Consensus in Bitcoin

Öz, B., Hoops, F., Gallersdörfer, U., & Matthes, F. (2023). “Blockchain-based Systems Engineering”. Lecture Slides. TU Munich.

Chair of Software Engineering for Business Information Systems (sebis)
Department of Computer Science
School of Computation, Information and Technology (CIT)
Technical University of Munich (TUM)
wwwmatthes.in.tum.de

Outline

1. Consensus
- Consensus in Distributed Systems
- The Byzantine Generals Problem
- Choosing a Leader
- Sybil-Control Mechanisms
- Block Propagation
- Transactions in Orphan Blocks

2. Proof-of-Work (Mining)
- Search Puzzle
- Difficulty Determination
- Incentives
- Amount of Bitcoin
- Mining Hardware
- Mining Pools

This lecture covers parts of the chapter 02 of „Bitcoin and Cryptocurrency Technologies“ by Arvind Narayanan.

Consensus in Distributed Systems

In simple terms, a blockchain is merely a replicated, distributed system[1] where network participants try to **keep in sync** (i) **while accepting new transactions** (ii).

i. Remember that, in a distributed system, not every node has the same worldview at every moment. Hence, they need to **synchronize to achieve consistency**.
ii. When a transaction enters a distributed system, it should **eventually be known by every participant**.

Definition *Distributed Consensus*[2]
A network consists of *N* nodes. Each node has an input value (e.g., a block) that they propose to other nodes. Some nodes are faulty (not responding) or malicious, trying to propose a wrong input.
Two properties must hold:
- The process has to terminate with all **honest nodes** in agreement on **the same input value**.
- The value must have been **generated** by an **honest node.**

- In blockchain networks, nodes try to agree on the following information the world state contains:
  - Which of the proposed **transactions** are **confirmed**?
  - In which **order** do the **transactions** appear in the ledger?

[1] These type of systems are also known as Replicated State Machines. We will go over this concept in the central exercise session.
[2] This is a very idealistic definition since Bitcoin does not 100% satisfy it. Bitcoin adopts a probabilistic consensus. The current state can be generated for a few minutes by a malicious node, or a fork can briefly split the nodes into two different world states.

Consensus in Distributed Systems – The Classic Problem

The Byzantine Generals Problem

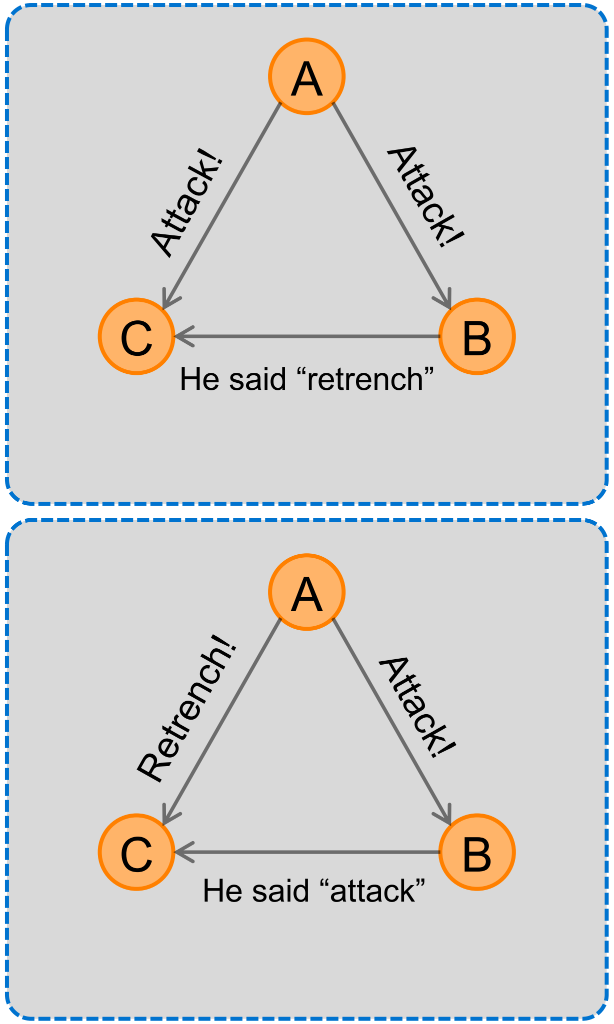
The Byzantine army wants to invade an enemy city. However, it is separated into multiple divisions. They want to attack at the same time, therefore they have to communicate in between the divisions to find a common time to attack.

A general is responsible for one division. These generals communicate by messenger. Some of the generals may be traitors, sending wrong messages to other generals. **The goal is for all loyal generals to derive the same plan without the traitors being able to convince other generals of the wrong plan.**

This property is called "Byzantine-Fault Tolerance" (BFT).

It can be shown that if **more or equal to one third of the generals are malicious**, it is **impossible** for the honest nodes to derive a **common plan** (i.e., consensus cannot be reached).

Three generals



C does not know what to agree on.

*[LAM*1982]* Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, *4*(3), 382-401.

Choosing a Leader – Imagine Bitcoin with a Central Authority

Remember the following questions from the first slide:

- Which of the proposed **transactions** are **confirmed**?
- In which **order** do the **transactions** appear in the ledger?

The answer is simple; **follow the block proposed by the leader**. But **who is the leader**?

What would it look like if we had designed a digital currency with a single central authority (CA)?

- Central authority **signs and creates** every new block and publishes it to the network.
- Other nodes **validate** the content and append the new block to their own copy of the chain.

What are the **disadvantages** of this approach? What could the CA do?

- The CA has to be **nominated** or somehow defined.
- The CA could **unilaterally ignore or delay transactions** of certain parties or with certain properties.
- The CA could render the network **unavailable** by being overloaded or intentionally shut down.

Bitcoin aims to democratize and decentralize the financial world, however, this approach would lead to a centrally controlled protocol operating under **dictatorship**.

Choosing a Leader – Fair and Random

- On the previous slide, we have seen the problems inherent in a digital currency run by a permanent central authority. What we need is an effective and efficient way to **democratize** and **decentralize leader selection**.

➔ **How can a blockchain protocol fairly and randomly choose a leader at every round?**

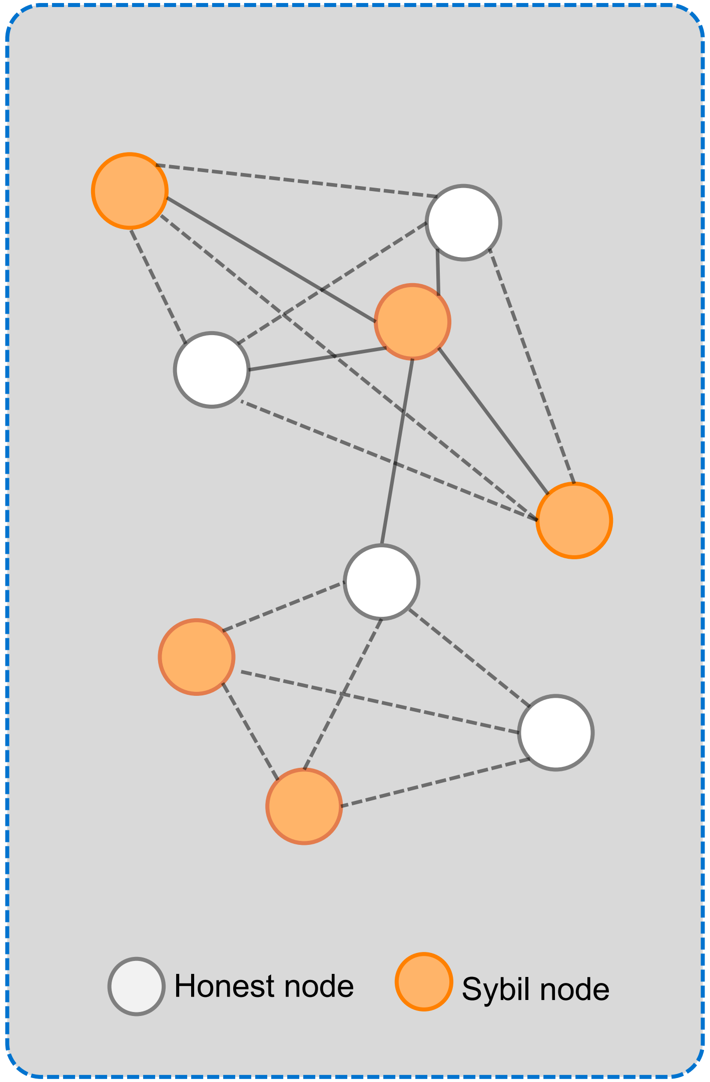
- We must ensure that **no one gets chosen more often simply by creating multiple identities**.
  - Remember, blockchain identities are (usually) free and easy to create.
- Many nodes join the network, also many leave after a short time. How do we know how many there are?
- Approaches like “more than 50% positive votes on a block” would not work, as we do not know how many nodes are in the network.

Sybil-Control Mechanisms

Sybil Attack

Creating multiple identities to gain an advantage over a system is known as a Sybil Attack[1].

- To avoid Sybil attacks, we need to bind the probability of getting chosen as the leader to a **scarce resource**.
- Bitcoin adopts **Proof-of-Work** (PoW)[2] as its Sybil-control mechanism where the scarce resource is the **computational** (hashing) **power**.
- PoW validates the expenditure of the computational work as your chance of getting chosen is **proportional to your computational power**.
- Thus, **creating new identities would not give you an advantage regarding how often you are chosen**.



[1] The name Sybil originates from the book "Sybil. The True Story Of A Woman Possessed By 16 Separate Personalities" by Flora Rheta Schreiber.
[2] Explained in detail in the following slides.

Sybil Control Mechanisms

Computational power	Available coins
↓	↓
Proof-of-Work (PoW)	**Proof-of-Stake (PoS)**

- Facilitates search puzzle
- Requires large amount of tries
- High investment costs
- High energy costs
- Leads to arms race
- High attack costs
- Fully anonymous mining
- More permissionless than PoS

- Coins are deposited to propose new block
- Requires large amount of stake
- Low energy costs
- "Rich people getting richer"
- Low attack costs (discouraged by penalties)
- Non-trivial to bootstrap

Bitcoin uses PoW!

Nakamoto Invented a New Approach

- Bitcoin's approach to distributed consensus was completely new and very different from older approaches that resembled traditional voting and scaled very poorly to more than a handful of nodes.

Ongoing consensus

Sybil-Control Mechanism

Incentives

Probabilistic consensus
The consensus mechanism is an ongoing process in Bitcoin. Therefore, the order of blocks or transactions is never 100% final.

Proof-of-Work (PoW)
The network selects a random node to propose a new block through PoW.[1] As we will see later, this ensures that probabilistic consensus can be reached assuming over 50% are honest.

Incentivized nodes
The network incentivizes nodes to participate block production by rewarding them Bitcoins for creating blocks which are included in the longest chain.

[1] Random in the sense that solving the mathematical PoW puzzle is a probabilistic process.

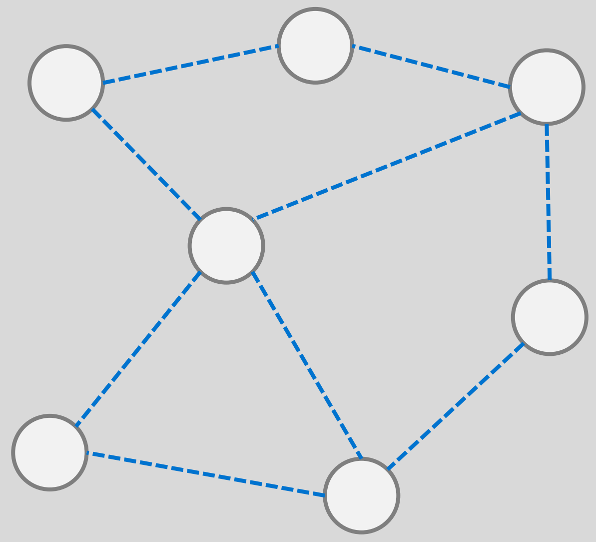
Simplified Consensus Mechanism of Bitcoin

1. *Transaction Broadcast*: Every node that receives or creates transactions broadcasts them to the network, making everyone aware of new transactions.
2. *Block Building*: Miners collect valid transactions, order them, and create a new block containing them.
3. *Leader Node Selection*: A miner is randomly chosen from the network through PoW mechanism as the new leader. The leader proposes his block to the network.
4. *Block Validation*: Other nodes receive the block from the leader node and validate its correctness. A correct block only contains valid transactions.
5. *Block Acceptance*: Other nodes show their acceptance for this block by building new blocks on top of it (i.e., extending it).

Block Propagation

- How do blocks propagate through the network?

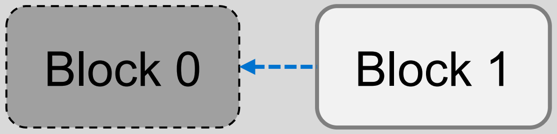
Network



1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.[1]
3. A node builds on the first block it hears of.

Rules

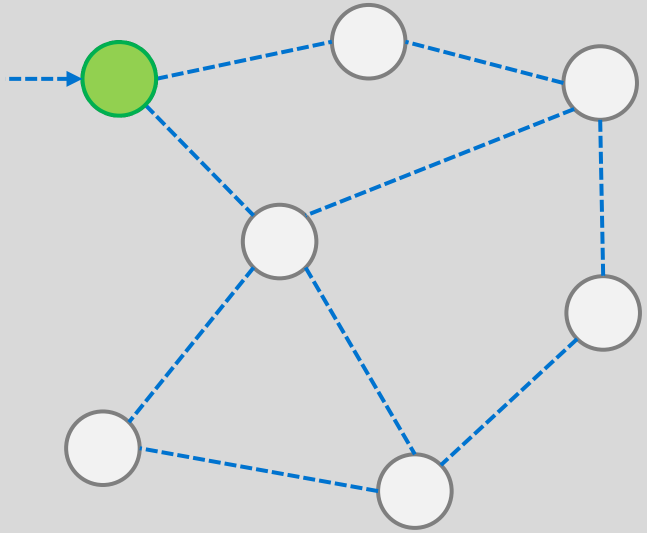
Blockchain



[1] The term “longest chain” refers to the chain with the highest weight in work done (i.e., the chain with the blocks that had the most difficult puzzles where miners submitted the most hashes while trying to solve them).

Block Propagation

Network

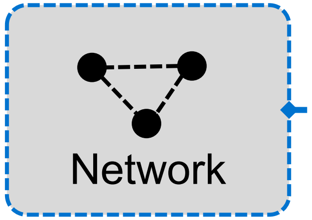


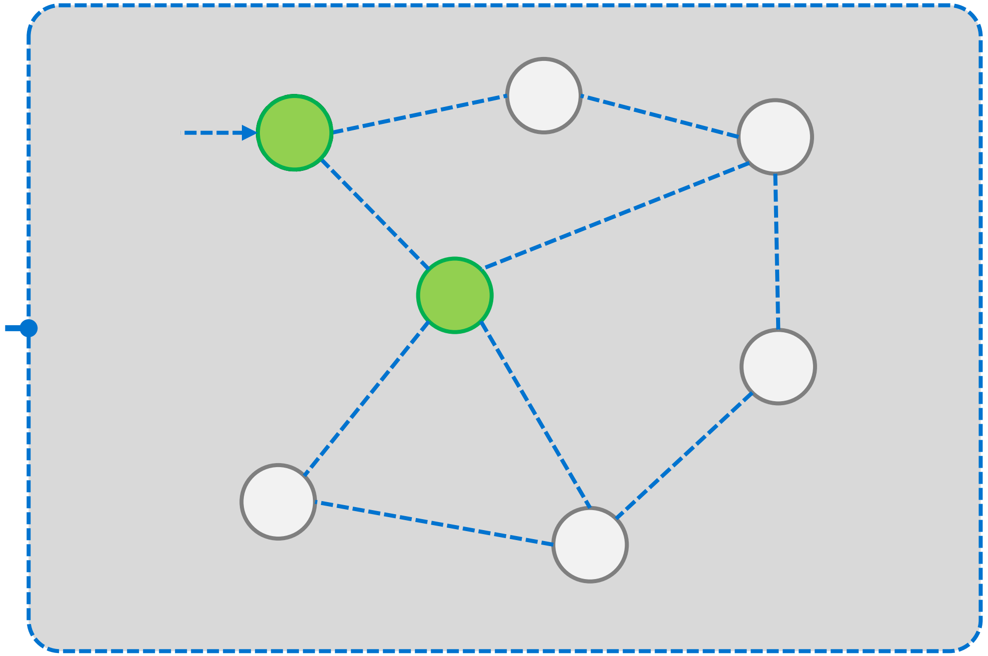
1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.

