CSE 3550/5000: Blockchain Technology

Lecture 11
Blockchain Scalability

Ghada Almashaqbeh
UConn - Spring 2026

Outline

- Motivation.
- Layer 1 scalability solutions.
  - Parameter change.
  - Sharding.
- Layer 2:
  - Probabilistic micropayments.
  - Payment channels.
  - Rollups.

Scalability of Blockchains

- A major issue hindering many applications and leading to high transaction fees.
  - Bitcoin has an average throughput of 5-7 transactions/sec, while Etehreum supports around 12 - 15 transactions/sec.
  - Visa supports around 65,000 transactions/sec!!!
- Also the size of the blockchain is growing at massive rate, leading to high storage, and information lookup, overhead.

Solutions

- Scalability is a very active research area.
- Several solutions exist that can be categorized into:
  - **Layer 1 solutions:** target the consensus layer (most of the time leads to a hard fork).
    - Changing consensus parameters.
    - Sharding.
  - **Layer 2 solutions:** process transactions off-chain and (usually) log only state changes on-chain.
    - Probabilistic micropayments.
    - Payment channels.
    - Rollups.

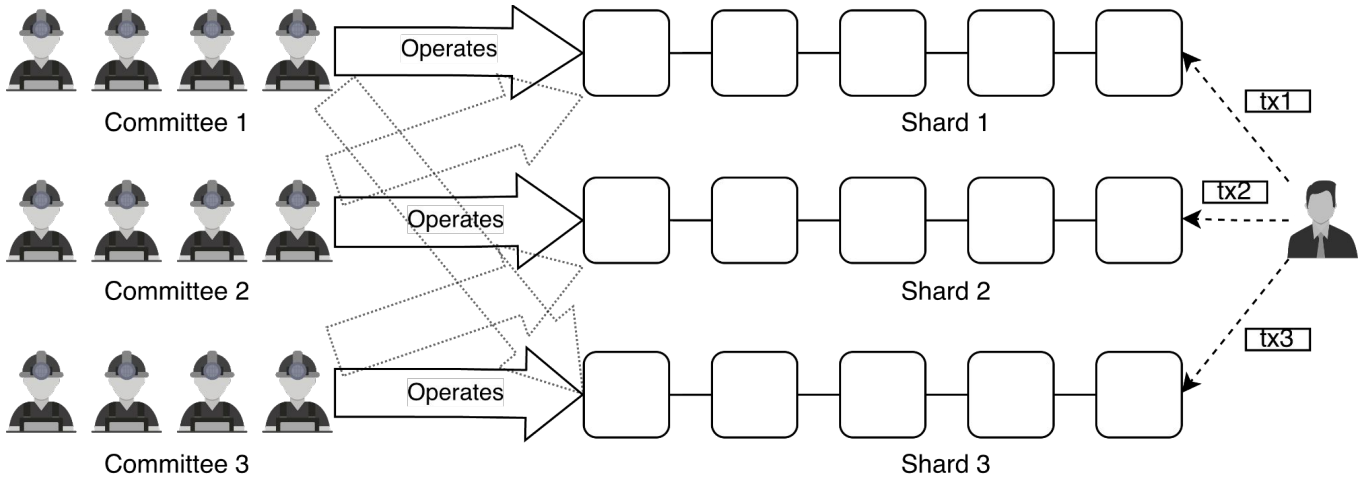
Layer 1 Solutions

Parameter Change

- This includes increasing the block size, reducing the round (or block) time.
  - Bitcoin Cash was a hard fork of Bitcoin that increased block size from 1 MB to up to 32 MB (started with 8 MB).
- Another example: Seggregated Witness (SegWit) in Bitcoin
  - Soft fork that was implemented in 2017.
  - It separates the signature, i.e., witness, from the transaction body.
    - Only the transaction body is counted in the block size.
  - So the signature is no longer part of the transaction ID.
    - Recall that a transaction ID is the hash of the transaction.
  - In theory, this will increase the effective block size to 4 MB (so still Bitcoin block size is 1 MB but it is only counted for transaction body), and hence, will increase the transaction throughput.
    - However, in practice Bitcoin throughput did not improve much!

Sharding

Figure credit: M. Najd



- Aims to scale the blockchain with the miner population.
- Multiple blockchains (or shards) work in parallel, each is managed by a committee of the miners—so split the miners among the shards.
- Shards usually employ BFT-based consensus with rotating committees.
- Incoming traffic usually assigned at random to the shards (to split the load uniformly or in balanced way)--usually hash of tx modulus no. of shards.
  - A huge problem; random assignment results in large number of expensive cross-shard transactions.
  - Lead to several approaches to achieve transaction locality.

