Session #9

Consensus in Bitcoin

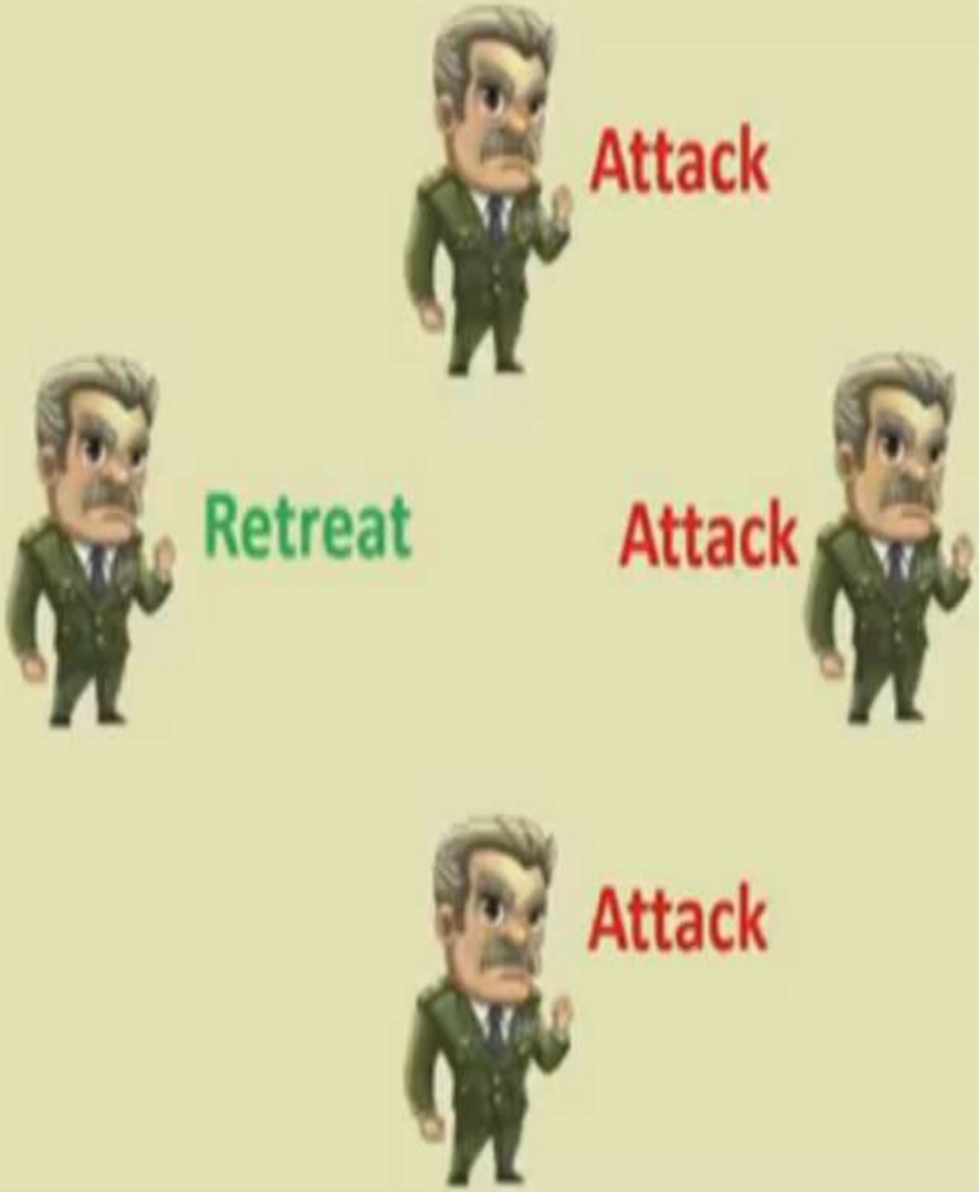
Agenda

- What is Consensus and Why Consensus is difficult?
- Distributed Consensus Properties
- Synchronous Message passing system and Asynchronous Message passing system
- Correctness of Distributed Consensus Protocol
- Consensus in Bitcoin Network

Consensus

- A procedure to reach in a common agreement in a distributed or decentralized multi-agent platform

- Important for a message passing system



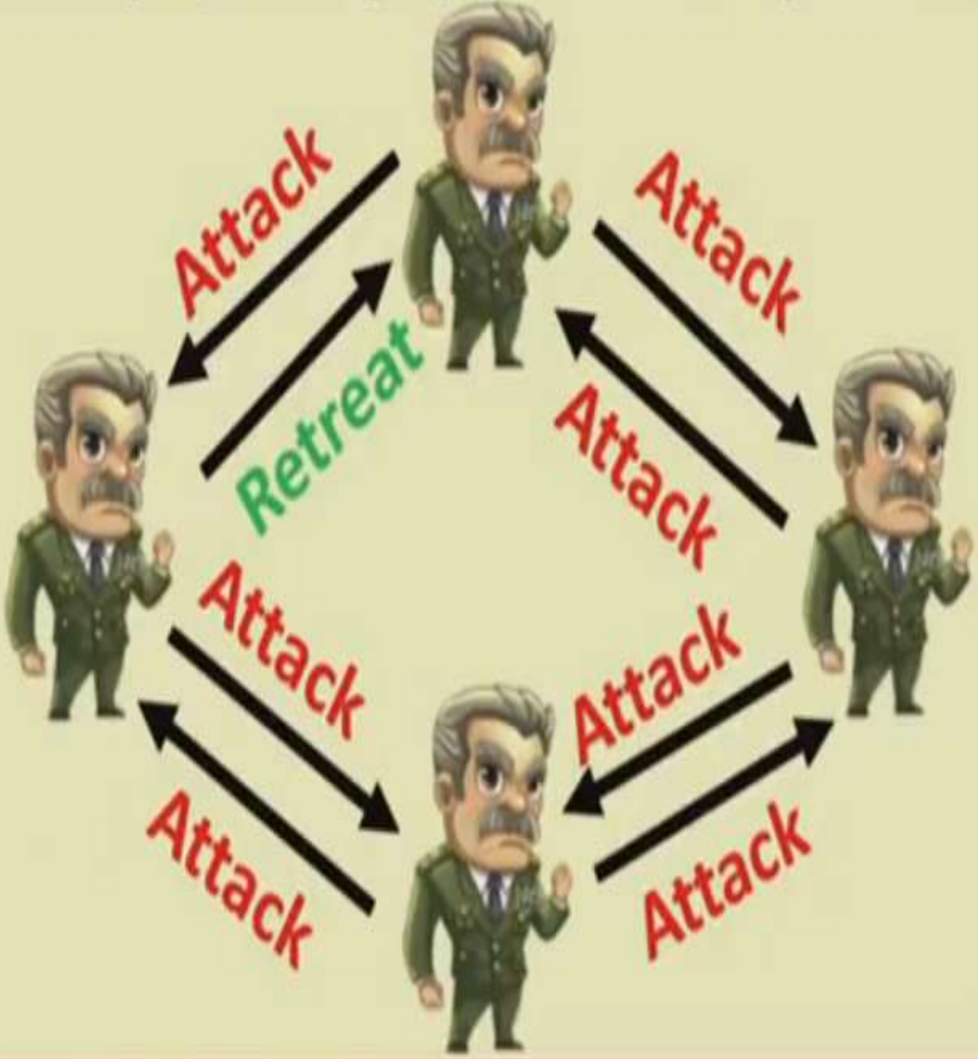
Why Consensus

- Reliability and fault tolerance in a distributed system
  - Ensure correct operations in the presence of faulty individuals

- Example:
  - Commit a transaction in a database
  - State machine replication
  - Clock synchronization

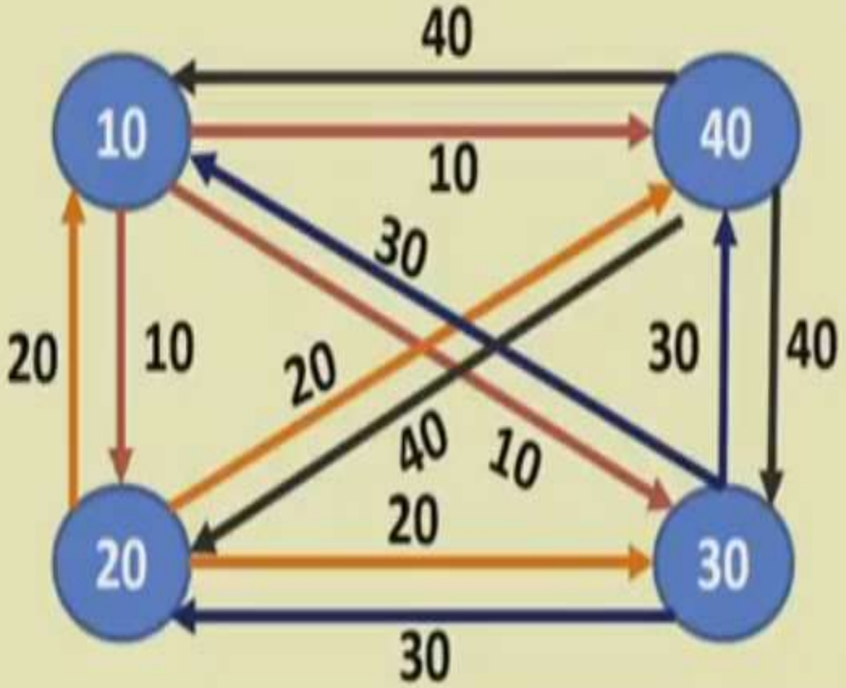
Why Consensus Can be Difficult in Certain Scenarios

