

# Consensus OR “On How to Descroogify ScroogeCoin”

BASED ON SLIDES FROM:

1. [S. CHAKRABORTY, S. SURAL](#)
2. [P. VISWANATH](#)

# Permissioned vs Permissionless Blockchains



We look at  
permissionless

- Access Control
  - Permission required, known entities vs open access
- Decentralization
  - More centralized, smaller number of nodes central authority vs consensus mechanisms
- Transaction Verification
  - Verified by a small group of nodes, central authority vs consensus mechanisms
- Privacy and Transparency
  - Privacy and more control on who can view what vs high level of transparency since ledger and transaction data is public and pseudoanonymous
- Use Cases
  - Enterprises and organizations where some form of control on the network is required (e.g., supply chain management) vs when openness and censorship resistance are critical (e.g., cryptocurrency, tokenization, smart contracts)

# Permissionless Model

---

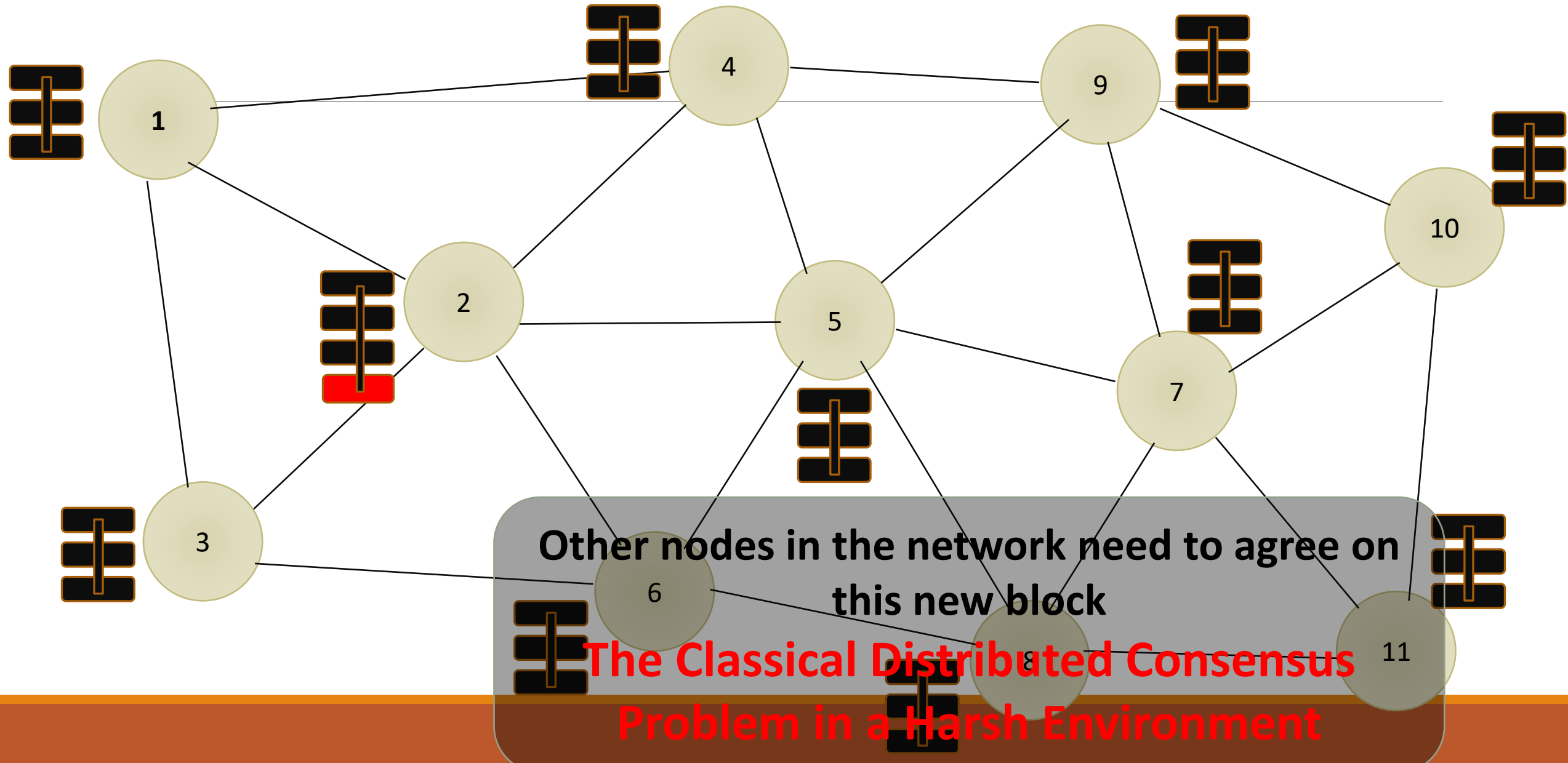
## Open network

- Anyone can join in the network and initiate transactions
- Participants are free to leave the network, and can join later again

## **Assumption: More than 50% of the participants are honest**

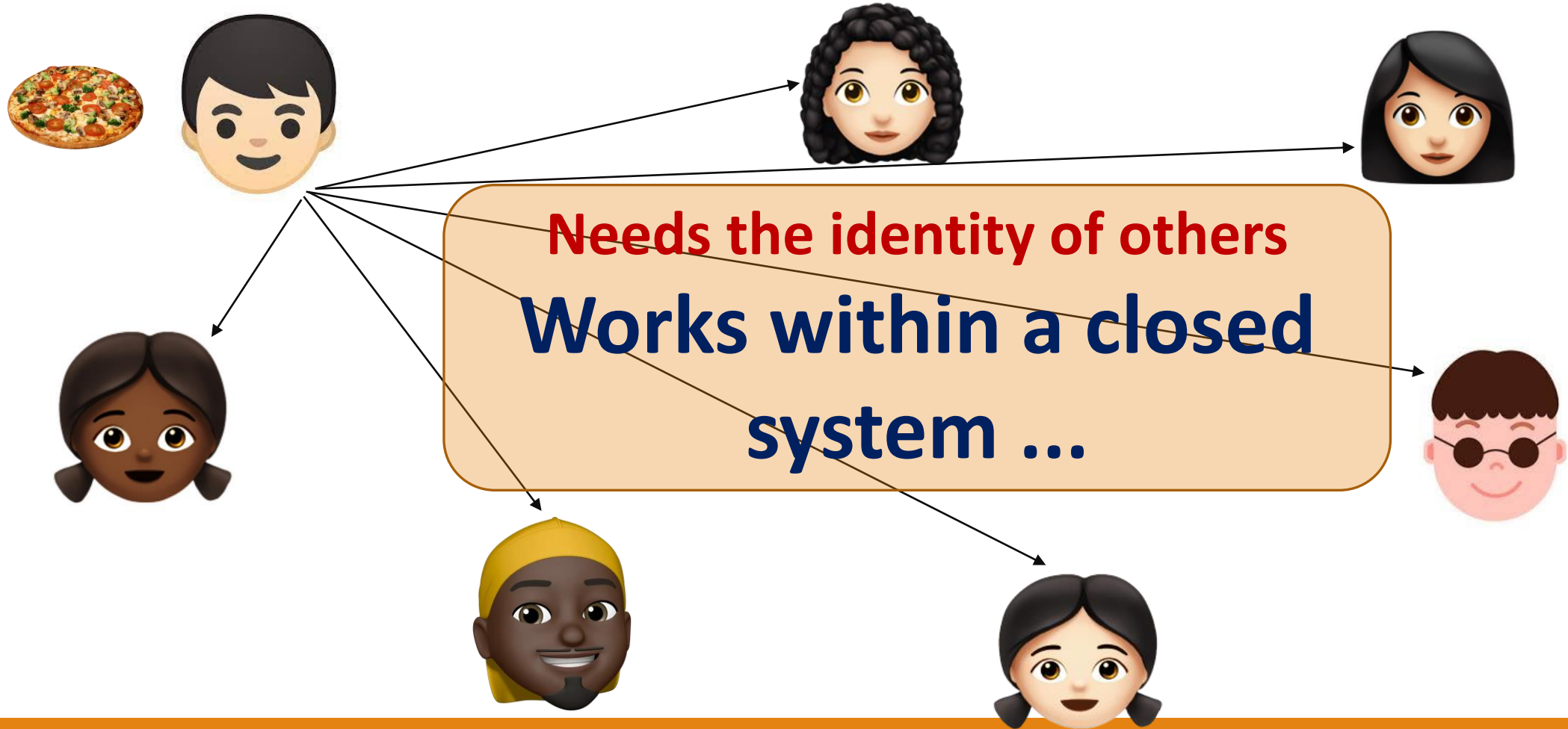
- A society cannot run if majority of its participants are dishonest !!!

# Our Core Problem



# What is the Issue with Classical Distributed Consensus?

---



# Core Questions

(slightly affected by Bitcoin...)

---

1. Who maintains the ledger of transactions?
  2. Who has authority over which transactions are valid?
  3. Who creates new bitcoins?
  4. Who determines how the rules of the system change?
  5. How do bitcoins acquire exchange value?
- 
- Technical
- Organizational - Economics

# Decentralization & Blockchains

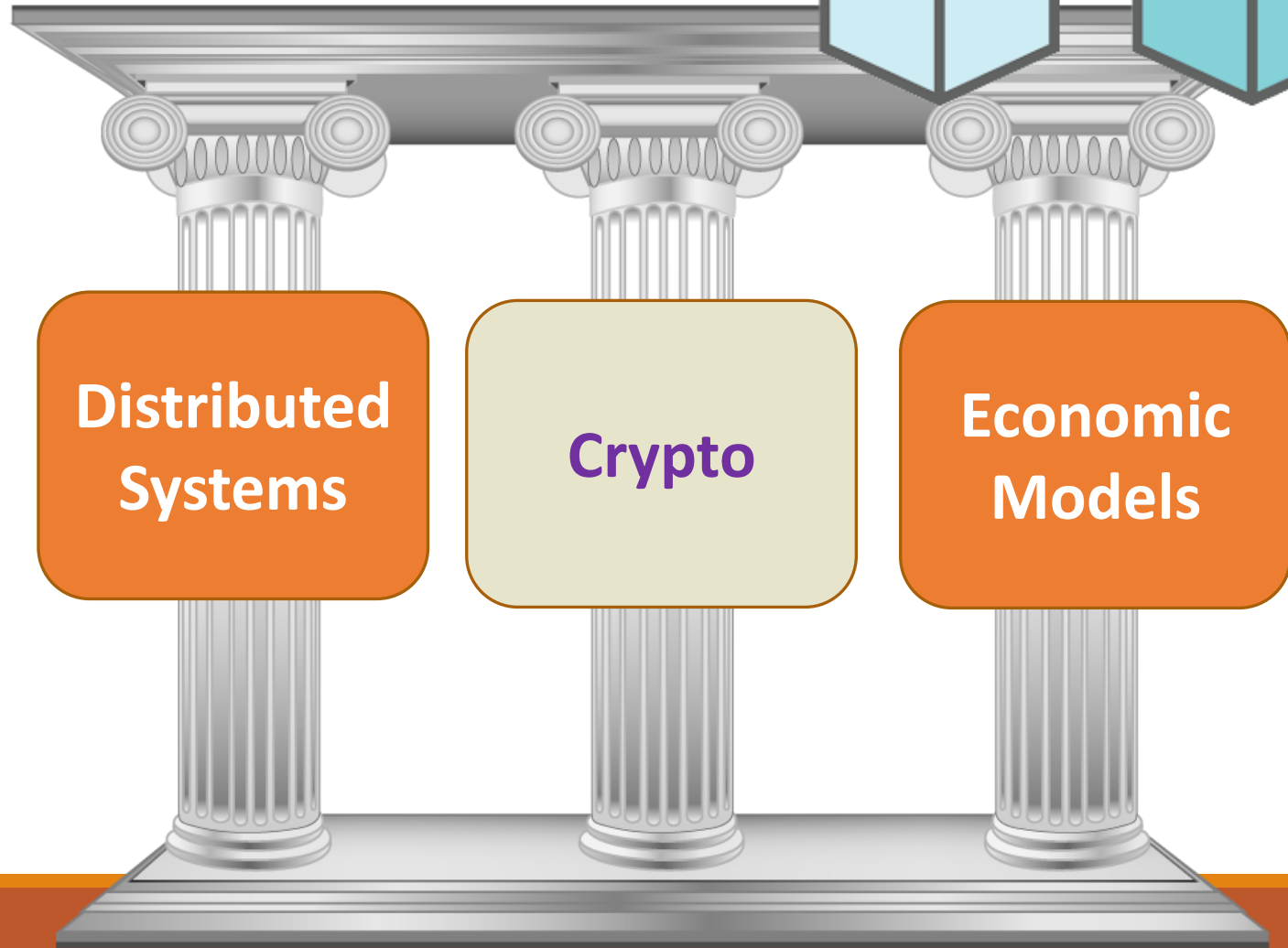


The Three Pillars

Distributed Systems

Crypto

Economic Models



# Distributed Consensus

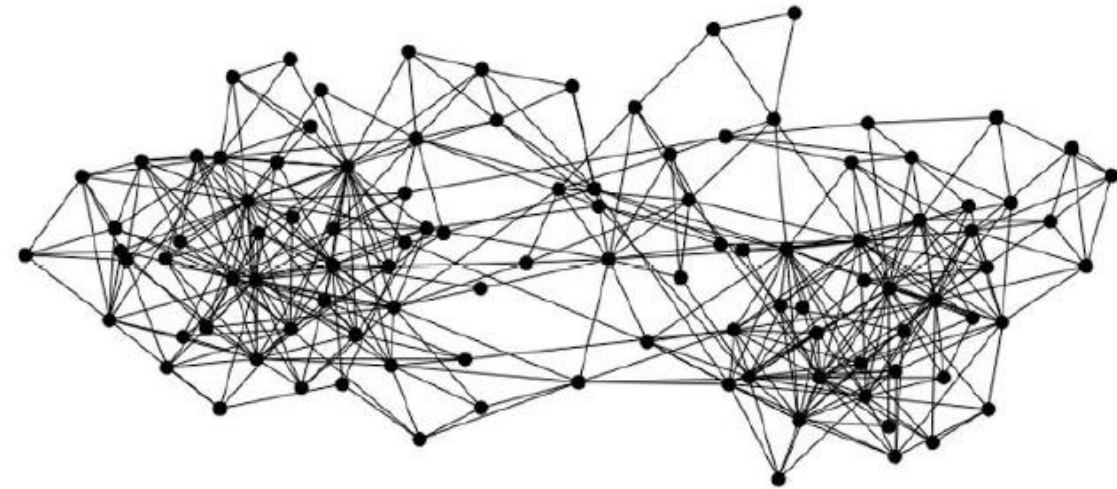
---

There are  $n$  nodes that each have an input value. Some of these nodes are faulty or malicious. A distributed consensus protocol has the following two properties:

- It must terminate with all honest nodes in agreement on the value (liveness)
- The value must have been generated by an honest node (safety)



signed by Alice  
Pay to  $pk_{\text{Bob}} : H( )$





# The Nodes Must Agree on:

---

1. The valid transactions that are broadcast in the network
2. The order of these valid transactions

subject to:

1. Nodes crash
2. Nodes may be malicious
3. The P2P is imperfect and not all nodes are directly connected to all
4. Latency issues (no global time) and network faults

# Consensus in a Permissionless Model - Challenges

Participants do not know others

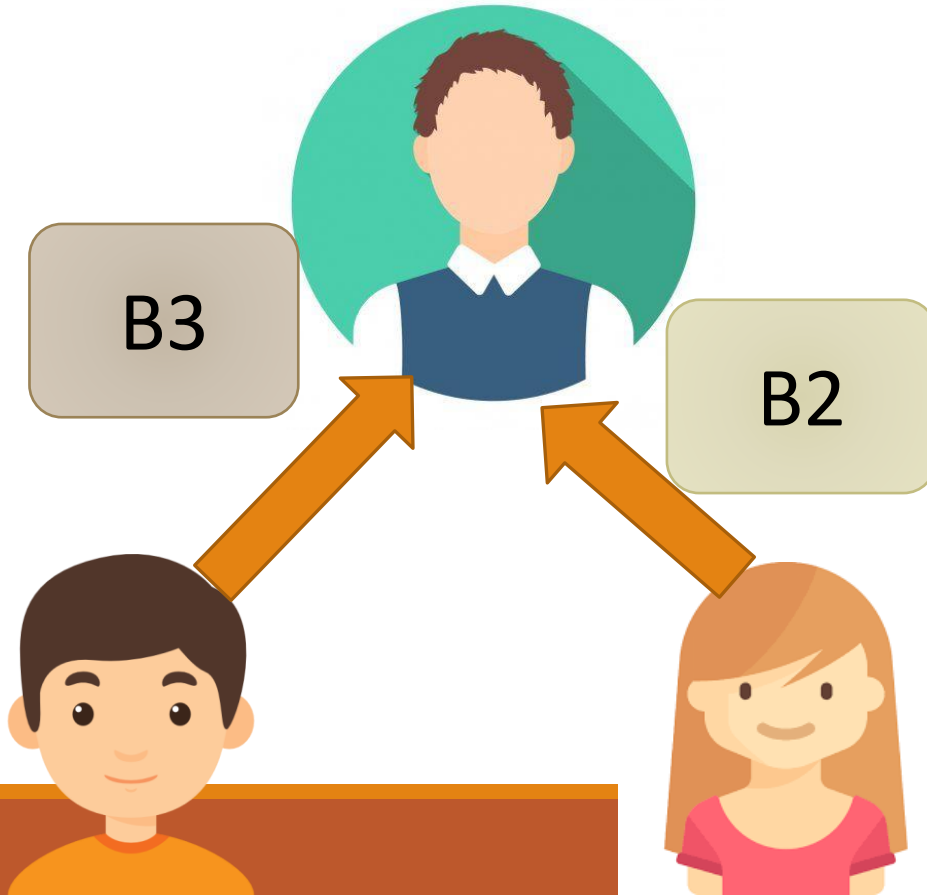
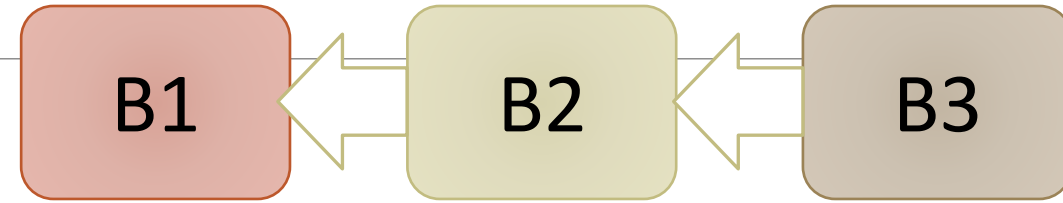
- Cannot use message passing !!

Anyone can propose a new block

- Who is going to add the next block in the blockchain?

