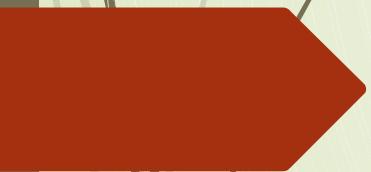


CSM3062: Blockchain Technology



**Dr. Mohammad Sajid
Assistant Professor
Department of Computer Science
Aligarh Muslim University, Aligarh-202002**

Books

Text Book

- ***Blockchain Technology*** by Chandramouli Subramanian, Asha A George, Abhilash KA, and Meena Karthikeyan, Orient Blackswan

Reference Books/Papers

- Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
- Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies, O'Reilly, 2014.
- Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008
- Gavin Wood, ETHEREUM: A Secure Decentralized Transaction Ledger, Yellow paper.2014.
- Henning Diedrich, Ethereum: Blockchain, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations, 2016
- Imran Bashir, "Mastering Blockchain: Distributed Ledger Technology, Decentralization and Smart Contracts Explained", Second Edition, Packt Publishing, 2018.
- Arshdeep Bahga, Vijay Madisetti, "Blockchain Applications: A Hands-On Approach", VPT, 2017.
- Roger Wattenhofer, "The Science of the Blockchain" CreateSpace Independent Publishing, 2016.
- Alex Leverington, "Ethereum Programming" Packt Publishing, 2017.

Course Goals

OBJECTIVES OF THE COURSE:

- To learn the basics of blockchain and cryptocurrency.
- To introduce the fundamental concepts of public and private blockchain systems.
- To learn issues, applications, and limitations of blockchain.

COURSE OUTCOMES:

The students would be able to:

- Understand the basic concepts of blockchain.
- Explain the creation of Blockchain.
- Understand the structure of public and private blockchain systems.
- Explain smart contracts and protocols for blockchain systems.
- Explain the applications and Limitations of blockchain.

Blockchain Trends

Blockchain Technology is “All Hype” or “Game Changer”

- <https://stackoverflow.blog/2021/06/07/most-developers-believe-blockchain-technology-is-a-game-changer-3/>
- <https://stackoverflow.com/questions/tagged/blockchain>
- <https://www.gartner.com/en/articles/what-is-blockchain#:~:text=Gartner%20research%20shows%20that%20from,technologies%20and%20processes%20can't.>
- <https://www.gartner.com/en/information-technology/insights/blockchain>
- <https://www.idc.com/getdoc.jsp?containerId=prUS47617821>
- https://www.idc.com/getdoc.jsp?containerId=IDC_P36801

Abstract from A Peer-to-Peer Electronic Cash System, 2008

- A PURELY PEER-TO-PEER VERSION OF electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.
- The longest chain not only serves as proof of the sequence of events witnessed but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure.
- Messages are broadcast on a best-effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Conclusion from A Peer-to-Peer Electronic Cash System, 2008

- WE HAVE PROPOSED A SYSTEM FOR electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity.
- Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best-effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

Blockchain

- Idea was coined by Stuart Haber and W. Scott Stornetta in 1991
 - How to time-stamp a digital document.
- Blockchain Technology was first published in 2008 by Satoshi Nakamoto.
 - A Peer-to-Peer Electronic Cash System, 2008
- In 2009, Satoshi Nakamoto implemented the first application of Blockchain, i.e., Bitcoin.

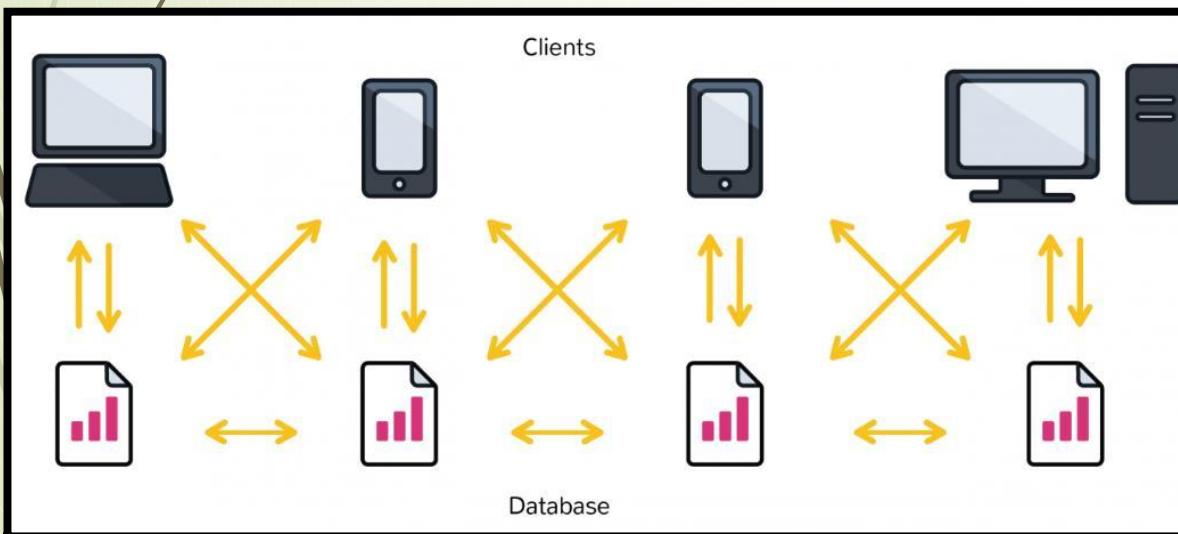
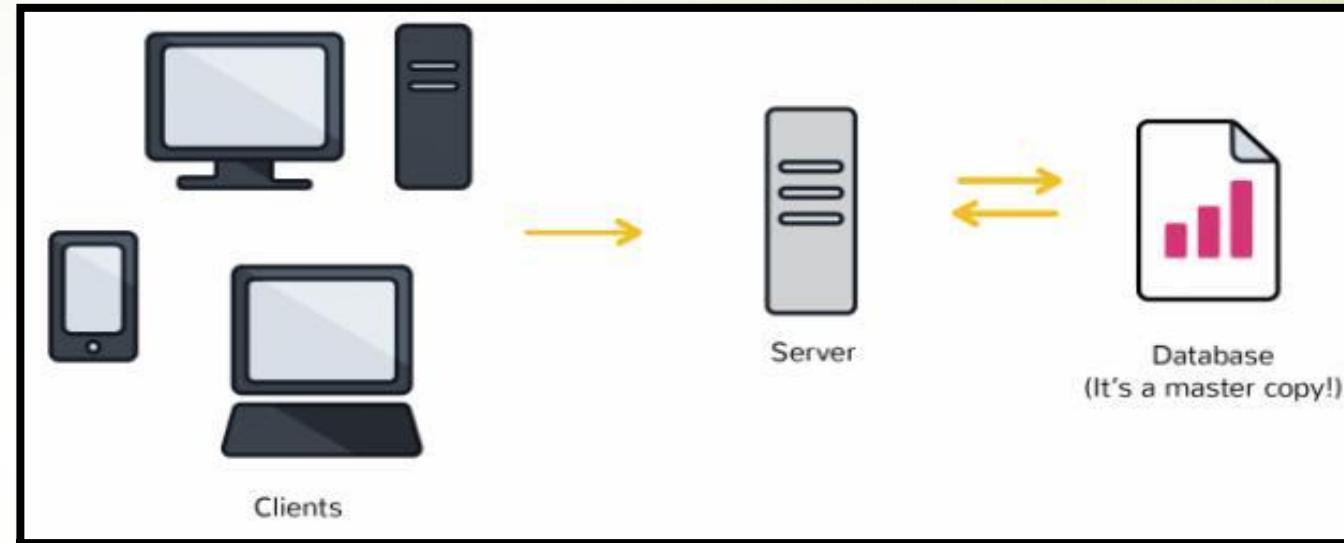
Blockchain Idea

- The Idea was to create a decentralized digital currency free from government regulations.
- Two people can confidently transact directly without any mediator.
- It reduces Government Control over cross-border transactions.
- It speeds up the transaction process by the need for third-party intermediaries.
- Blockchain technology redefines how we store, update, and move data (**Is it DBMS ?**).

Distinction between databases and blockchain ledgers

- Distinction between databases and blockchain ledgers
 - *It begins with architecture*

- **Blockchain ledgers**



- **Databases**

Source:

<https://www.coindesk.com/information/what-is-the-difference-blockchain-and-database/>

Distinction between databases and blockchain ledgers

Databases	VS	Blockchains
		
Databases have admins & centralized control		No one is the admin or in-charge
Only entities with rights can access database		Anyone can access (public) blockchain
Only entities entitled to read or write can do so		Anyone with the right proof of work can write on the blockchain
Databases are fast		Blockchains are slow
No history of records & ownership of digital records		History of records & ownership of digital records

Source: <https://coinsutra.com/blockchain-vs-database/>

Blockchain: An introduction

Definition

- Blockchain is defined as a **distributed, replicated peer-to-peer network of databases** that allows multiple non-trusting parties to transact **without a trusted intermediary** and maintains an ever-growing, append-only, **tamper-resistant list** of time-sequenced records.
- Blockchain can be considered as a type of **distributed ledger** that sits on the internet for recording transactions and maintaining a permanent and **verifiable record-set of information**.

Blockchain: An introduction

▪ What is it?

- Aka DLT (Distributed Ledger Technology) - rudimentary shared accounting system
- Technologically, it is :
 - Distributed database – public ledger (you can insert, and select new data, but **can't** update or delete old data).
 - Based on **P2P** (peer-to-peer) technology, cryptology, and API.
 - Distributed computer – execute digital contracts.

