Web3@KAIST

Building Web3 Apps to Solve Real Problems

*Building Web3 & Blockchain Applications (CS492 Special Topics in Computer Science) Spring 2023*

# Blockchain 101: Bitcoin

*Lecture 3 (2023-03-15)*

Min Suk Kang

Assistant Professor

School of Computing/Graduate School of Information Security

NetS&P Research Lab @ KAIST

# Logistics

- 2:30 hr lecture (we’ll end by 6:30pm)
- Short (3-min?) bathroom break in the middle

# Min Suk Kang (https://netsp.kaist.ac.kr)

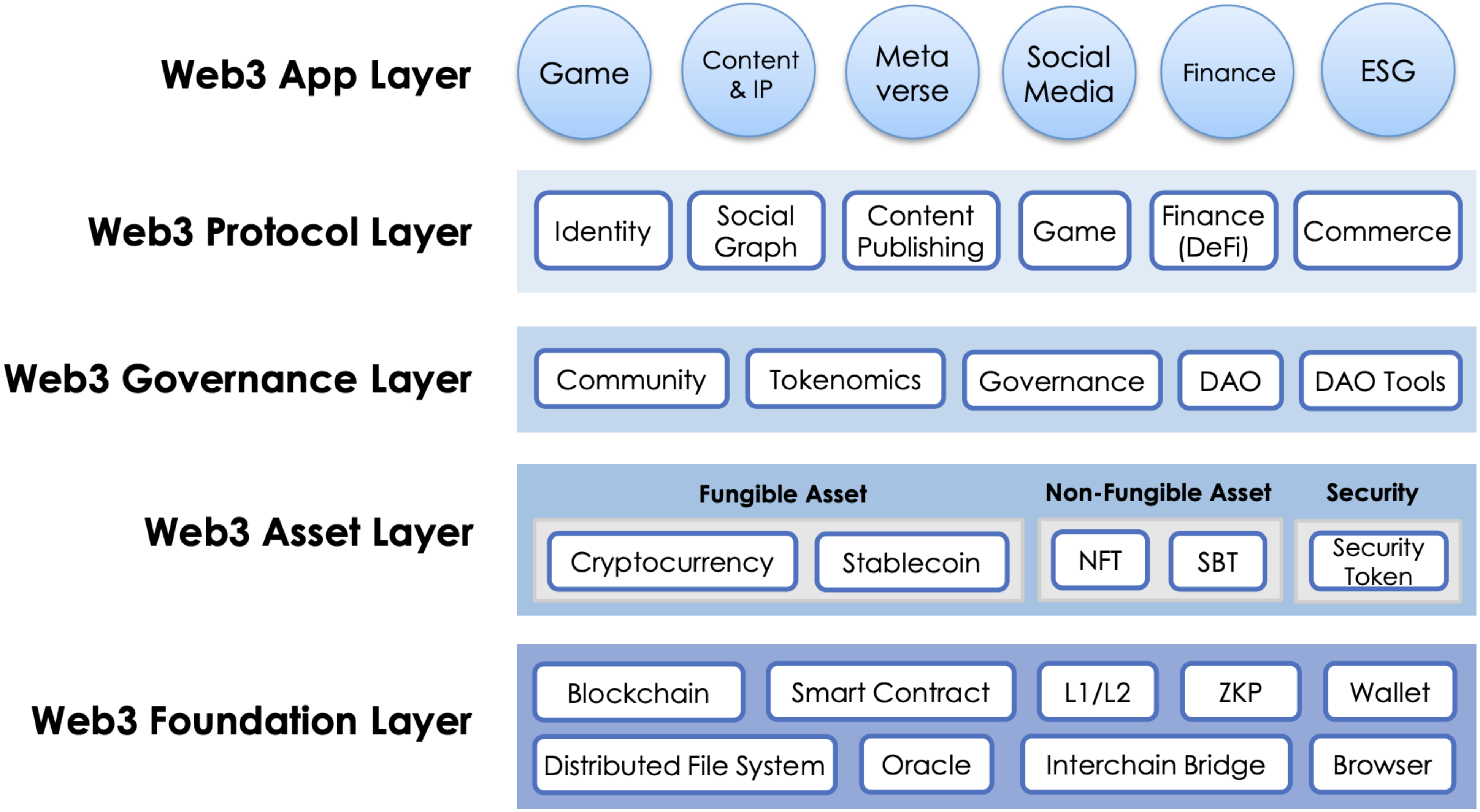
- Assistant Prof., School of Computing, KAIST (Since Aug 2020)
- Assistant Prof., School of Computing, NUS (2016-2020)
- Ph.D. ECE, Carnegie Mellon Univ (2016)
- MS & BS, EE, KAIST, South Korea (2008 & 2006)

List of blockchain research projects:

- Partitioning Bitcoin peer-to-peer networks
- Guaranteeing partition-resistant blockchain p2p
- Low-cost eclipse attacks in Ethereum
- Mixing Bitcoin transactions for better privacy
- Discovering consensus bugs in Bitcoin and Ethereum
- Enforcing network service guarantees for public blockchains
- …

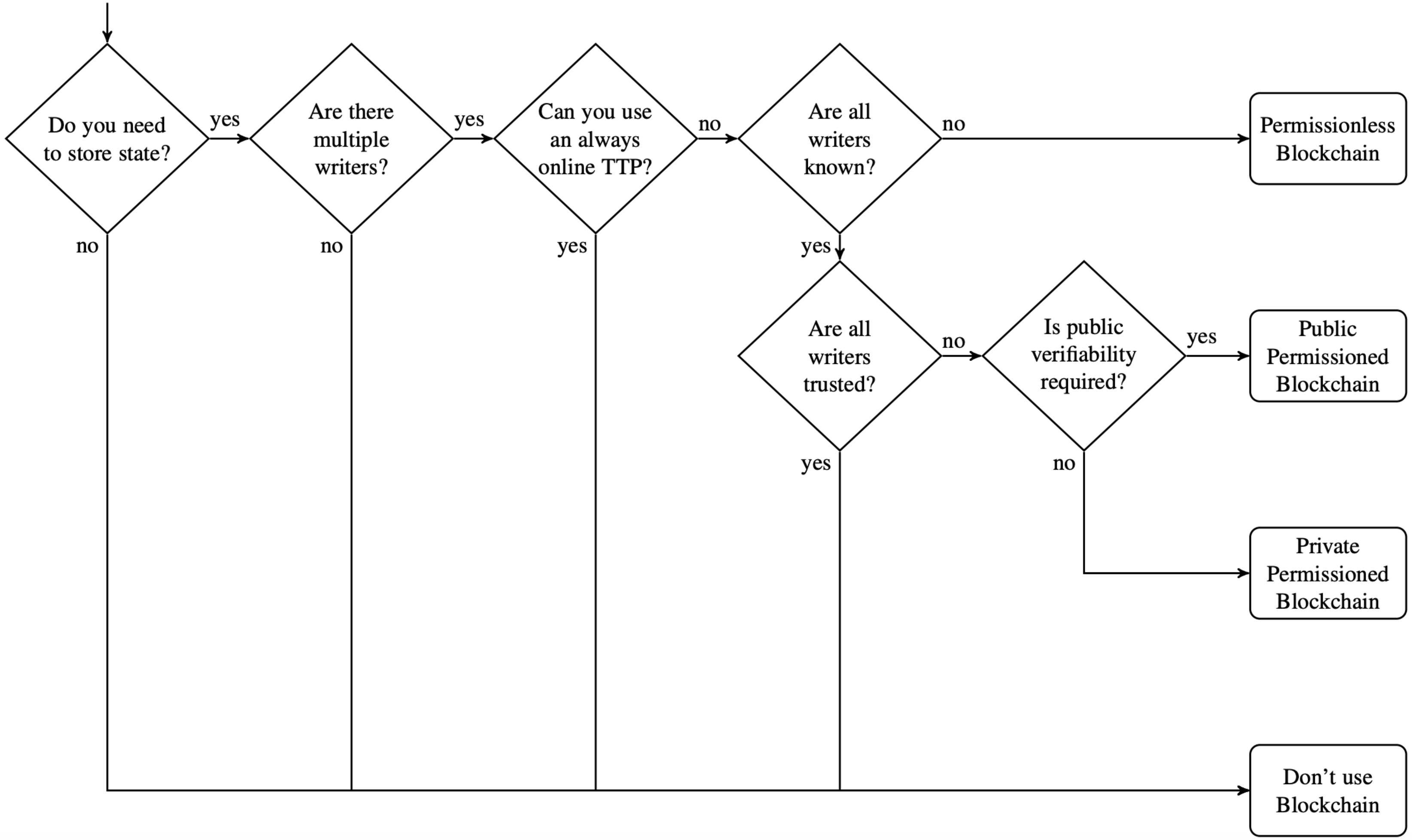
Privacy

# Web3 Stack (in the view of data)




(from Jason’s lecture slides)

# Why blockchain?
# What about Web3 without blockchain?



# Agenda

- Digital currency
  - Why is it hard?
  - What properties should we achieve?
- Nakamoto consensus
  - How Bitcoin solved it?
- Ethereum as the world computer
  - Smart contracts
  - Proof of stake
- What’s more? (next week)

Many slides borrowed from good researchers
Prateek Saxena, Dan Boneh, Ertem Nusret Tas

# Online Transactions

- Physical cash
  - Non-traceable (well, mostly!)
  - Secure (mostly)
  - Low inflation

- Can’t be used online directly
- Electronic credit or debit transactions
  - Bank sees all transactions
  - Merchants can track/profile customers

# E-Cash

- Secure
  - Single use
  - Reliable
- Low inflation
- Privacy-preserving

# E-Cash Crypto Protocols

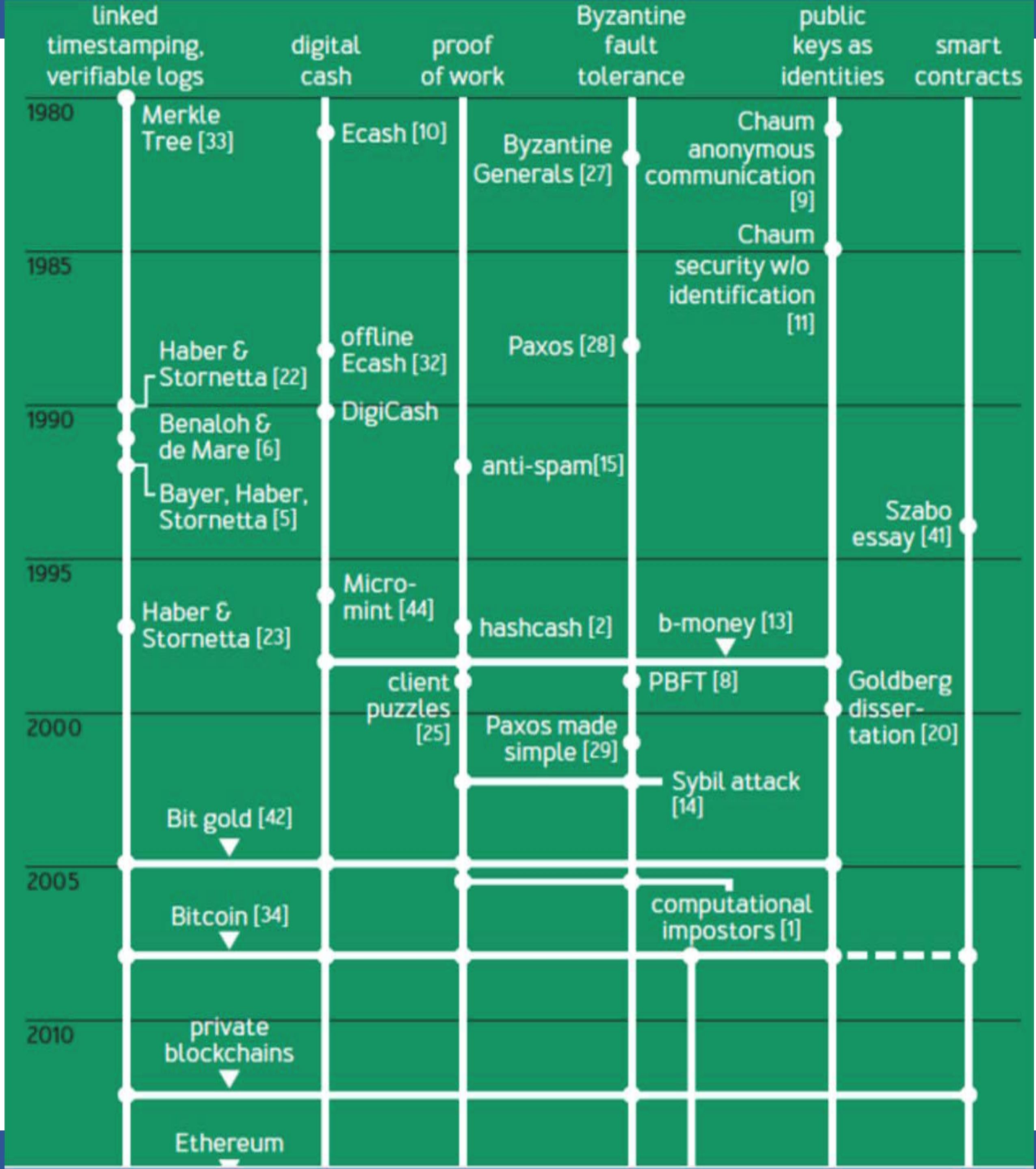
- Chaum82: blind signatures for e-cash
- Chaum88: retroactive double spender identification
- Brandis95: restricted blind signatures
- Camenisch05: compact offline e-cash

- Various practical issues:
  - Need for trusted central party
  - Computationally expensive
  - Etc.

# Bitcoin

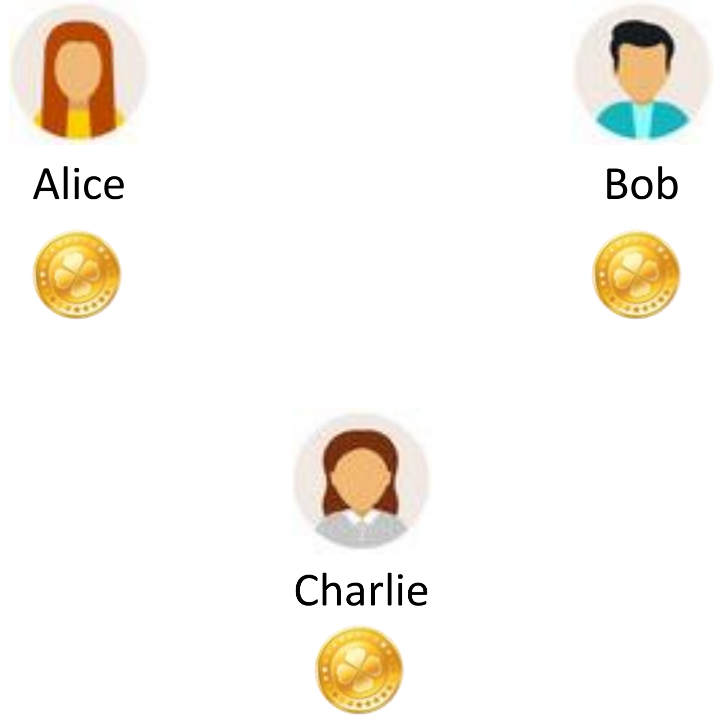
- A distributed, decentralized digital currency system
- Released by Satoshi Nakamoto 2008
- Effectively a bank run by an ad hoc network
  - Digital checks
  - A distributed transaction log

# Chronology of Ideas in Bitcoin




(Narayanan and Clark)