- Consider a message passing system, and a general behaves maliciously



Distributed Consensus

- If there is **no failure**, it is easy and trivial to reach in a consensus
  - **Broadcast** the personal choice to all
  - Apply a **choice function**, say the maximum of all the values



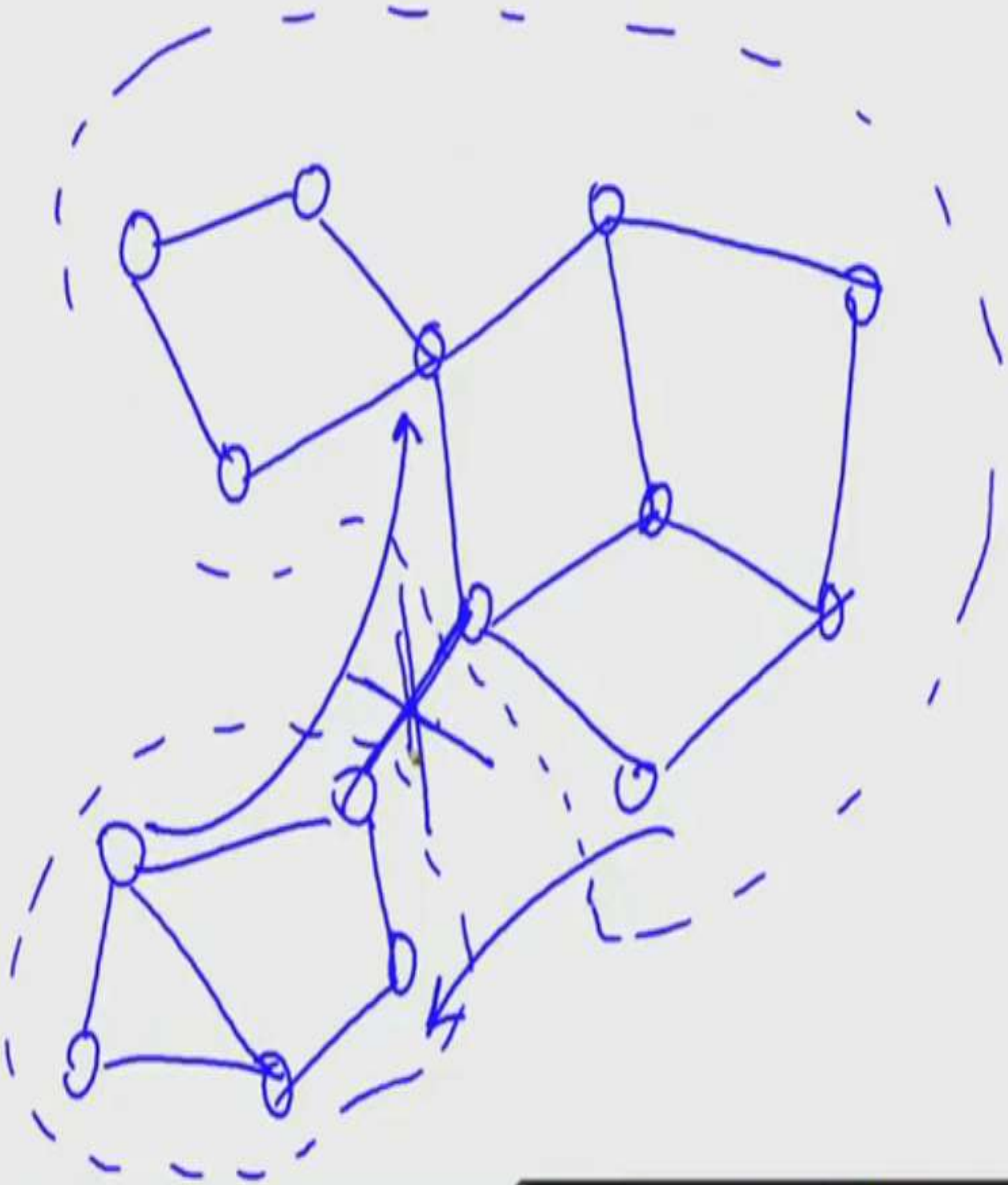
Distributed Consensus

- There can be various types of faults in a distributed system.

- **Crash Fault:** A node suddenly crashes or becomes unavailable in the middle of a communication

- **Network or Partitioned Faults:** A network fault occurs (say the link failure) and the network gets partitioned

- **Byzantine Faults:** A node starts behaving maliciously



Distributed Consensus - Properties

- **Termination**: Every correct individual decides some value at the end of the consensus protocol

- **Validity:** If all the individuals proposes the same value, then all correct individuals decide on that value

- **Integrity:** Every correct individual decides at most one value, and the decided value must be proposed by some individuals

- **Agreement:** Every correct individual must agree on the same value

Synchronous vs Asynchronous Systems

- **Synchronous Message Passing System:** The message must be received within a predefined time interval
  - Strong guarantee on message transmission delay

- **Asynchronous Message Passing System:** There is no upper bound on the message transmission delay or the message reception time
  - No timing constraint, message can be delayed for arbitrary period of times

Asynchronous Consensus

- **FLP85 (Impossibility Result):** In a **purely asynchronous distributed system**, the consensus problem is **impossible** (**with a deterministic solution**) to solve if in the presence of a **single crash failure**.
  - Results by Fischer, Lynch and Patterson (most influential paper awarded in ACM PODC 2001)
  - Randomized algorithms may exist

Synchronous Consensus

- Various consensus algorithms has been explored by the distributed system community
  - Paxos
  - Raft
  - Byzantine fault tolerance (BFT)

We'll look into these consensus algorithms, but later !!

Correctness of a Distributed Consensus Protocol

- **Safety:** Correct individuals must not agree on an incorrect value
  - Nothing bad happend

- **Liveliness** (or **Liveness**)**:** Every correct value must be accepted eventually
  - Something good eventually happens

Correctness of Distributed Consensus

- **Safety:** Correct individuals must not agree on incorrect value
  - Nothing bad happened
- **Liveliness (or Liveness):** Every correct value must be accepted eventually
  - Something good eventually happens

