

ThunderCore Consensus 101

PaLa Explained



@_kitchen

@thunderprotocol



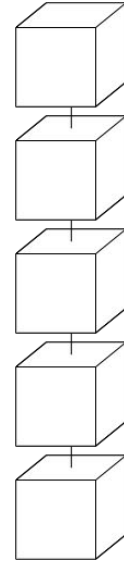
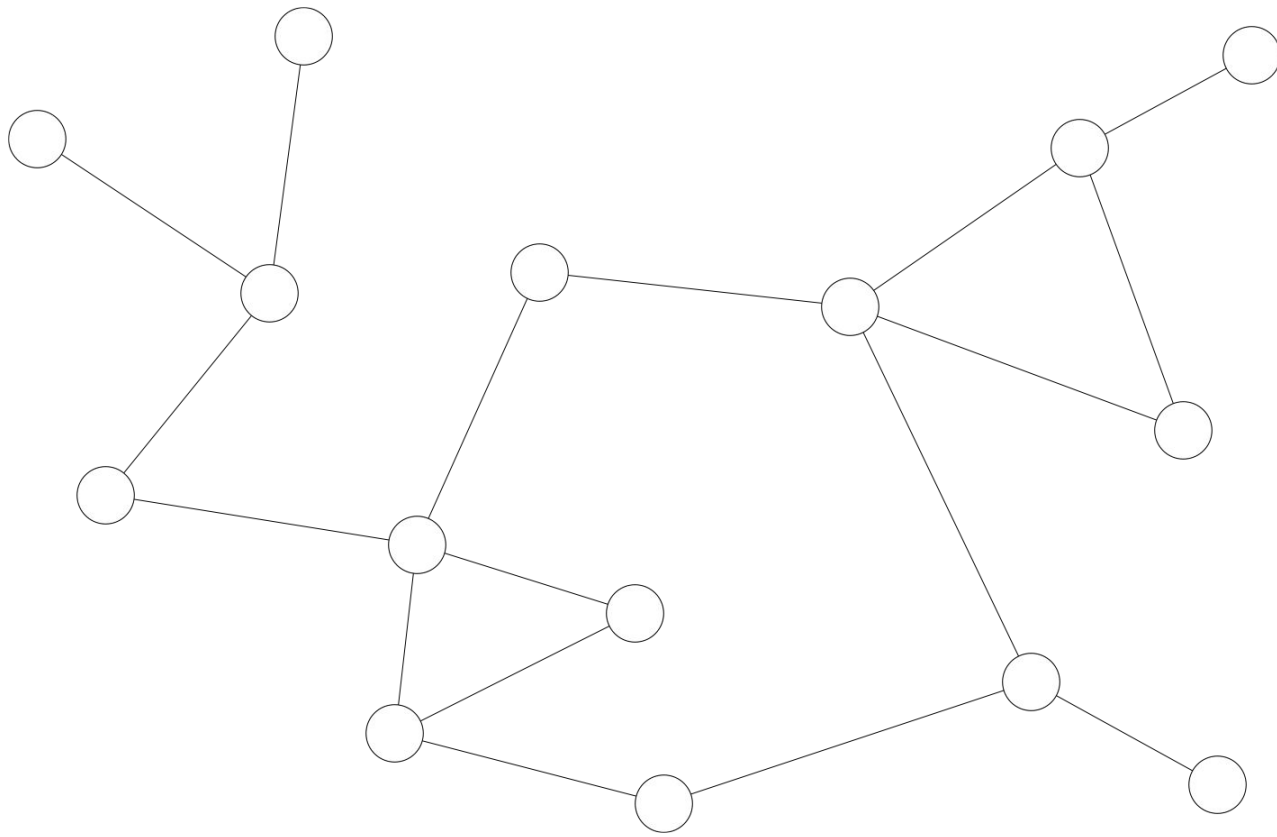
First generation blockchains such as bitcoin and ethereum are powered by the groundbreaking Proof-of-Work (PoW) consensus algorithm



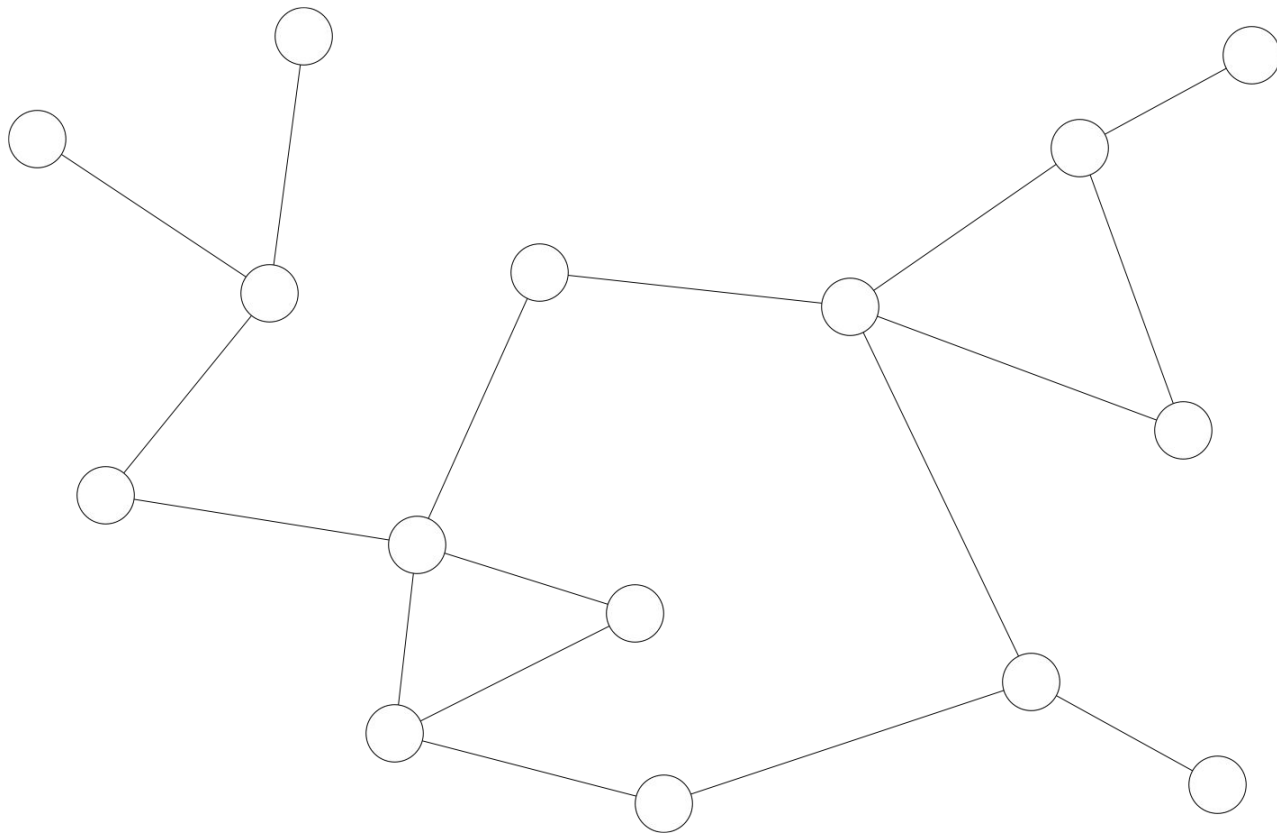
In PoW any node can join the distributed p2p network



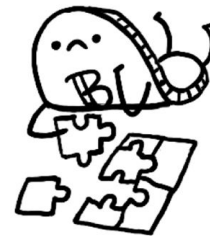
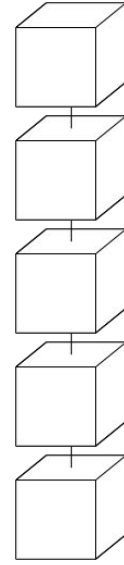
In PoW any node can join the distributed p2p network where they can collaborate

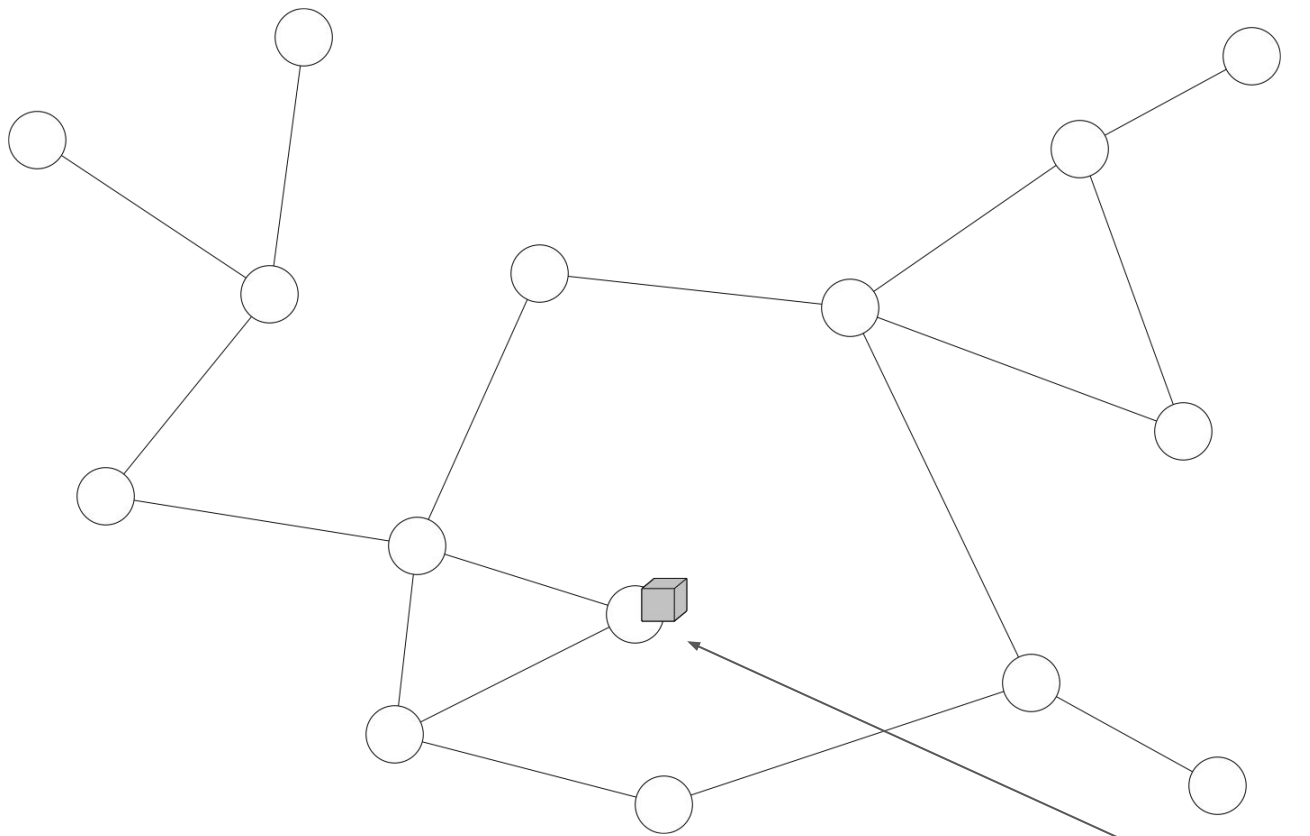


In PoW any node can join the distributed p2p network where they can collaborate to build a single global blockchain

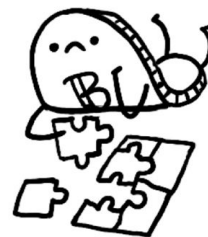
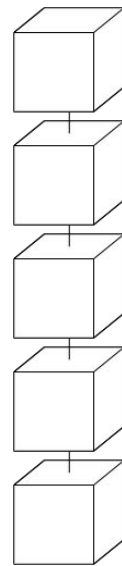