# Self-regulating currency

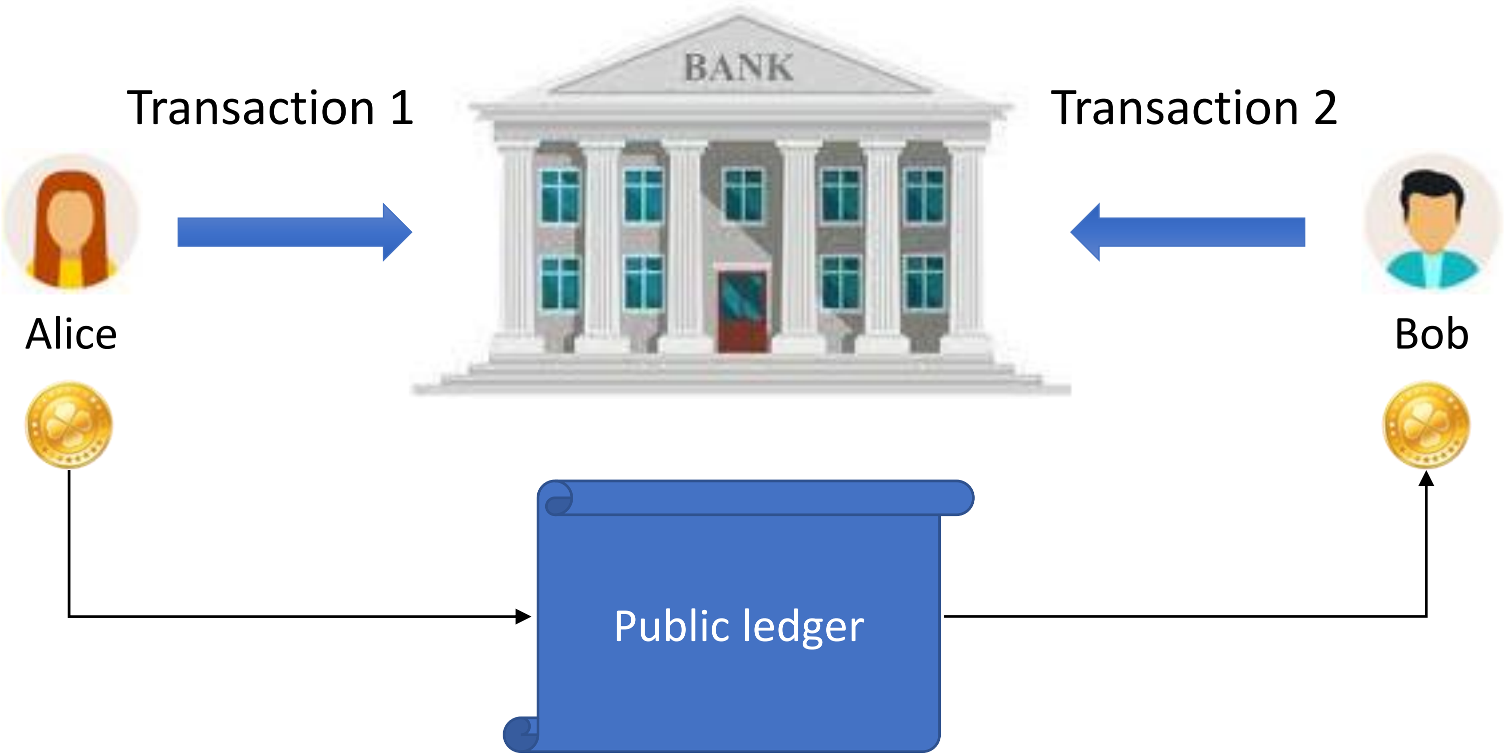



TX-1: Alice -> Bob
TX-2: Bob -> Charlie

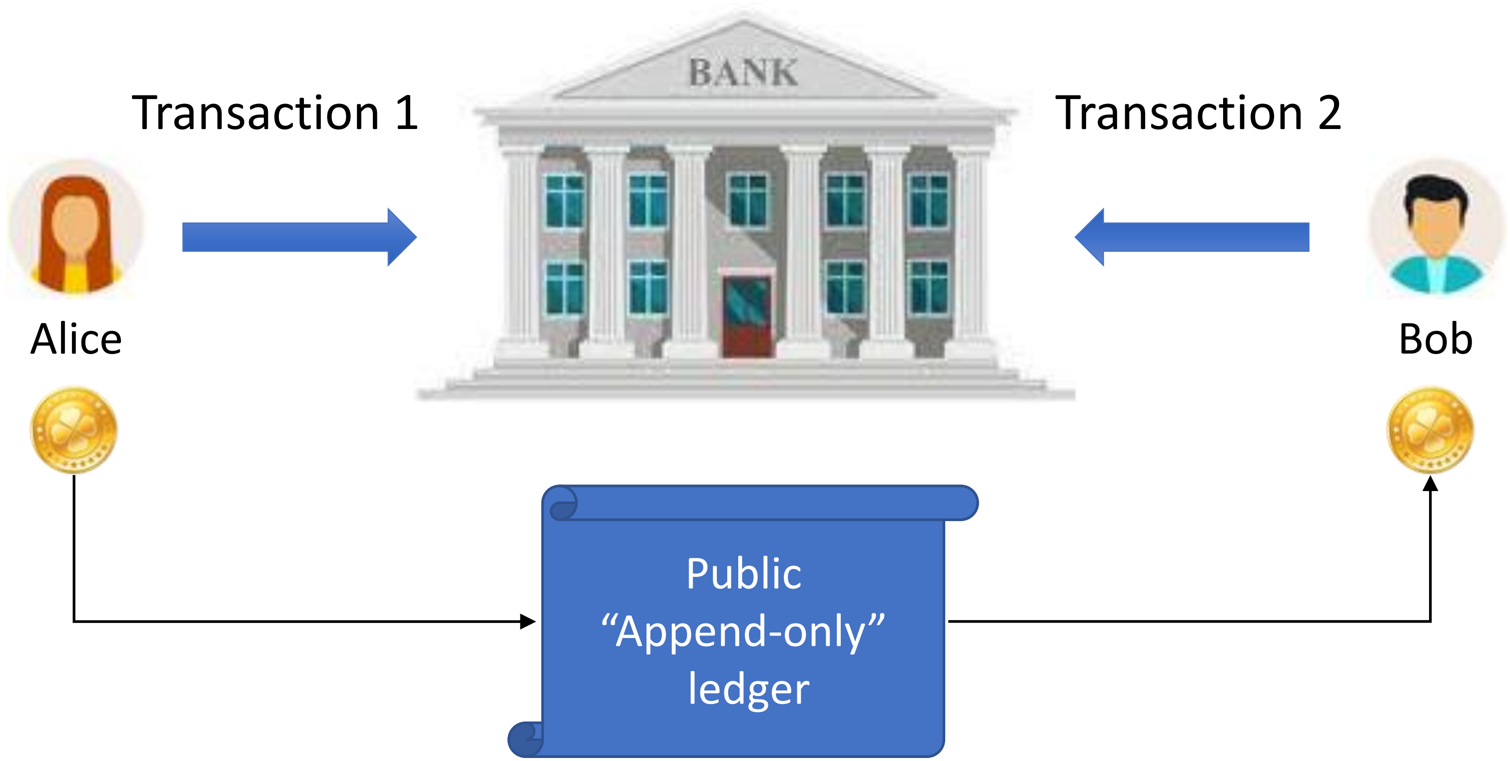
TX-1: Bob -> Charlie
TX-2: Alice -> Charlie

TX-1: Alice -> Bob
TX-2: Alice -> Charlie

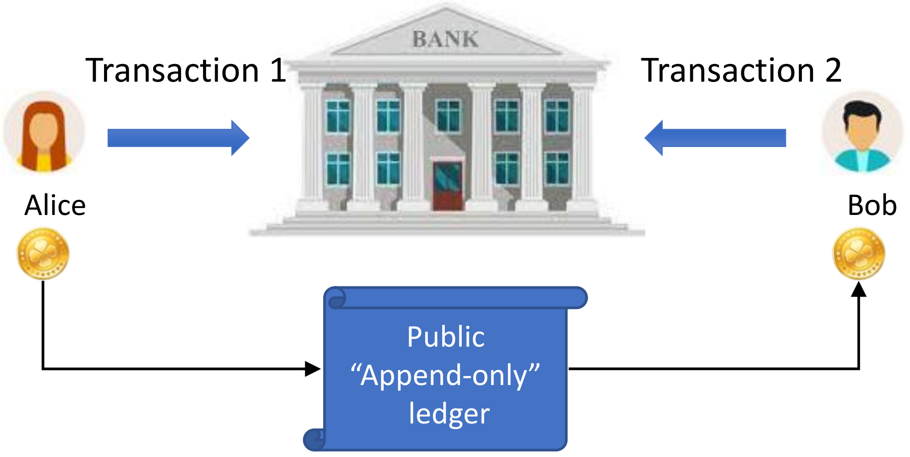
# Self-regulating currency



# Almost a solution



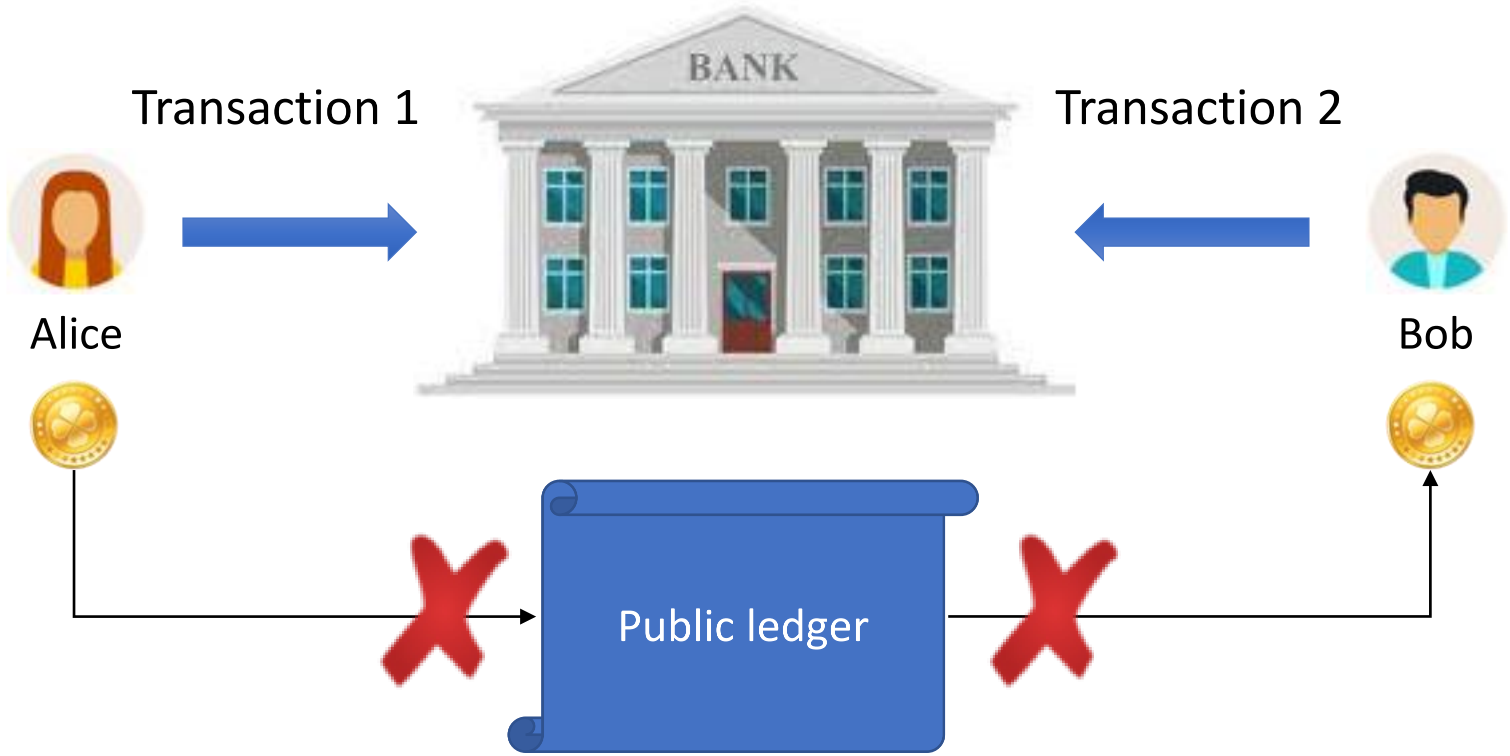
# Almost a solution




- Anyone can verify
  - Alice has enough balance
  - She **authorized** a transaction to Bob
  - New balances credit-debited correctly

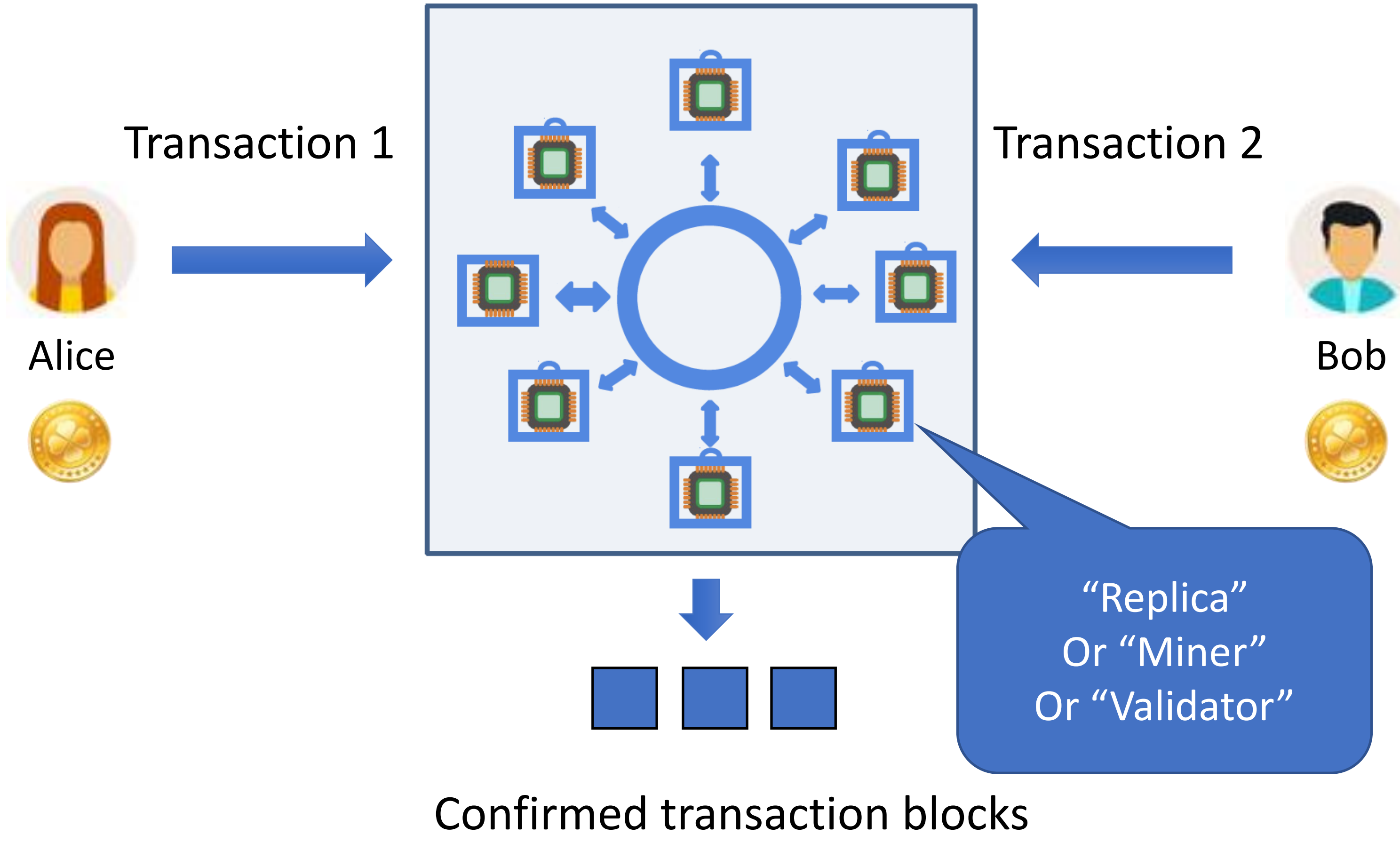
- E.g., Alice digitally signs "I want to pay Bob $45"
  - Digital signatures: authenticity and integrity
  - Alice publishes her **public key**
    - Does not need to reveal her real identity
    - Keeps her private key secret

# So, what’s difficult in Bitcoin-like systems?




- Provide **correctness of a distributed append-only ledger** (fault-tolerance)
- Prevent **censorship of transactions** for some users (fairness)

# State machine replication (SMR)



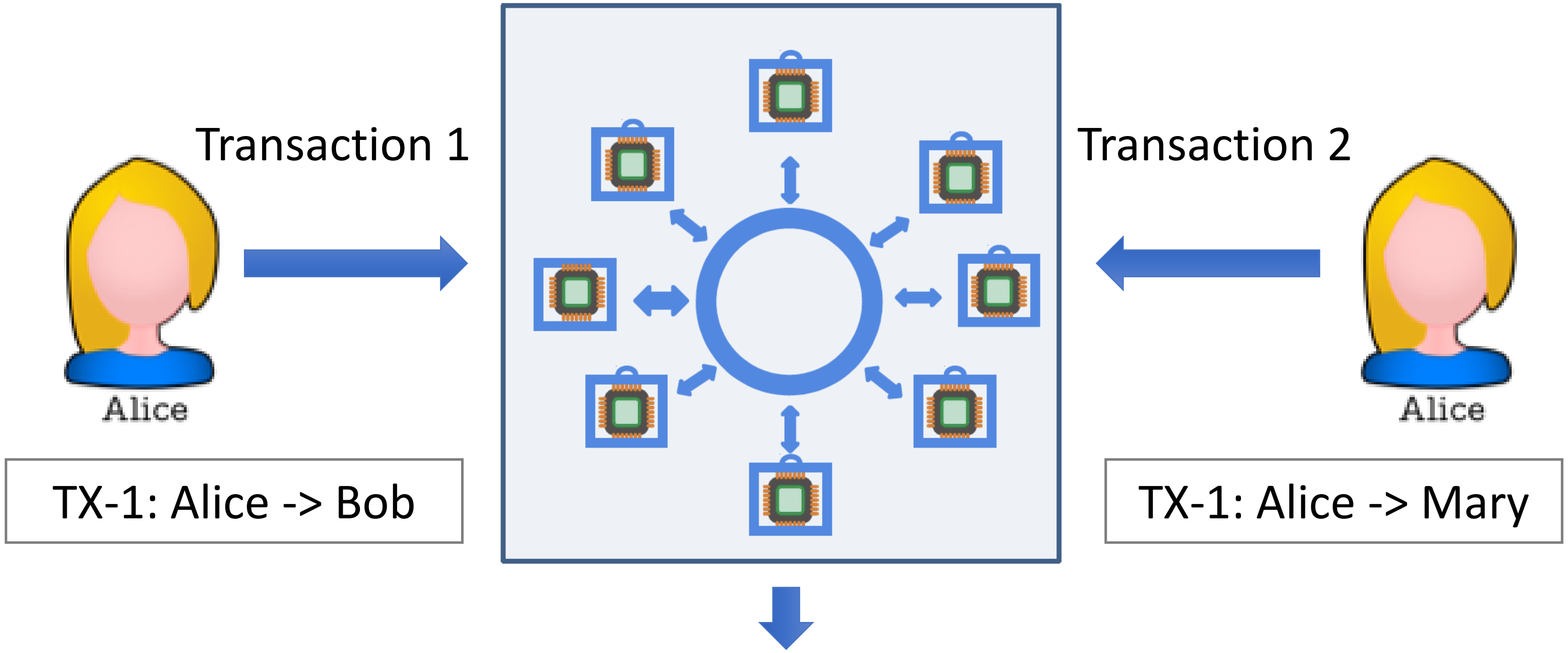
# Goals of blockchain consensus

- A continuous process… 1 block every 10 minutes

| | | |
|---|---|---|
| TX-1: Alice -> Bob<br>TX-2: Bob - > Mary<br>… | TX-1: Mary -> Bob<br>TX-2: Bob - > Alice<br>… | TX-1: Mary -> Alice<br>TX-2: Bob - > Alice<br>… |
| 10:30 AM<br>May 1, 2021 | 10:40 AM<br>May 1, 2021 | 10:50 AM<br>May 1, 2021 |

- Transactions are totally ordered in “blocks”
- Blocks are totally ***ordered in time***
  - Anyone can verify their order

# Key Challenge:
# Agreement over Transaction Ordering




| TX-1: Alice -> Bob<br>TX-2: Alice - > Mary | ≠ | TX-1: Alice -> Mary<br>TX-2: Alice - > Bob |
|---|---|---|

Ordering Transactions is sufficient to prevent double spending

# What is a blockchain?

Abstract answer: a blockchain provides
coordination between many parties,
when there is no single trusted party

if trusted party exists $\Rightarrow$ no need for a blockchain

[financial systems: often no trusted party]

# Blockchains: what is the new idea?

2009

**Bitcoin**

Several innovations:

- A practical **public append-only data structure**, secured by replication and incentives
- A fixed supply asset (BTC). Digital payments, and more.