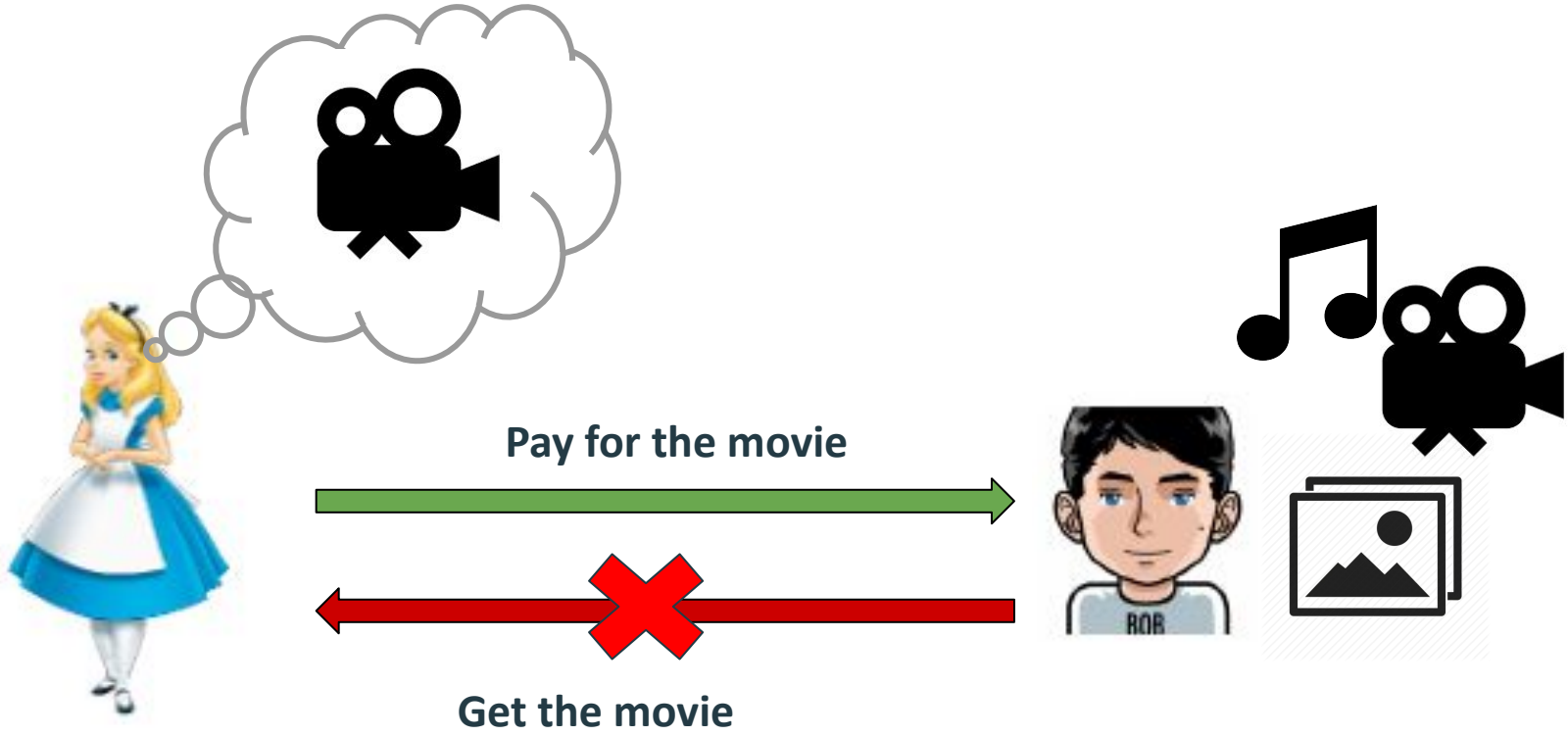
Layer 2 Solutions

Micropayments

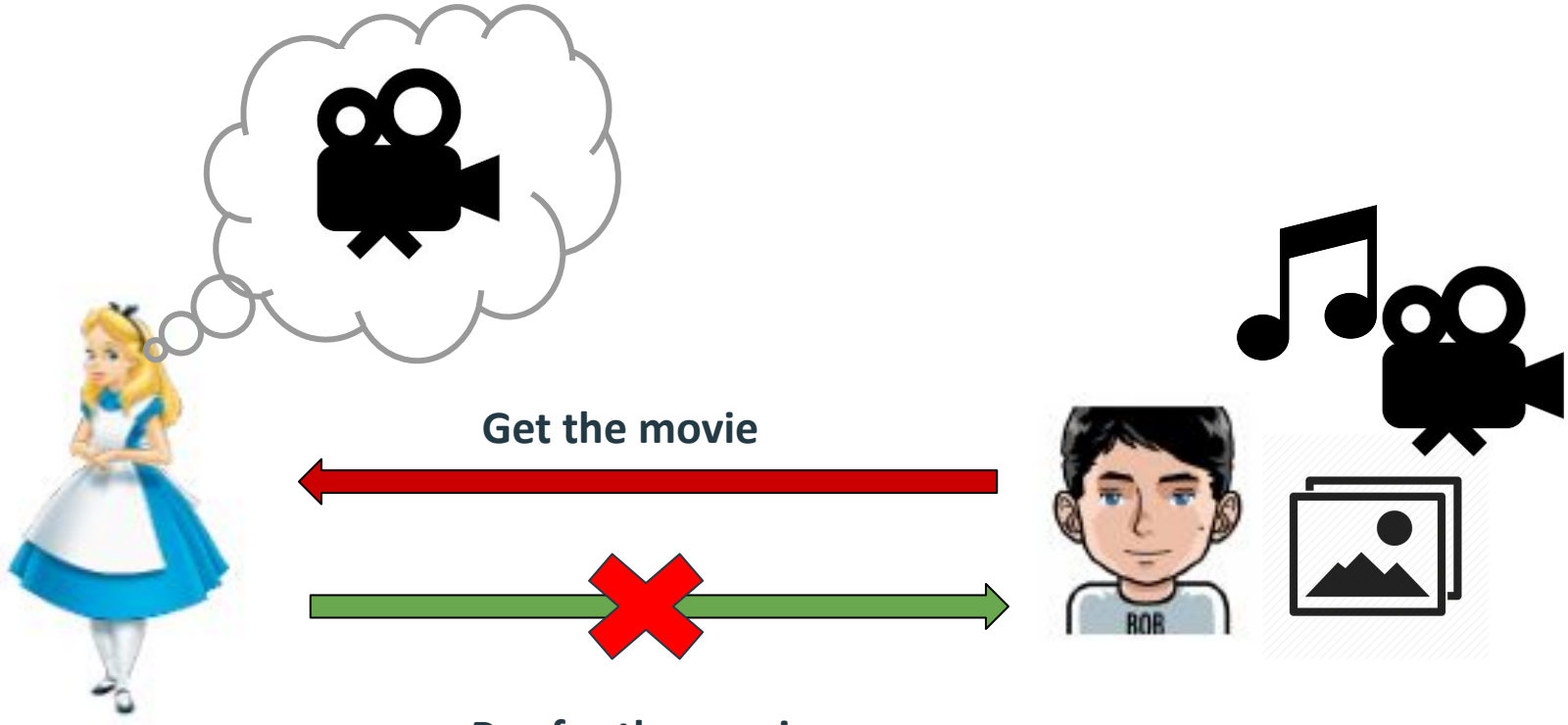
Why micropayments are useful?