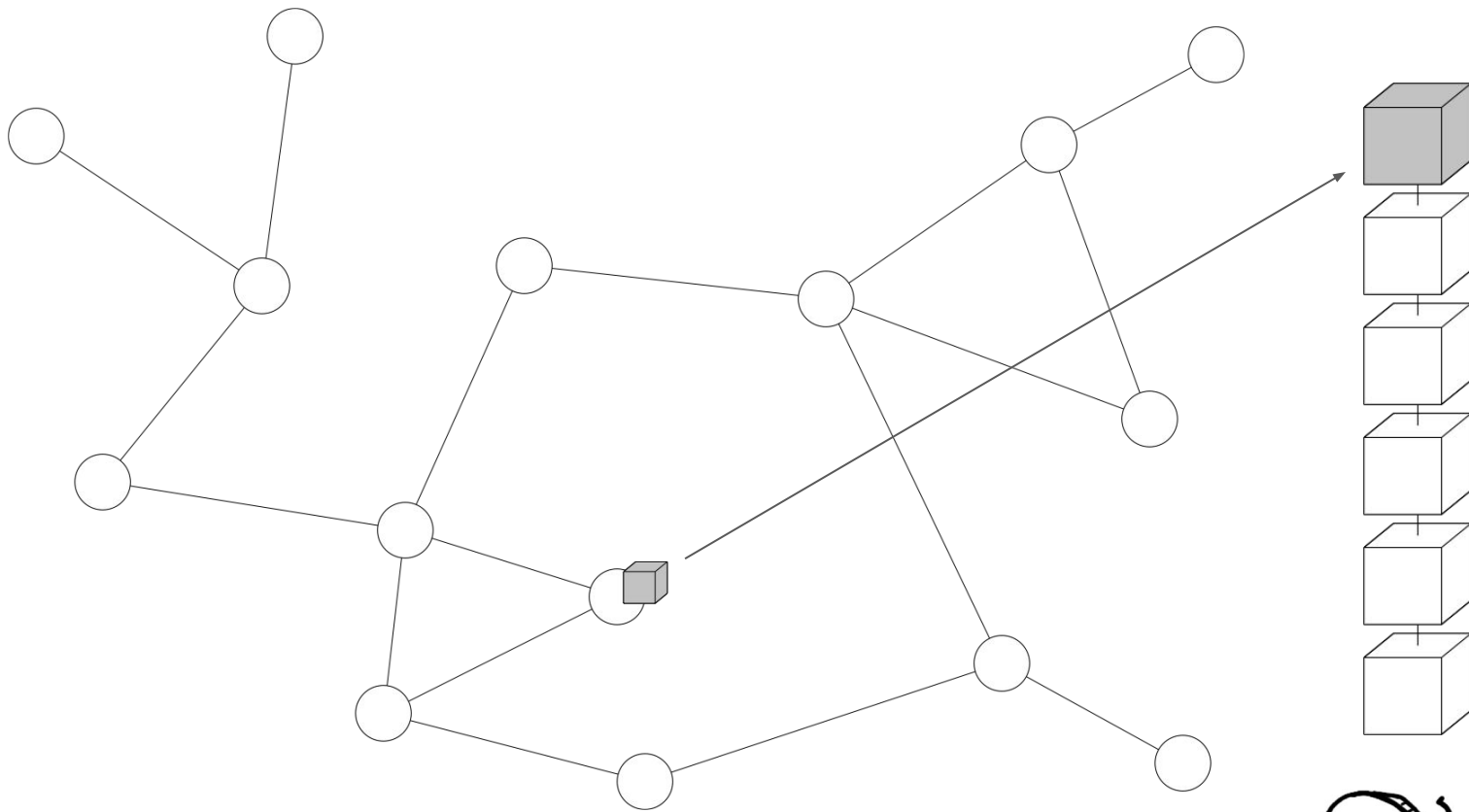
In order to extend the blockchain, nodes in the network compete to solve hard cryptographic puzzles



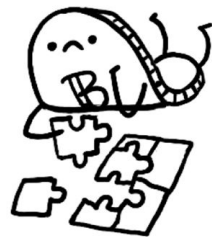


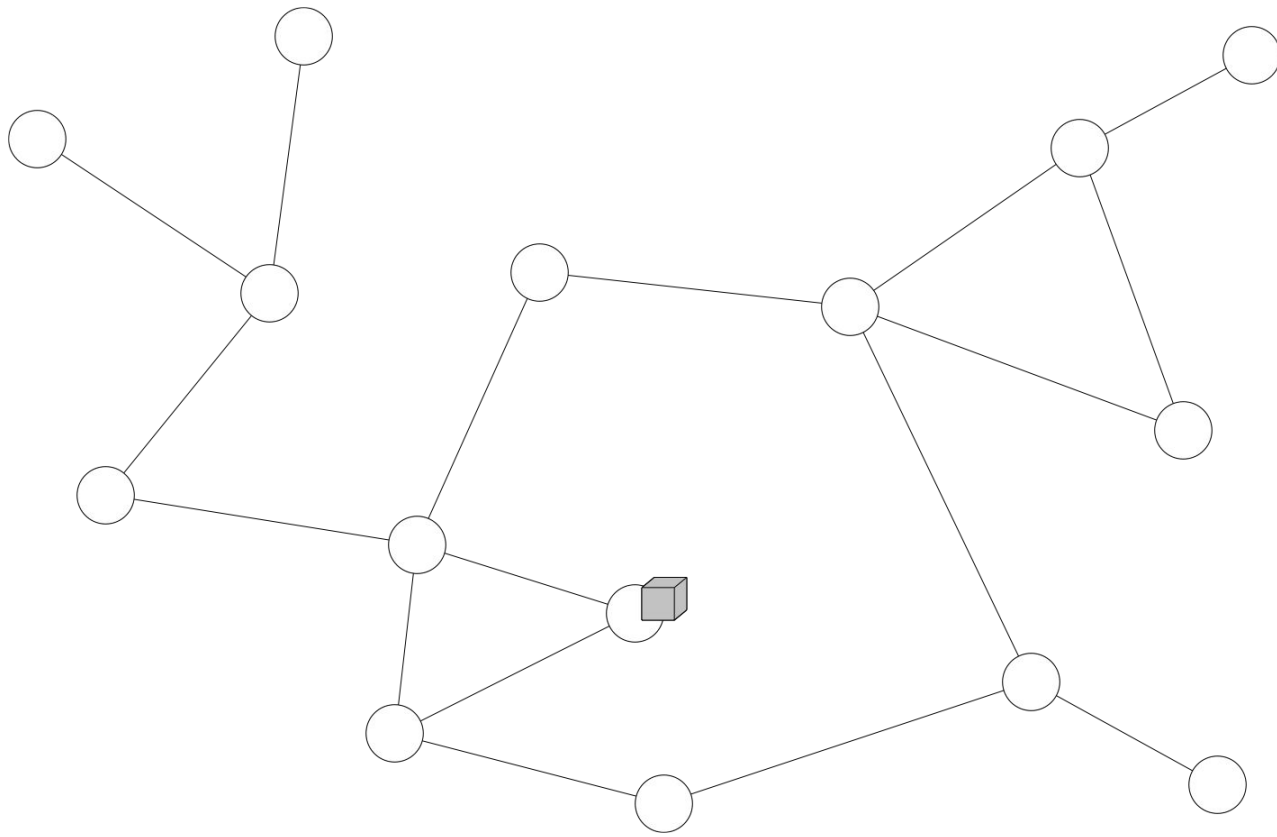
The first to solve the problem



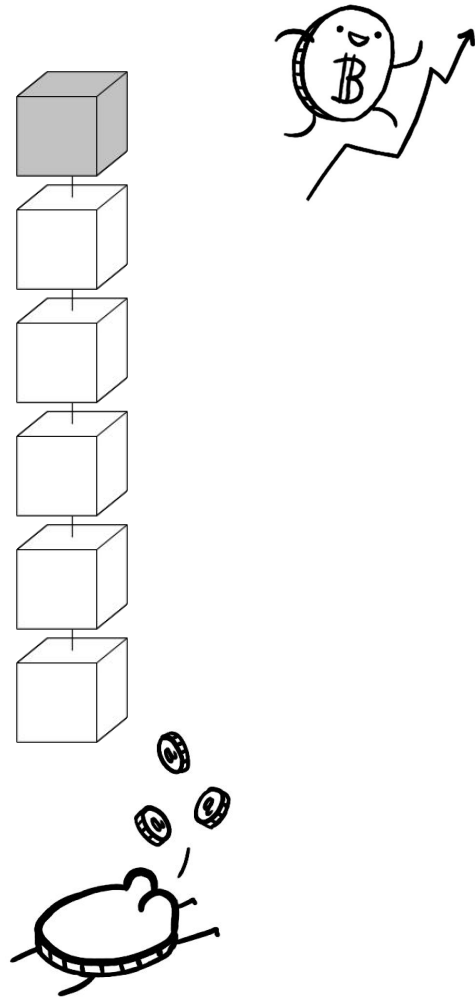


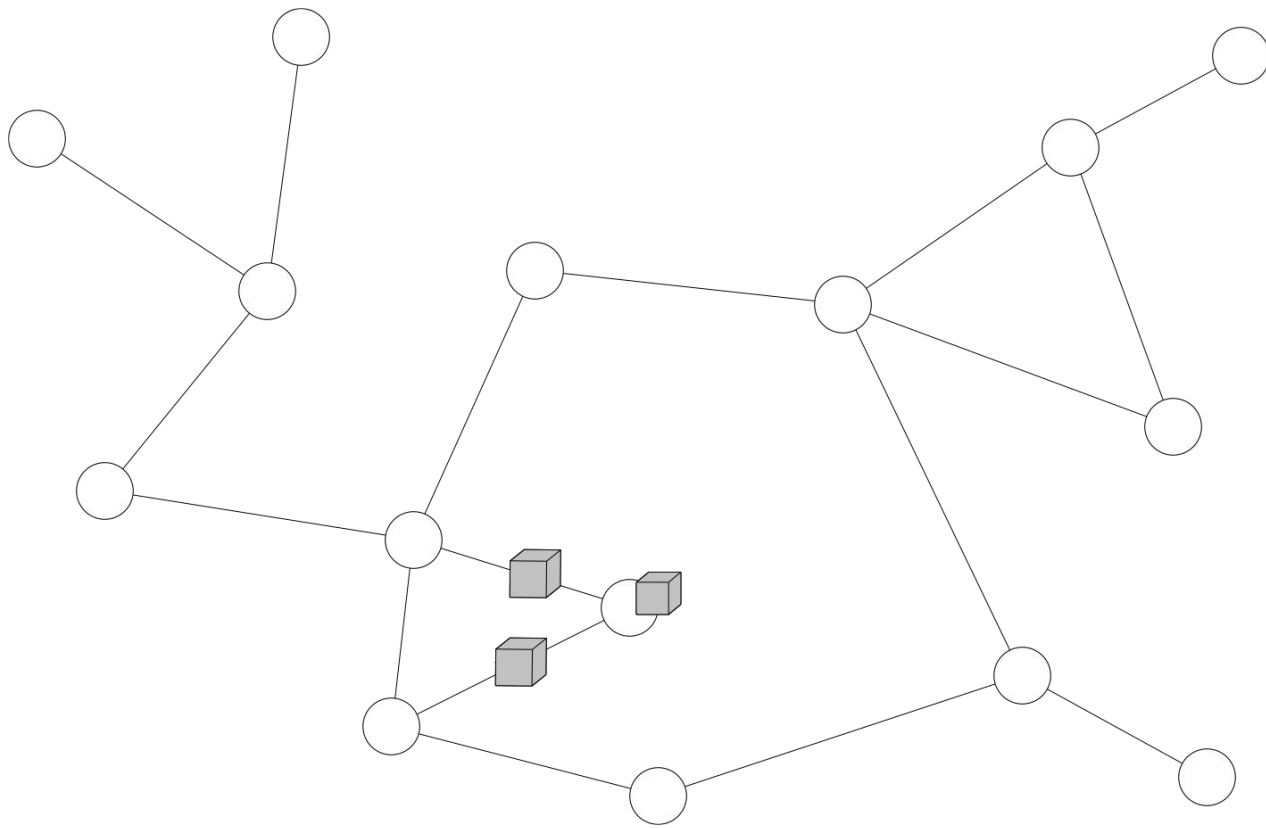
The first to solve the problem extends the blockchain



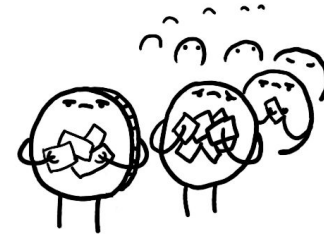
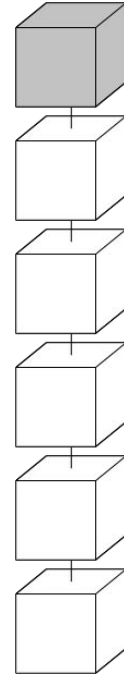


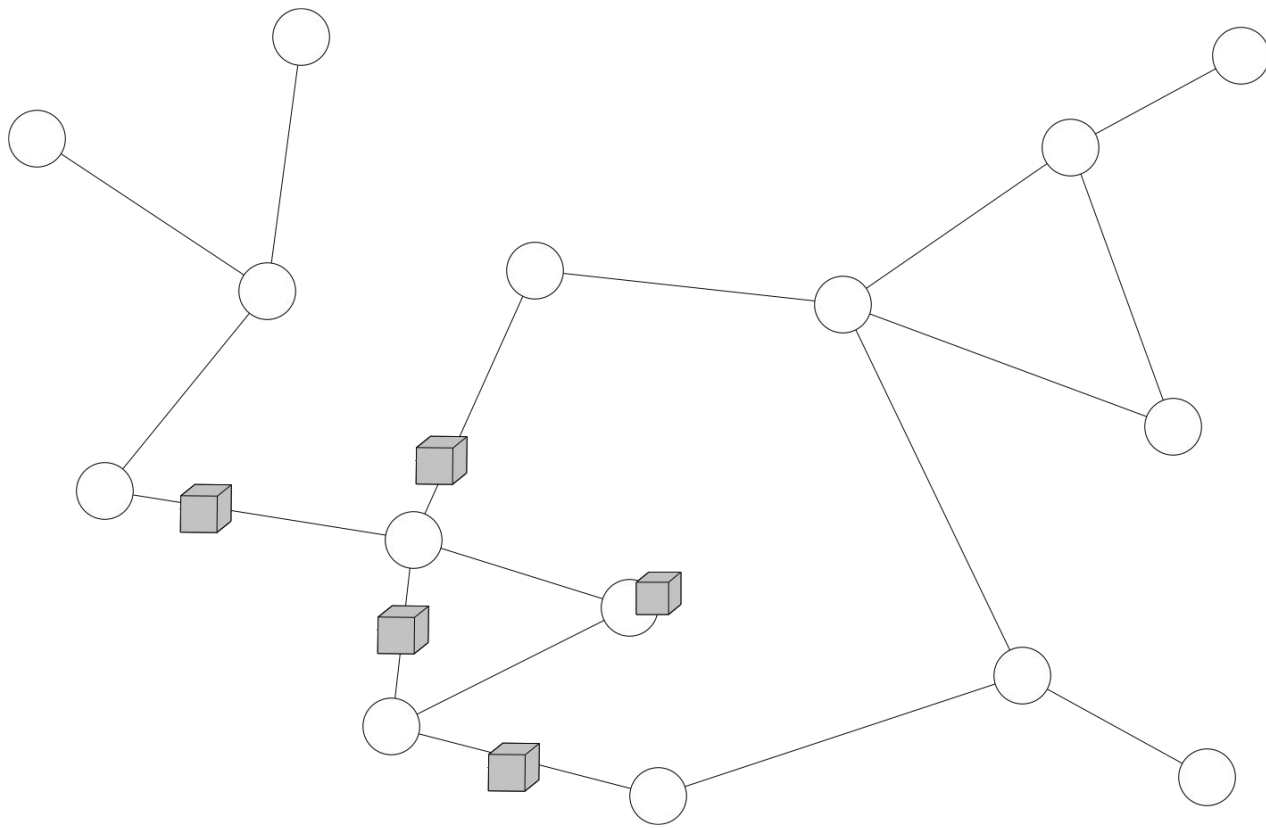
The first to solve the problem extends the blockchain and earns a prize for their hard work