# Blockchains: what is the new idea?

2009 — **Bitcoin**

2015 — **Ethereum**

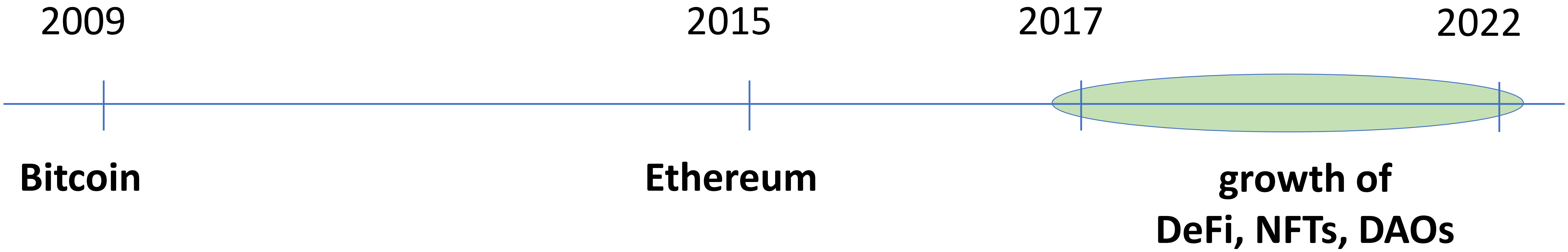
Several innovations:

- **Blockchain computer**: a fully programmable environment

  $\Longrightarrow$ public programs that manage digital and financial assets

- **Composability**: applications running on chain can call each other

# Blockchains: what is the new idea?



# Consensus layer (informal)

achieved by replication

A **public** append-only data structure:

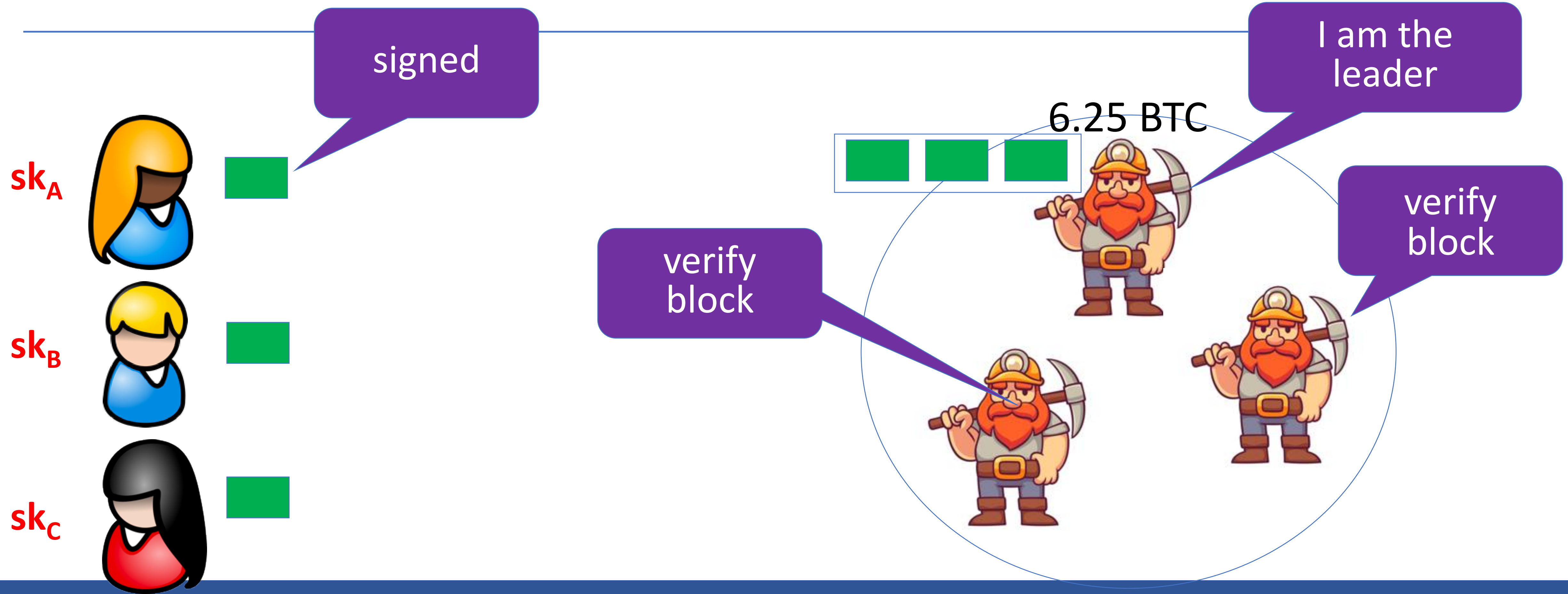
- **Persistence**: once added, data can never be removed*
- **Safety**: all honest participants have the same data**
- **Liveness:** honest participants can add new transactions
- **Open(?)**: anyone can add data (no authentication)

# Other desired properties

- ***Fairness***: Your confirmed blocks are proportional to the computational power you have connected
- ***Throughput***: Lots of transactions per unit time
- ***Latency***: Short timeframe to confirm a transaction
- ***Decentralization***: Large # of miners proposing transaction blocks

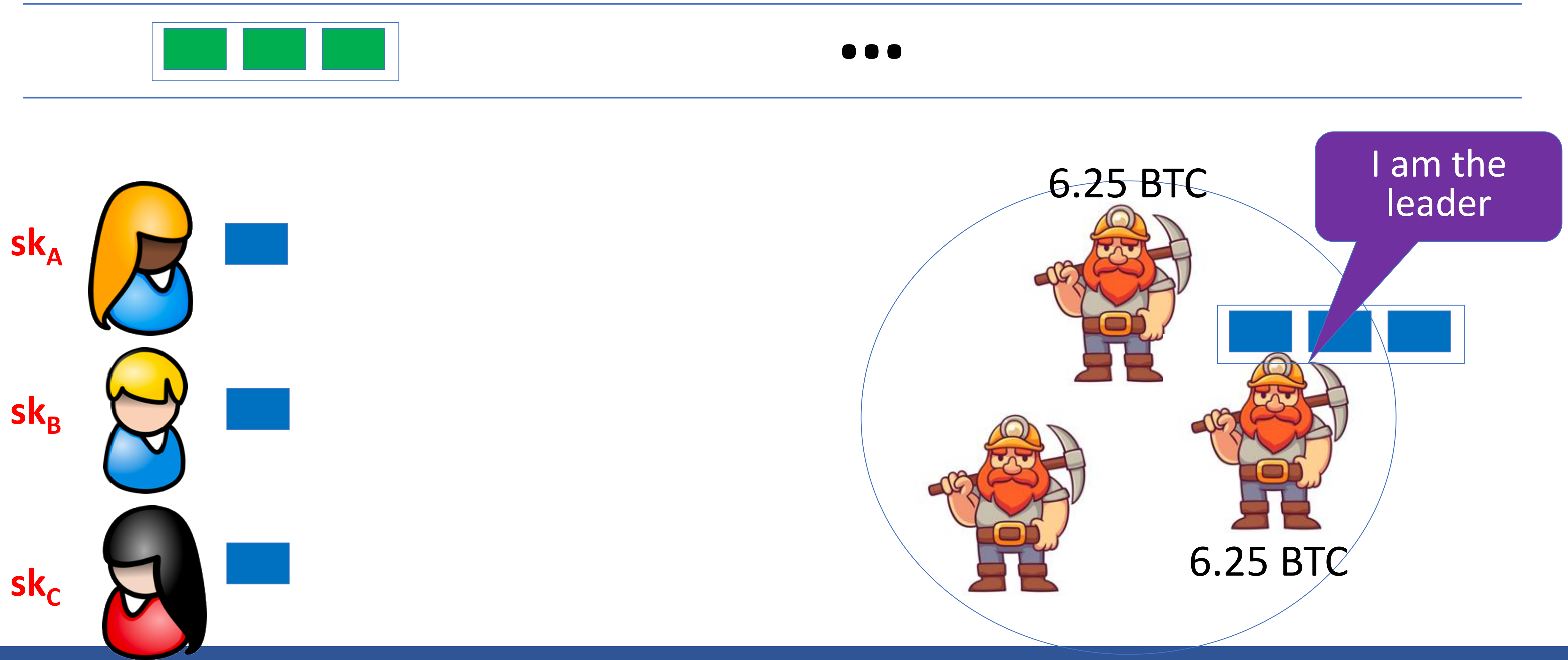
# How are blocks added to chain?

blockchain

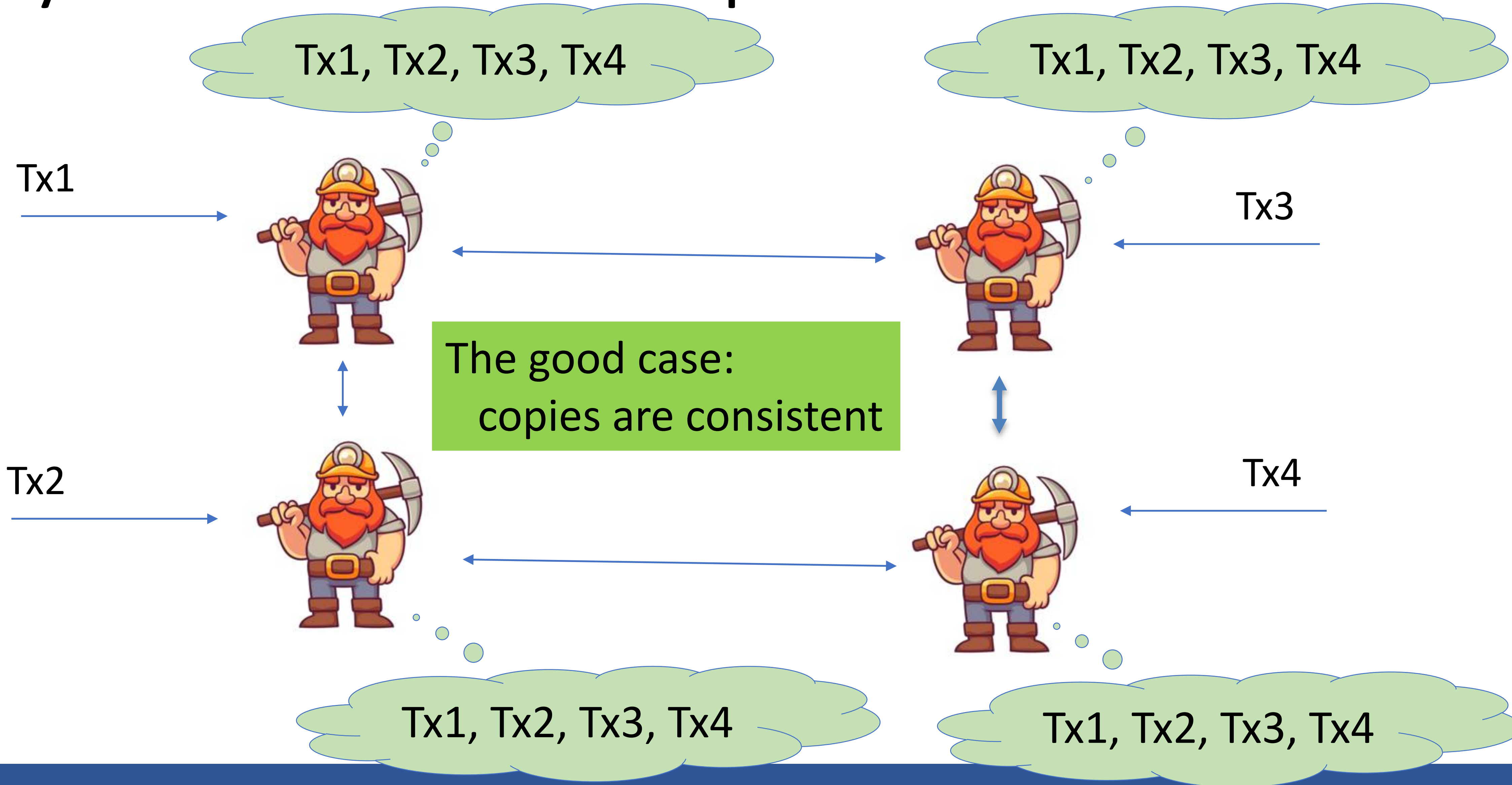


# How are blocks added to chain?

blockchain



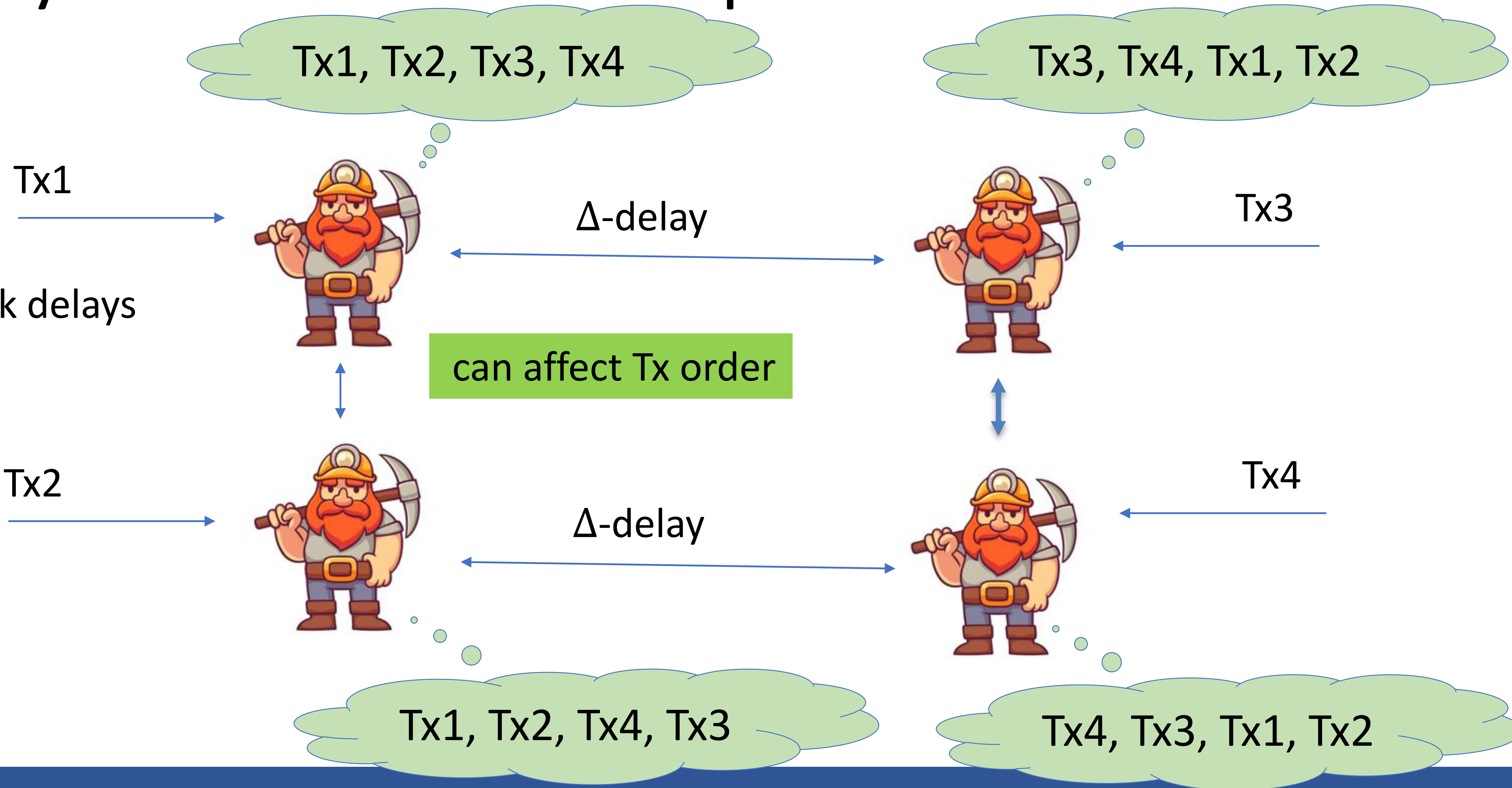
# Why is consensus a hard problem?



# Why is consensus a hard problem?

Problems:

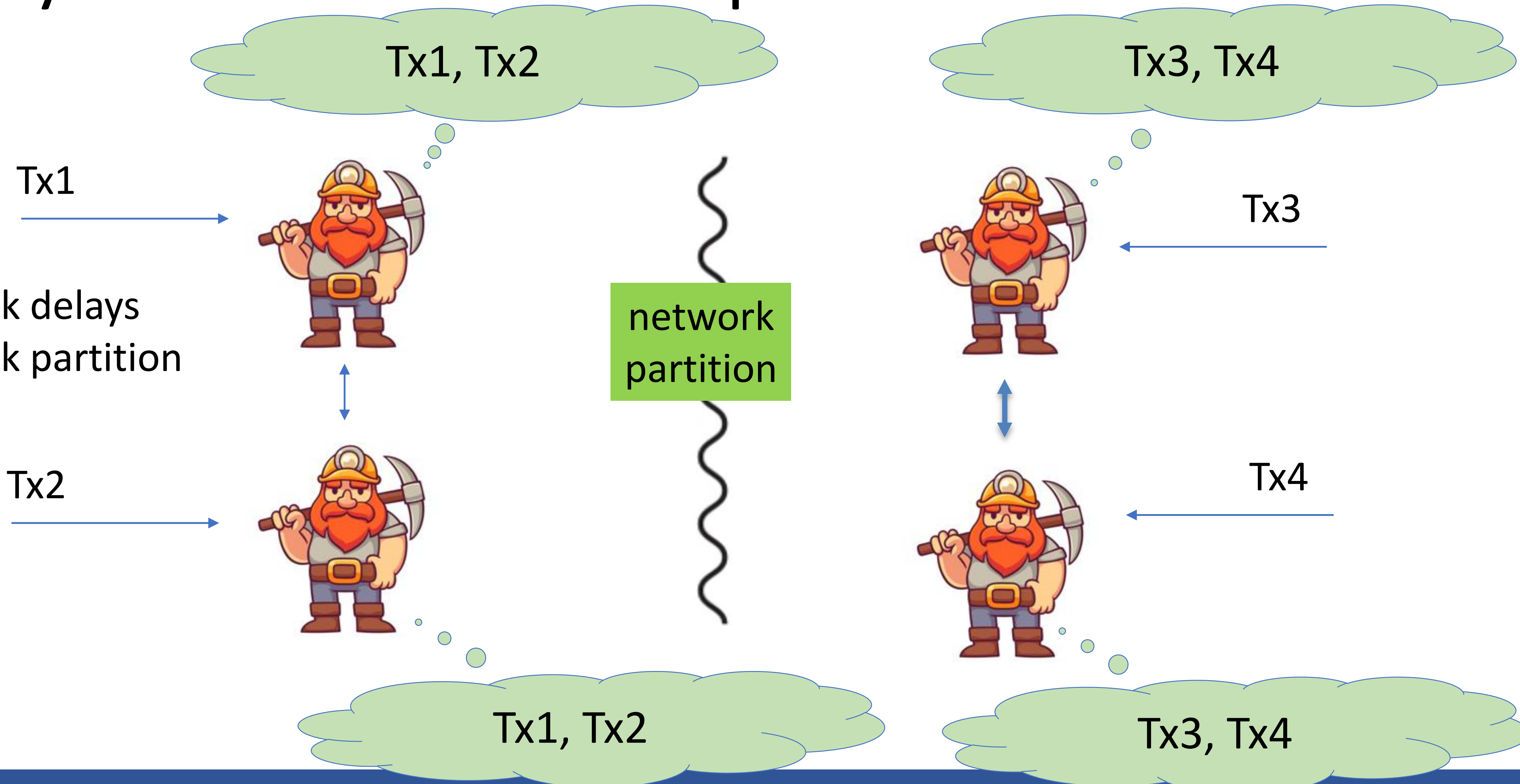
- Network delays



# Why is consensus a hard problem?

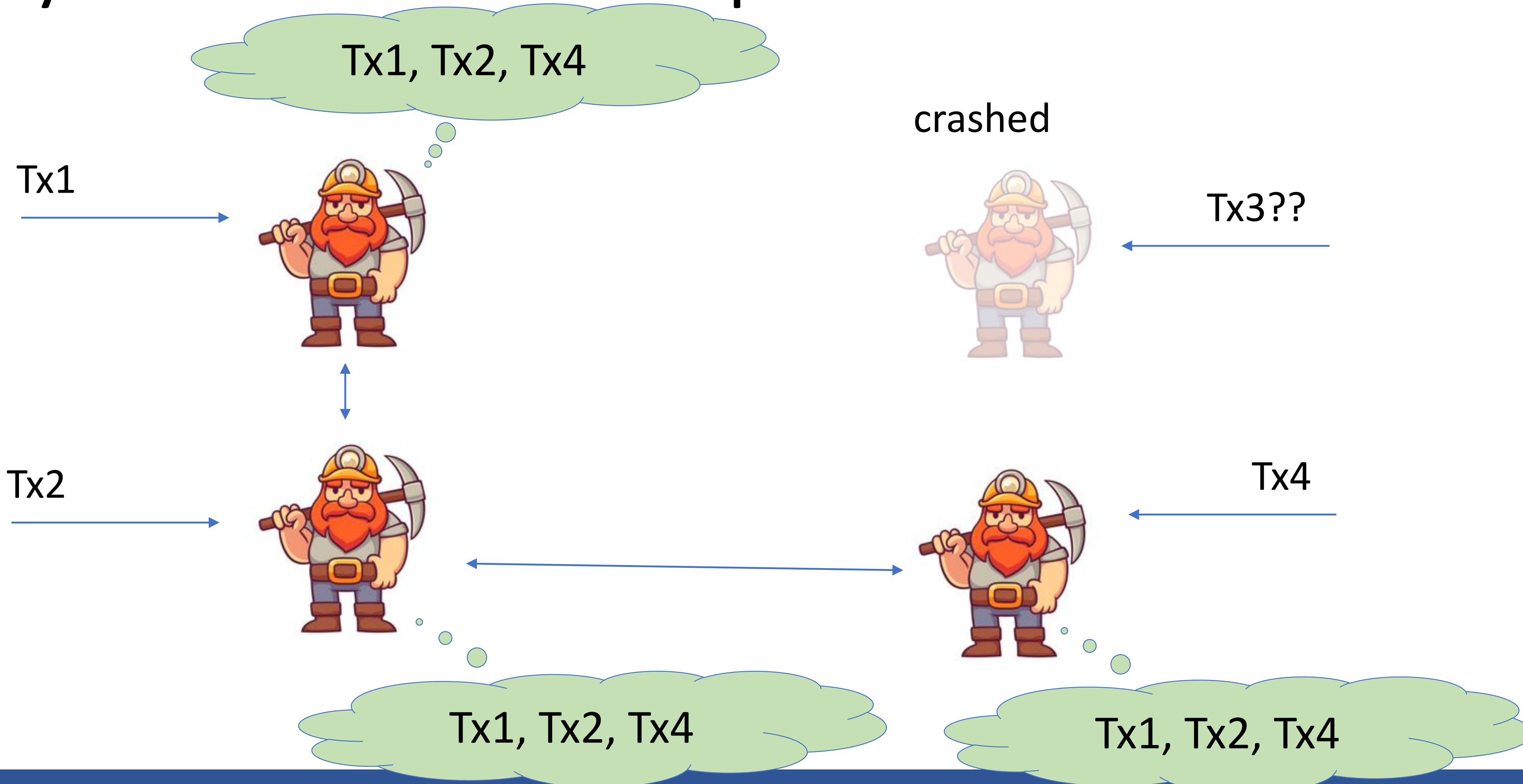
Problems:

- Network delays
- Network partition

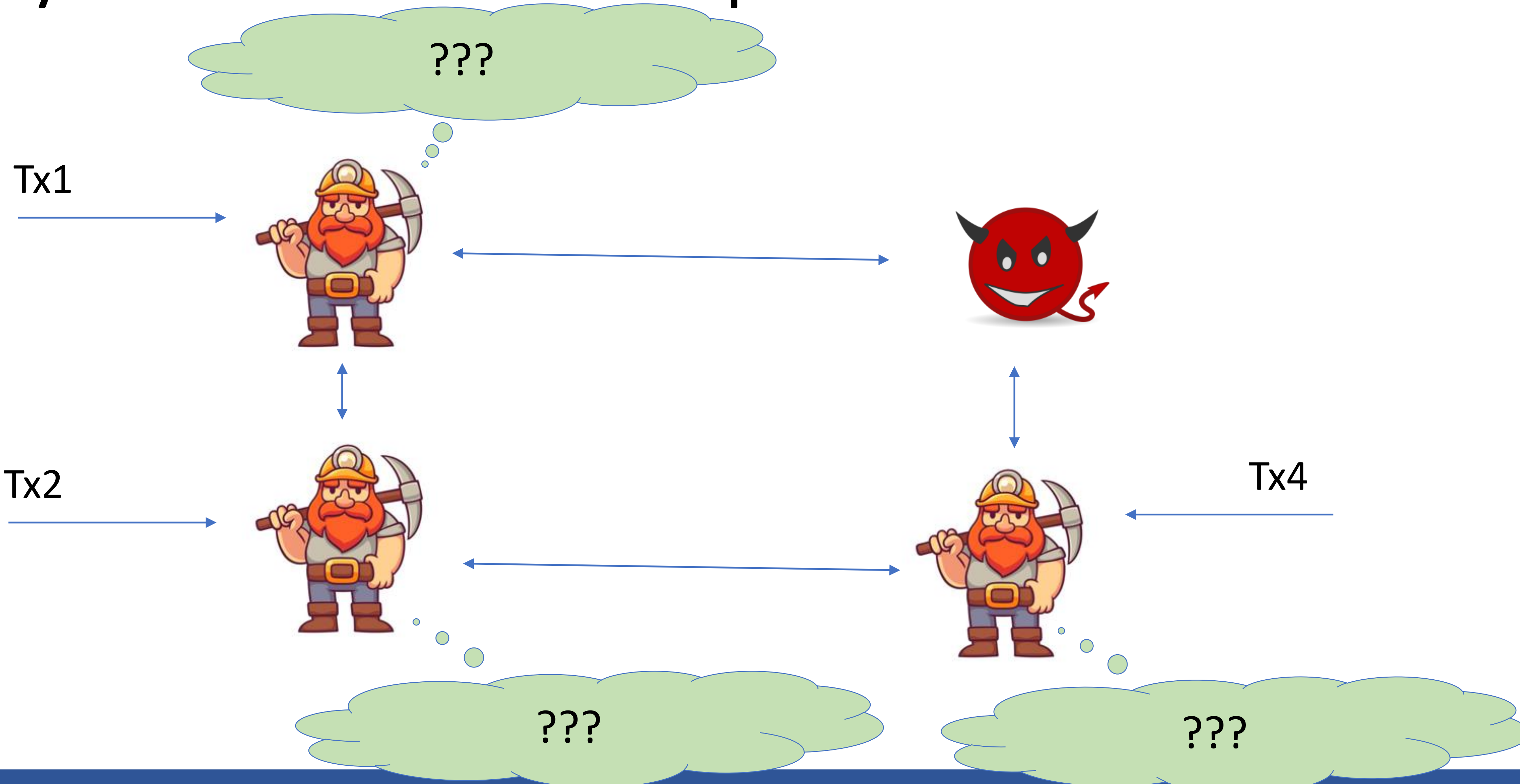


# Why is consensus a hard problem?

Problems:
- crash

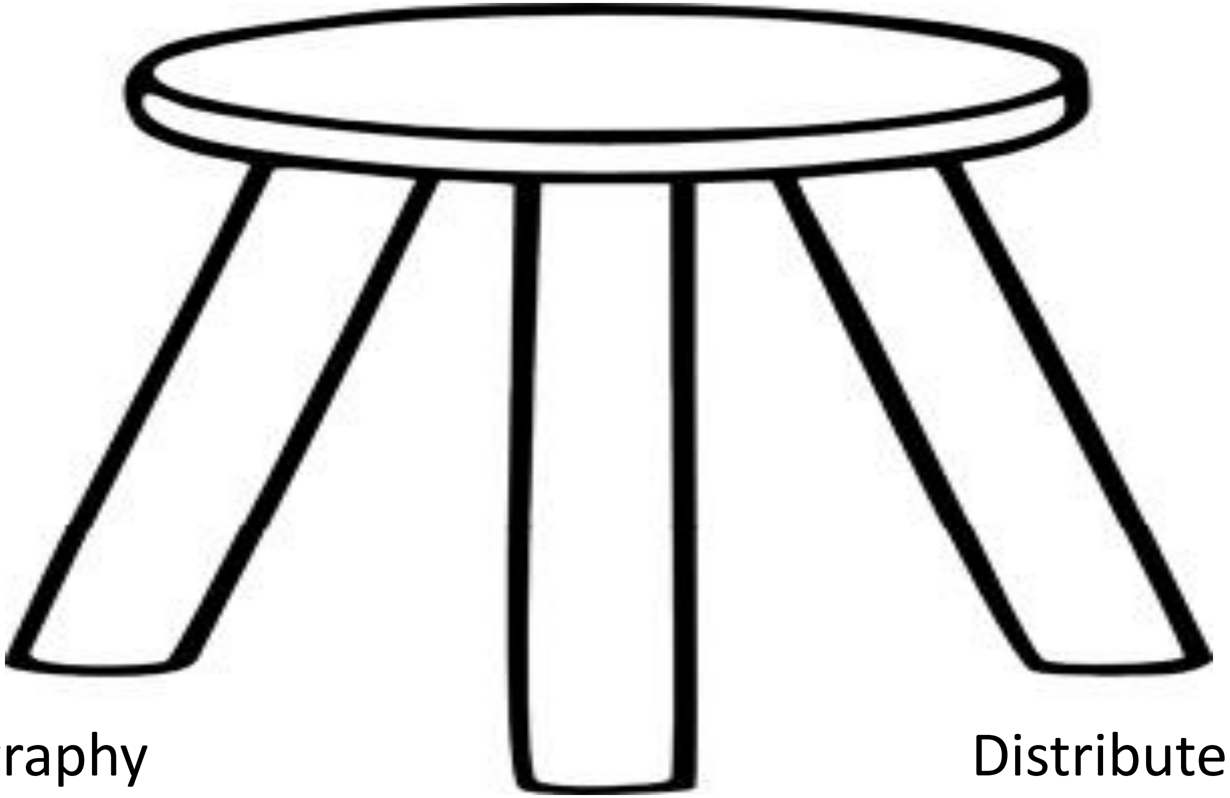


# Why is consensus a hard problem?




Problems:
- crash
- malice

# Blockchain systems…



# Cryptography Background

(1) cryptographic hash functions

An efficiently computable function $H\colon\ M \to T$

where $|M| \gg |T|$


$T = \{0,1\}^{256}$

# Collision resistance

**Def**: a **collision** for $H: M \to T$ is pair $x \neq y \in M$ s.t. $\boxed{H(x) = H(y)}$

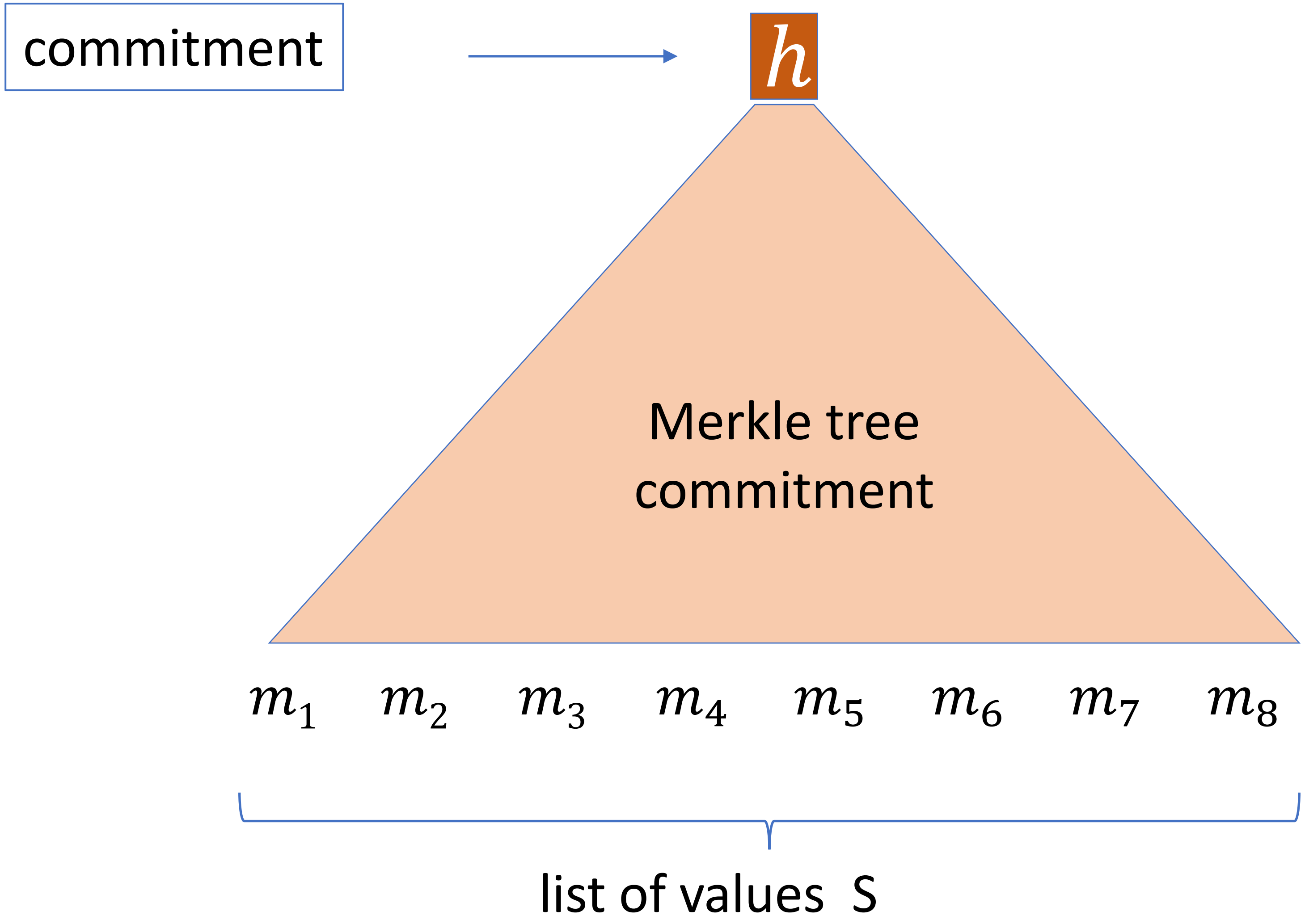
$|M| \gg |T|$ implies that <u>many</u> collisions exist

**Def:** a function $H: M \to T$ is **collision resistant** if it is “hard” to find even a single collision for $H$ (we say $H$ is a CRF)

Example: **SHA256**: $\{x : \text{len}(x) < 2^{64} \text{ bytes}\} \to \{0,1\}^{256}$

(output is 32 bytes)

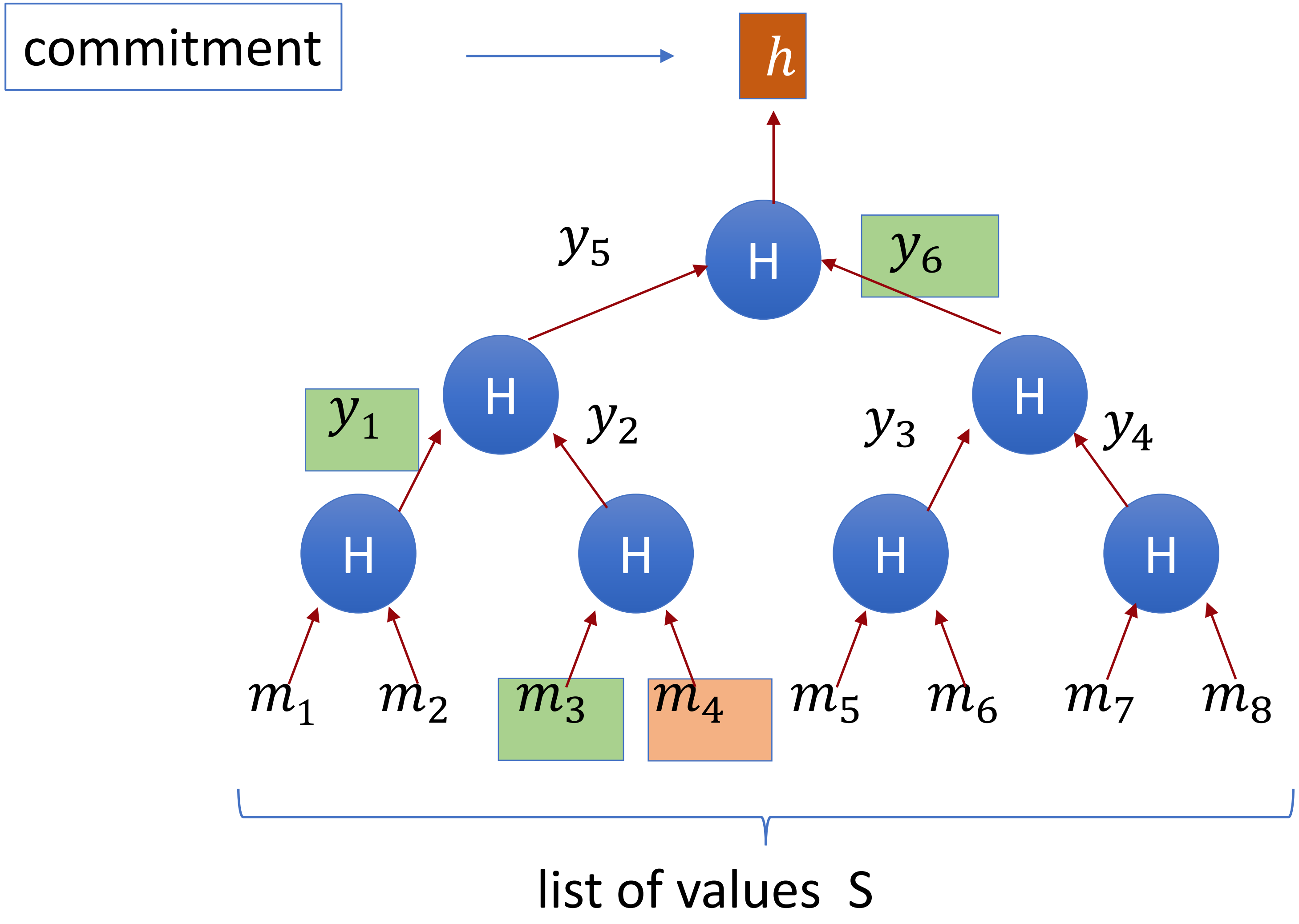
# Merkle tree (Merkle 1989)




Goal:

- commit to list S of size n
- Later prove $S[i] = m_i$

# Merkle tree (Merkle 1989) [simplified]




Goal:

- commit to list S of size n
- Later prove $S[i] = m_i$

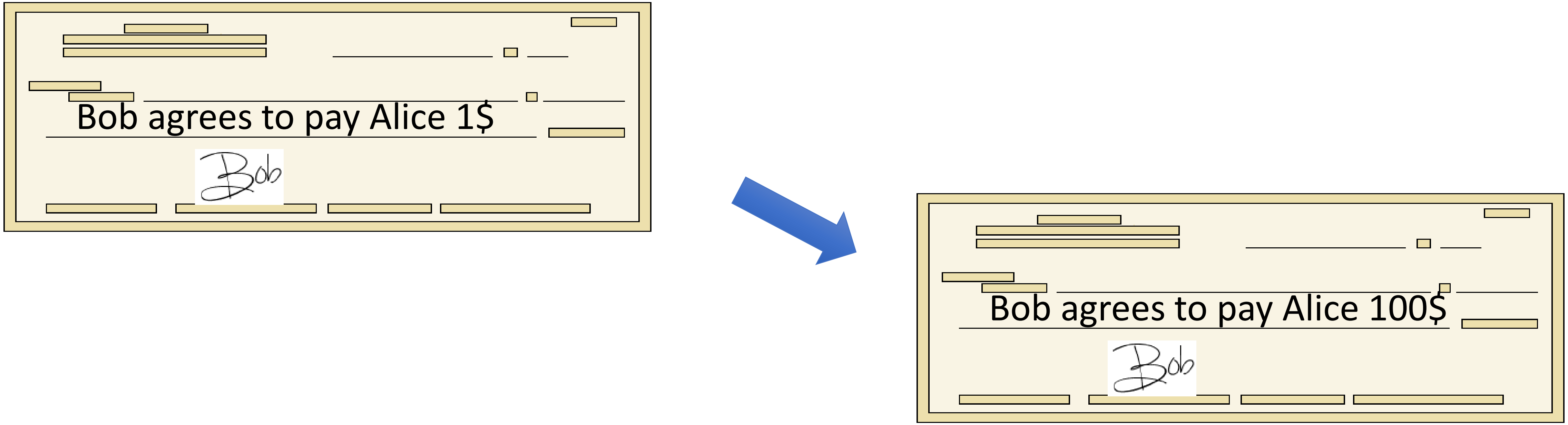
To prove $S[4] = m_4$ ,

proof π $= (m_3, y_1, y_6)$

length of proof: $\log_2 n$

# Signatures

Physical signatures: bind transaction to author

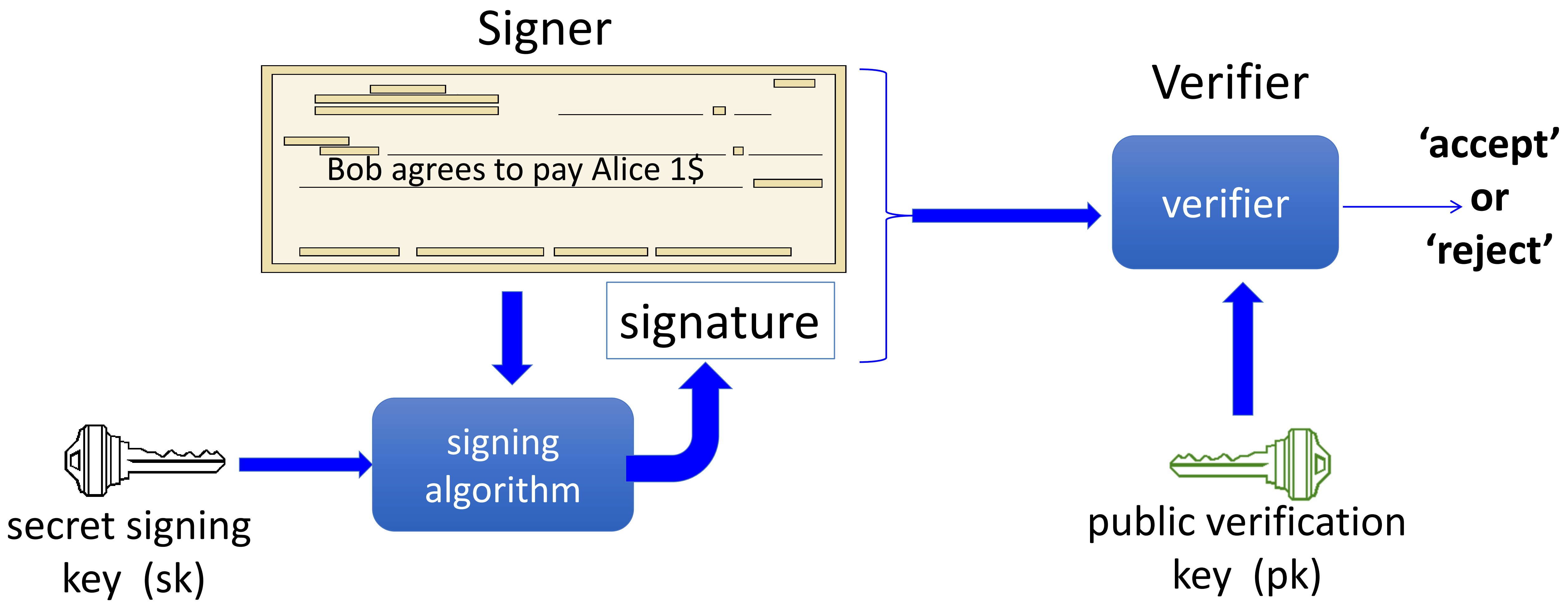



Problem in the digital world:

anyone can copy Bob's signature from one doc to another

# Digital signatures

Solution:  make signature depend on document



# Families of signature schemes

1. RSA signatures (old … not used in blockchains):
   - long sigs and public keys (≥256 bytes), fast to verify

2. Discrete-log signatures: Schnorr and ECDSA (Bitcoin, Ethereum)
   - short sigs (48 or 64 bytes) and public key (32 bytes)

