Consensus in Bitcoin

Blockchain-based Systems Engineering

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

1. Consensus
- Imagine - Bitcoin with a Central Authority
- Random Dictatorship
- Sybil Control Mechanism
- Byzantine Fault Tolerance
- 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

Imagine[1] - Bitcoin with a Central Authority (CA)

If we would have designed a digital currency with a central authority, what would it look like?

- 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.

Bitcoins aims to democratize the financial world, however, this approach would lead to a “dictatorship”.

[1]There is no Bitcoin with a central authority, this is just a thought experiment.

Random Dictatorship

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 block creation.

How do we do that?

- Looking at social choice theory, we could establish an agreed-upon world state by fairly randomizing a dictator, creating a "random dictatorship".[1]

Problem:

Now, we are facing the problem of fairly choosing a random participant as the central authority (i.e., the random dictator). Additionally, we must ensure that no one is getting chosen more often by creating multiple identities (i.e., Sybil attack). Remember, blockchain identities are (usually) free and easy to create.

Note that:
The consensus mechanism is often used to refer to the mining puzzles, however, mining is not a part of the consensus mechanism. We will take a deeper look at it in the next slides, using the imaginary central authority example.

[1] If you want to learn more about this, you can take the “Computational Social Choice” lecture from Prof. Brandt.

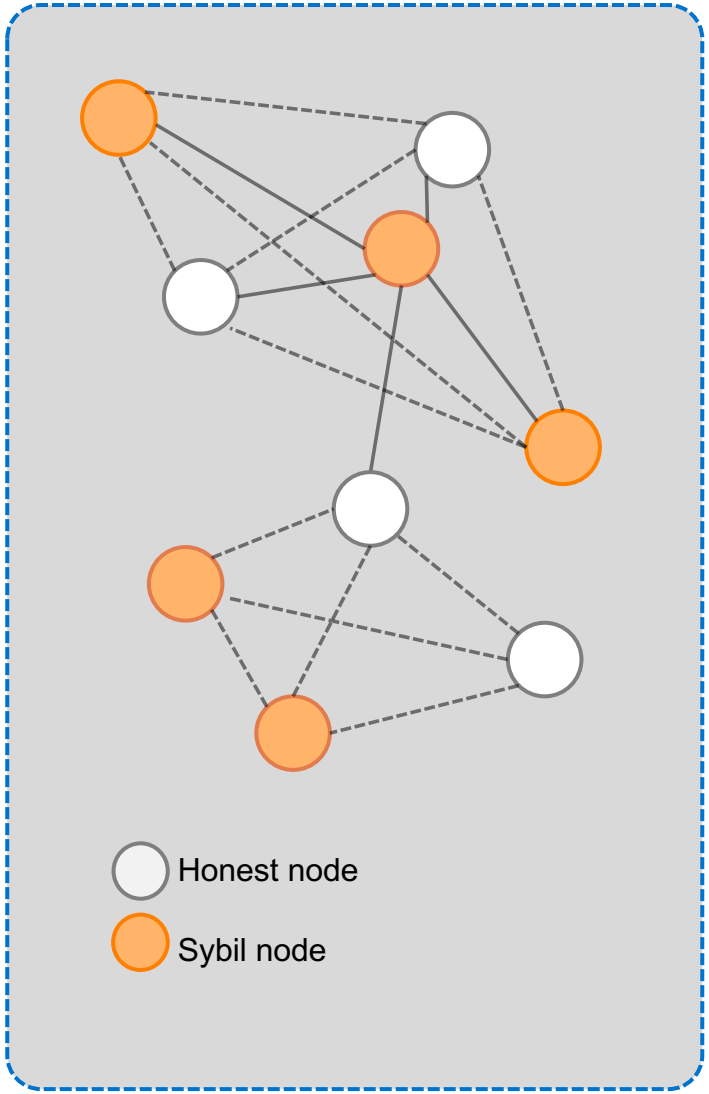
Sybil Control Mechanism

Sybil attack:

- Sybil attack is a type of attack where a user creates and controls multiple identities for a malicious purpose.
- In our case, the attacker can create an arbitrary number of nodes to increase his chance to be selected as the central authority.

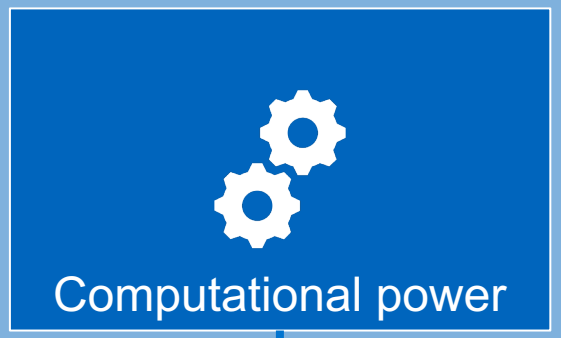
- Bitcoin adopts Proof-of-Work (PoW)[1] as its Sybil control mechanism for validating the expenditure of the computational work.

- To avoid sybil attacks, we need to bind the probability of getting selected to a **scarce resource**. In PoW, that resource is the *computational (hashing) power*. Now, 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]Explained in detail in the following slides.

Sybil Control Mechanism (cont.)



Proof of Work (PoW)

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

Proof of Stake (PoS)

- Coins are deposited to propose new block
- Requires large amount of stake[1]
- Low energy costs
- “Rich people getting richer”
- Low attack costs (discouraged by penalties)

Bitcoin uses PoW!

[1] Note that PoS is not suited to bootstrap a blockchain, as the value of the currency is not bound to another currency.

Difficulties with Reaching a Common World State

- We now have solved the problem of defining who is allowed to create new blocks (i.e., central authority).
- However, we still have to ensure that a common world state (i.e., consensus) is reached. This can be a difficult process as some of the chosen central authorities may not stick to the protocol.
- In game theory, this problem is known as the *Byzantine Generals Problem*.

Byzantine Fault Tolerance

The Byzantine Generals Problem

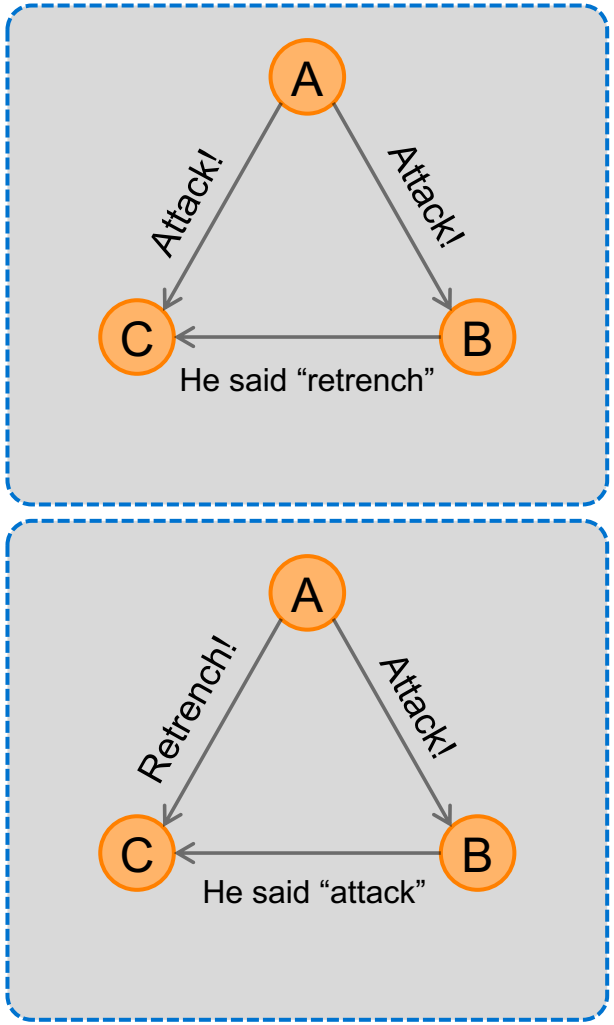
The Byzantine army wants to invade an enemy city. However, it is separated into multiple divisions. They want to attack at the same time, therefore they have to communicate in between the divisions to find a common time to attack.

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 the “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.

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.

Distributed Consensus

- Thanks to cryptography, especially digital signatures, message authenticity is guaranteed. Thus, the only problem left to overcome is agreeing upon a chain to represent the current world state.

Definition *Distributed Consensus*[1]
A network consists of N nodes. Each node has an input value that they propose to other nodes. Some of the 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.**

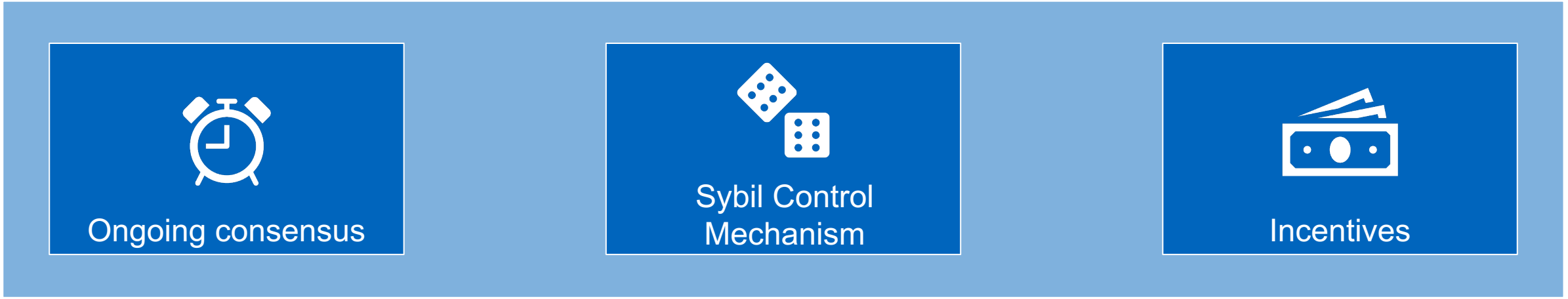
- We want the network to agree on the information the world state contains:
  - Which of the proposed **transactions** are **valid**?
  - In which **order** do the **transactions** appear in the ledger?

- Bitcoin agrees on the **longest chain of blocks** as the world state.

[1] 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.

Bitcoin Invented a New Approach

- Bitcoin's approach to decentralized 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.



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:
The network selects a random node to propose a new block using Proof-of-Work. As we will see later, this ensures that probabilistic consensus can be reached assuming over 50% are honest.

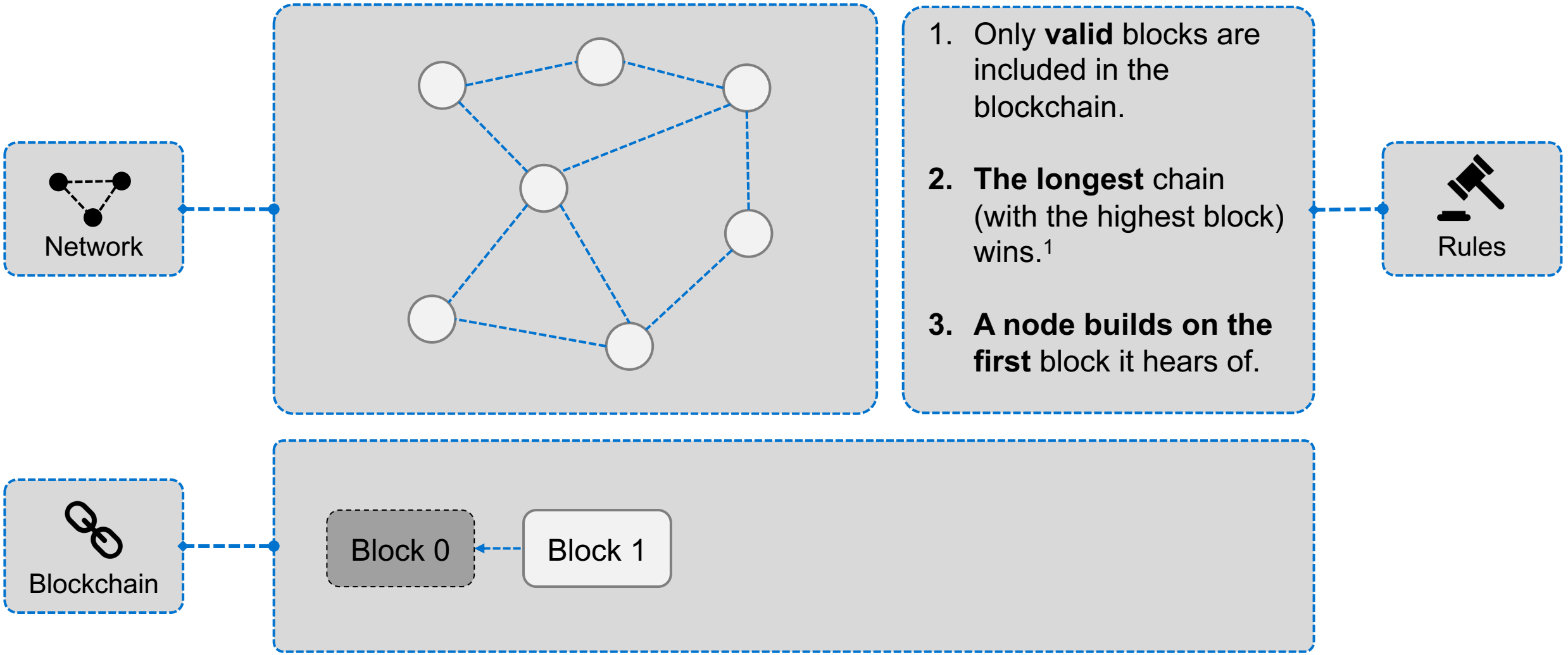
Incentivized nodes:
The network incentivizes nodes to participate in the consensus algorithm. They receive Bitcoins for created blocks which are included in the longest chain.

Simplified Consensus of Bitcoin

1. *Transaction Broadcast*: Every node who receives transactions or creates them, broadcasts them to the network, making everyone aware of new transactions.
2. *Block Building*: Every node collects the valid transactions, orders them and creates a new block containing the transactions.
3. *Random Node Selection*: A node is randomly chosen out of the network. It is able to propose its block to the network.
4. *Block Validation*: Other nodes receive the block from the randomly chosen node and validate whether it is correct. A correct block only contains valid transactions.
5. *Block Acceptance*: Other nodes show their acceptance for this block if the nodes build new blocks on top of the recently proposed block.

Block Propagation

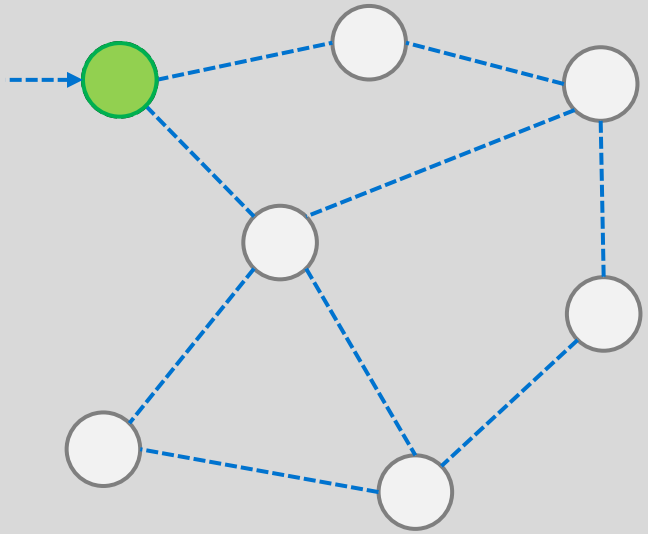
- How do blocks propagate through the network?



[1] The term “longest chain” is actually the chain with the highest weight. We will explain this term later on.

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.