The network is asynchronous

- We do not have any global clock
- Theoretically, a node may see the messages in different orders



# FLP Impossibility

---

## Synchronous vs Asynchronous Networks

- **Synchronous:** I am sure that I'll get the message within a predefined time threshold
- **Asynchronous:** I am not sure whether and when the message will arrive

## Failures in a network --

- **Crash Fault:** A node stops responding
- **Link Fault (or Network Fault):** A link fails to deliver the message
- **Byzantine Fault:** A node starts behaving maliciously

**The Impossibility Theorem:** Consensus is not possible in a perfect asynchronous network even with a single crash failure

# FLP Impossibility

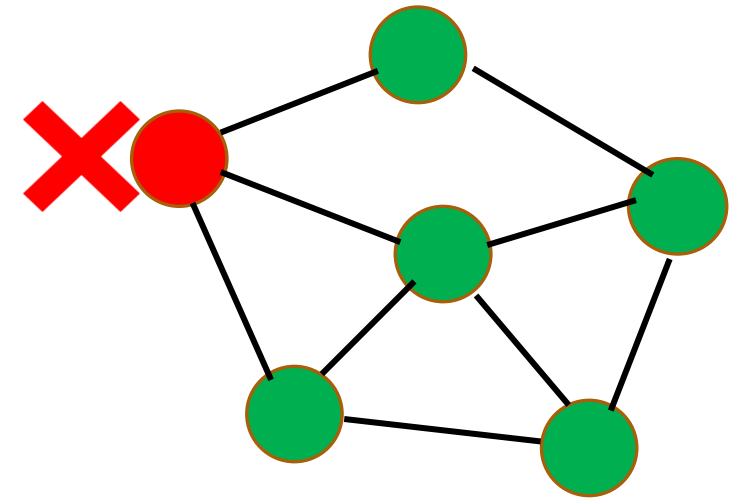
---

## Synchronous vs Asynchronous Networks

- **Synchronous:** I am sure that I'll get the message within a predefined time threshold
- **Asynchronous:** I am not sure whether and when the message will arrive

## Failures in a network --

- **Crash Fault:** A node stops responding
- **Link Fault (or Network Fault):** A link fails to deliver the message
- **Byzantine Fault:** A node starts behaving maliciously



**The Impossibility Theorem:** Consensus is not possible in a perfect asynchronous network even with a single crash failure

# FLP Impossibility

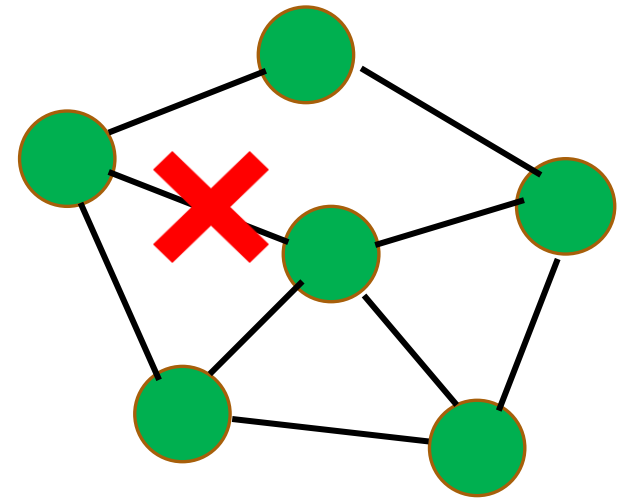
---

## Synchronous vs Asynchronous Networks

- **Synchronous:** I am sure that I'll get the message within a predefined time threshold
- **Asynchronous:** I am not sure whether and when the message will arrive

## Failures in a network --

- **Crash Fault:** A node stops responding
- **Link Fault** (or Network Fault): A link fails to deliver the message
- **Byzantine Fault:** A node starts behaving maliciously



**The Impossibility Theorem:** Consensus is not possible in a perfect asynchronous network even with a single crash failure

# FLP Impossibility

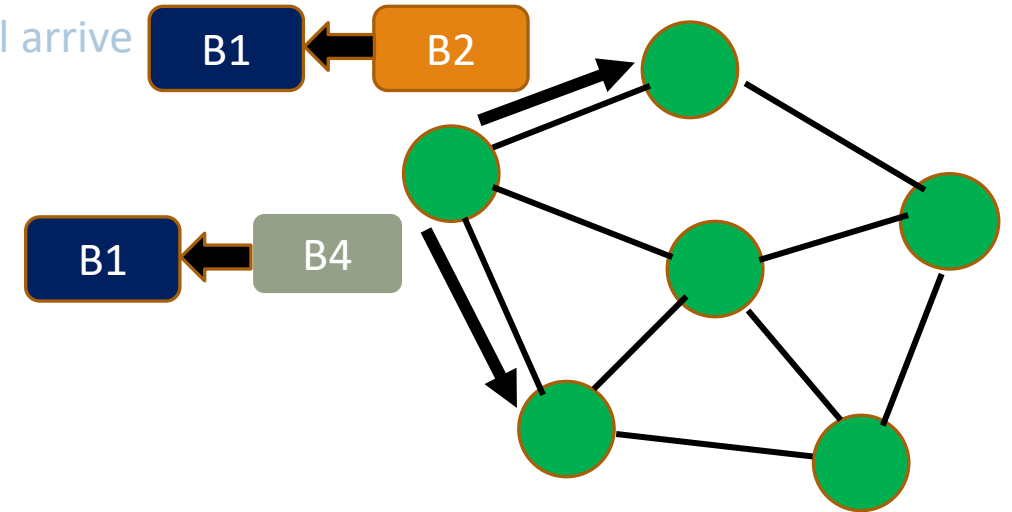
---

## Synchronous vs Asynchronous Networks

- **Synchronous:** I am sure that I'll get the message within a predefined time threshold
- **Asynchronous:** I am not sure whether and when the message will arrive

## Failures in a network --

- **Crash Fault:** A node stops responding
- **Link Fault (or Network Fault):** A link fails to deliver the message
- **Byzantine Fault:** A node starts behaving maliciously



**The Impossibility Theorem:** Consensus is not possible in a perfect asynchronous network even with a single crash failure

# FLP Impossibility

---

## Synchronous vs Asynchronous Networks

- **Synchronous:** I am sure that I'll get the message within a predefined time threshold
- **Asynchronous:** I am not sure whether and when the message will arrive

## Failures in a network --

- **Crash Fault:** A node stops responding
- **Link Fault (or Network Fault):** A link fails to deliver the message
- **Byzantine Fault:** A node starts behaving maliciously

FLP Impossibility Theorem – Fischer, Lynch, Paterson (1985): Consensus is not possible in a perfect asynchronous network even with a single crash failure

- Cannot ensure safety and liveness simultaneously

**Correct processes will yield the correct output**

**The output will be produced within a finite amount of time (eventual termination)**

# Safety vs Liveness Dilemma

---

## Synchronous vs Asynchronous Networks

- Synchronous
- Asynchronous

### The Nakamoto Consensus (Proof of Work)

**Liveness** is more important than **Safety**

**Immediate focus is on liveness with a minimum safety guarantee, full safety will be ensured eventually**

## Failures in a network

- Crash Fault
- Link Fault ( )
- Byzantine F

**The Impossibility Theorem:** Consensus is not possible in a perfect asynchronous network even with a single crash failure

- Cannot ensure safety and liveness simultaneously



# Consensus in an Open Environment

---

2008: A whitepaper got floated on the Internet

- Hash Chain + Puzzle Solving as a Proof (from Bit Gold) + Coin Mining in an open P2P setup
- **Proof of Work (PoW)** -- Nakamoto Consensus

The Key to Success:

Give more emphasis on  
"Liveness" rather than "Safety"

Participants may agree on a transaction that is not  
the final one in the chain

# Consensus

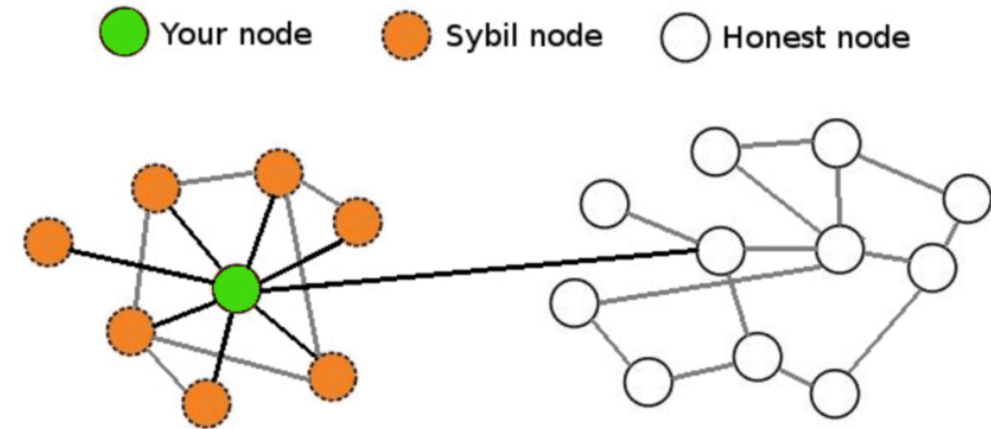
FOLLOWING BITCOIN

THERE ARE NO REAL IDENTITIES...

# Decentralized Identity

---

**Public keys** are used as **identity**



**Single entity can create vast number of identities**

Sybil attacks

Cannot do majority or super-majority voting

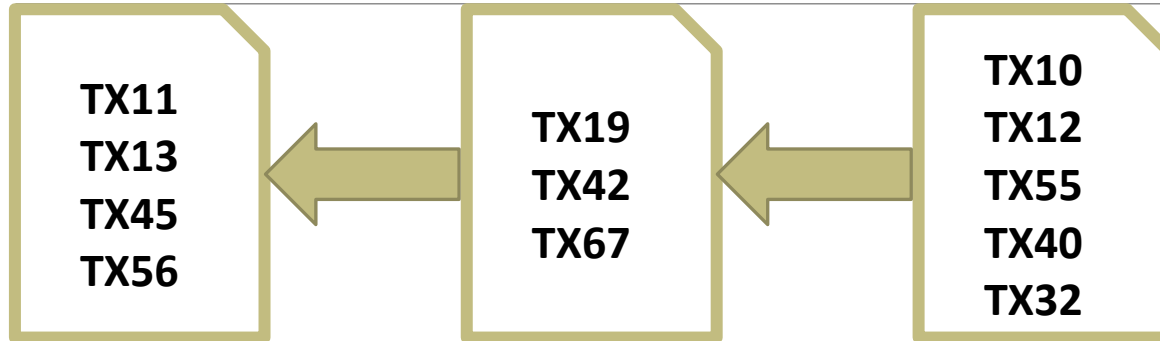
# Temporary Assumption

---

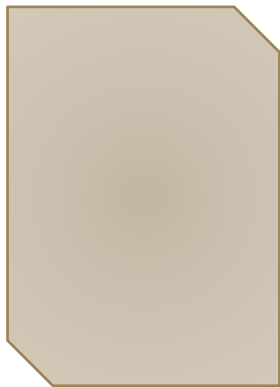
Assume that a node can be randomly chosen to propose the next block.

- Give him a ticket for a lottery
- A single token for an adversary irrespectively of its Sybil nodes (smart lottery 😊)

# Breaking the "Safety vs Liveness" Dilemma



Unconfirmed TX



Miner 1

Unconfirmed TX



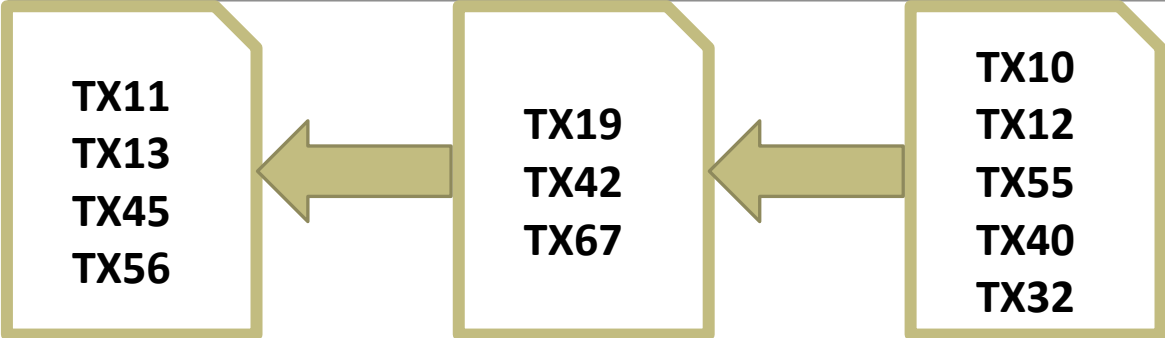
Miner 2

Unconfirmed TX

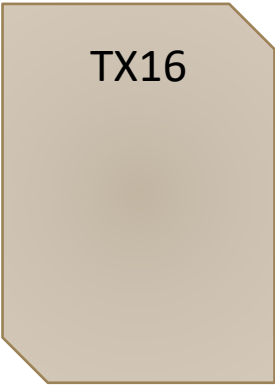


Miner 3

# Breaking the "Safety vs Liveness" Dilemma



Unconfirmed TX



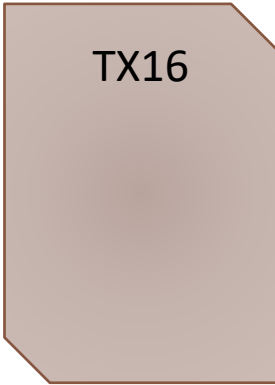
Miner 1

Unconfirmed TX



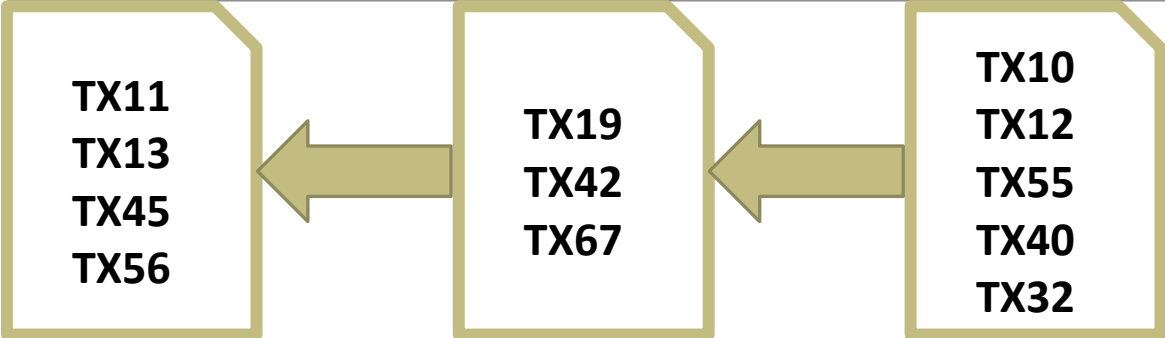
Miner 2

Unconfirmed TX

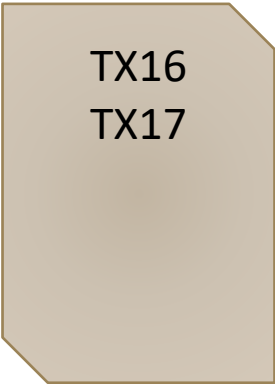


Miner 3

# Breaking the "Safety vs Liveness" Dilemma

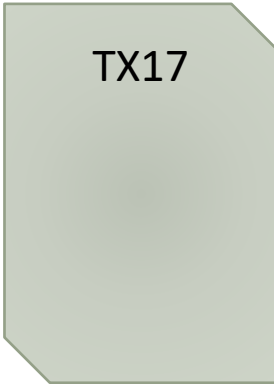


Unconfirmed TX



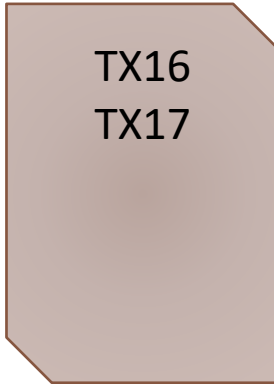
Miner 1

Unconfirmed TX



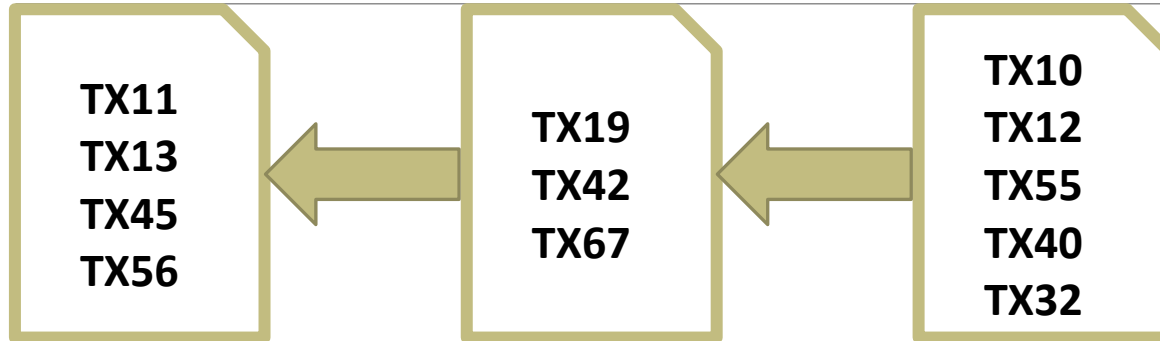
Miner 2

Unconfirmed TX

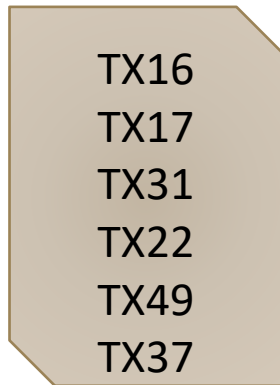


Miner 3

# Breaking the "Safety vs Liveness" Dilemma

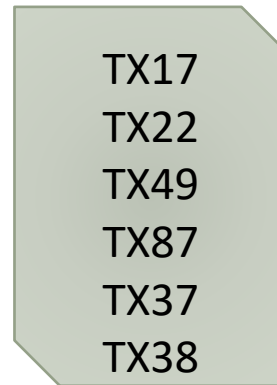


Unconfirmed TX



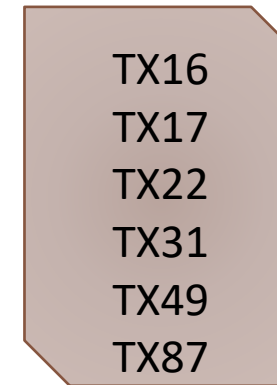
Miner 1

Unconfirmed TX



Miner 2

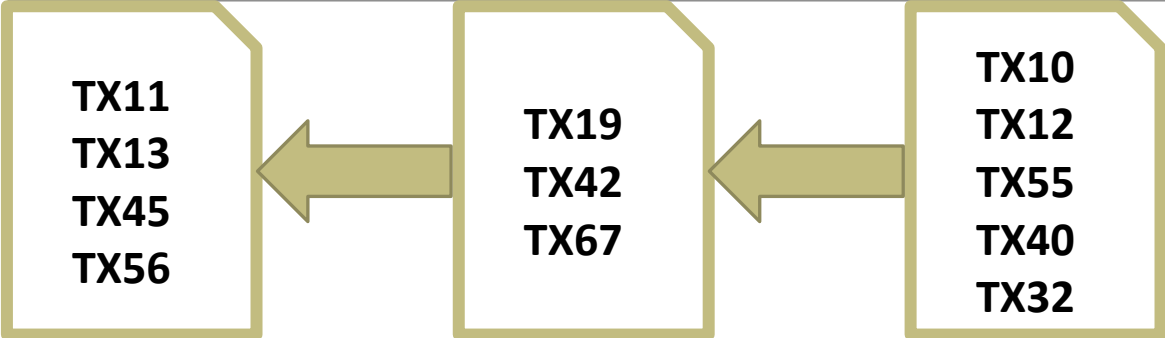
Unconfirmed TX



Miner 3



# Breaking the "Safety vs Liveness" Dilemma



Unconfirmed TX

- TX16
- TX17
- TX31
- TX22
- TX49
- TX37



Miner 1

Unconfirmed TX

- TX17
- TX22
- TX49
- TX87
- TX37
- TX38



Miner 2

Unconfirmed TX

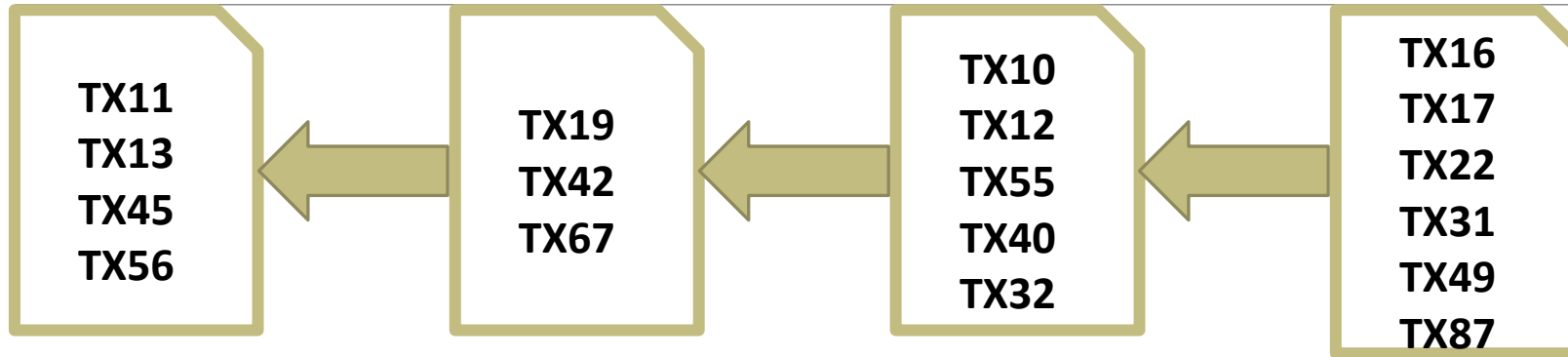
- TX16
- TX17
- TX22
- TX31
- TX49
- TX87