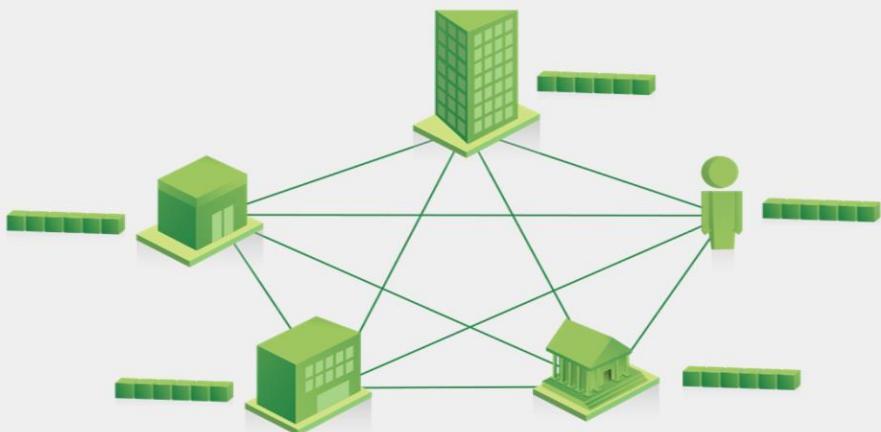
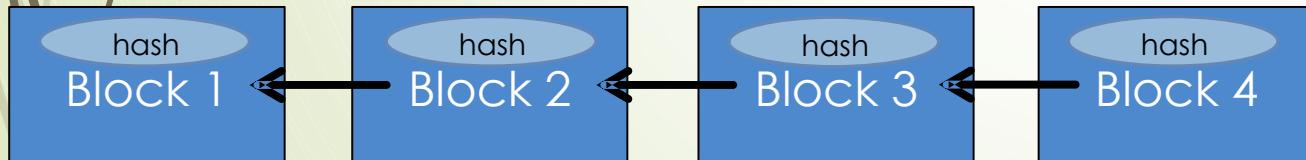


Image source:
https://www.ibm.com/blockchain/assets/images/landing/blockchain_shared_ledger.png

Blockchain: An introduction

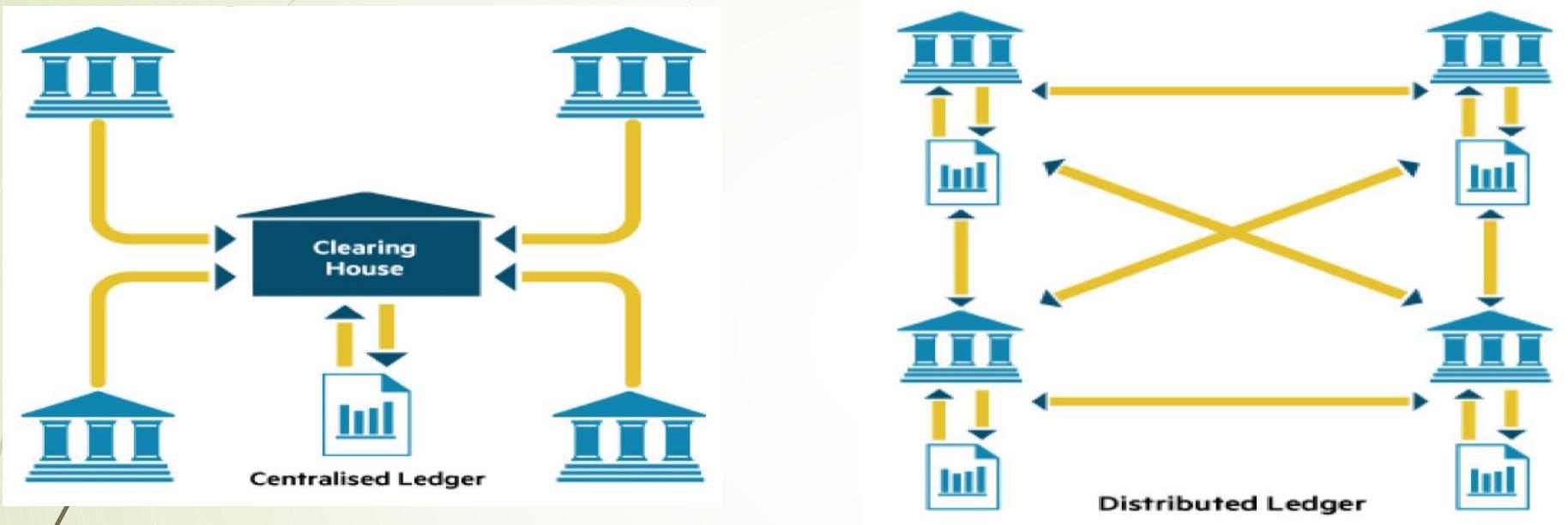
- **What is it? It**
 - usually contains financial transactions
 - is replicated across a number of systems in almost real-time
 - Uses cryptography and digital signatures to prove identity, and authenticity and enforce read/write access rights
 - Can be written by everyone in a public blockchain (but only certain participants in a private blockchain)
 - Can be read by participants, often a wider audience
 - Has mechanisms to make it hard to change historical records, or at least make it easy to detect when someone is trying to do so



Source: <https://miethereum.com/wp-content/uploads/2017/11/A.-A-Gentle-Introduction-To-Blockchain-Technology.pdf>

Blockchain: An introduction

- **Distributed ledger?**

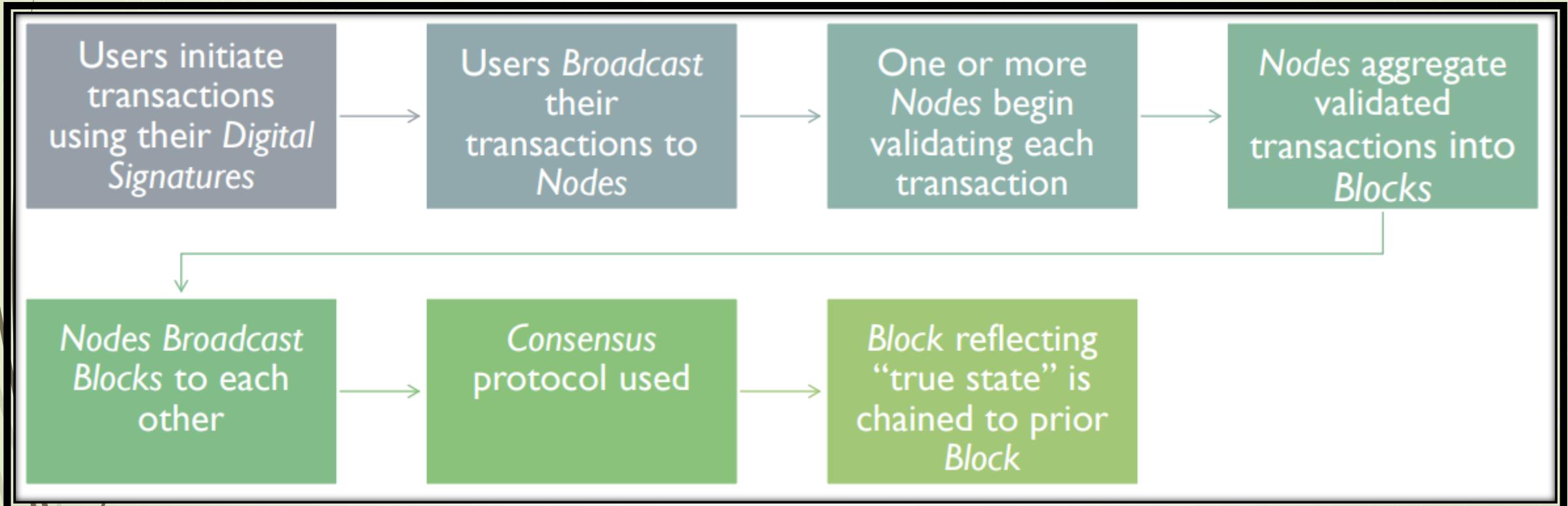


Source: <https://tradeix.com/distributed-ledger-technology/>

Image source: <https://knowledgecrypto.com/the-difference-between-blockchains-distributed-ledger-technology/>

Blockchain: An introduction

- **Distributed ledger – How it works?**



Source:

[https://ccl.yale.edu/sites/default/files/files/A%20Brief%20Introduction%20to%20Blockchain%20\(Final%20without%20Notes\).pdf](https://ccl.yale.edu/sites/default/files/files/A%20Brief%20Introduction%20to%20Blockchain%20(Final%20without%20Notes).pdf)

Blockchain: An introduction

Transaction & blocks

- A transaction is a value transfer
- A block is a collection of transactions on the blockchain network, gathered into a block that is hashed and added to the blockchain.