Rules

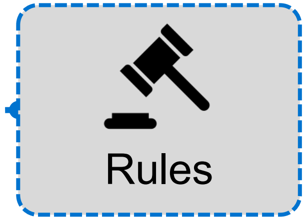
Blockchain

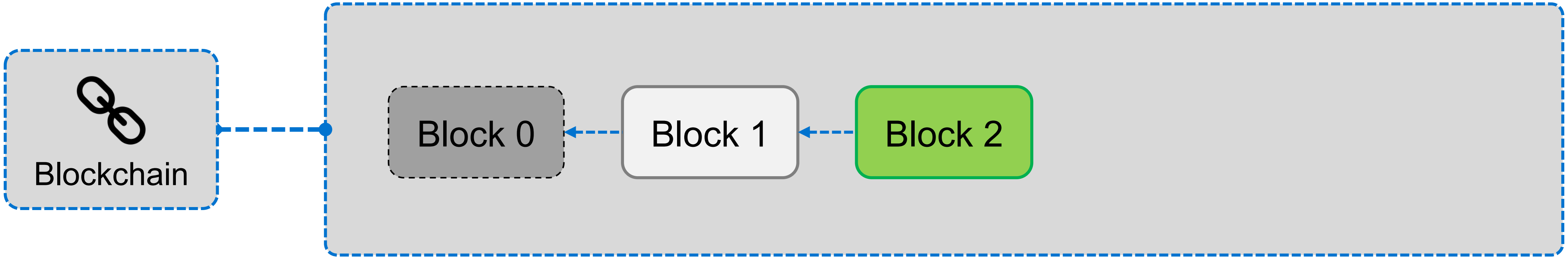
Block Propagation





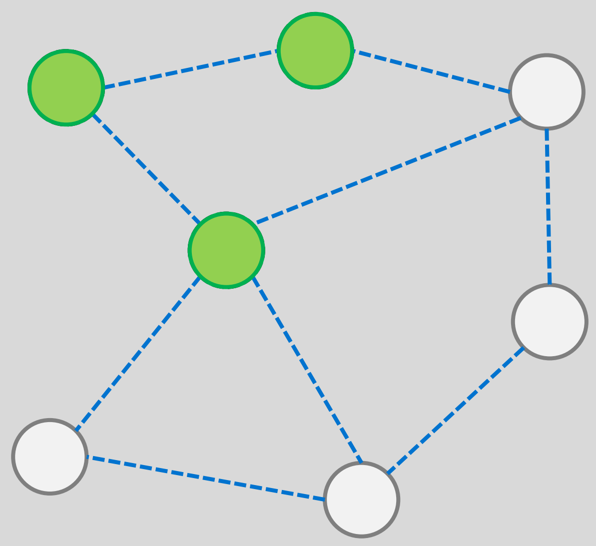
1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.





Block Propagation

Network



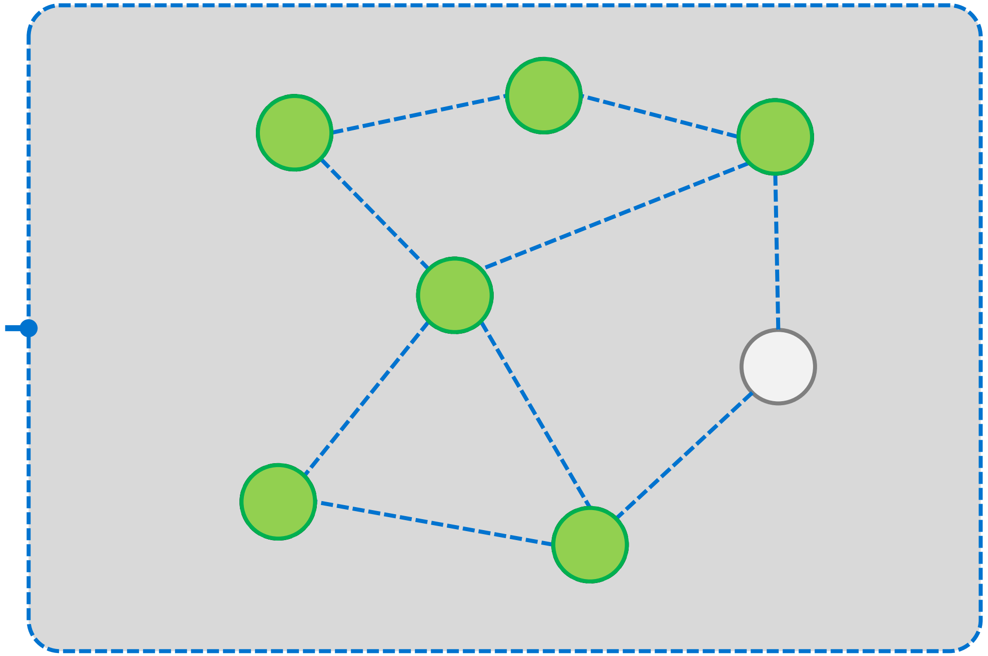
1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.

Rules

Blockchain

Block Propagation

Network

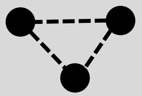


1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.

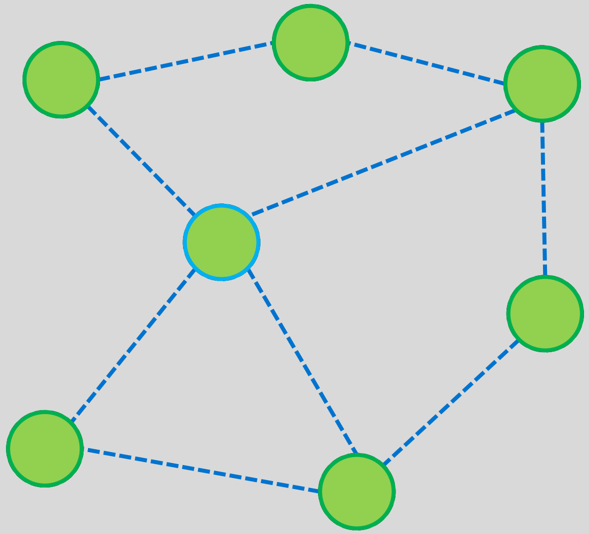
Rules

Blockchain

Block Propagation



Network

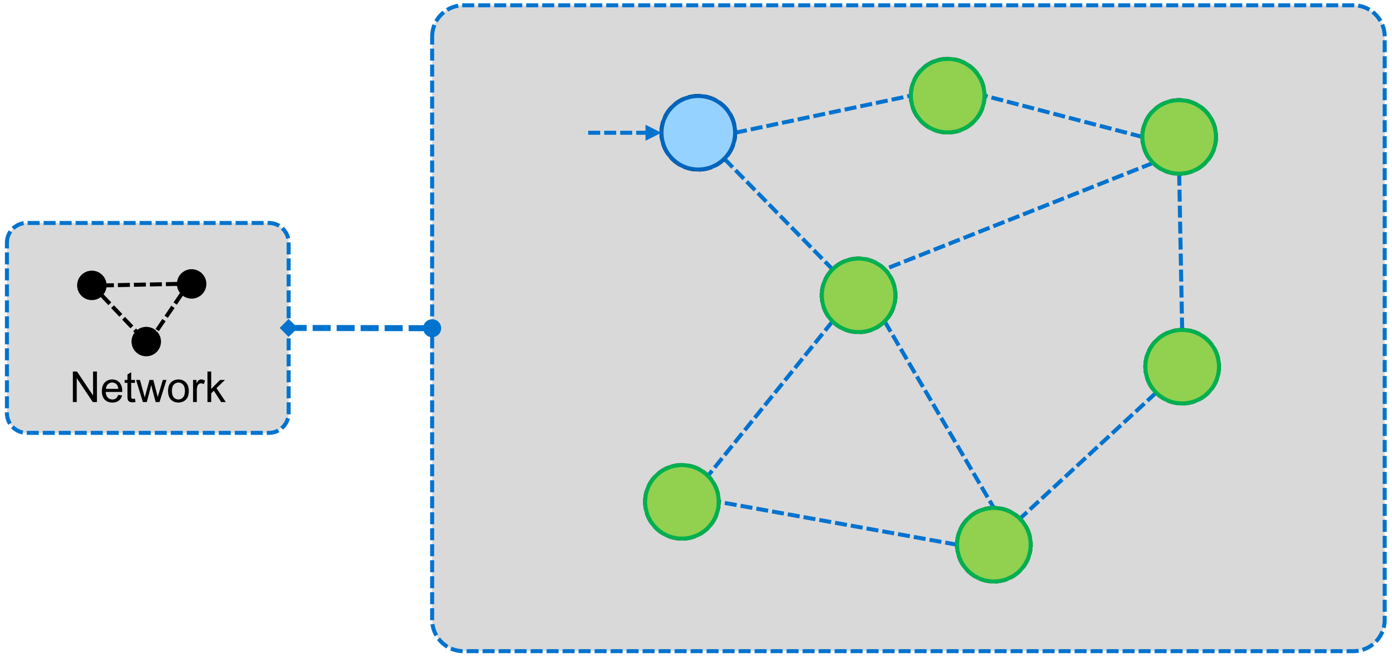


1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.

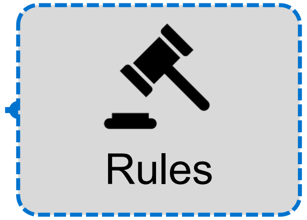
Rules

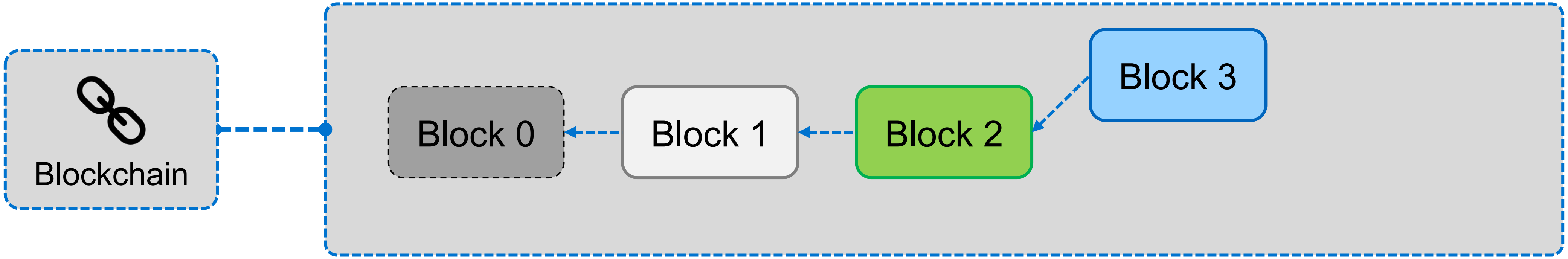
Blockchain

Block Propagation



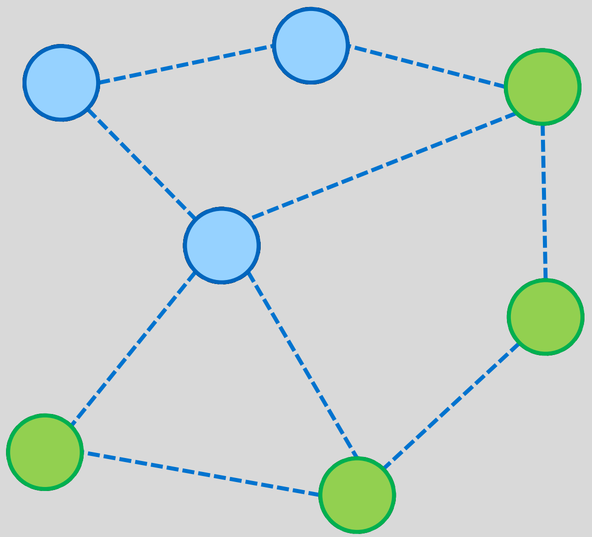
1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.





Block Propagation

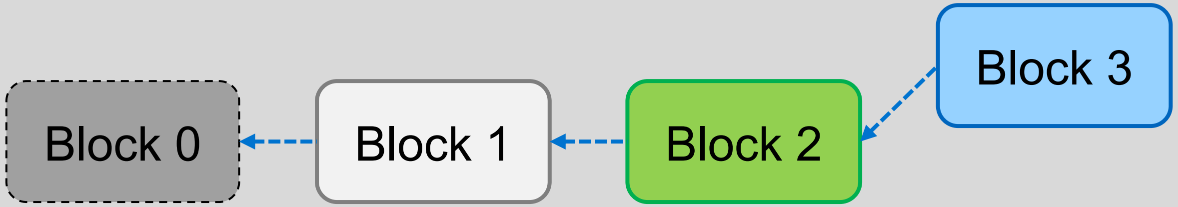
Network



1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.

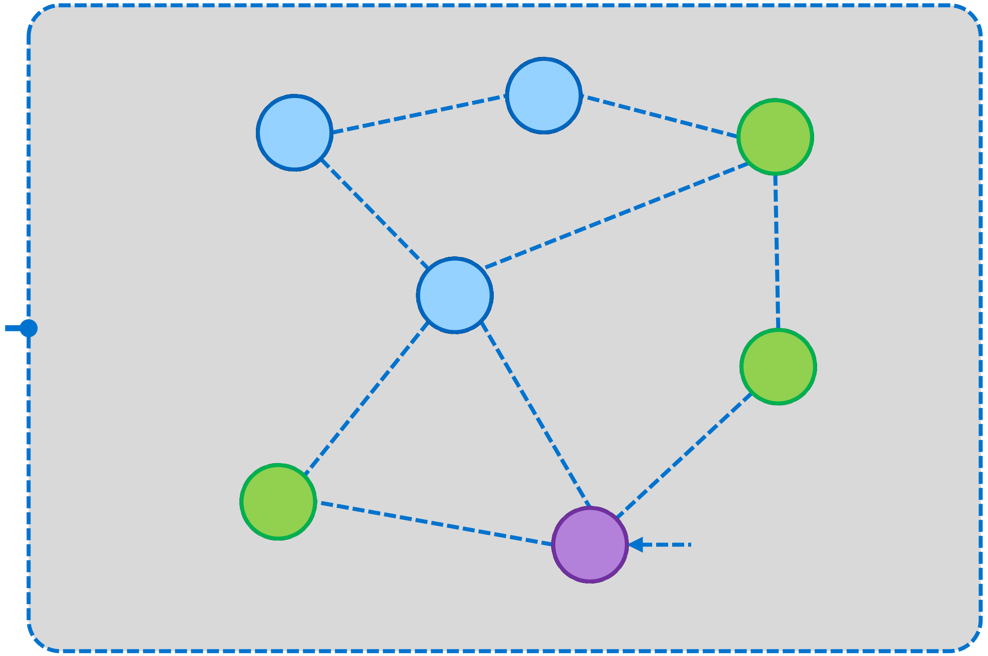
Rules

Blockchain



Block Propagation

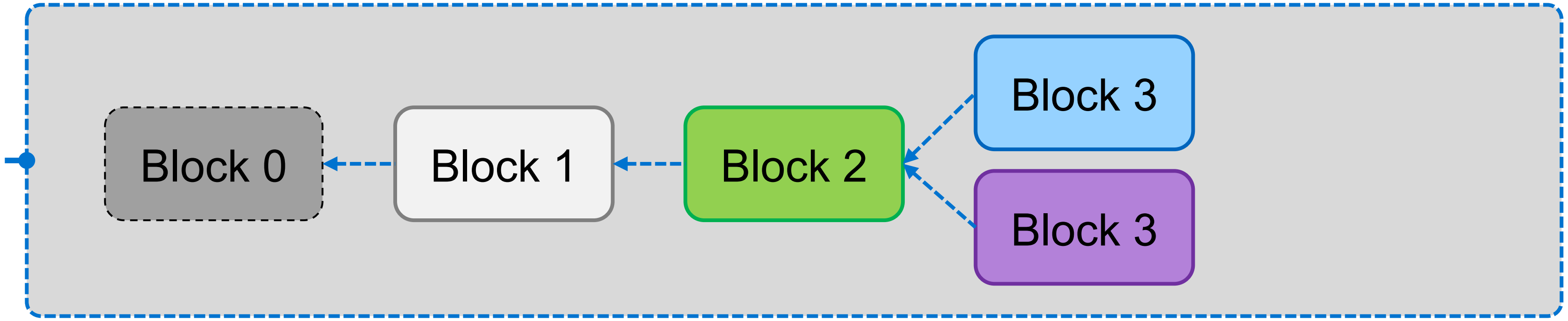
Network



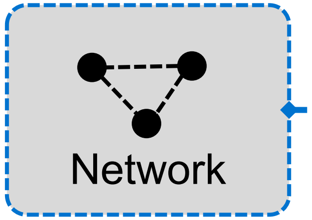
1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.