3. BLS signatures: 48 bytes, aggregatable, easy threshold

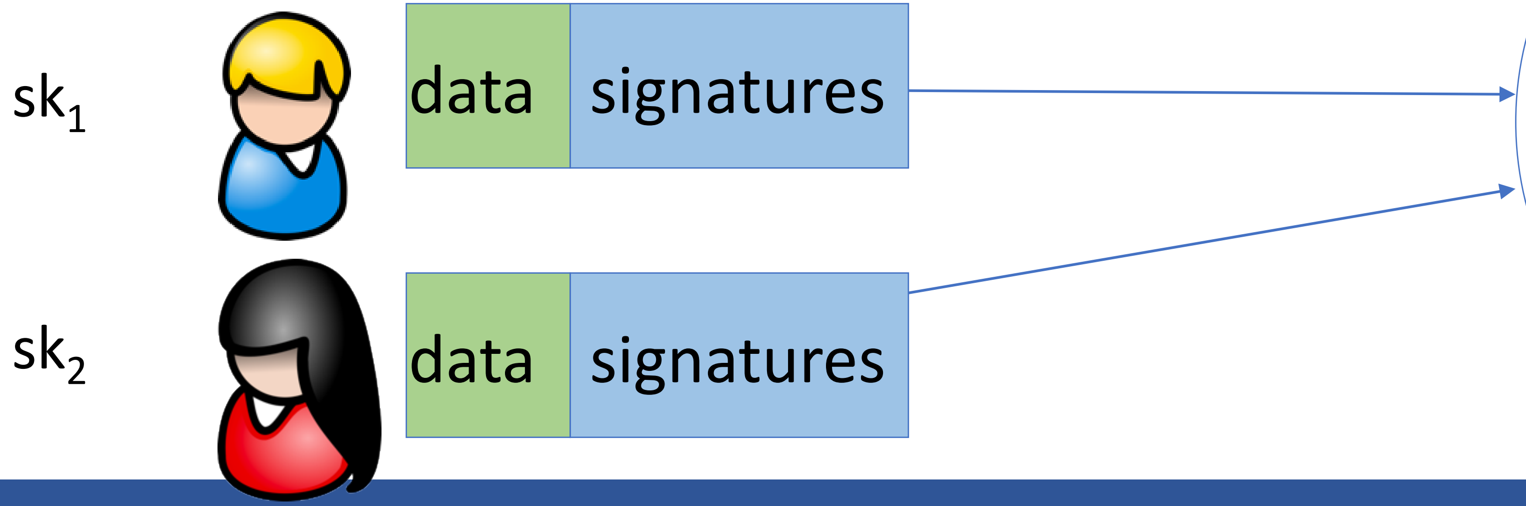
   (Ethereum 2.0, Chia, Dfinity)

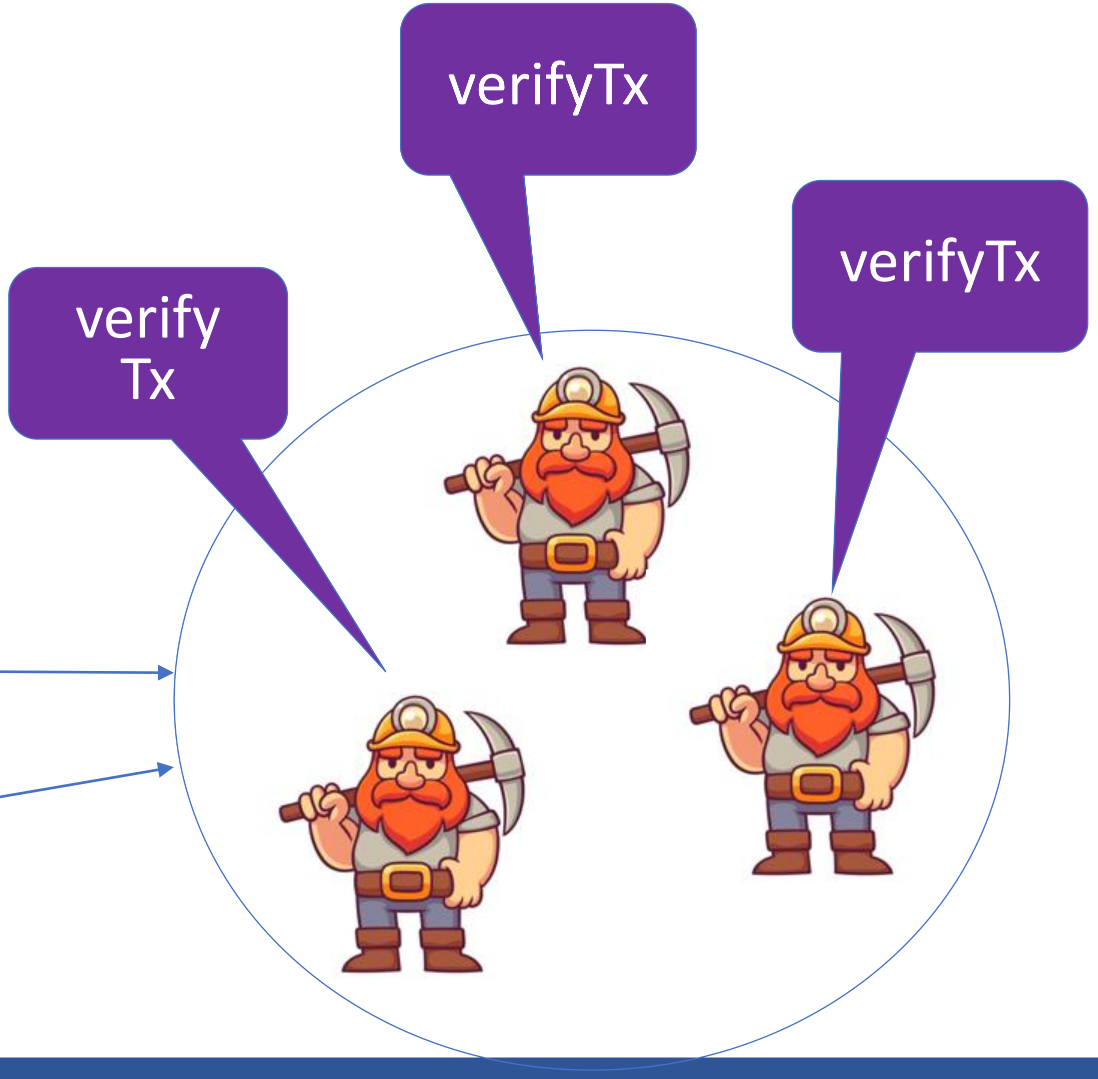
4. Post-quantum signatures: long (≥600 bytes)

# Signatures on the blockchain

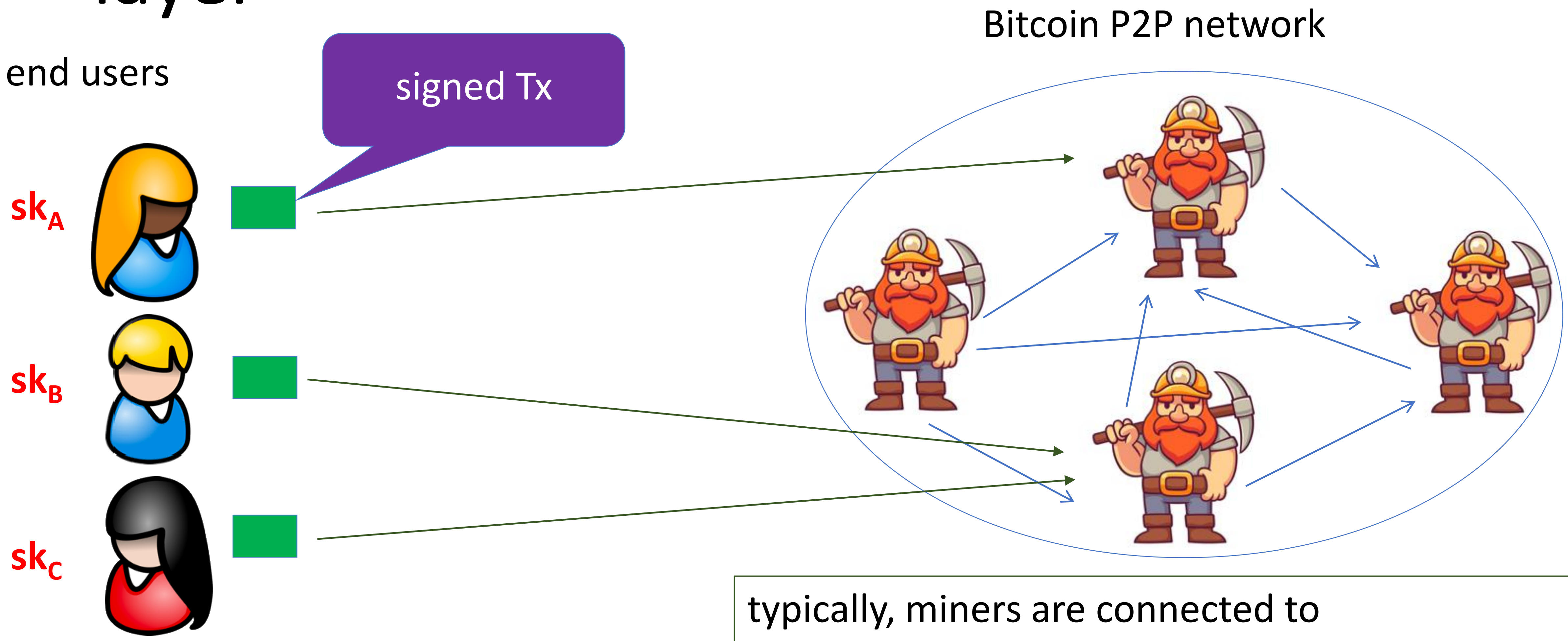
Signatures are used everywhere:

- ensure Tx authorization,
- governance votes,
- consensus protocol votes.



# First: overview of the Bitcoin consensus layer



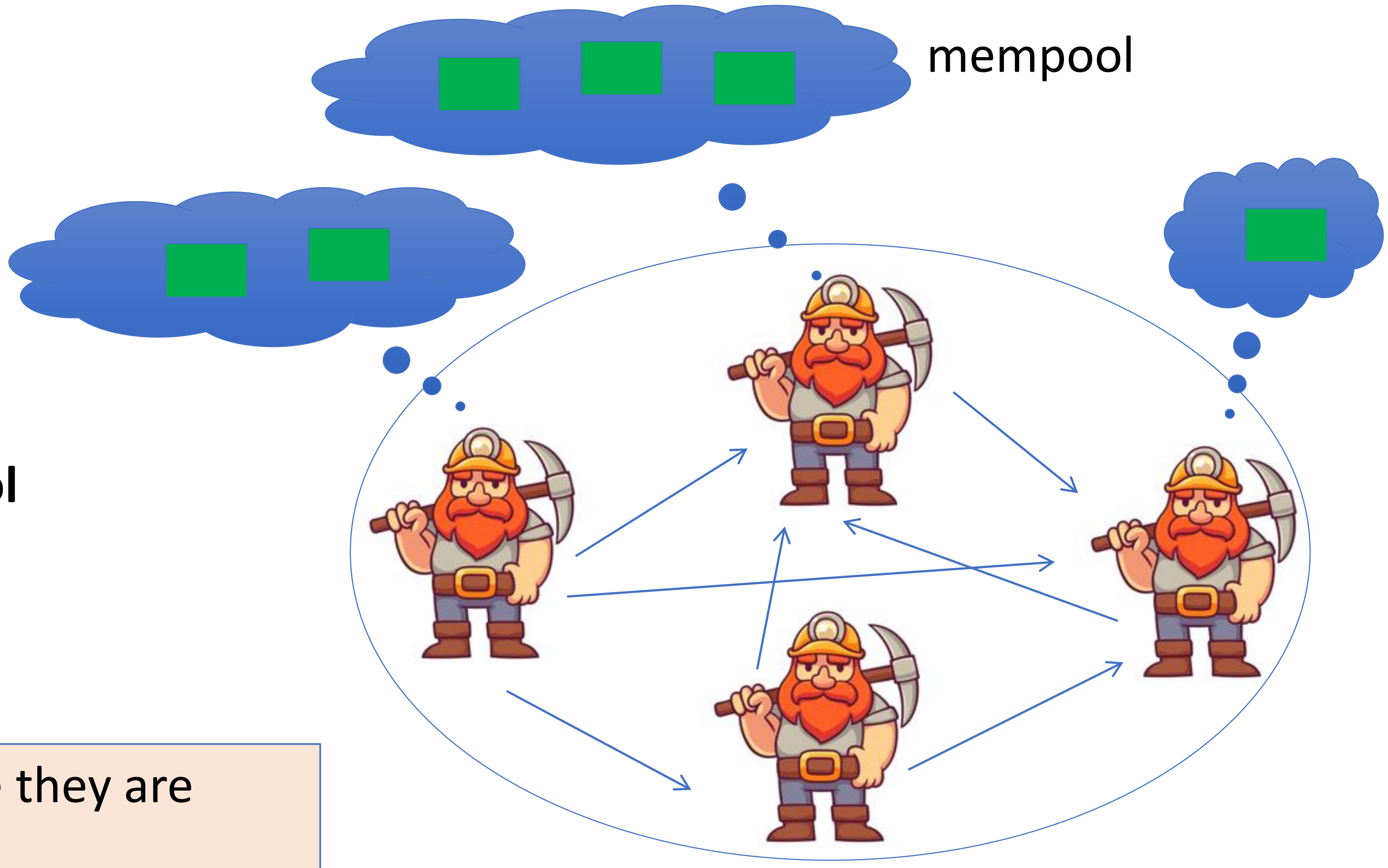

typically, miners are connected to many other peers (anyone can join)

# First: overview of the Bitcoin consensus layer

miners broadcast received Tx
to the P2P network

every miner:
  validates received Tx and
  stores them in its **mempool**
  (unconfirmed Tx)

note: miners see all Tx before they are posted on chain



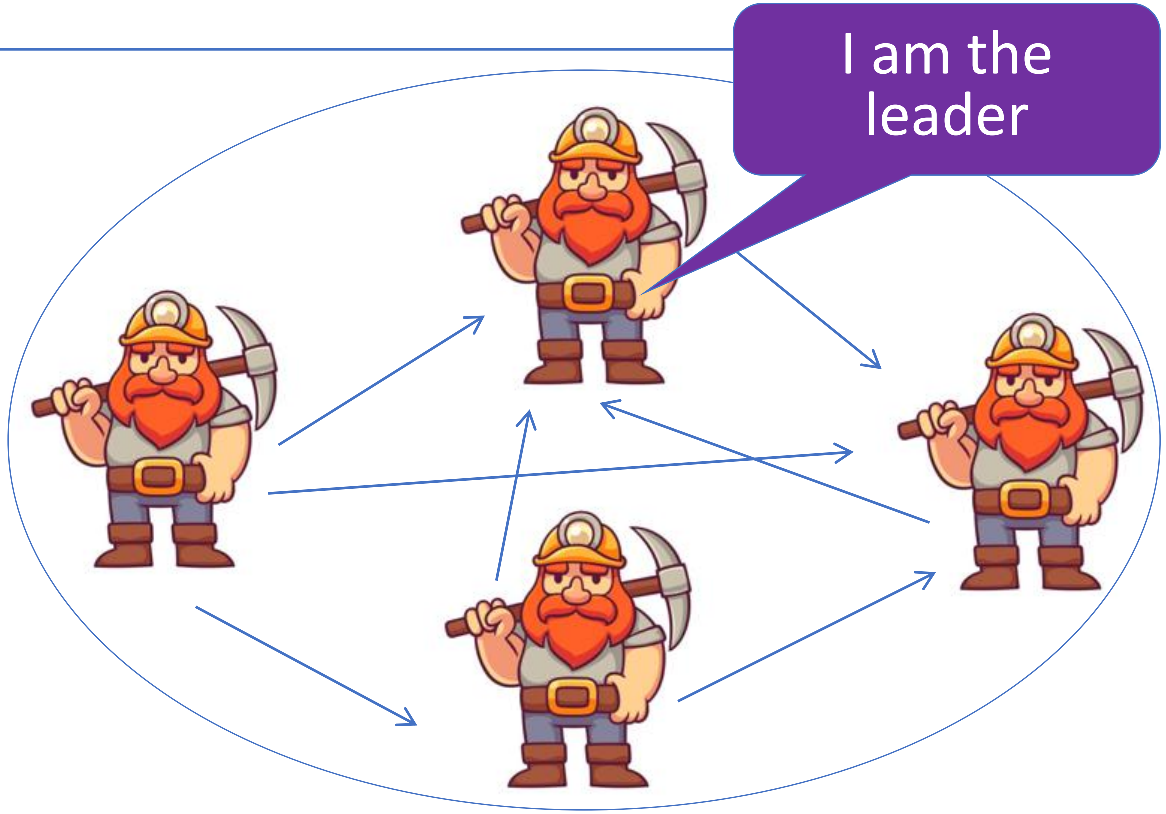

Bitcoin P2P network

# First: overview of the Bitcoin consensus layer

blockchain

Every ≈**10 minutes**:

- Each miner creates a candidate block from Tx in its mempool
- a “random” miner is selected (how?), and broadcasts its block to P2P network
- all miners validate new block




Bitcoin P2P network

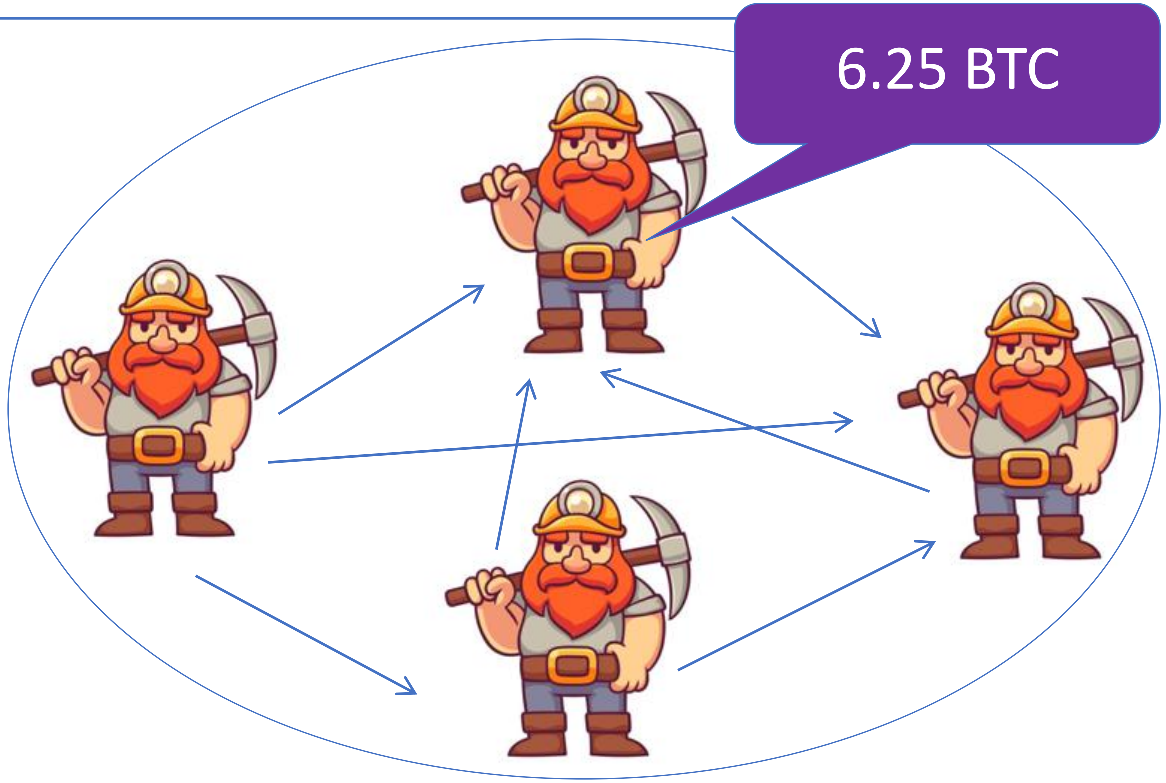
# First: overview of the Bitcoin consensus layer

blockchain

Selected miner is paid 6.25 BTC
in **coinbase Tx** (first Tx in the block)

- only way new BTC is created
- block reward halves every four years

  ⇒ max 21M BTC (currently 19.1M BTC)




note: miner chooses order of Tx in block

# Properties (very informal)

**Safety / Persistence**:

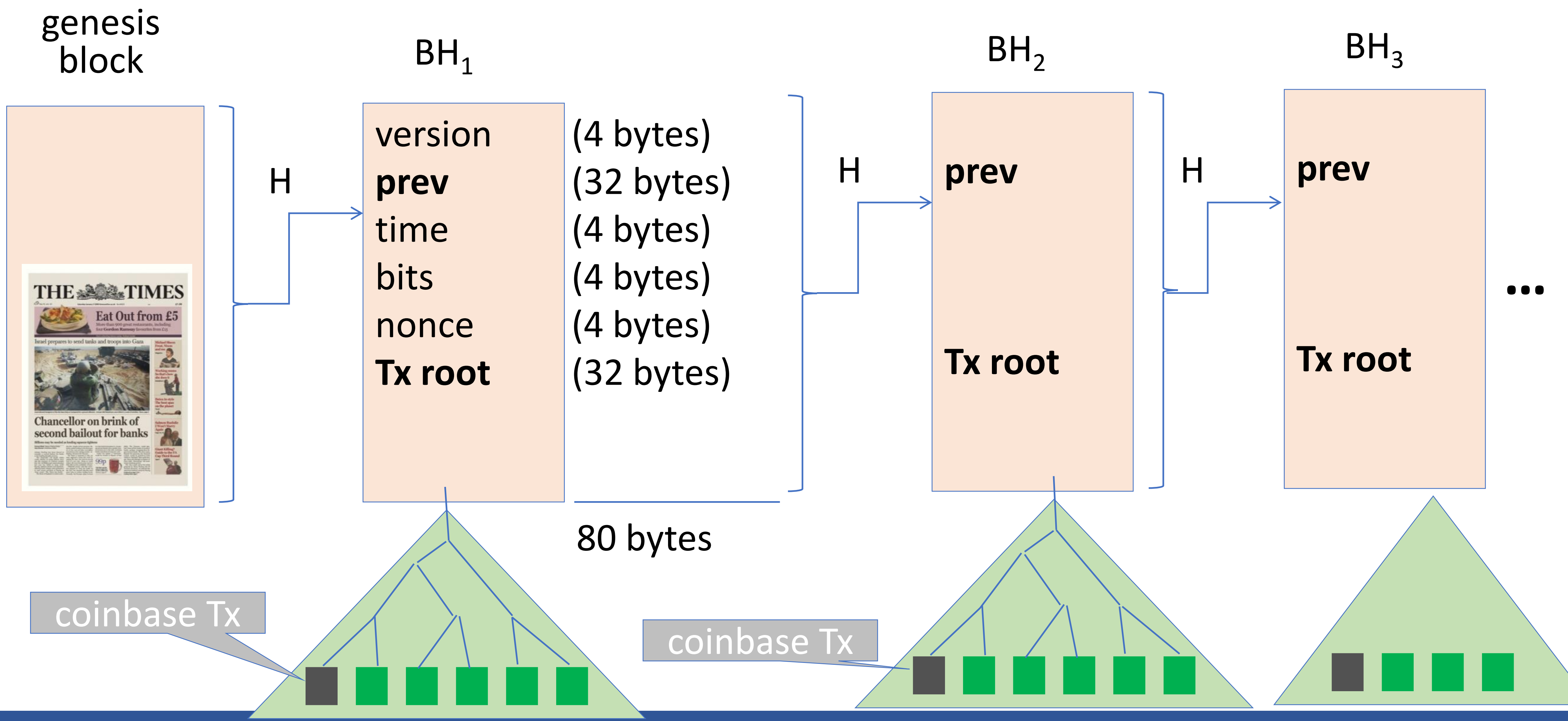
- to remove a block, need to convince 51% of mining power *

**Liveness**:

- to block a Tx from being posted, need to convince 51% of mining power **

(some sub 50% censorship attacks, such as feather forks)

# Bitcoin blockchain:  a sequence of block headers, 80 bytes each



Bitcoin blockchain: a sequence of block headers, 80 bytes each

**time**: time miner assembled the block. Self reported.
(block rejected if too far in past or future)

**bits**: proof of work difficulty
**nonce**: proof of work solution

for choosing a leader (next week)

**Merkle tree**: payer can give a short proof that Tx is in the block

new block every ≈10 minutes.

# An example (Sep. 2020)

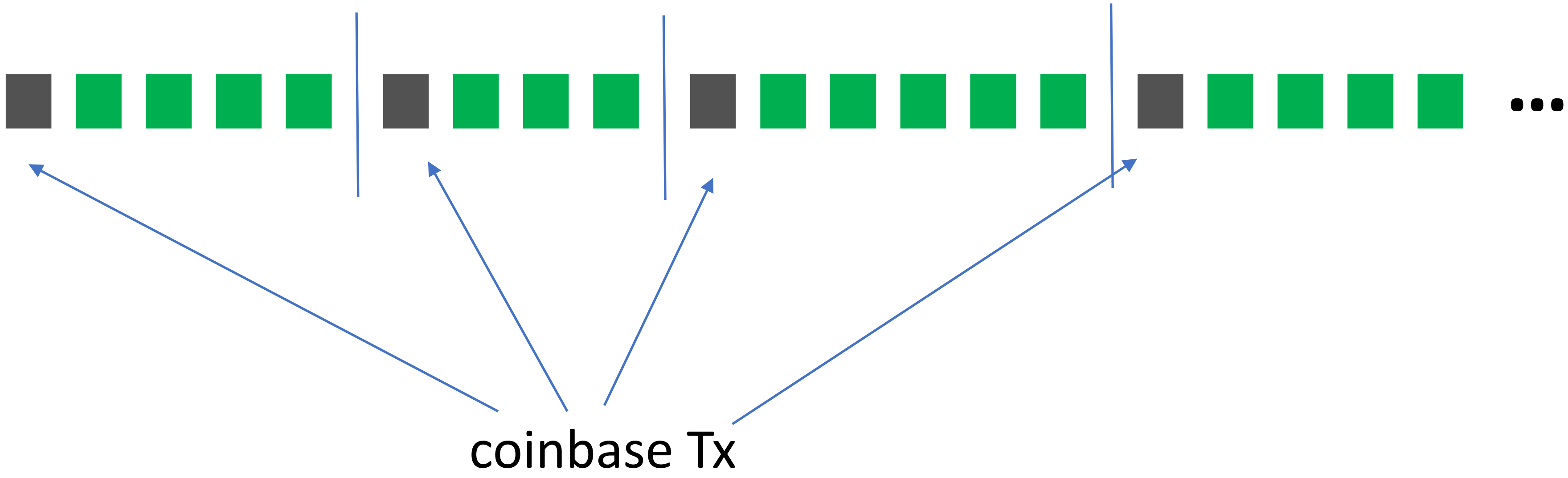
Tx data

| Height | Mined | Miner | Size | #Tx |
|---|---|---|---|---|
| 648494 | 17 minutes | Unknown | 1,308,663 bytes | 1855 |
| 648493 | 20 minutes | SlushPool | 1,317,436 bytes | 2826 |
| 648492 | 59 minutes | Unknown | 1,186,609 bytes | 1128 |
| 648491 | 1 hour | Unknown | 1,310,554 bytes | 2774 |
| 648490 | 1 hour | Unknown | 1,145,491 bytes | 2075 |
| 648489 | 1 hour | Poolin | 1,359,224 bytes | 2622 |

# Block 648493

| | | |
|---|---|---|
| Timestamp | 2020-09-15 17:25 | |
| Height | 648493 | |
| Miner | SlushPool | (from coinbase Tx) |
| Number of Transactions | 2,826 | |
| Difficulty (D) | 17,345,997,805,929.09 | (adjusts every two weeks) |
| Merkle root | 350cbb917c918774c93e945b960a2b3ac1c8d448c2e67839223bbcf595baff89 | |
| Transaction Volume | 11256.14250596 BTC | |
| Block Reward | 6.25000000 BTC | |
| Fee Reward | 0.89047154 BTC | (Tx fees given to miner in coinbase Tx) |

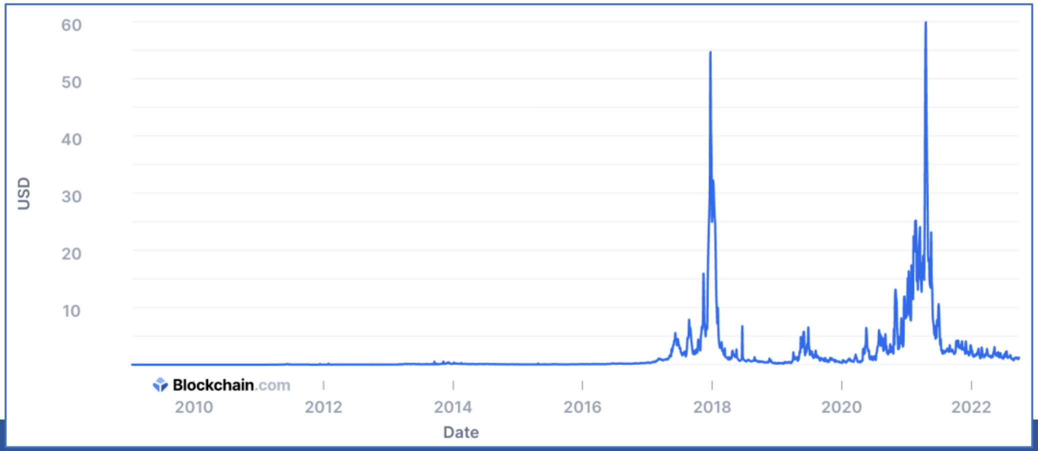
# View the blockchain as a sequence of Tx (append-only)



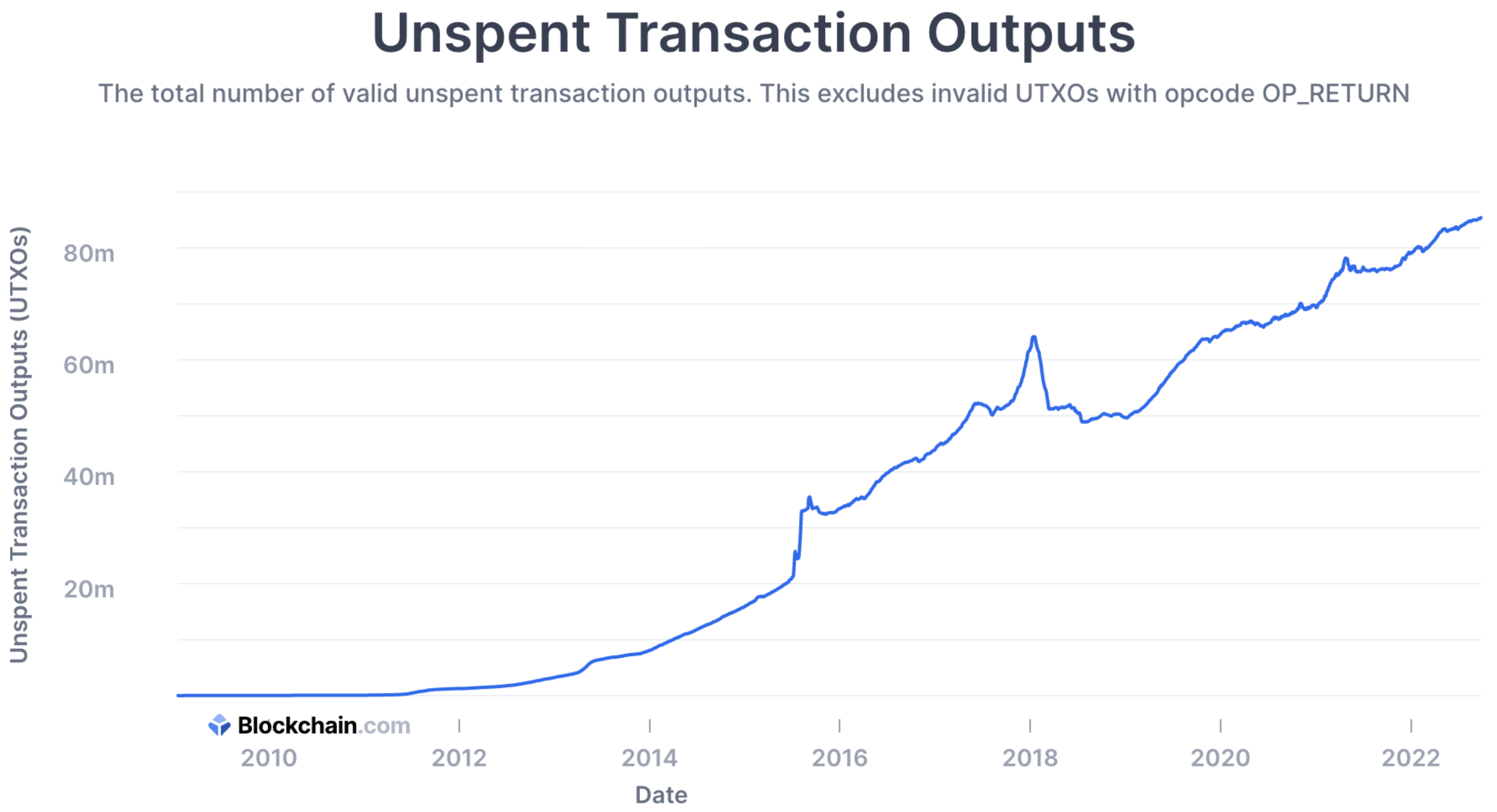
# Tx fees

Bitcoin average Tx fees in USD  (last 60 days, sep. 2022)


Bitcoin average Tx fees in USD  (all time)