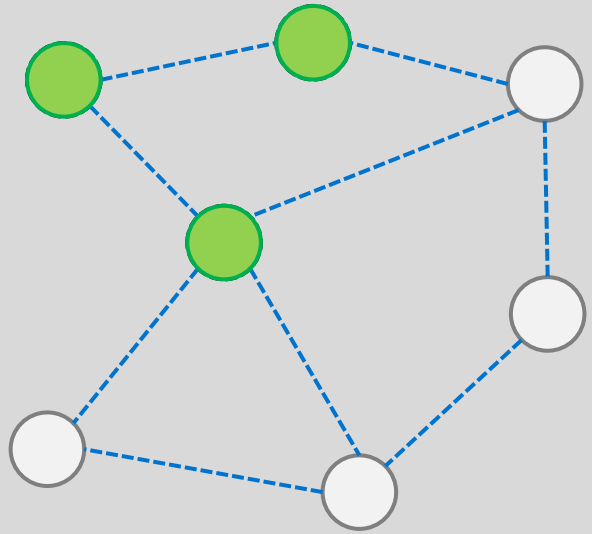
Rules

Blockchain

Block 0 ← Block 1 ← Block 2

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.

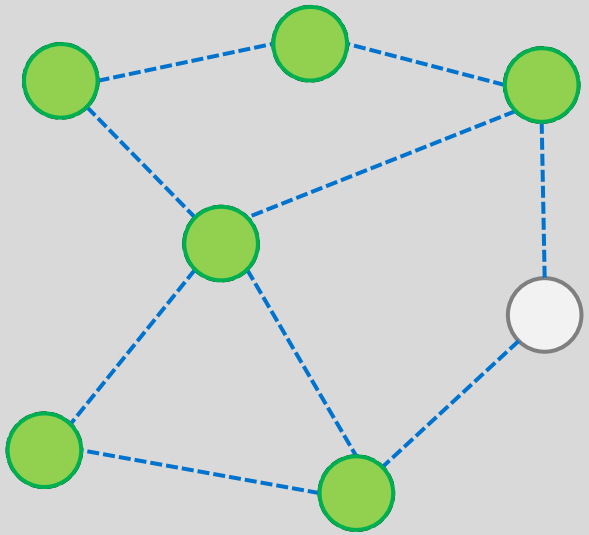
Rules

Blockchain

Block 0 ← Block 1 ← Block 2

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.

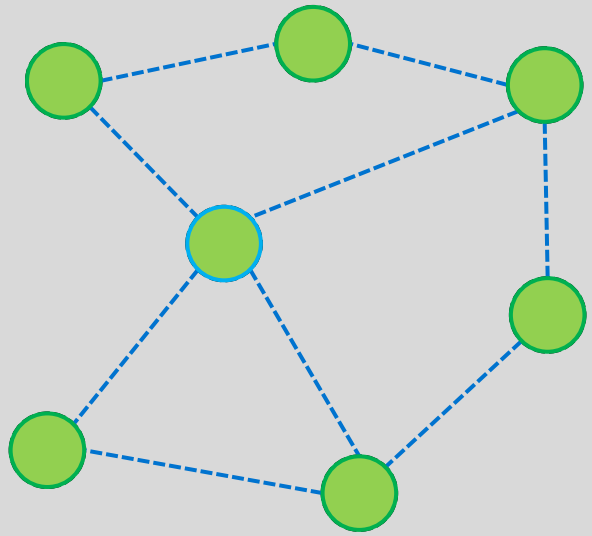
Rules

Blockchain

Block 0 ← Block 1 ← Block 2

Block Propagation

Network



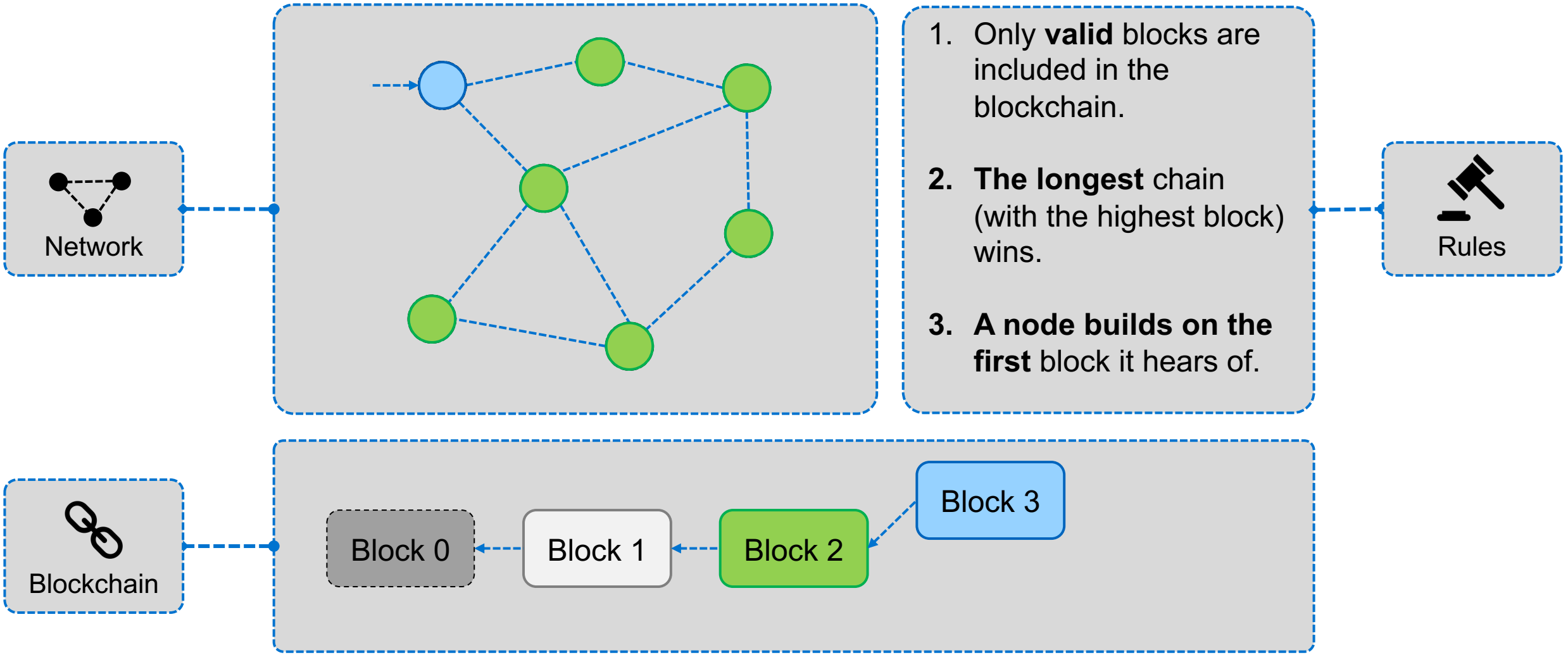
1. Only **valid** blocks are included in the blockchain.
2. **The longest** chain (with the highest block) wins.
3. **A node builds on the first** block it hears of.

