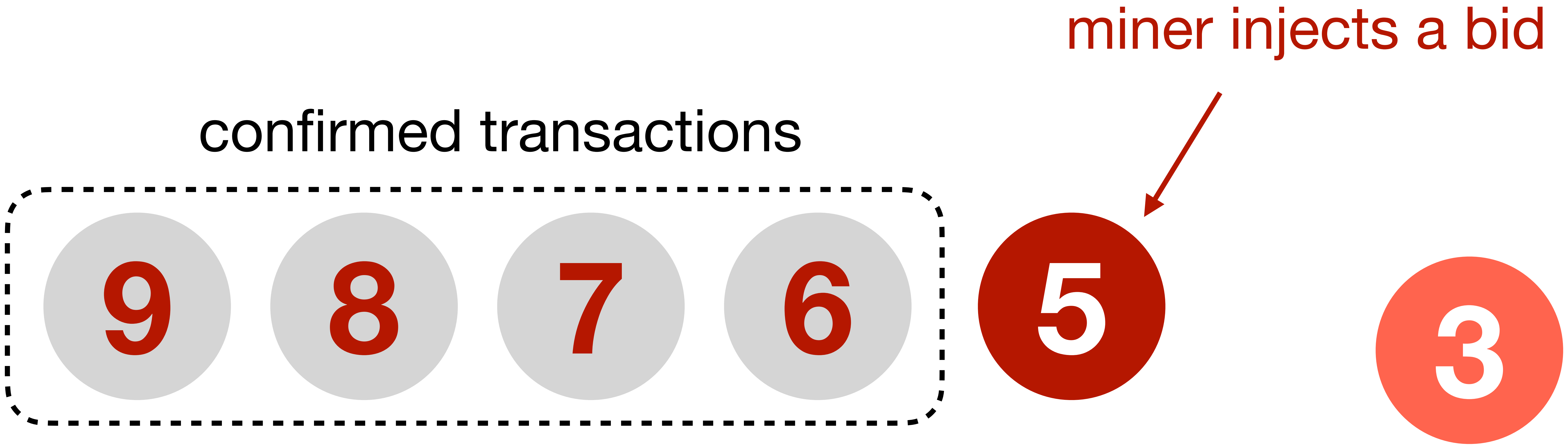
Encourage untruthful bidding

Truthful bidding by classical mechanisms

We can run 2nd price auction



Classical Mechanisms Fail!



Three desired properties

UIC (user incentive compatibility)

- A user's best strategy is to bid truthfully

MIC (miner incentive compatibility)

- Miner's best strategy is to implement the mechanism honestly

c-SCP (c-side-contract-proofness)

- A coalition of the miner and at most c users doesn't want to deviate

New challenges in decentralized context!

Three desired properties

UIC (user incentive compatibility)

- A user's best strategy is to bid truthfully

MIC (miner incentive compatibility)

- Miner's best strategy is to implement the mechanism honestly

c-SCP (c-side-contract-proofness)

- A coalition of the miner and at most c users doesn't want to deviate

New challenges in decentralized context!

!

2nd price auction is UIC, but not MIC and 1-SCP

1st price auction is MIC and c-SCP, but not UIC

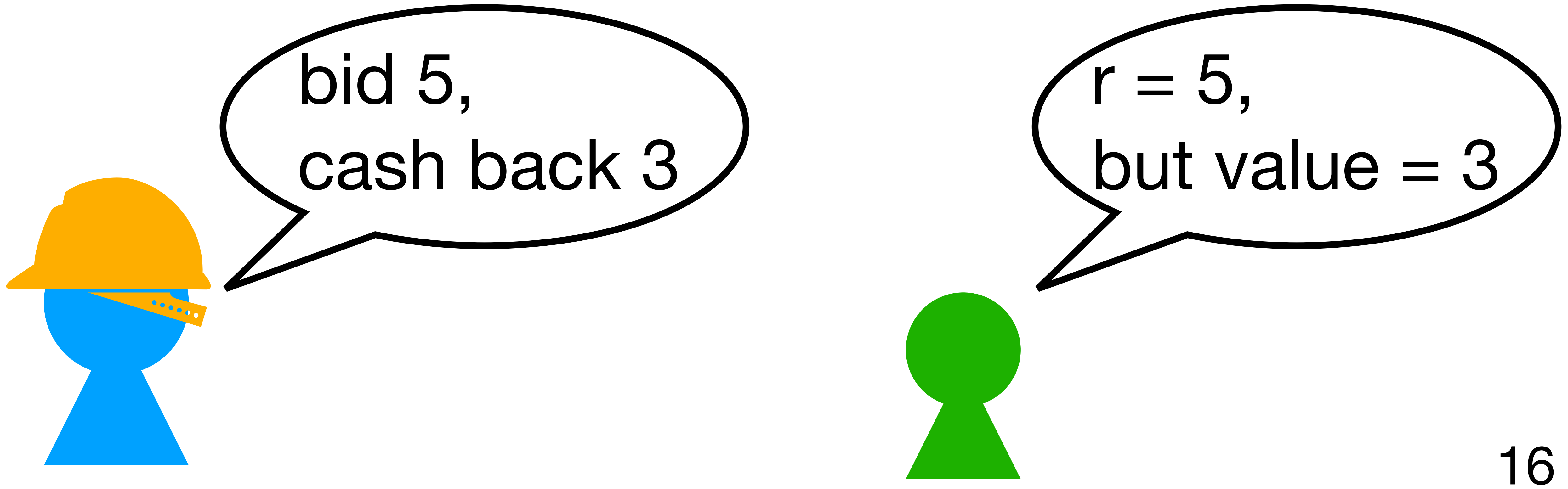
Ethereum's EIP-1559 achieves all properties assuming infinite block size