Customer

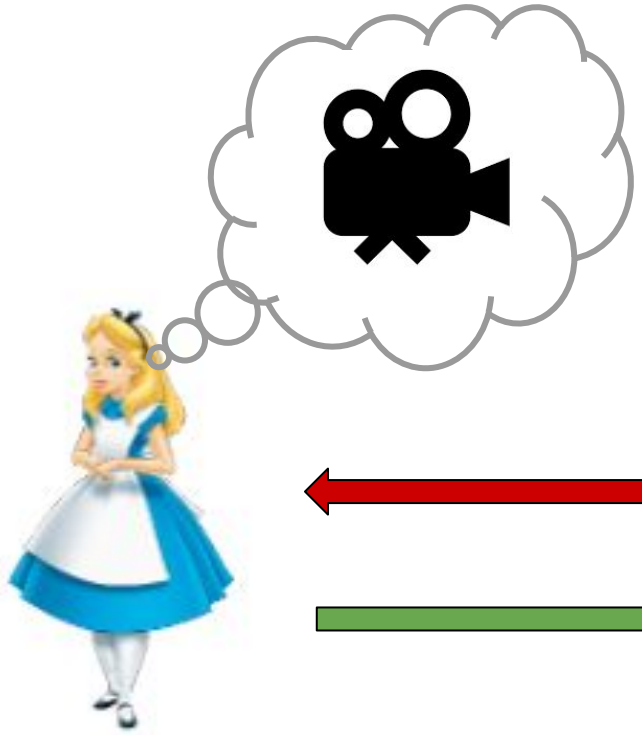
Merchant

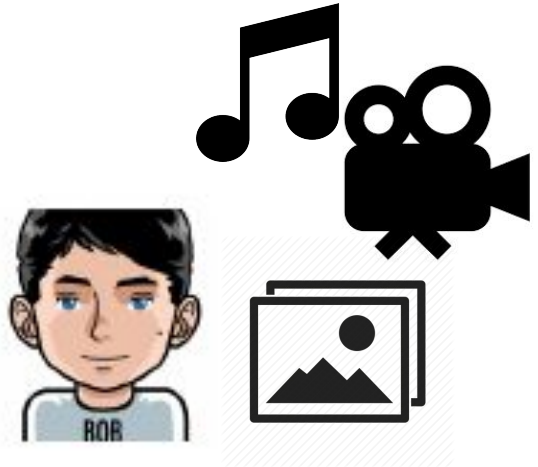


The merchant does not provide the service but keep the customer's money



The customer does not pay after obtaining the service



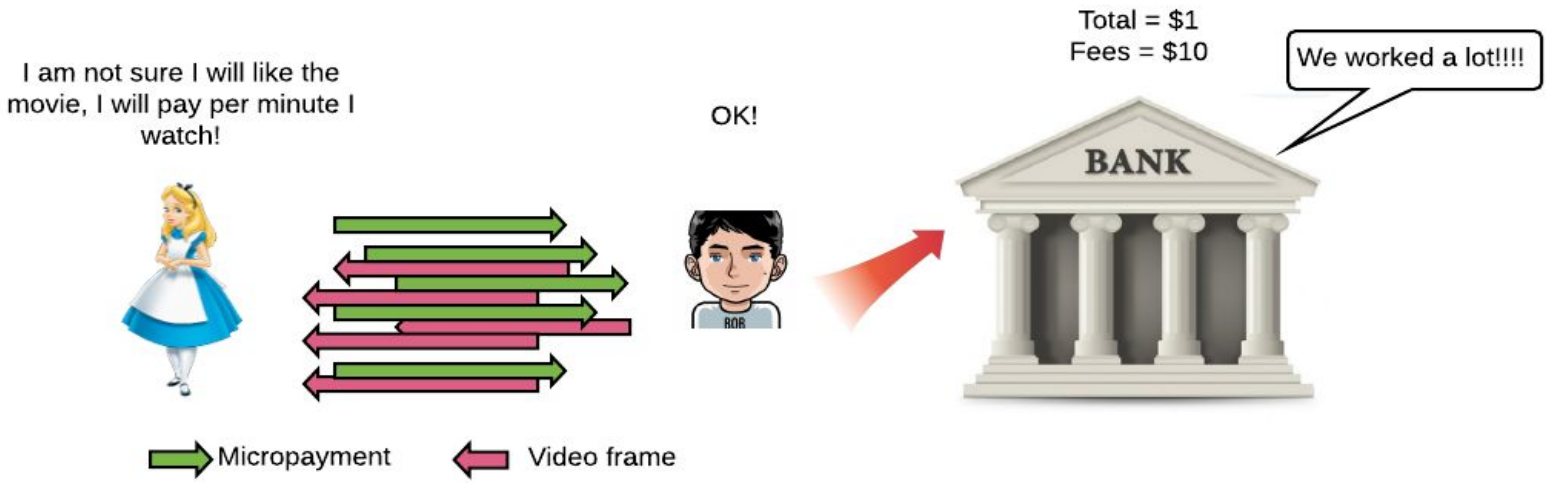


Customer

Merchant

I did not like this movie, I just watched the first 30 min but I have already paid in full!

Micropayments



- Payments of a micro value, i.e., pennies or fractions of pennies.
- Several applications, e.g., ad-free web, online gaming, etc.
  - Used extensively in blockchain-based distributed services.
  - Main motivation is a workaround the impossibility of fair service-payment exchange. ***Do you see how?***
  - Another motivation is flexibility; customers can stop the service at anytime.