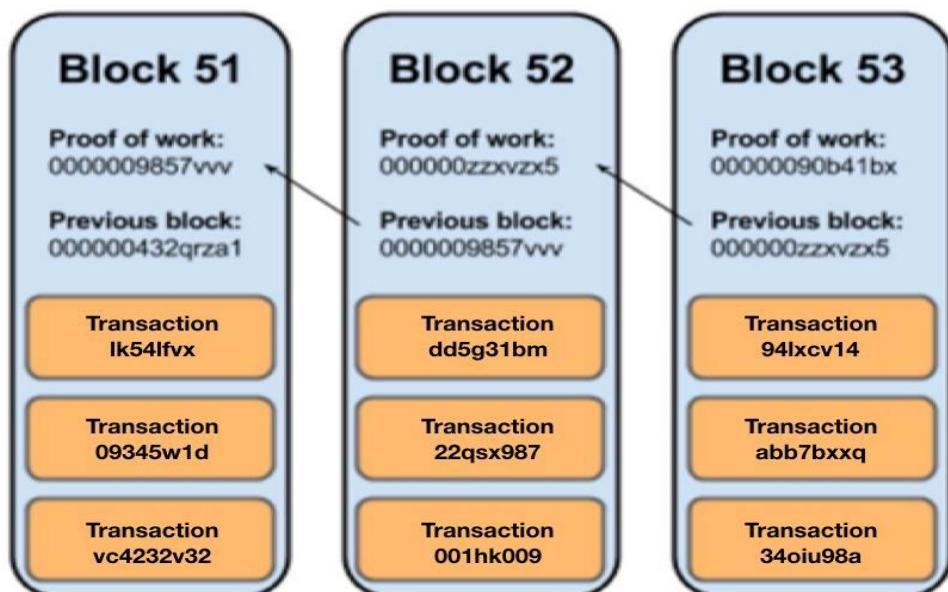
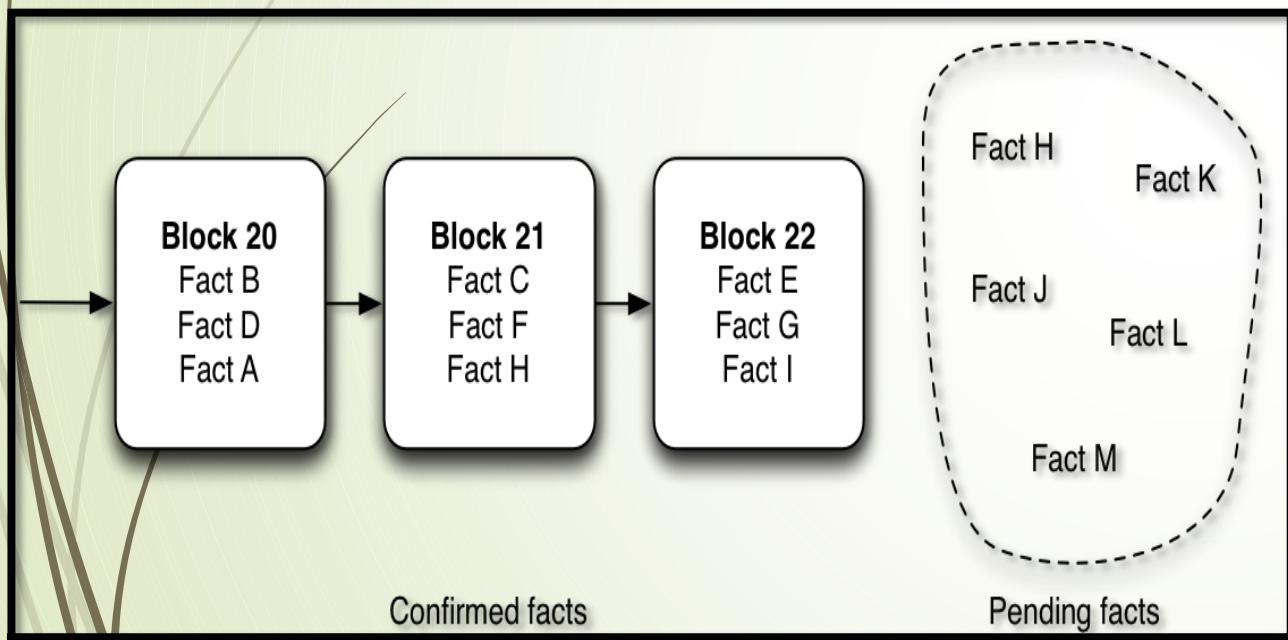


Image source:
<https://pplware.sapo.pt/informacao/monero-xmr-uma-moeda-segura-privada-e-sem-rasto/>

Blockchain: An introduction

▪ Mining

- The process of solving cryptographic problems using computing hardware also triggers the release of cryptocurrencies.
- The process of finding a new block (with a unique hash and some transactions).



Source:
<https://marmelab.com/blog/2016/05/12/blockchain-expliquee-aux-developpeurs-web-la-theorie.html>

Blockchain: An introduction

- **Mining**

- Miners on the network select transactions from pools and form them into a ‘block’.

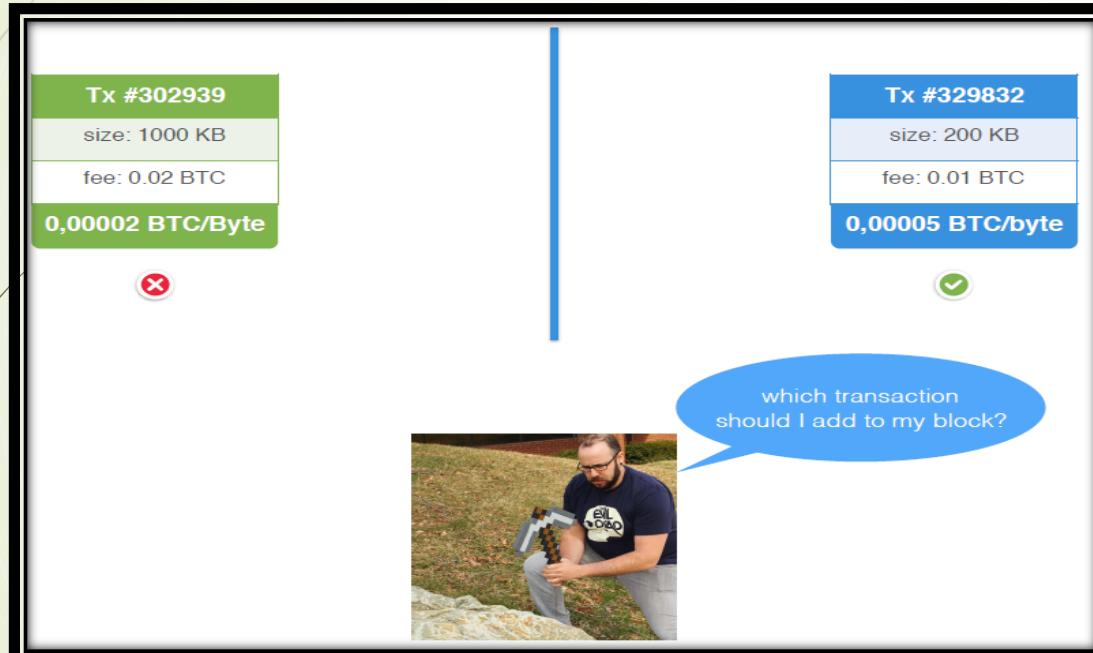


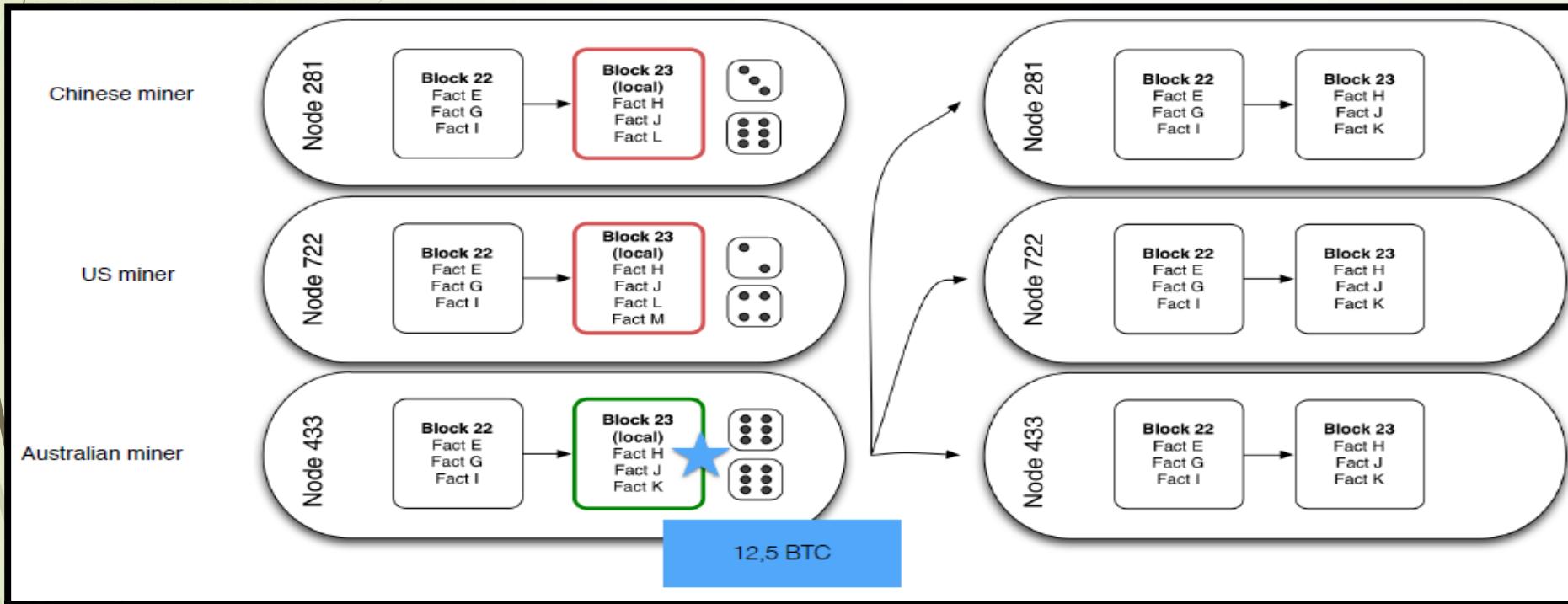
Image source:

https://www.thinkgeek.com/images/products/additional-carousel/e847_minecraft_pickaxe_inuse.jpg

Blockchain: An introduction

- **Consensus Mechanism**

- The process by which transactions (Blocks) are verified and added to a Blockchain.