Rules

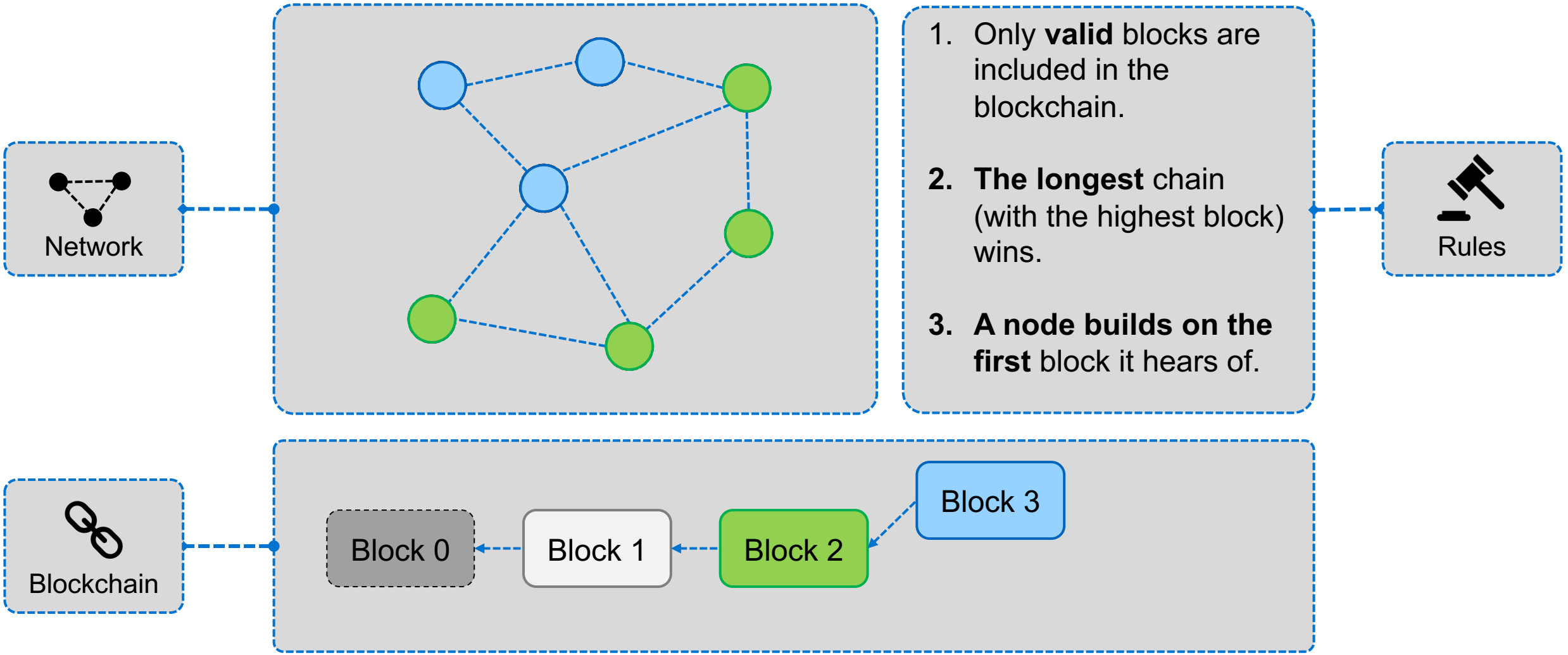
Blockchain

Block 0 ← Block 1 ← Block 2

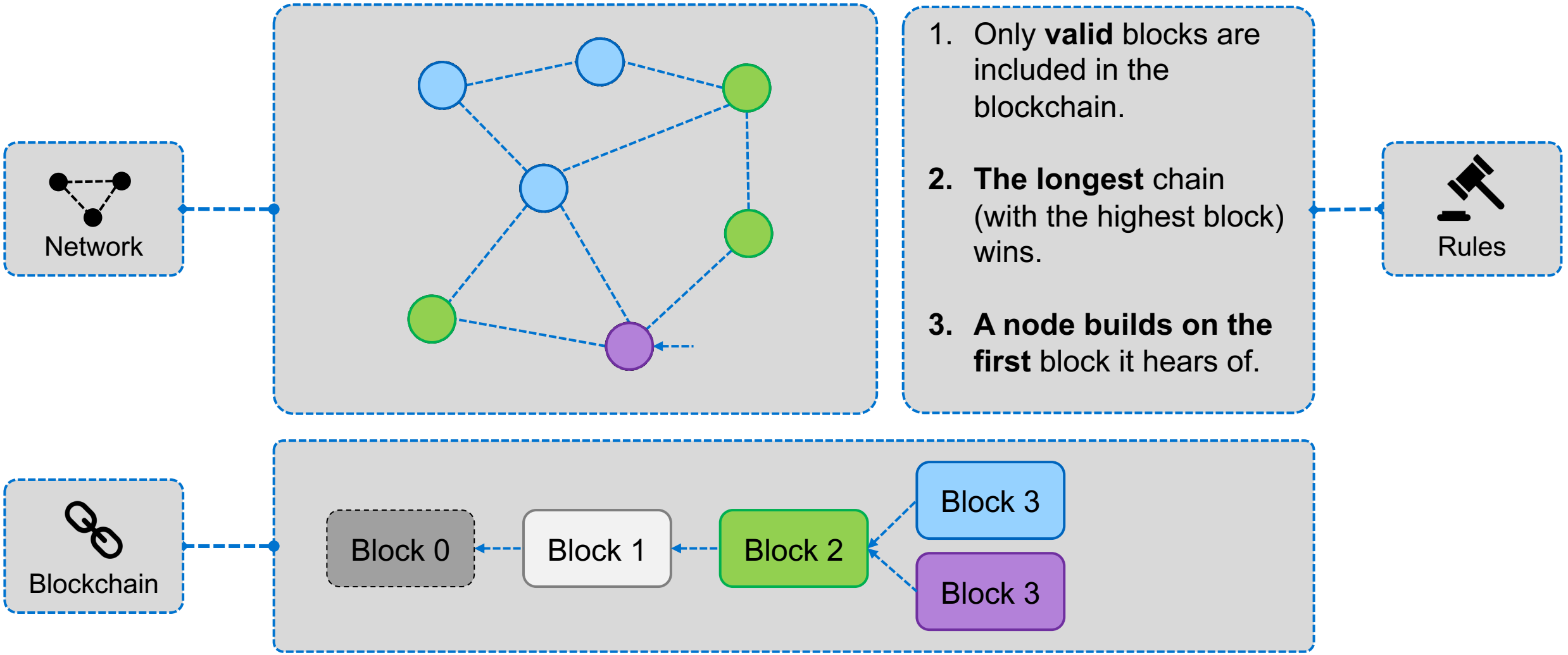
Block Propagation



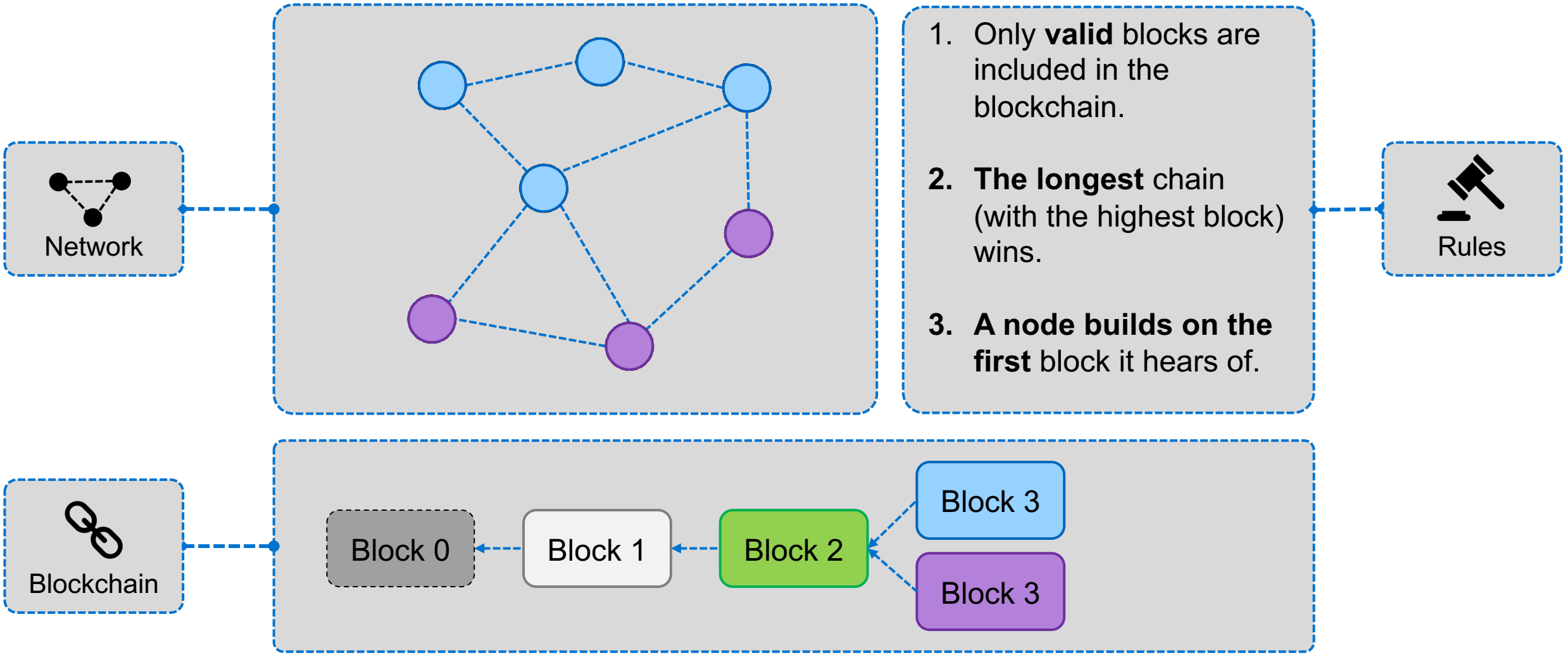
Block Propagation



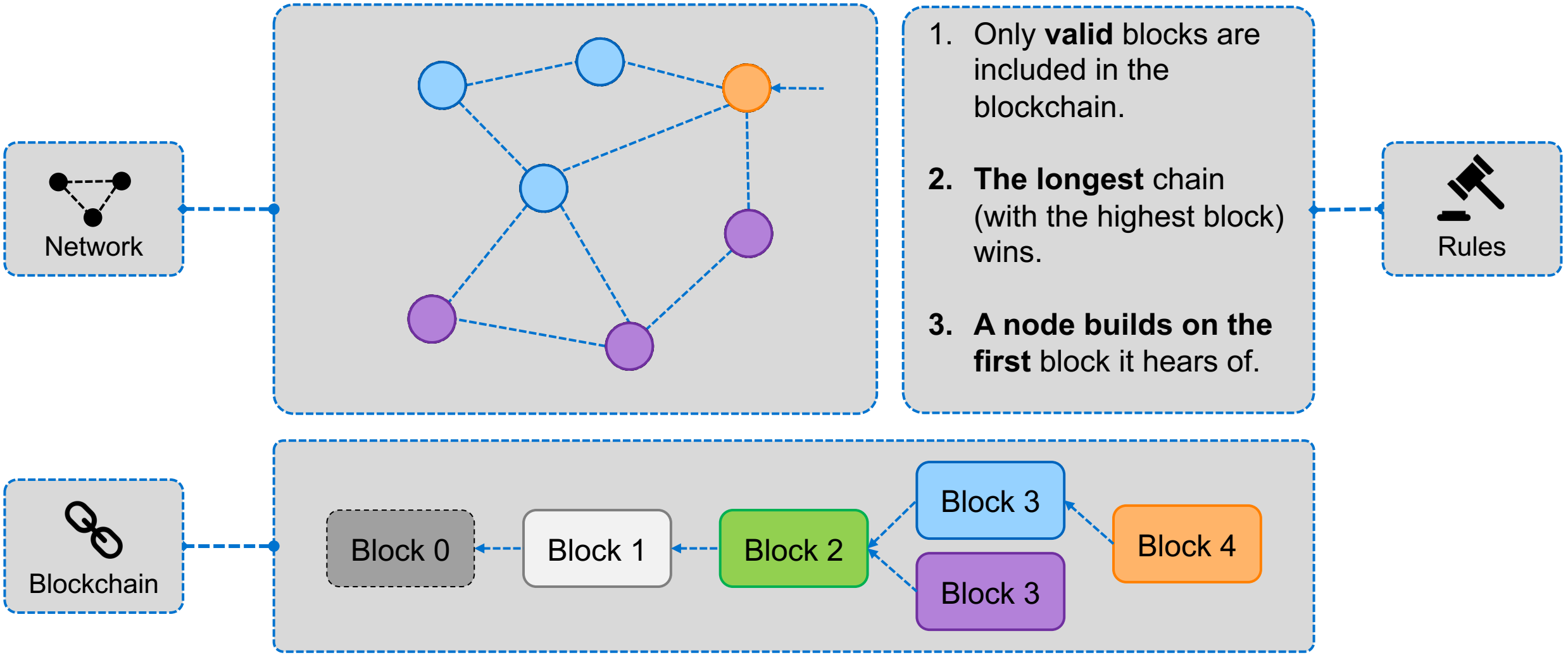
Block Propagation



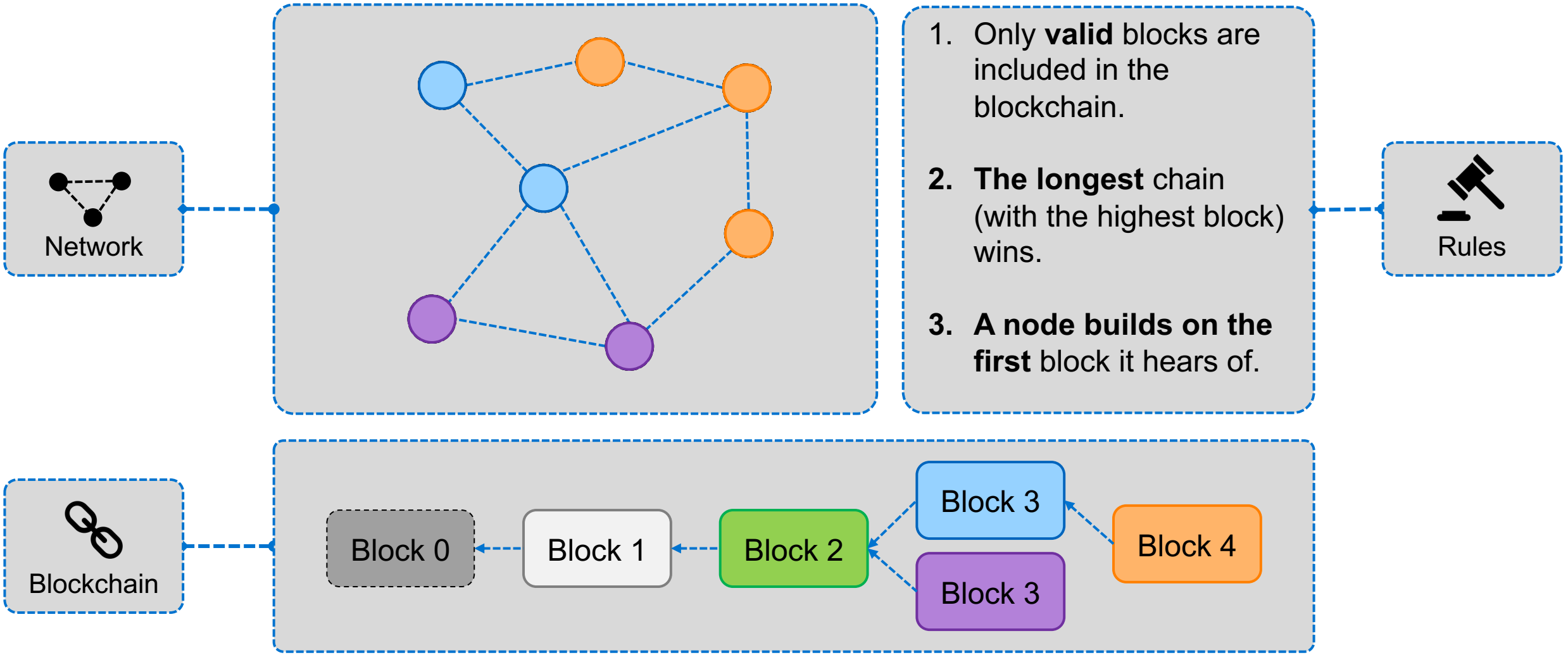
Block Propagation



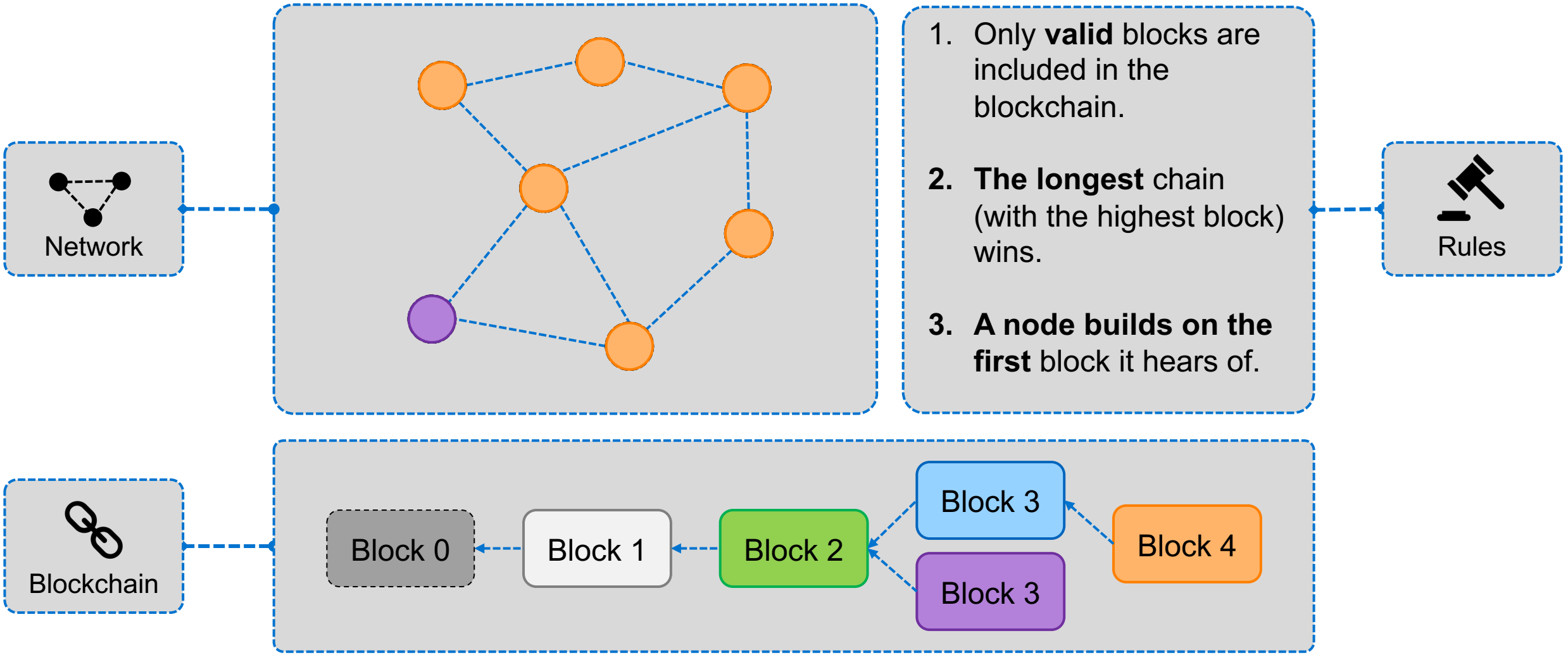
Block Propagation



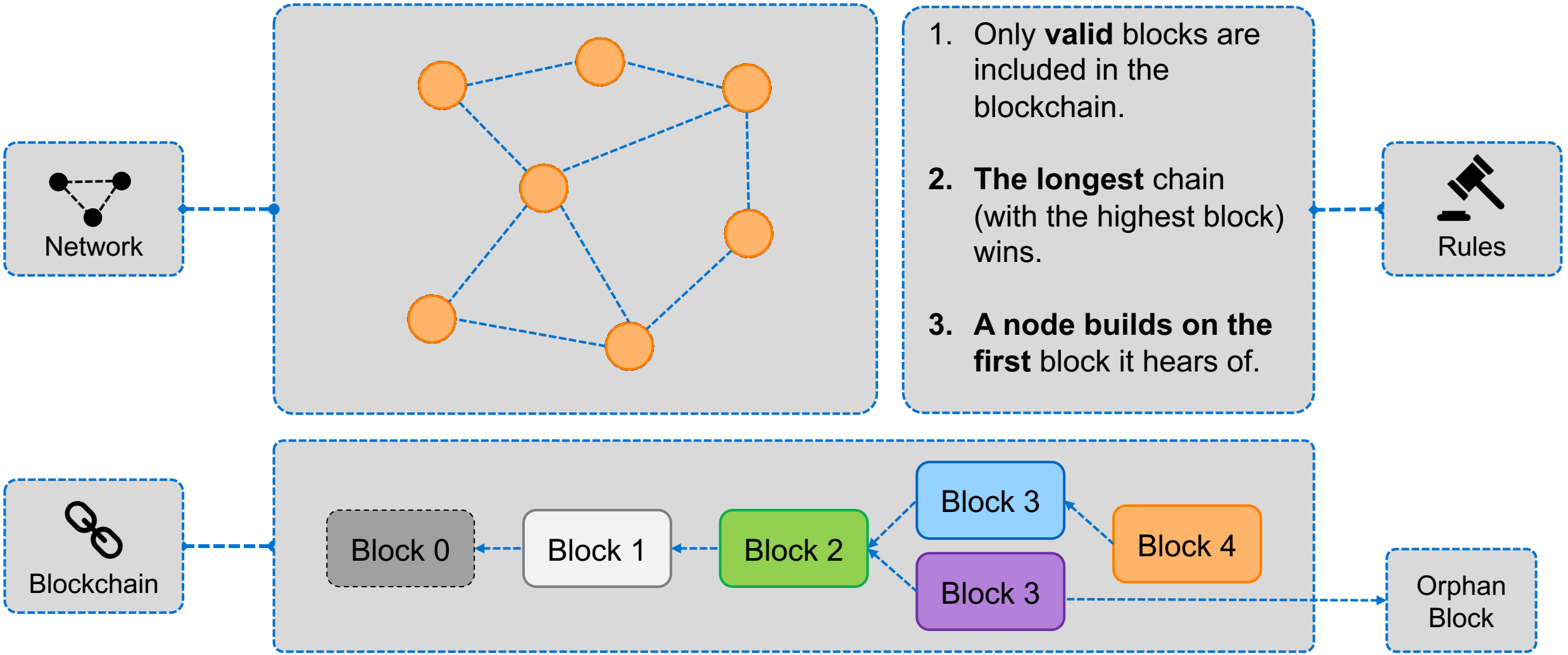