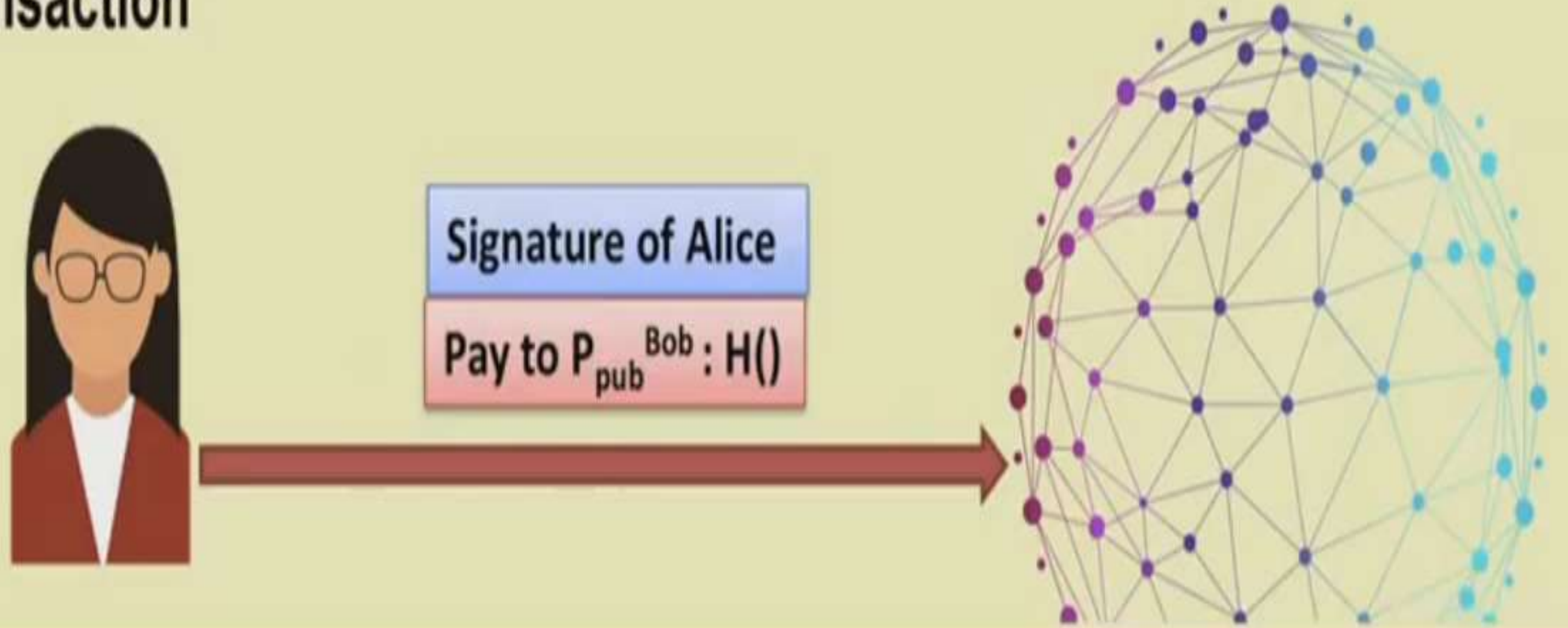
Consensus in an Open System

- The tradition distributed consensus protocols are based on
  - Message passing (when individuals are connected over the Internet)
  - Shared memory (when a common memory place is available to read and write the shared variables that everyone can access)

- Message passing requires a **closed** environment – everyone need to know the identity of others

Why Do We Require Consensus in Bitcoin Network

- Bitcoin is a peer-to-peer network
- Alice broadcast a transaction in this peer-to-peer network
- **All the nodes in this network need to agree on the correctness of this transaction**

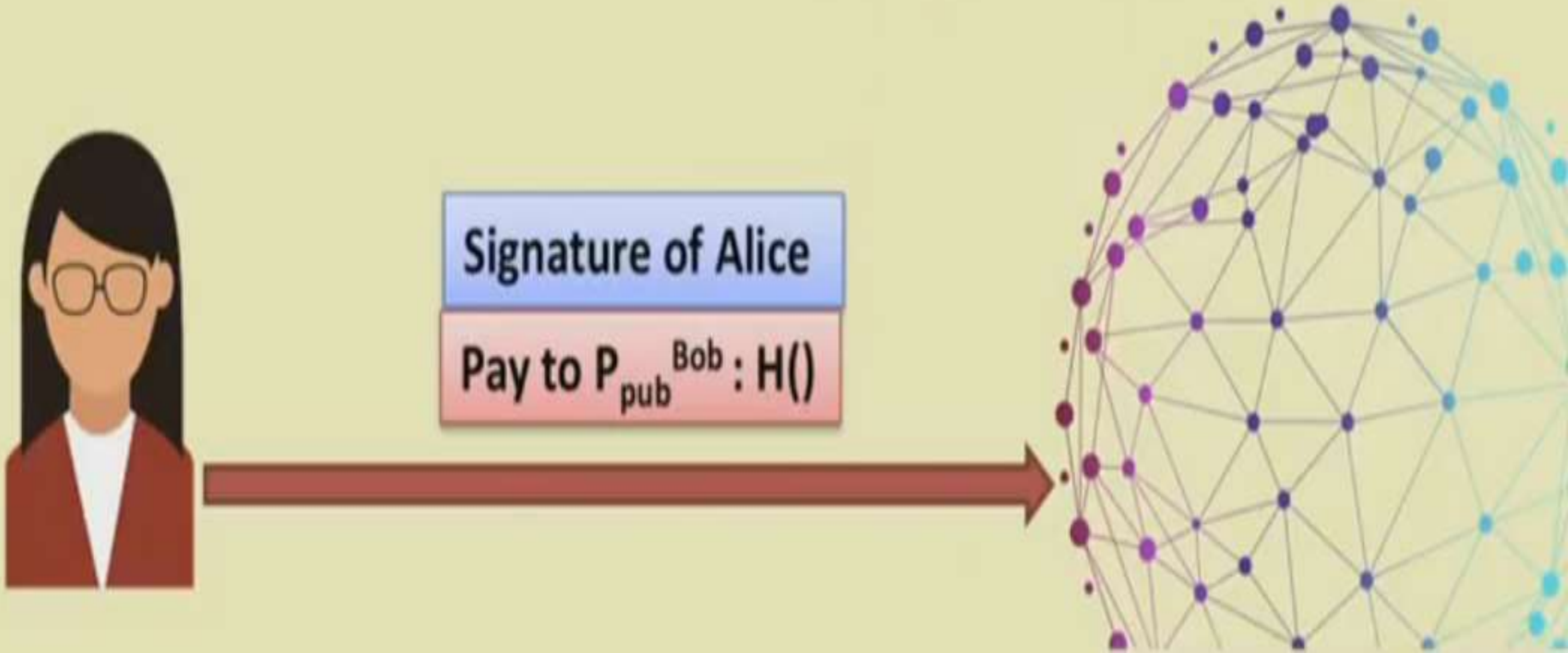


Signature of Alice

Pay to P_{pub}^{Bob} : H()

Why Do We Require Consensus in Bitcoin Network

- A node does not know all the peers in the network – this is an **open network**
- Some nodes can also initiate **malicious transactions**



Signature of Alice

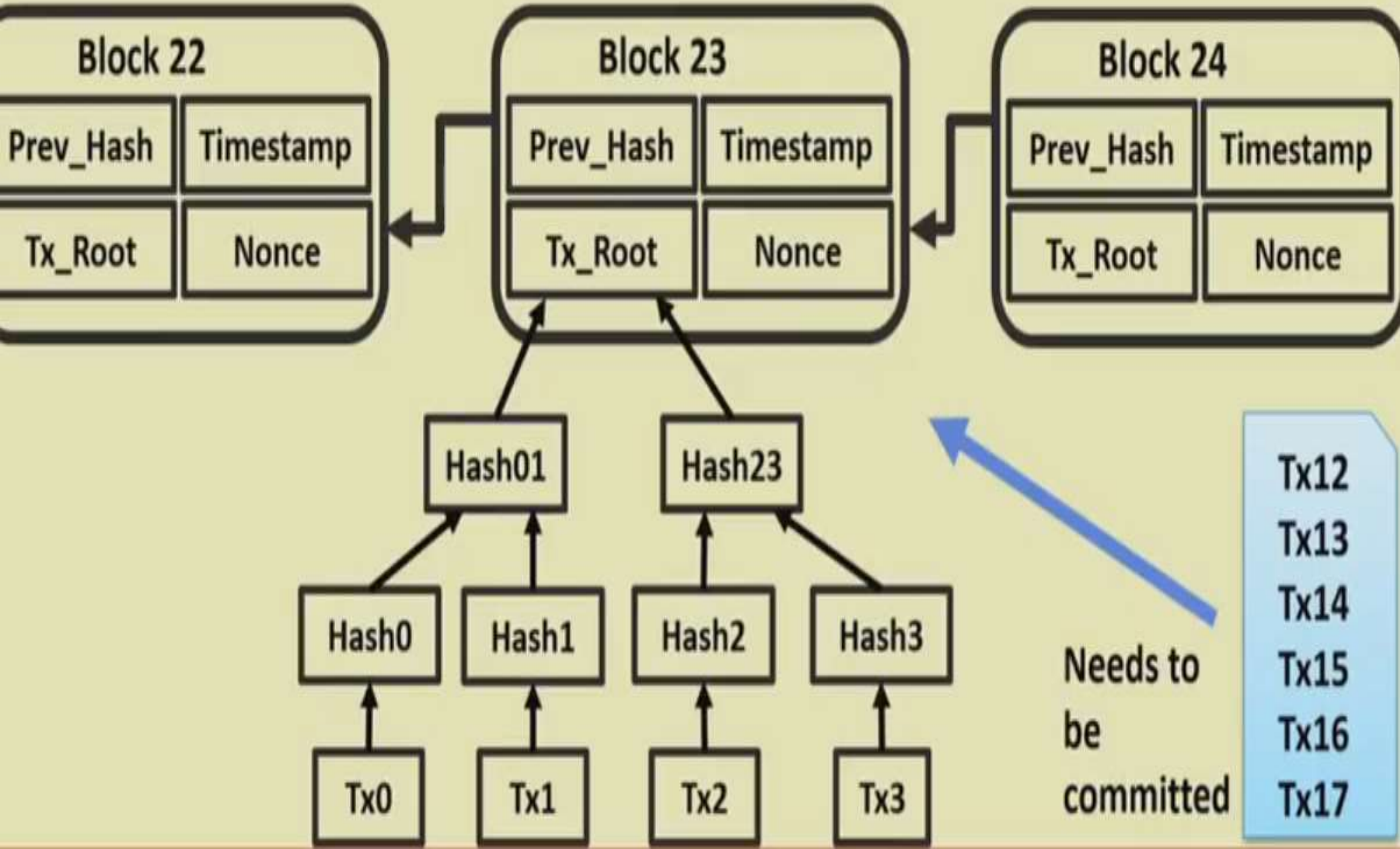
Pay to P_{pub}^{Bob} : H()