Challenges

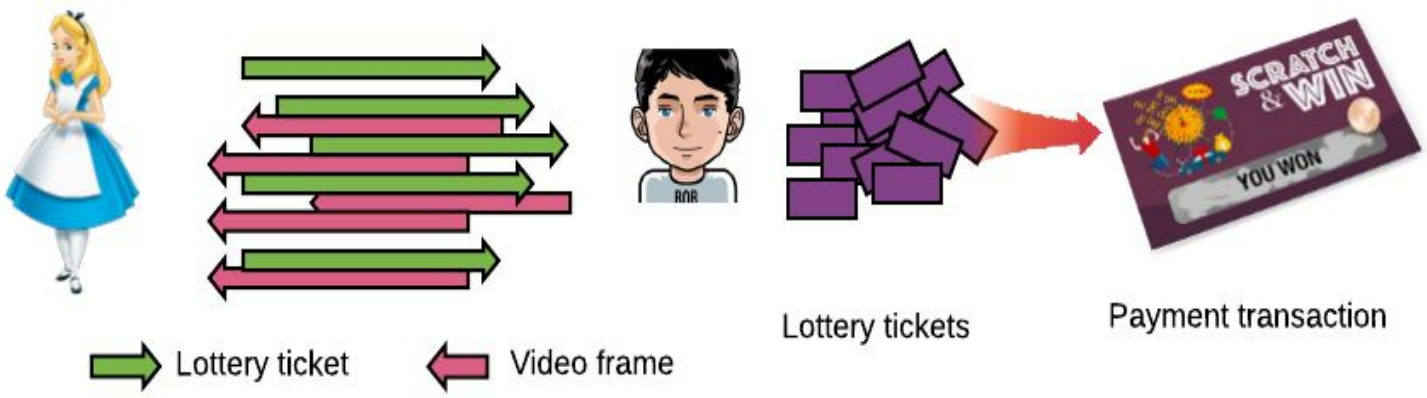
- Produce a huge number of small-value transactions.
  - Overwhelm the system.
  - Explode the payment log, i.e., blockchain storage footprint.
  - Do not scale for large demands or large number of users.
    - What is the throughput of Bitcoin in tx/sec?! Can Bitcoin accommodate thousands of micrpayments per customer?
  - High transaction fees.
    - Each transaction must pay a fee.
    - This fee may exceed the payment value itself.

Aggregate the small (off-chain) payments into a few larger ones so only the large ones are logged on-chain!

A layer-2 solution!

Probabilistic Micropayments (lottery-based)

- A solution to aggregate tiny payments.
- Dated back to Rivest [Rivest, 1997] and Wheeler [Wheeler, 1996]--these are centralized solutions that involve centralized banks.



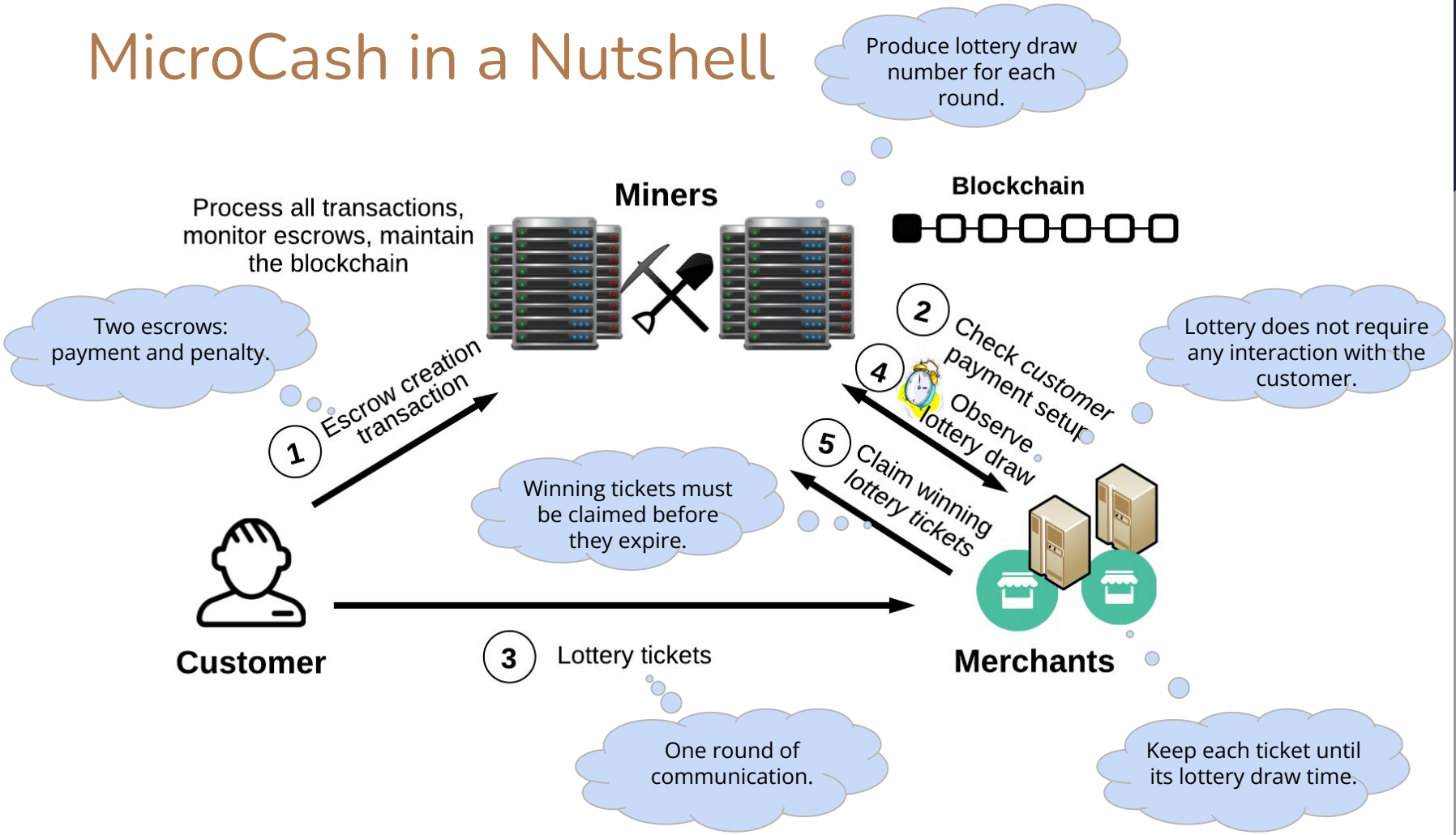
Decentralized Probabilistic Micropayments