Miner 3

Hooray!!! I am the lucky guy to put the next block

# Breaking the "Safety vs Liveness" Dilemma



**Safety-1:** The next block should be "correct"

- Transactions are verified, the block is correct

Unconfirmed TX

TX16  
TX17  
TX31  
TX22  
TX49  
TX37



Miner 1

Unconfirmed TX

TX17  
TX22  
TX49  
TX87  
TX37  
TX38



Miner 2

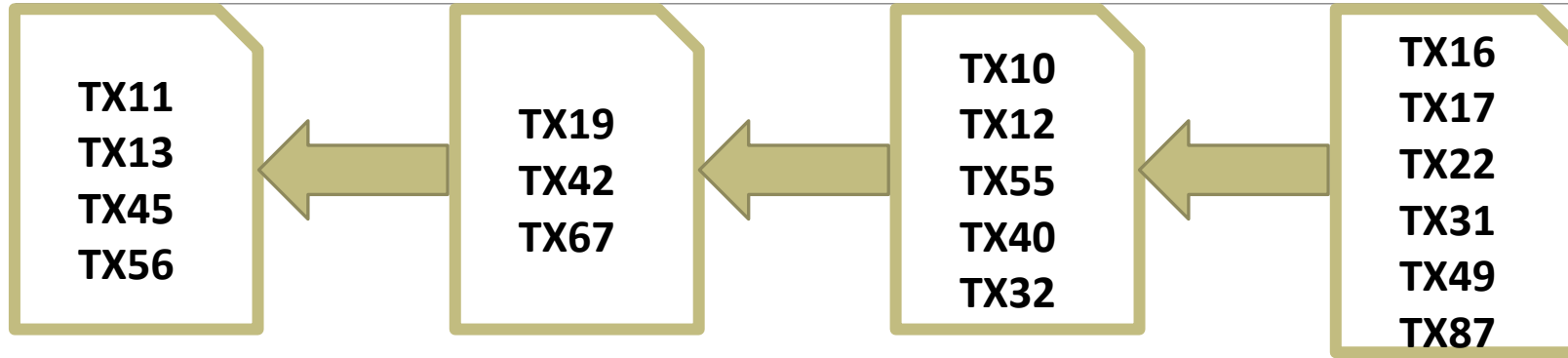
Unconfirmed TX

TX16  
TX17  
TX22  
TX31  
TX49  
TX87



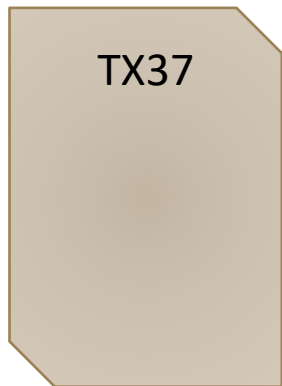
Miner 3

# Breaking the "Safety vs Liveness" Dilemma



The unconfirmed transactions are updated

Unconfirmed TX



Miner 1

Unconfirmed TX



Miner 2

Unconfirmed TX



Miner 3

# Consensus Algorithm

---

*This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.*

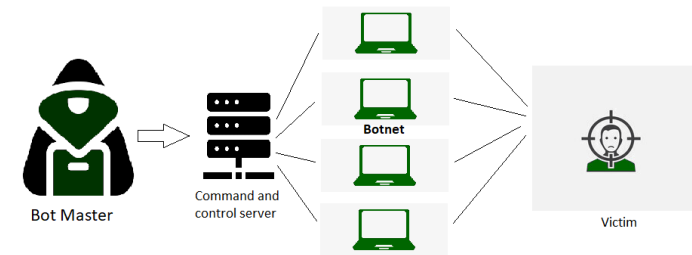
1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

# Does it Work?

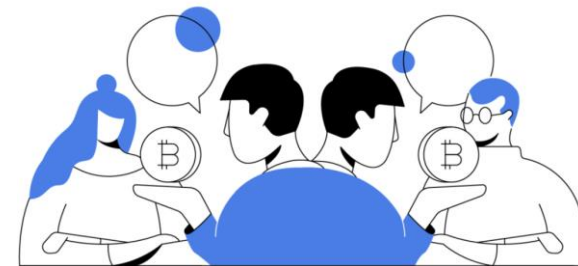
1. Can you steal coins?

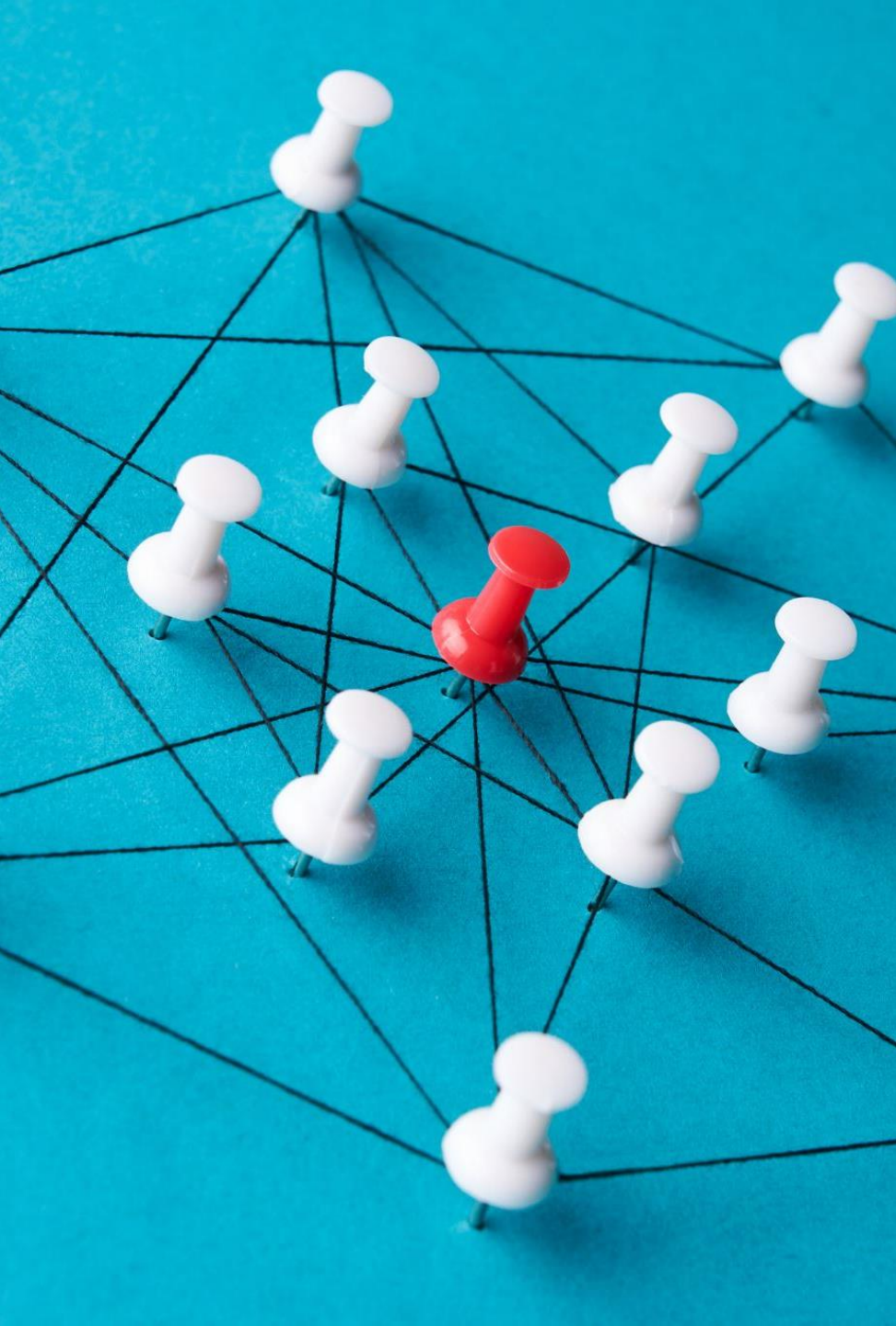


2. Can you make a DoS attack?



3. Double-spend attack?





# How is the Random Node Chosen?

---

ON INCENTIVE ENGINEERING

# Incentives

---

Can we penalize dishonest nodes (e.g., for the double spending we mentioned)?

- Not really (hm...) and not directly

Can we reward honest nodes?

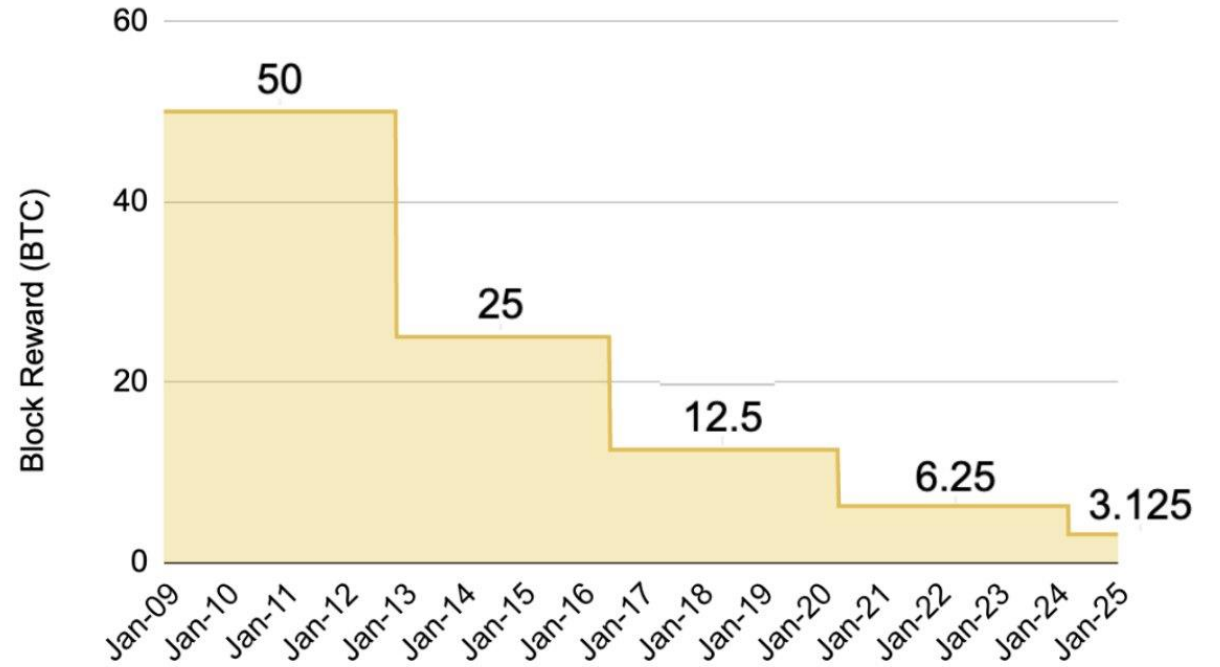
- Of course.
- Use a coin to do that

# Incentive Mechanism: Rewards

---

## Block Rewards

## Transaction Fees



Source: Binance Research

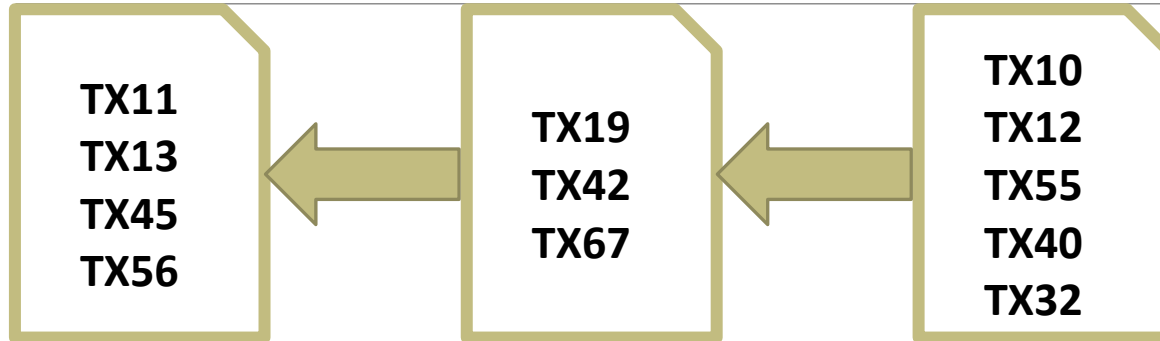


# Problems Again

---

1. How to pick a random node?
2. Unstability of the system because all want to become block proposers
3. Sybil nodes?

# Breaking the "Safety vs Liveness" Dilemma



Miners do not know each other – how can they agree on the same block?

**Safety-2:** All the miners should agree on a single block

- The next block of the blockchain should be selected unanimously

Hey... I was the chosen one

Unconfirmed TX

TX16  
TX17  
TX31  
TX22  
TX49  
TX37



Miner 1

Unconfirmed TX

TX17  
TX22  
TX49  
TX87  
TX37  
TX38



Miner 2

Unconfirmed TX

TX16  
TX17  
TX22  
TX31  
TX49  
TX87



Miner 3

# Solution: Mining

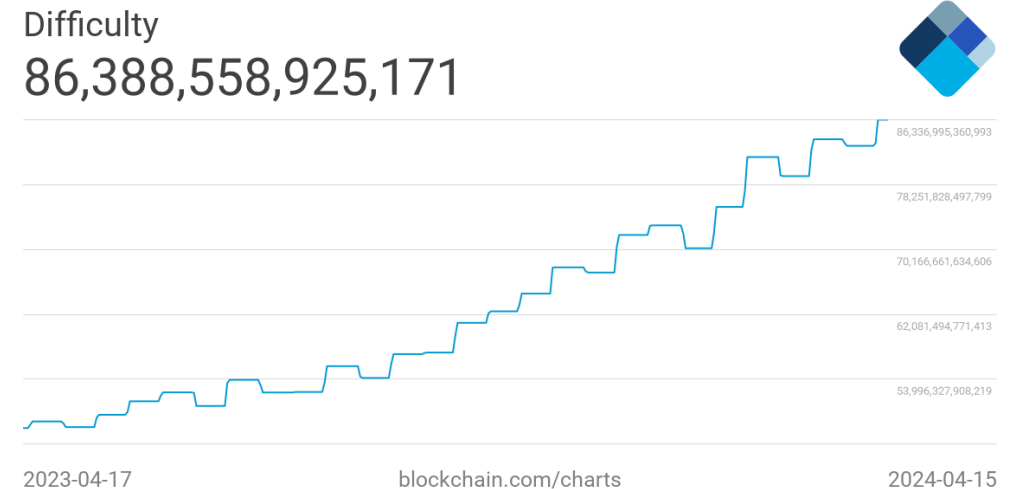
---

Instead of selecting randomly select in proportion to a resource that cannot be monopolized:

- Proof-of-Work (resource: computing power – PoS)
- Proof-of-Stake (resource: ownership of currency – PoW)

# Proof-of-Work (Bitcoin)

- Competition based on computing power. But how?
  - Hash puzzles
    - Difficulty Target
    - Target: 1 block every 10 minutes (bitcoin)
    - Adapted every 2016 blocks



- Can be considered as a tax on identities (not easy to make Sybil identities)

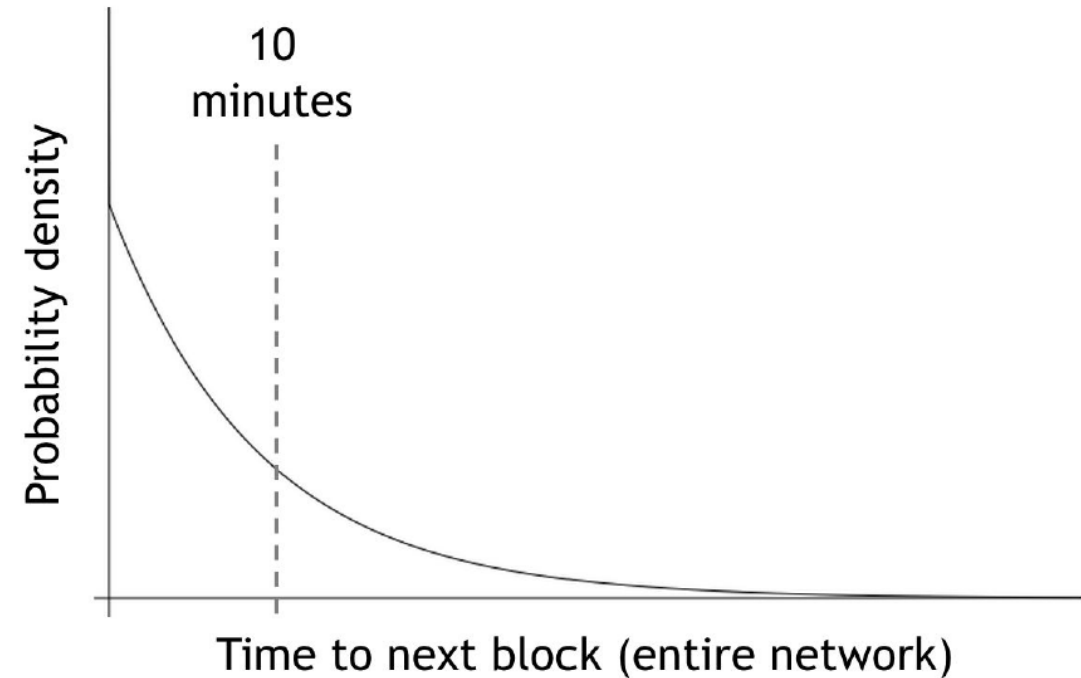
# When does it Fall?