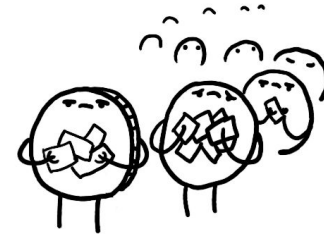
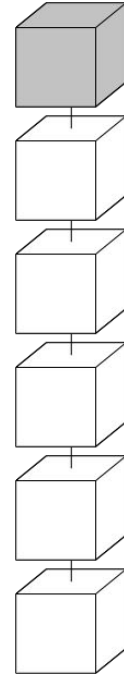


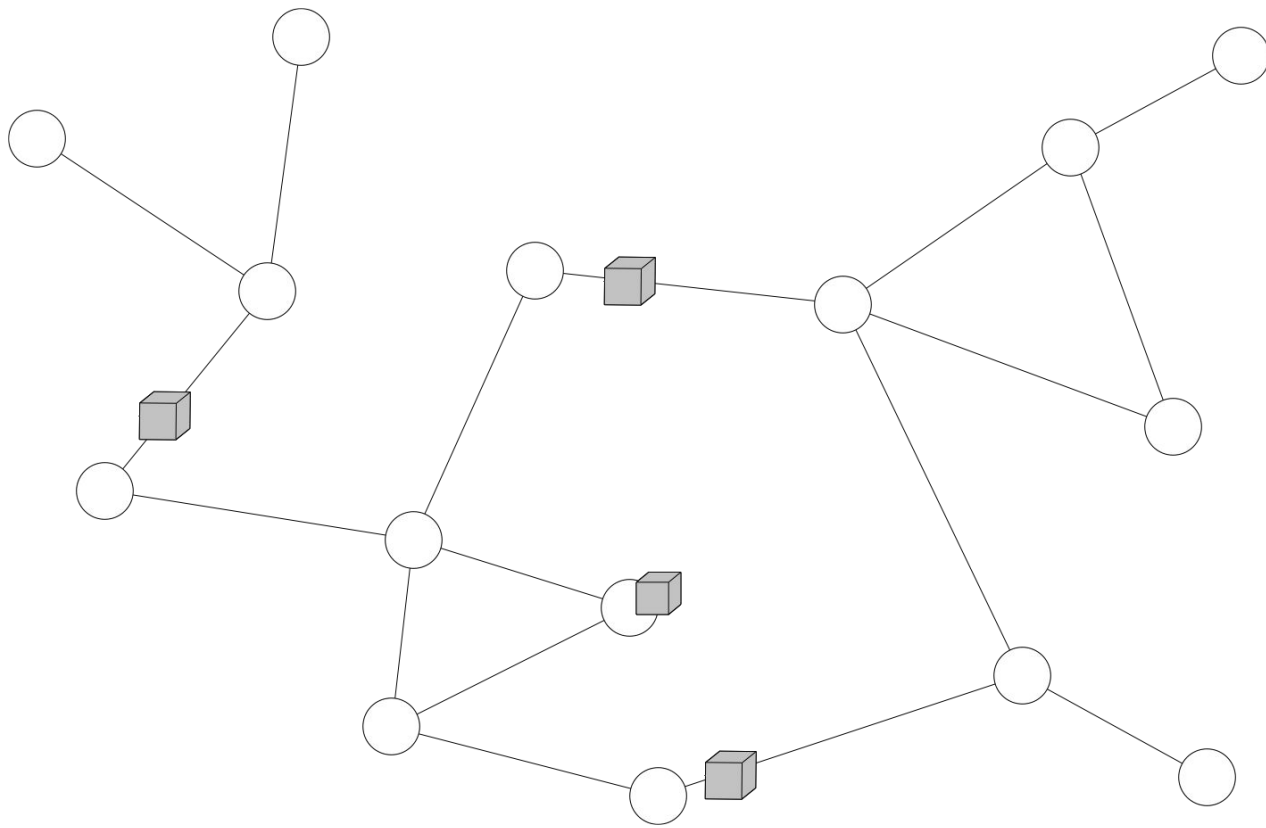
But PoW is slow because data must propagate through



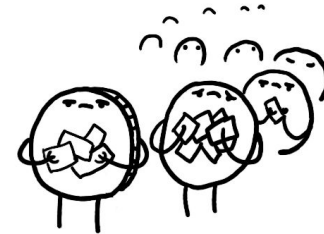


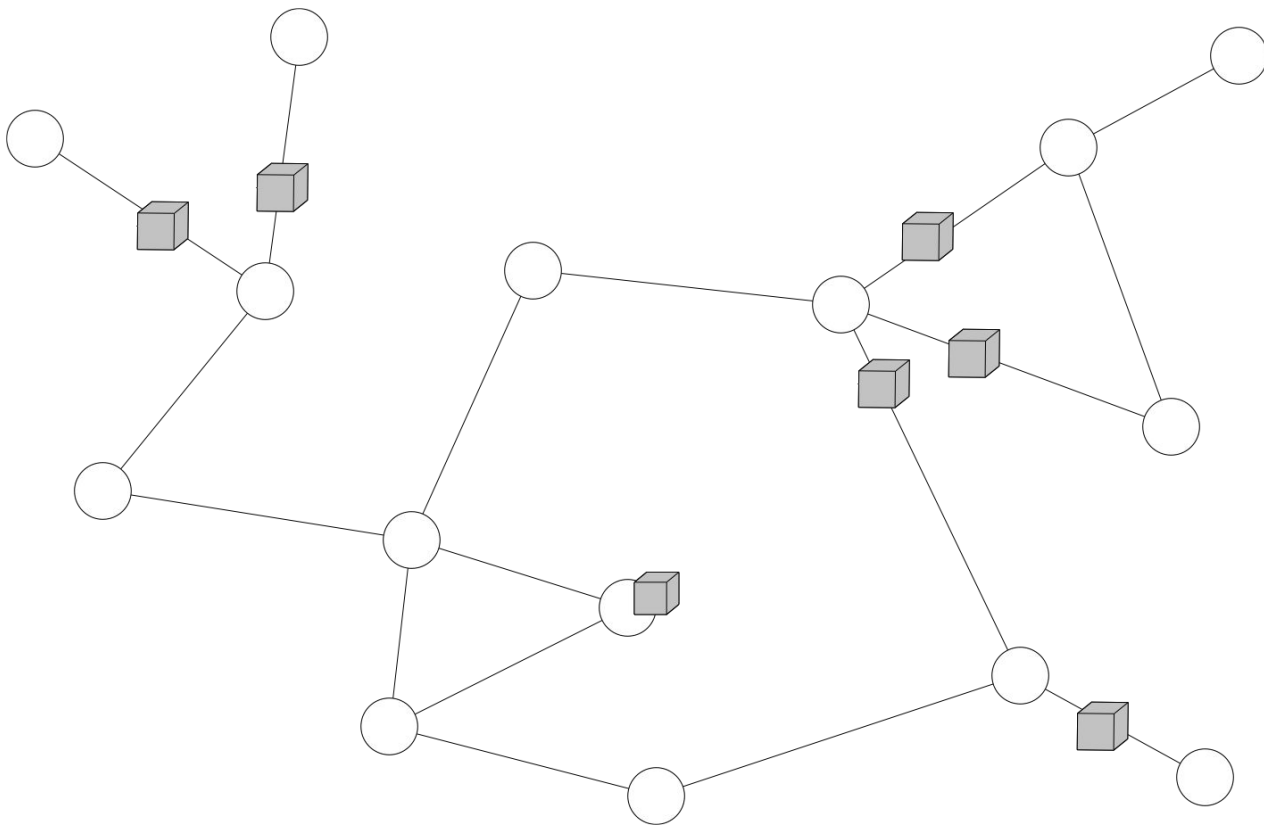
But PoW is slow because data must propagate through the



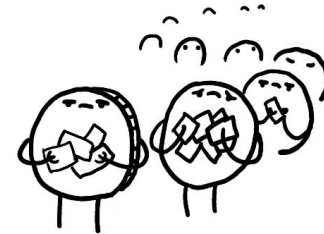
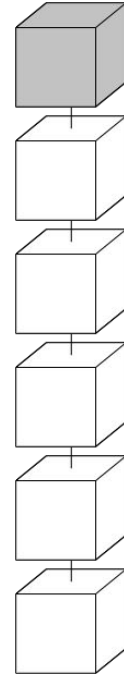


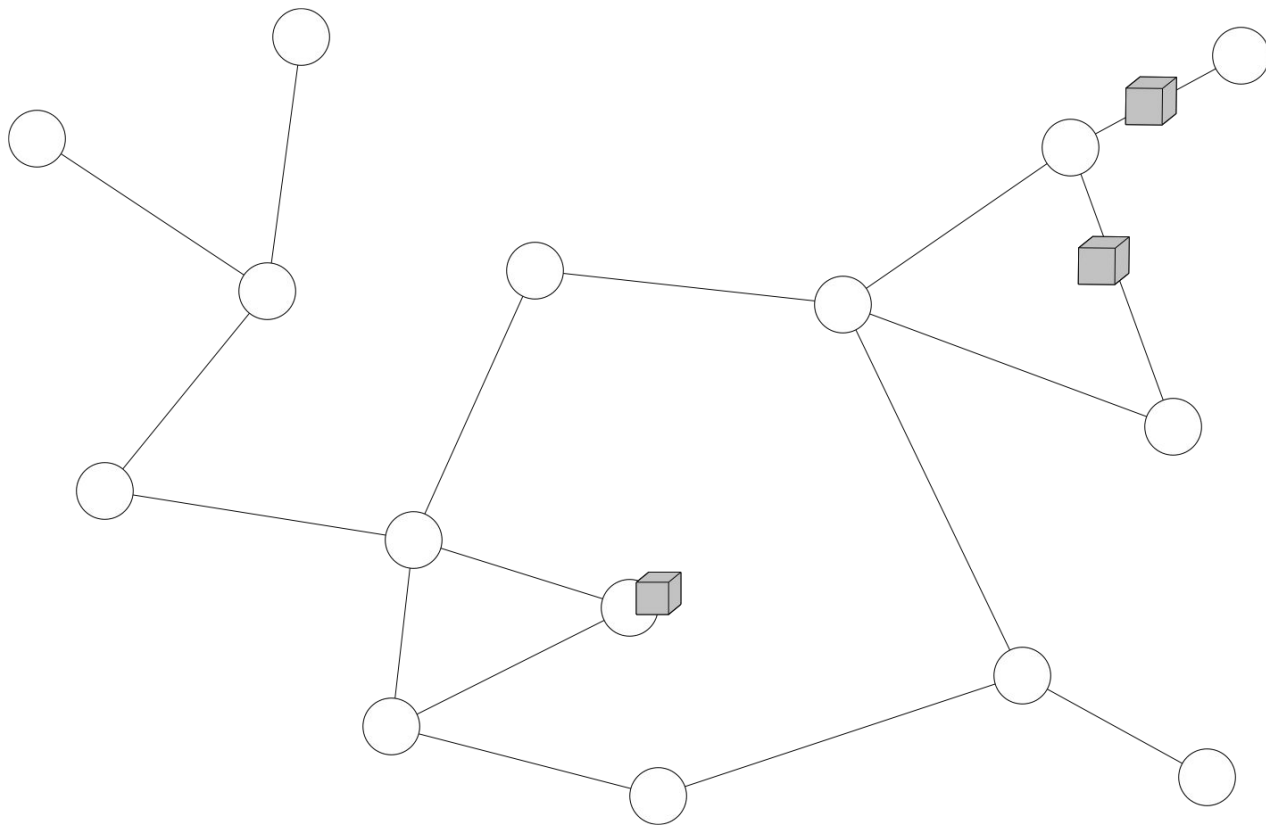
But PoW is 🐌slow🐌 because data must propagate through the entire