- Utilize blockchain/cryptocurrencies to convert centralized schemes into distributed ones.
- Ingredients:
  - The bank is replaced with the miners.
  - Escrows/accounts are created on the blockchain.
  - Consensus/governance rules to manage escrows/accounts, claim/verify winning tickets, and punish cheaters.
- Will explore one system:
  - **MicroCash** [Almashaqbeh et al., 2020].

MicroCash

- The ***first*** decentralized probabilistic micropayment scheme that supports **concurrent micropayments**.
- The ***first*** to introduce a lottery with ***exact win rate.***
  - Non-interactive lottery requiring only secure hashing (in the random oracle model).
- Compared to older sequential micropayment schemes, it reduces the amount of data on the blockchain by around **50%.**
  - This is due to the fact an escrow can pay multiple winning tickets.

MicroCash in a Nutshell



Escrows and Micropayment Concurrency

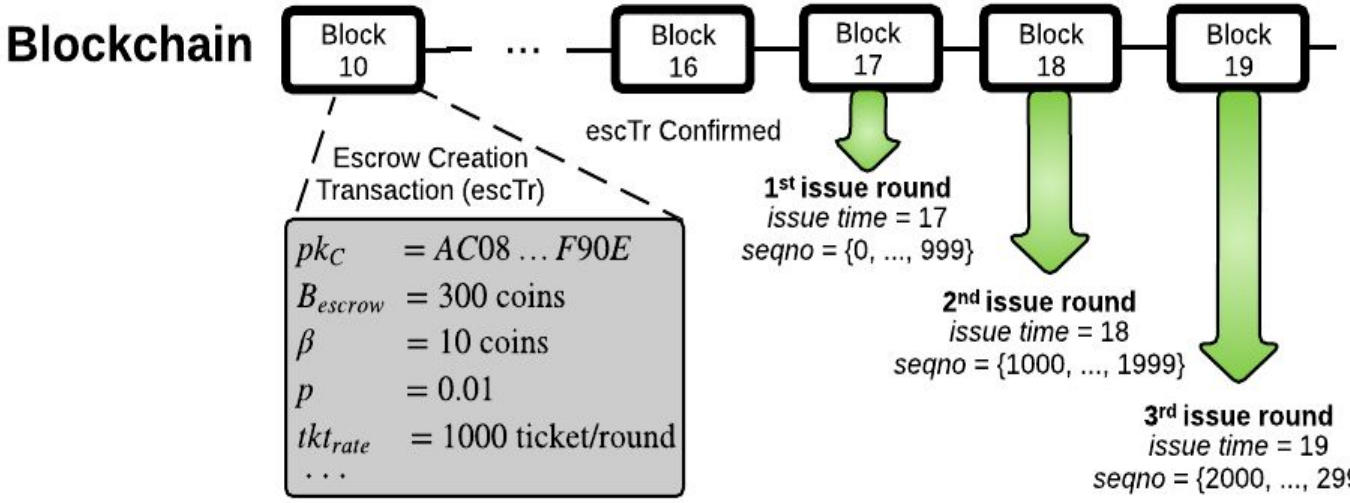
- The payment escrow balance covers all winning tickets.
  - A winning probability $\boldsymbol{p}$, ticket issue rate $\boldsymbol{tkt_{rate}}$, lottery phase length (in rounds) $\boldsymbol{draw_{len}}$, and escrow lifetime (in rounds) $\boldsymbol{l_{esc}}$.
  - For each lottery there are $\boldsymbol{p\ tkt_{rate} draw_{len}}$ winning tickets, each with value $\boldsymbol{\beta}$ coins, then the payment escrow balance is $\boldsymbol{\beta\ p\ tkt_{rate} draw_{len}}$
- Track tickets in the system based on their sequence numbers.
- Miners control escrows in the system.
- Each escrow must identify a set of beneficiary merchants.
- A customer can create an escrow that is sufficient to pay merchants for days.

Lottery Ticket Issuance

- Each ticket is a simple structure consisting of (idenitiy of the escrow, index of the merchant, tkt sequence number, and a signature over the tkt):

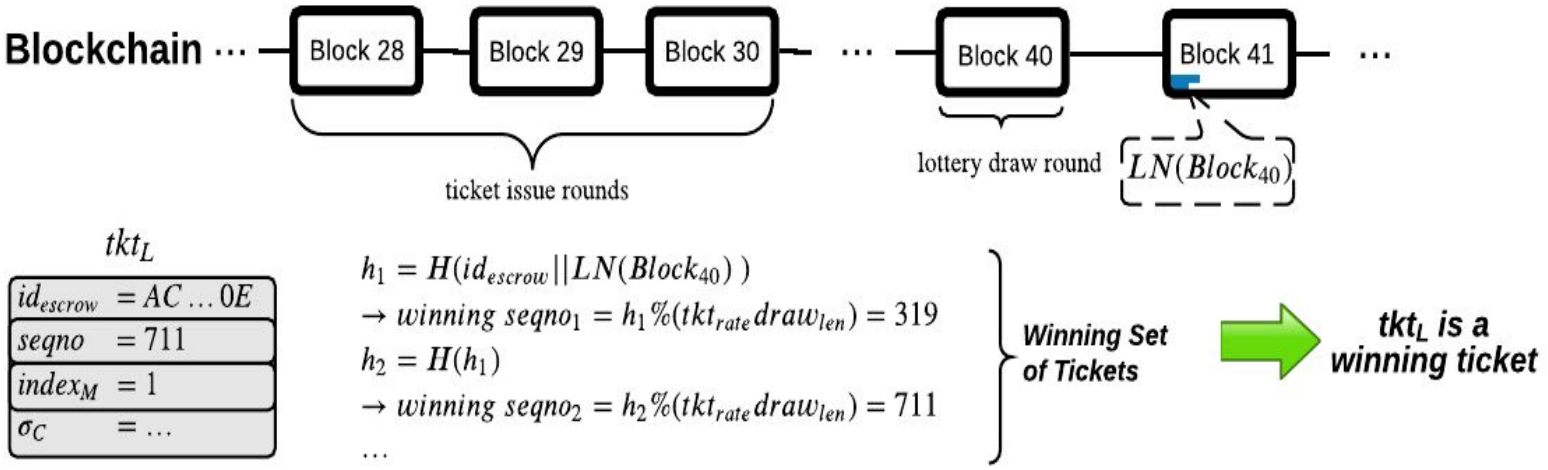
$$tkt_L = id_{esc} || index_M || seqno || \sigma_C$$