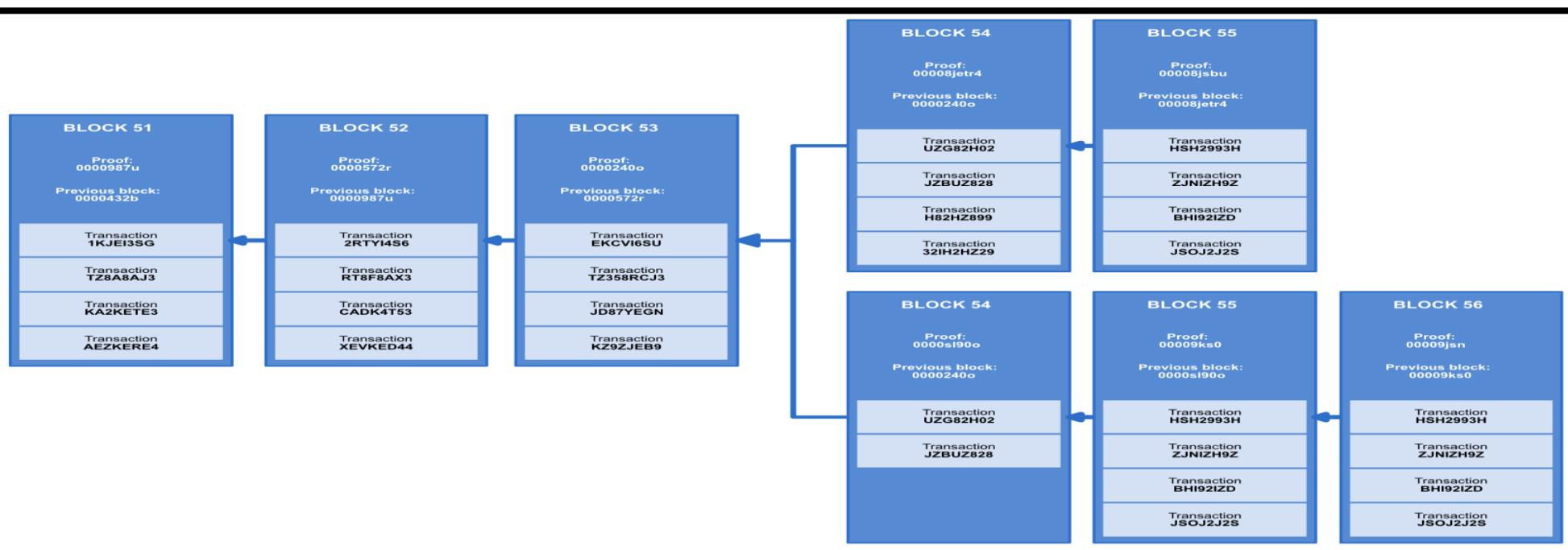
Source:
<https://marmelab.com/blog/2016/05/12/blockchain-expliquee-aux-developpeurs-web-la-theorie.html>

Blockchain: An introduction

❑ Forks

- ❑ A fork is the creation of an ongoing alternative version of the blockchain, by creating two blocks simultaneously on different parts of the network.
- ❑ This creates two parallel blockchains, where one of the two is the winning Blockchain.
- ❑ When does it happen?
 - Block found at the same time
 - Software incompatibility
 - “We don’t agree” split

Source: <https://medium.com/my-blockchain-bible/101-blockchain-terminology-874f007c0270>



Blockchain terminologies



- **Bitcoin**
 - Monetary creation

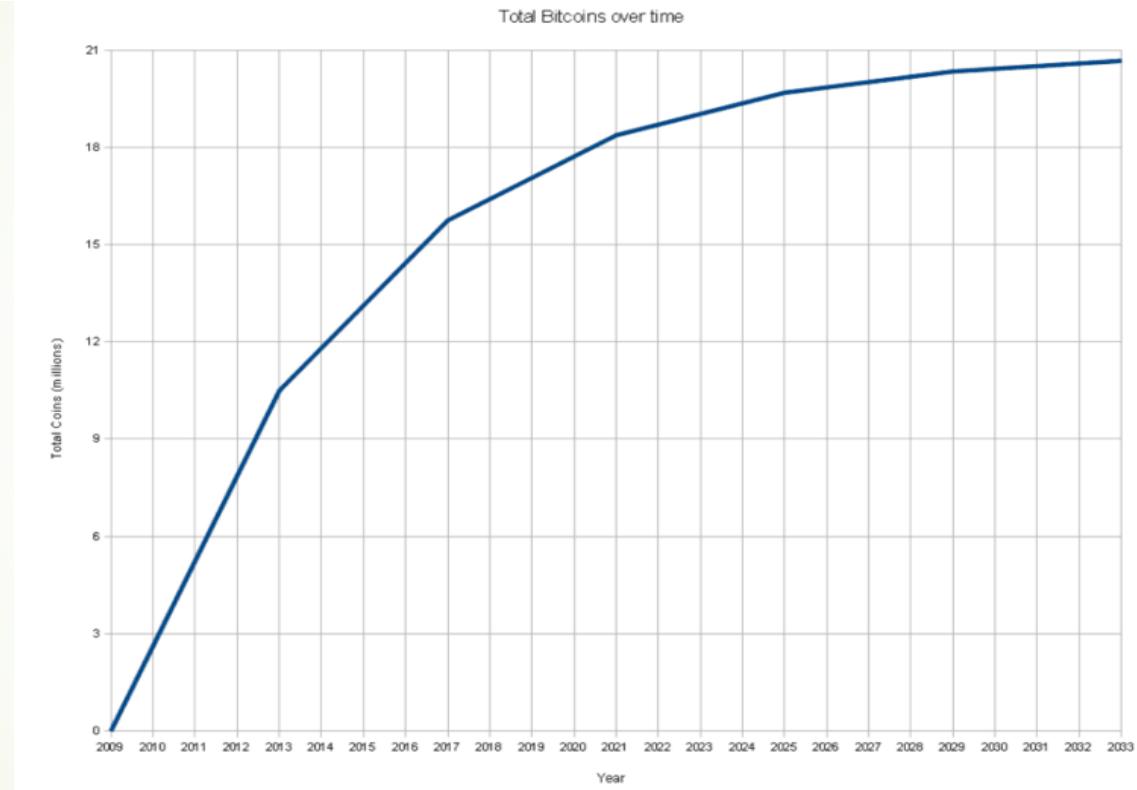


Image source:

https://upload.wikimedia.org/wikipedia/commons/thumb/5/54/Total_bitcoins_over_time.png/740px-Total_bitcoins_over_time.png

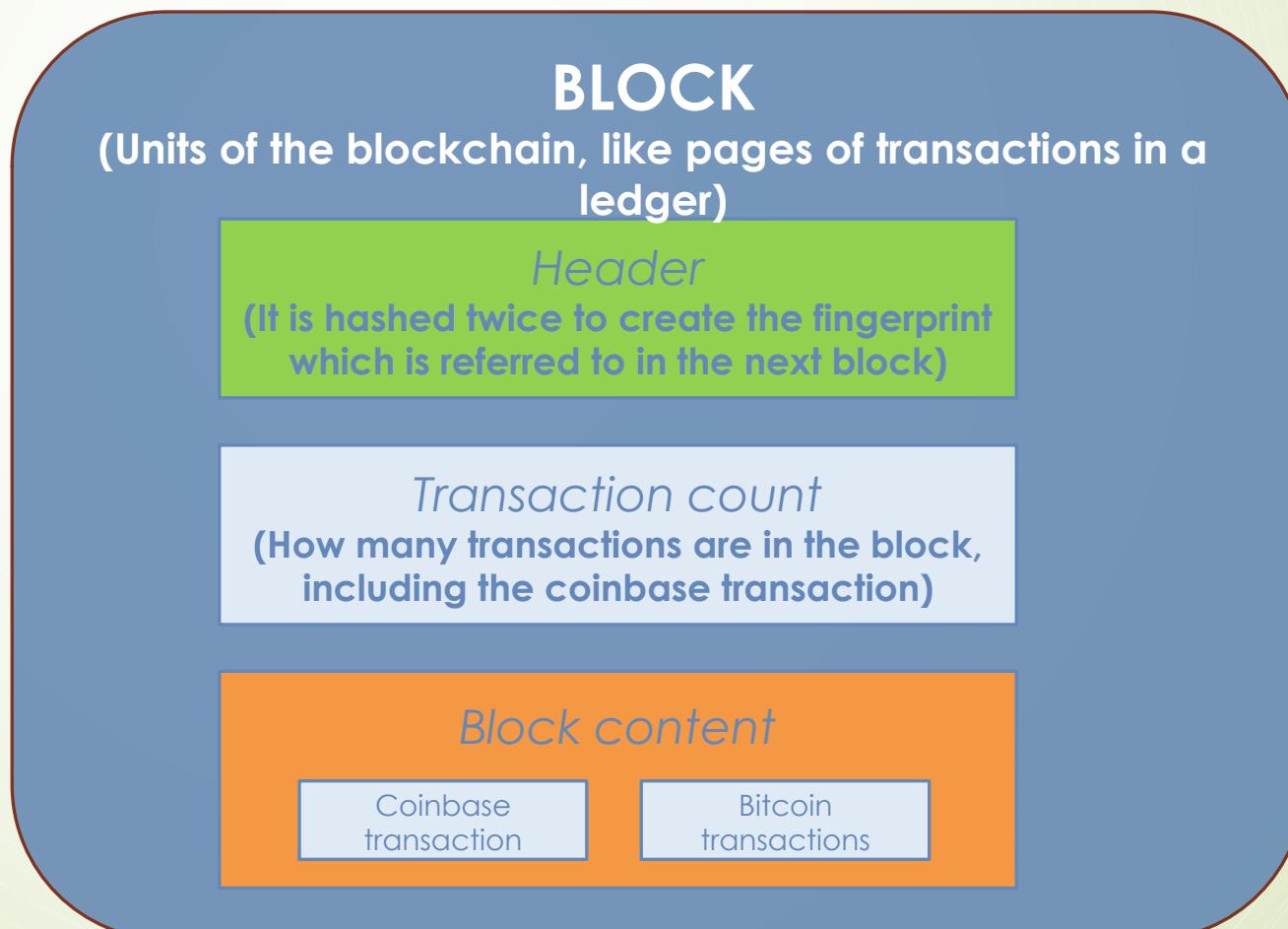
Source: https://en.bitcoin.it/wiki/Controlled_supply

Blockchain terminologies

- **Bitcoin**
 - Inside Bitcoin's Blockchain



Source:
https://bitsonblocks.files.wordpress.com/2015/09/bitcoin_blockchain_infographic1.jpg



Blockchain terminologies



- **Bitcoin**

- Inside Bitcoin's Blockchain
 - *Block Header*: includes Technical data, Previous block hash, Merkle Root, Timestamp, Difficulty target,Nonce.