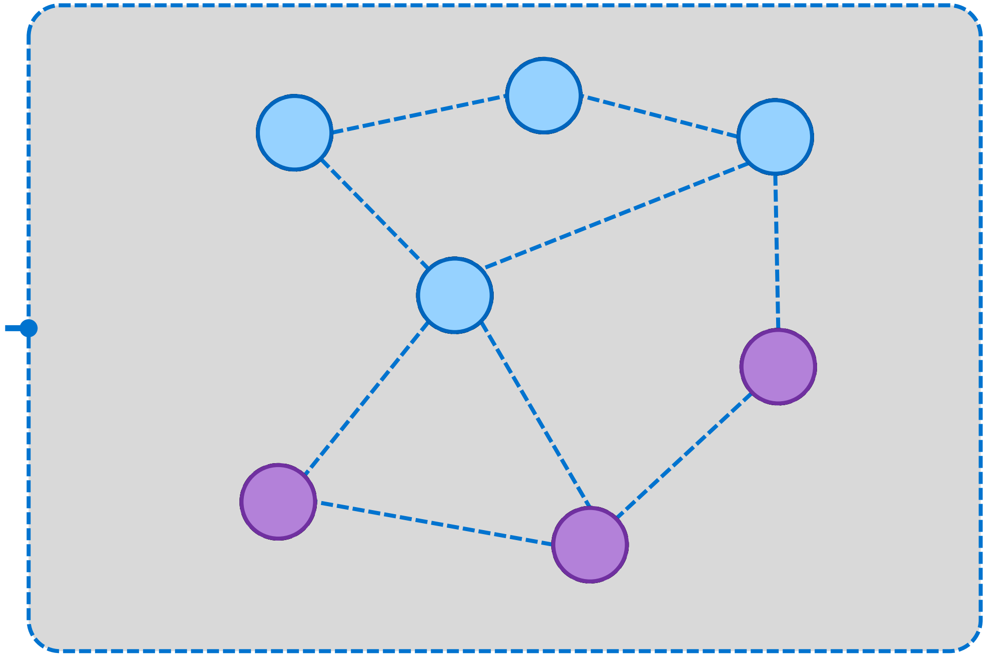
Rules

Blockchain



Block Propagation

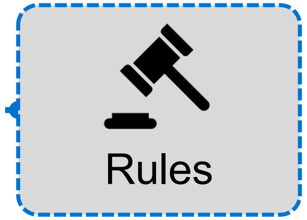


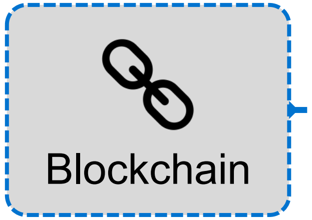


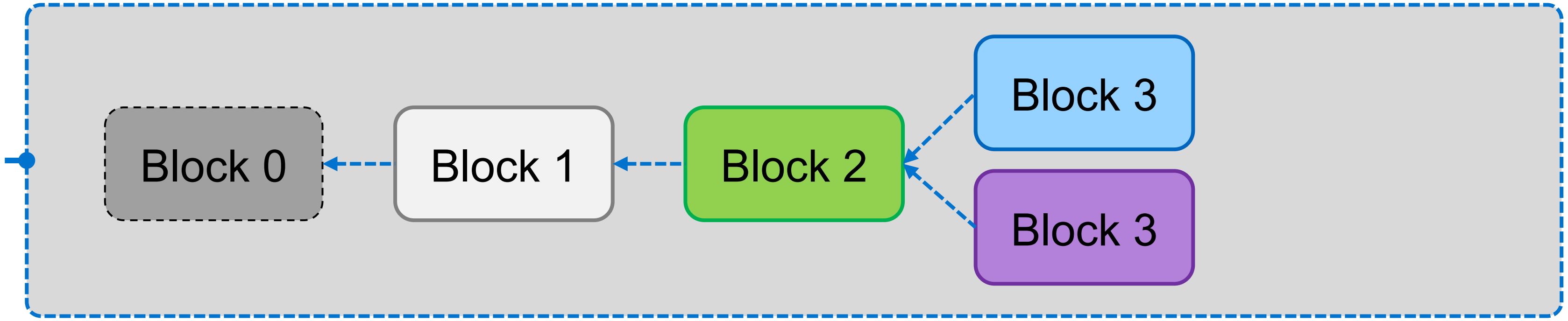
1. Only **valid** blocks are included in the blockchain.

2. The **longest chain** (with the highest block) wins.

3. A node builds on the first block it hears of.

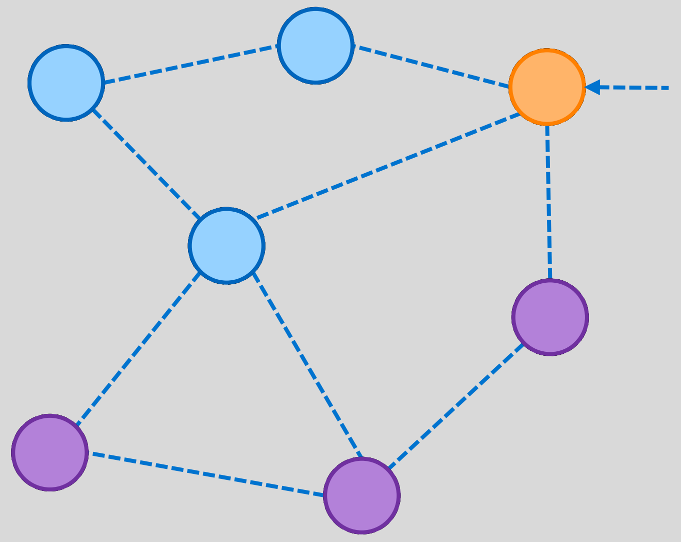






Block Propagation

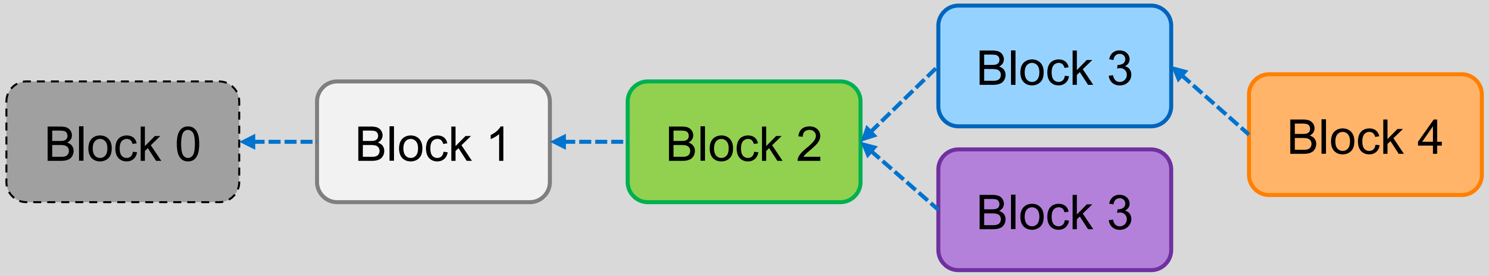
Network



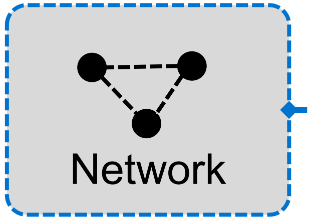
1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.

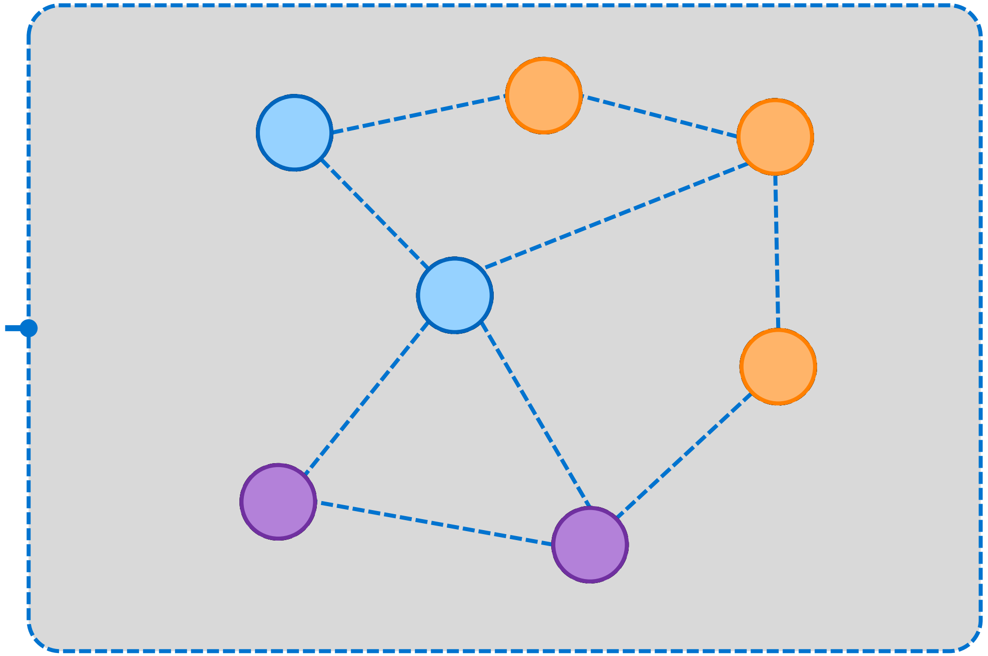
Rules

Blockchain

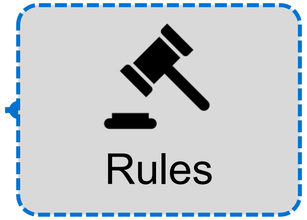


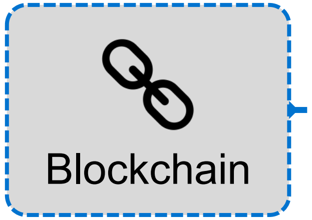
Block Propagation

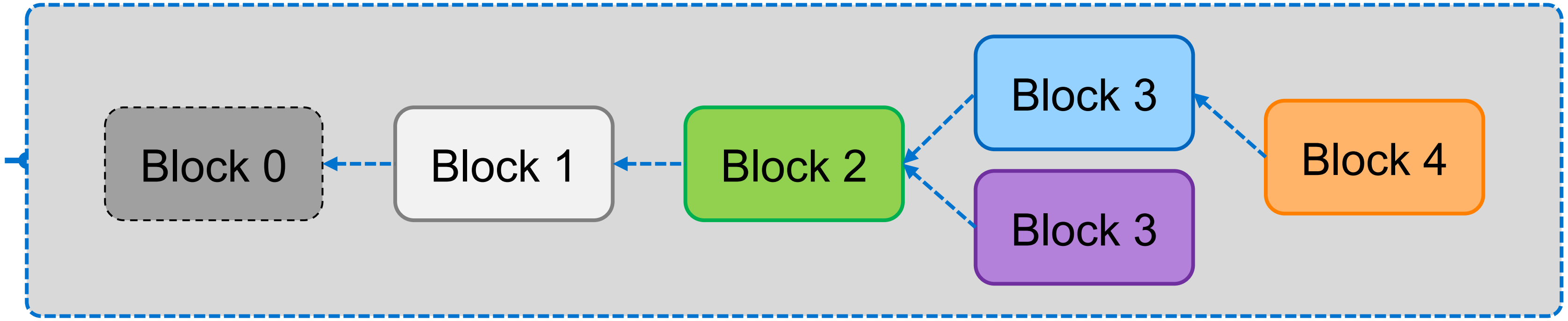




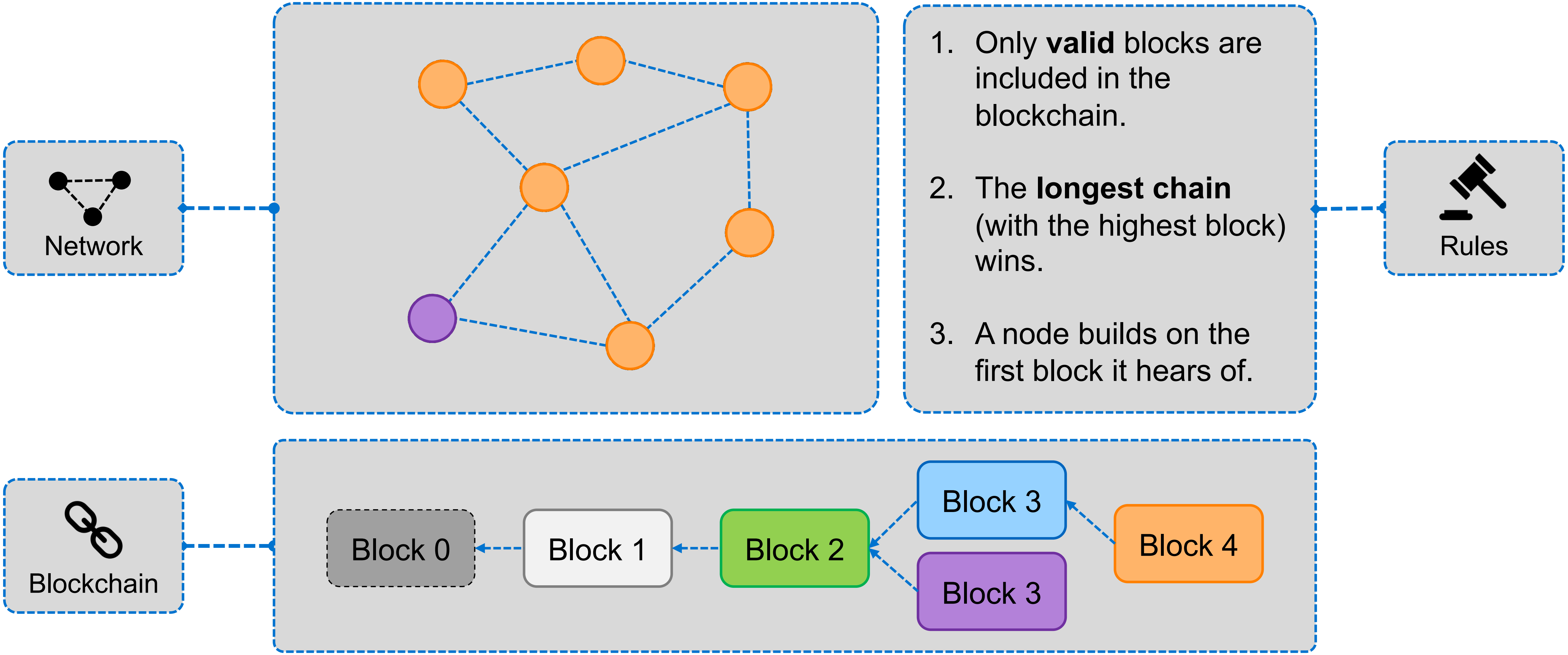
1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.