Block Propagation



Block Propagation



Block Propagation



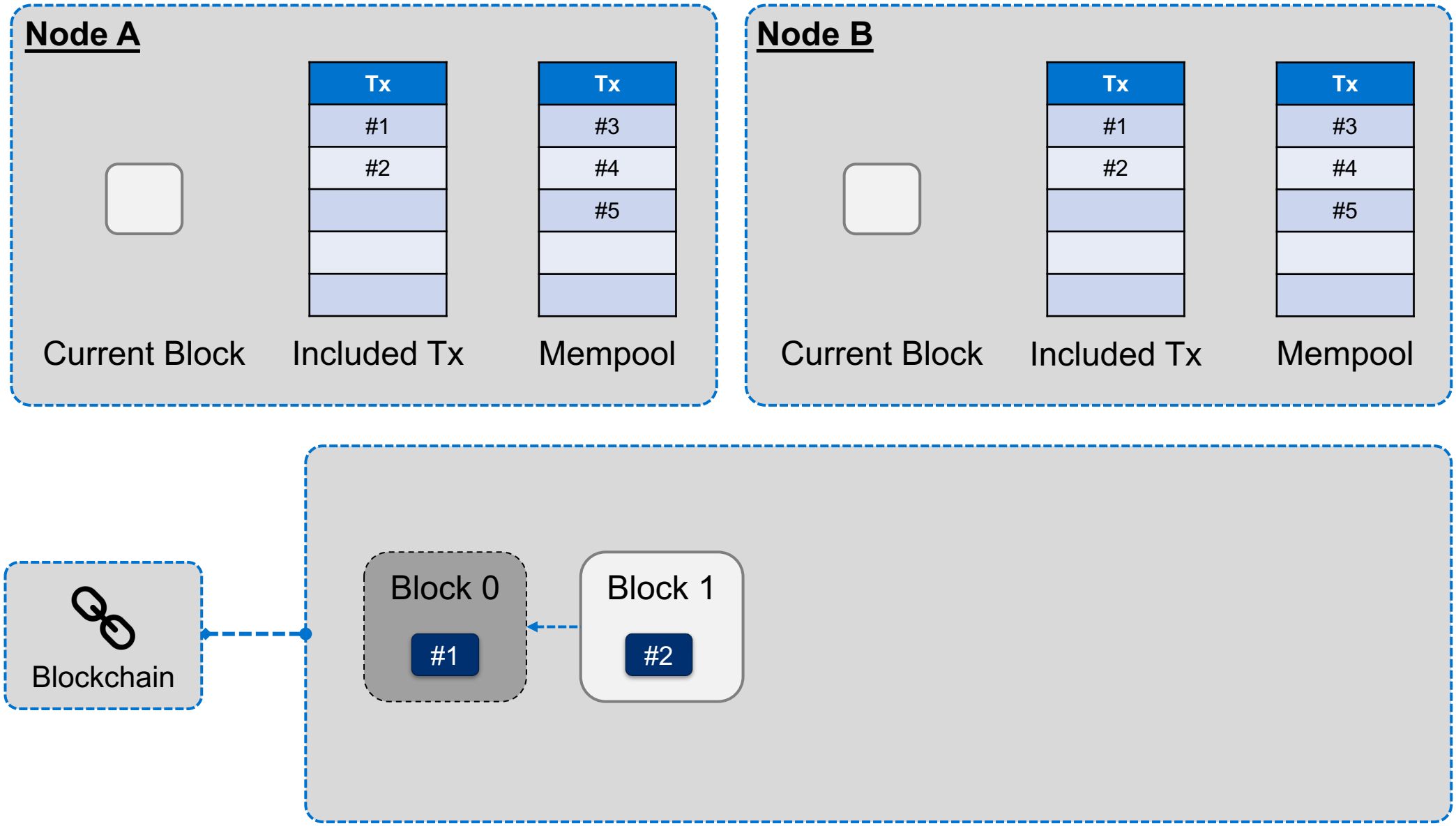
Transactions in Orphan Blocks

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

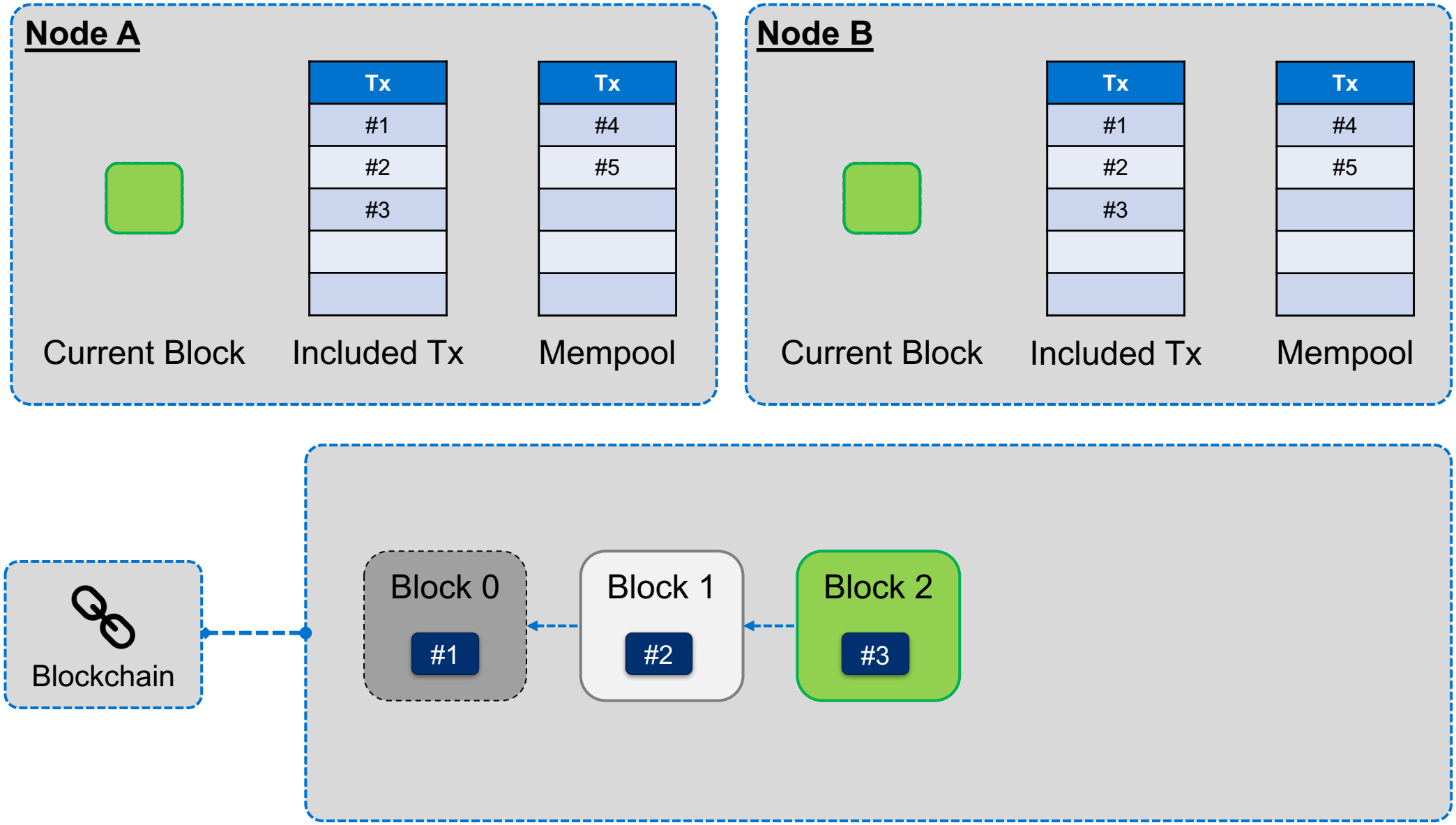
What happens to transactions which are included in the orphan block(s)?

- 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 know of 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.