---

- No need for the majority to be honest
- A lot of attacks are infeasible if the majority of the miners **weighted by computing power** are honest

For a specific miner:

$$\text{mean time to next block} = \frac{10 \text{ minutes}}{\text{fraction of hash power}}$$



# Cost of Mining

---

If

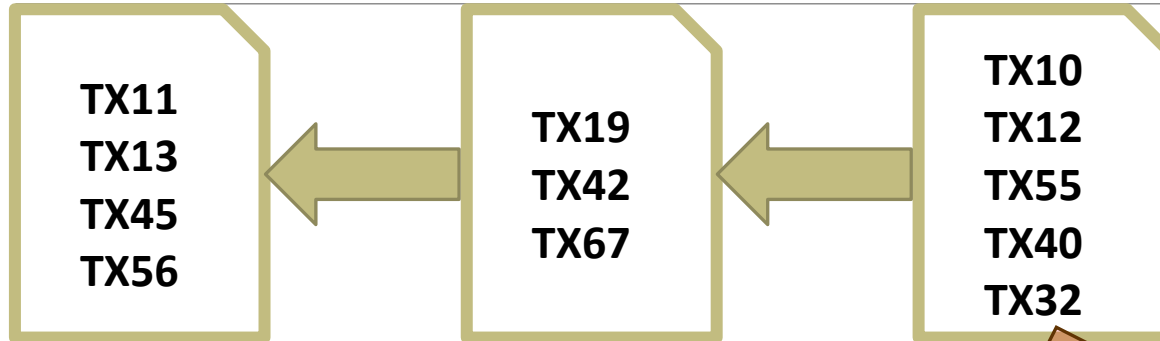
- **mining reward > mining cost**

then miner **profits**

where

- **mining reward = block reward + tx fees**
- **mining cost = hardware cost + operating costs (electricity, cooling, etc.)**

# Breaking the "Safety vs Liveness" Dilemma



Miners do not know each other – how can they agree on the same block?

**Safety-2:** All the miners should agree on a single block

- The next block of the blockchain should be selected unanimously

Compromise

Hey... I was the chosen one

Unconfirmed TX

TX16  
TX17  
TX31  
TX22  
TX49  
TX37



Miner 1

Unconfirmed TX

TX17  
TX22  
TX49  
TX87  
TX37  
TX38



Miner 2

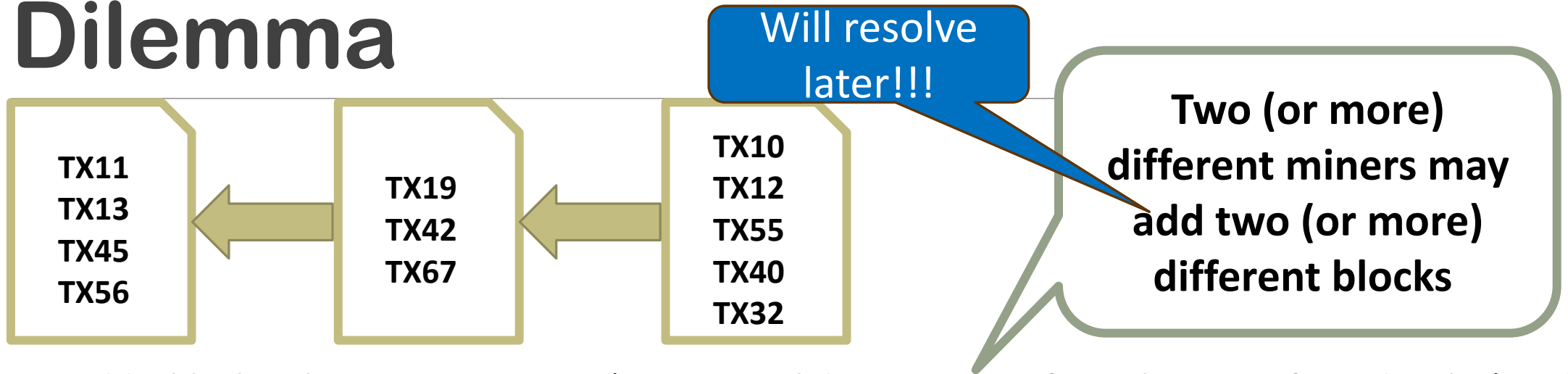
Unconfirmed TX

TX16  
TX17  
TX22  
TX31  
TX49  
TX87



Miner 3

# Breaking the "Safety vs Liveness" Dilemma



**Liveness:** Add a block as long as it is correct (contains valid transactions from the unconfirmed TX list) and move further

Unconfirmed TX

- TX16
- TX17
- TX31
- TX22
- TX49
- TX37



Miner 1

Unconfirmed TX

- TX17
- TX22
- TX49
- TX87
- TX37
- TX38



Miner 2

Unconfirmed TX

- TX16
- TX17
- TX22
- TX31
- TX49
- TX87

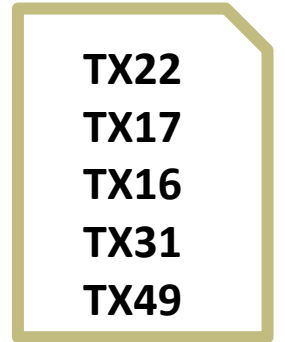
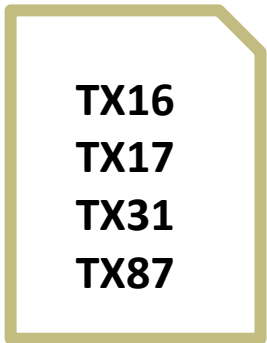
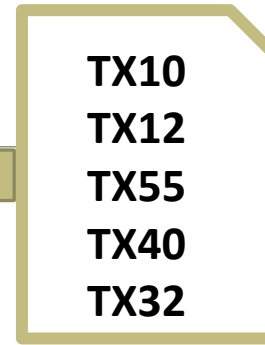
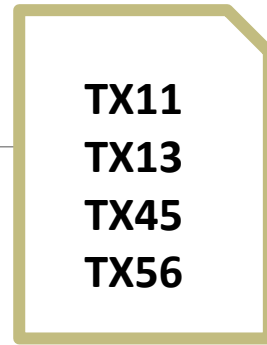


Miner 3

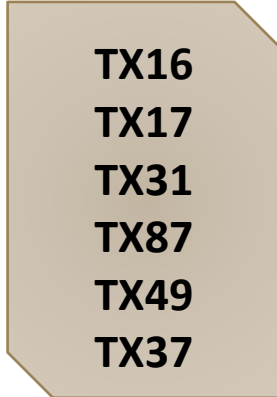


# Nakamoto Consensus (Proof of Work)

- No fixed ordering of transactions
- No fixed number of transactions per block
- **Limit on the Block size**

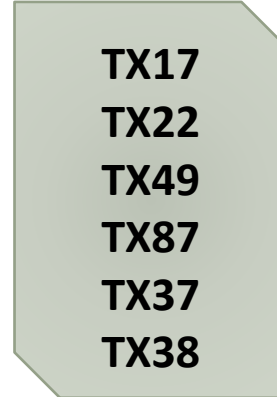


Unconfirmed TX



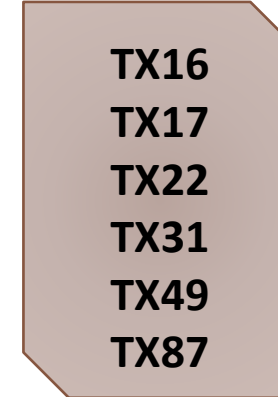
Miner 1

Unconfirmed TX



Miner 2

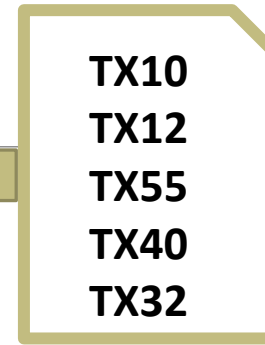
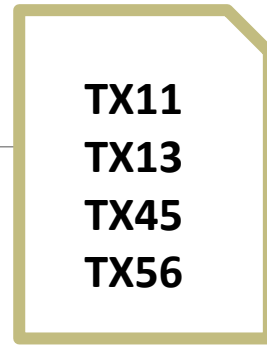
Unconfirmed TX



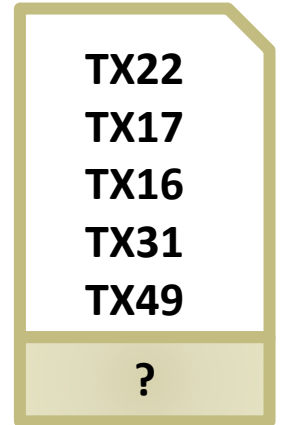
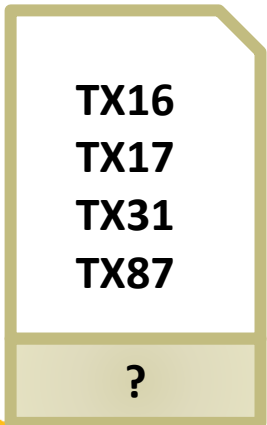
Miner 3

# Nakamoto Consensus (Proof of Work)

Expectation: One of the miners will be able to generate the proof



- Generate the proof (nonce)
  - **Generation: Complex**
  - **Verification: Easy**

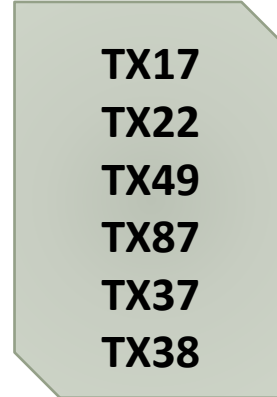


Unconfirmed TX



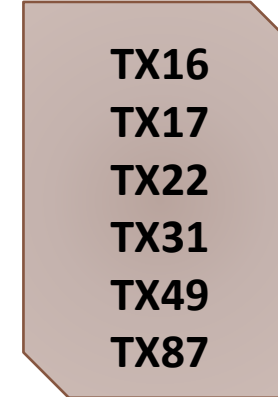
Miner 1

Unconfirmed TX



Miner 2

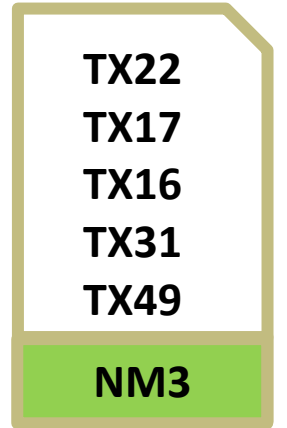
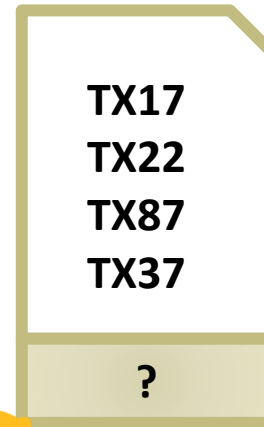
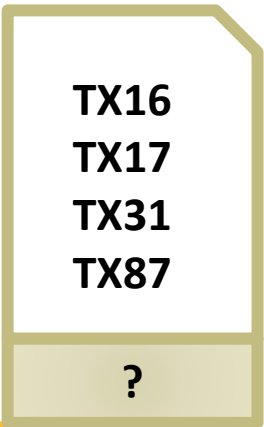
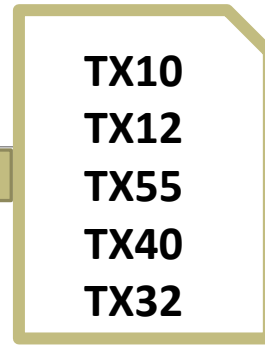
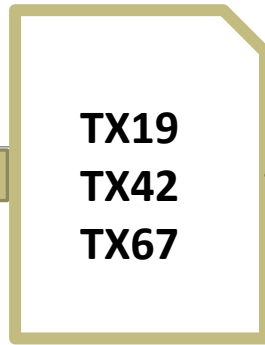
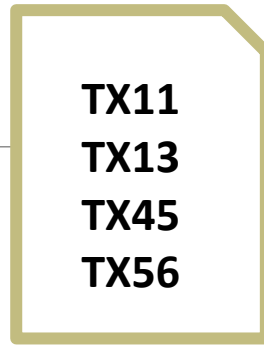
Unconfirmed TX