Here is an example:

Source:

https://bitsonblocks.files.wordpress.com/2015/09/bitcoin_blockchain_infographic1.jpg

Image source: www.blockchain.com

Height	448909
Block time	2017-01-19 09:32:58
Trades sum	5,340.87080329 BTC
Nb txs	1637
Difficulty	336,899,932,795.81
Fee	0.41239309 BTC
Hash	000000000000000000dbc2853f4939baad1f09d086fa68a0105d79378bf7629
Version	127
Confirmations	1
Merkle root	a4772eff88cbe645bba832d31730f0b42ea4d8d05d02ea62be533316bd3fb197
Prev block hash	0000000000000000000015278f089845eaa41753e61a0f97c54b364325ca74a6275
Size	947.32 kB
Coin days destroyed	2,913.95 ?

Blockchain terminologies



- **Bitcoin**
 - Inside Bitcoin's Blockchain
 - *Block content* : Transaction Flow

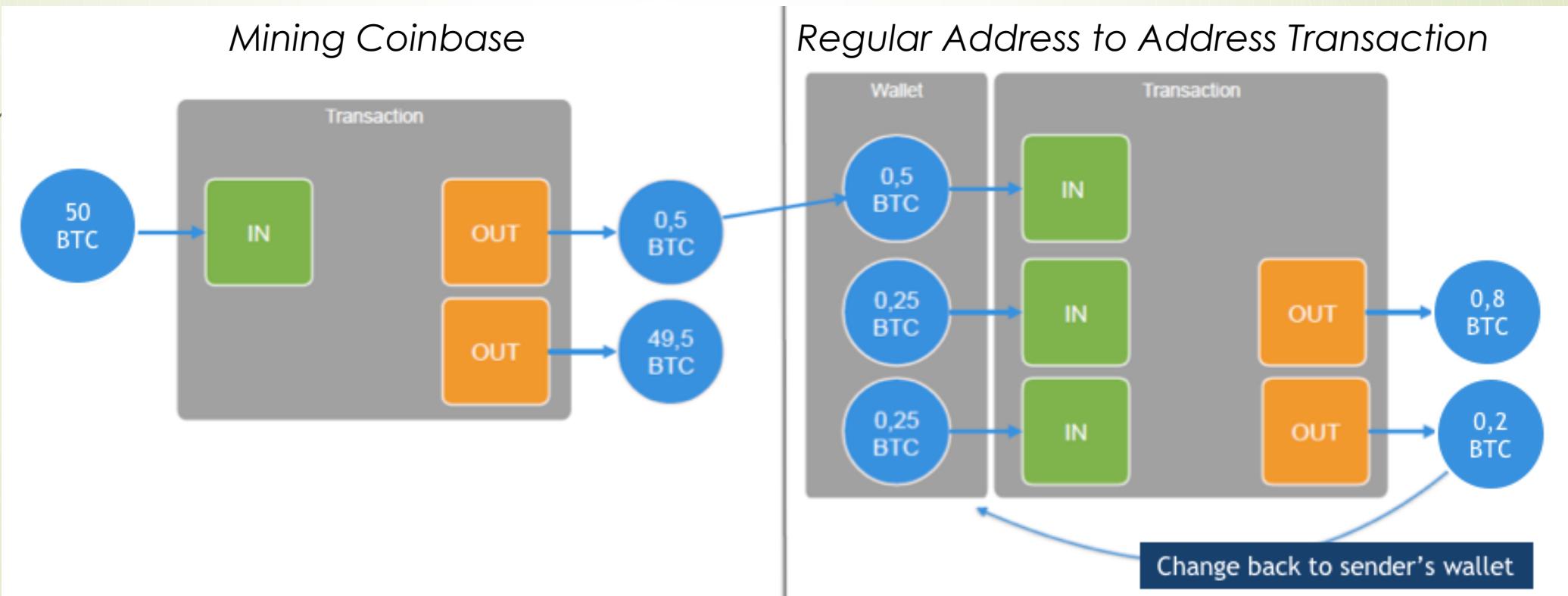


Image source: Scorech

Blockchain terminologies

- Bitcoin
 - Inside Bitcoin's Blockchain
 - Block Transaction example:



coinbase 86c3532df82e5746611cb640fd2482b8c0794fe3c0c1ea5bb4a2bea2317db293			
Newly generated coins		3NA8hsjfdgVkmVS9moHmkZsVCoLxUkvv	12.91239309
	NONSTANDARD		0
		Fee:	0.00000000
Transaction sum: 12.91239309			
34ae7288e0d245f0c1642c726c71aa72156923dbf16a1fa6f7aba6493f7290d1			
1Ku2paKQx4Syy2dx6x7wkUSxRpgr1U1oyq	-3.4871	1NYHREgzVYoA38Zv6tdpcHSkn9bpVRreWy	1.1
	1JE5db6FabSg8cpw1VKvi9hsXU4vgiQhJW		2.3862
		Fee:	0.00090000
Transaction sum: 3.48710000			
b2b8f254c9af388cea47cd63ad7856b70ce976c6ce5e89516c4fc8315fc0e8c			
1JE5db6FabSg8cpw1VKvi9hsXU4vgiQhJW	-2.3862	1ANyp8aNehCJ29fDevnEPwEFFmXZ2eRoym	1.1
	1FmiZLGEP7WvQSvJqZXNNDc8EUdZ9zTqVr		1.2853
		Fee:	0.00090000
Transaction sum: 2.38620000			

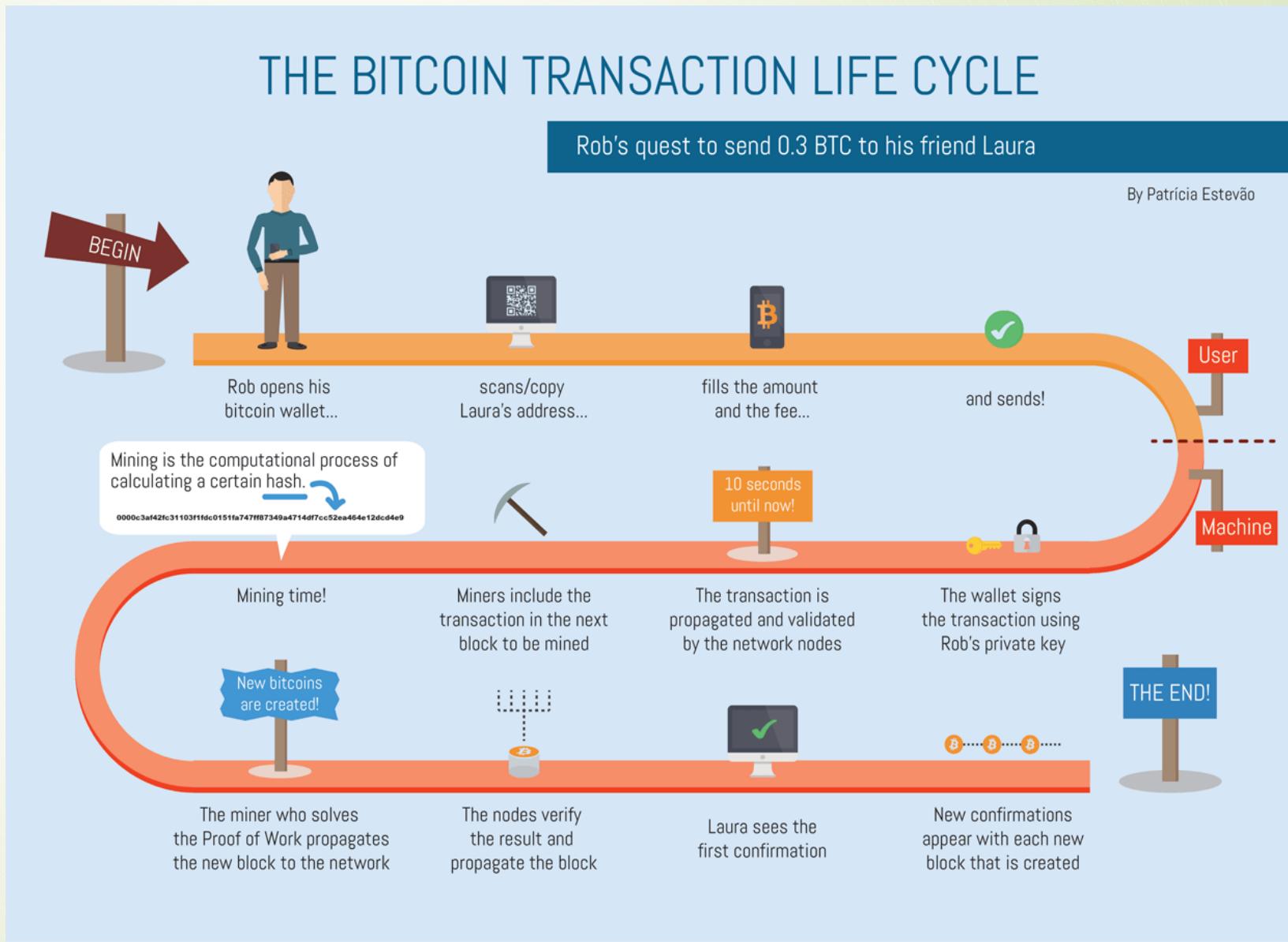
Image source:
www.blockchain.com

Blockchain terminologies

- Bitcoin
 - How the money transfer works?

Image source:

<https://www.weusecoins.com/images/bitcoin-transaction-life-cycle-high-resolution.png>



Blockchain terminologies

- **Ethereum**

- Proposed in late 2013 by Vitalik Buterin (cryptocurrency researcher and programmer)
- Online crowdsale during summer 2014
- Bitcoin on steroids!