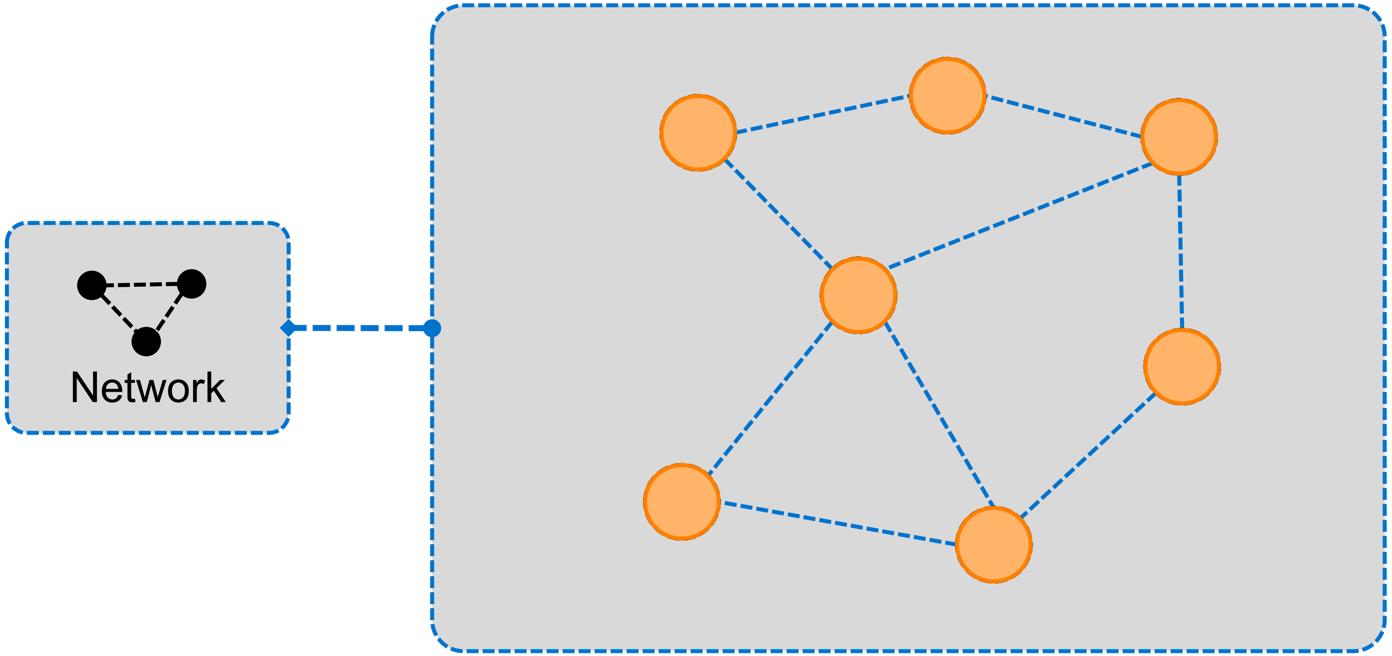




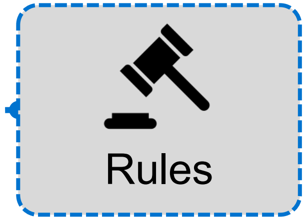
Block Propagation

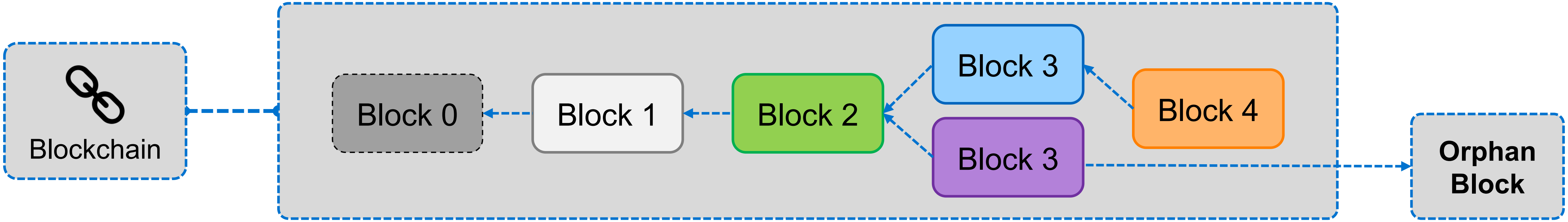


Block Propagation



1. Only **valid** blocks are included in the blockchain.
2. The **longest chain** (with the highest block) wins.
3. A node builds on the first block it hears of.





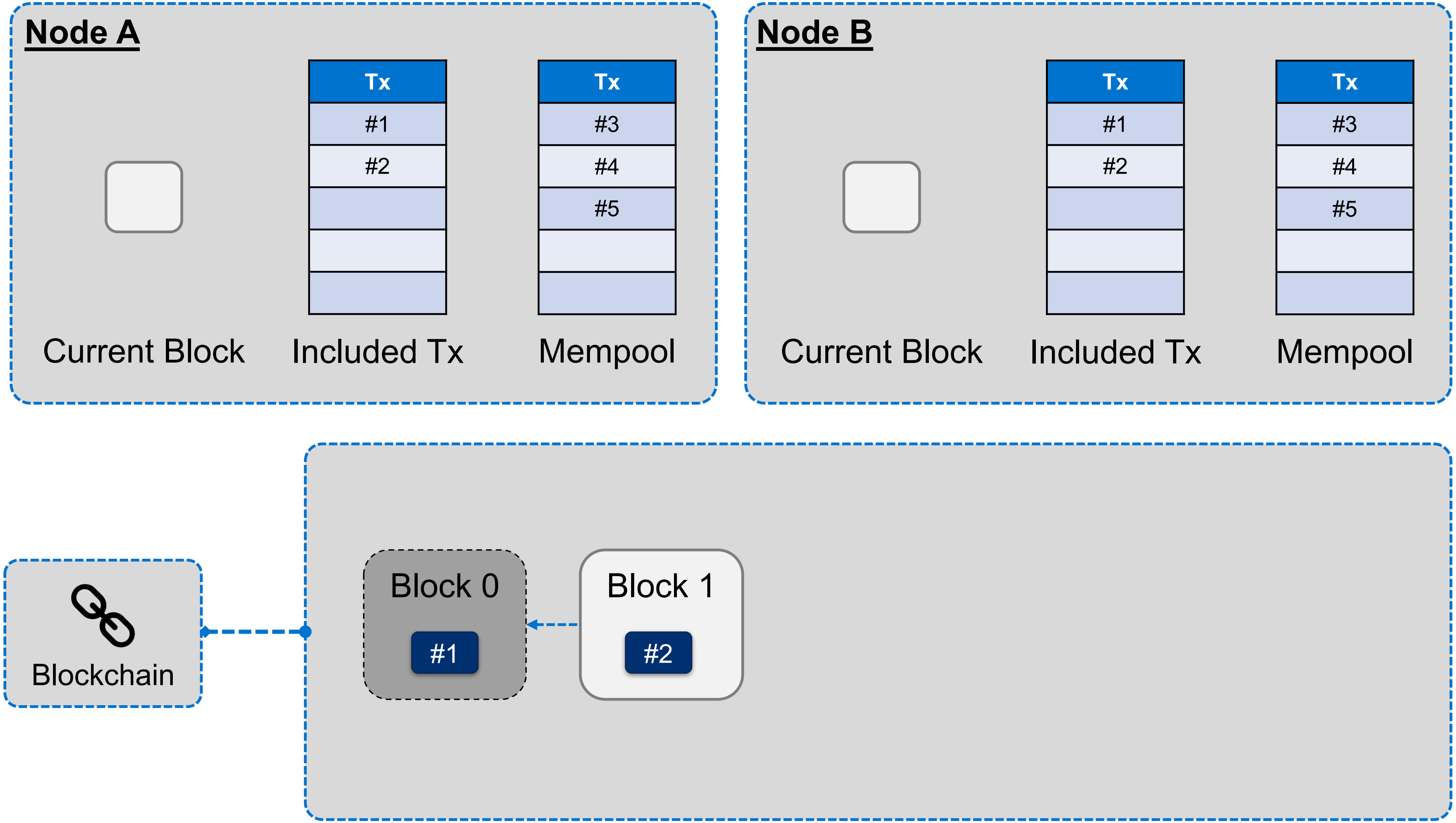
Transactions in Orphan Blocks

- An **orphan block** is a block that has been proposed in the network but has not been included in the longest chain.

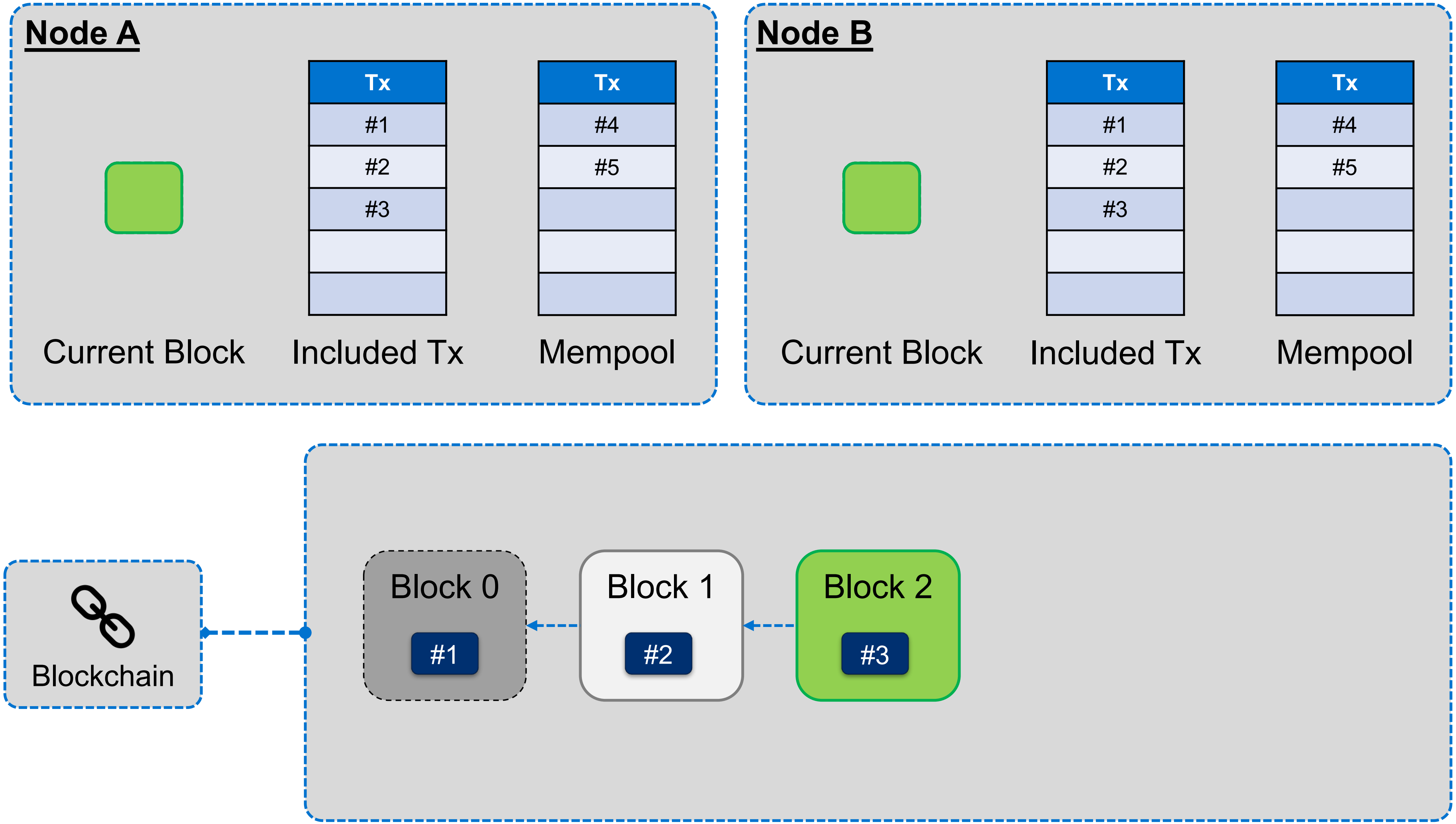
What happens to the transactions that are included in an orphan block?

- Unconfirmed transactions are stored in the mempool before they get added to a block.
- As unconfirmed transactions get “gossiped” in the network, every node will about all transactions.
- As a new block is proposed, all nodes update their mempool and remove the transactions which were included.
- As a consequence, the transactions in an orphan block are simply considered as unconfirmed, waiting to be included in a later block.