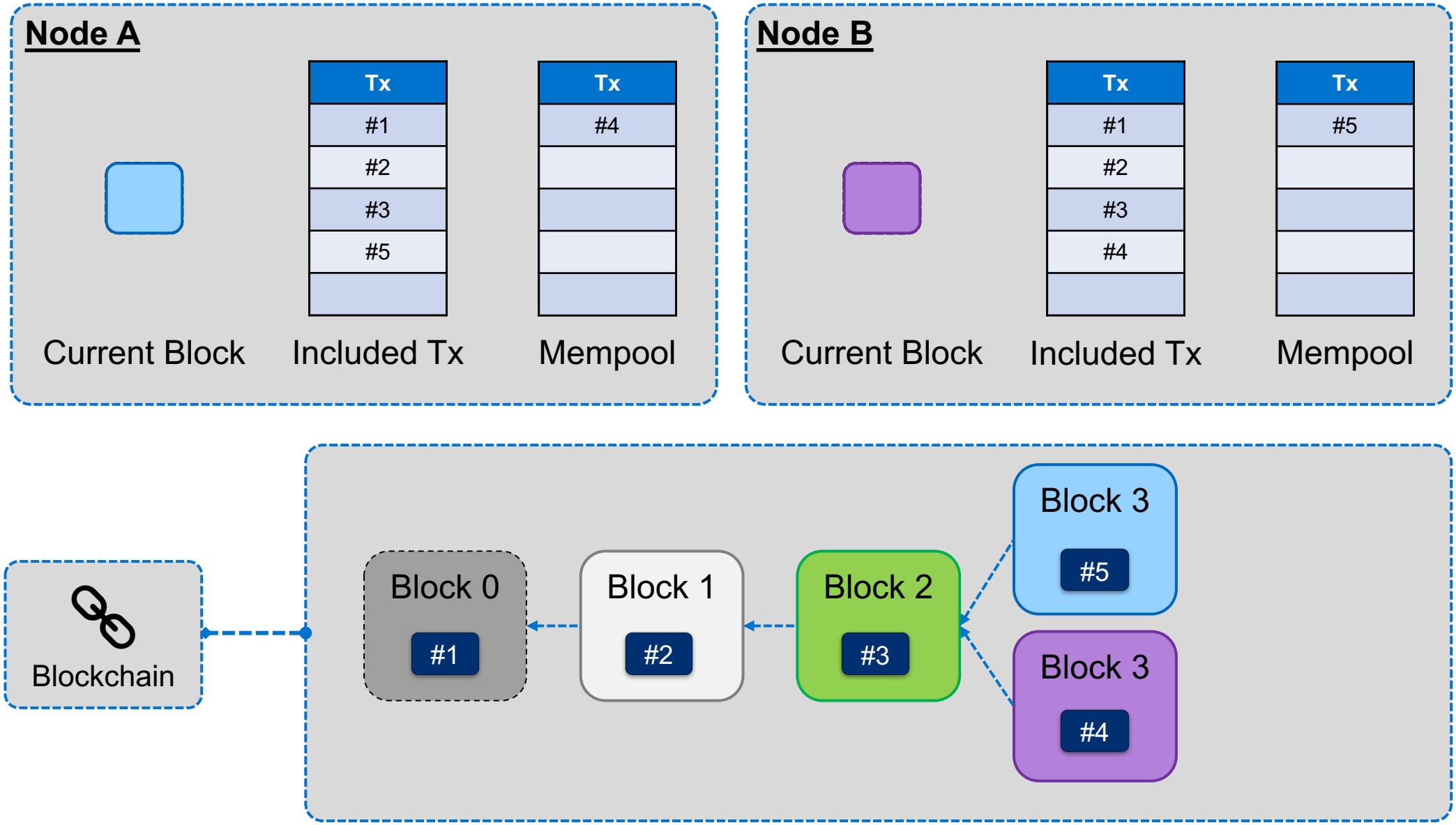
Transactions in Orphan Blocks



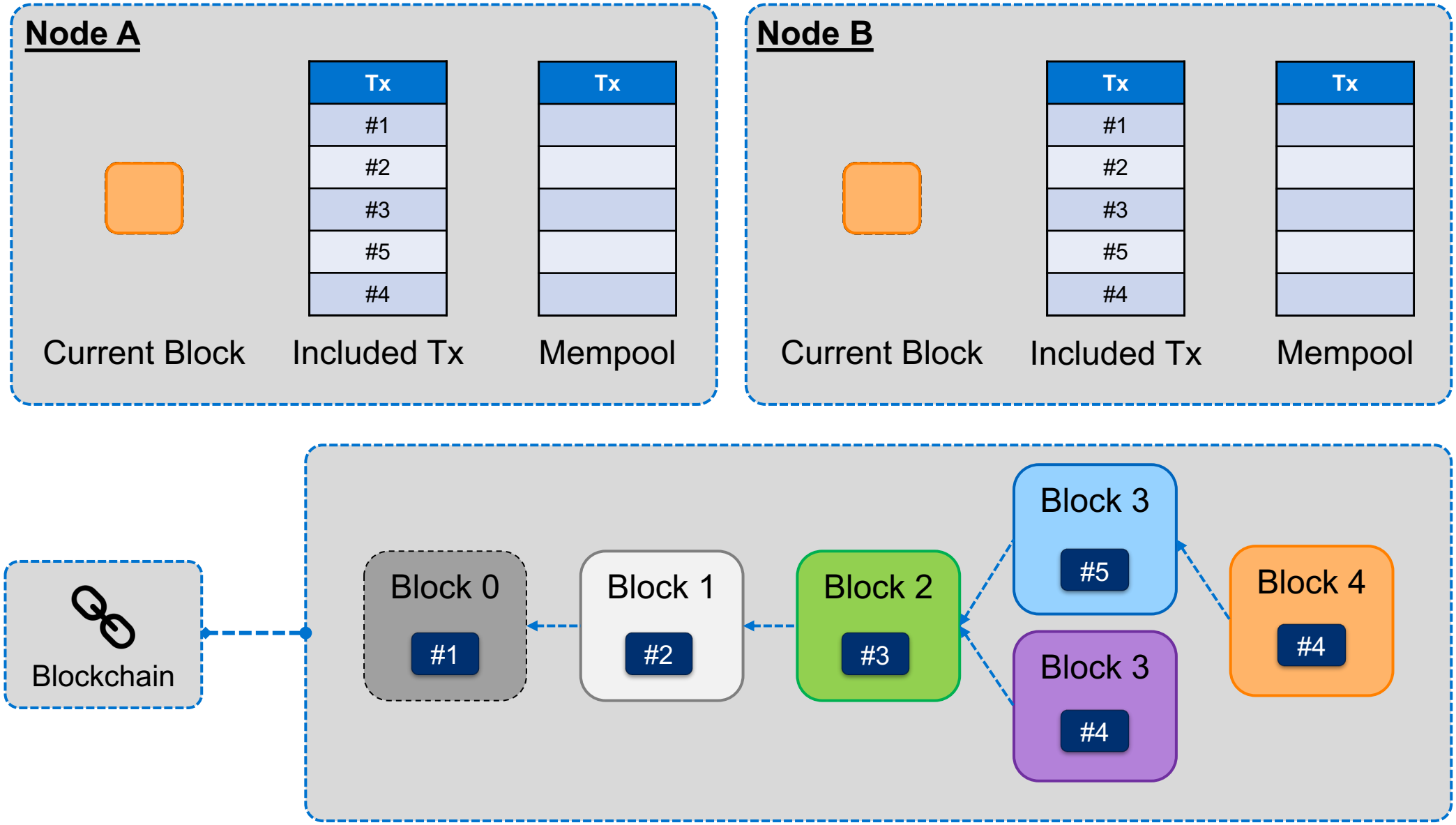
Transactions in Orphan Blocks



Transactions in Orphan Blocks



Transactions in Orphan Blocks



Outline

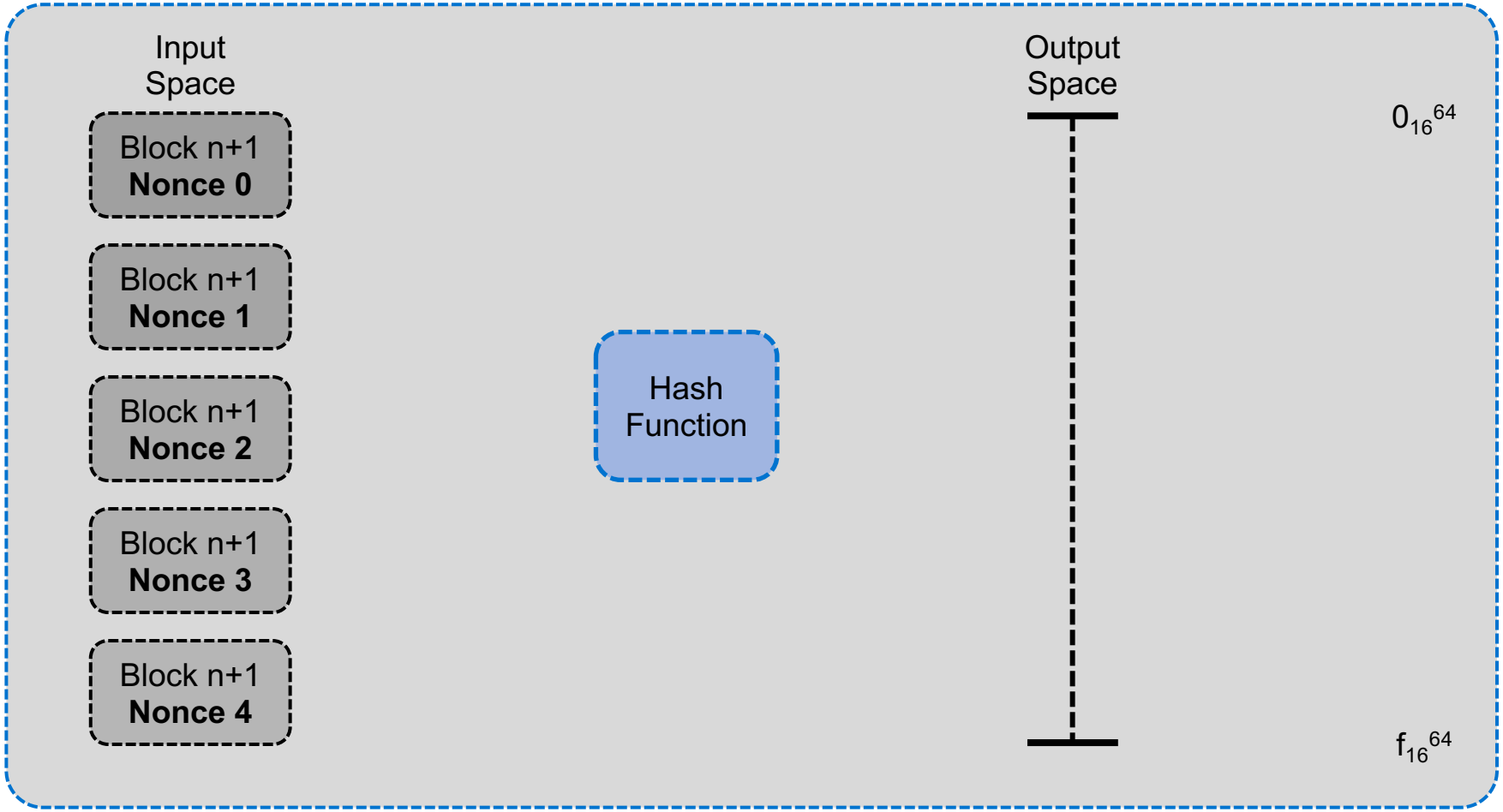
1. Consensus
- Imagine - Bitcoin with a Central Authority
- Random Dictatorship
- Sybil Control Mechanism
- Byzantine Fault Tolerance
- Block Propagation
- Transactions in Orphan Blocks

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

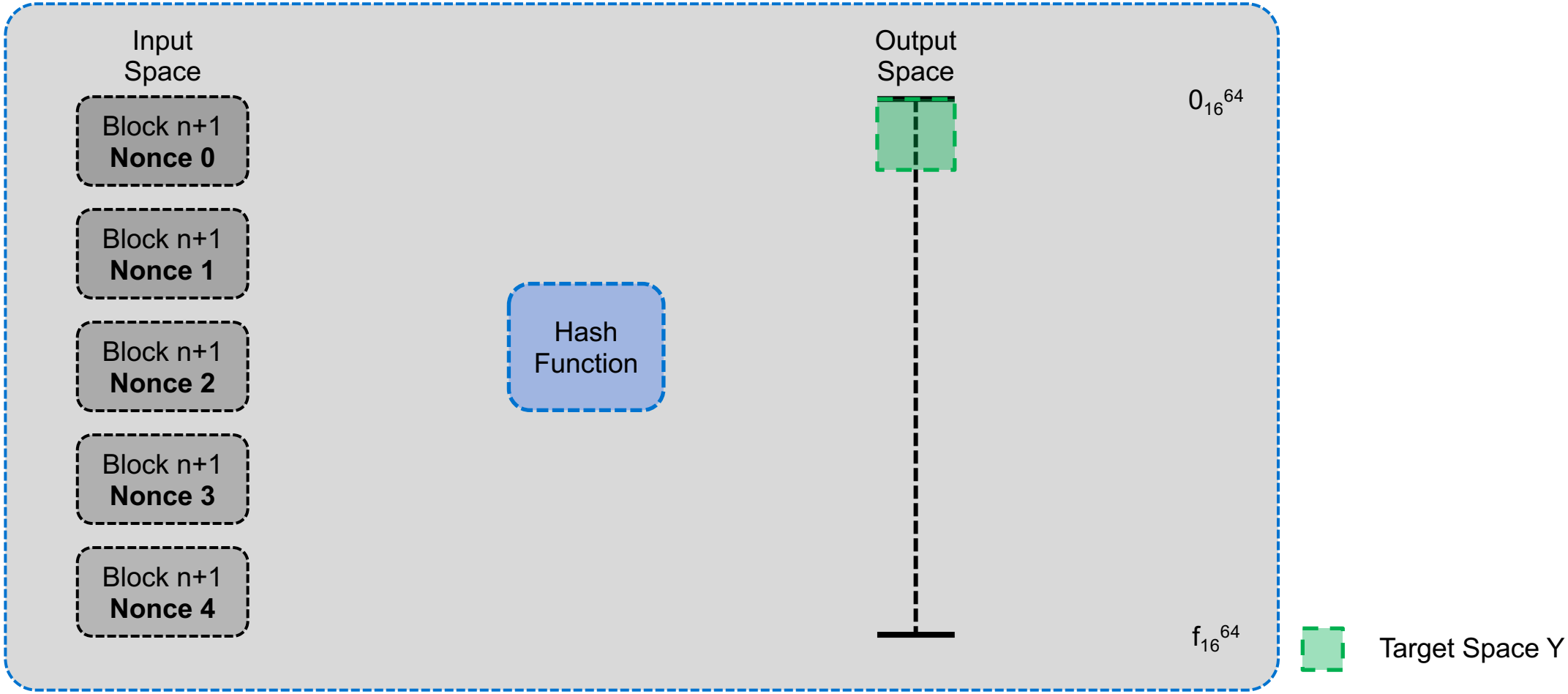
The Mining Puzzle – Proof of Work (PoW)

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)))



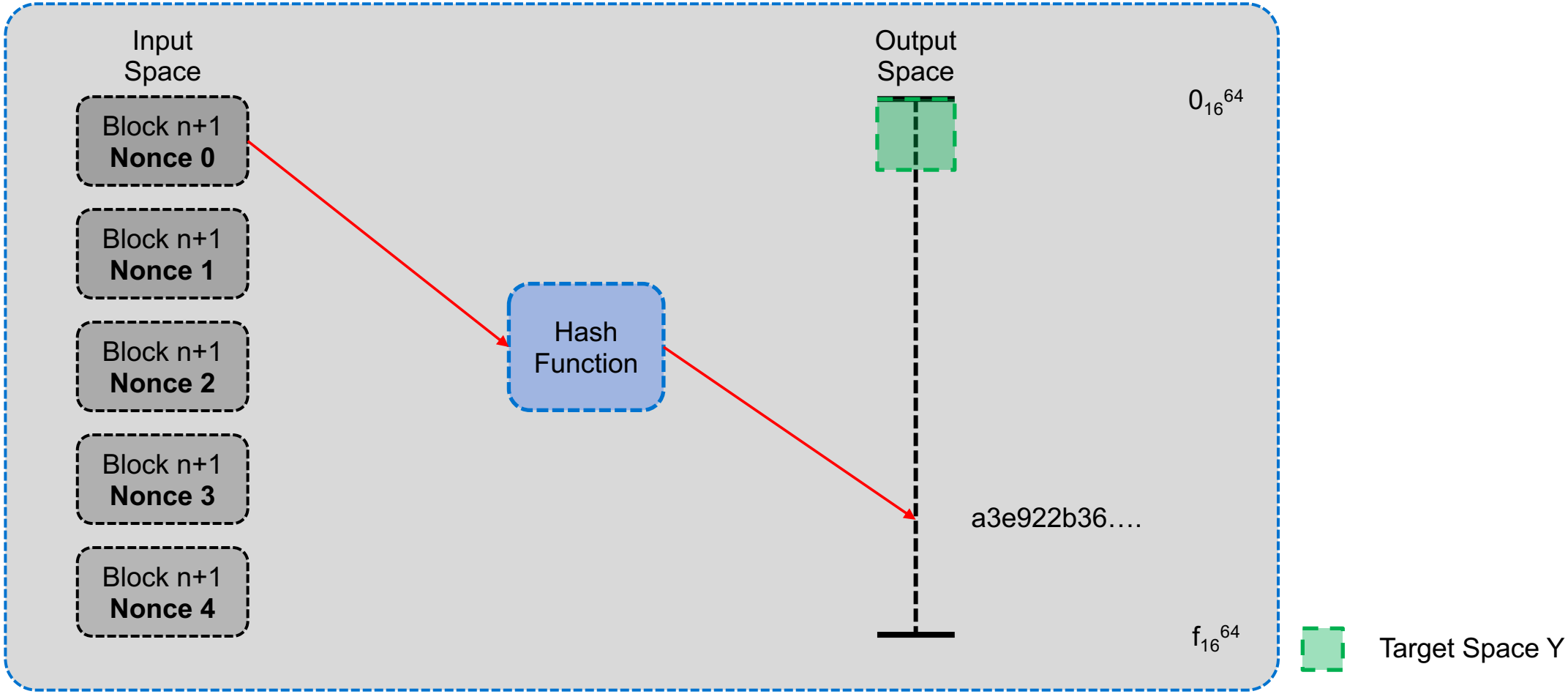
The Mining Puzzle – Proof of Work (PoW)

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)))



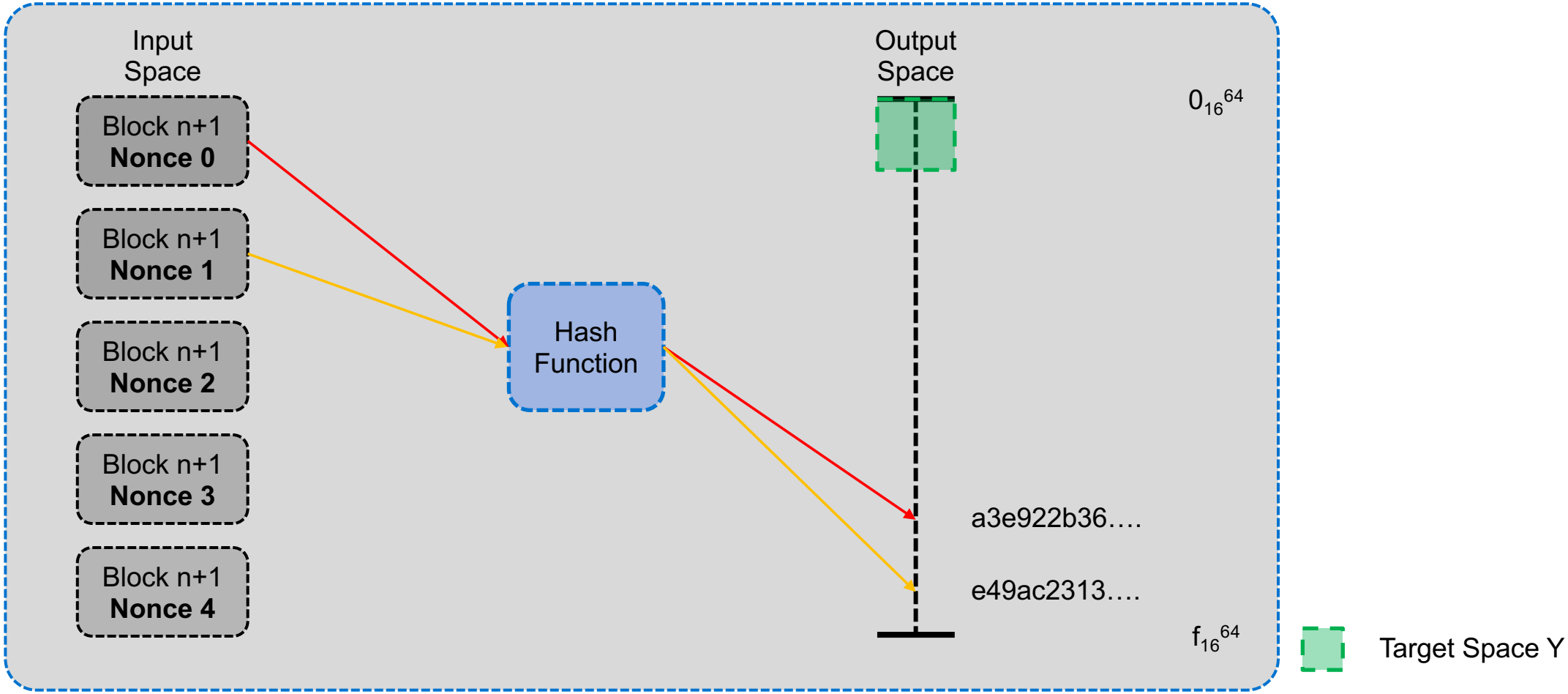
The Mining Puzzle – Proof of Work (PoW)

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)))



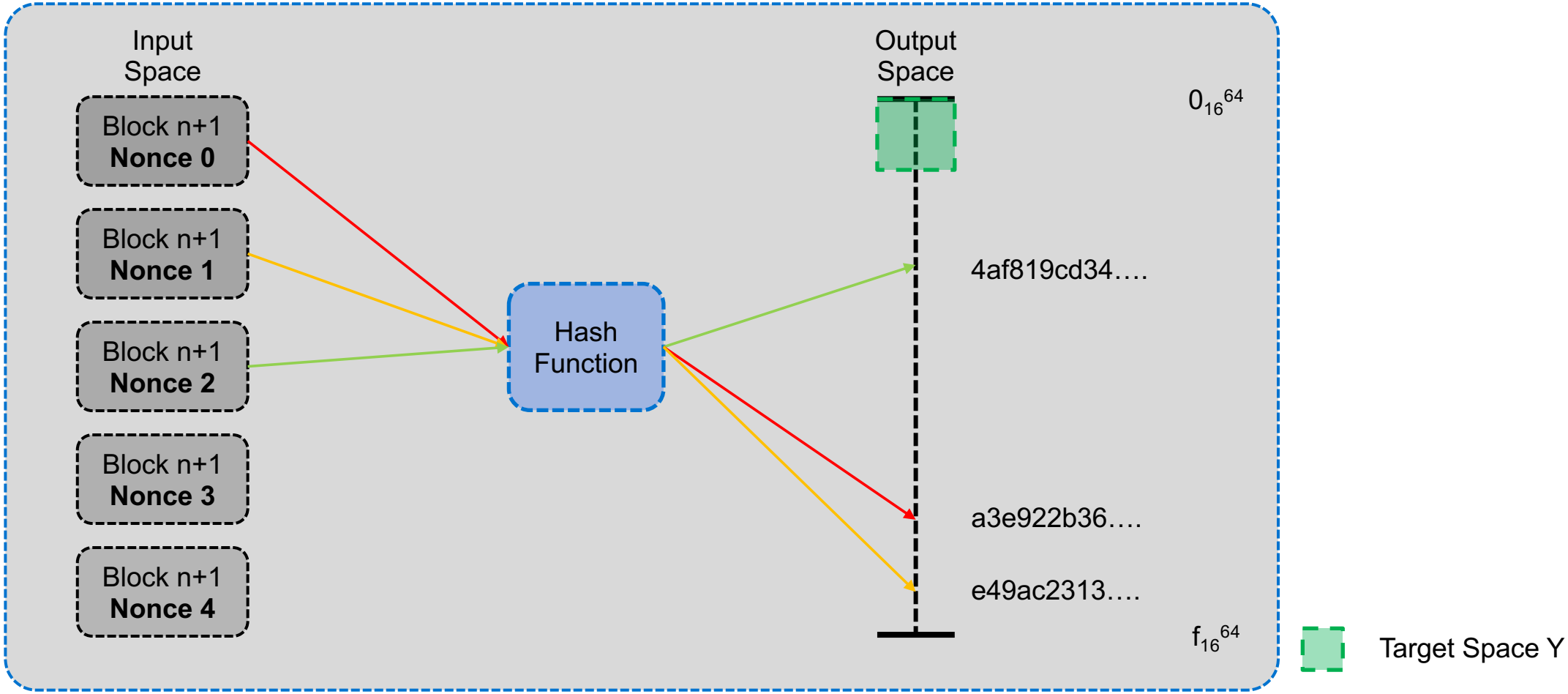
The Mining Puzzle – Proof of Work (PoW)