- Ticket issuance must follow a ticket issuing schedule.



The lottery Protocol

- Lightweight, non-interactive, and supports exact win rate.
  - Based on the blockchain view and requires only secure hashing.
- LN is a lottery number which is a verifiable delay function (VDF), which is repeated hashing) evaluated over the block.
  - **Why do not we use the block hash directly? *(hint: recall miner who mines a block sees the block hash before anyone else)***



Proof-of-cheating and Penalty Deposit

- Any party can issue a proof-of-cheating against the customer if it detects:
  - Duplicate ticket issuance, i.e., seeing more than one ticket with the same seqno under the same escrow for different merchants.
  - Issuing more tickets than the escrow can support, i.e., tickets with out-of-range sequence numbers.
- The miners burn the customer's penalty deposit.
  - This deposit must be large enough to make cheating unprofitable.
  - Its lower bound is derived using a game theoretic analysis of MicroCash setup.

MicroCash Security Properties

- Prevents **escrow overdraft**.
  - Front running attacks are not possible, i.e., customer cannot withdraw the escrow before paying off the merchants their winning tickets).
  - Ticket tracking prevent issuing more tickets than what can be covered.
- Prevents **escrow-withholding**.
  - An escrow will be refunded once all tickets expire.
- Prevents **manipulating the lottery** outcome.
  - Achieved by the use of VDFs and ticket issuing schedule.
- Addresses **duplicated ticket issuance**.
  - Using detect-and-punish approach.

MicroCash - Issues

- Not fully compatible with any of the cryptocurrencies out there.
  - *Why?*
- To address double spending, the set of merchants that can be paid by using an escrow must be set in advance.
  - Needed for the lower bound analysis of the penalty deposit.