Consensus in a Bitcoin Network

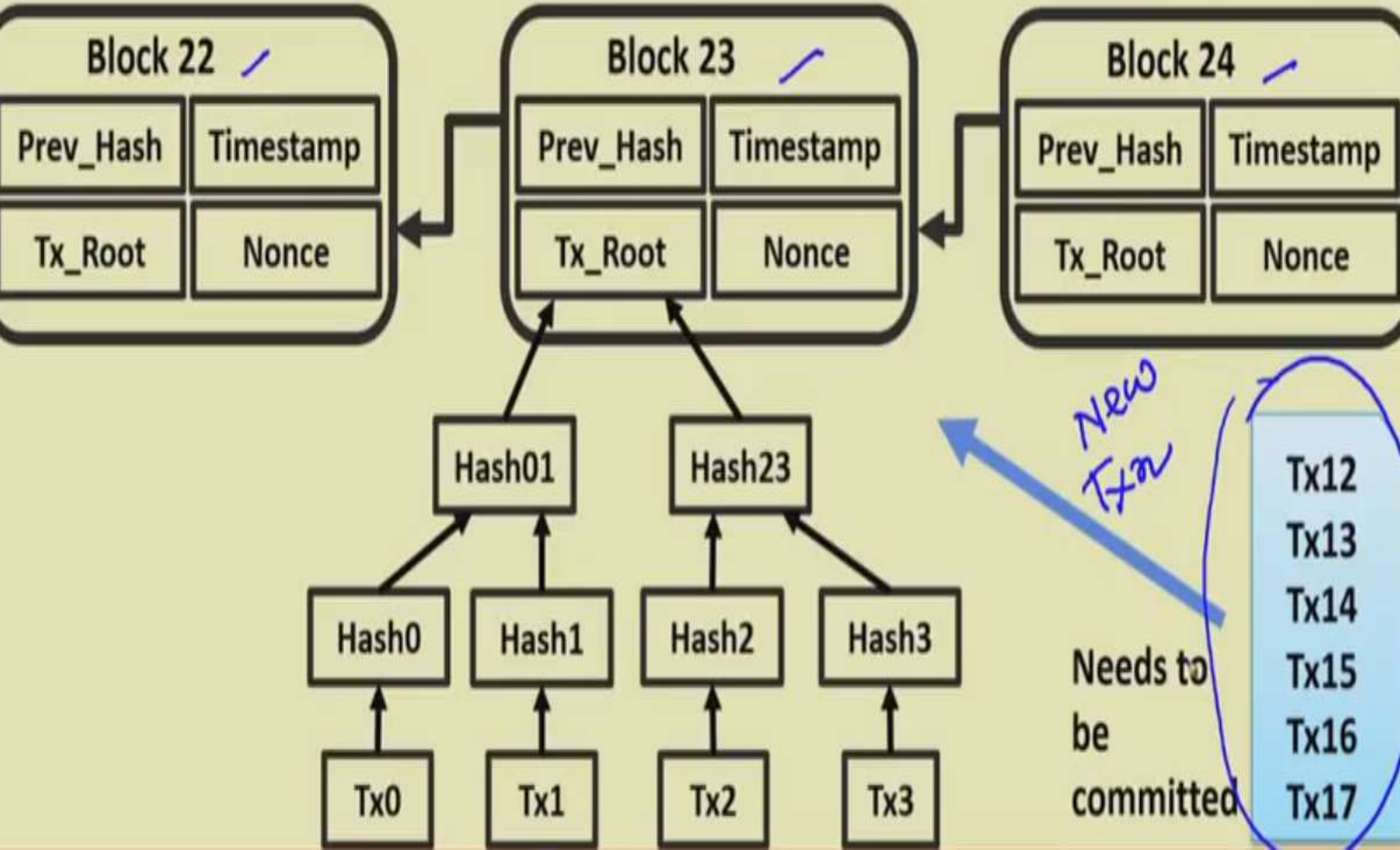
- Every node has **block of transactions** that has already reached into the consensus (**block of committed transactions**)

- The nodes also has a list of outstanding transactions that need to be validated against the block of committed transactions