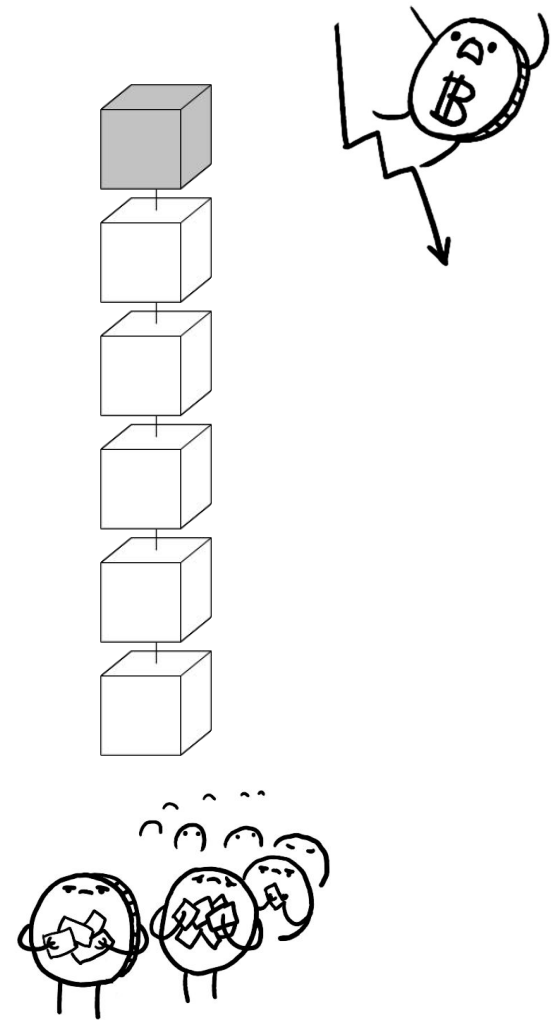


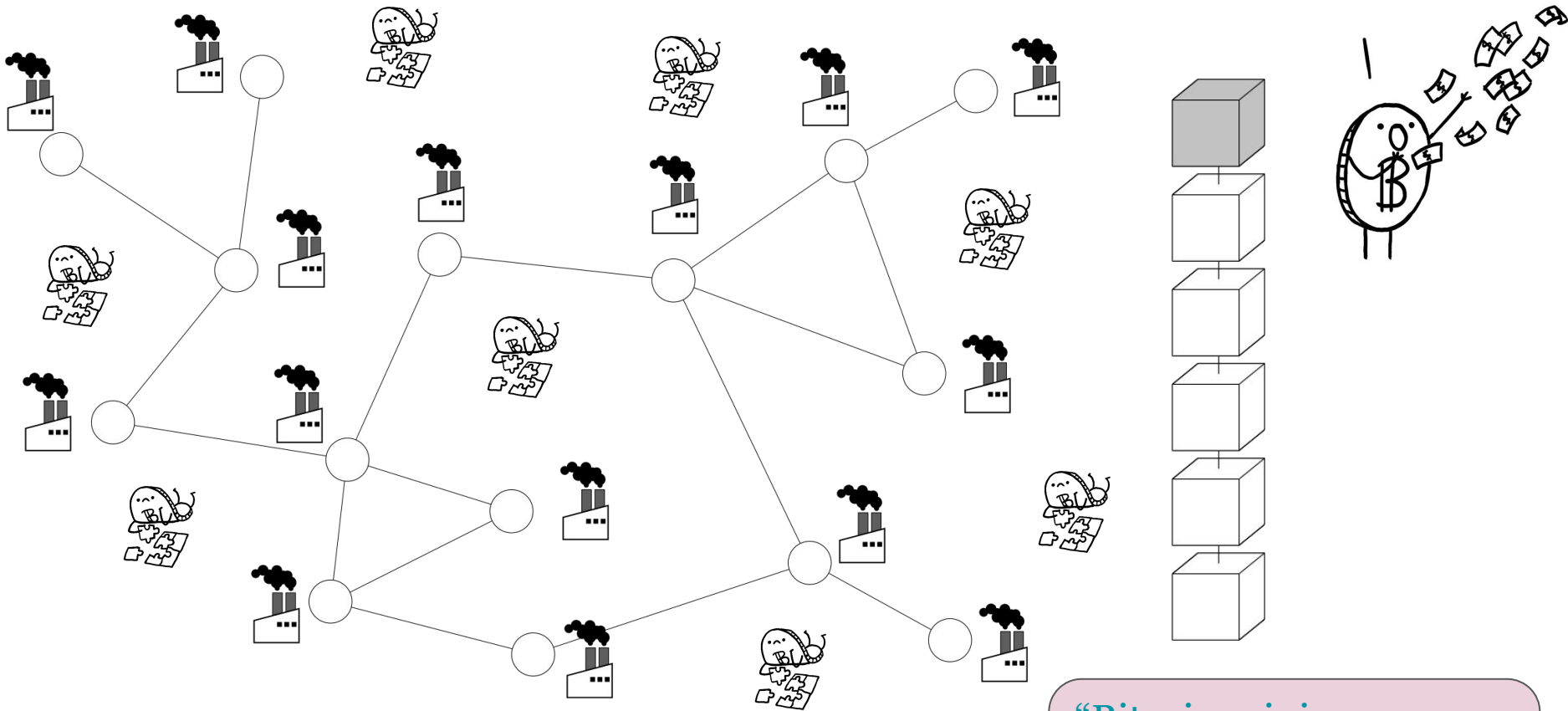
But PoW is 🐌slow🐌 because data must propagate through the entire network





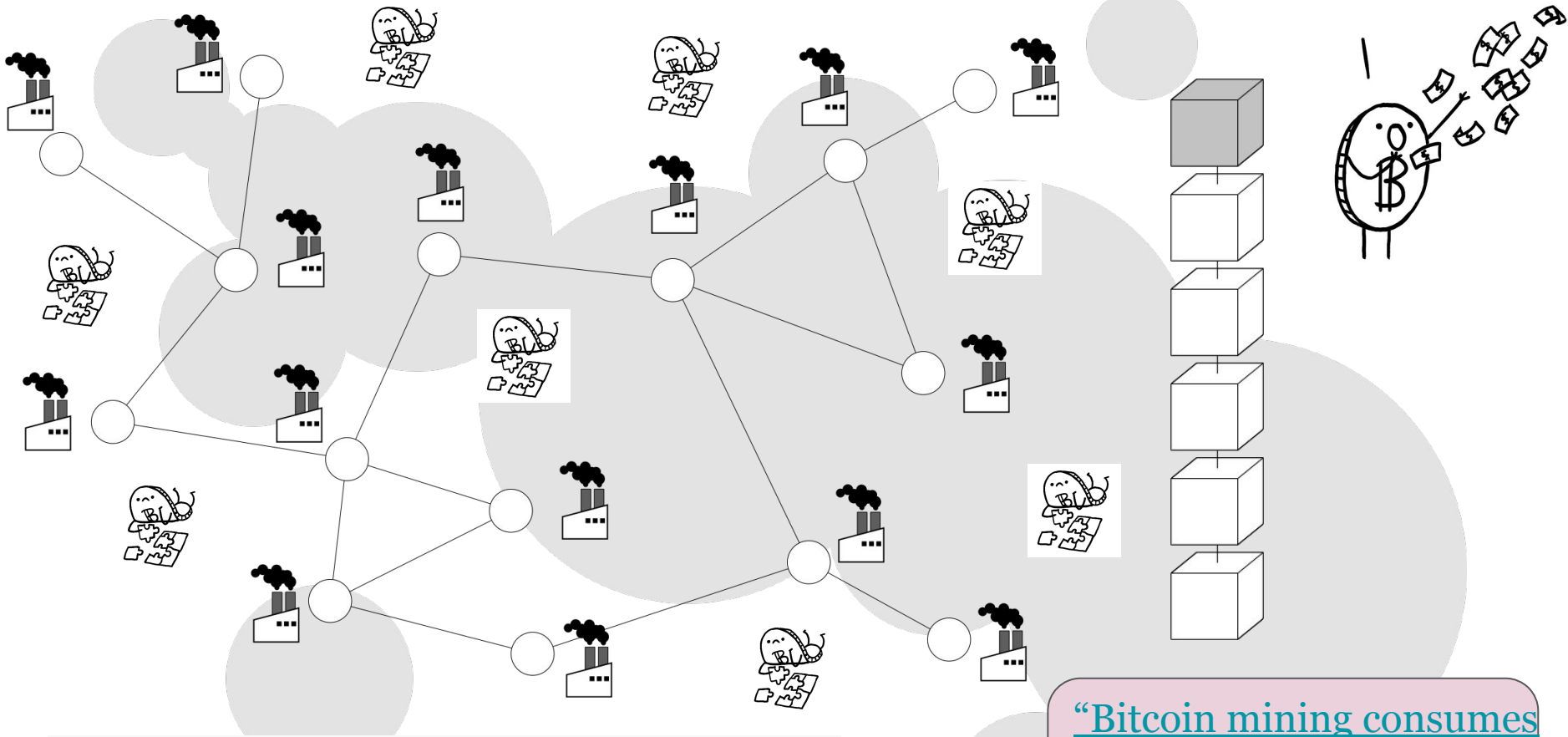
But PoW is 🐌slow🐌 because data must propagate through the entire network...





and having so many nodes compete to solve cryptographic puzzles uses LOTS of electricity

“Bitcoin mining consumes more electricity a year than Ireland”

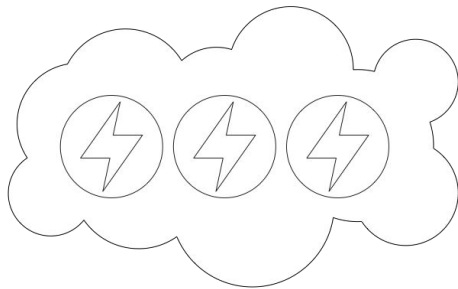


and having so many nodes compete to solve cryptographic puzzles is bad for the environment

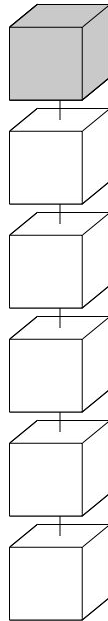
“Bitcoin mining consumes more electricity a year than Ireland”

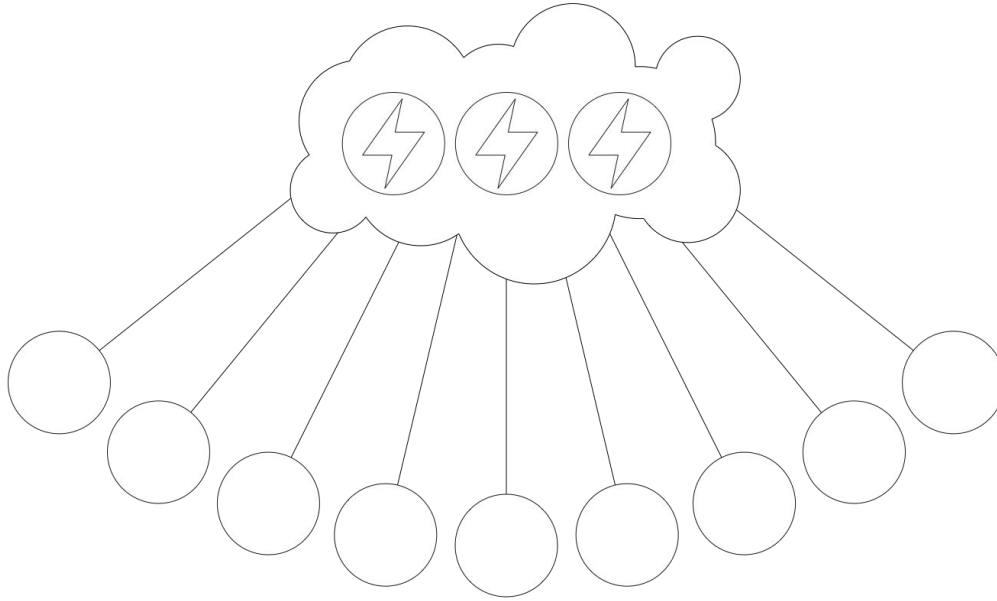


To solve these problems, we introduce
the **PaLa consensus algorithm**

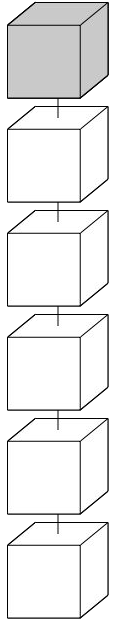


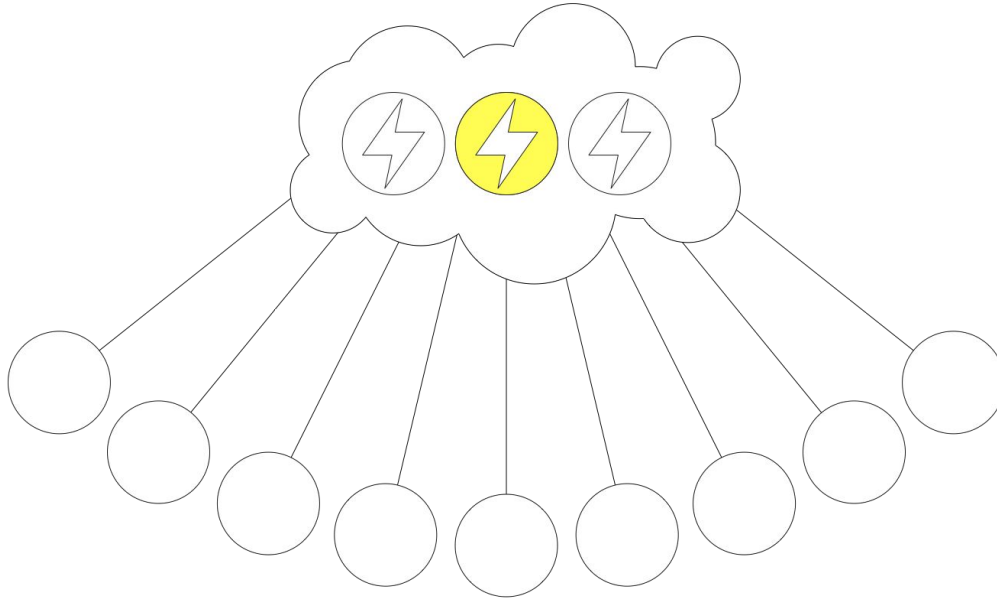
In Pala only selected **proposers** have privileges to extend blocks