“A blockchain is a magic computer that anyone can upload programs to and leave the programs to self-execute, where the current and all previous states of every program are always publicly visible, and which carries a very strong cryptoeconomically secured guarantee that programs running on the chain will continue to execute in exactly the way that the blockchain protocol specifies.” Vitalik Buterin



Source: <https://medium.com/blockchain-review/how-does-the-blockchain-work-for-dummies-explained-simply-9f94d386e093>

Image source: https://znews-photo-td.zadn.vn/w660/Uploaded/lce_uxlcq/2017_06_27/20DBBITCOIN4master675.jpg

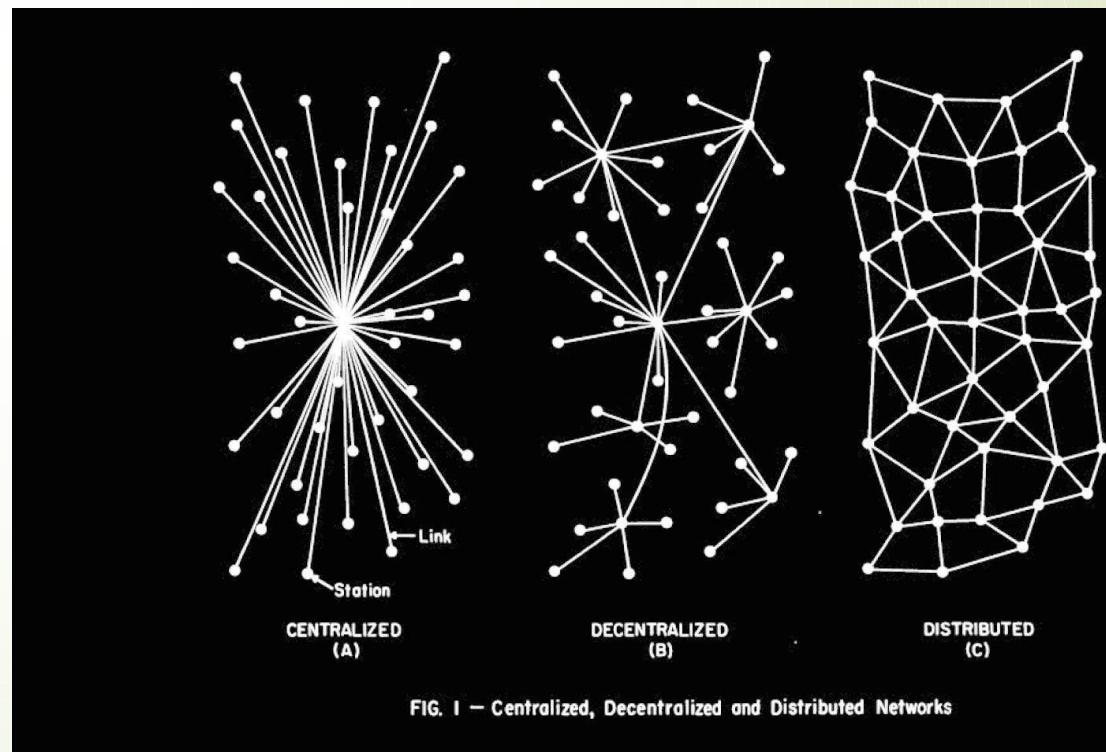
Blockchain terminologies

- **Ethereum**

- Decentralised app platform (dapps)
- Transaction & smart-contracts ledger
- Based on the Ethereum Virtual Machine (EVM)
- Cryptocurrency called ether (ETH)

Image source:

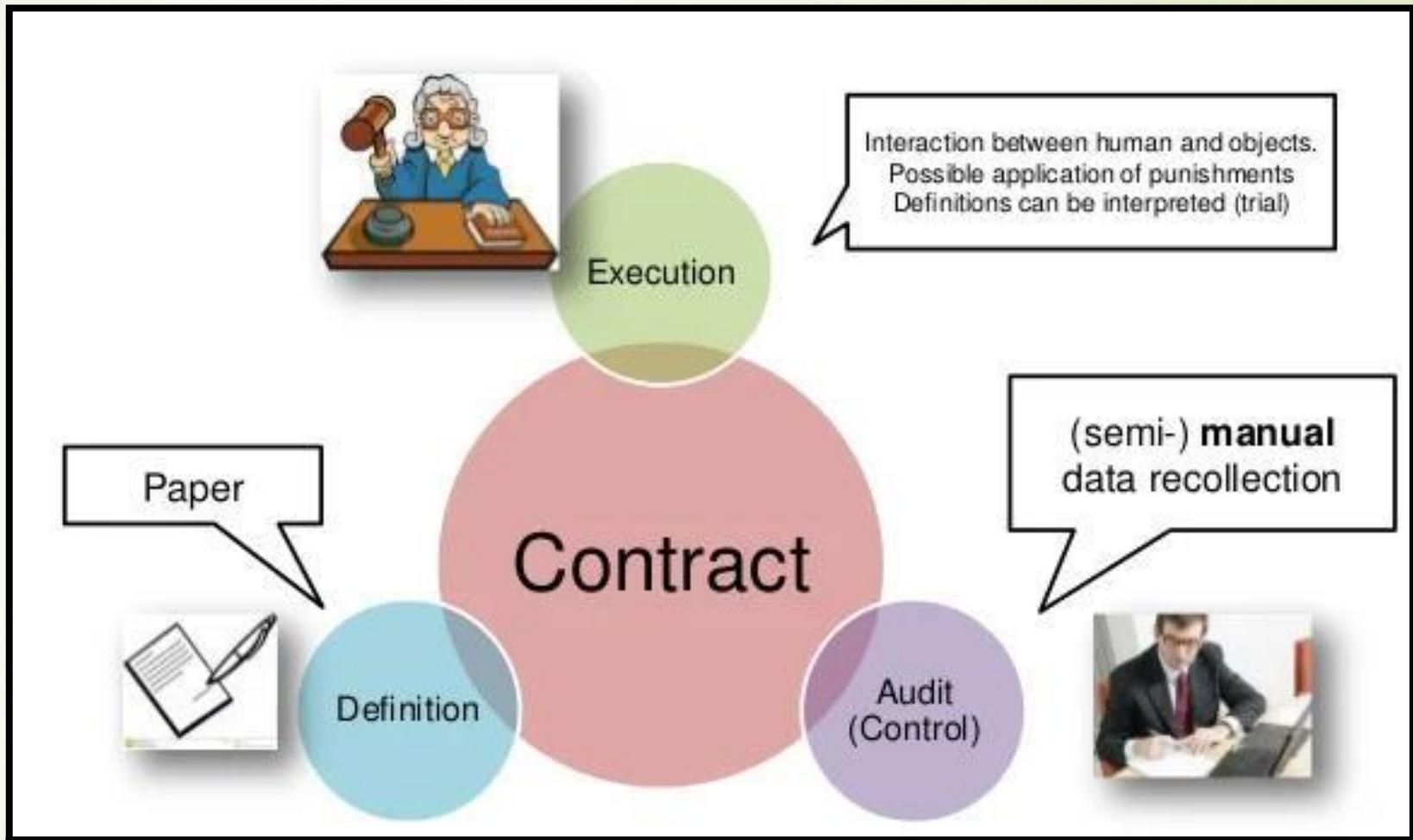
<https://image.slidesharecdn.com/empresaeinovaonasociedadeemredemarо2013-130717064842-phpapp01/95/empresa-e-inovao-na-sociedade-em-rede-84-638.jpg?cb=1374043787>



Blockchain terminologies

- Ethereum
 - *Smart Contract*

How a “Traditional” contract works?



Source: <https://www.investopedia.com/terms/s/smart-contracts.asp>

Image source: <https://image.slidesharecdn.com/smart-contracts-150925125324-lva1-app6892/95/smart-contracts-4-638.jpg?cb=1443185644>