Consensus in a Bitcoin Network



Consensus in a Bitcoin Network



Needs to be committed

Consensus in Bitcoin

- Per transaction consensus
  - Inefficient

- Block based consensus

New Block of Transactions

Tx12

Tx13

Tx14

Tx15

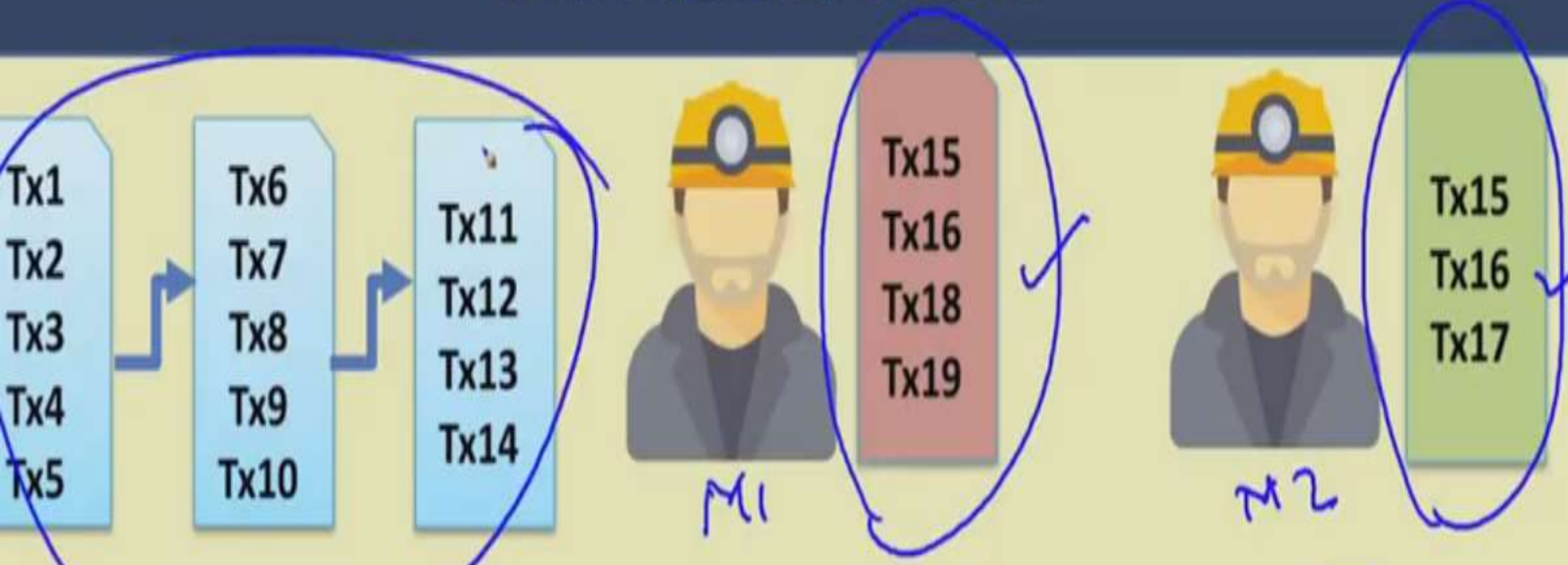
Tx16

Tx17

Apply consensus over the entire block of transactions

- Here comes the Blockchain

Consensus in Bitcoin



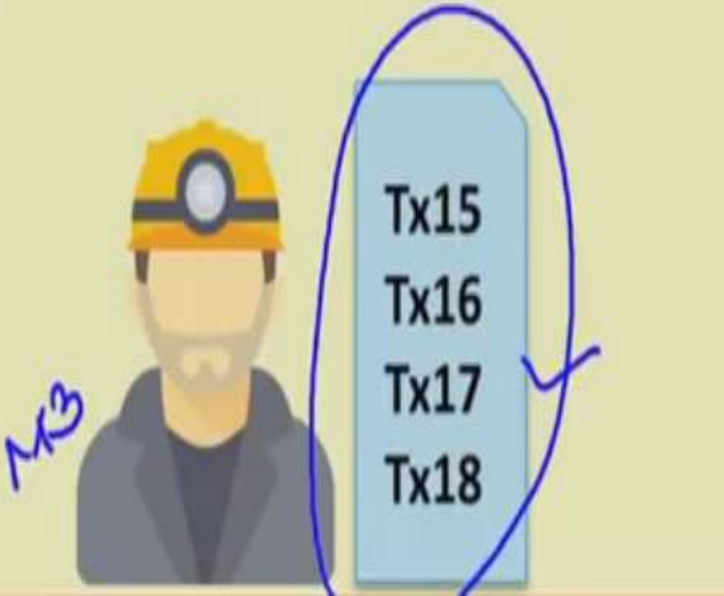
Tx1
Tx2
Tx3
Tx4
Tx5

Tx6
Tx7
Tx8
Tx9
Tx10

Tx11
Tx12
Tx13
Tx14

Tx15
Tx16
Tx18
Tx19

Tx15
Tx16
Tx17

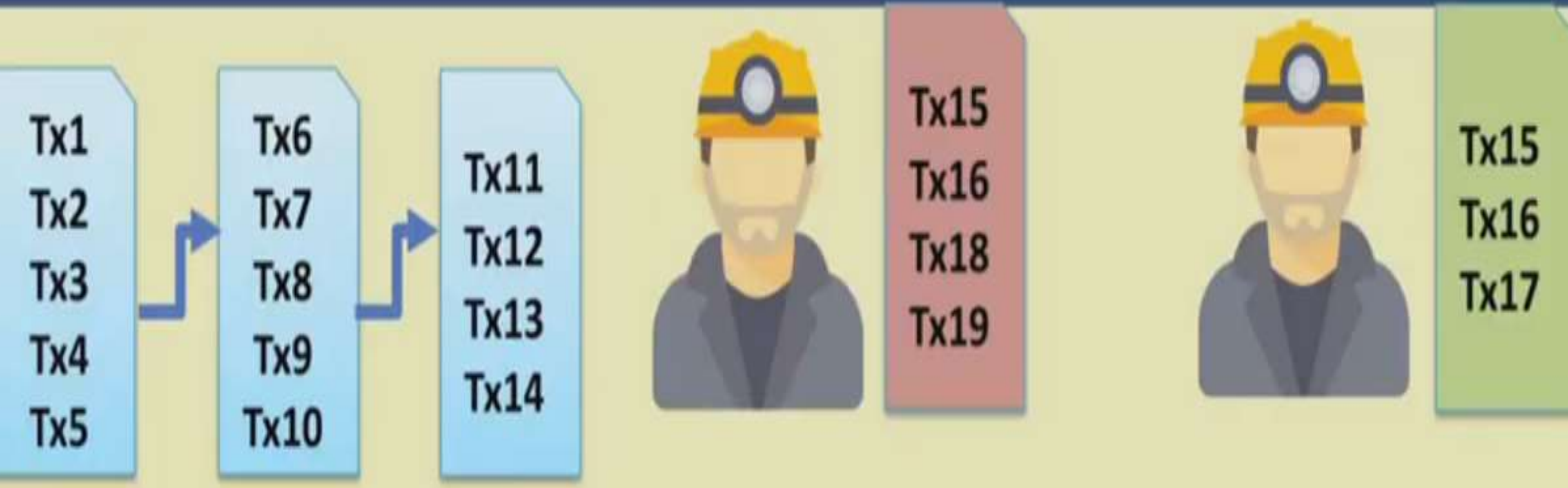


Tx15
Tx16
Tx17
Tx18

Bitcoin Consensus Objective:

Which block do we add next?

Consensus in Bitcoin

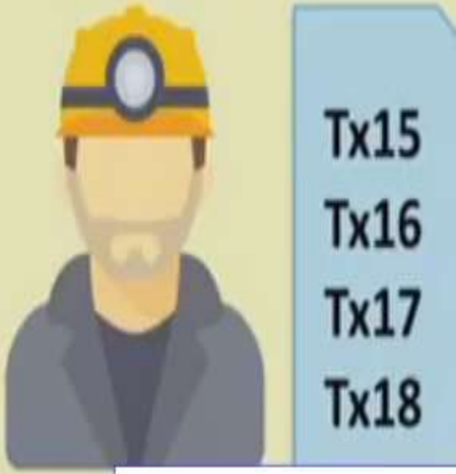


Bitcoin Consensus Objective:

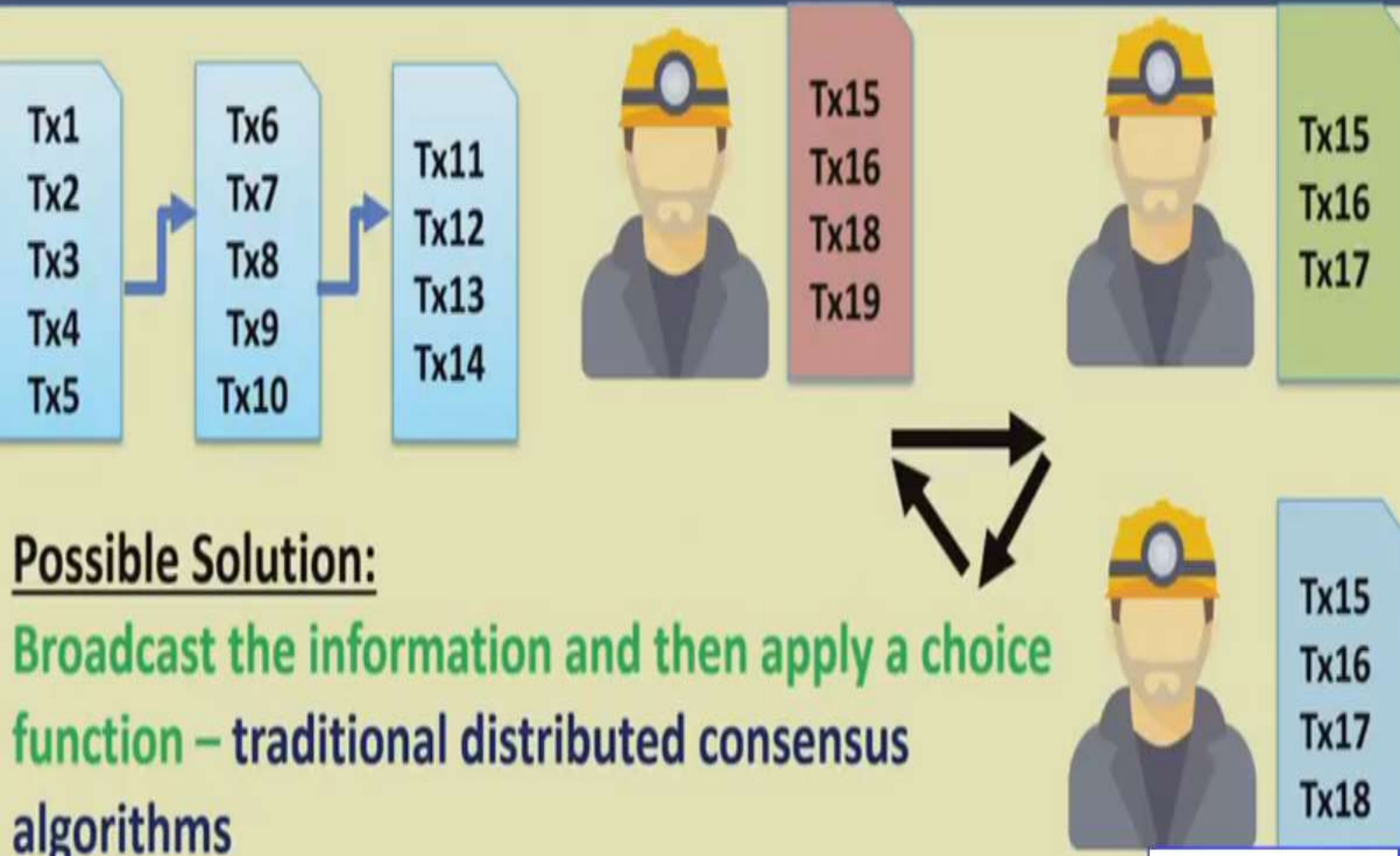
Which block do we add next?