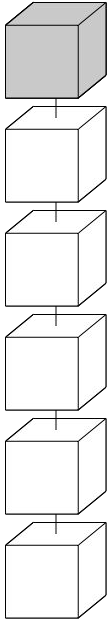


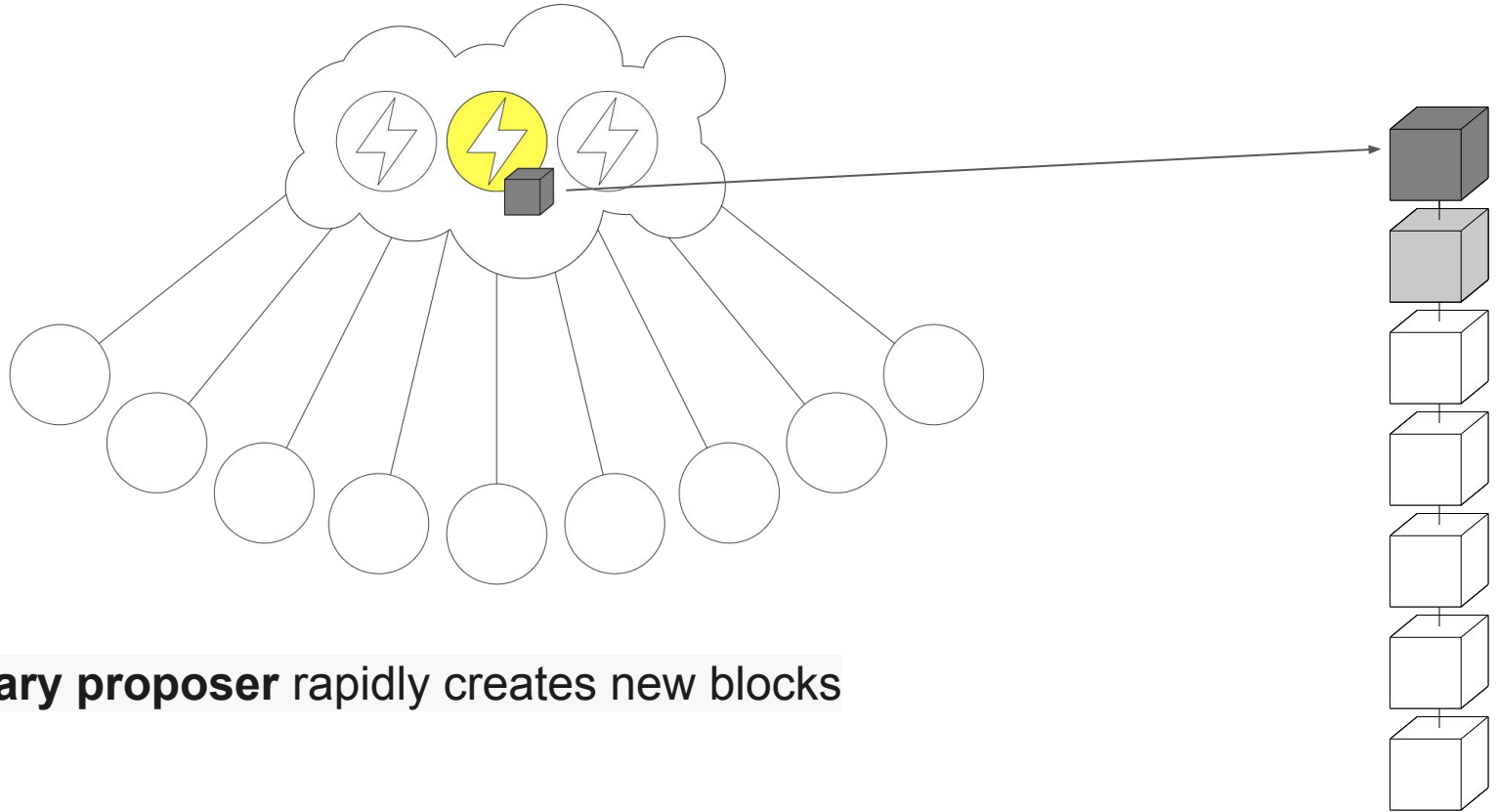
In order to ensure proposers behave honestly, their blocks must be voted on by a **committee of voters**



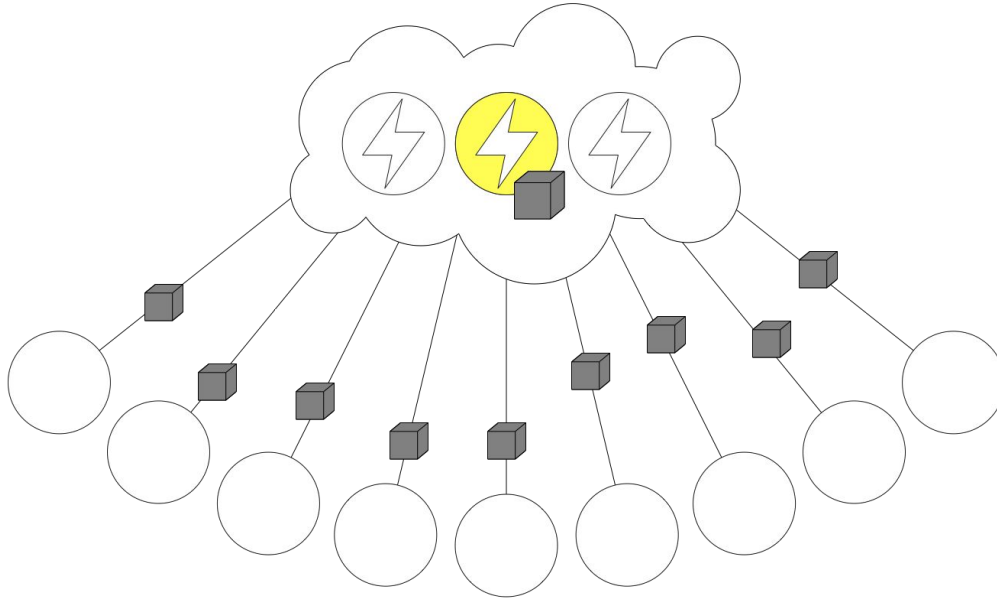


The **primary proposer**

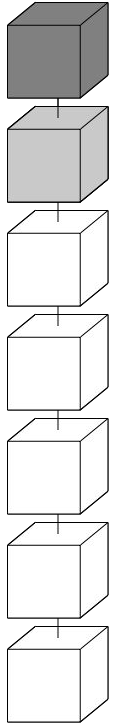


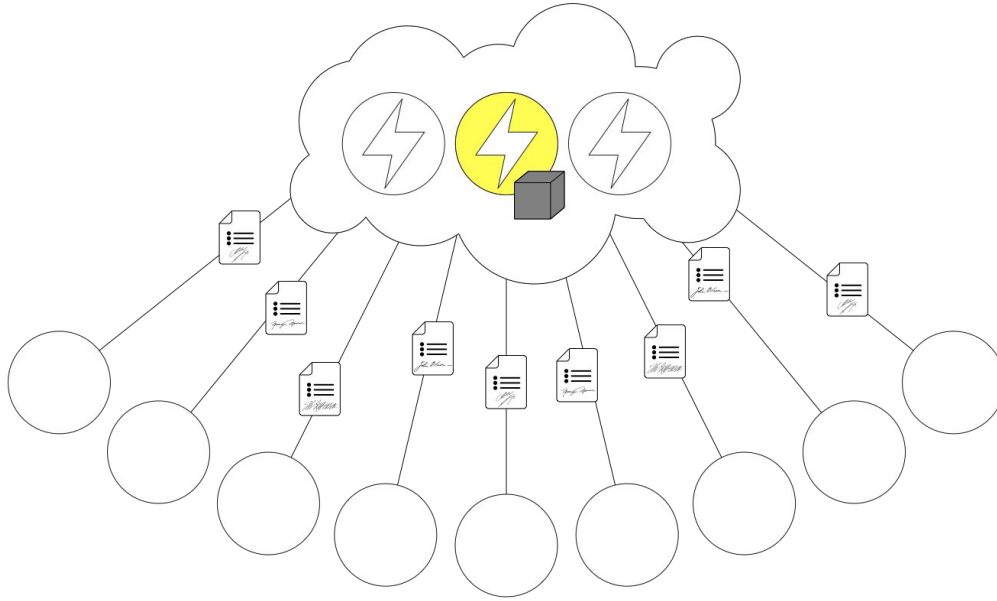


The **primary proposer** rapidly creates new blocks

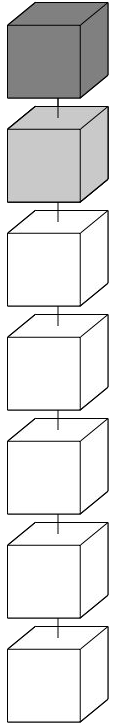


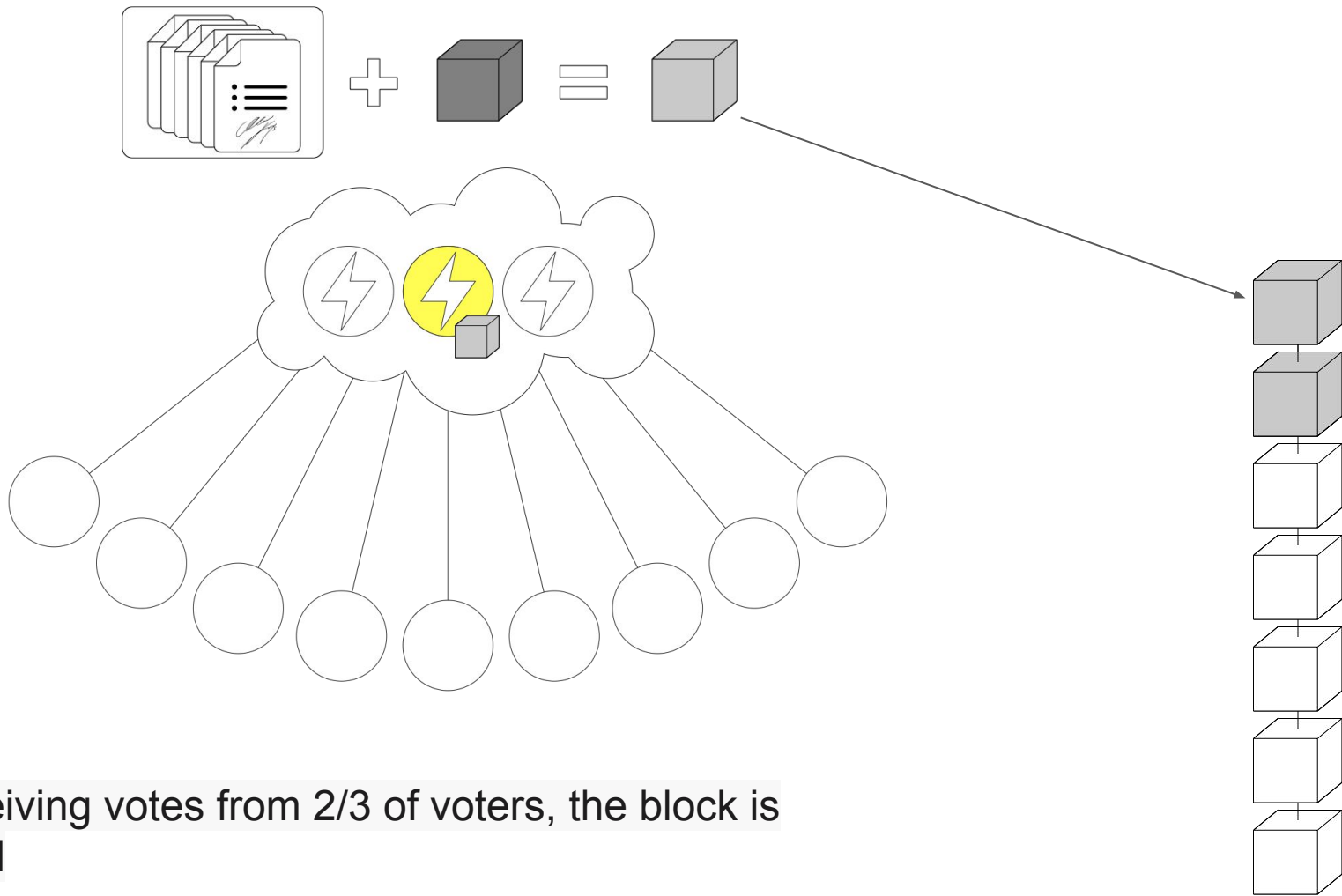
The **primary proposer** rapidly creates new blocks and sends it over a high speed network connection to all voters