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)))



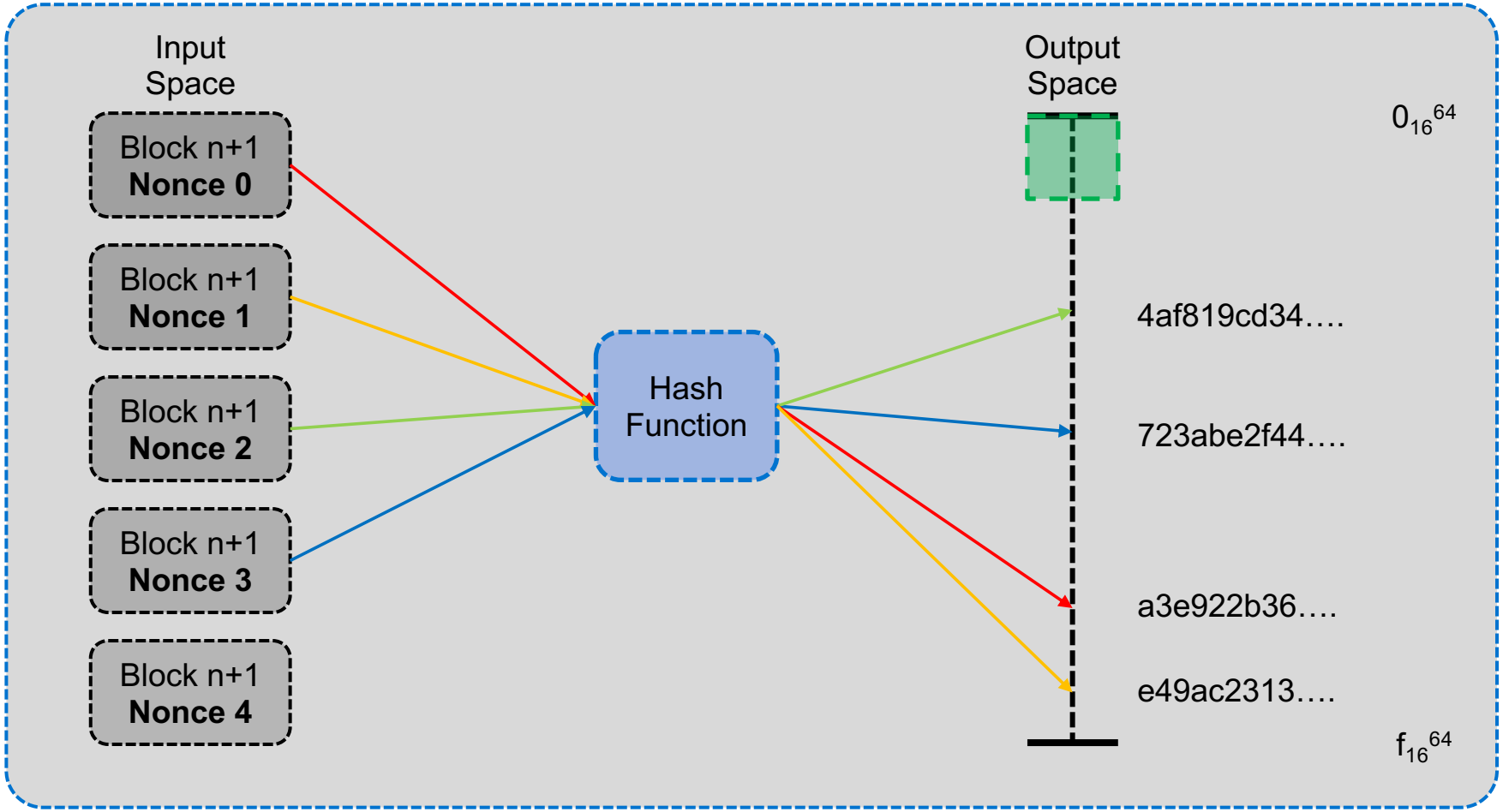
The Mining Puzzle – Proof of Work (PoW)

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)))



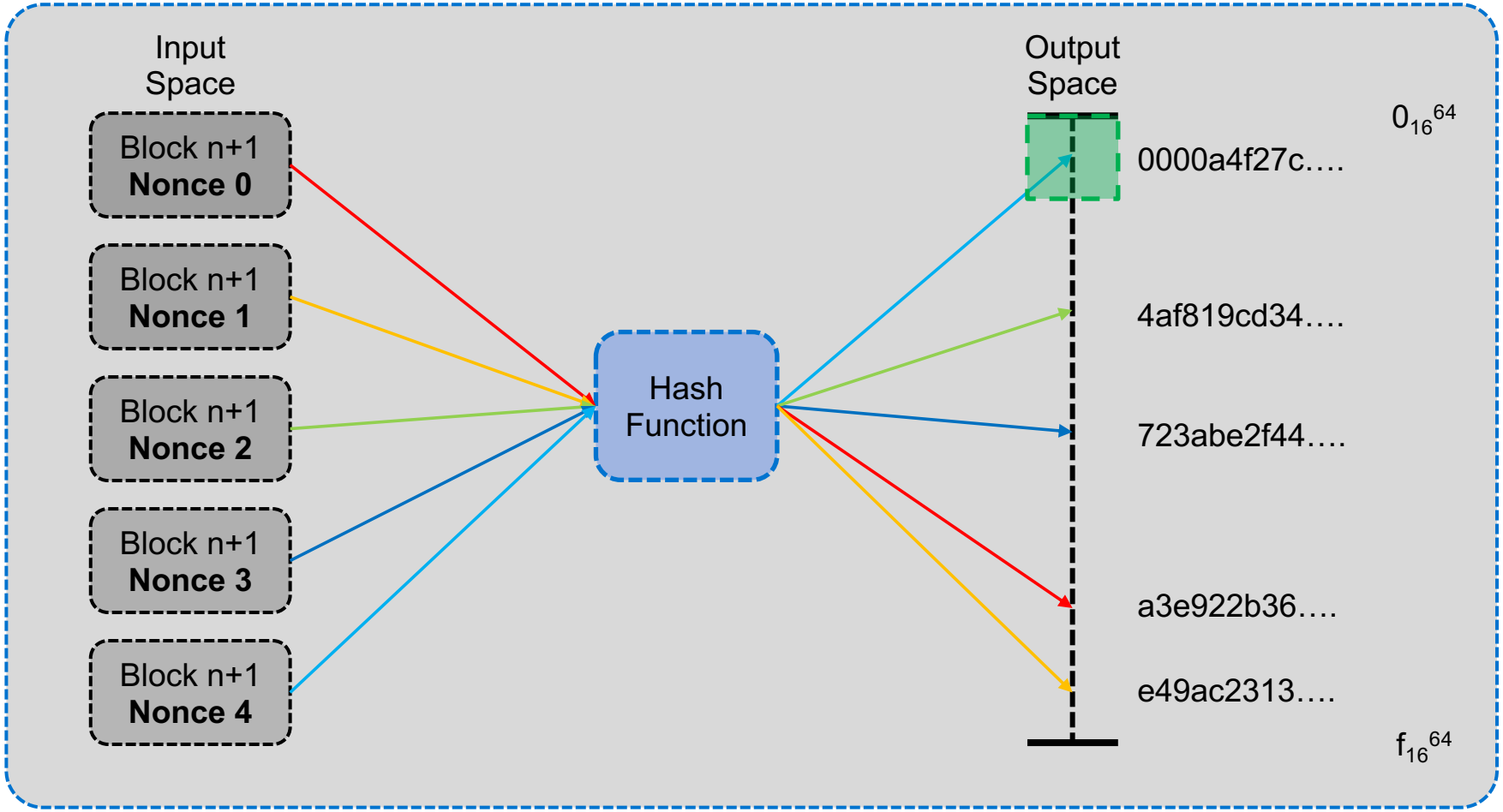
The Mining Puzzle – Proof of Work (PoW)

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)))



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)))



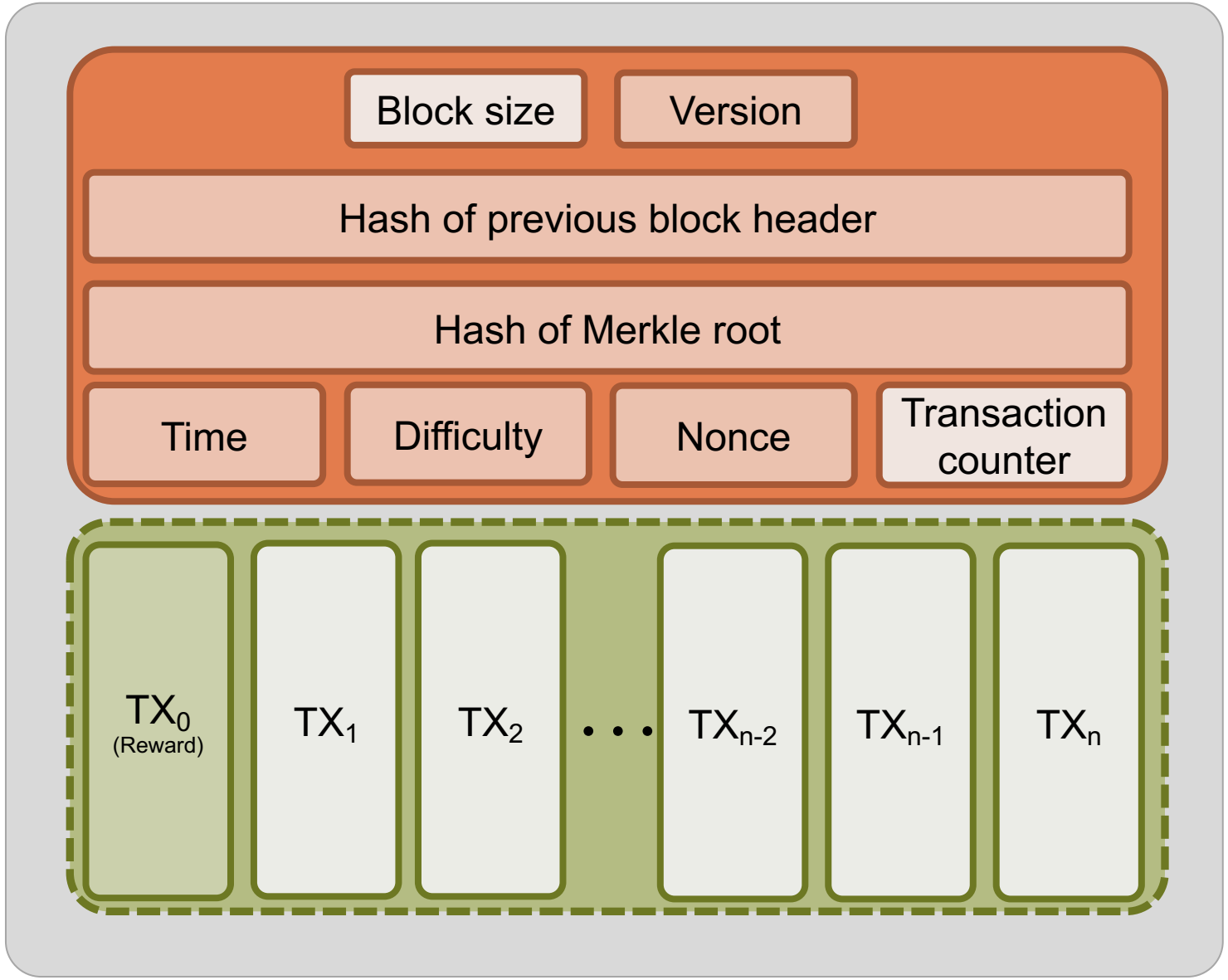
Does Everyone Have the Same Search Puzzle?

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 not?



Does Everyone Have the Same Search Puzzle?

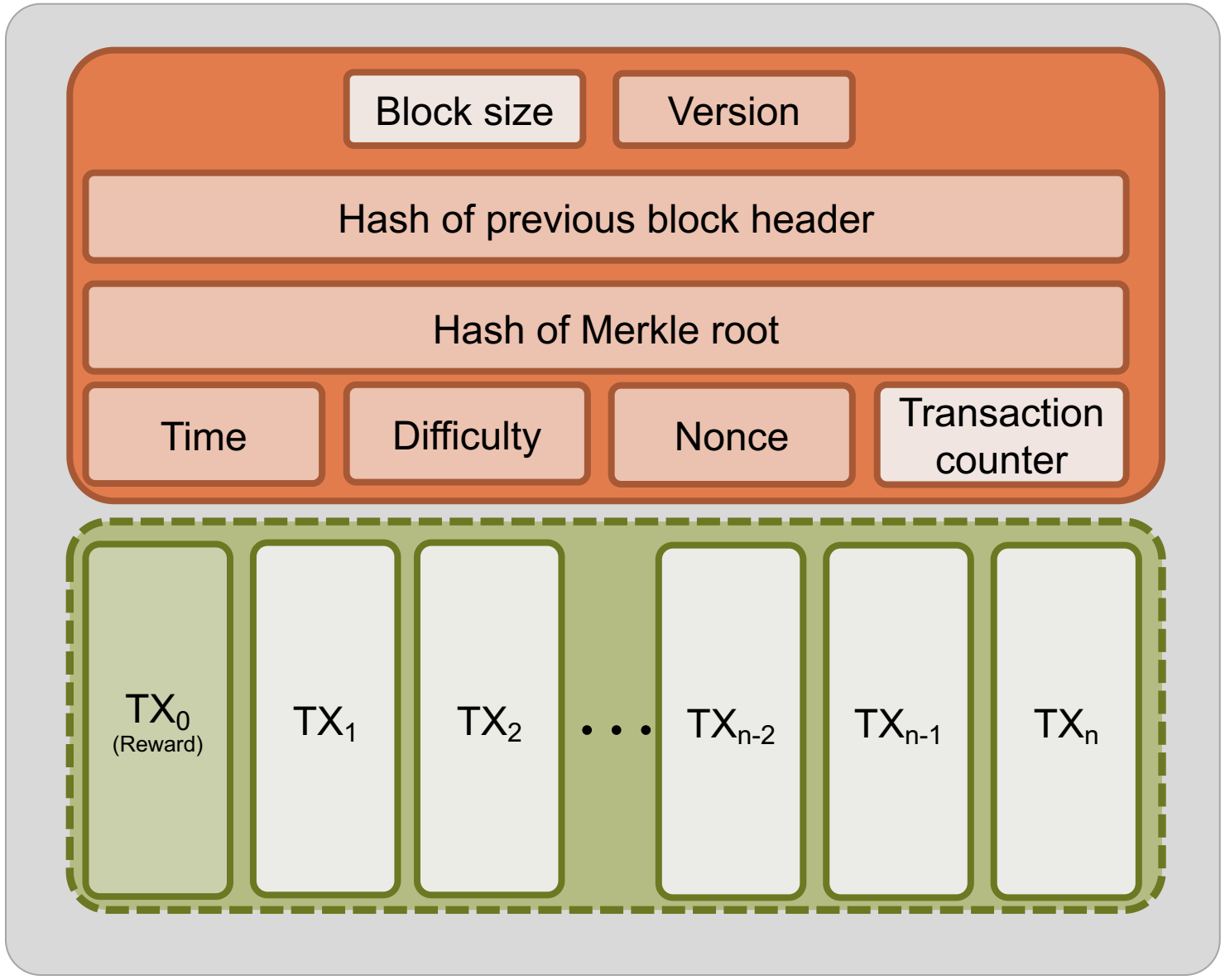
Recap:

- 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 not?