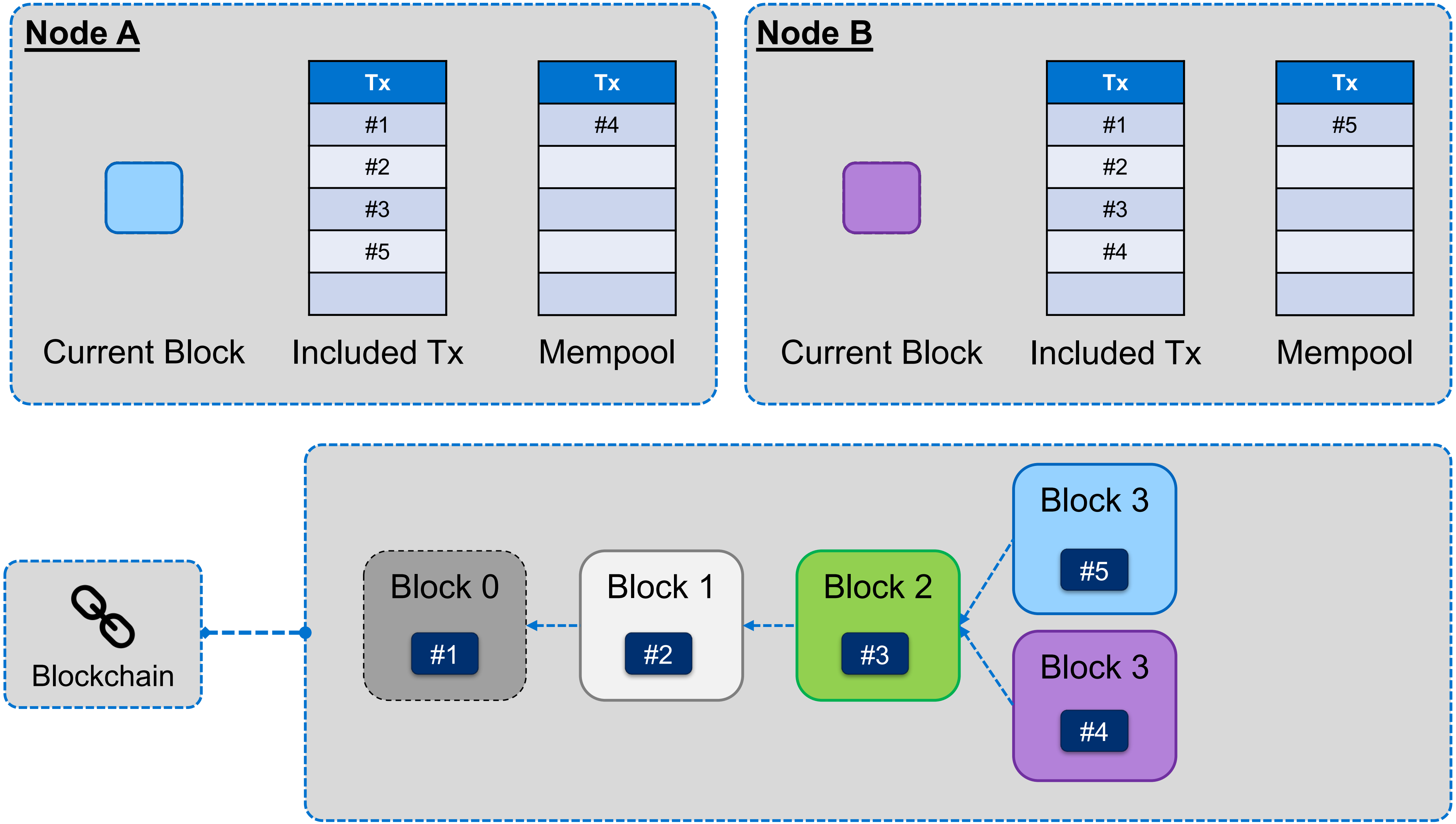
Transactions in Orphan Blocks



Transactions in Orphan Blocks



Transactions in Orphan Blocks



Transactions in Orphan Blocks

Node A

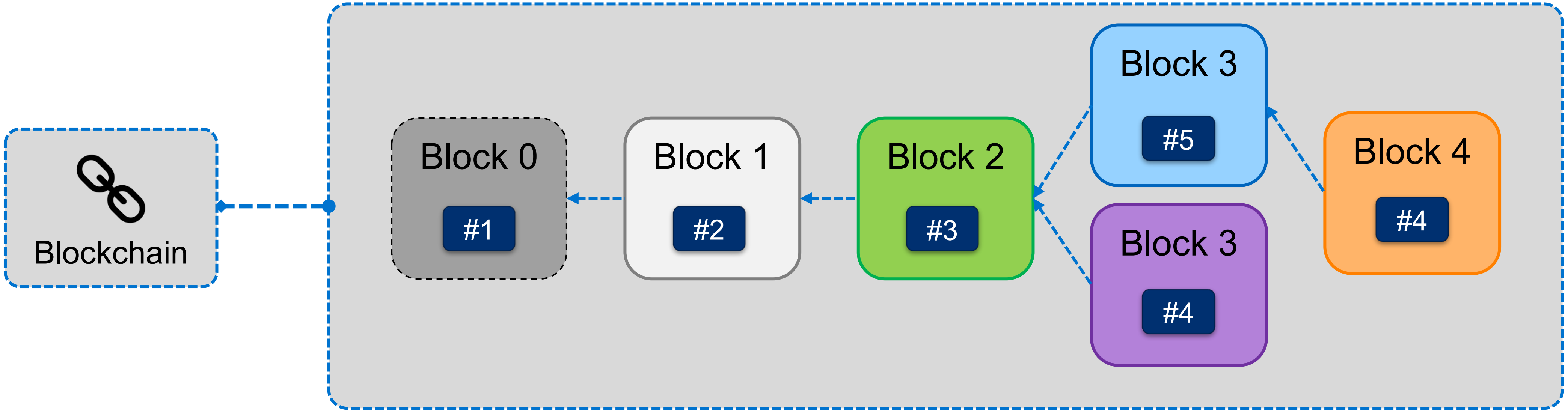
Included Tx (Tx)	Mempool (Tx)
#1	
#2	
#3	
#5	
#4	

Current Block · Included Tx · Mempool

Node B

Included Tx (Tx)	Mempool (Tx)
#1	
#2	
#3	
#5	
#4	

Current Block · Included Tx · Mempool



Consensus Mechanism ≠ Sybil Control Mechanism

- The term “consensus mechanism” is often used to refer to the mining puzzles; however, mining is not a direct concern of a consensus mechanism.
- A consensus mechanism deals with how the nodes in the protocol agree upon the history of events, while Sybil-control mechanisms (e.g., PoW - mining puzzles) aim to provide a fair leader selection.
- A permissionless blockchain requires **both** of them.
- Satoshi Nakamoto aimed to solve the permissionless consensus problem by using **Longest-chain consensus** and **PoW Sybil-control mechanisms**. This combination is also known as ***Nakamoto Consensus***.

Outline

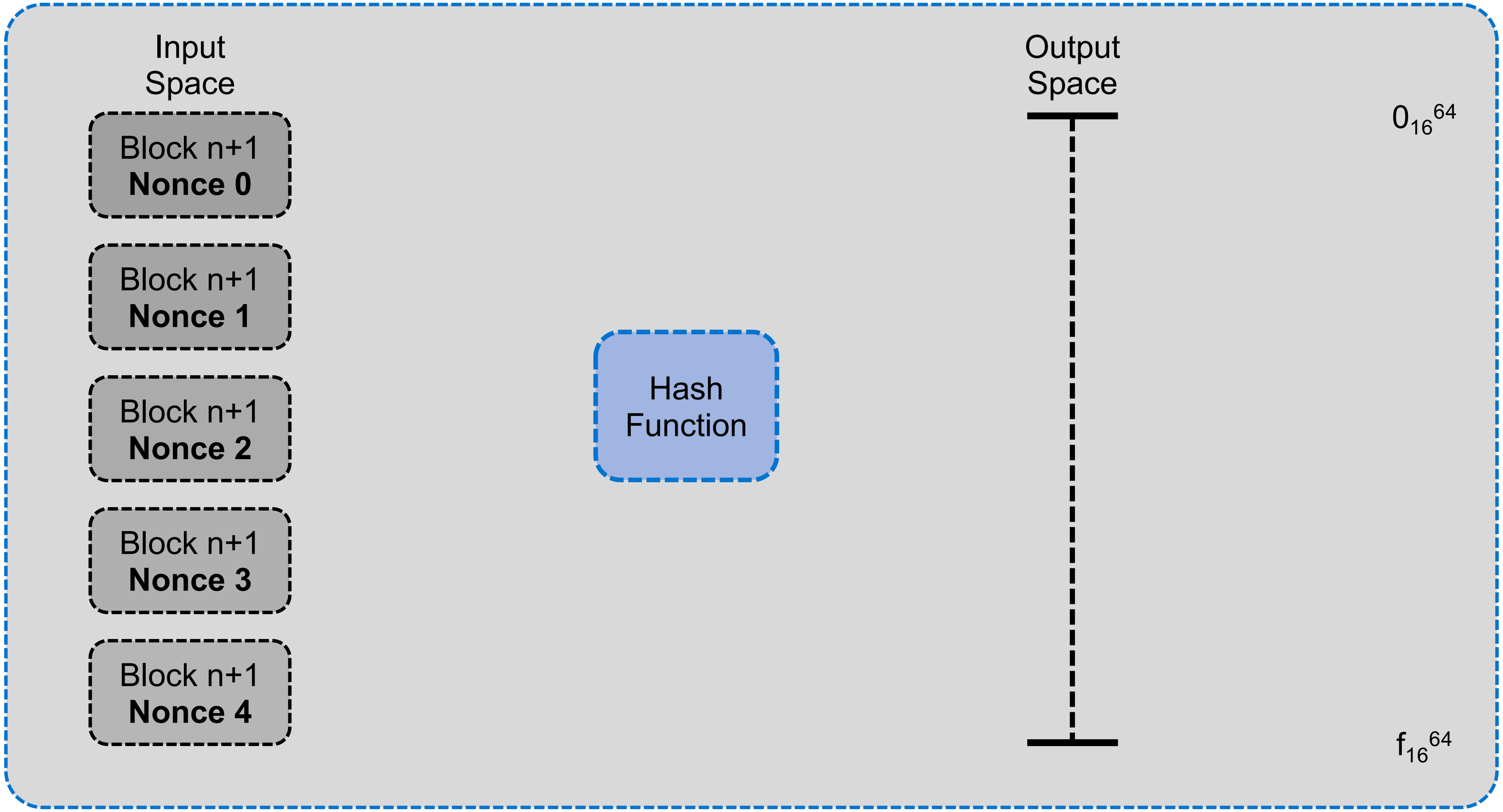
1. Consensus
- Consensus in Distributed Systems
- The Byzantine Generals Problem
- Choosing a Leader
- Sybil-Control Mechanisms
- Block Propagation
- Transactions in Orphan Blocks

2. Proof-of-Work (Mining)
- Search Puzzle
- Difficulty Determination
- Incentives
- Amount of Bitcoin
- Mining Hardware
- Mining Pools

This lecture covers parts of the chapter 02 of „Bitcoin and Cryptocurrency Technologies“ by Arvind Narayanan.

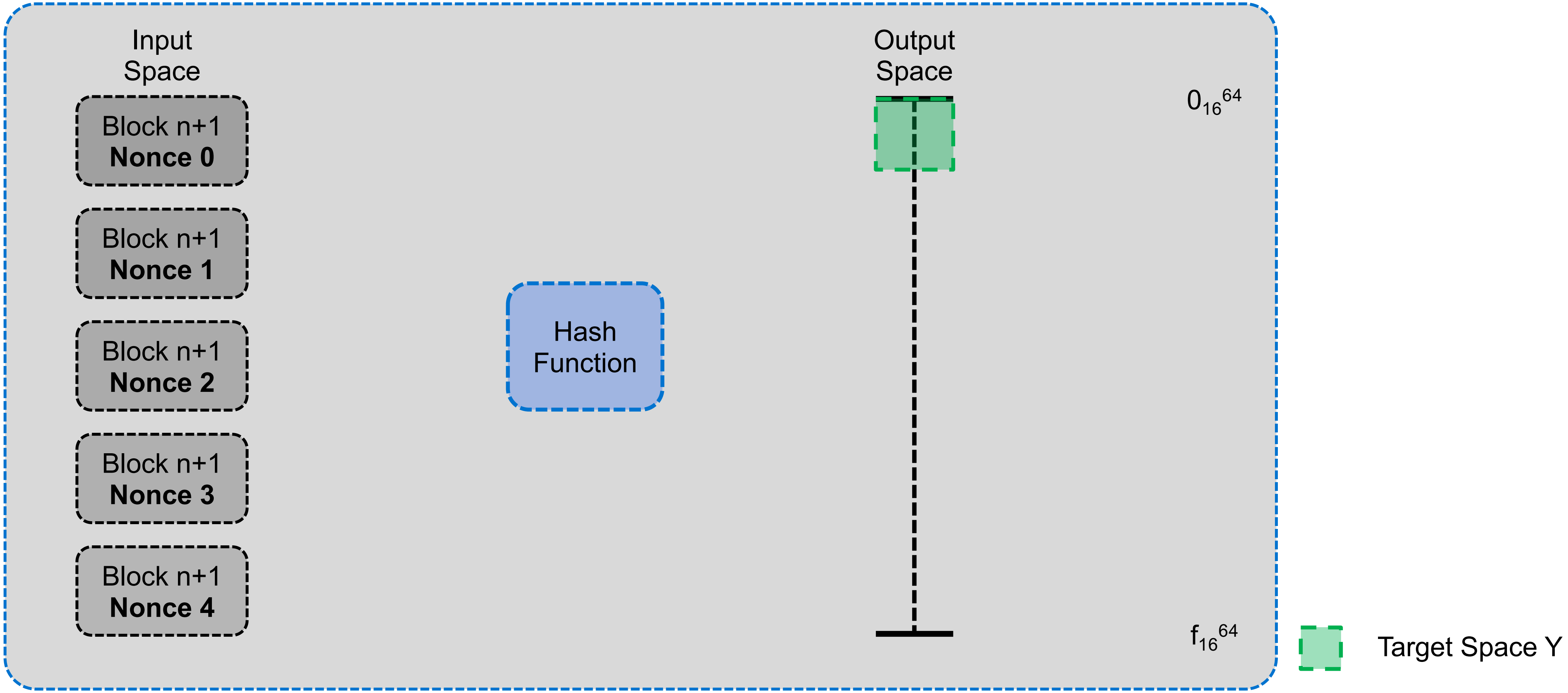
The Mining Puzzle – Proof of Work (PoW)

Idea: We use the search puzzle introduced in the chapter about cryptographic foundations. The header of the hash has to be included in Y. Bitcoin uses double SHA-256. (*sha256*(*sha256*(block)))



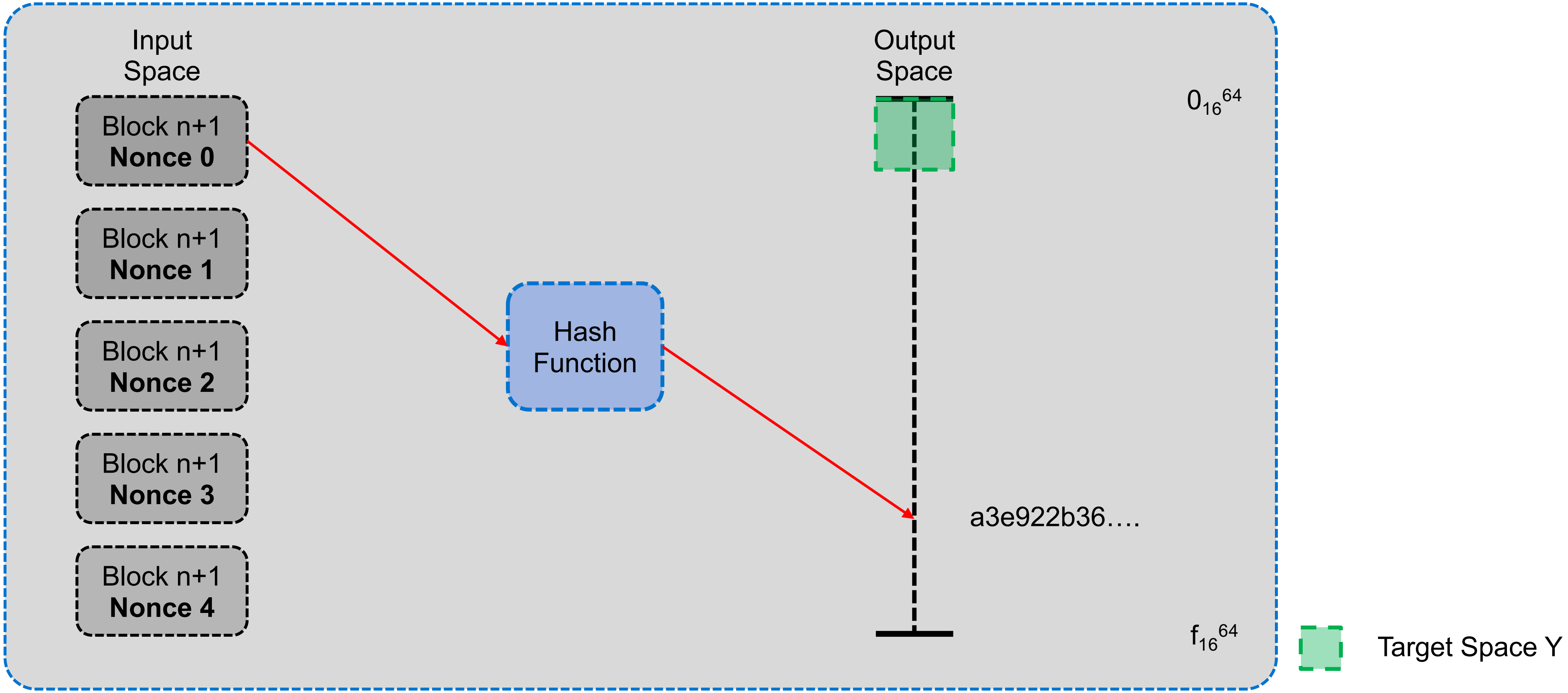
The Mining Puzzle – Proof of Work (PoW)

Idea: We use the search puzzle introduced in the chapter about cryptographic foundations. The header of the hash has to be included in Y. Bitcoin uses double SHA-256. (*sha256*(*sha256*(block)))