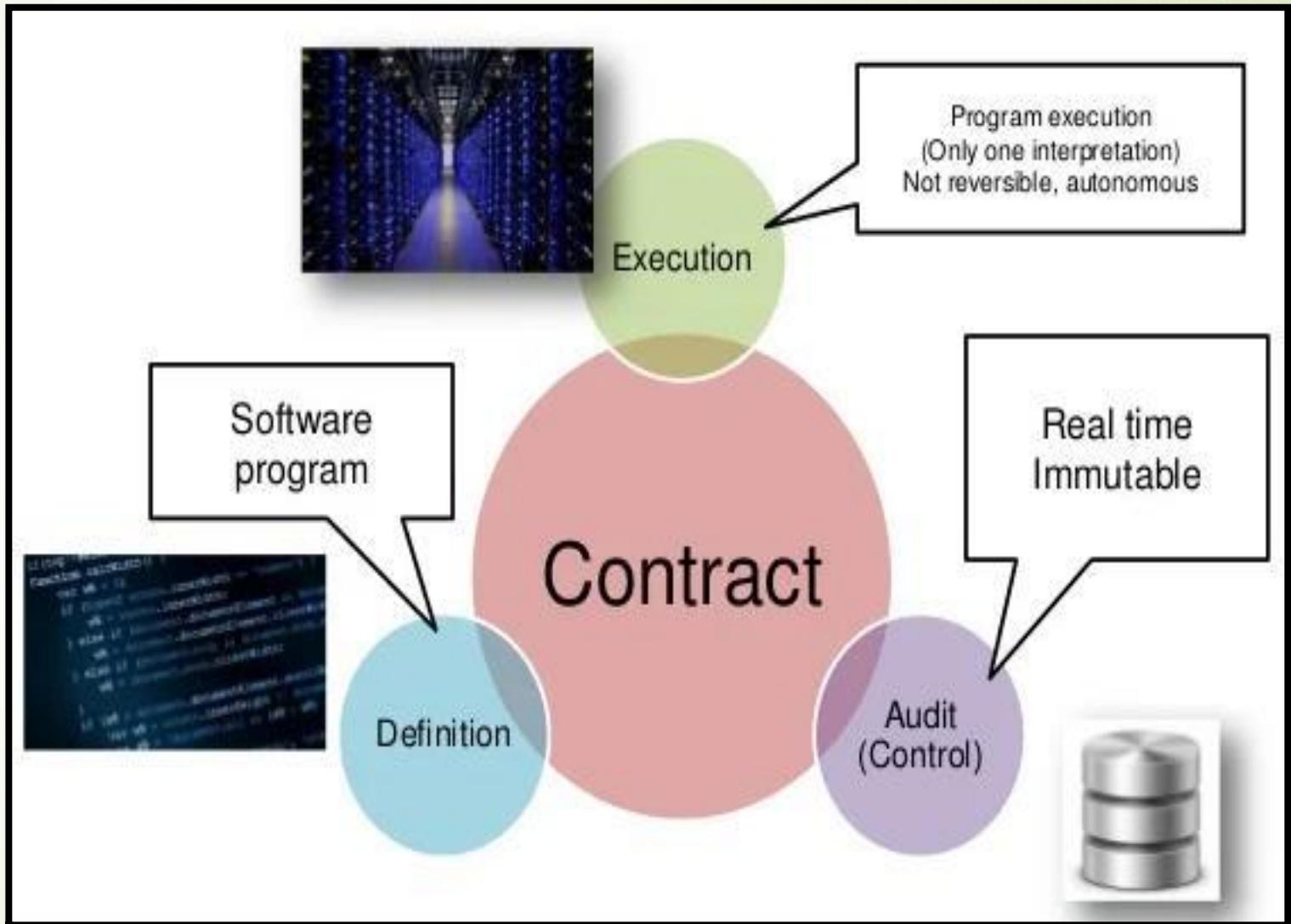
Blockchain terminologies

- **Ethereum**
 - *Smart Contract*

How a “*Smart Contract*” contract works:

Source:
<https://www.investopedia.com/terms/s/smart-contracts.asp>

Image source:
<https://image.slidesharecdn.com/smart-contracts-150925125324-lva1-app6892/95/smart-contracts-5-638.jpg?cb=1443185644>



Bitcoin and Blockchain

- Though the terms ‘Bitcoin’ and ‘Blockchain’ are often used interchangeably, they are not the same.
- Blockchain is the underpinning technology that Bitcoin was built on.
- *“A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network.”*
 - Part of the abstract from the paper “A Peer-to-Peer Electronic Cash System.”

Bitcoin and Blockchain

'Bitcoin' and 'Blockchain' are not the same.

- **Meaning**
 - Bitcoin is a Cryptocurrency, while Blockchain is a technology (a database of information).
 - Blockchain provides a secure environment that Bitcoin needs for peer-to-peer transactions.
 - Blockchain acts as a bitcoin's ledger and maintains all the transactions of the bitcoin.
- **Transparency**
 - Bitcoin has a high degree of anonymity.
 - Blockchain is quite transparent as it is expected to work across multiple industry applications.

Bitcoin and Blockchain

- **Scope of Usage**

- Bitcoin is only one application of Bitcoin Technology.
- Blockchain can also be used for Identity management, Records Management, research management, and others.
- It can also be used in various industry sectors like the Financial sector, Digital identity, Education sector Logistics sector, Health sector, Insurance sector, Retail sector Energy sector, Blockchain in Agriculture Land registrations, and others

Blockchain-related initiatives (India)

Andhra Pradesh

- Blockchain Database
- Cybersecurity
- Healthcare
- Land Registry
- Vehicle Registration

Assam

- Public Service Delivery

Delhi

- Monitoring Growth and Maintenance of Saplings and Plants

Goa

- Land Registry

Gujarat

- Fertilizer Subsidy Management
- e-Governance

Karnataka

- Agriculture
- Digital Certificates
- Forest and Land Acquisition
- Public Service Delivery
- Idea Marketplace
- IP Protection

Kerala

- Farm Insurance
- Agriculture Insurance

Maharashtra

- Land Registry
- Digital Certifications
- Organ Transplants
- Rationing Distribution
- Farm Insurance

Rajasthan

- Electronic Health records (EHR)
- Land Registry

Tamil Nadu

- Agriculture
- Healthcare
- Education

Telangana

- Land Registry
- Chit Funds Operations
- Digital Education Certificates

Uttar Pradesh

- Land Registry
- Power Sharing

West Bengal

- Land Registration
- Duty Payments
- Record Management
- Cybersecurity
- Digital Birth Certificates
- Data Management

Career Opportunities in Blockchain

- **Blockchain Developer**
- **Bitcoin cryptocurrency developer**
- **Blockchain Software Engineer**
- **Blockchain Principal Program Manager**
- **Business Analytics Associate**
- **Cloud engineer with bitcoin protocol/blockchain**
- **Cryptocurrency analyst**
- **Cryptocurrency developer**
- **Cryptocurrency mining engineer**
- **Financial analyst**
- **Research analyst: blockchain**

Key Concepts of Blockchain Technology

1. Peer-to-peer network
2. Public Key Cryptography
3. Distributed Consensus

Peer-to-Peer network

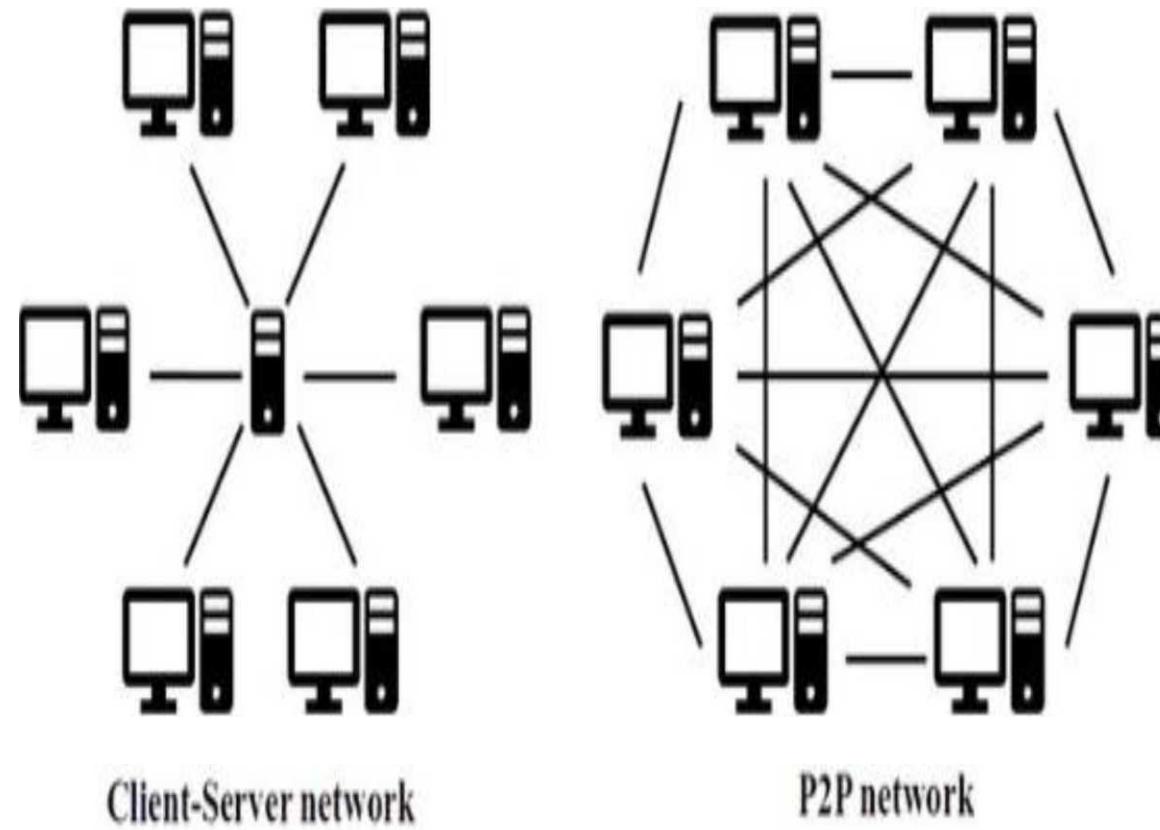
Client-Server Network

- A client-server network is a communications model in which multiple client programs share the services of a common server program.

Peer-to-Peer Network

- The P2P concept is not new as its first use-case dates back to the 1990s when it was first used in the first file-sharing programs.
- A group of nodes that are connected together to create a network.
- The network can be used to share files and store them as well. All nodes can act as servers and a client.
- All the nodes generally have equal power and can perform the same tasks.
- In a P2P network, peers can exchange digital assets or cryptocurrencies/ Information/ files without the need to go through any intermediaries.

Peer-to-Peer network



Peer-to-Peer network

Examples-

- There are other use-cases of P2P including P2P loans, P2P car rental, P2P payments, and so on. Another useful use case is P2P insurance.
- There exist more than 2000+ crypto currencies that take advantage of these networks.
- The P2P networks are also used in distributed computing applications such as streaming platforms, web search engines, online marketplaces, and so on. It is also part of the InterPlanetary File System(IPFS) web protocol and BitTorrent.
- Blockchain technology has also been the front-runner when it comes to using P2P networks.
 - Blockchain is a P2P network where peers can communicate and do transactions without the need for centralized authority.

Peer-to-Peer network

In the client-server network, there is a centralized server from which the client downloads files.

How P2P Works ?

- Every node is responsible for maintaining the distributed network.
- Each node needs to act as both client and server to other nodes on the server.
- Each of the nodes has a copy of the file. Each node acts as a server and needs to either download files from other nodes or upload them to other nodes.
- Each node can act as a server when it is required to serve files to other nodes. This sharing and receiving aspect can be done by a node simultaneously, which makes the P2P network so efficient and fast. The network tends to become more efficient as the network grows.

Advantages

- The P2P distributed architecture is secure and can defend against cyber attacks in a much better way.
- No central point of failure in a P2P network.
- It can also handle malicious attacks (false obsolete records/files).