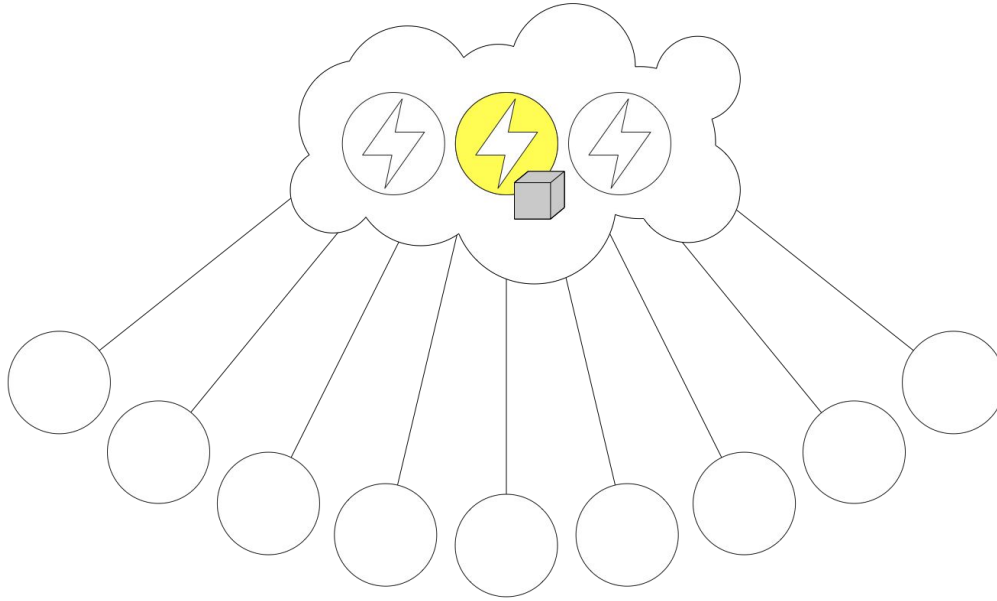


The voters respond with **votes**

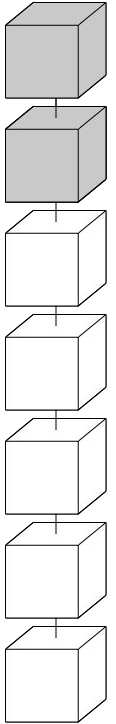


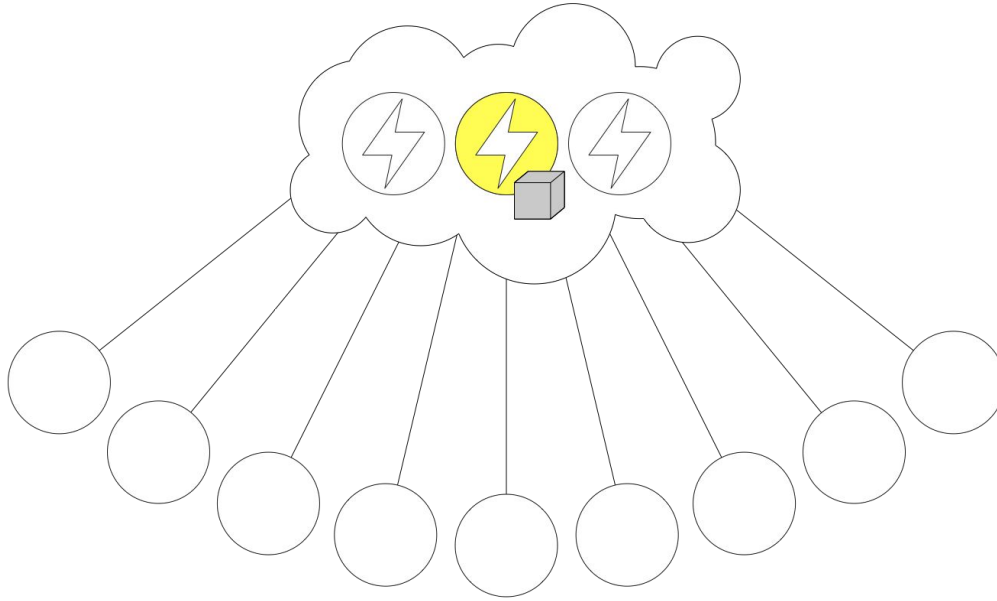


upon receiving votes from 2/3 of voters, the block is
notarized

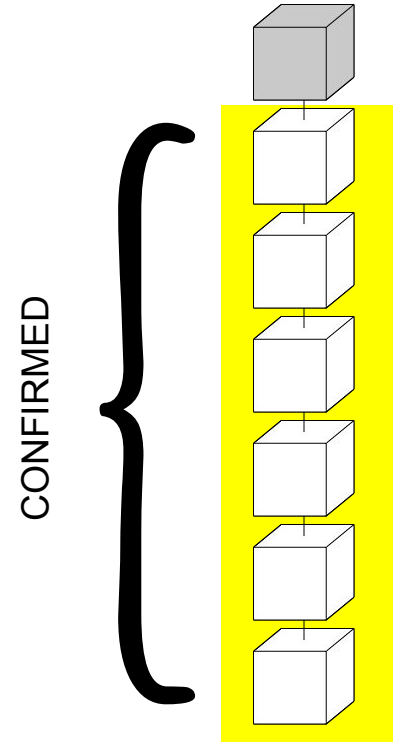


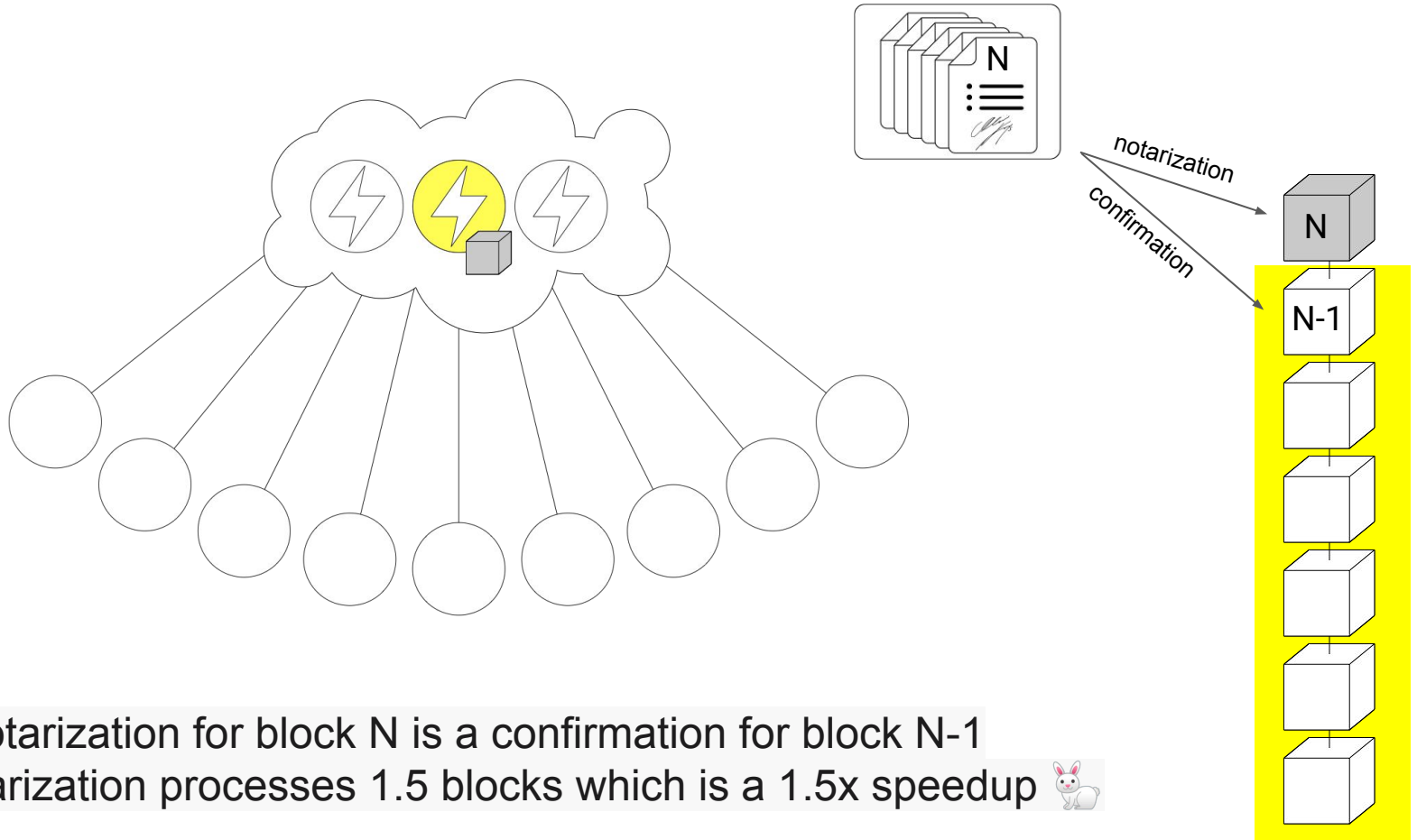
When there are 2 notarized blocks in a row





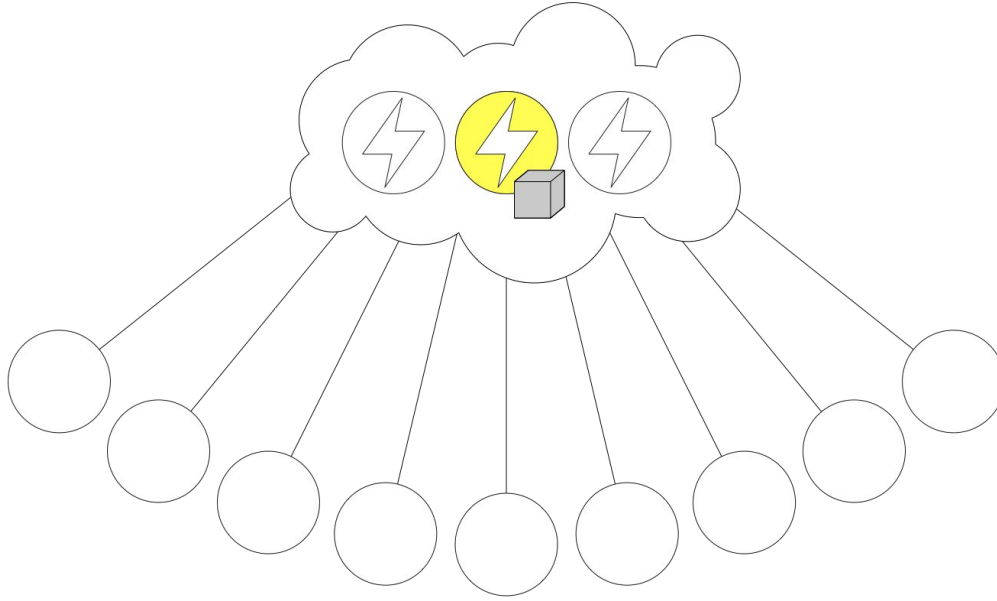
When there are 2 notarized blocks in a row, the second block becomes **confirmed** meaning it will never change and is now part of the **immutable** blockchain history



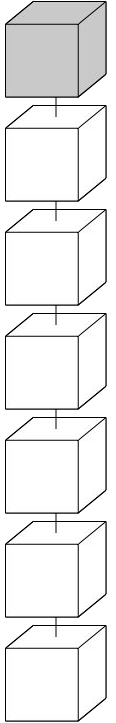


Thus a notarization for block N is a confirmation for block N-1
Each notarization processes 1.5 blocks which is a 1.5x speedup



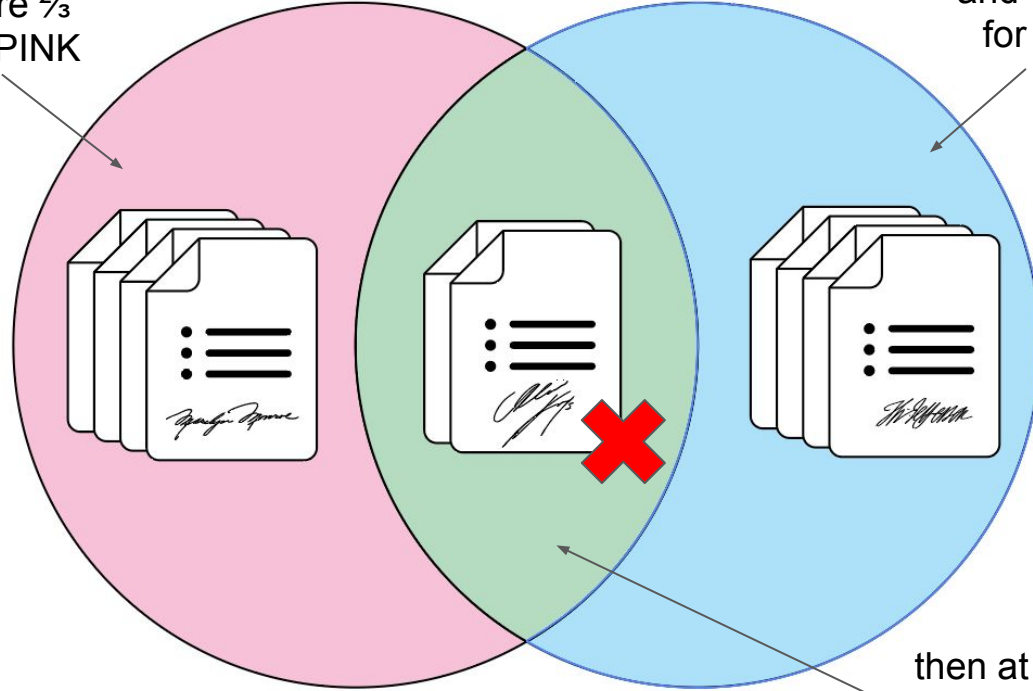


With this process PaLa achieves high throughput and
⚡ ⚡ lighting ⚡ ⚡ fast confirmation times