- 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 has the block time 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 accumulated highest 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

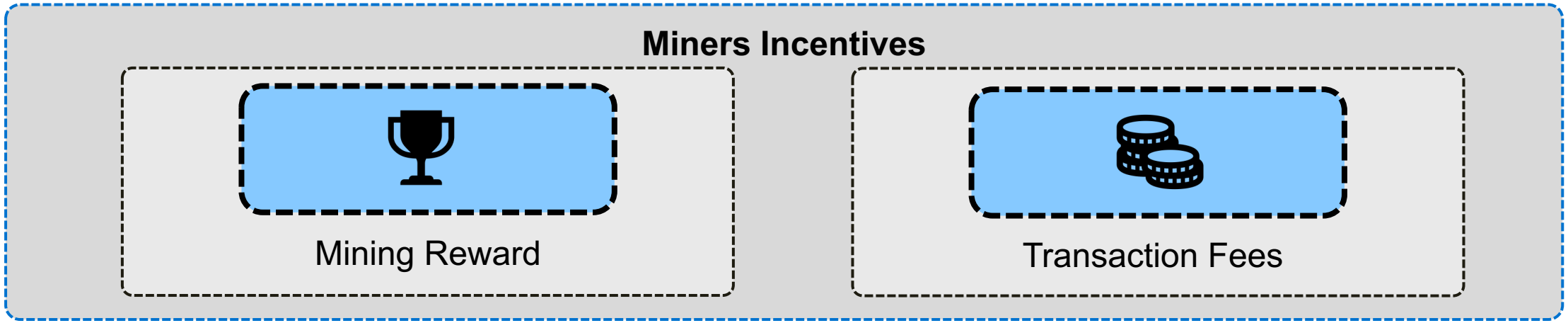
The Bitcoin Currency

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!



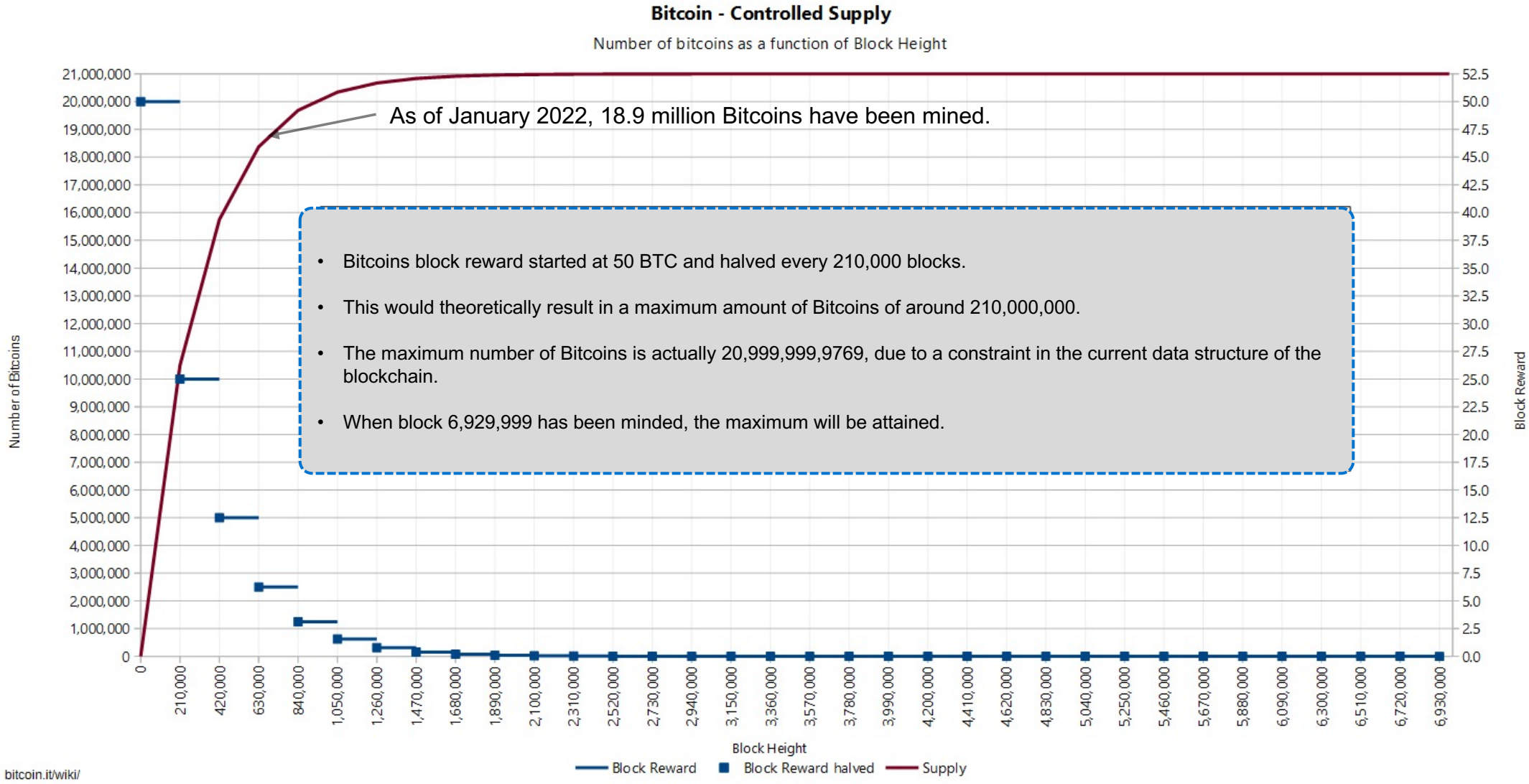
Mining Reward

- 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

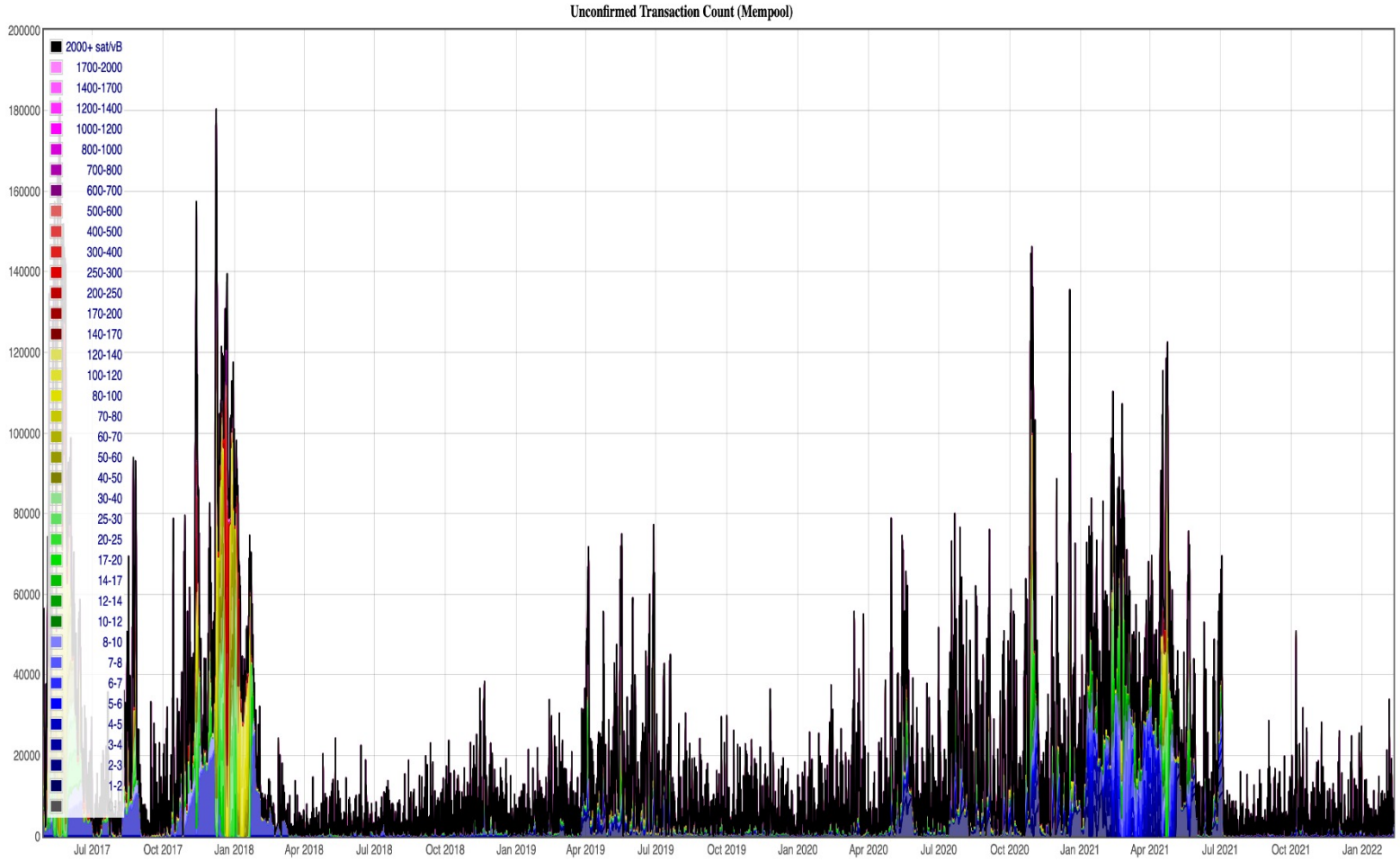


- Bitcoins block reward started at 50 BTC and halved every 210,000 blocks.
- This would theoretically result in a maximum amount of Bitcoins of around 210,000,000.
- The maximum number of Bitcoins is actually 20,999,999,9769, due to a constraint in the current data structure of the blockchain.
- When block 6,929,999 has been minded, the maximum will be attained.

bitcoin.it/wiki/

Transaction Fee

- Every transaction includes a transaction fee. 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 6 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*)

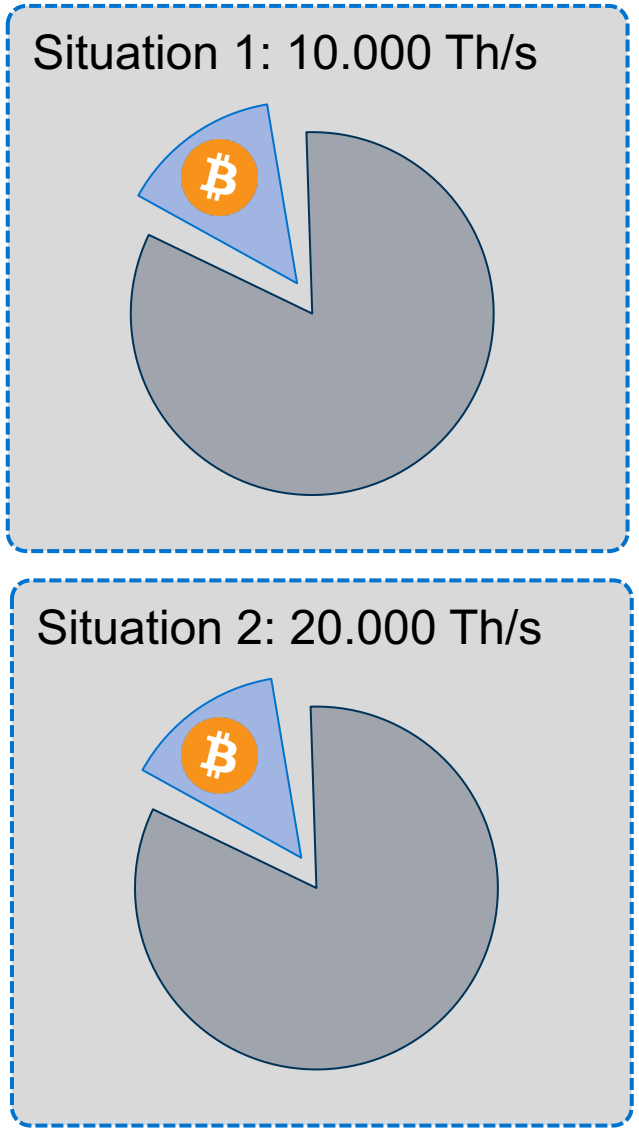
Arms Race in Mining

- The process of mining can be a profitable business. However, there are some remarks:

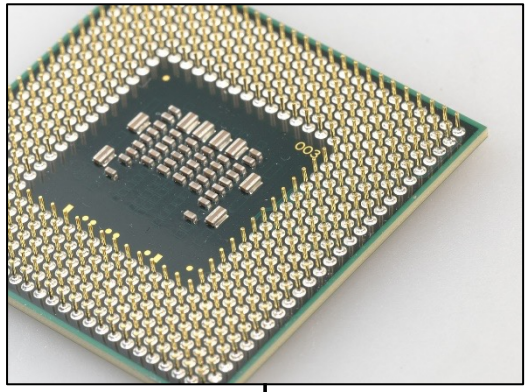
- Approximately 900 new Bitcoins are mined per day. On a daily basis, 144 new blocks are produced. If computing power increases, the difficulty of the search puzzle increases, too.
  ➔ The overall output of the mining reward stays the same.