Challenge:

The miners do not know each other



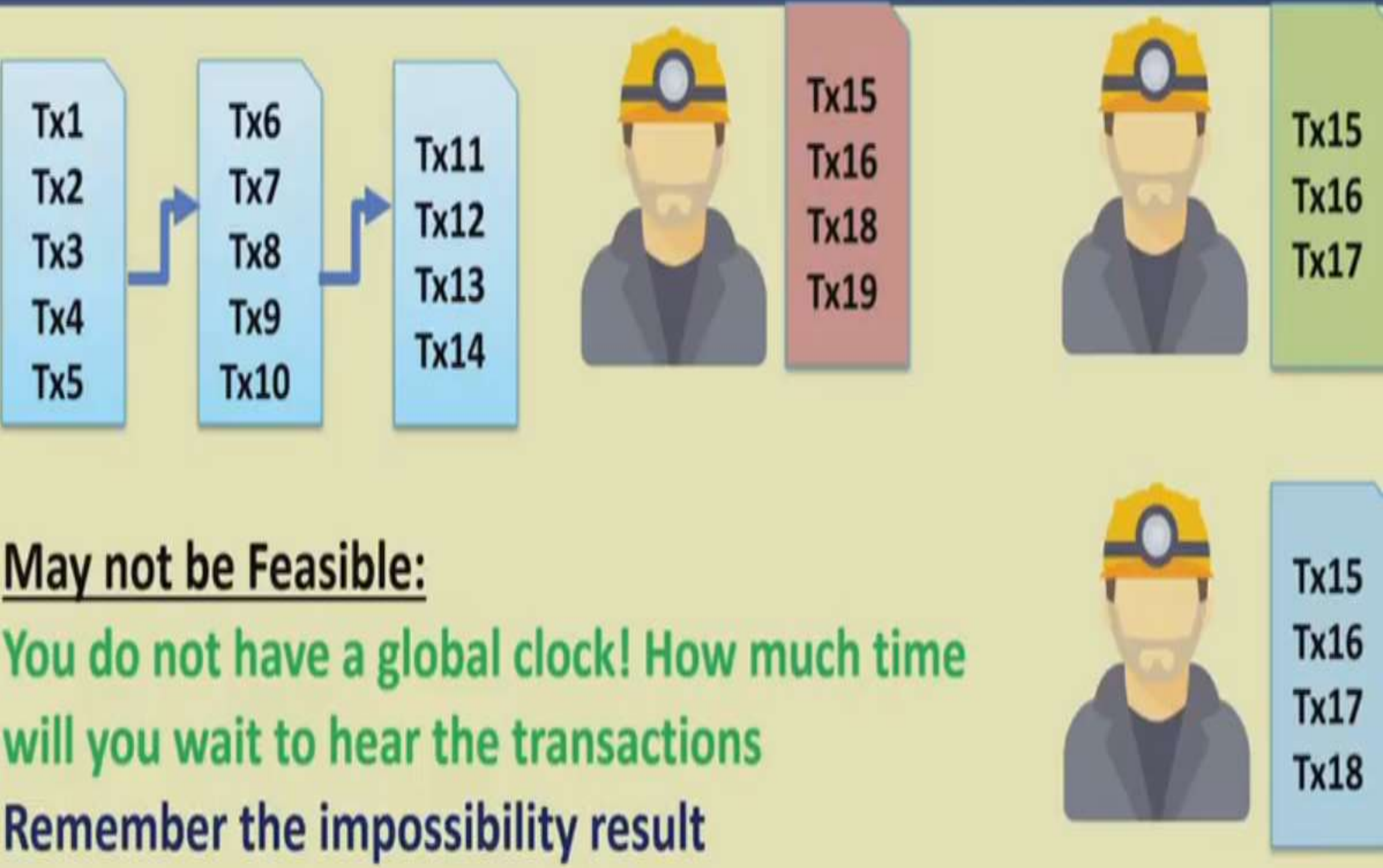
Consensus in Bitcoin



Possible Solution:

Broadcast the information and then apply a choice function – traditional distributed consensus algorithms

Consensus in Bitcoin



Tx1
Tx2
Tx3
Tx4
Tx5

Tx6
Tx7
Tx8
Tx9
Tx10

Tx11
Tx12
Tx13
Tx14

Tx15
Tx16
Tx18
Tx19

Tx15
Tx16
Tx17

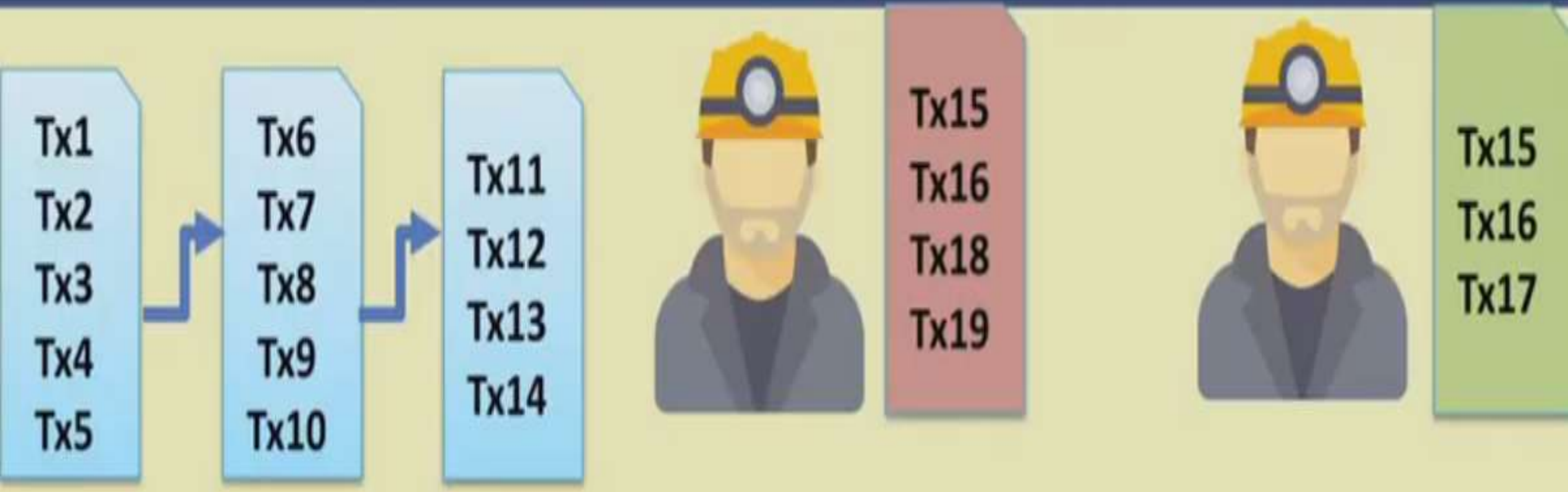
Tx15
Tx16
Tx17
Tx18

May not be Feasible:

You do not have a global clock! How much time will you wait to hear the transactions