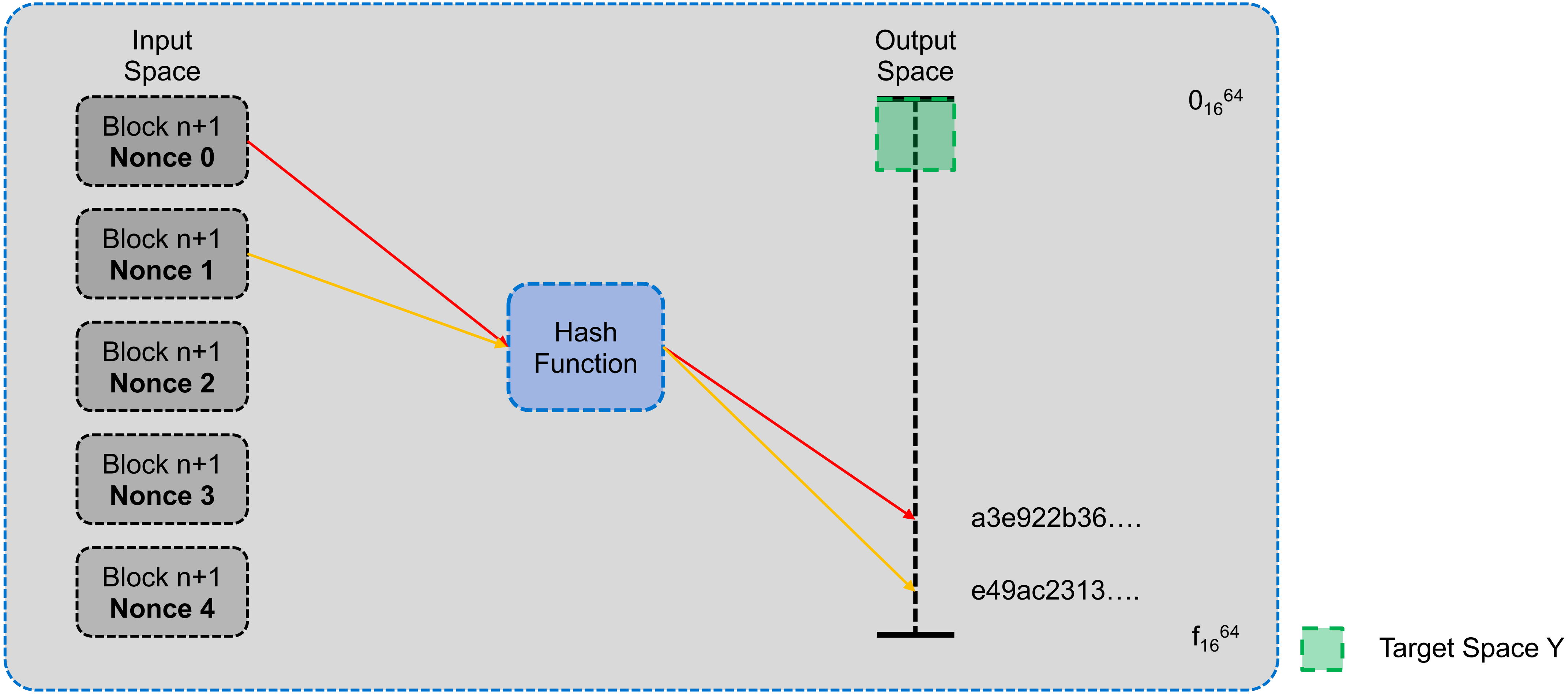
The Mining Puzzle – Proof of Work (PoW)

Idea: We use the search puzzle introduced in the chapter about cryptographic foundations. The header of the hash has to be included in Y. Bitcoin uses double SHA-256. (*sha256*(*sha256*(block)))



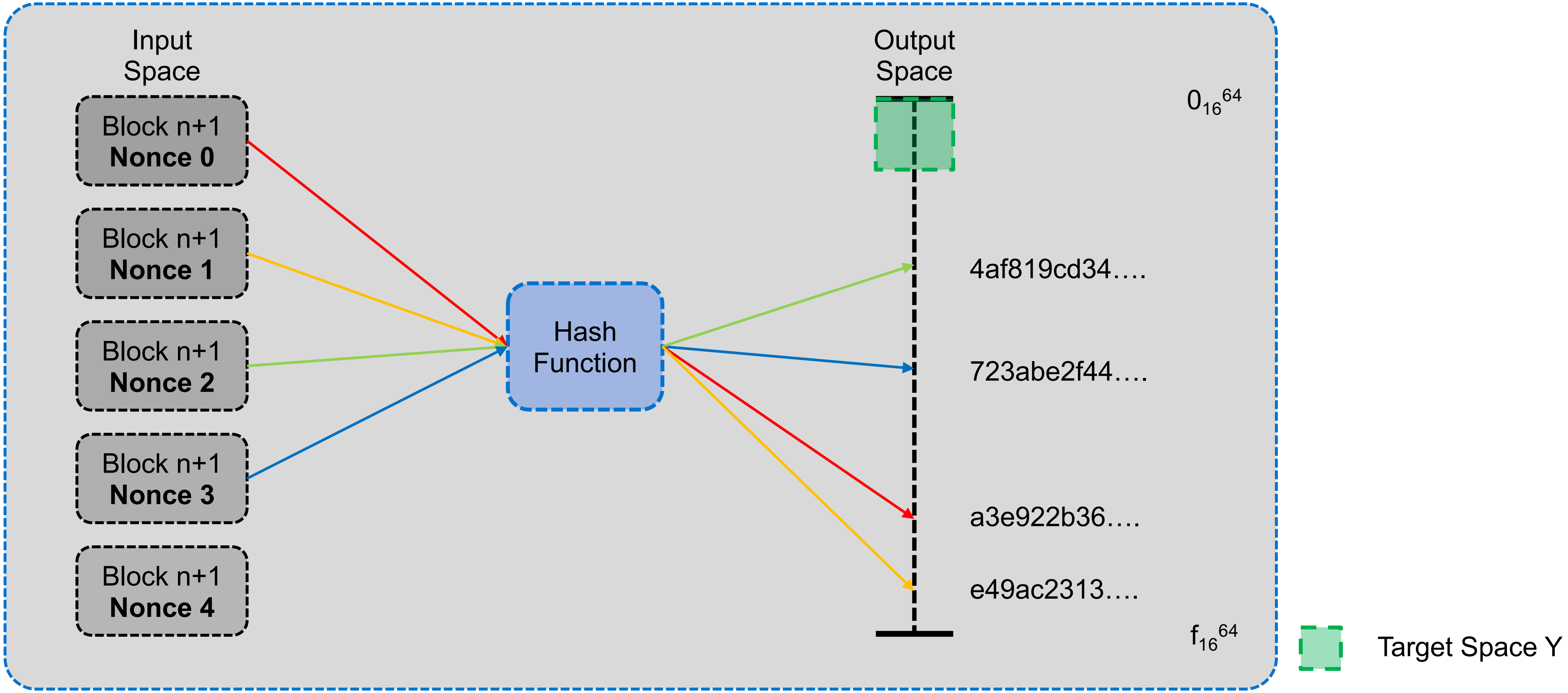
The Mining Puzzle – Proof of Work (PoW)

Idea: We use the search puzzle introduced in the chapter about cryptographic foundations. The header of the hash has to be included in Y. Bitcoin uses double SHA-256. (*sha256*(*sha256*(block)))



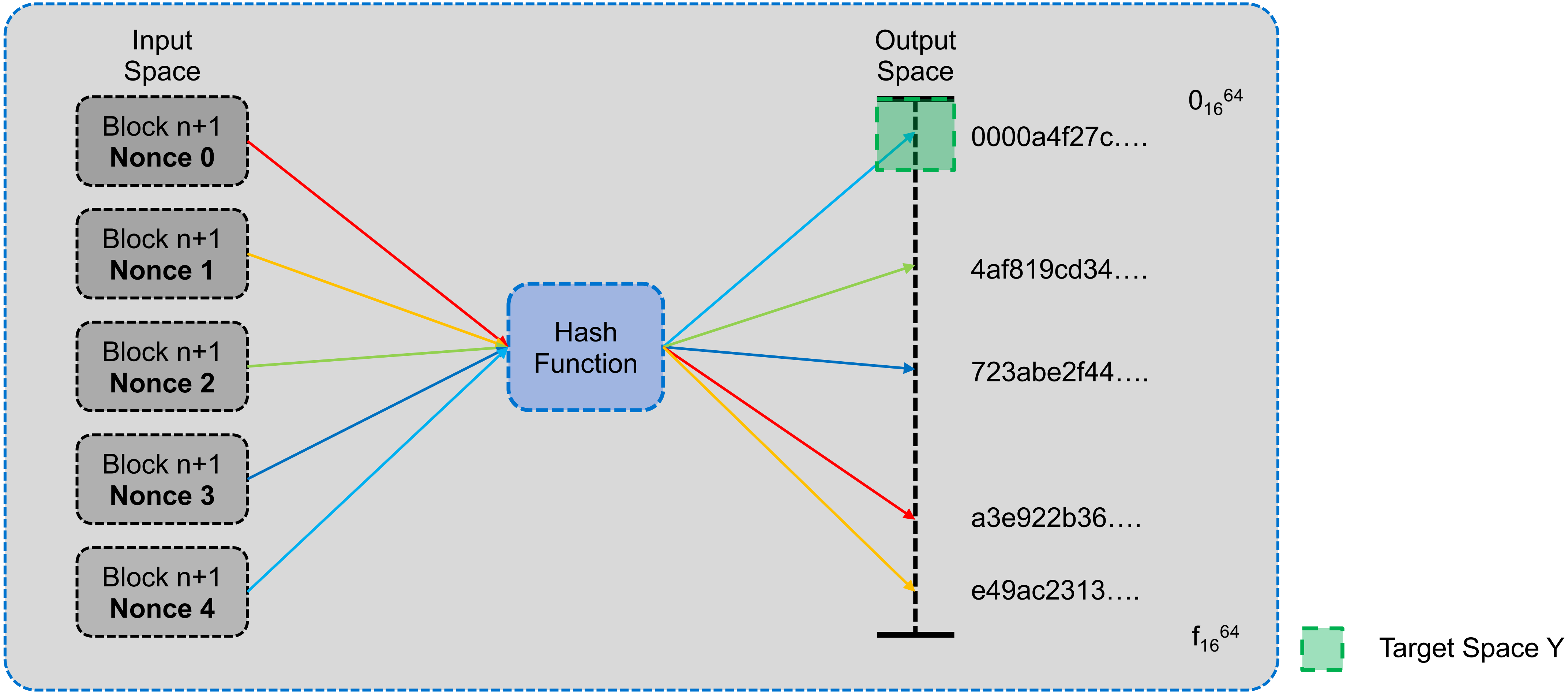
The Mining Puzzle – Proof of Work (PoW)

Idea: We use the search puzzle introduced in the chapter about cryptographic foundations. The header of the hash has to be included in Y. Bitcoin uses double SHA-256. (*sha256*(*sha256*(block)))



The Mining Puzzle – Proof of Work (PoW)

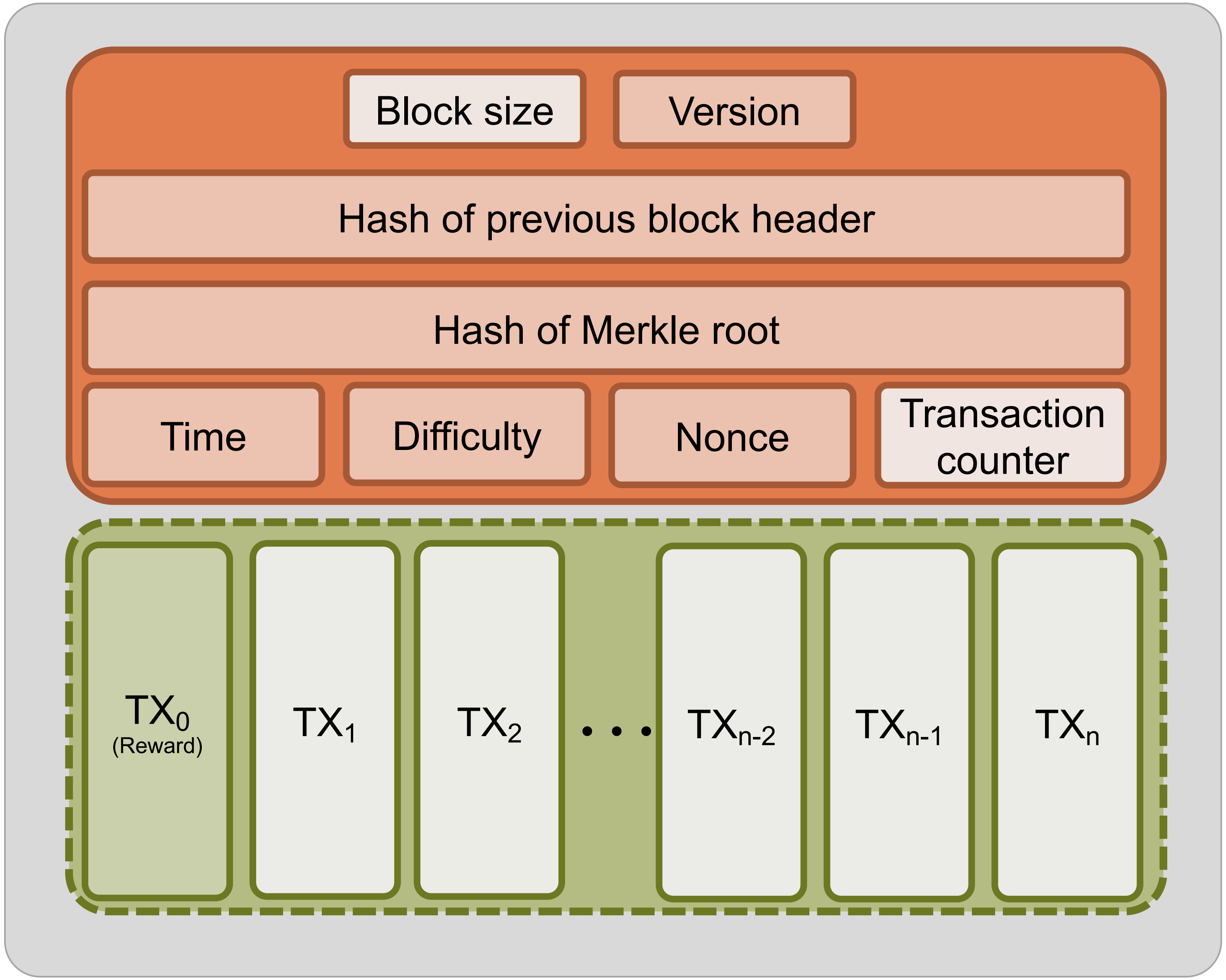
Idea: We use the search puzzle introduced in the chapter about cryptographic foundations. The header of the hash has to be included in Y. Bitcoin uses double SHA-256. (*sha256*(*sha256*(block)))



Does Everyone Have the Same Search Puzzle?