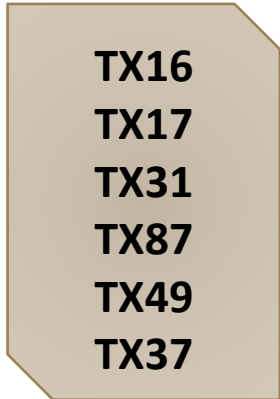
Miner 3

# Nakamoto Consensus (Proof of Work)

Expectation: One of the miners will be able to generate the proof

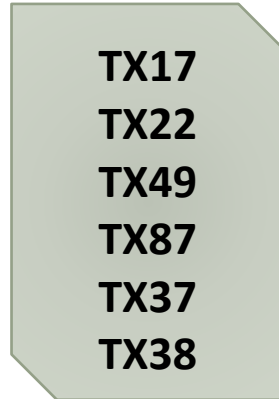


Unconfirmed TX



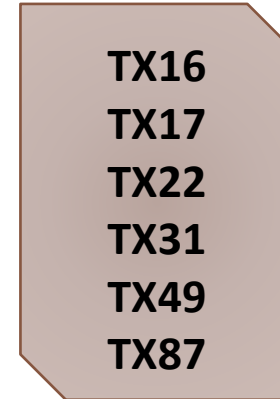
Miner 1

Unconfirmed TX



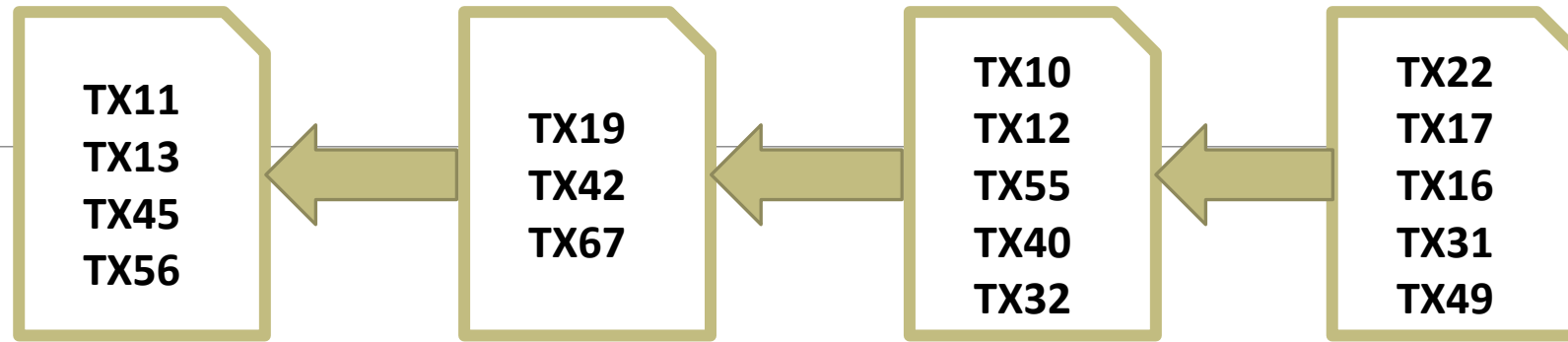
Miner 2

Unconfirmed TX



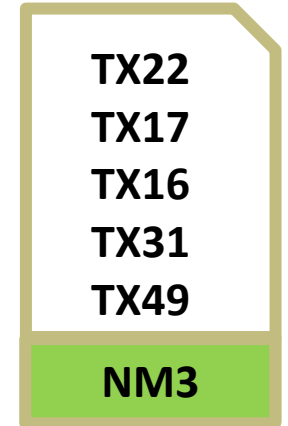
Miner 3

# Nakamoto Consensus (Proof of Work)

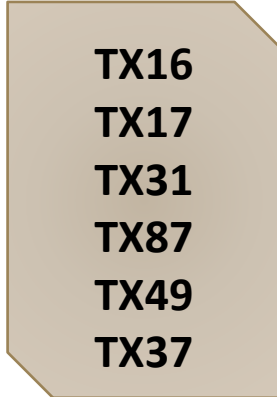


Sign the block and broadcast

- Gossip over the P2P network

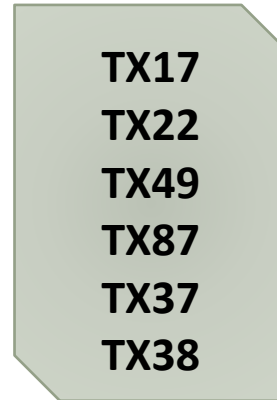


Unconfirmed TX



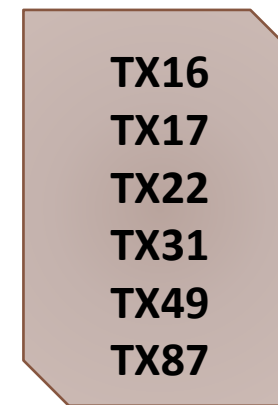
Miner 1

Unconfirmed TX



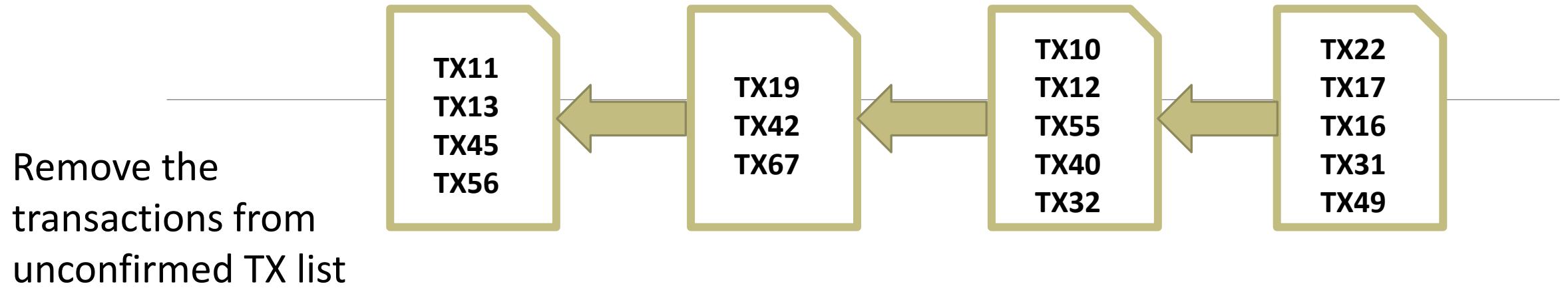
Miner 2

Unconfirmed TX

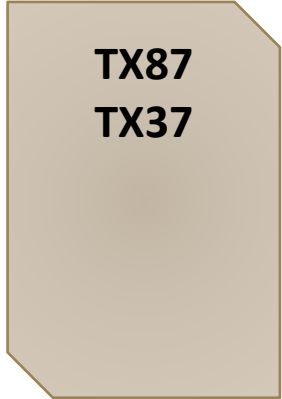


Miner 3

# Nakamoto Consensus (Proof of Work)



Unconfirmed TX



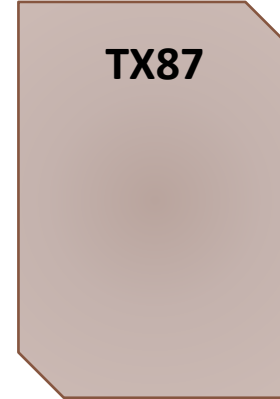
Miner 1

Unconfirmed TX



Miner 2

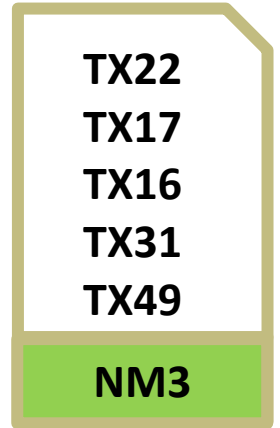
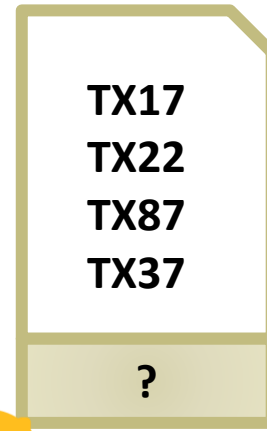
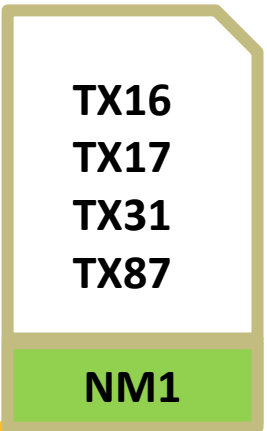
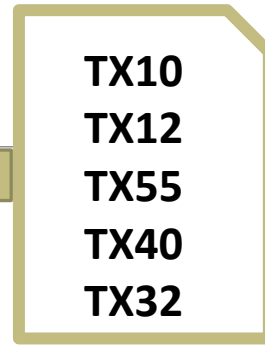
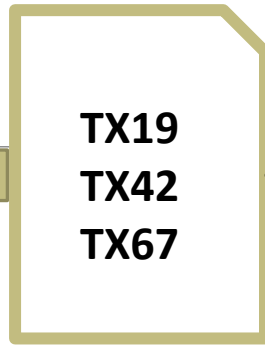
Unconfirmed TX



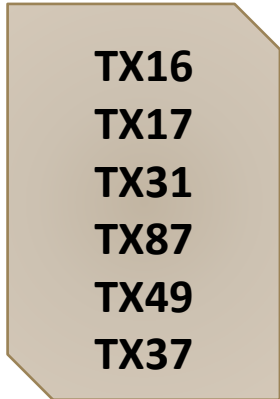
Miner 3

# Nakamoto Consensus (Proof of Work)

Reality: (Not likely but possible) More than one miners generate the proof simultaneously

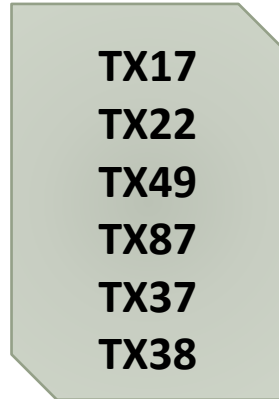


Unconfirmed TX



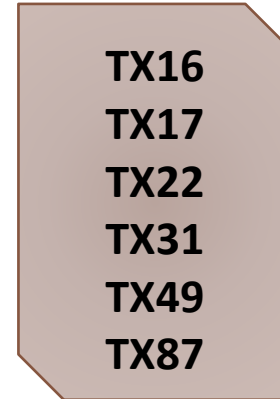
Miner 1

Unconfirmed TX



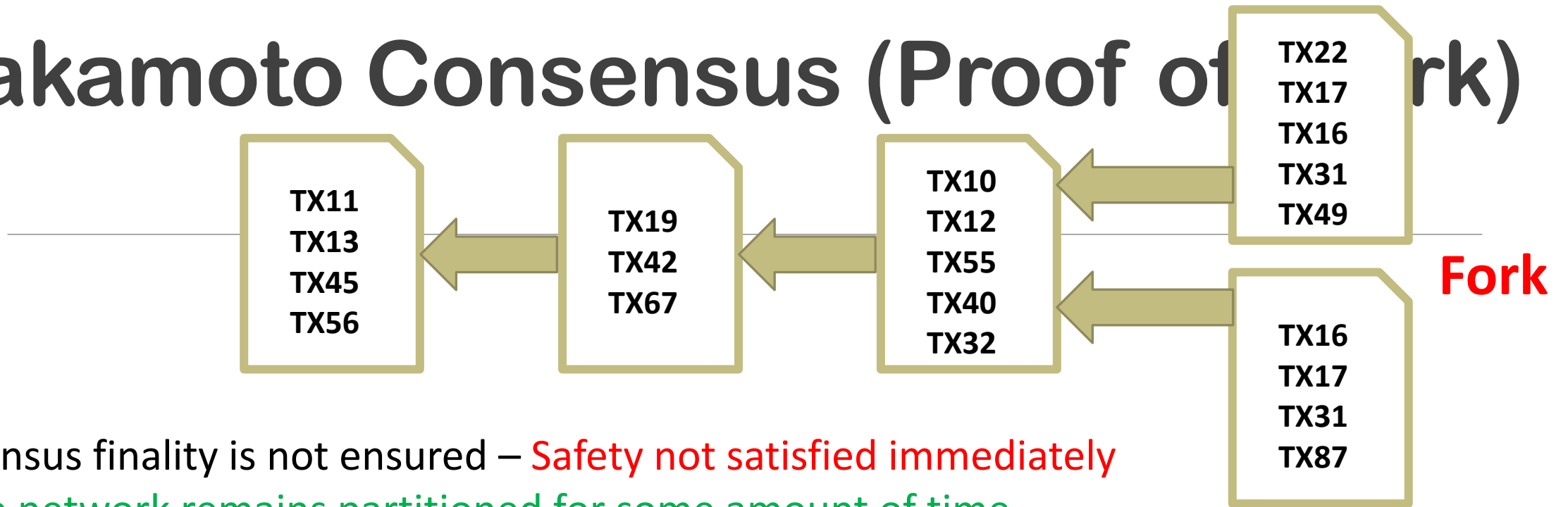
Miner 2

Unconfirmed TX



Miner 3

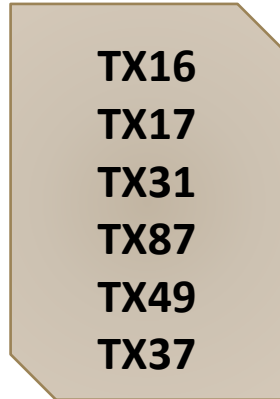
# Nakamoto Consensus (Proof of Work)



Consensus finality is not ensured – **Safety not satisfied immediately**

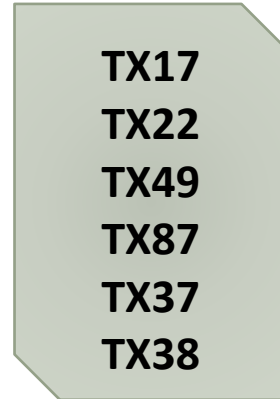
- The network remains partitioned for some amount of time

Unconfirmed TX



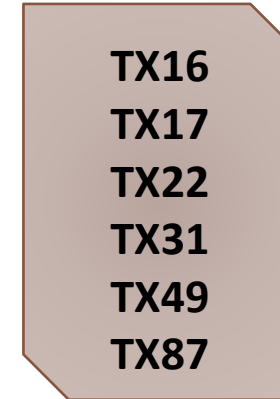
**Miner 1**

Unconfirmed TX



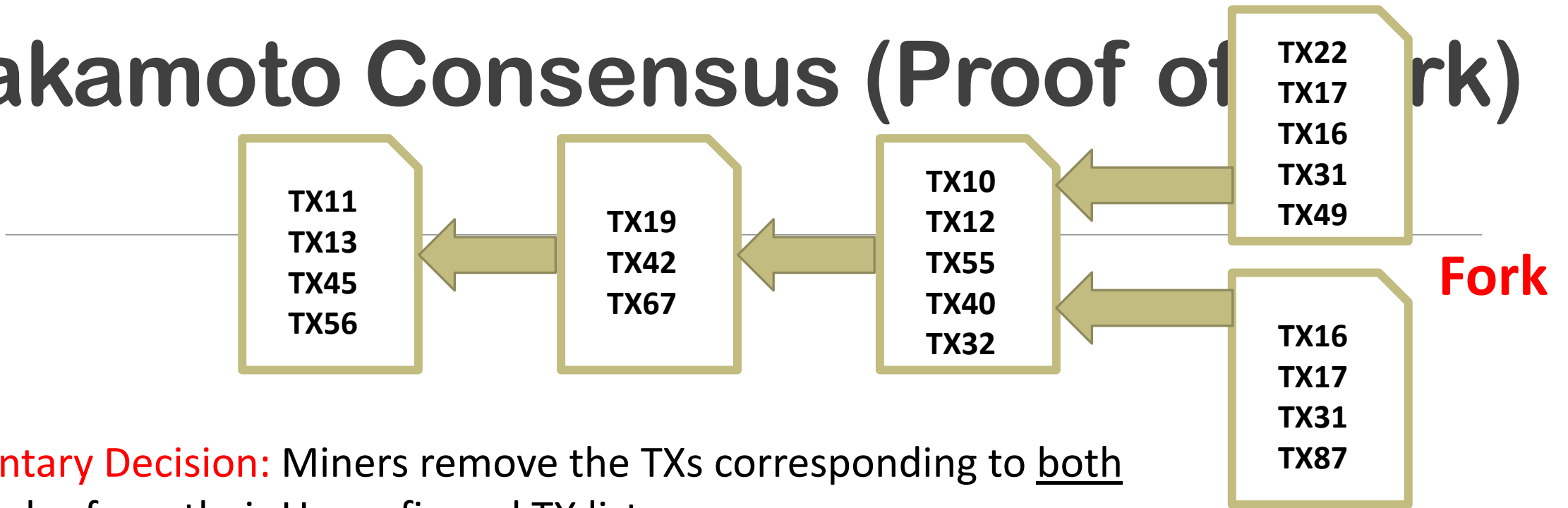
**Miner 2**

Unconfirmed TX



**Miner 3**

# Nakamoto Consensus (Proof of Work)



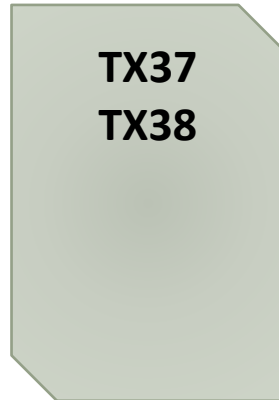
**Momentary Decision:** Miners remove the TXs corresponding to both the blocks, from their Unconfirmed TX list

Unconfirmed TX



Miner 1

Unconfirmed TX



Miner 2

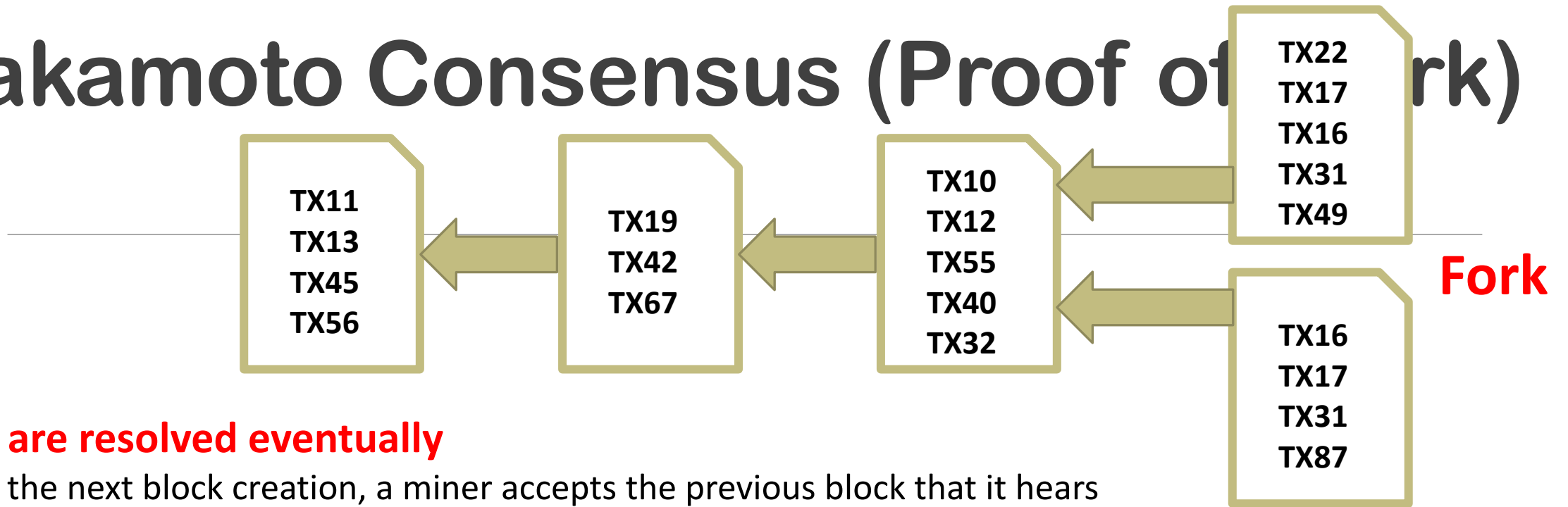
Unconfirmed TX



Miner 3



# Nakamoto Consensus (Proof of Work)



## Forks are resolved eventually

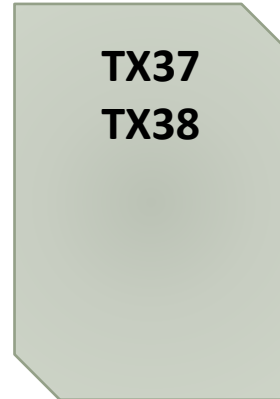
- For the next block creation, a miner accepts the previous block that it hears from the majority of the neighbor

Unconfirmed TX



Miner 1

Unconfirmed TX



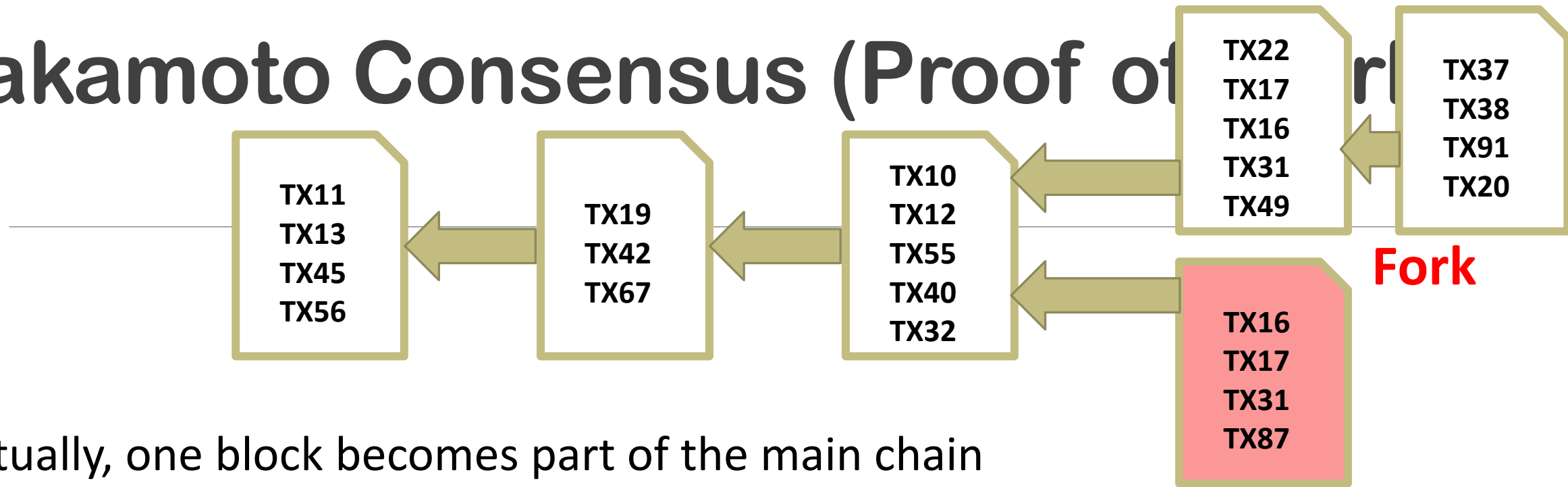
Miner 2

Unconfirmed TX



Miner 3

# Nakamoto Consensus (Proof of Work)



Unconfirmed TX



**Miner 1**

Unconfirmed TX



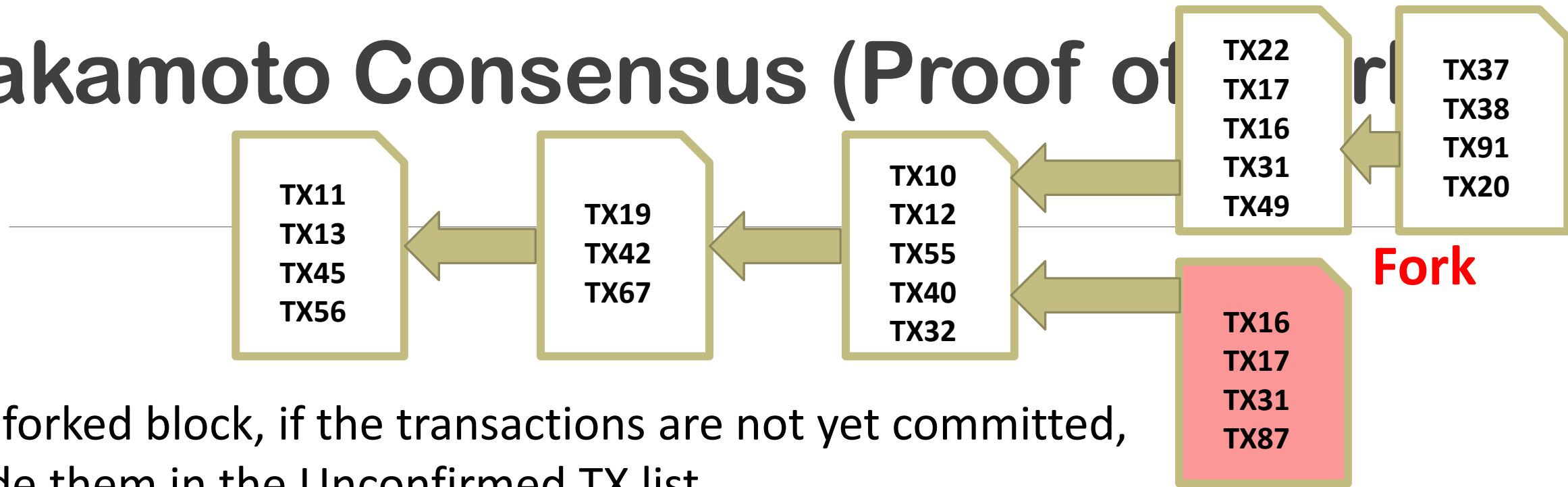
**Miner 2**

Unconfirmed TX



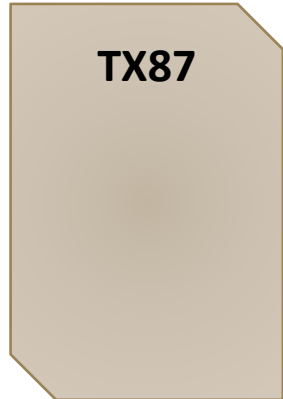
**Miner 3**

# Nakamoto Consensus (Proof of Work)



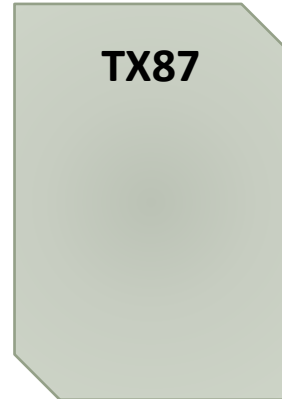
For a forked block, if the transactions are not yet committed, include them in the Unconfirmed TX list