Remember the impossibility result

Consensus in Bitcoin



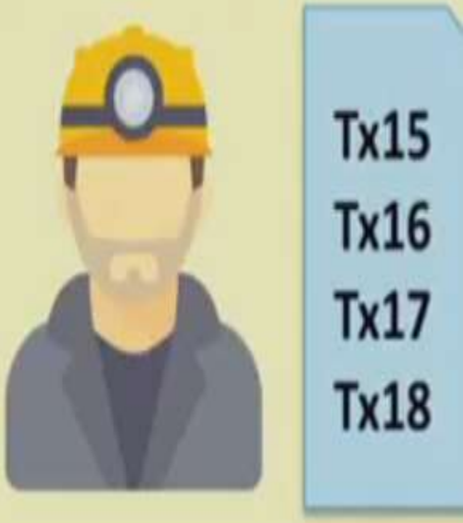
Tx1
Tx2
Tx3
Tx4
Tx5

Tx6
Tx7
Tx8
Tx9
Tx10

Tx11
Tx12
Tx13
Tx14

Tx15
Tx16
Tx18
Tx19

Tx15
Tx16
Tx17

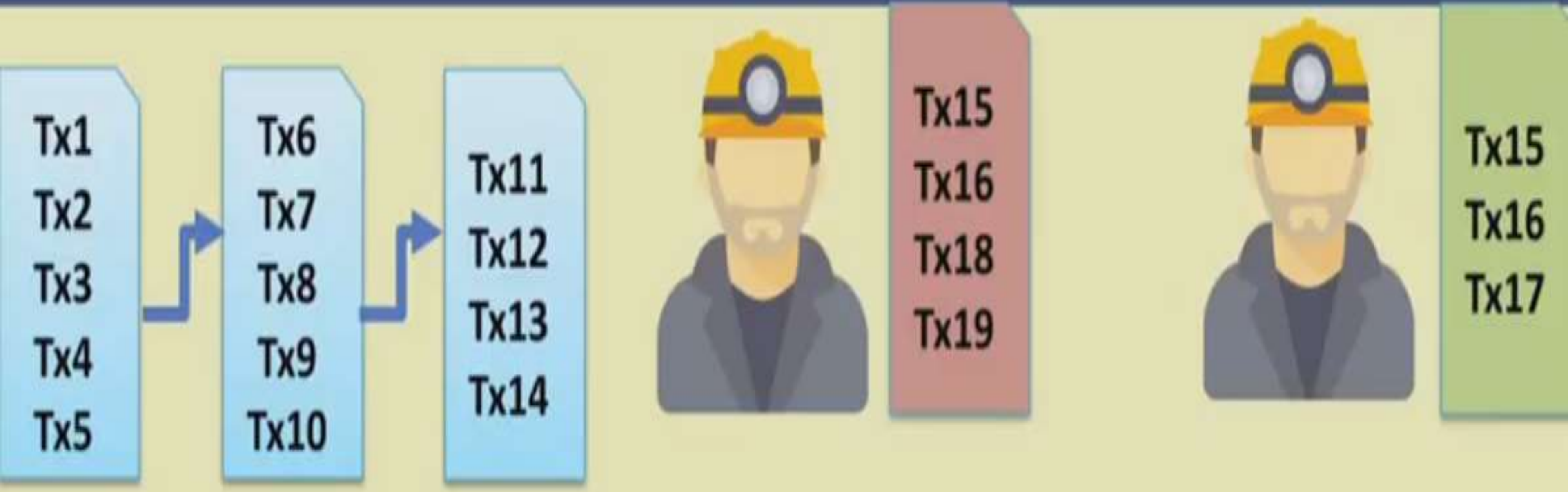


Tx15
Tx16
Tx17
Tx18

Observation - 1:

- Any valid block (a block with all valid transactions) can be accepted, even if it is proposed by only one miner

Consensus in Bitcoin



Tx1
Tx2
Tx3
Tx4
Tx5

Tx6
Tx7
Tx8
Tx9
Tx10

Tx11
Tx12
Tx13
Tx14

Tx15
Tx16
Tx18
Tx19

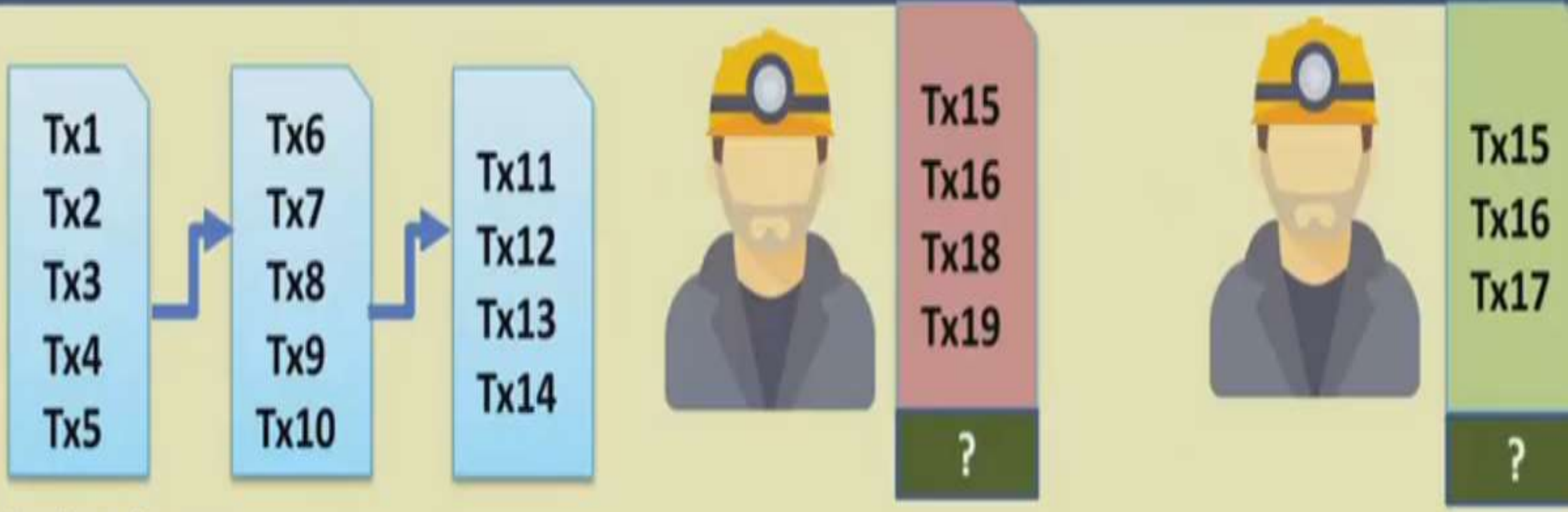
Tx15
Tx16
Tx17

Tx15
Tx16
Tx17
Tx18

Observation - 2:

- The protocol can work in rounds
  - Broadcast the accepted block to the peers
  - Collect the next set of transactions

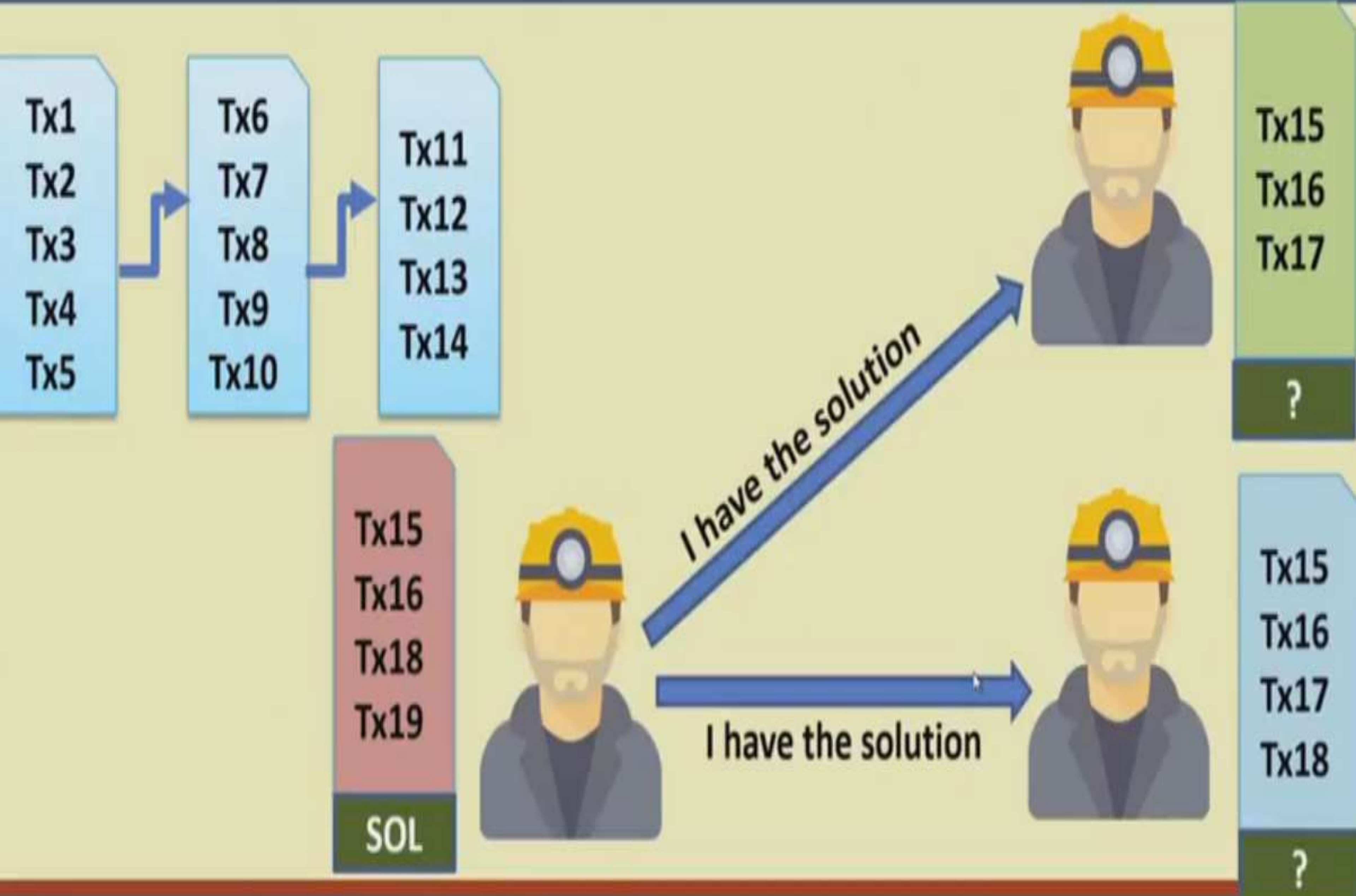
Consensus in Bitcoin



Solution:

- Every miner independently tries to solve a challenge
- The block is accepted for the miner who can prove first that the challenge has been solved