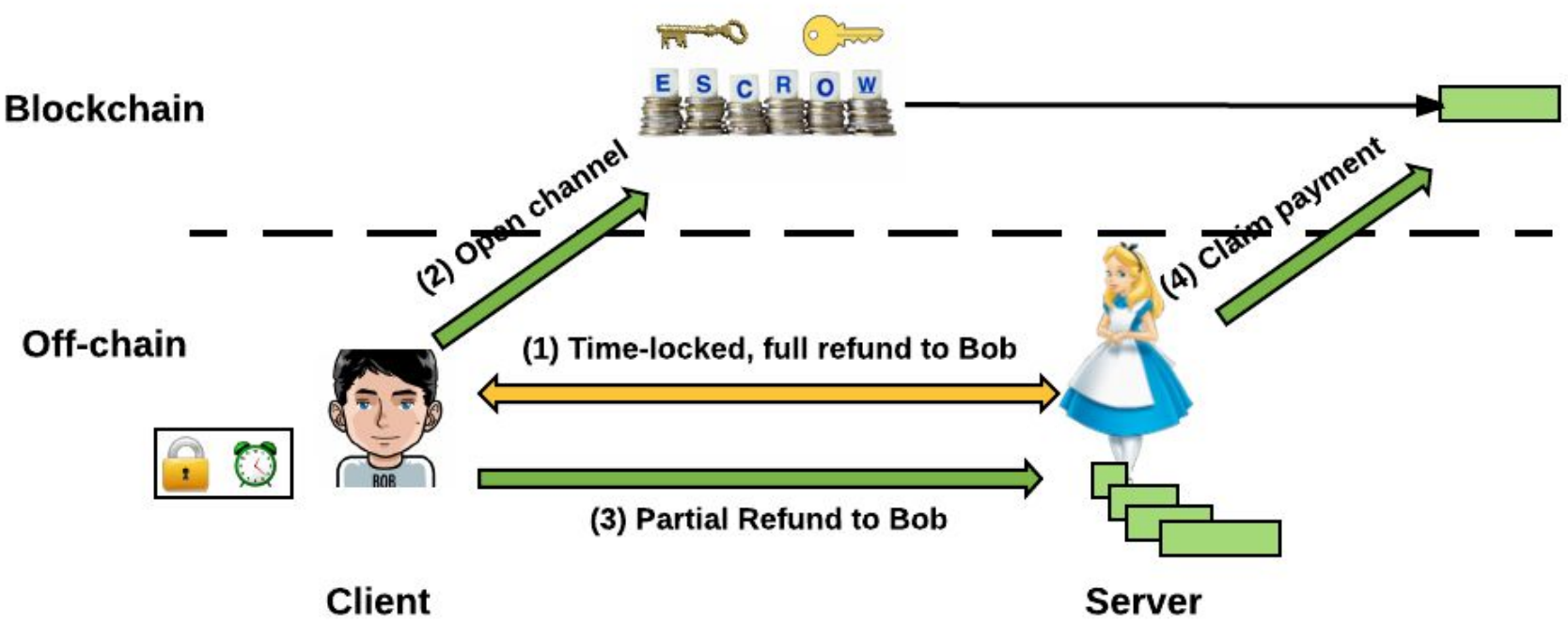
Layer 2 Solutions

Payment Channels/Networks

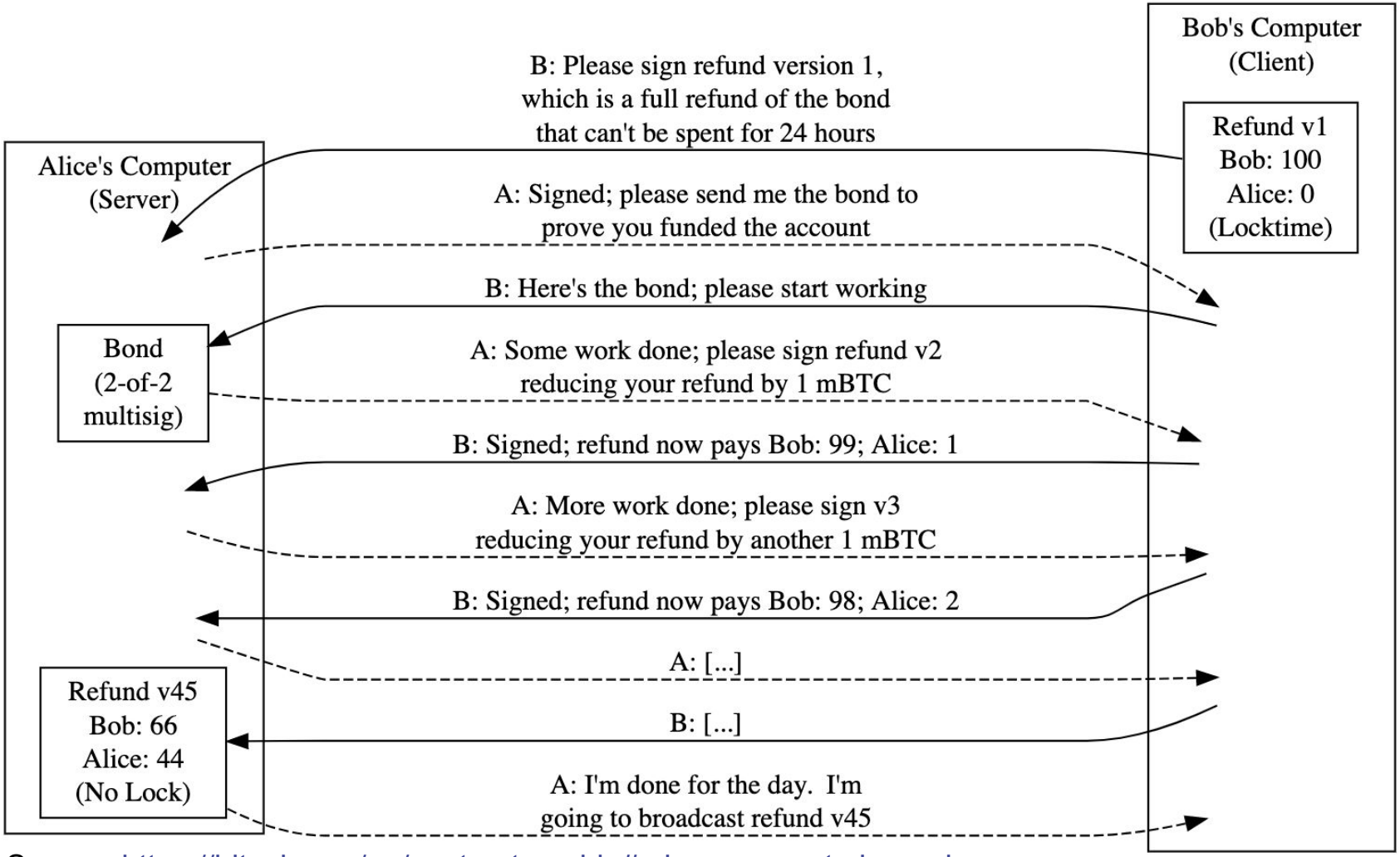
Payment Channels

- A payment channel is a contract between two parties locking a shared fund such that:
  - the amount of this shared fund owned by each party is adjusted over time to reflect the payments made so far.
- It is a way of processing transactions off-chain to reduce the number of on-chain transactions.
- A channel consists of two transactions:
  - Channel opening: a multi-sig transaction locking funds in an escrow.
  - Channel closing: a refund transaction expressing the latest state of the shared fund (how it is divided between the payer and the payee).
- Over time, the two parties exchange off-chain transactions, that reflect the change of the channel balance over time (i.e., how it is divided by the parties).

A Payment Channel Pictorially I



A Payment Channel Pictorially II



Source: https://bitcoin.org/en/contracts-guide#micropayment-channel

Payment Networks

- Payment channels allow only two parties to exchange payments.
- Payment networks allow any two parties that share a payment path to exchange payments.
  - A payment path is a set of contiguous payment channels connecting the payer and the payee.
  - Parties in between are called payment hubs, they may charge a fee for relaying payments.
- Main example is lightning networks [Poon et al., 2014].

Issues

- Drive the system toward centralization.
  - Only wealthy parties can afford to be payment hubs.
- Hubs charge fees for relaying payments.
  - ***Fees are back!*** They may exceed the micropayment value itself.
- But, payment channels (rather than networks) between long-term transacting parties (two parties) are still useful to handle micropayments.
- Currently payment channels/networks are more geared towards enhancing scalability (i.e., transaction throughput rate).
  - For **macropayments** (or regular size payments) rather than micropayments (i.e., small size ones).

Layer 2 Solutions

Rollups

Optimistic Rollups

- They batch transactions to be executed off-chain and only state changes are submitted on-chain.
  - As a matter of fact also the tarsnaction batch is published on-chain (for data availability), many projects are investigating how to avoid that.
- The blockchain (usually called mainchain) optimistically assumes that the transaction data and execution results are valid.
  - However, it does not finalize a processed rollup until a contestation period has elapsed.
  - During that period, external verifiers check the validity of the rollup result and issue fraud proofs if they disagree.