Recap:

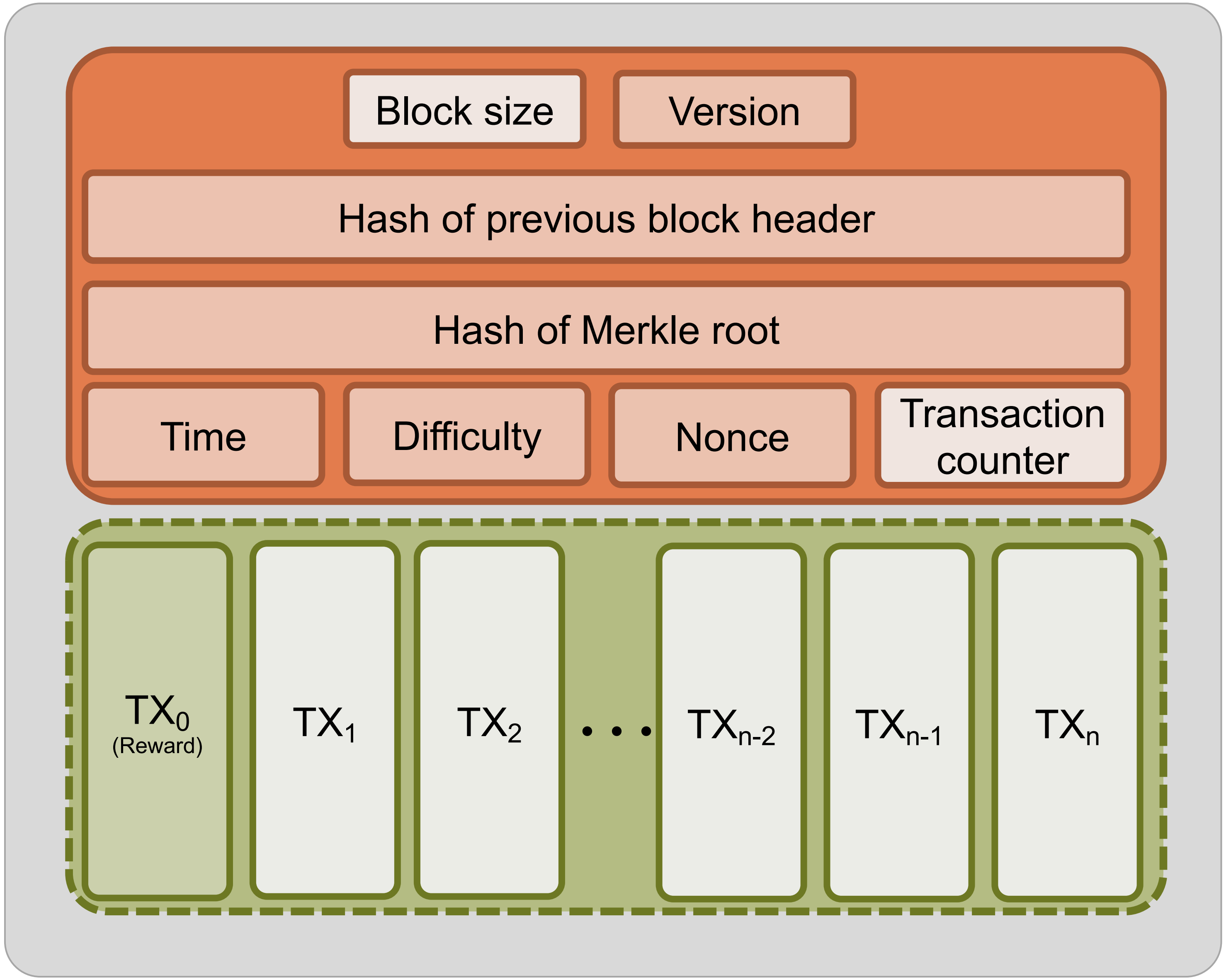
- The block's hash used for chaining is calculated from the *version* until the *nonce* field.
- Assume:
  - There are only 10 Tx in the memory pool. Every node includes all of them in the new block.
  - Every node uses the same time and version.
- Does everyone have the same search puzzle? If not, why?



Does Everyone Have the Same Search Puzzle?

Recap:

- The block's hash used for chaining is calculated from the *version* until the *nonce* field.
- Assume:
  - There are only 10 Tx in the memory pool. Every node includes all of them in the new block.
  - Every node uses the same time and version.
- Does everyone have the same search puzzle? If not, why?
  - No, every node has a different puzzle, as the TX_0 (the reward-address) is different from node to node.



Difficulty Calculation & Block Time

- The block time defines the average time between the creation of two blocks (In Bitcoin, block time = 10 minutes)
- **Why does the block time needs to be constant**?
  - Too slow:
    - Transactions take longer to be included
    - Network capacity decreases
  - Too fast:
    - Higher possibility of chain forking, leading to multiple “realities”.
    - Network has to keep track of these forks even if many will be orphaned.
    - Empty blocks
- How do we design the search puzzle in such way that it keeps a constant block time?
  - Every 2016 blocks, the difficulty of the puzzle is adapted to the current network speed.
- The longest chain is considered as the **chain with the highest accumulated difficulty**.

1. Measure, how long the last 2016 blocks took to get mined. (=T)
2. Calculate the factor of speed (two Weeks / T) (=F)
3. The difficulty gets increased (F > 1) or decreased (F < 1).

   3a. Maximum increase: 4. Maximum decrease: 0,25.
4. The process is done every 2016[1] blocks.

[1] 14 Days x 24 Hours x 6 (every 10 mins) = 2016

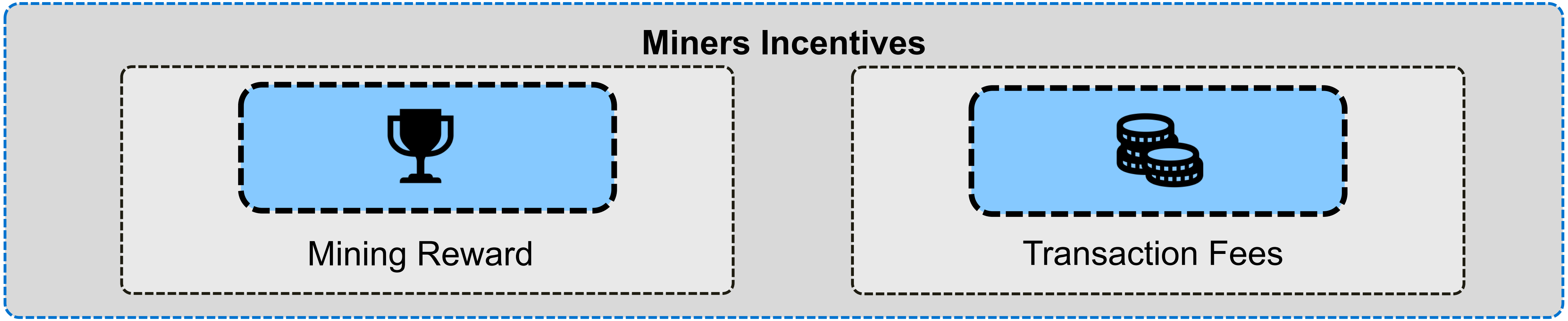
The Bitcoin Currency

Description	Bitcoin
Size of Data Field	8 Byte
Representation	Unsigned Integer
Smallest Unit	1 Satoshi
Base Unit	1 BTC = 100.000.000 Satoshi
Maximum Amount of BTC	20.999.999,9769 BTC

Mining Puzzle

"Why would anyone waste energy on solving a stupid puzzle?"

Because of incentives!



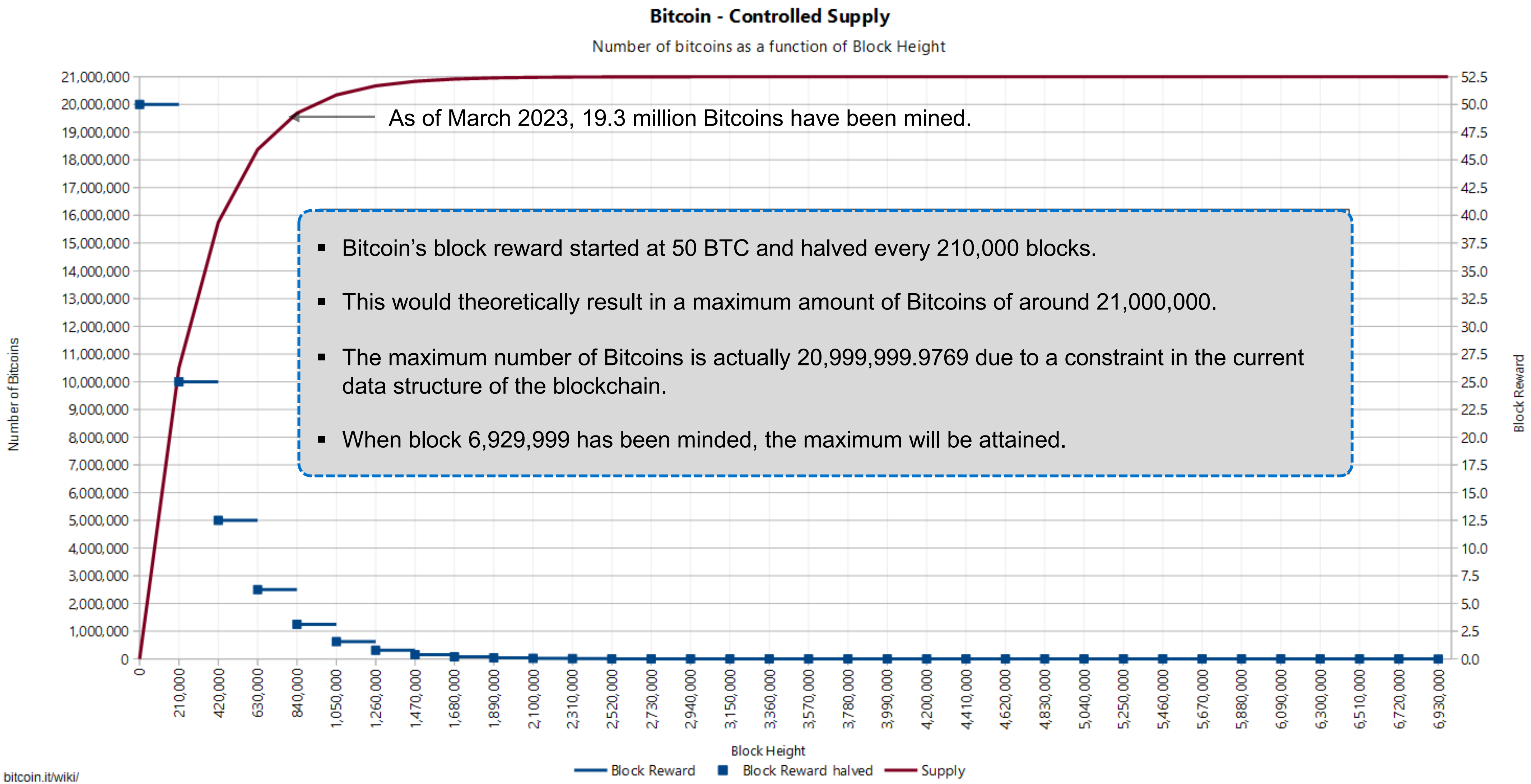
Mining Reward

- For a newly created block, the miner is allowed to issue new Bitcoins to his wallet.
- The mining reward was **6.25 Bitcoins** as of November 2021. This incentive, which was originally 50 Bitcoins, is cut in half roughly every four years or after each set of 210,000 blocks are mined. This is known as **halving** and it limits the total global supply of Bitcoin, so prices could rise if demand remains strong.

Transaction Fee

- Every transaction includes a transaction fee.
- It is the difference between all inputs and outputs.

Upper Bound of Bitcoins

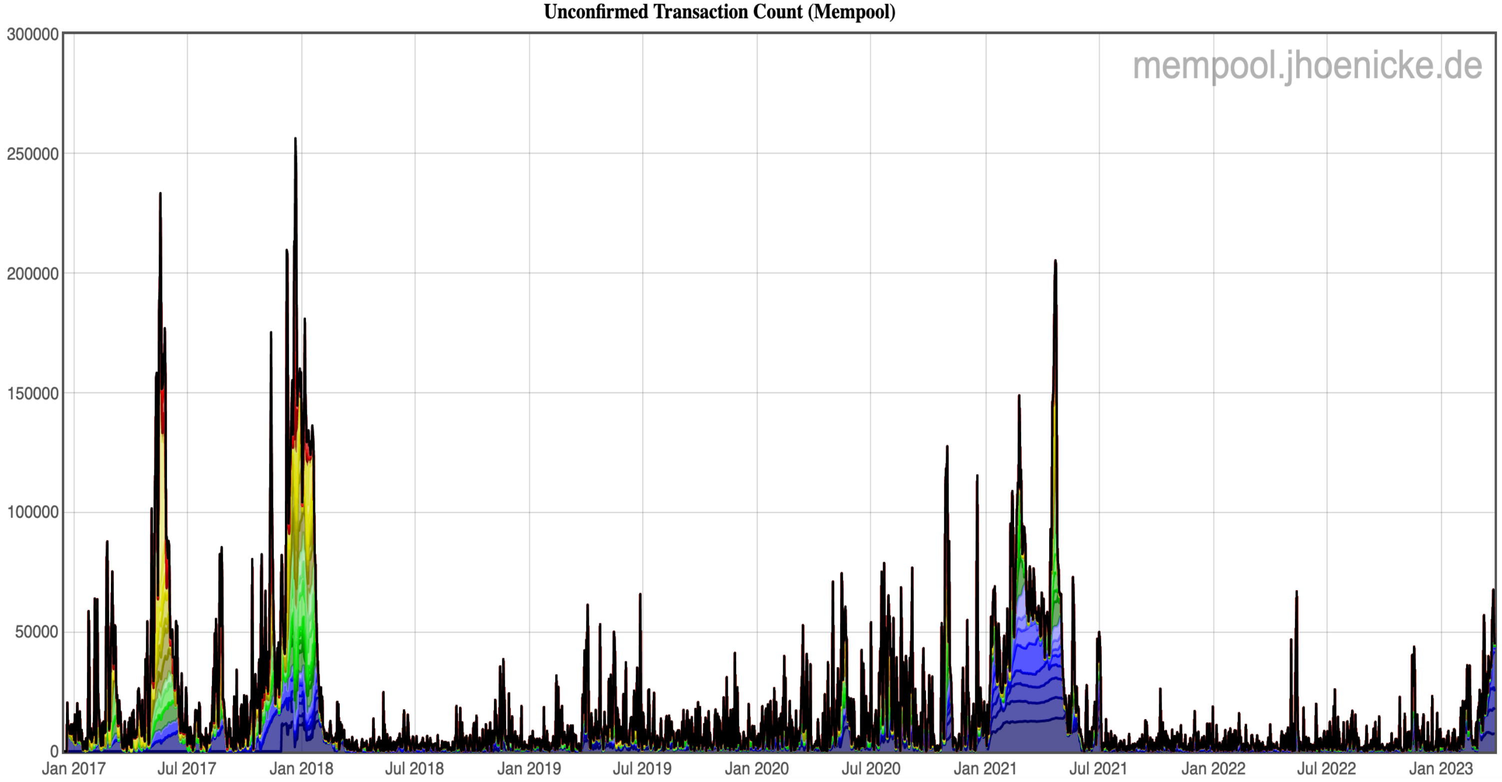


- Bitcoin's block reward started at 50 BTC and halved every 210,000 blocks.
- This would theoretically result in a maximum amount of Bitcoins of around 21,000,000.
- The maximum number of Bitcoins is actually 20,999,999.9769 due to a constraint in the current data structure of the blockchain.
- When block 6,929,999 has been minded, the maximum will be attained.

bitcoin.it/wiki/

Transaction Fee

- Every transaction includes a transaction fee. In Bitcoin, it is the **difference between all inputs and outputs**.
- The miner of the block obtains the transaction fees in addition to the block reward.
- The network can only process between 3 and 7 transactions per second, therefore **some transactions have to wait longer**.
- **The higher the transaction fee, the faster the transaction gets included in the blockchain**.
- The miners are incentivized to mine the high-fee transactions first.
- The fee is calculated in Satoshi[1]/byte.