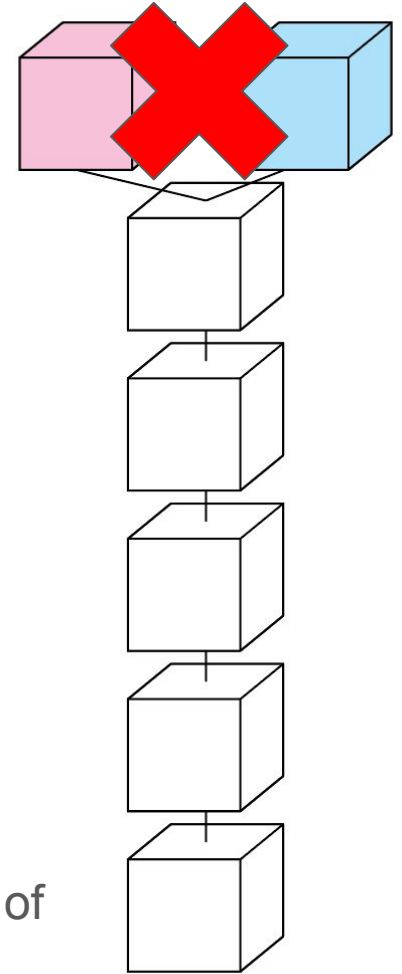
if there are $\frac{2}{3}$ votes for PINK

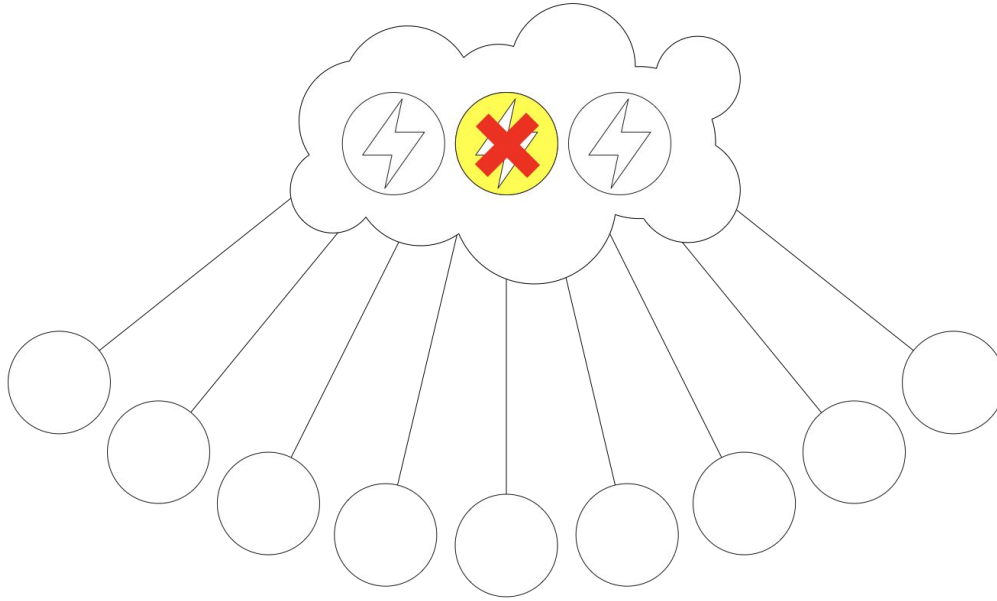
and $\frac{2}{3}$ votes for BLUE



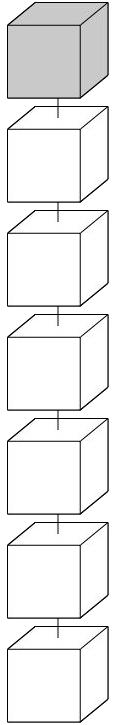
then at least $\frac{1}{3}$ voted for BOTH

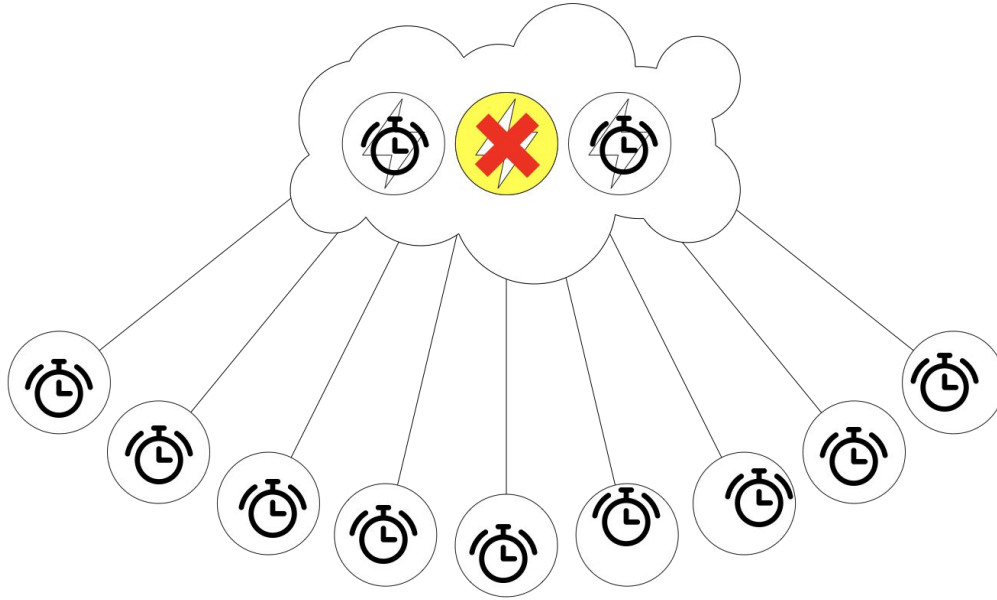
requiring $\frac{2}{3}$ of committee votes for a notarization means it's impossible to finalize two conflicting blocks unless at least $\frac{1}{3}$ of voters voted twice--**honest voters do not double vote!**



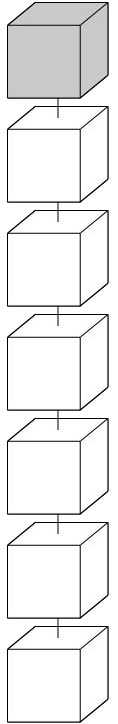


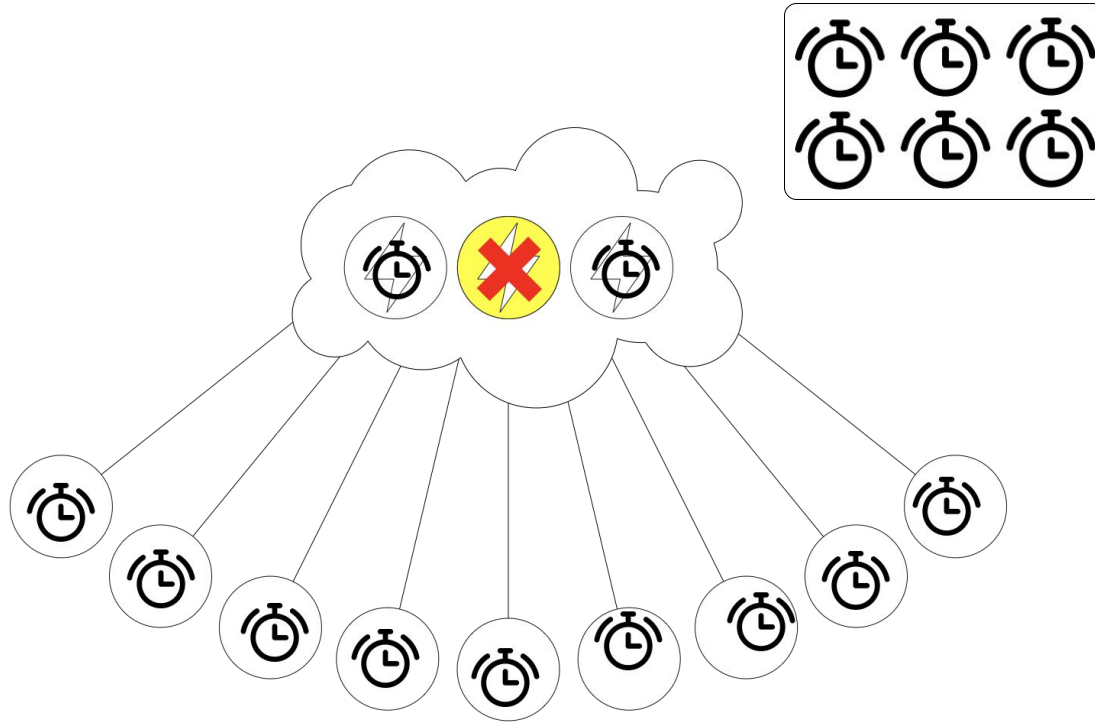
Oh no, the proposer crashed, what now?



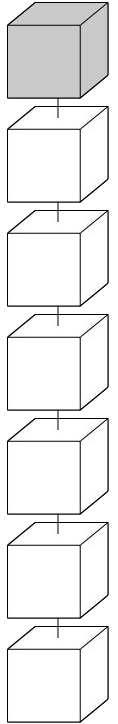


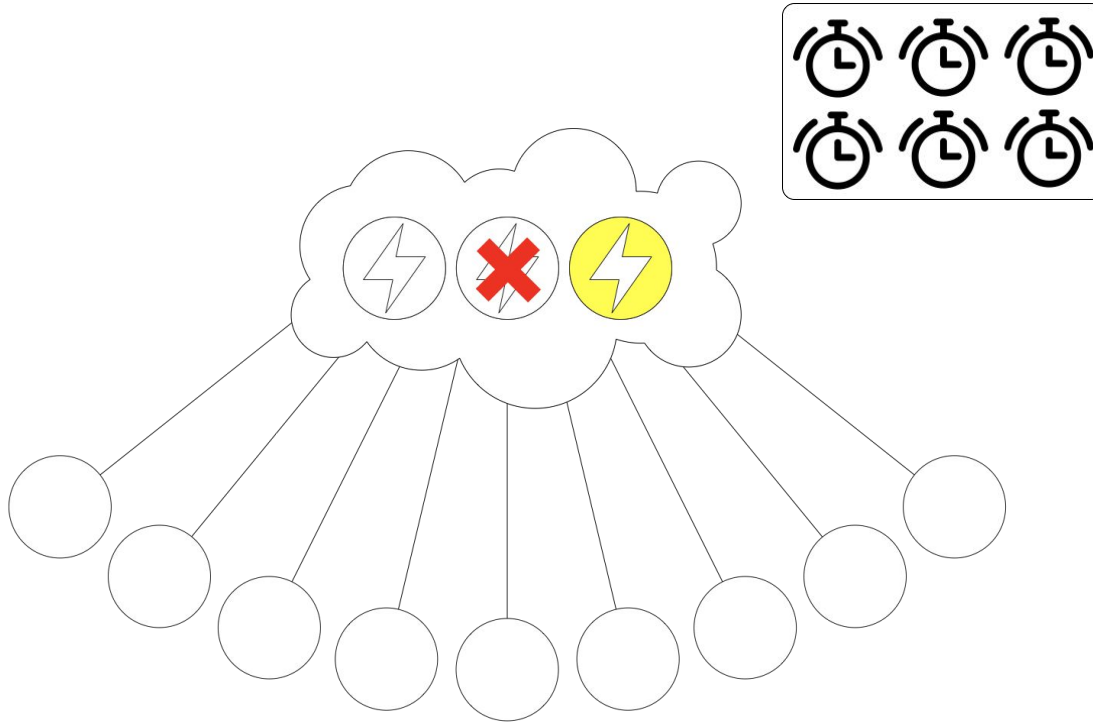
If no blocks are seen for 6 seconds, voters will signal to each other using **clock messages** that it's time to switch proposers



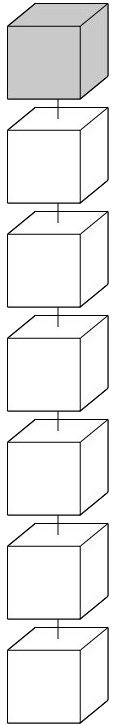


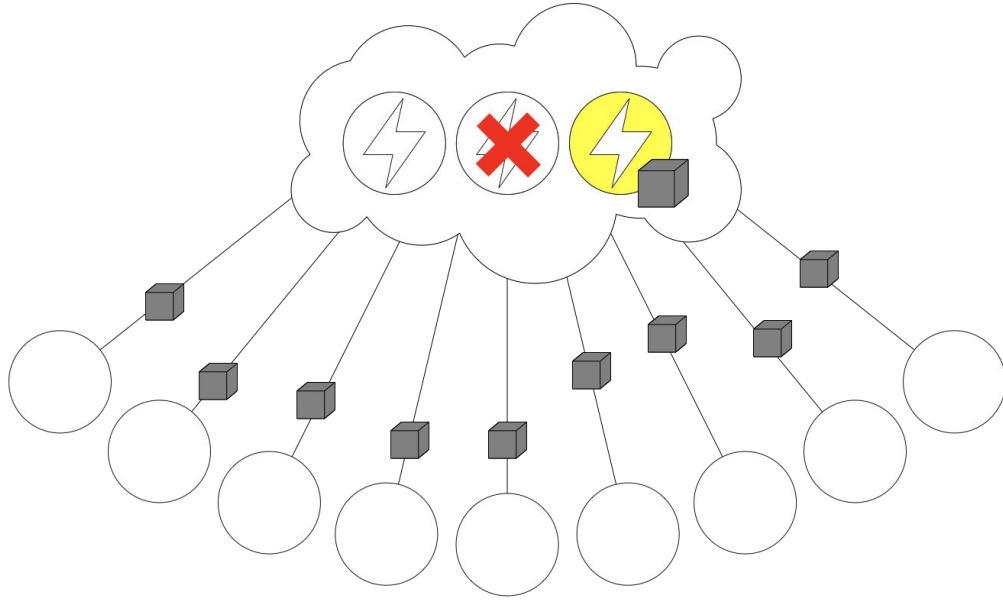
When clock messages from $\frac{2}{3}$ of voters are collected