Uncongested ⟹ posted-price auction

- All bids $\geq$ posted price r are confirmed, and pay r
- miner gets nothing; all payment is burnt

Congested ⟹ first price auction

Without burning, miner-user coalition can bypass the price r



Dream mechanism is impossible!

Theorem

Suppose the block size is finite.

No non-trivial TFM can satisfy UIC and 1-SCP at the same time.

Foundations of Transaction Fee Mechanism Design. Hao Chung, and Elaine Shi. SODA'23

Zero miner revenue is inherent

Theorem

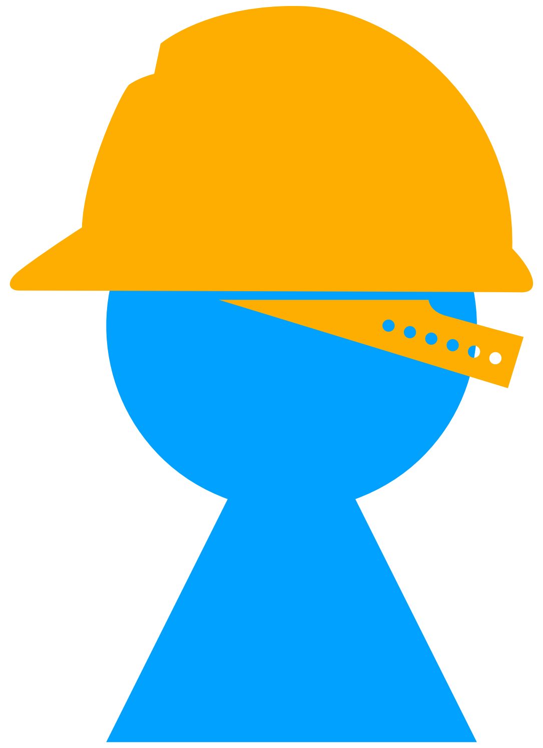
For any TFM that satisfies UIC and 1-SCP,

miner revenue must be zero.

Burning in EIP1559 is necessary!

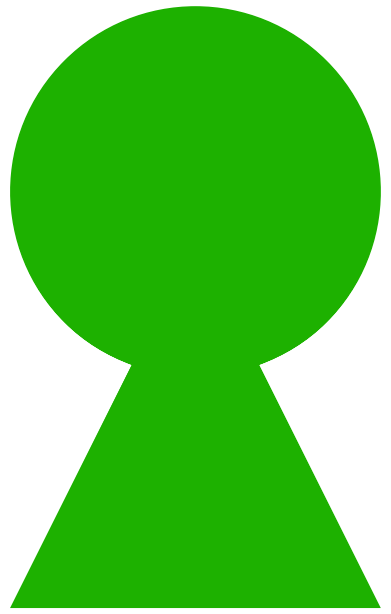
Foundations of Transaction Fee Mechanism Design. Hao Chung and Elaine Shi. SODA'23

Strategy Space in Plain Model



After seeing others’ bids, a **miner** can

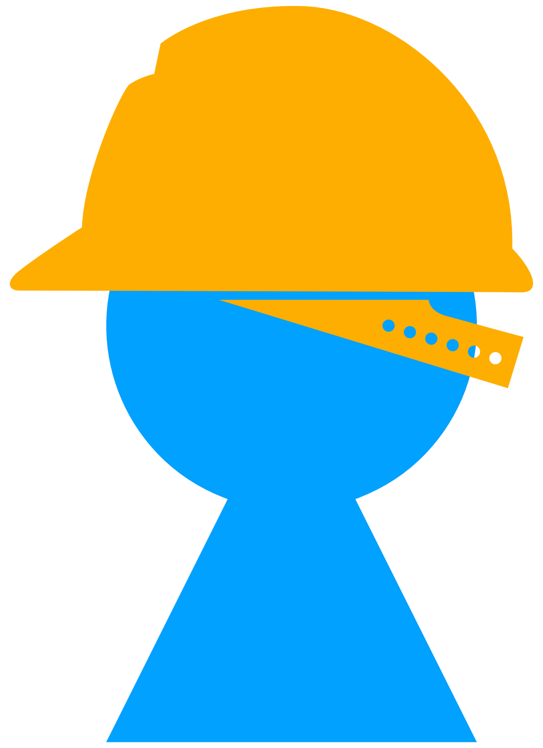
- inject fake bids
- create a block arbitrarily