Consensus in Bitcoin



Tx1
Tx2
Tx3
Tx4
Tx5

Tx6
Tx7
Tx8
Tx9
Tx10

Tx11
Tx12
Tx13
Tx14

Tx15
Tx16
Tx18
Tx19

SOL

I have the solution

I have the solution

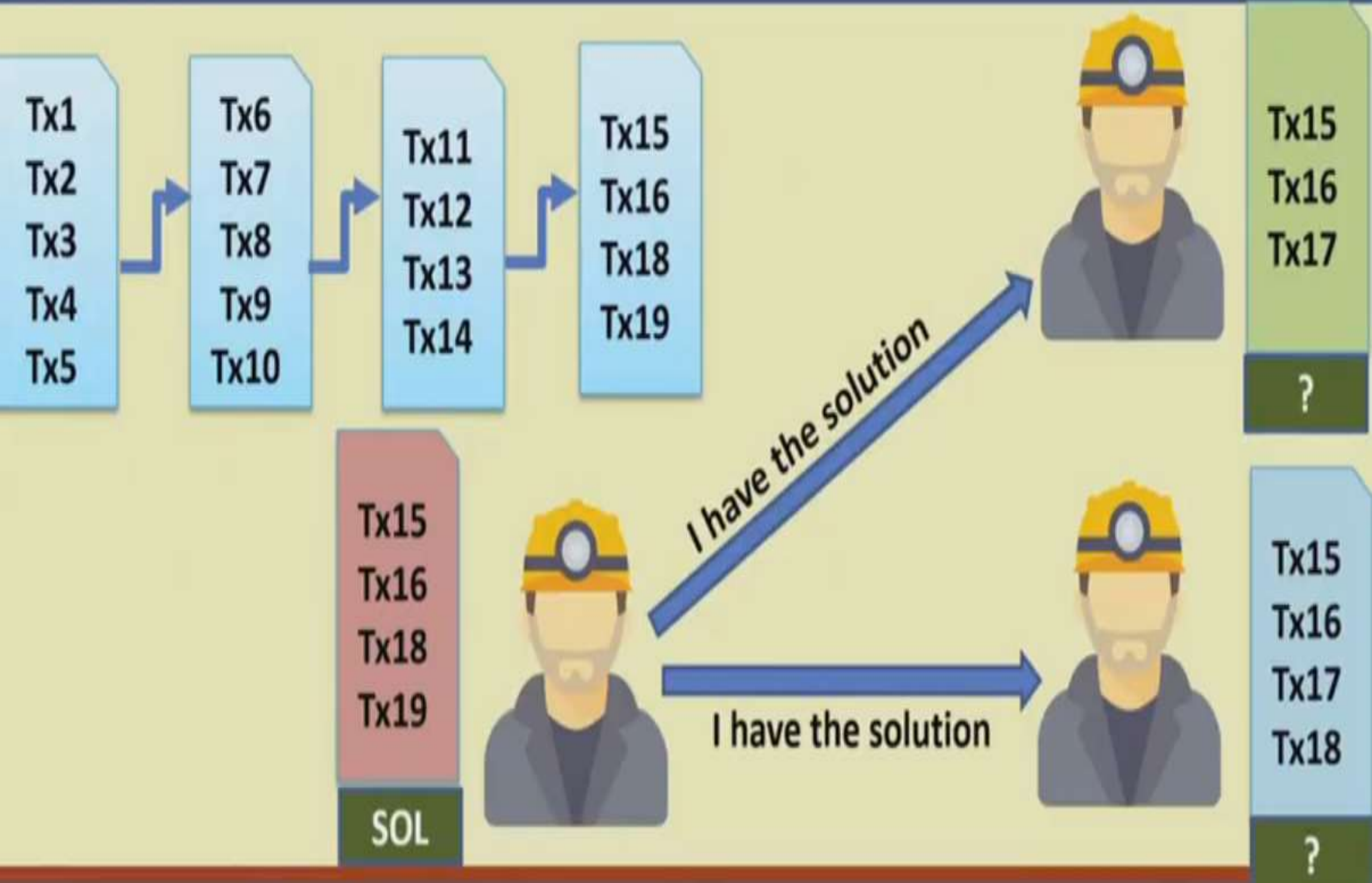
Tx15
Tx16
Tx17

?

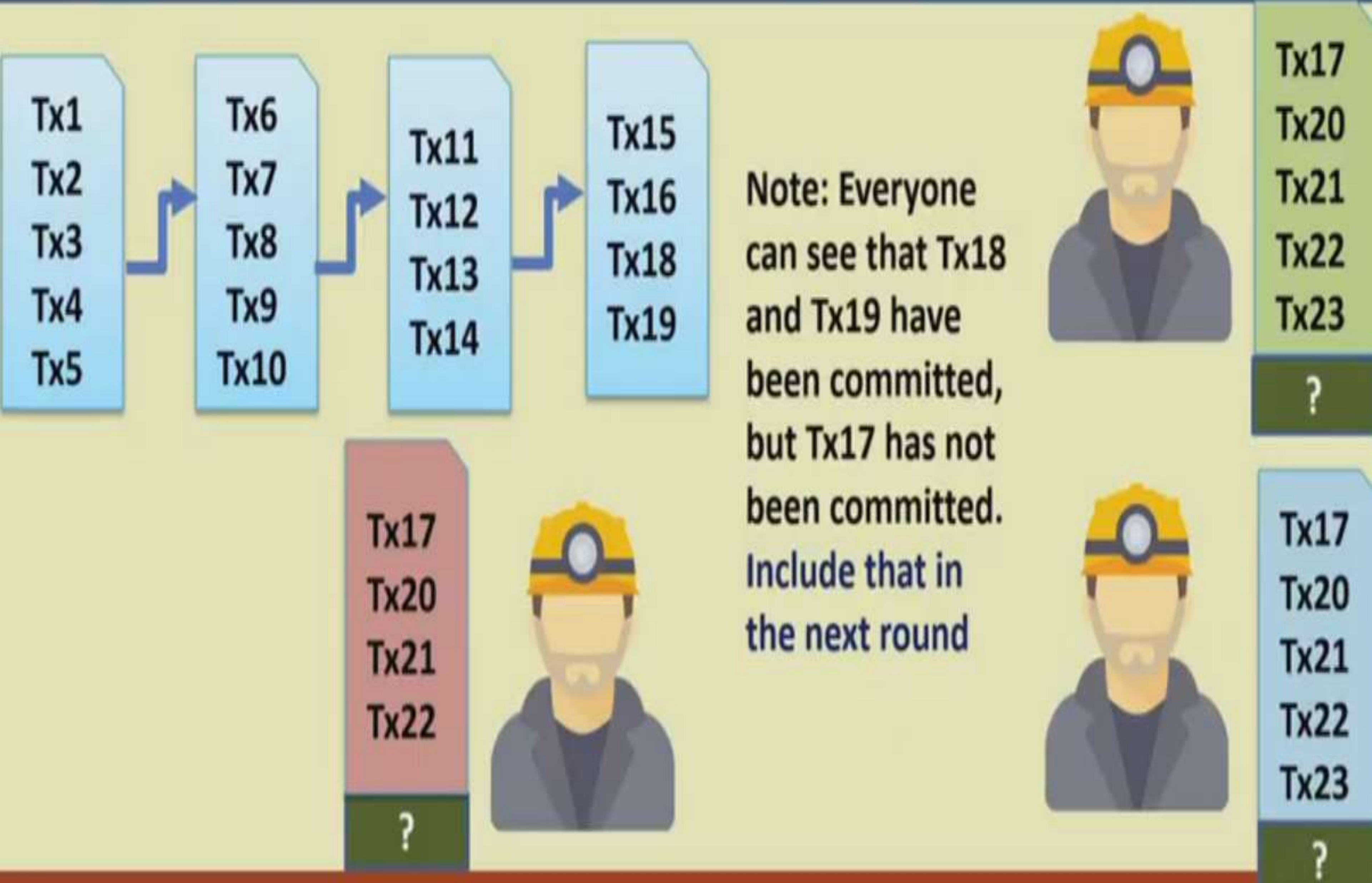
Tx15
Tx16
Tx17
Tx18

?

Consensus in Bitcoin



Consensus in Bitcoin



Tx1 Tx2 Tx3 Tx4 Tx5

Tx6 Tx7 Tx8 Tx9 Tx10

Tx11 Tx12 Tx13 Tx14

Tx15 Tx16 Tx18 Tx19

Tx17 Tx20 Tx21 Tx22

?

Tx17 Tx20 Tx21 Tx22 Tx23

?

Tx17 Tx20 Tx21 Tx22 Tx23

?

Note: Everyone can see that Tx18 and Tx19 have been committed, but Tx17 has not been committed. Include that in the next round

THANK YOU