# All value in Bitcoin is held in UTXOs




Sep. 2022: miners need to store ≈85M UTXOs in memory

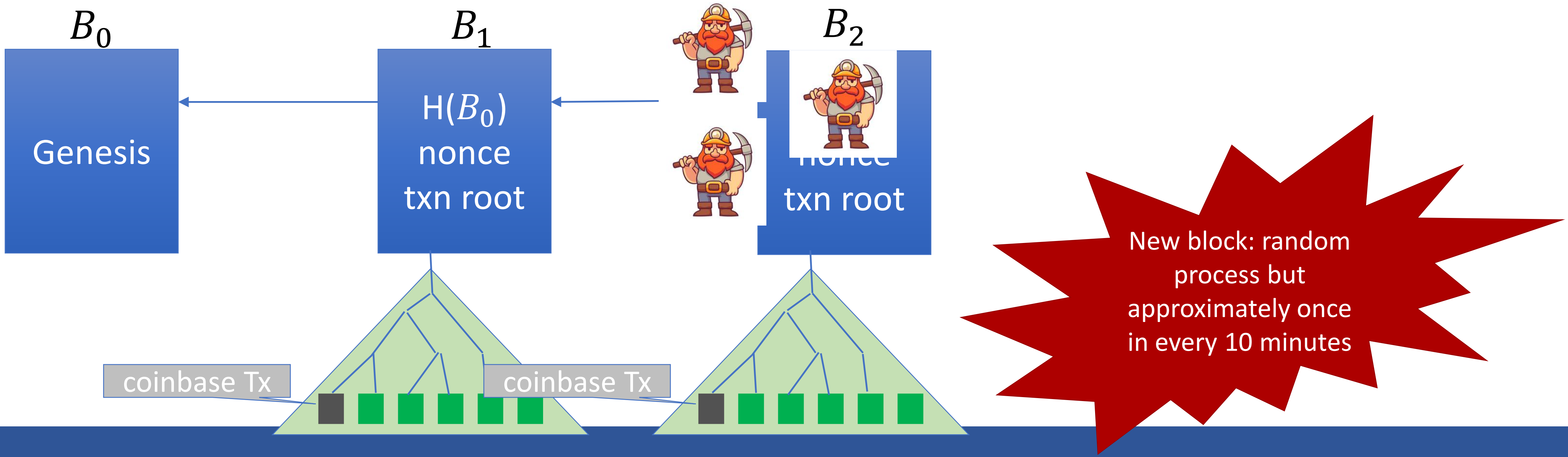
# Bitcoin: Mining

To mine a new block, a miner must find *nonce* such that

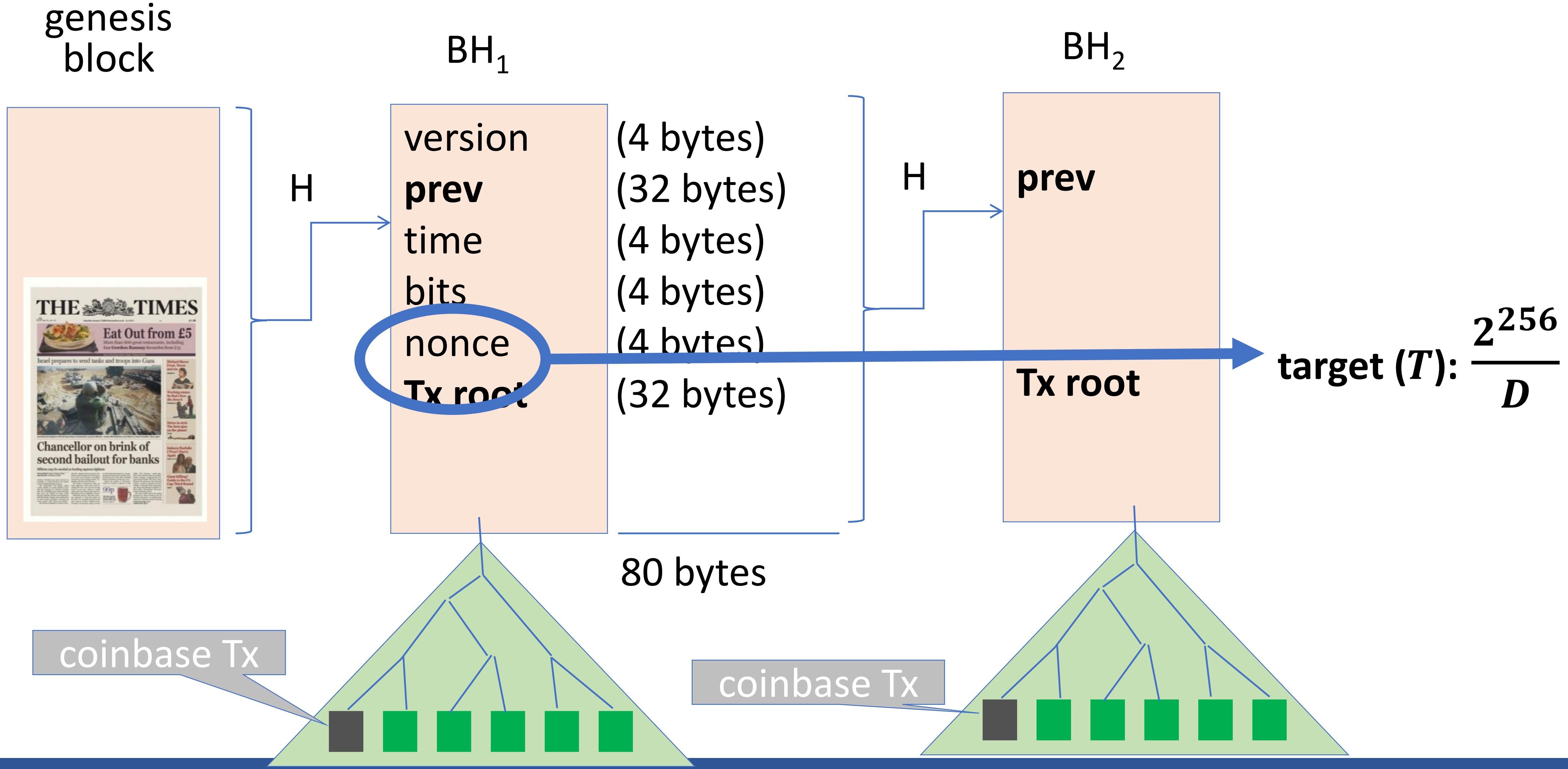
$$H\left(h_{prev}, txn\,root, nonce\right) < \text{Target} = \frac{2^{256}}{D}$$

**Difficulty:** How many nonces on average miners try until finding a block?

Each miner tries different nonces until one of them finds a nonce that satisfies the above equation.



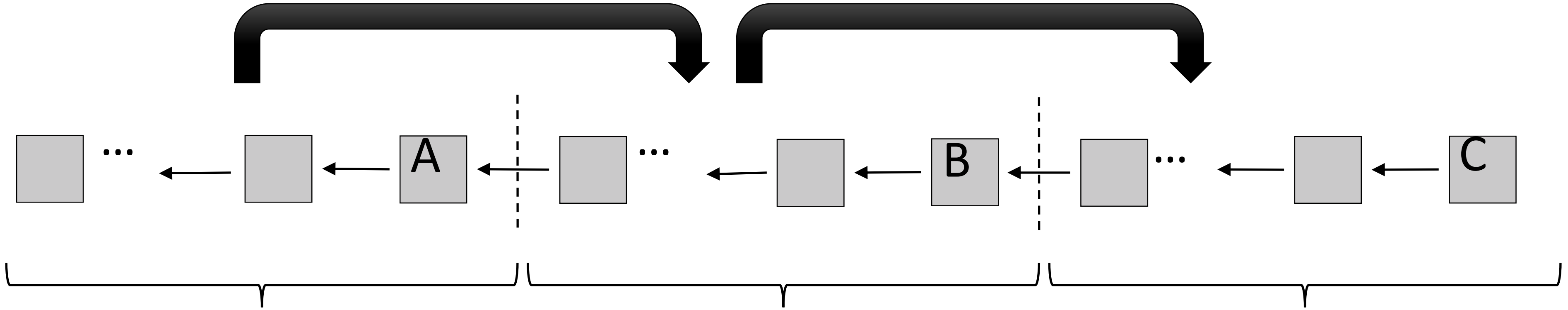
# Bitcoin: Block Headers



# Bitcoin: Difficulty Adjustment

New target: $T_2 = T_1 \frac{t_1}{2016 \times 10\ mins}$

New target: $T_3 = T_2 \frac{t_2}{2016 \times 10\ mins}$




2016 blocks
Time it took to mine: $t_1$(min)
Target: $T_1$

2016 blocks
Time it took to mine: $t_2$(min)
Target: $T_2$

2016 blocks
Time it took to mine: $t_3$(min)
Target: $T_3$

New target is not allowed to be more than 4x old target.
New target is not allowed to be less than ¼ x old target.

The Bitcoin Backbone Protocol with Chains of Variable Difficulty (2016)

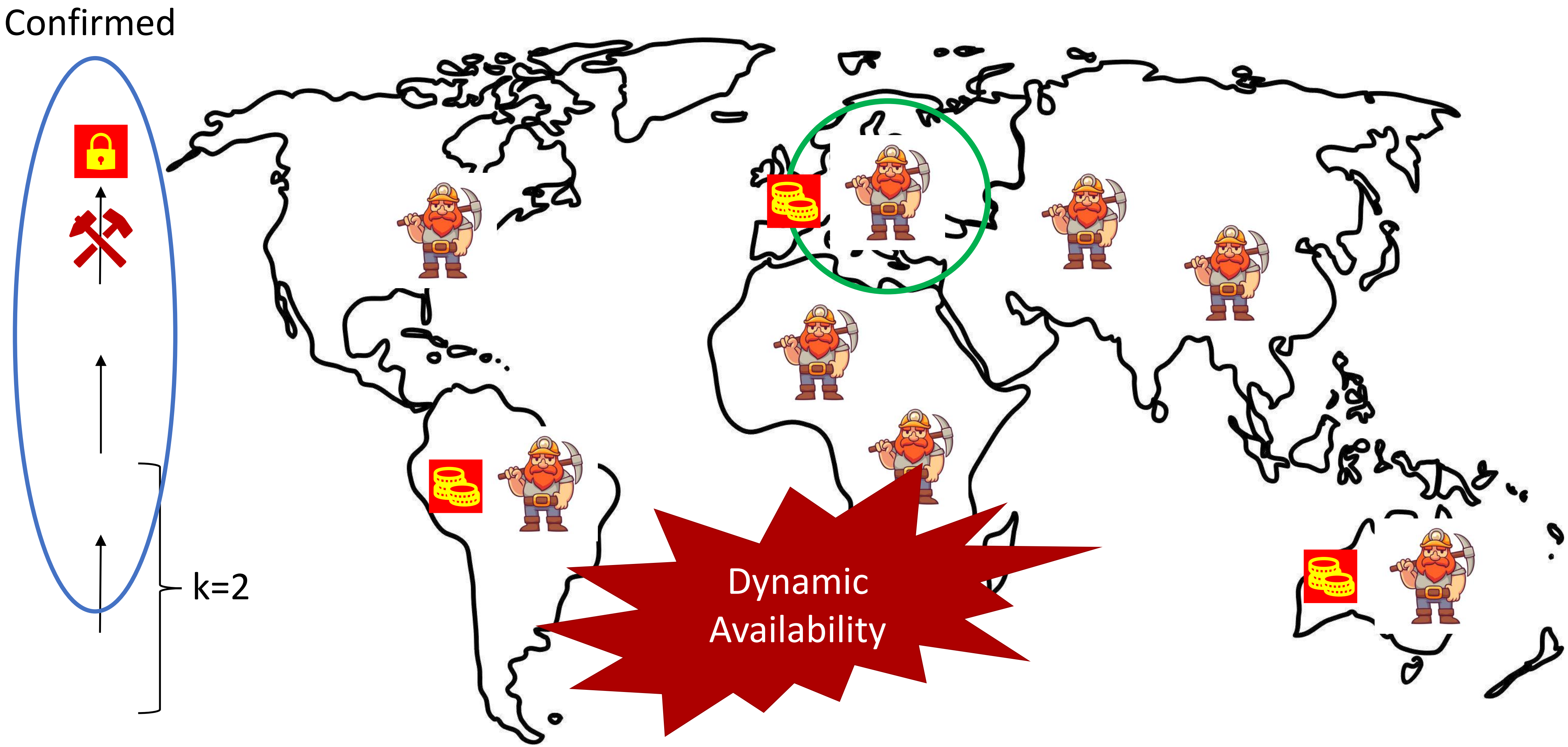
# Nakamoto Consensus

Chain with the highest difficulty!

Bitcoin uses **Nakamoto consensus**:

- **Fork-choice / proposal rule:** At any given time, each honest miner attempts to extend (i.e., mines on the tip of) the <u>heaviest</u> (longest for us) chain in its view (Ties broken adversarially).
- **Confirmation rule:** Each miner confirms the block (along with its prefix) that is *k*-deep within the longest chain in its view.
  - In practice, $k = 6$.
  - Miners and clients accept the transactions in the latest confirmed block and its prefix <u>as their log</u>.
  - Note that *confirmation* is different from *finalization*.
- **Leader selection rule:** Proof-of-Work.

# Nakamoto Consensus



# Consensus in the Internet Setting

Characterized by *open participation*:

- Adversary can create many Sybil nodes to take over the protocol.
- Honest participants can come and go at will.

**Goals:**

- Limit adversary's participation.
  - **Sybil resistance (e.g., Proof-of-Work)!**
- Maintain availability (liveness) of the protocol against changing participation by the honest nodes.
  - **Dynamic availability!**

# Security

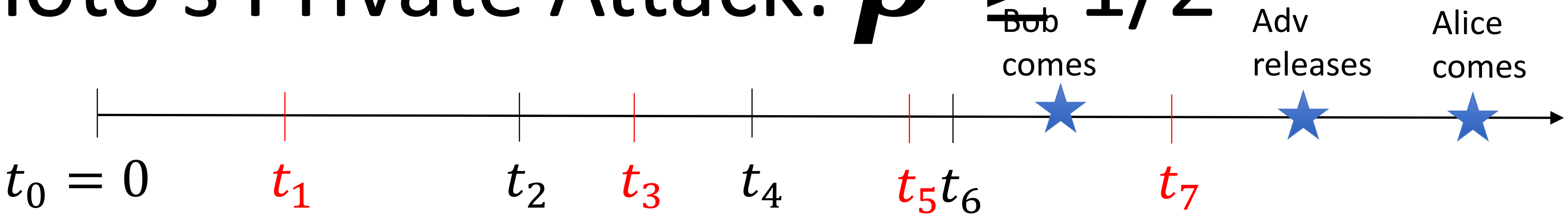
Can we show that Bitcoin is secure under synchrony against a Byzantine adversary?

What would be the best possible resilience?

$$\beta < 1/2?$$

**Fraction of the mining power controlled by the adversary.**

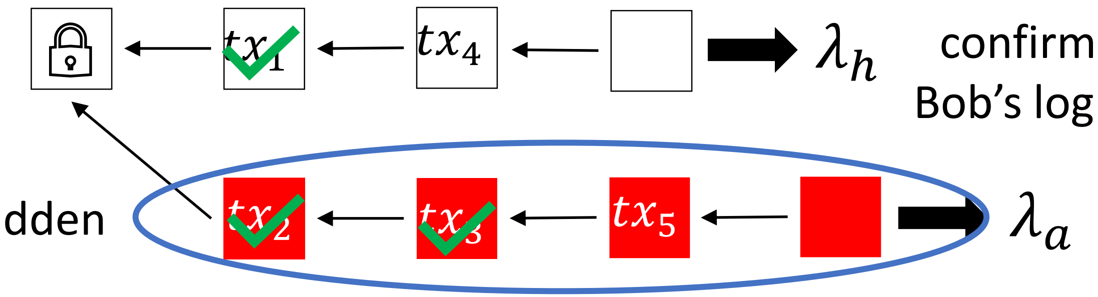
# Nakamoto's Private Attack: $\boldsymbol{\beta} \geq 1/2$




**$tx_1$ got 'reorged':** It was part of the longest chain before but not anymore!!

k deep confirmation rule (k=3 in our example)

Hidden




Bob sees $tx_1$ as confirmed.
Bob's log: $tx_1$

Now, Alice comes, in her view:
The red chain is the longest chain.
$tx_1$ is not confirmed!
Alice's log: $tx_2 tx_3$

Private attack succeeds!

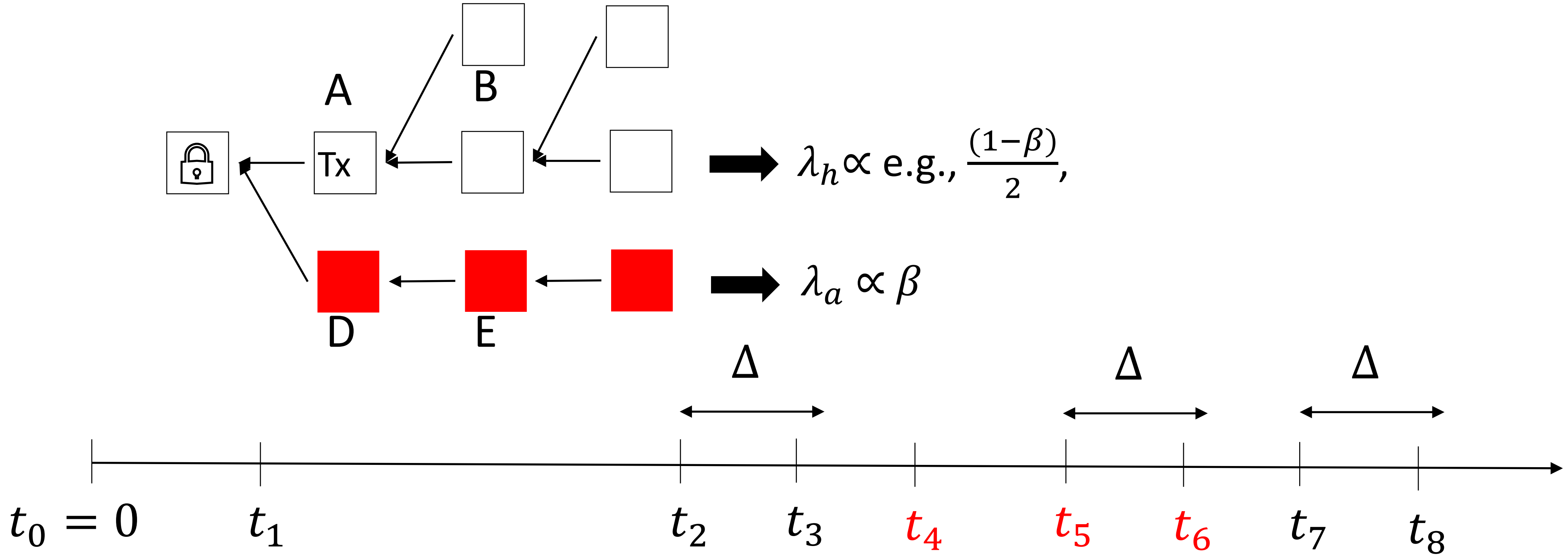
Private attack (mostly) succeeds if $\lambda_a \geq \lambda_h$, i.e., if $\beta \geq 1 - \beta$, i.e., if $\beta \geq \frac{1}{2}$.

Private attack (mostly) fails if $\lambda_a < \lambda_h$, i.e., if $\beta < 1 - \beta$, i.e., if $\beta < \frac{1}{2}$.

Can another attack succeed?

A Peer-to-Peer Electronic Cash System (2008)

# Forking



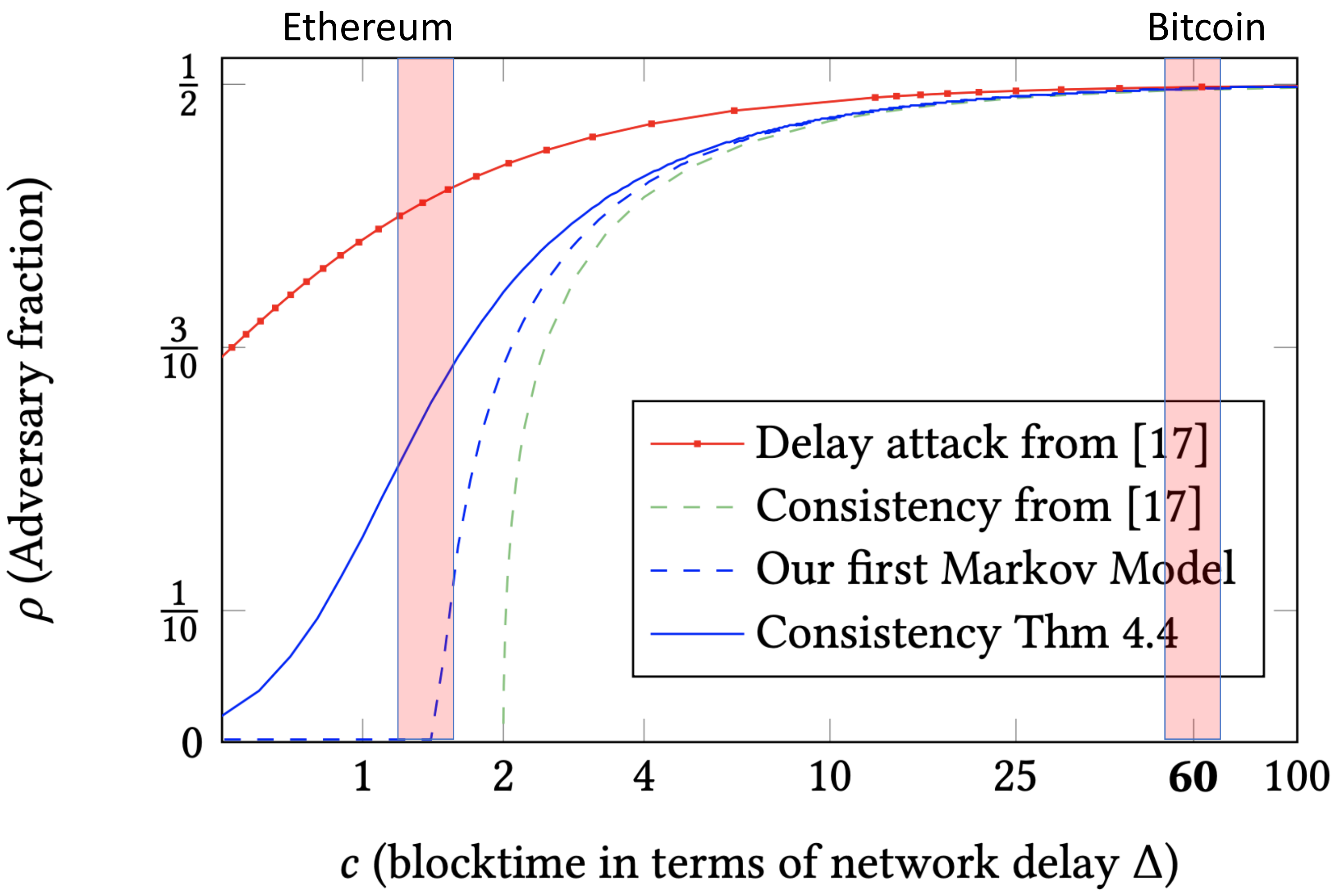

Multiple honest blocks at the same height due to network delay.