Unconfirmed TX



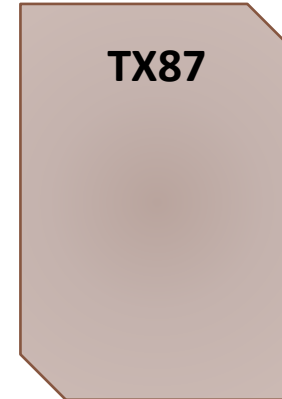
Miner 1

Unconfirmed TX



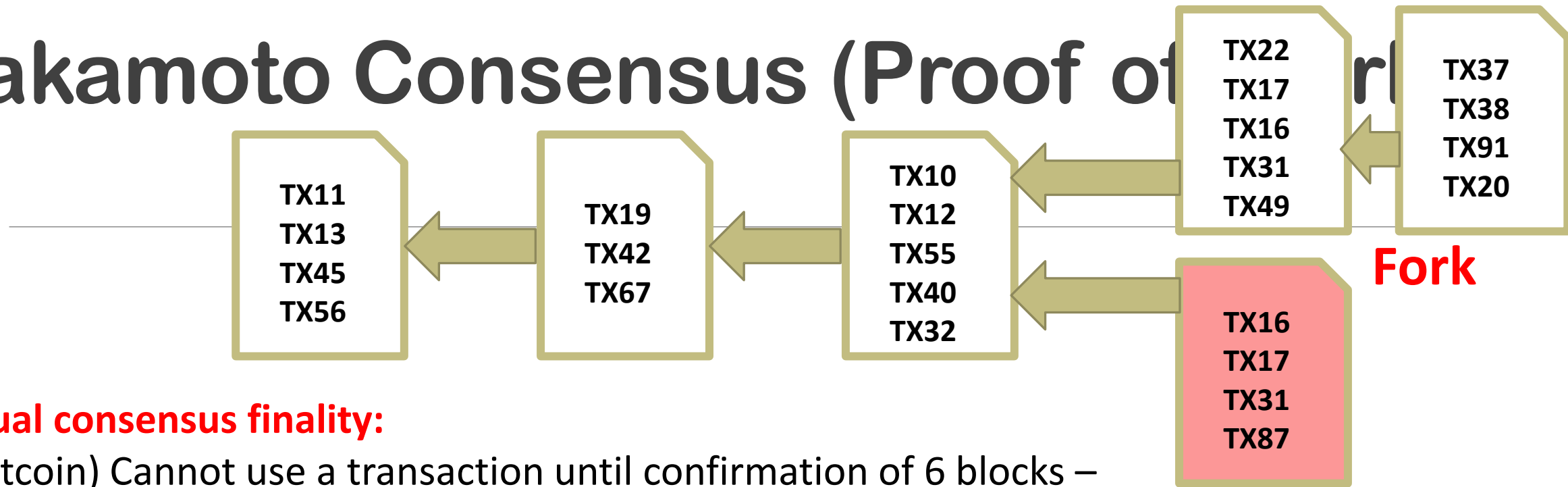
Miner 2

Unconfirmed TX



Miner 3

# Nakamoto Consensus (Proof of Work)



## Eventual consensus finality:

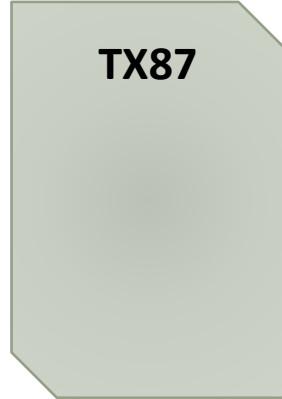
- (Bitcoin) Cannot use a transaction until confirmation of 6 blocks – ensured through scripts

Unconfirmed TX



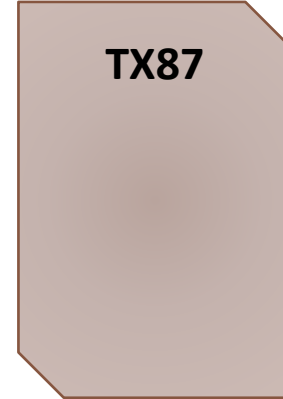
Miner 1

Unconfirmed TX



Miner 2

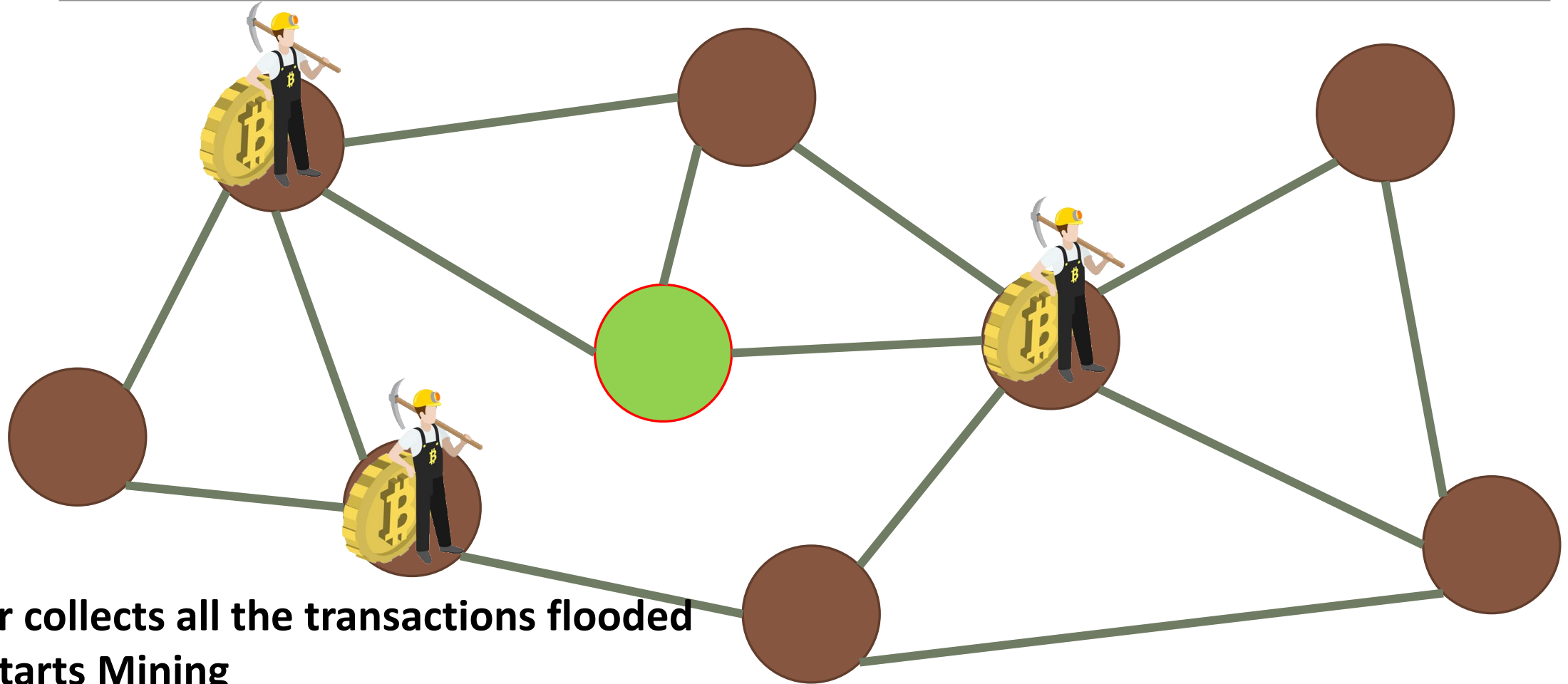
Unconfirmed TX



Miner 3

# Mining in a Bitcoin Network

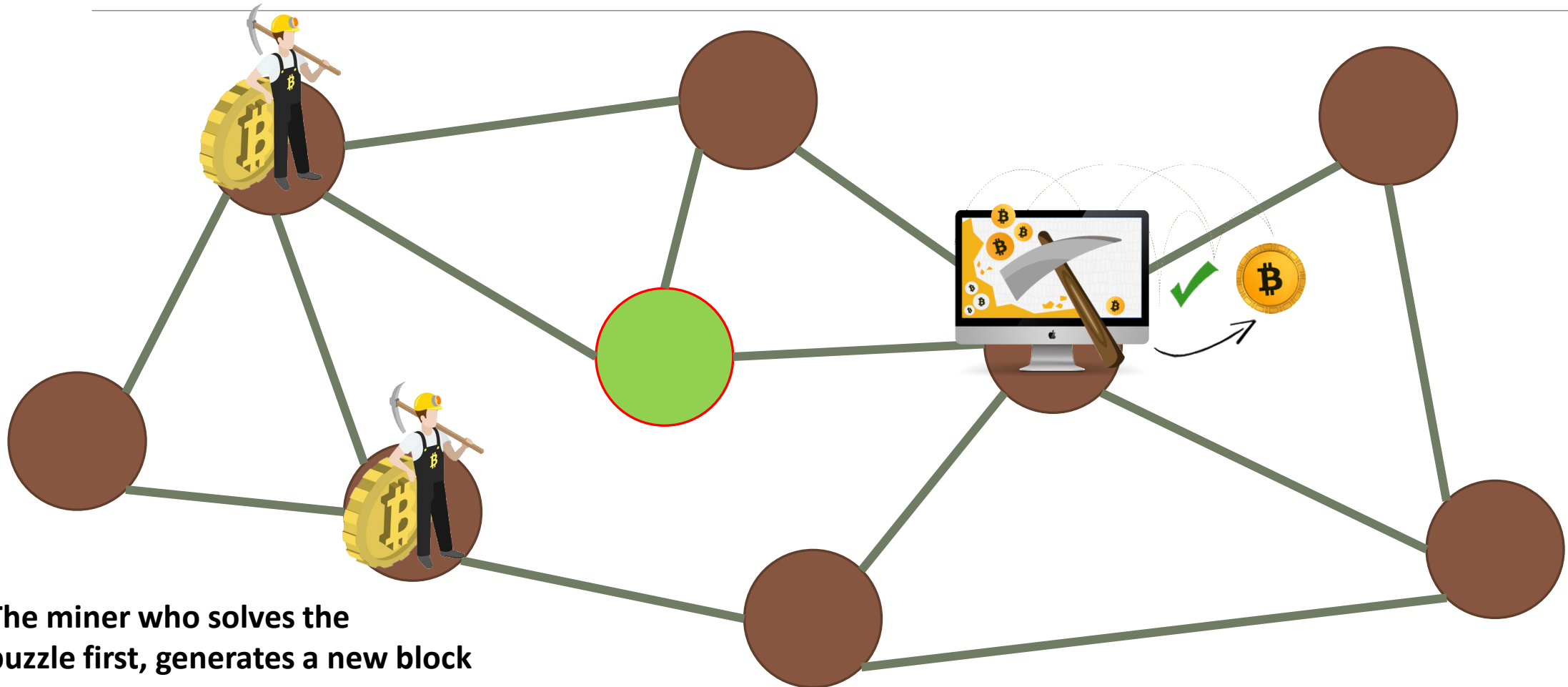
---



**Miner collects all the transactions flooded and starts Mining**

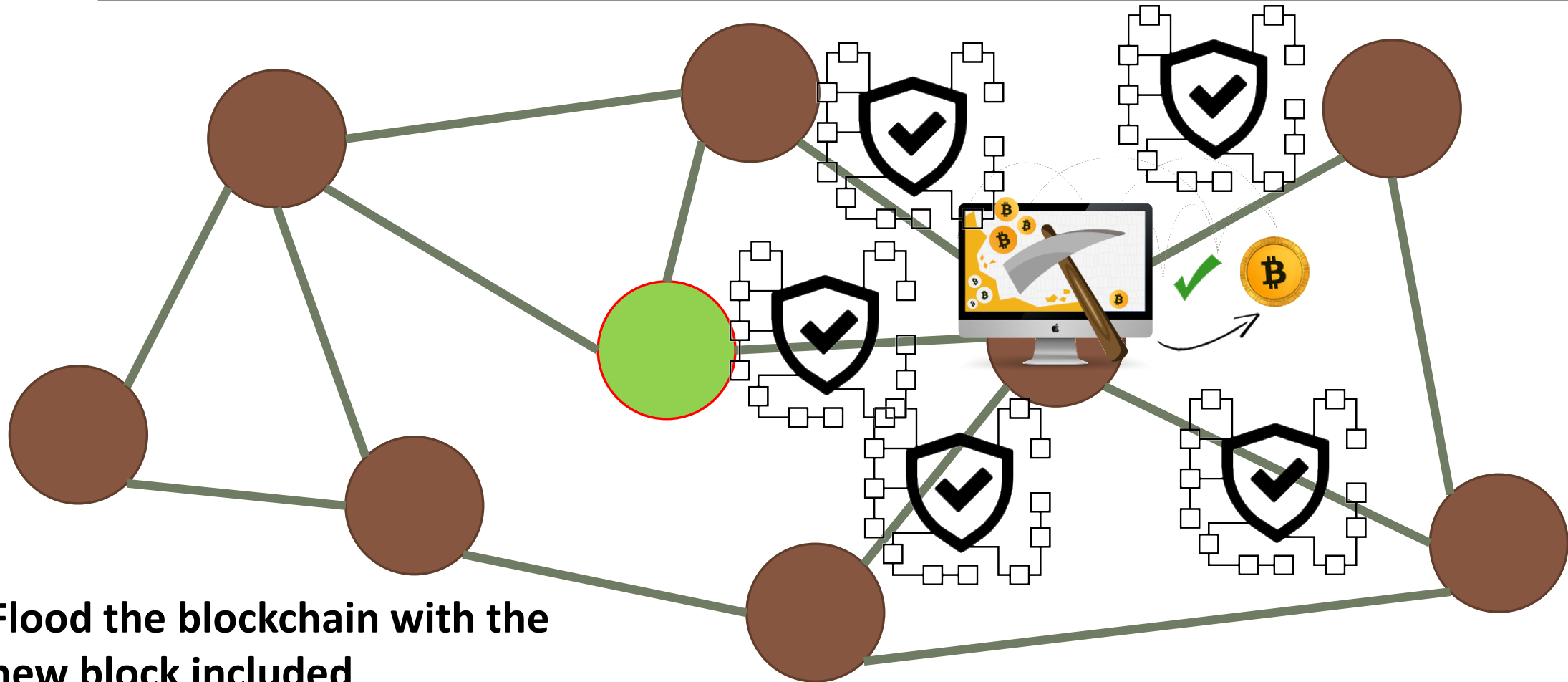
# Block Generation

---



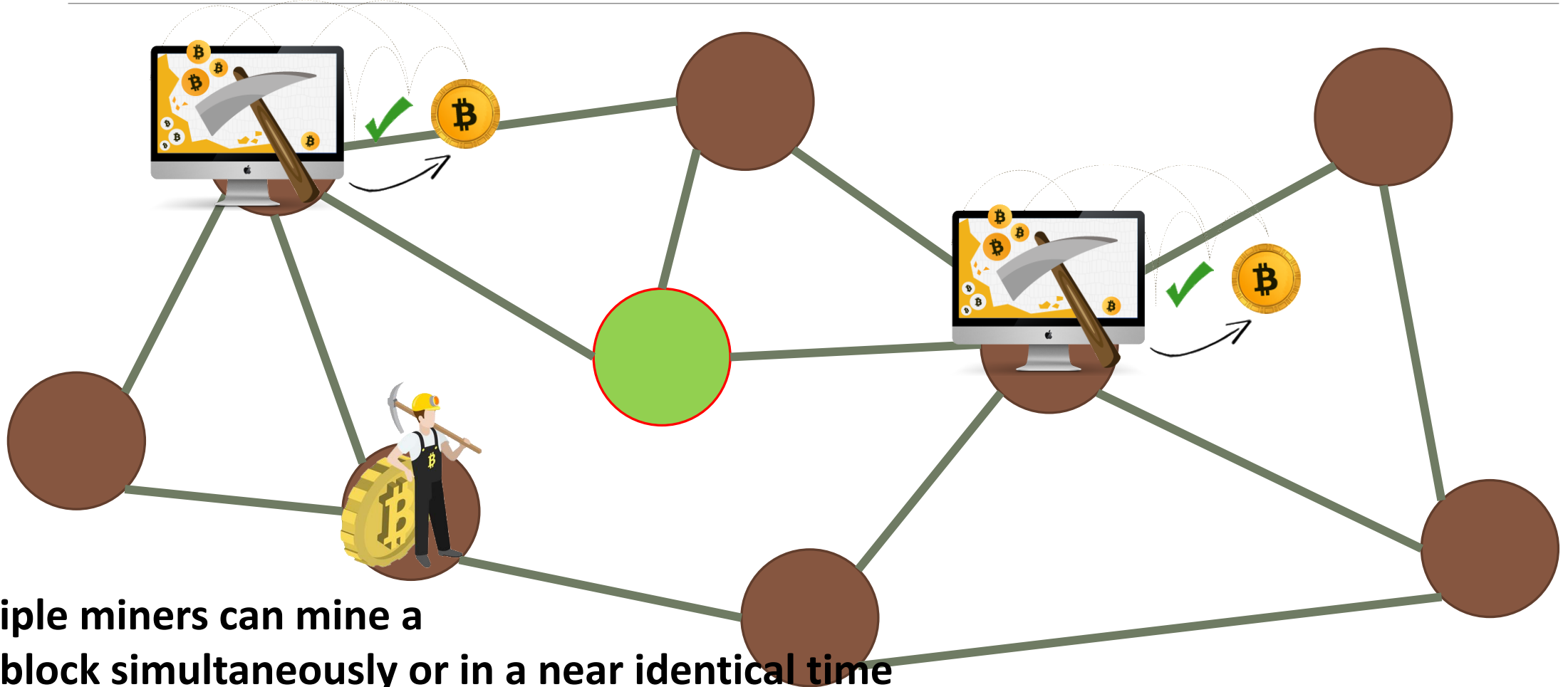
**The miner who solves the puzzle first, generates a new block**