Optimistic Rollup Components

- **(1) The bridge smart contract:**
  - deployed on the mainchain, it is responsible for communication with the rollup. It sends transactions to the rollups for processing and posts the rollup results back to the mainchain.
- **(2) Sequencers:**
  - are responsible for collecting and ordering rollup transactions. They can receive these transactions either directly from rollup users or from the mainchain via the bridge.
- **(3) Executors:**
  - receive the ordered rollup transactions, execute them, and then produce the result (state changes) to be posted on the mainchain.
- **(4) Verifiers:**
  - They monitor the results posted by executors, verify their validity, and issue fraud-proof transactions if they find discrepancies.

Optimistic Rollups Pictorially

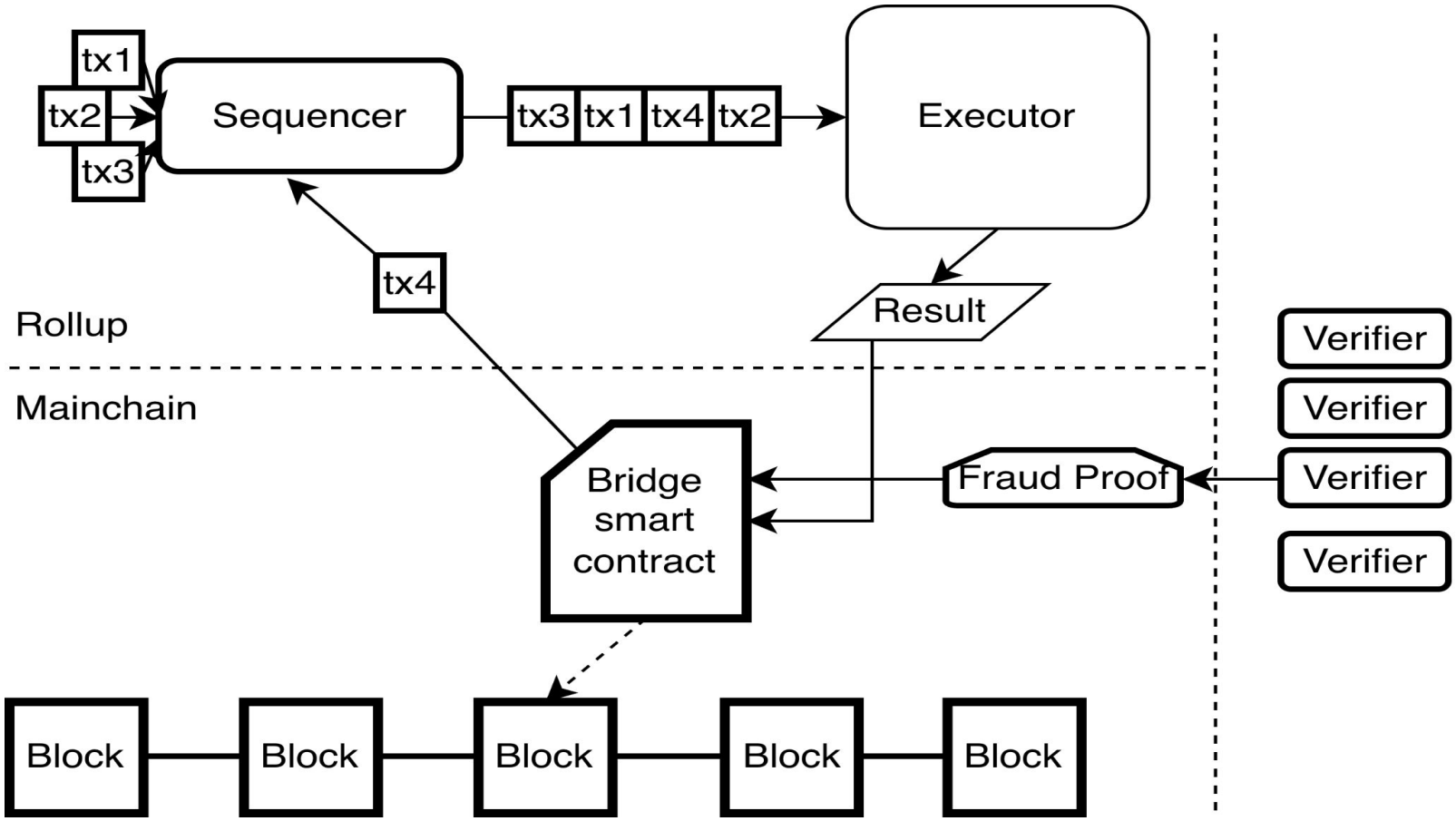


Figure credit: M. Najd

Issues

- **Centralization:** All deployed optimistic rollups to date have a single sequencer!
- **Long contestation period;** state changes are not considered final until this period is over. This period is usually around one week.
- **Incentive compatibility of verifiers is an open question;** it is unclear how to incentivize verifiers in a way that will ensure that they will do their job.
  - What happens if verifiers do not do their work?
- **Data availability;** the transaction batch must be made available for the verifiers (and everyone) to verify the produced state changes.
  - Generally it is published on-chain, but this will impact blockchain size!
  - New suggested approaches to store them temporarily outside the blockchain for pretty much the contestation period.

References

- [Rivest, 1997] Ronald Rivest.1997.Electronic lottery tickets as micropayments. In International Conference on Financial Cryptography. Springer, 307–314.
- [Wheeler, 1996] David Wheeler. 1996. Transactions using bets. In International Workshop on Security Protocols. Springer, 89–92.
- [Almashaqbeh et al., 2020] Ghada Almashaqbeh et al. "MicroCash: Practical Concurrent Processing of Micropayments." In Financial Cryptography, 2020.