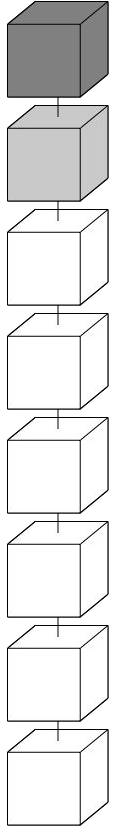


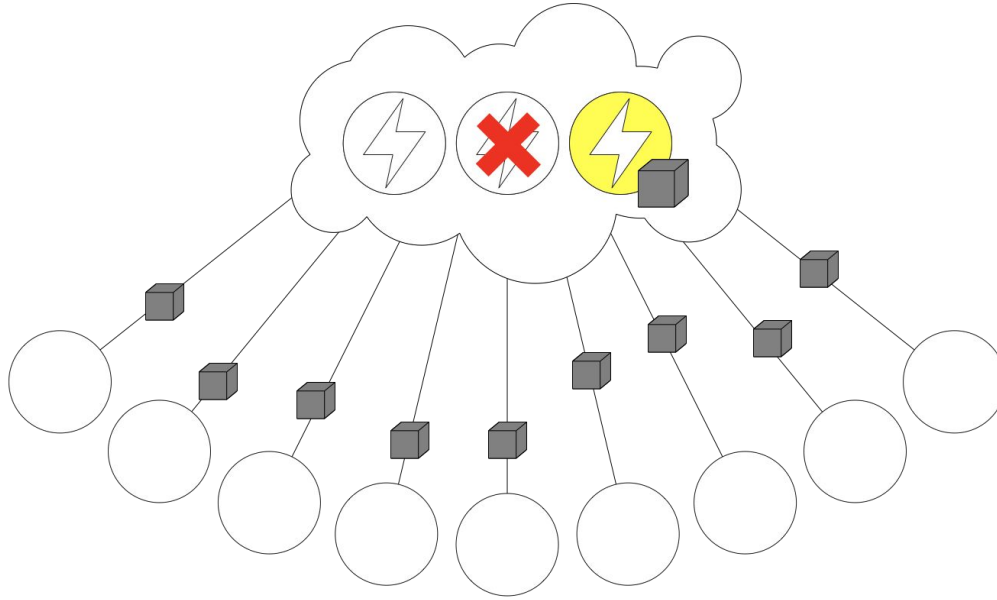
When clock messages from $\frac{2}{3}$ of voters are collected, the next proposer becomes the primary proposer



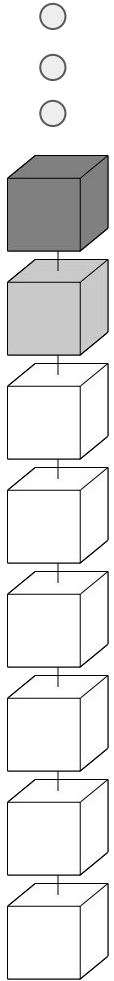


When clock messages from $\frac{2}{3}$ of voters are collected, the next proposer comes online and continues where the previous proposer left off

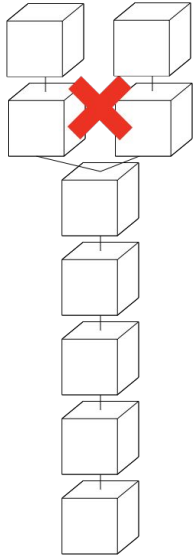




with proposer switch, the chain will always be able to make progress even if the active proposer crashes or behaves maliciously

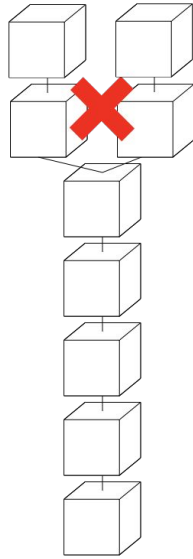


In this manner PaLa achieves

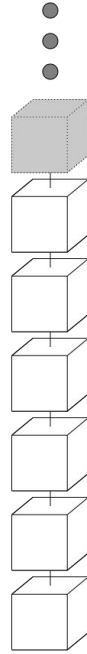


no double spends

In this manner PaLa achieves consistency



no double spends



always progress

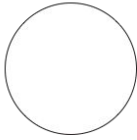
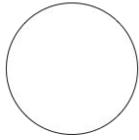
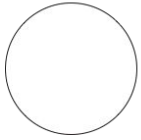
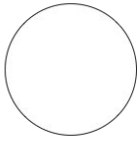
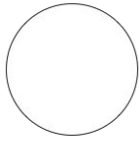
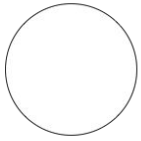
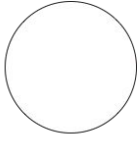
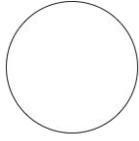
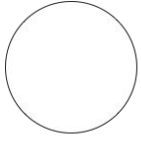
In this manner PaLa achieves consistency and liveness

PaLa Takeaway

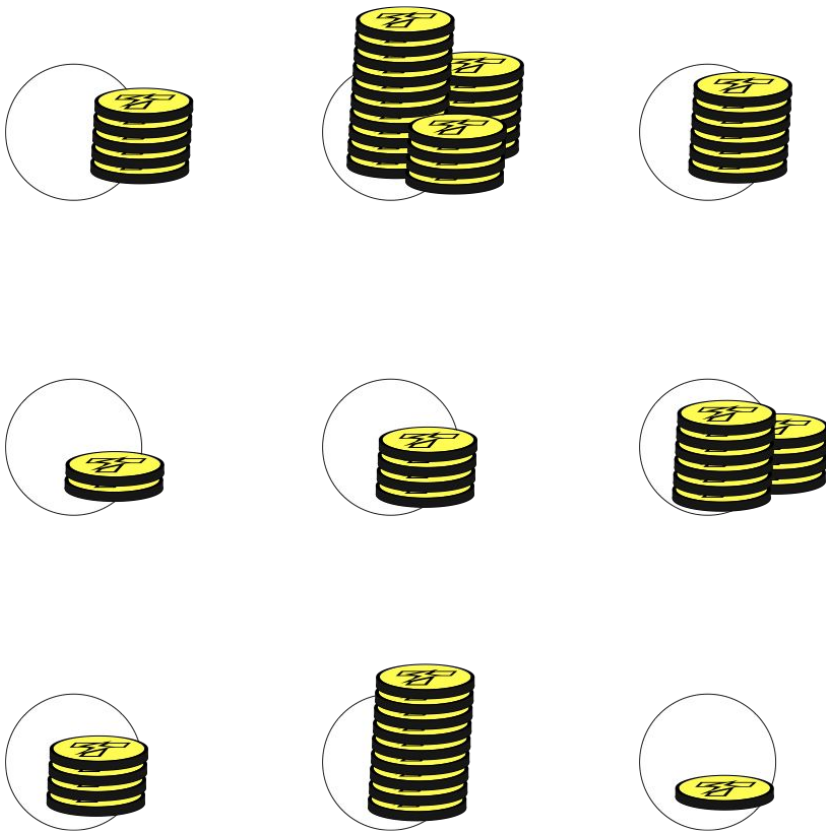
PaLa achieves consensus faster and with fewer messages than any other consensus algorithm. It is the best consensus protocol of its class. Its simple and elegant design is natural to implement and has rigorously proven consistency and liveness properties.

- 100x faster than PoW
- Sub-second confirmation times
- Very high throughput

Proof-of-Stake



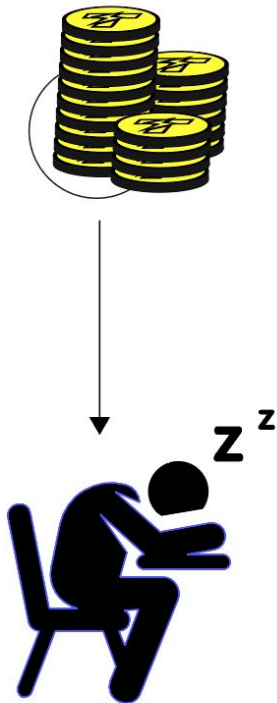
In ThunderCore **Proposers** and **Voters** are selected
using a **Proof-of-Stake** based election



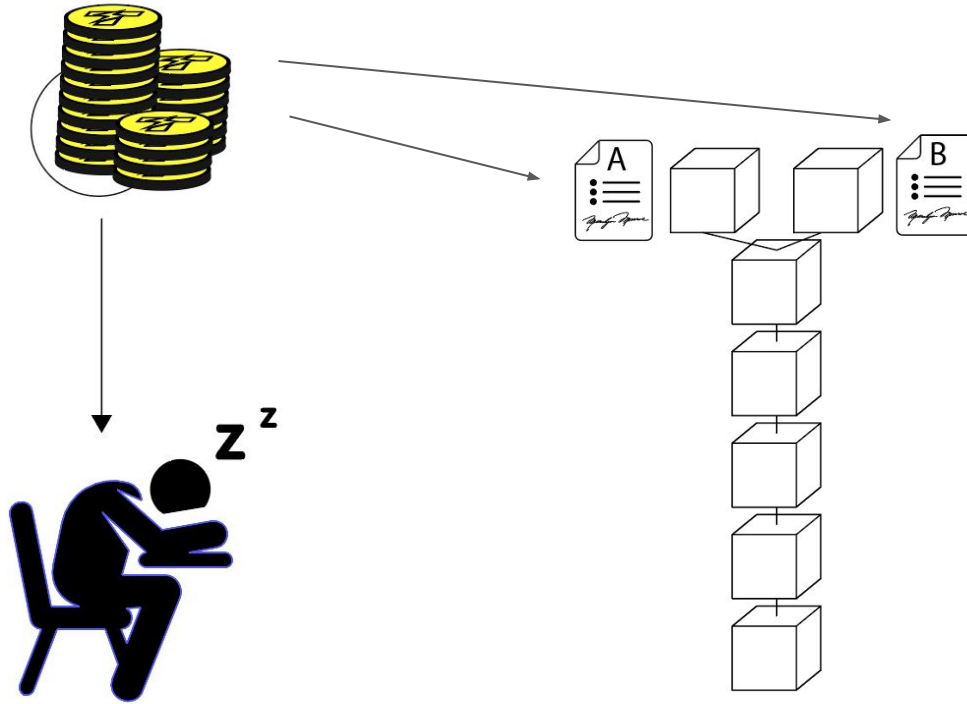
Anyone can become a voter or proposer by staking in
Thunder Token into the ThunderCore election



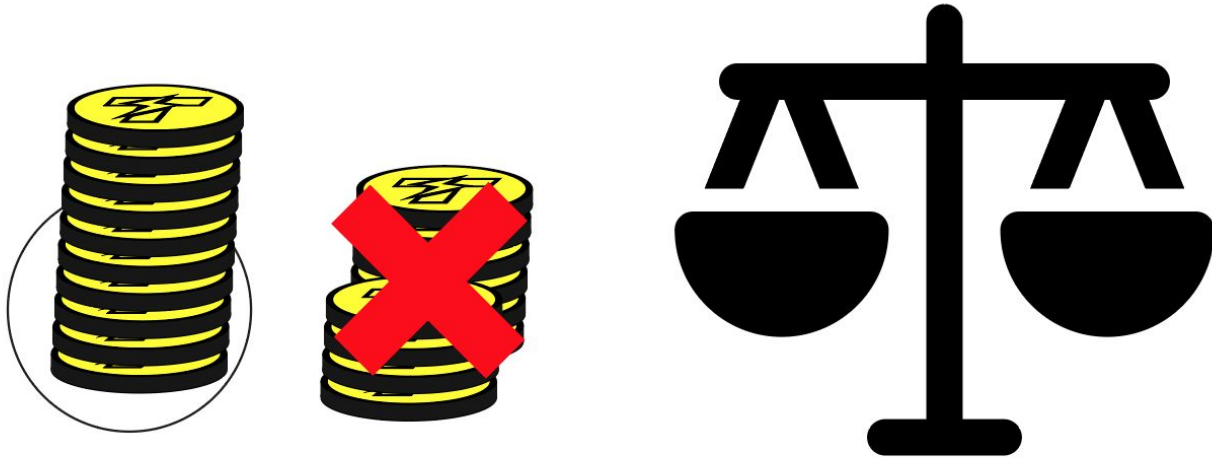
In return they receive a rewards based on their staked-in funds



If a voter node never votes



If a voter node never votes or makes two conflicting votes



Their staked-in funds get slashed

Thus ThunderCore attains security using incentives for participation and high cost of attack through slashing

ThunderCore blockchain built on PaLa
consensus algorithm is fast secure and
decentralized



Join us:

Check out our developer portal:

<https://developers.thundercore.com>



thundercore.com



twitter.com/Thunderprotocol



reddit.com/r/thunder_official



t.me/thunder_official



discordapp.com/invite/5EbxXfw



medium.com/thunderofficial



ThunderCore Consensus 101