- 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.**



Mining Hardware

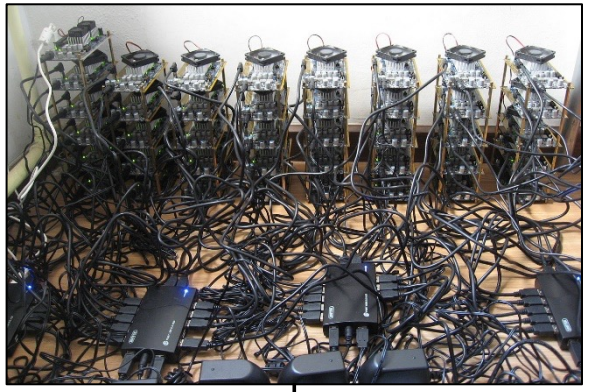


2009
CPU

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

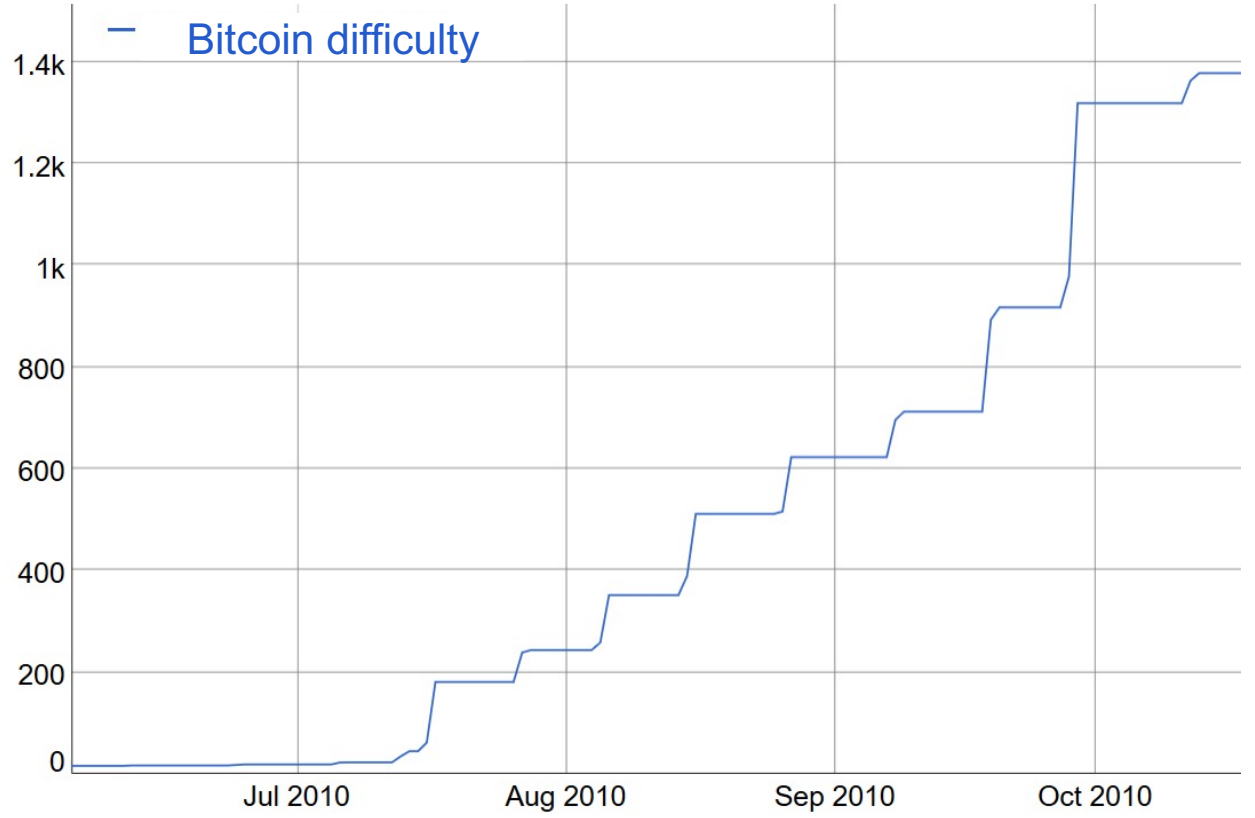
Re: A few suggestions
December 12, 2009, 05:52:44 PM

#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.

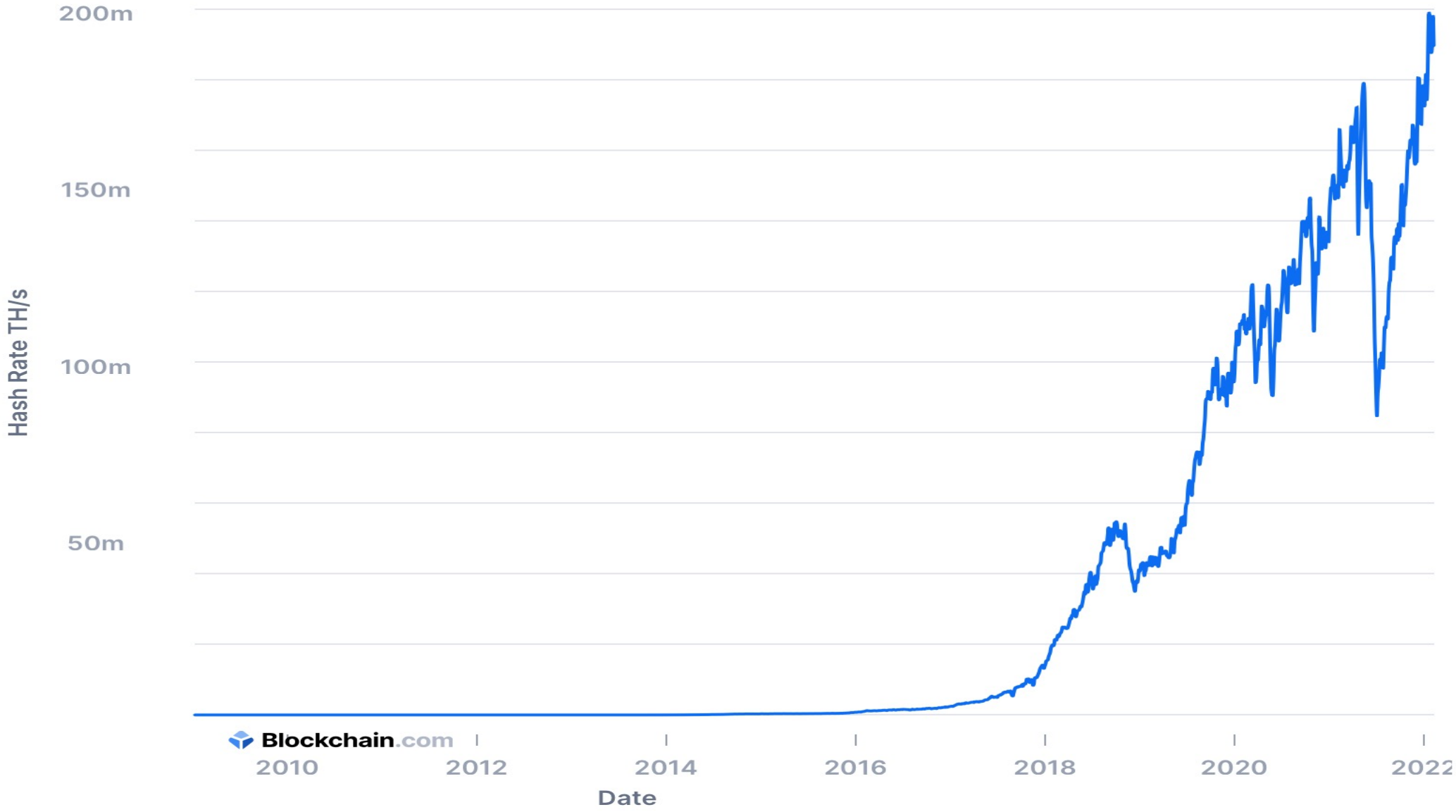


See this post of Satoshi Nakamoto: https://bitcointalk.org/index.php?topic=12.msg54#msg54

A historic chart of Bitcoin difficulty: https://bitinfocharts.com/de/comparison/bitcoin-difficulty.html

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

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[1] = 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*

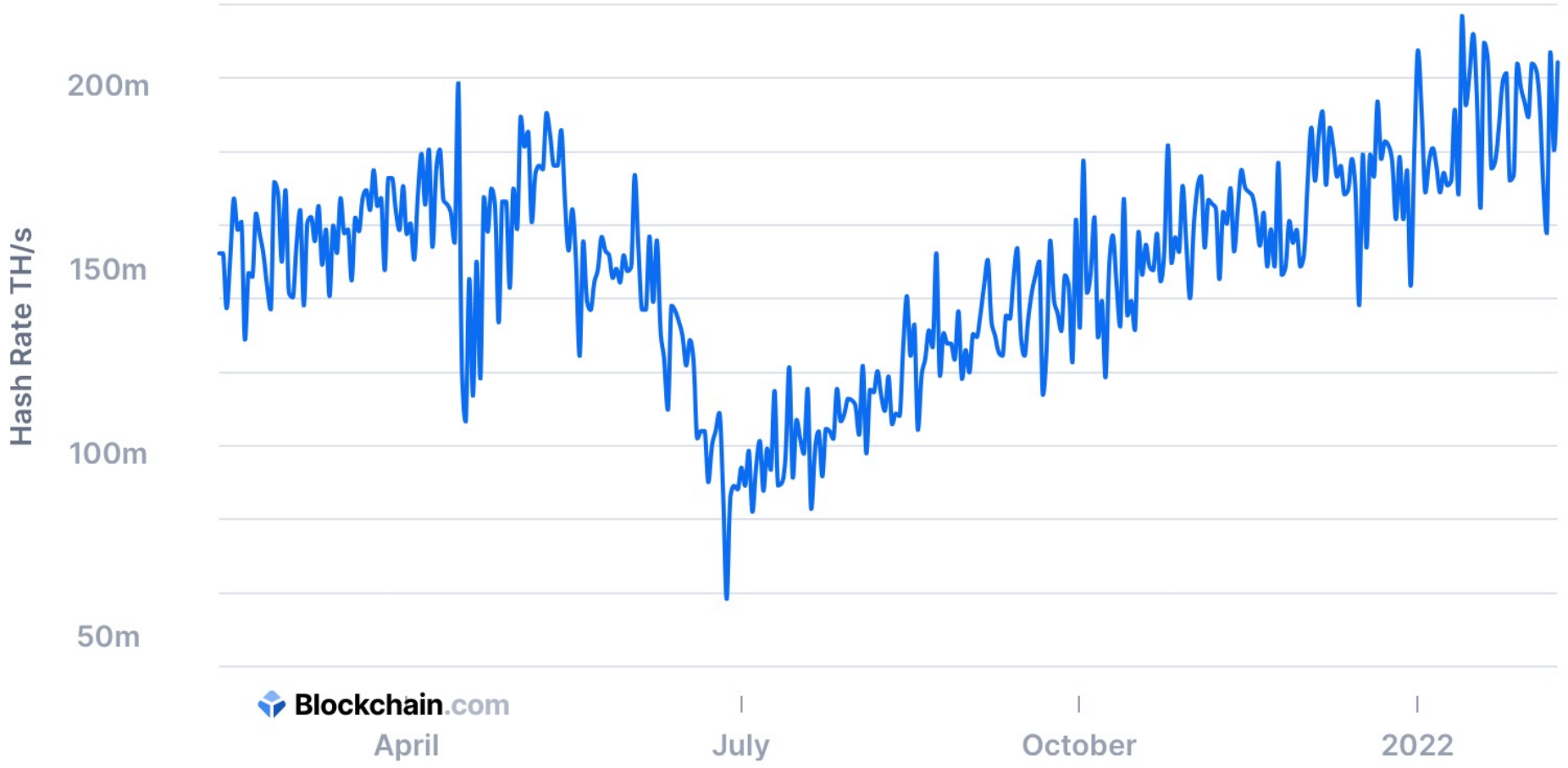
How does a mining pool work?

- 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.

[1]We will see later where this number comes from.

Hash Rate

- The hash rate is a stochastic variable.

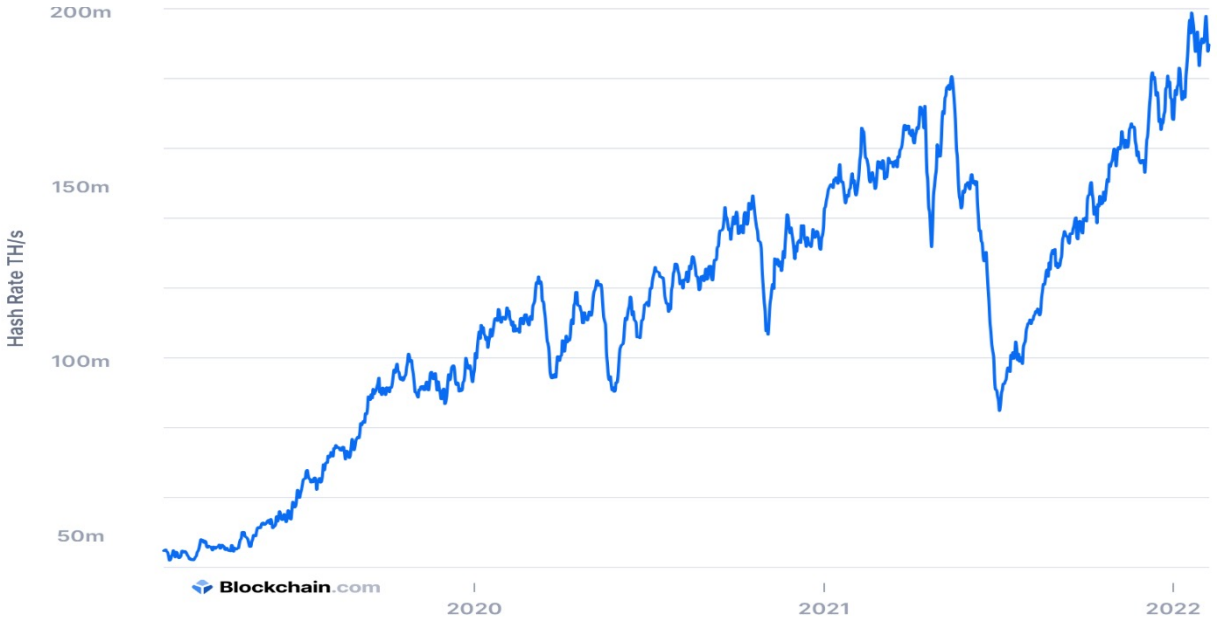


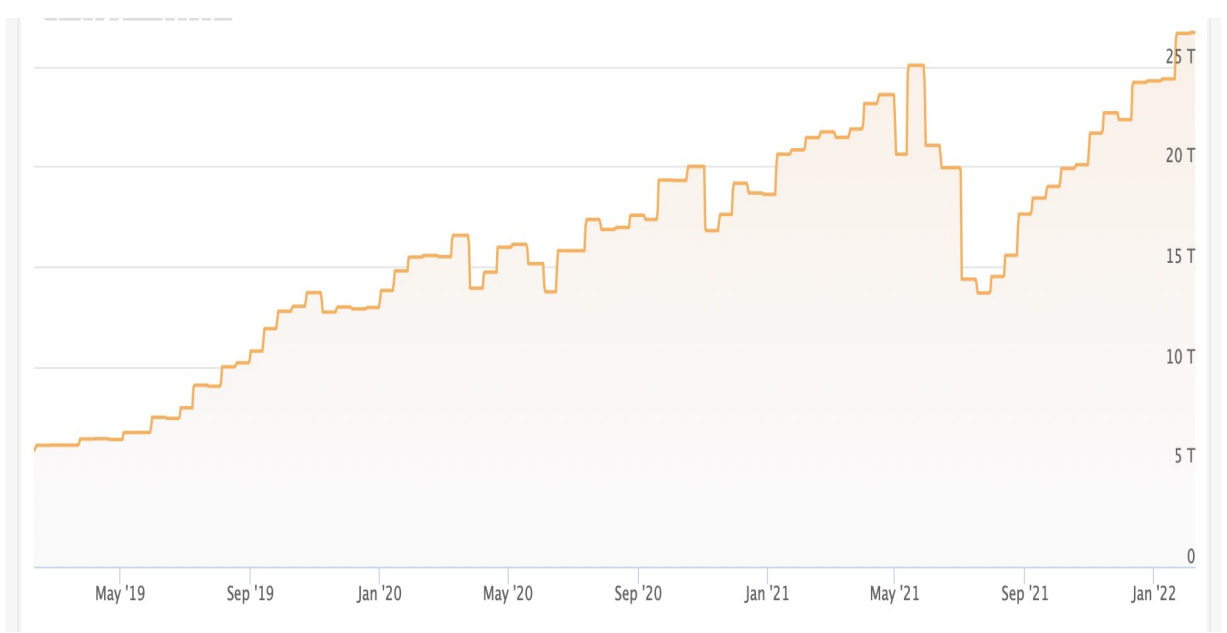
Graph at https://bitcoinwisdom.com/bitcoin/difficulty
The graph of the hash rate: https://blockchain.info/de/charts/hash-rate?timespan=all

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

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