# Block Flooding



**Flood the blockchain with the new block included**

# Block Propagation



**Multiple miners can mine a new block simultaneously or in a near identical time**

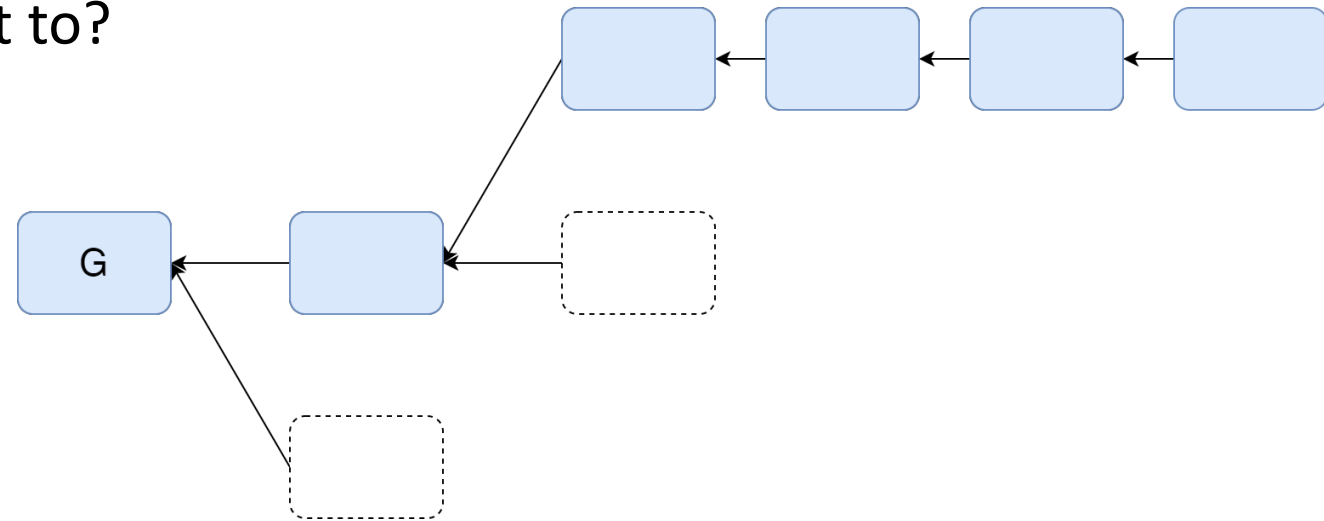


# Longest Chain Protocol

---

Where should the mined block hash-point to?

Blockchain may have **forks**  
because of network delays  
because of adversarial action

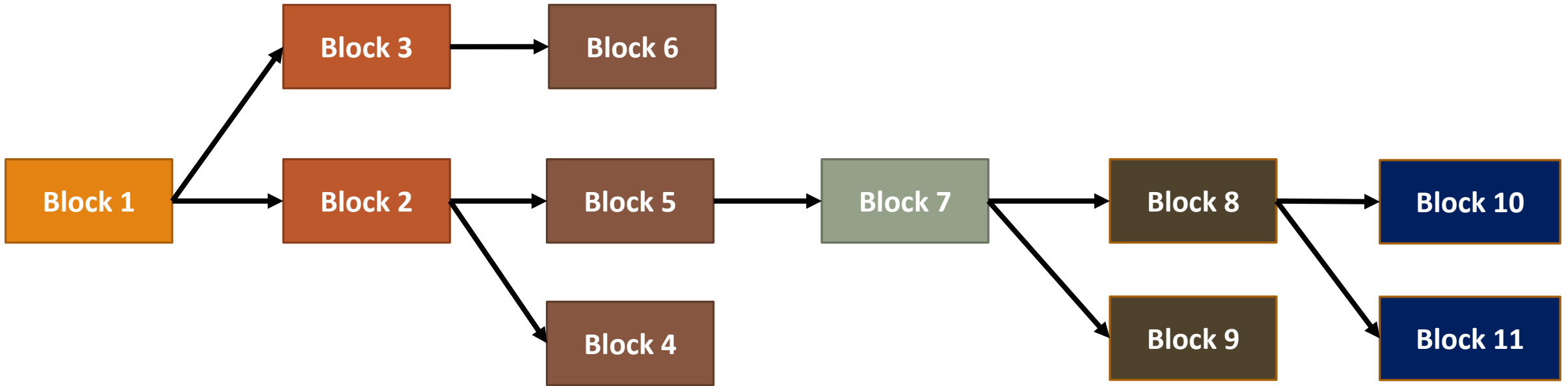


## **Longest chain protocol**

attach the block to the leaf of the longest chain in the block tree  
what if two equal length chains? Discussed previously

# Block Propagation – Accept One of the Longest Chains

---



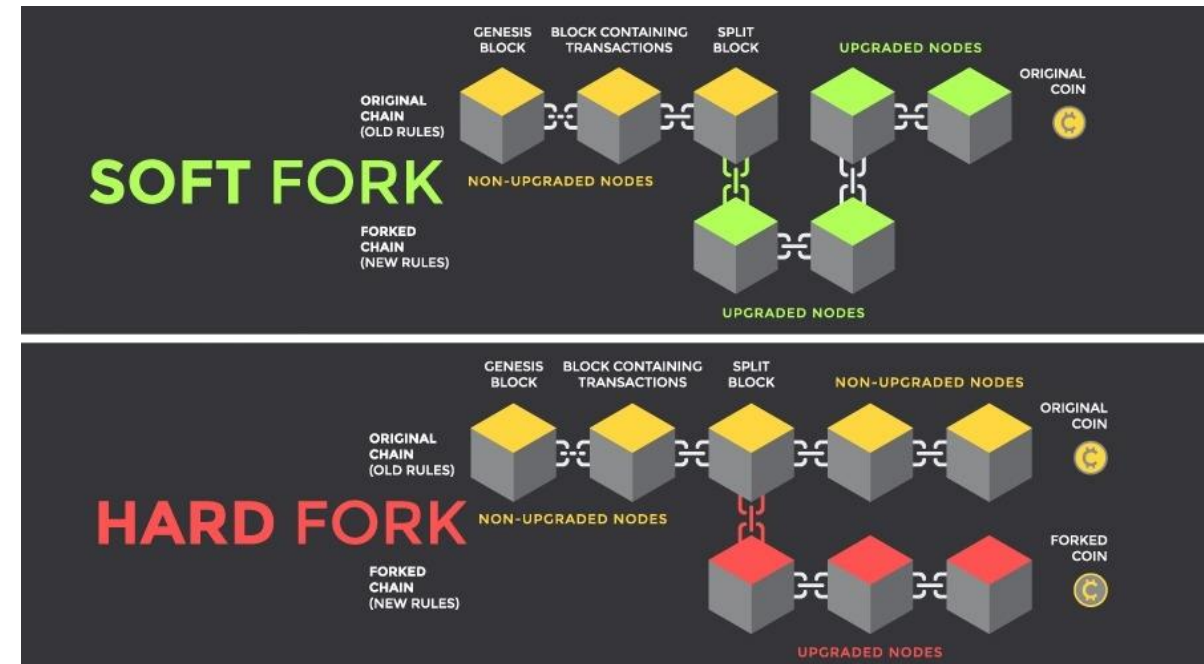
“Accidental” forks occur rarely. Even if they occur, eventually only one becomes part of the longest chain

There are “intentional” forks of two types: [hard forks](#) and [soft forks](#) to come up with new versions like Bitcoin Cash, etc., or to upgrade software versions

# Changing the Protocol

Because not all nodes are updated:

- [Hard forks](#)
  - Introduction of features that previously were considered invalid
  - Examples:
    - [Bitcoin Cash](#)
    - Ethereum: The DAO Heist
- Soft forks
  - Stricter validation rules
  - Examples:
    - SegWit (blockchain)



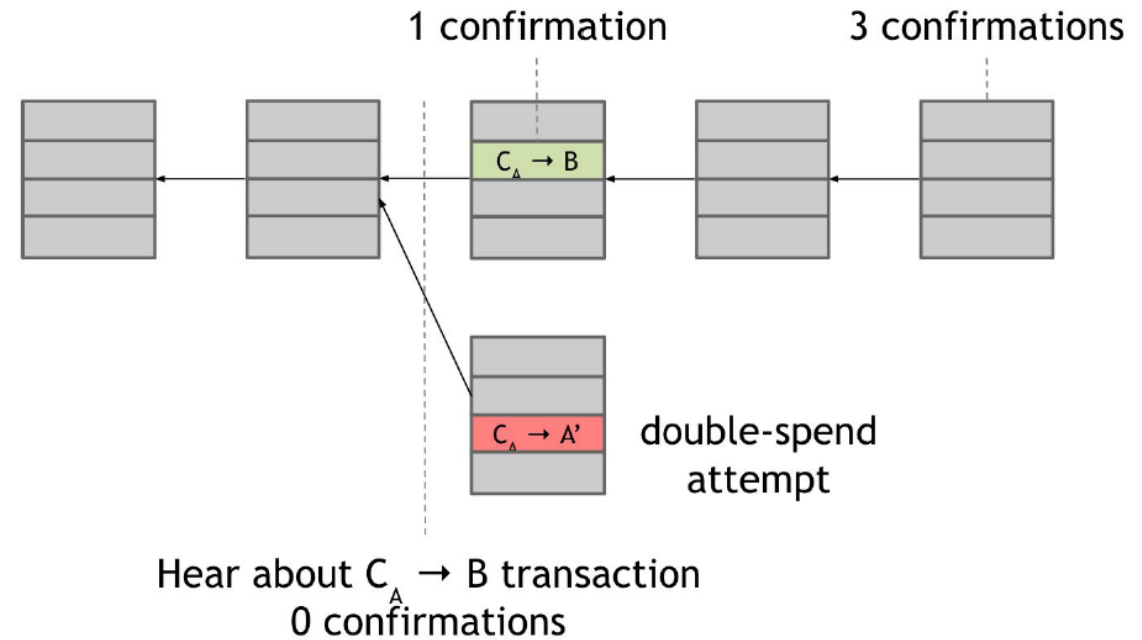
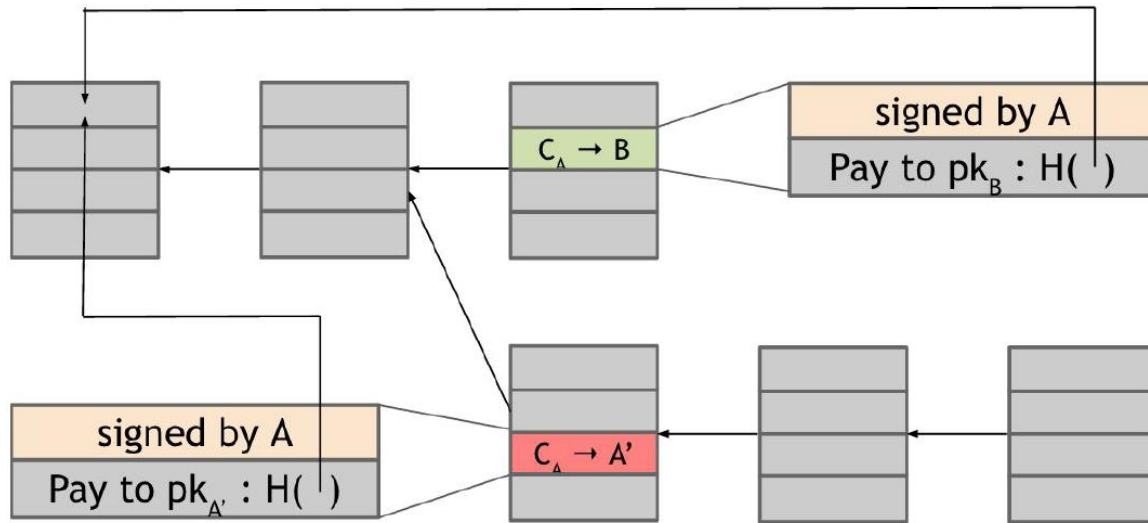
# What would happen if?

---

Assume a 51-percent attacker:

- Can he steal coins?
- Can he suppress transactions?
- Can he change the block reward?
- Can he destroy confidence in the coin?

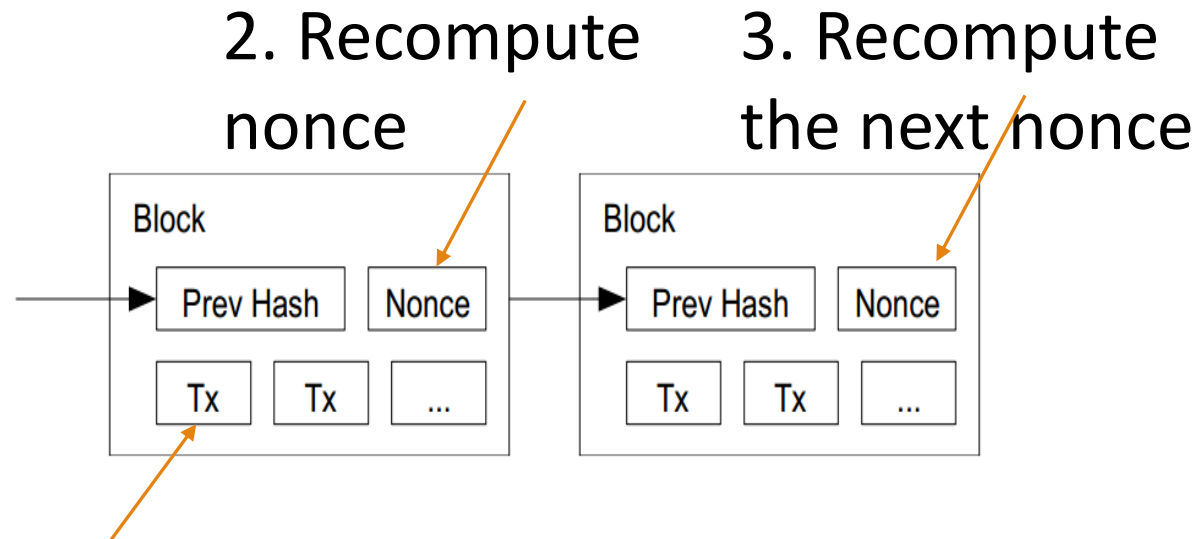
# Double-Spend Attack



# Reverting is Hard

---

Reverting gets exponentially hard as the chain grows.



1. Modify the transaction  
(revert or change the payer)

# Protection

---

## 1. Against invalid transactions:

1. Cryptography
2. Enforced by Consensus

## 2. Against double-spending:

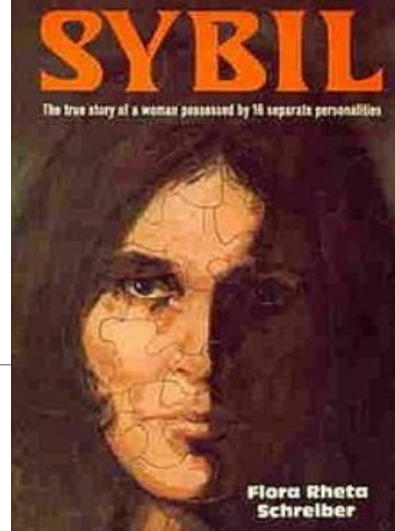
1. Consensus determines which transaction will end up in the blockchain
2. High probability but never sure that the transaction is in the consensus branch.
3. For BitCoin, after 6 confirmations it is OK.
4. For Ethereum after 30 confirmations it is OK.

# Sybil Attack

---

A *Sybil Attack* is an attack on a peer-to-peer network (not necessarily blockchain) that involves a single party controlling many nodes on the given network, unbeknownst to the rest of the network.

This is carried out in an attempt to gain more power over the network that one individual would have





# Sybil Attack

---

There are two popularly cited varieties: **hijacking** and **forging**

- **Hijack:** the attack is able to take control of other network members nodes (usually without the network member knowing it) and can use them for malicious purposes
- **Forge:** the attacker creates many of their own nodes through the use of multiple computers, VPNs, or other means of spoofing identity, appearing to be multiple people acting on their own (**important for blockchain**)

Named after a case study of dissociative identity disorder (person's name was Sybil)

# Sybil Attack

---

## Common Prevention Techniques:

- Require identity verification of some kind before one can join the network (Permissioned blockchain)
- Require identities to be expensive to create (e.g., there is a large fee to join the network)
- Do not rely on identity as a means of power or network influence (e.g., PoW relies on hash power, not identity)
- Weight identities differently based on other network factors (e.g., trust networks determining voting power)

# The Monopoly Problem

---

PoW depends on the computing resources (or other resources, e.g., PoS) available to a miner

- Miners having more resources have more probability to complete the work

Monopoly can increase over time (*Tragedy of the Commons*)

- Miners will get less reward over time
- Users will get discouraged to join as the miner
- Few miners with large computing resources may get control over the network

**51% Attack:** A group of miners control more than 50% of the hash rate of the network

- Hypothetical as of now for Bitcoin (as the network is large), but not impossible (happened for Krypton – Ethereum based blockchain, in August, 2016)

# The Limit of PoW

---

**The Good:** A fully decentralized consensus for permissionless models

- works good for cryptocurrencies – serves its purposes

**The Bad:** Do not trust the individuals, but trust the society as a whole

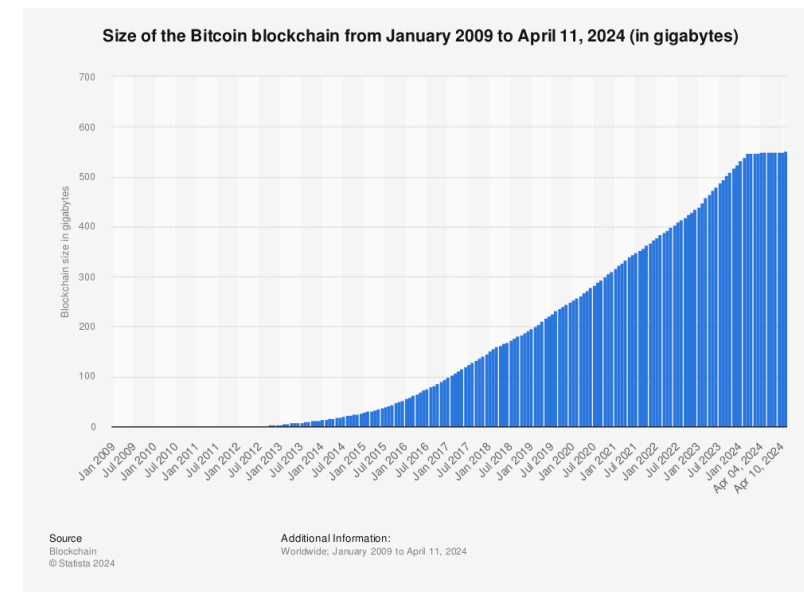
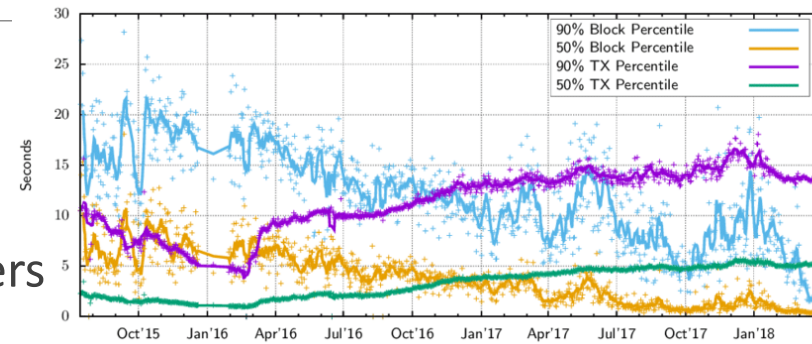
- You need a real large network to prevent the 51% attack – **not at all suitable for enterprise applications**