[1] 1 Satoshi equals 10^{-8} Bitcoin. It is the smallest value in the Bitcoin network.
The website representing this graph can be found here: https://jochen-hoenicke.de/queue/

Coinbase Transaction

- The coinbase transaction is the first transaction in a block
  - It has a Txin that references no Txout (called **coinbase**)

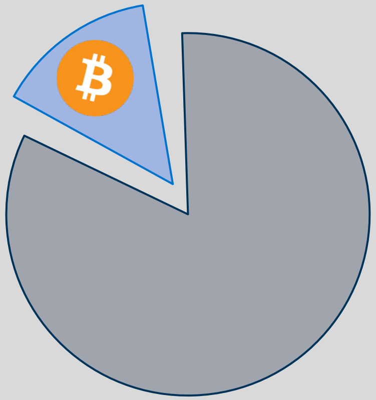
- The miner who finds the block is entitled to the coinbase transaction and therefore the block reward consisting out of
  - Block reward (newly available Bitcoins which are introduced in the system)
  - Transaction fees

- The contents of the coinbase transaction are
  - The block height
  - Up to 100 arbitrary bytes that can be put into the transaction input (*scriptSig*)

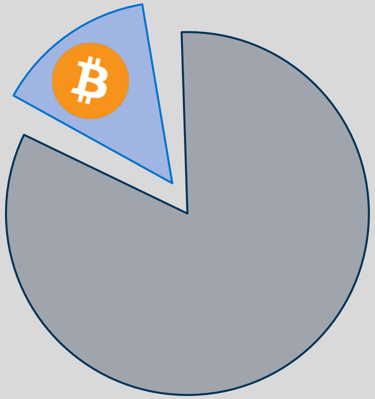
Arms Race in Mining

- The process of mining can be a profitable business. However, there are some remarks:
- Approximately 900 new Bitcoins are mined per day. On a daily basis, 144 new blocks are produced. If computing power increases, the difficulty of the search puzzle increases, too.
  ➔ The overall output of the mining reward stays the same.
- Example
  10 entities own hardware with 10.000 Th/s. Each receives roughly 180 Bitcoins per day. Each of them decides to double their hash rate to gain more Bitcoins. Now everyone has 20.000 Th/s, but still earns 180 Bitcoins per day as his share stays at 10% of the network hash rate.
- ➔ If a miner wants to increase his revenue, it has to invest more than the others. As everyone thinks this way (otherwise his revenue will decrease), this leads to an **arms race.**

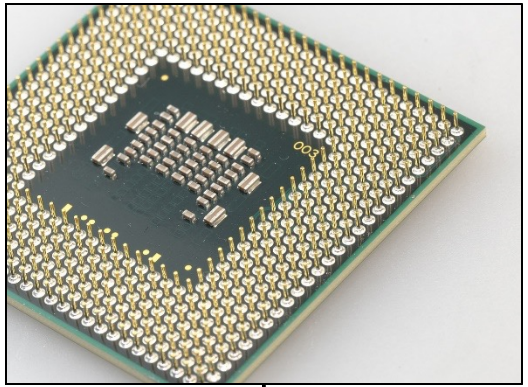
Situation 1: 10.000 Th/s



Situation 2: 20.000 Th/s



Mining Hardware

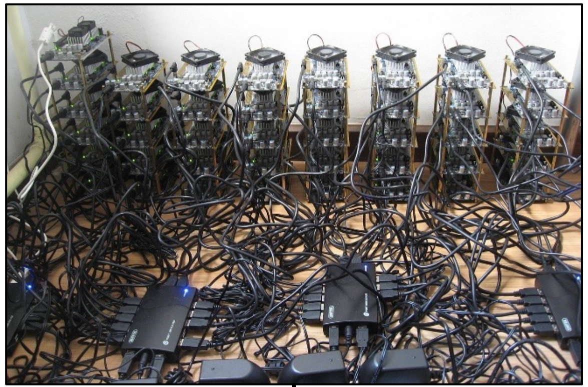


2009
CPU

CPUs were the first hardware to mine Bitcoins.

2010
GPU

GPUs are faster than CPUs. First mining software was introduced in 2010.



2011
FPGA

FPGA (field programmable gate array) are much more energy effective than GPUs.

2013
ASIC

ASIC (application-specific integrated circuit) are chips specially designed for mining. Fastest mining.

FPGA image taken from https://en.wikipedia.org/wiki/File:Icarus_Bitcoin_Mining_rig.jpg by Xiangfu, cropped. CC BY-SA 4.0
ASIC image taken from https://commons.wikimedia.org/wiki/File:Cryptocurrency_Mining_Farm.jpg by Marco Krohn, not modified. (CC BY-SA 4.0)

Mining Hardware and Difficulty

satoshi
Founder
Sr. Member

Activity: 364
Merit: 1224

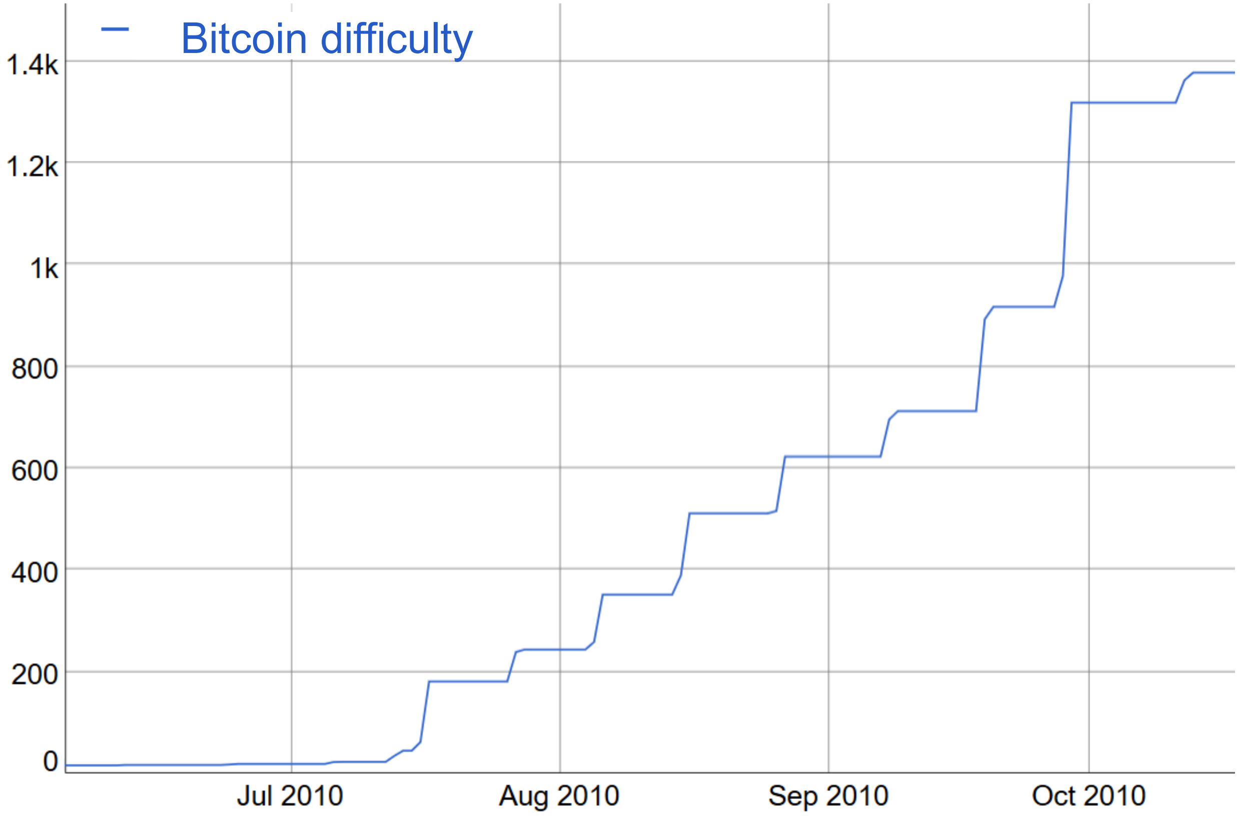
Re: A few suggestions
December 12, 2009, 05:52:44 PM #10

The average total coins generated across the network per day stays the same. Faster machines just get a larger share than slower machines. If everyone bought faster machines, they wouldn't get more coins than before.

We should have a gentleman's agreement to postpone the GPU arms race as long as we can for the good of the network. It's much easer to get new users up to speed if they don't have to worry about GPU drivers and compatibility. It's nice how anyone with just a CPU can compete fairly equally right now.

- Satoshi suggested in December 2009 a gentleman's agreement to postpone the "arms race" that would come with the introduction of mining software for GPUs.

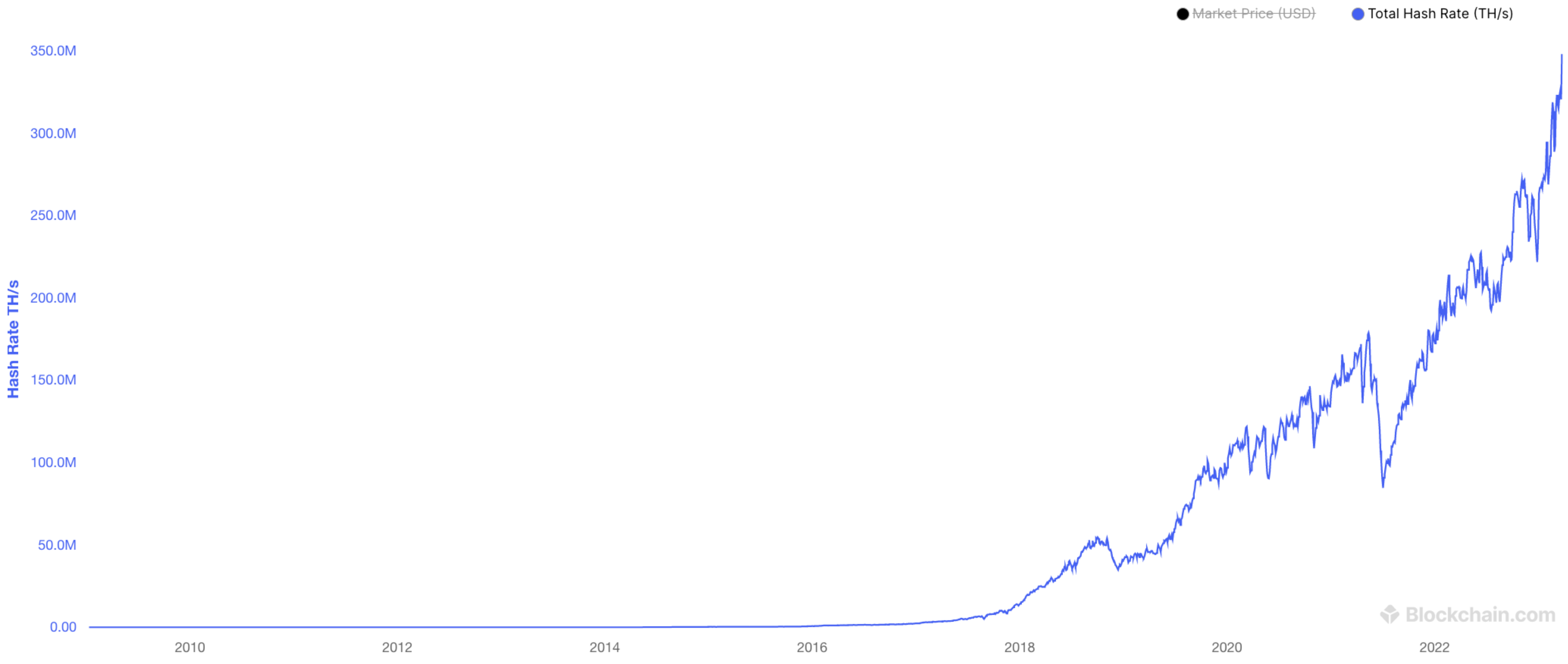
- As of 2010, this agreement was broken, the first GPU-miners were used, and the difficulty rose.



At the moment, ~2,8*10^19 attempts per second => 16 zetta hashes attempts / block.

See this post of Satoshi Nakamoto: https://bitcointalk.org/index.php?topic=12.msg54#msg54
A historic chart of Bitcoin difficulty: https://bitinfocharts.com/de/comparison/bitcoin-difficulty.html

Mining Hardware and Difficulty (cont.)



The graph of the hash rate: https://blockchain.info/de/charts/hash-rate?timespan=all

Mining Pools

With increasing difficulty, miners face problems:

- Hardware costs are high (high fixed costs)
- Electricity and cooling costs are high (high variable costs)

- Decreasing market share (own hash rate vs. overall hash rate)
- A block is either found or not → no condolence reward

Solution:

- Miners work together in mining pools to stabilize their monthly income
- A pool is organized by the pool manager

Assume percentage of overall hash rate:
`1 / 2.000.000 = 0,0000003 (0,00003%)`

Blocks proposed per year:
`6 * 24 * 365 = 52.560 blocks`

Expected number of blocks per year:
`0,0000003 * 52.560 = 0,0158 Blocks / year`

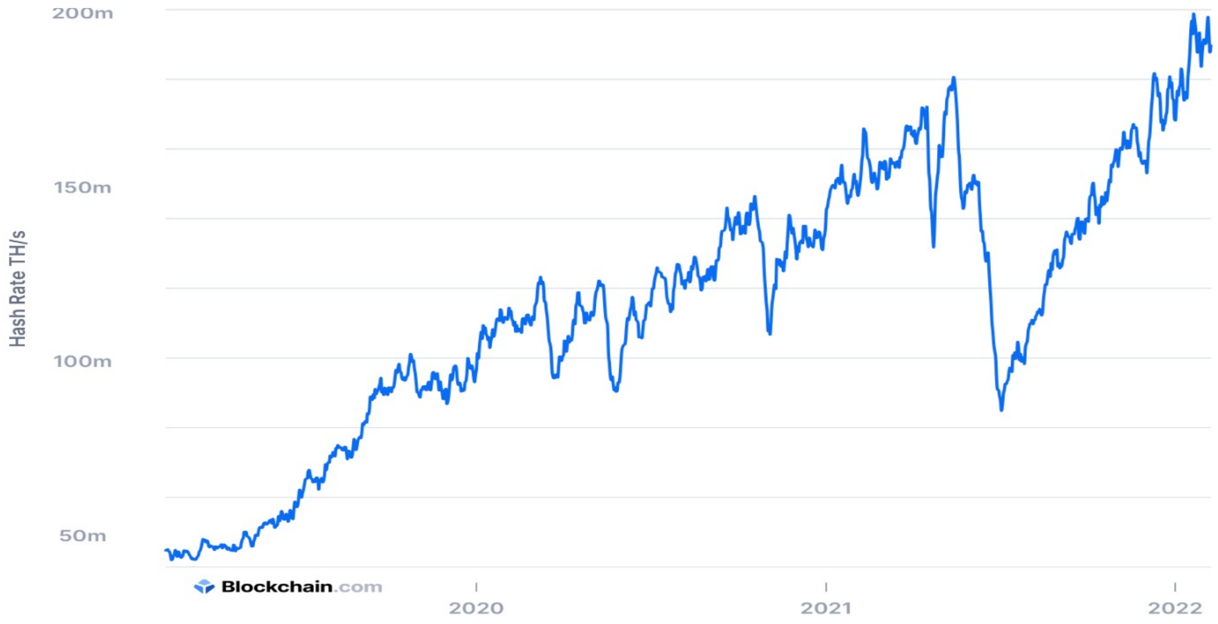
How does a mining pool work?

- The pool manager proposes for each new block height a “block prototype” to his pool, requiring the PoW done. (TX_0 → pool manager)
- Near-solutions are sent to the pool manager to prove some work. Each proof is called a share.
- Solutions are also sent to the pool manager and pushed into the network.
- Pool manager receives mining reward and distributes it among the shares, keeps a fee.

Difficulty vs Hash Rate

- The difficulty adapts to the hash rate.

Bitcoin Hash Rate vs Difficulty (3 years)



Screenshot from https://www.blockchain.com/charts/hash-rate. Accessed in 2022.

Screenshot from https://www.coinwarz.com/mining/bitcoin/difficulty-chart. Accessed in 2022.