After seeing others’ bids, a **user** can

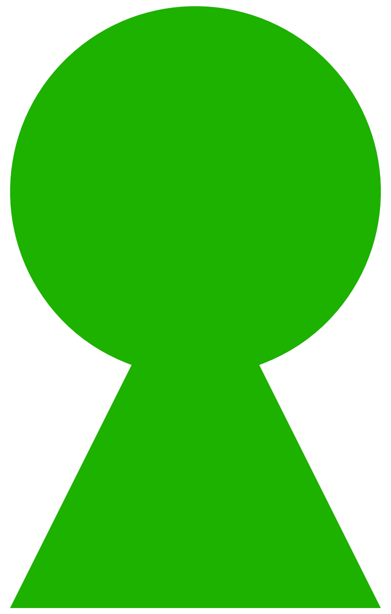
- bid untruthfully
- inject fake bids

Strategy Space in MPC-assisted Model



~~After seeing others' bids,~~ a **miner** can

- inject fake bids
- ~~create a block arbitrarily~~



~~After seeing others' bids,~~ a **user** can

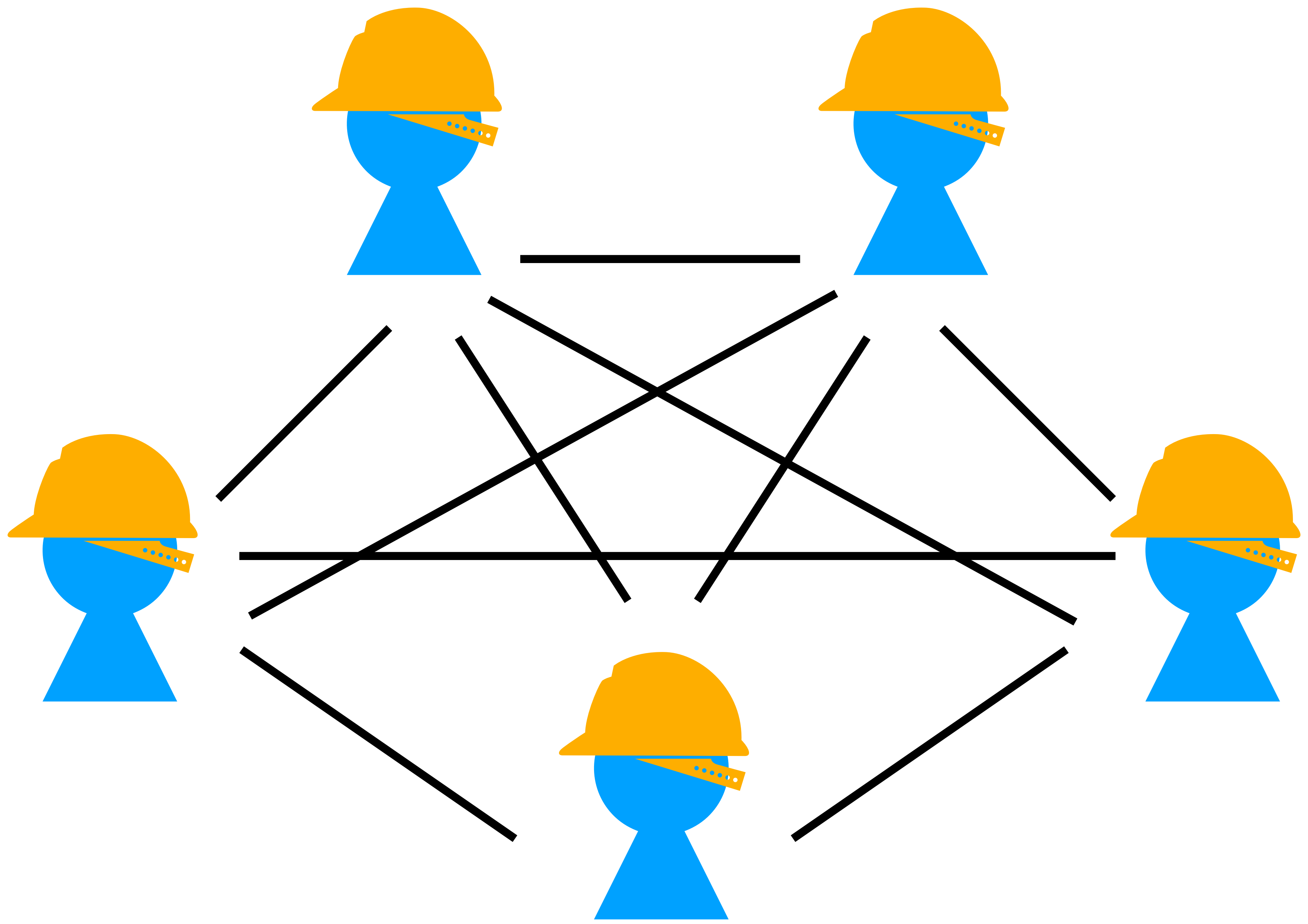
- bid untruthfully
- inject fake bids

Posted-price with random selection

- All bids $\geq$ **posted-price** r are **eligible**
- Randomly choose k eligible bids to confirm
- Each confirmed bid pays r
- All payments are **burnt**

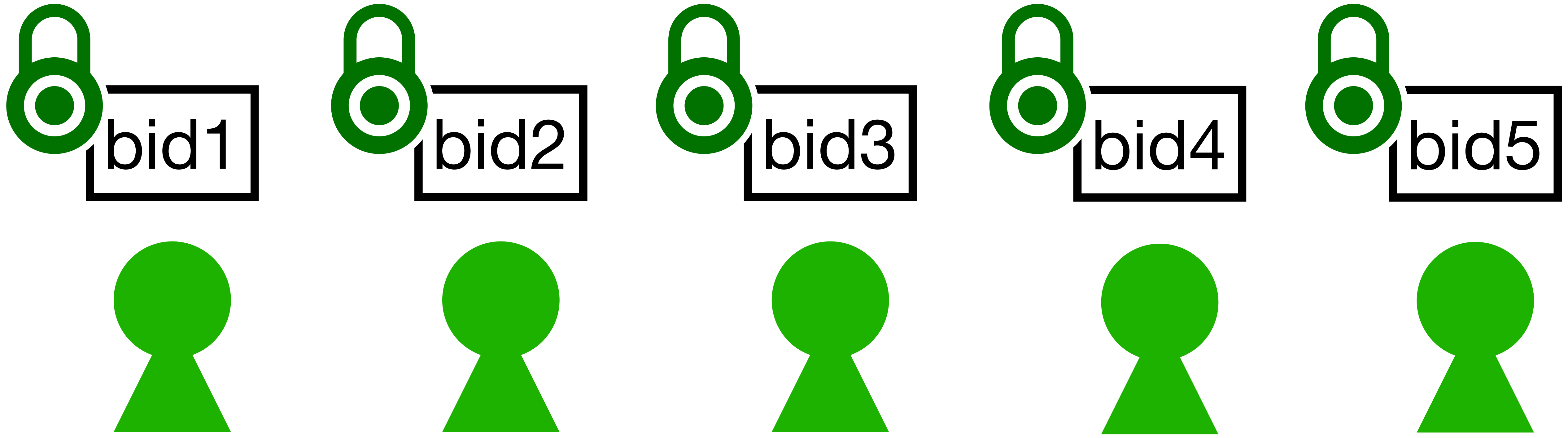
This mechanism is **UIC + MIC + 1-SCP**

MPC-Assisted Model



Multiple miners jointly run **multiparty computation (MPC)**

A user secret-shares its bid,
and sends each share to each miner



What Can Cryptography Do For Decentralized Mechanism Design. Elaine Shi, Hao Chung, Ke Wu. ITCS'23

Take away from lecture 1

- A blockchain is a public computer
- Transaction fee mechanisms (TFMs) allocate block space
- New design feature: burning
- New challenge: miner and miner-user deviation
- In plain model: UIC + 1-SCP $\Longrightarrow$ trivial mechanism

Some simplifications in lecture 1

- Focus on a **single** block
  - In practice: multiple block in the long term (lecture 2)
- All transactions have **equal size**
  - In practice: different size (“gas” model in Ethereum)
- Transaction **order** does not matter in the block
  - In practice: order matters! (lecture 3)
- A single miner fully controls one block
  - Depend on protocols, e.g. MPC-assisted mechanism (end of this lecture) or proposer-builder separation (lecture 3)

Thank you!

backup: Ethereum’s EIP-1559

- Each bid specifies $(\overset{\text{base fee}}{r}, \overset{\text{tip}}{t})$
- All bids $\geq$ **base-fee** r are **eligible**
- Miner confirms up to k eligible bids with highest tips
- Each confirmed bid pays $r + t$
- Miner gets all the tips, the base fee is **burnt**