Adversary's chain grows at rate proportional to (shown by $\propto$) $\beta$!

Honest miners' chain grows at rate less than $1 - \beta$ because of forking!

Now, adversary succeeds if $\beta \geq \frac{(1-\beta)}{2}$, which implies $\beta \geq \frac{1}{3}$!!

# Security of Bitcoin against other attacks

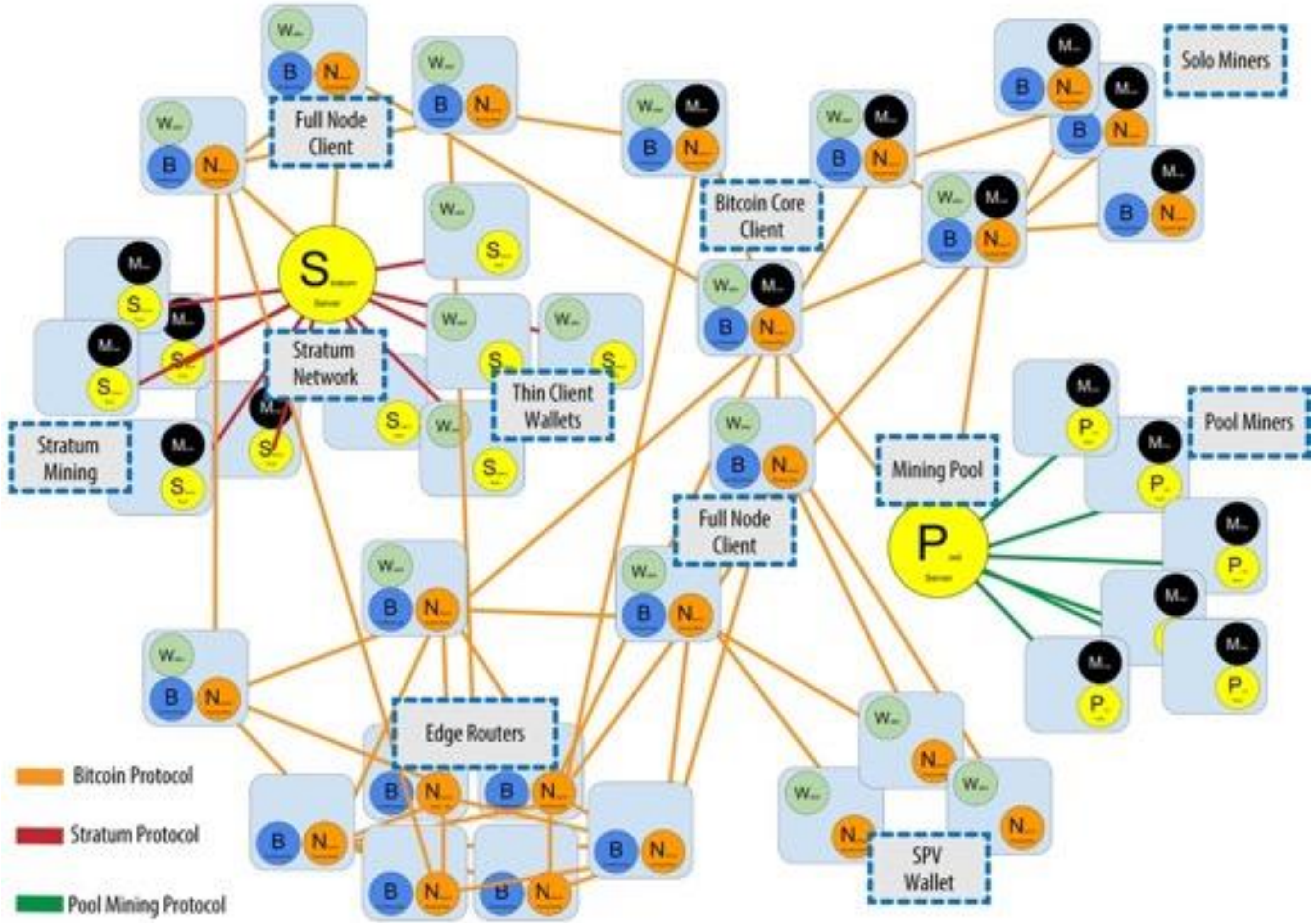



Kiffer, Lucianna, Rajmohan Rajaraman, and Abhi Shelat. "A better method to analyze blockchain consistency." *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 2018.

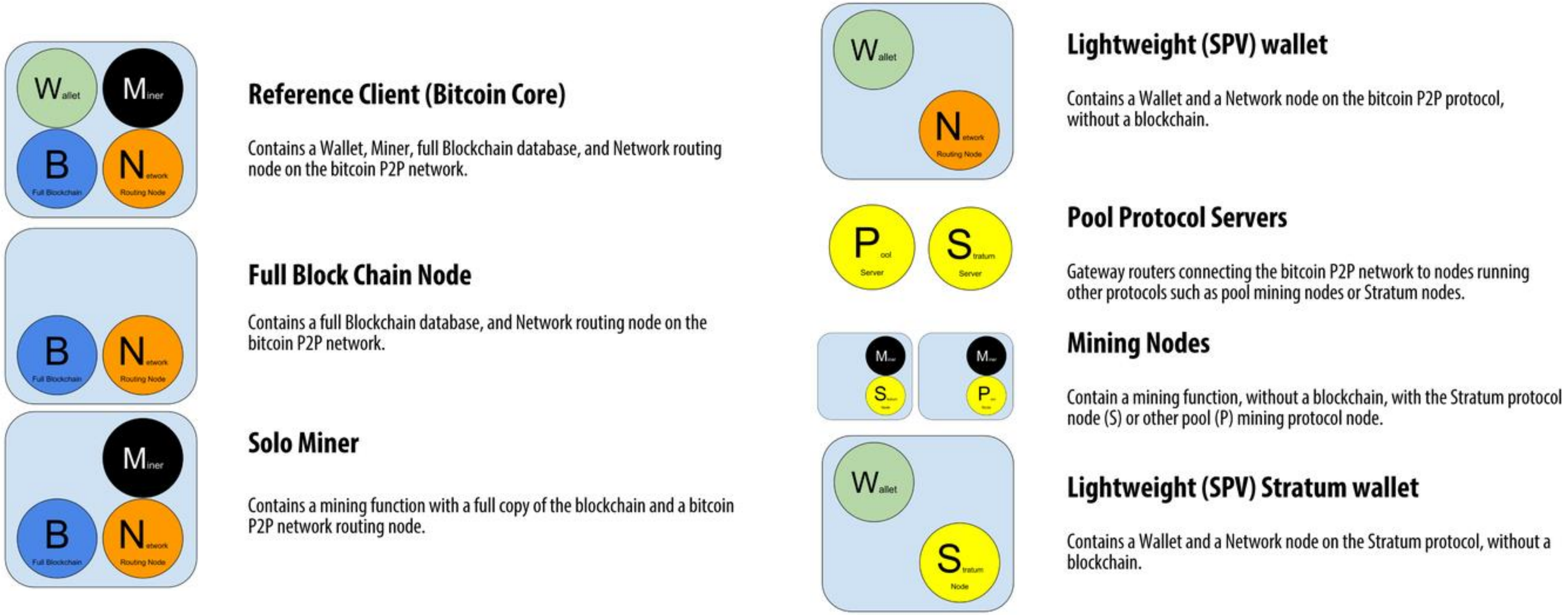
# Peer-to-Peer Communication Network

- Decentralized, permissionless peer-to-peer broadcast network used to announce new transactions and proposed blocks
- Requirements
  - low latency
    - 10 minute block creation time handles latency issues
  - robust against malicious miners
    - e.g., censor transactions
- Network topology and discovery
  - Bitcoin: 8 outgoing, 117 incoming connections
- Communication protocol
  - Flooding new blocks and pending transactions

# Extended Bitcoin network



# Various types of nodes in Bitcoin



# Large peer-to-peer network

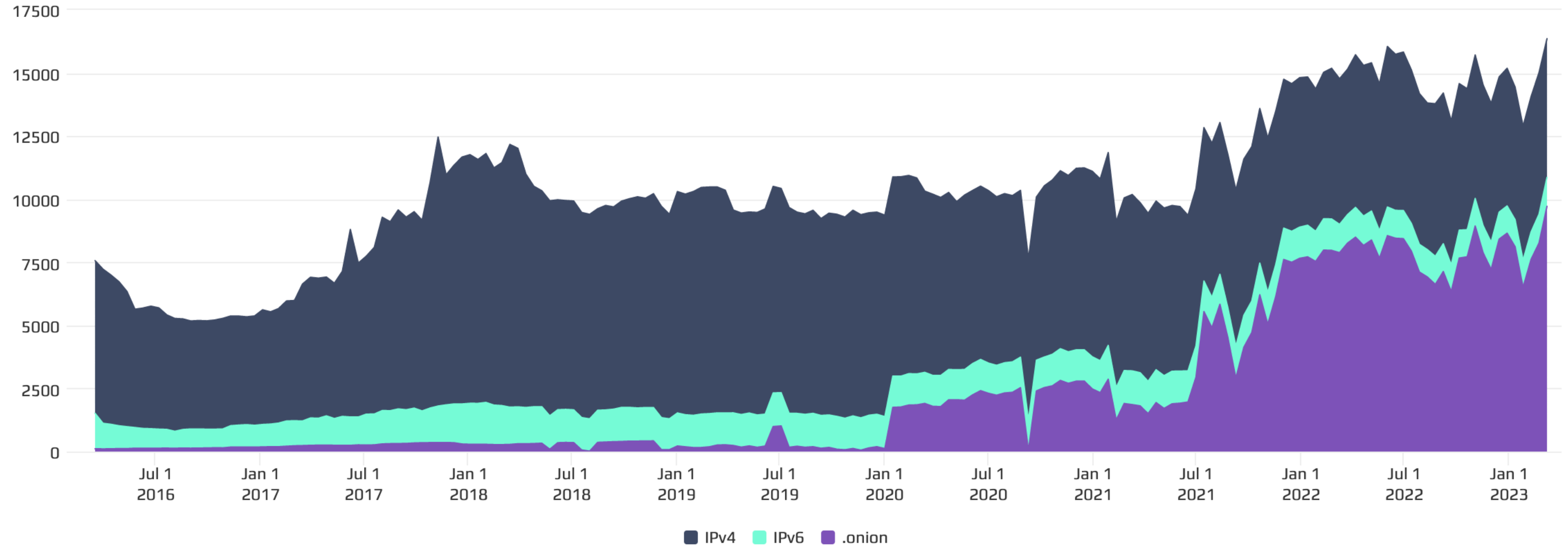
| 16359 | 10370 | 8793 ▲ 116.22% |
|---|---|---|
| Reachable nodes | Average | Since 7 years ago |

**NODES**

Chart shows the number of reachable Bitcoin nodes during the last 7 years. Series can be enabled or disabled from the legend to view the chart for specific networks.

24h 90d 1y 7y

Lo 5176 Hi 16359 Avg 10370 Last 16359 nodes



# Is Bitcoin the Endgame?

- Bitcoin provides Sybil resistance and dynamic availability.
- It can be made secure for any $\beta < \frac{1}{2}$.
- Is it the Endgame for consensus?

No!

- Bitcoin is secure only under synchrony but not under partial synchrony.
- It *confirms* blocks with an error probability as a function of *k*, not *finalizes* blocks.
- Energy?

# Dark Side of Bitcoin: Energy

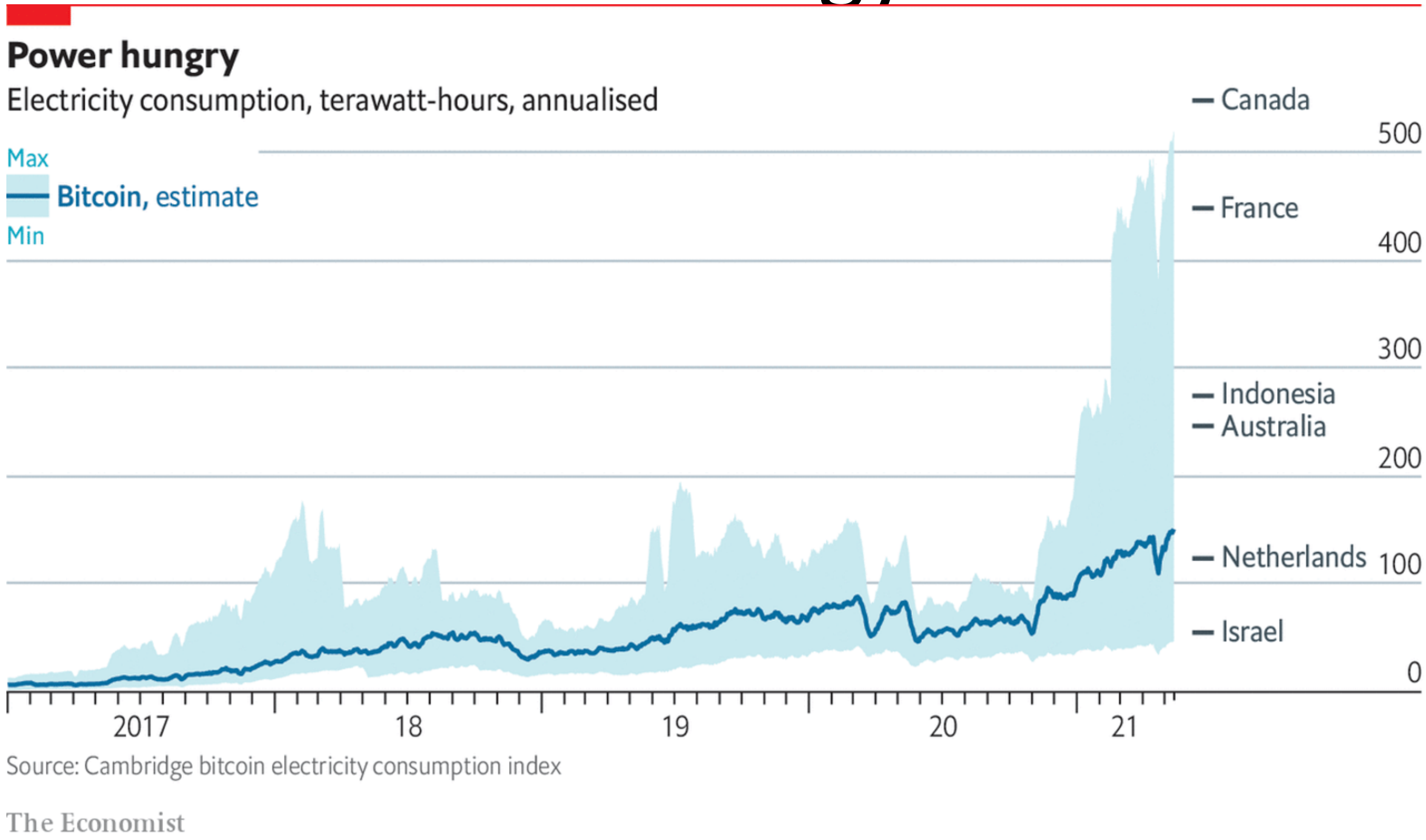
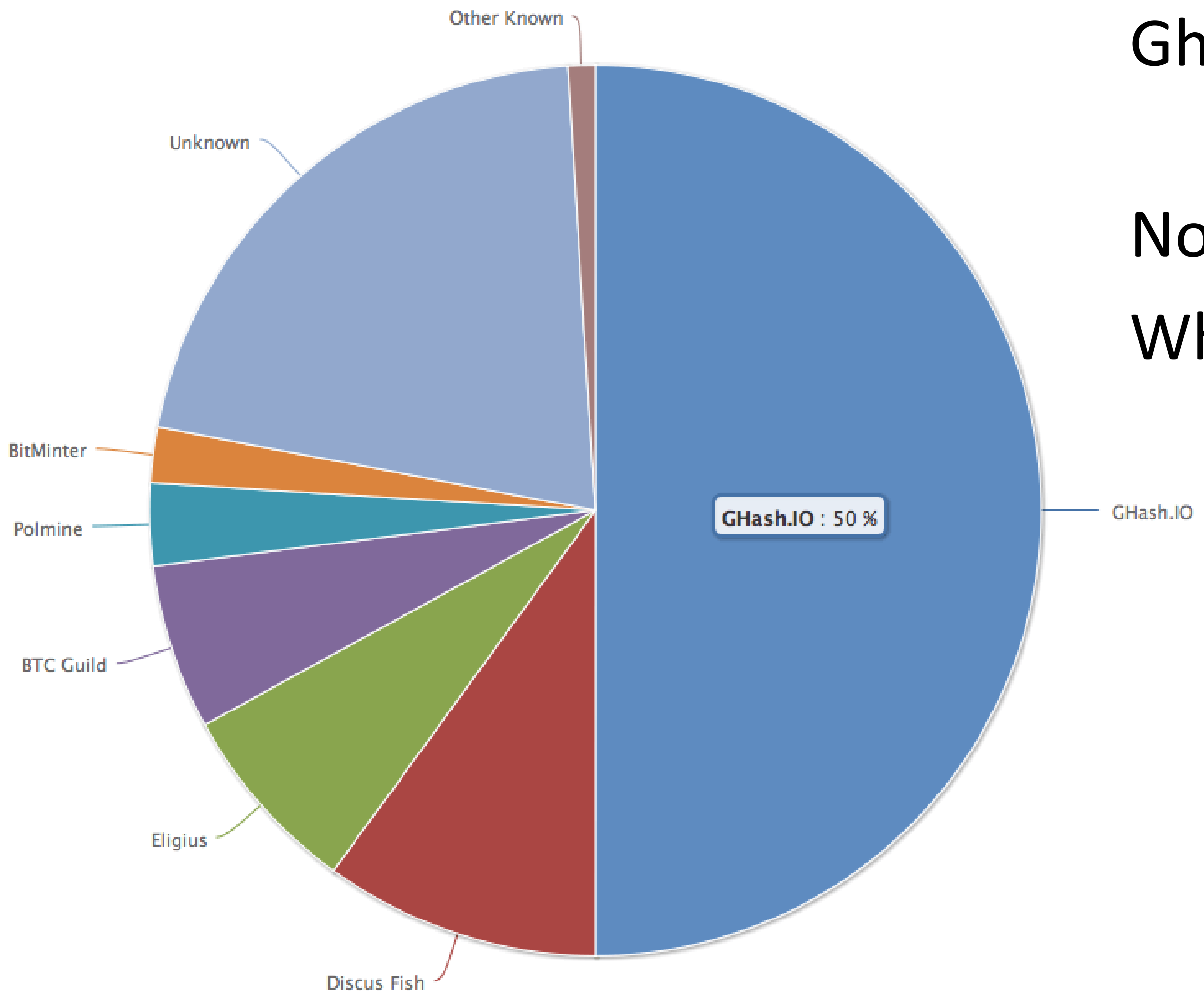



Photo taken from the article "As the price of bitcoin has climbed, so has its environmental cost" that appeared at The Economist on May 14th 2021.

# No Attacks on Bitcoin?




Ghash.IO had >50% in 2014

- Gave up mining power

No Selfish mining attacks?

Why are visible attacks not more frequent?

- Miners care about the Bitcoin price.
- Might not be rational to attack.
- No guarantees for the future.