**The Ugly:** Low transaction throughput, Overuse of computing power !!

- (Bitcoin) 3.3 to 7 transactions per second, (Ethereum) ~15 transactions per second
- Millions of miners – thousands tries, but only one gets the success

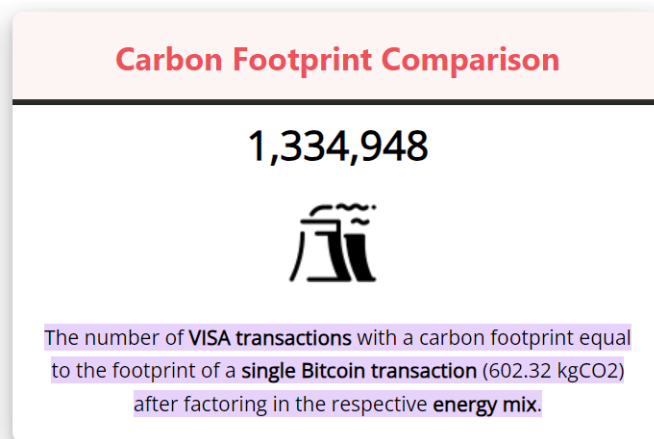
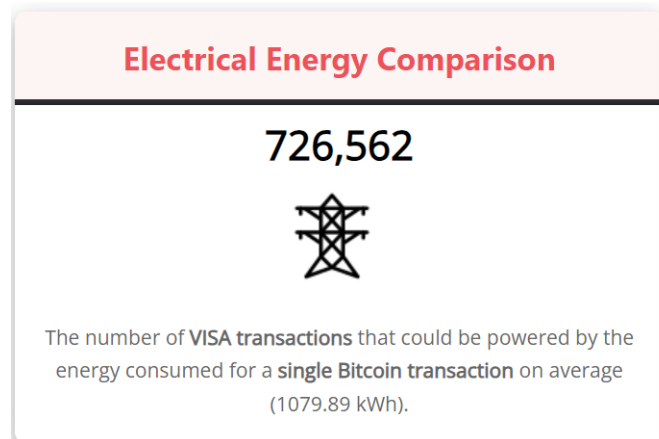
# Remarks on the Bitcoin Network









- P2P network
- Initially, you connect to the seed node
- By message passing you query nodes and connect to random peers
- If a new transaction, flood the network
- Blockchain size
- Fully validating nodes vs Lightweight nodes
- Size of the network (~18000 fully validating nodes)



# Limitations for Bitcoin

- on the total number of Bitcoins (by 2140)
- On small transaction output
  - on the size of the block
  - on the difficulty of the hash puzzle
- Energy



	Crypto Currency	kWh per transaction
	Bitcoin	1173
	Ethereum	87.29
	Bitcoin Cash	18.957
	Litecoin	18.522
	Ethereum 2.0*	0.8729*
	Cardano	0.547
	Dogecoin	0.12
	XRP	0.0079

\*Ethereum 2.0 will use 90% less energy

# Bitcoin Energy Consumption

## Bitcoin Energy Consumption

— Estimated TWh per Year    - - - Minimum TWh per Year

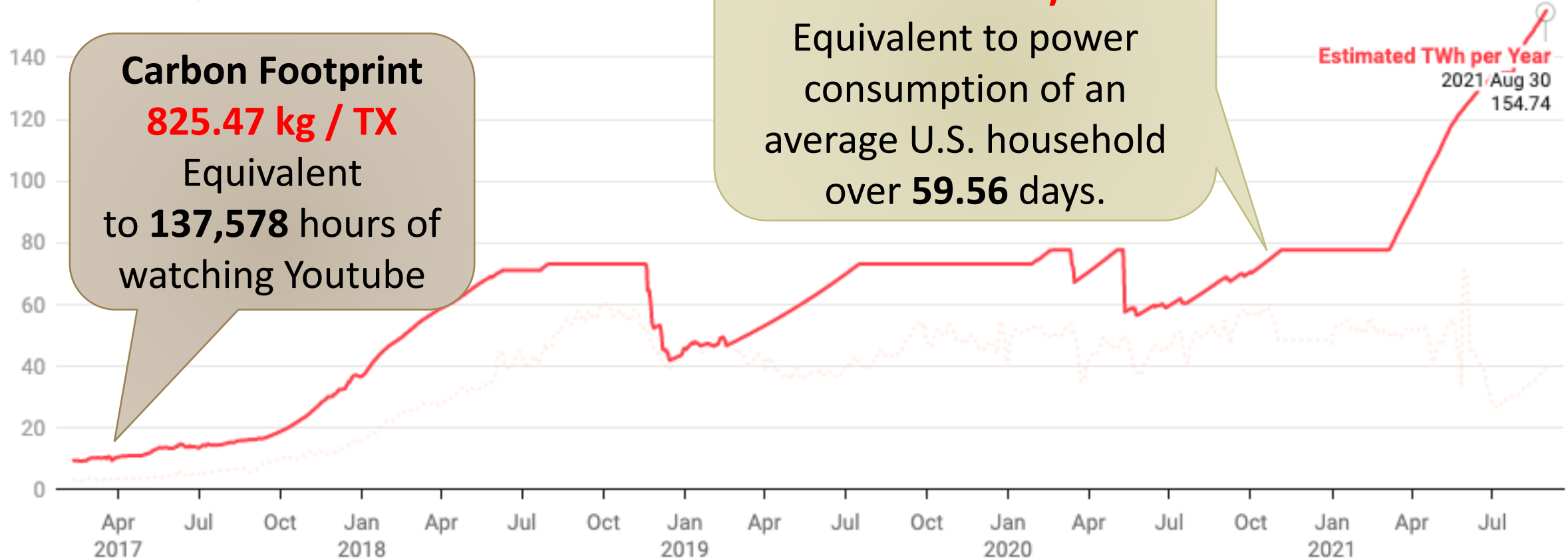


Image Source: Digiconomist Bitcoin Energy Consumption Index

# Other Consensus Mechanisms

---



# Proof of Stake (PoS)

---

Possibly proposed in 2011 by a Member in Bitcoin Forum -

<https://bitcointalk.org/index.php?topic=27787.0>

- Make a transition from PoW to PoS when bitcoins are widely distributed

PoW vs PoS

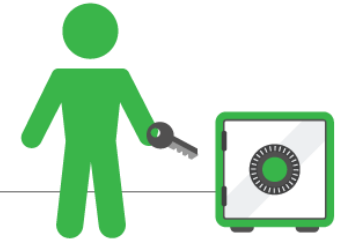
- **PoW**: Probability of mining a block depends on the work done by the miner
- **PoS**: Amount of bitcoin that the miner holds – Miner holding 1% of the Bitcoin can mine 1% of the PoS blocks.

Provides increased protection

- Executing an attack is expensive, you need more Bitcoins
- **Reduced incentive for attack** – the attacker needs to own a majority of bitcoins – an attack will have more affect on the attacker

# Proof-of-Stake (PoS)

---



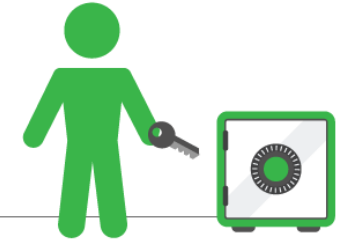
The right to mine blocks is given out randomly, but proportionally, based on 'stake'

Stake is defined as some form their share or involvement in the network

- Often the amount of the currency owned
- Example: If you owned 10% of all of the given coin, you could expect to win the right to mine 10% of all blocks

# Proof-of-Stake (PoS)

---



The chosen miners still do some form of guess-and-check to create the block:

- They try various combinations of features of their address and wallet, and previous block variables
- The number of combinations possible is based on their stake, hence why larger stakeholders have higher chances of successfully mining the block
- These combinations are quickly exhausted, making PoS significantly less computationally intensive

# Proof-of-Stake (PoS)

---

The miners are incentivized to only provide valid blocks, as they have great incentive to keep the network functioning correctly (their stake or holdings will be worthless if the network fails to function)

- Some implementations demand that miners put their coins into escrow that is lost if they break the rules

# Proof-of-Stake (PoS)

---

The block is validated as usual by the rest of the network before they continue to the next block

There are many variations on Proof of Stake (often named something slightly different), and the mechanisms by which rewards are distributed, validators are selected, and stake is determined

# PoS Strengths

---

**No useless mining:** there is no unnecessary use of resources to further power the blockchain

**Little to no hardware advantage**

- ASIC mining pools do not have a significant advantage over a powerful home computer

**Those 'guarding' the value of the coins have the most to lose if the network is compromised**

- The incentives to be honest are aligned with individuals motives

# PoS Strengths

---

The 51% attacks becomes essentially infeasible

- An attacker would need to accumulate 51% of all the coins on the network to accomplish this
- Currently for Ethereum this is \$6 Billion, which would be lost if the attack were successful

Proof of Stake has the potential to be magnitudes more efficient than PoW, making it significantly more scalable

- Very high transaction throughput is possible with PoS

# PoS Drawbacks

---

Theoretically encourages centralization:

- Higher stake means higher rewards, keeping the 'rich' richer

'Nothing at stake attack'

- Since forwarding the blockchain costs effectively nothing (compared to PoW), nodes are actually encouraged to work on every possible fork at once, as doing so increases the chance that they receive part of the reward in the event that the forked chain becomes longer
- Results in consensus being difficult to reach, or unreachable



# PoS Drawbacks

---

PoS is often claimed to be not as secure as PoW

- There are many implementations of various 'claimed' security, and most of these just need to stand the test of time to be considered more secure

Some implementations are vulnerable to a Sybil attack

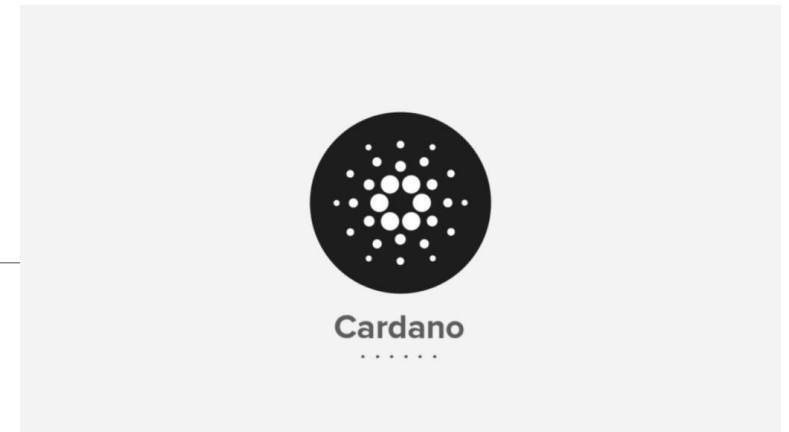
# Current PoS Systems

Ethereum (after Casper)

Cardano

Waves

Peercoin



# Casper: Ethereum Serenity

---

Serenity is the final planned phase of Ethereum (more may come later)

The major improvement in Serenity is Casper: a new consensus protocol transitioning the Ethereum network from Proof-of-Work to Proof-of-Stake

# Casper: Ethereum Serenity

---

Instead of mining nodes, Casper has **Validator** nodes, which are responsible for driving consensus

To become a validator, you must 'stake' some of your ethereum in a process called 'bonding'

- This serves as an escrow, and will be forfeited if your actions are malicious on the blockchain
- If you do not meet the minimum amount of ETH required (~1500 ETH), you may join a staking pool

# Casper: Ethereum Serenity

---

Validators place 'bets' on blocks: if the block is eventually validated, they win a reward, if the block isn't validated, they pay a penalty

- Thus, validators are incentivized to only vote on valid blocks that will become part of the dominant chain
- A block is confirmed once the 'bets' on a block converge to infinity

# Casper: Ethereum Serenity

---

Validators place 'bets' on blocks: if the block is eventually validated, they win a reward, if the block isn't validated, they pay a penalty

- Example: I am willing to take a bet with 99999:1 odds that block #2 in the Ethereum chain will not change

Validators are also responsible for making blocks, which functions similar to other consensus protocols

# Delegated-Proof-of-Stake

---

Developed by Dan Larimer in 2013

This consensus model is aimed at modeling a digital democracy

Token holders (stake holders) can vote for witnesses

- The number of votes they can cast is proportional to their token holdings

# Delegated-Proof-of-Stake

---

- Witnesses are the block creators, and are paid transaction fees when they create a new block
- Witnesses can be voted out at any time, and thus will lose their income if they do not create new blocks, or create blocks that are not trustworthy



# Delegated-Proof-of-Stake

---

- In some cases, witnesses are rotated on a regular basis to give more people opportunity to participate
- Current projects include Bitshares, Steem, EOS (all Dan Larimer founded), Lisk and Ark



EOS



LISK



STEEM

# Other Consensus Algorithms

---

- *Practical Byzantine Fault Tolerance* (pBFT) - Hyperledger Fabric
- *Federated Byzantine Fault Tolerance* (fBFT) - Stellar
- *Delegated Byzantine Fault Tolerance* - Neo
- *Proof-of-Importance* (PoI) - NEM
- *Proof-of-Elapsed-Time* (PoET) - Hyperledger Sawtooth
- *Proof-of-Capacity* (PoC - aka P-o-Space)
- *Proof-of-Authority* (PoA)
- *Raft* (more classical consensus, not blockchain specific)
- ...

# Anonymity

---

# Privacy Implications

---

- No anonymity, only pseudonymity (no **unlinkability**)
  - It should be hard to link together different addresses of the same user.
  - It should be hard to link together different transactions made by the same user.
  - It should be hard to link the sender of a payment to its recipient.
  
- All transactions remain on the block chain– indefinitely!
  
- Retroactive data mining
  - Target used data mining on customer purchases to identify pregnant women and target ads at them (NYT 2012), ended up informing a woman's father that his teenage daughter was pregnant
  - Probably not a true story but it gives a measure of what credit card companies could do with the data

# Zerocoin

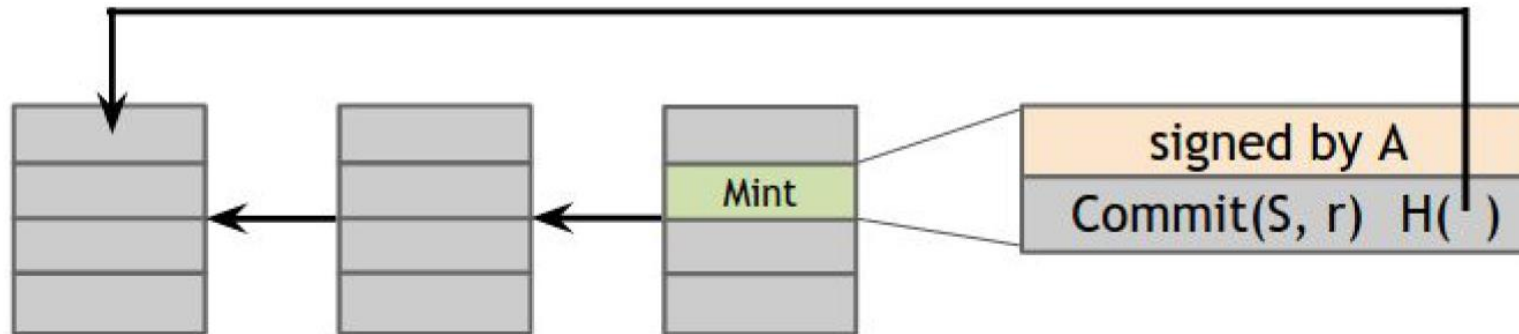
---

- A distributed approach to private electronic cash
- Extends Bitcoin (or any basecoin) by adding an anonymous currency on top of it
- Zerocoins are exchangeable for basecoins

# Minting a Zerocoin

---

1. Generate serial number  $S$  and a random secret  $r$
2. Compute  $Commit(S, r)$ , the commitment to the serial number
3. Publish the commitment onto the block chain as shown below. This burns a basecoin, making it unspendable, and creates a Zerocoin. Keep  $S$  and  $r$  secret for now.

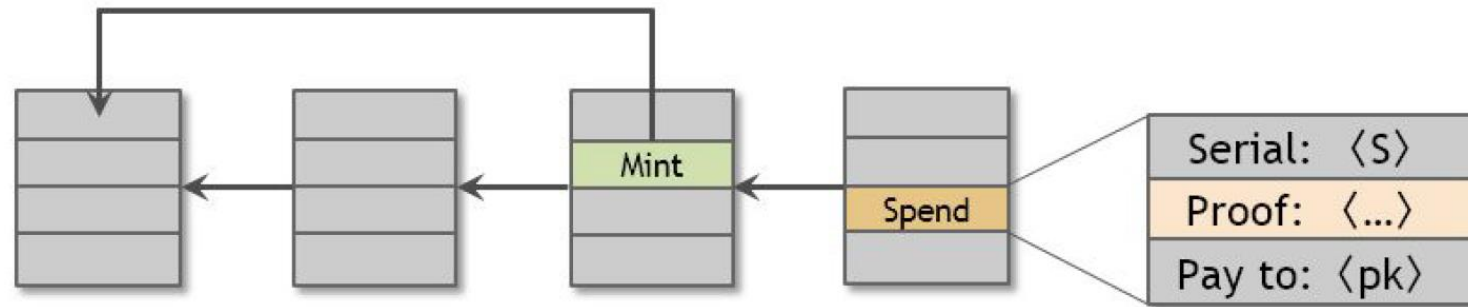


# Spending a Zerocoin

1. Create a special “spend” transaction that contains  $S$ , along with a zero-knowledge proof of the statement:

“I know  $r$  such that  $Commit(S, r)$  is in the set  $\{C_1, C_2, \dots, C_N\}$ ”

2. Miners will verify your zero-knowledge proof which establishes your *ability* to open one of the zerocoin commitments on the block chain, without actually opening it.
3. Miners will also check that the serial number  $S$  has never been used in any previous spend transaction (since that would be a double-spend).
4. The output of your spend transaction will now act as a new basecoin. For the output address, you should use an address that you own.



# That's all Folks!

## Βιβλιογραφία

1. [Αλυσίδες Συστοιχιών \(BlockChain\). Κάλλιπος.](#)
2. [Bitcoin and Cryptocurrency Technologies](#)