Role of P2P in Blockchain Network

- Bitcoin introduced a key concept of blockchain where a distributed ledger called blockchain is managed by the P2P.
- P2P helps the cryptocurrencies to be available almost everywhere around the world within an instant.
- There is no centralized server requirement to carry out the operation makes P2P architecture and blockchain technology so amazing!
- Anyone can participate in the Bitcoin network and help in validating and verifying blocks, similar to that of an open P2P network where anyone can join and participate in the network.
- There is no need for a central authority to record or process transactions due to P2P architecture.
- If someone tries to play with the data, and try to modify it, then it will result in a malicious activity in which the network is capable of stopping. It will discard any inaccurate data.

Role of P2P in Blockchain Network

- One more way the P2P network has influence in blockchains is how nodes participate in the network activities. Not all nodes have the same role. There are nodes with different roles.
- For example, there are full nodes that are capable of verifying transactions using the consensus algorithm set by the network. They help make the network more secure. The full nodes are also responsible for having a complete and updated copy of blockchain's ledger.

Advantages of P2P architecture in Blockchains

There is no doubt that P2P architecture used in blockchains brings a lot of benefits.

- The architecture is more secure compared to the client-server.
- No central point of failure
- Large number of nodes distributed across the network, Denial-of-Service(DoS) attacks are not possible.
- Another benefit is data immutability, where the data once is written cannot be altered. The bigger the network, the less chance it can be altered.
- To alter the data, the majority of the nodes need to be controlled by one entity to carry out a 51% attack.

Key Concepts of Blockchain Technology

1. Peer-to-peer network
2. **Public Key**

Cryptography

3. Distributed Consensus

Public Key Cryptography

William Stallings, “Cryptography and Network Security”, 5/e,
Chapter 9 - “Public Key Cryptography and RSA”. Lecture
slides by Lawrie Brown

Some Basic Terminology



- **plaintext** - original message
- **ciphertext** - coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (codebreaking)** - study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - field of both cryptography and cryptanalysis

Cryptography



It can be characterized by:

- type of encryption operations
 - substitution
 - transposition
 - product

- number of keys used
 - single-key or private
 - two-key or public

- way in which plaintext is processed
 - block
 - stream



Cryptanalysis

Objective of cryptanalysis is to recover key not just message General approaches:

- cryptanalytic attack

- **Chosen Plaintext Attack (CPA)**

- In this method, the attacker has the text of his choice encrypted. Thus, he has the ciphertext-plaintext pair of his choice. This simplifies his task of determining the encryption key.

- A popular public key cryptosystem, RSA is also vulnerable to chosen-plaintext attacks.

- brute-force attack

- If either type of attack succeeds in deducing the key, the effect is catastrophic

Symmetric Encryption

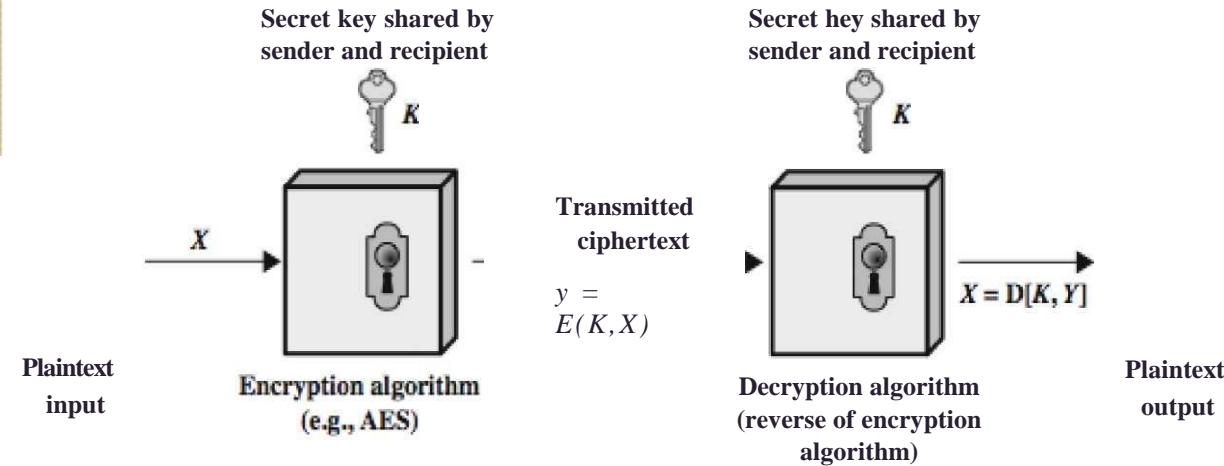


- Conventional / private-key / single-key Sender and recipient share a common key All classical encryption algorithms are private-key Only type prior to invention of public-key in 1970's

- Example-
 - Advanced Encryption System (AES)
 - Data Encryption System (DES)



Symmetric Cipher Model





Symmetric Cipher Model

- Plaintext - original message
- Encryption algorithm - performs substitutions/transformations on plaintext
- Secret key - control exact substitutions/transformations used in encryption algorithm
- Ciphertext - scrambled message
- Decryption algorithm - inverse of encryption algorithm



Symmetric Cipher Model-Requirements

- Requirements for secure use of symmetric encryption:
 - a strong encryption algorithm
 - a secret key known only to sender / receiver
- It can be defined mathematically as:
$$Y = E(K, X)$$
$$X = D(K, Y)$$
- It is assume that encryption algorithm is known
- A secure channel is needed to distribute key



Classical Substitution Ciphers

- The two basic building blocks of all encryption technique are substitution and transposition.
- Letters of plaintext are replaced by other letters/numbers/symbols
- If plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

Caesar Cipher

Earliest known substitution cipher by Julius Caesar

- First attested use in military affairs Uses a simple letter shift example:
 - meet me after the toga party
 - PHHW PH DIWHU WKH WRJD SDUWB



Caesar Cipher

- Message is transformed as:

a b c d e f g h i j k l m n o p q r s t u v w x y z D E F G H I J K L M N O P Q R S
T U V W X Y Z A B C

- Mathematically assign each letter a number

a b c d e f g h i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 19 20 21 22 23 24 25

- Mathematically Caesar cipher can be defined as:

$$c = E(k, p) = (p + k) \bmod 26$$

Where p is an alphabet in the plain text message, k is shift value and c is an alphabet in the cipher text.

- Example - Decode your name



Cryptanalysis of Caesar Cipher

- > Only have 26 possible ciphers • A maps to A,B,..Z
- > **brute force search:** Try all possibilities
- > given ciphertext, just try all shifts of letters
- > Example> Break ciphertext "GCUA VQ DTGCM"



Advanced Private Ciphers

- > Data Encryption System (DES)
- > Advanced Encryption System (AES)
- > Blowfish
- > Twofish
- > Etc.



Public-Key Cryptography

- All previous cryptographic systems have been based on the elementary tools of **substitution and permutation**.
- Traditional **private/secret/single key** cryptography uses **one** key shared by both sender and receiver
If this key is disclosed, the communications are compromised
- Also it is **symmetric**, parties are having same key.



Public-Key Cryptography

Probably most significant advancement in the 3000 year history of cryptography

Uses **two** keys - a public & a private key

Asymmetric since parties are **not** having same keys.

Complements **rather than** replaces private key crypto



Why Public-Key Cryptography?

Developed to address two key issues:

- **key distribution** - how to have secure communications in general without having to trust a KDC with your key
- **digital signatures** - how to verify a message comes intact from the claimed sender

Invention due to Whitfield Diffie & Martin Hellman at Stanford University in 1976

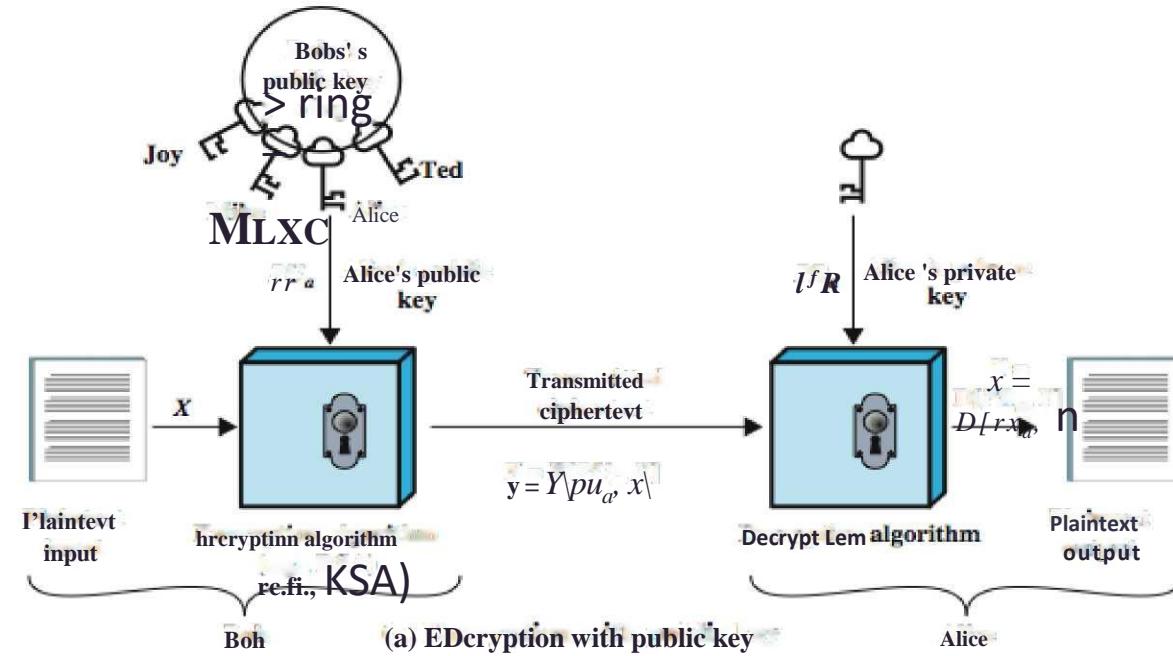
- known earlier in classified community



Public-Key Cryptography

- **Public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
 - a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
 - a related **private-key**, known only to the recipient, used to **decrypt messages**, and **sign (create) signatures**
- **Infeasible to determine private key from public**
- **Asymmetric** because
 - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

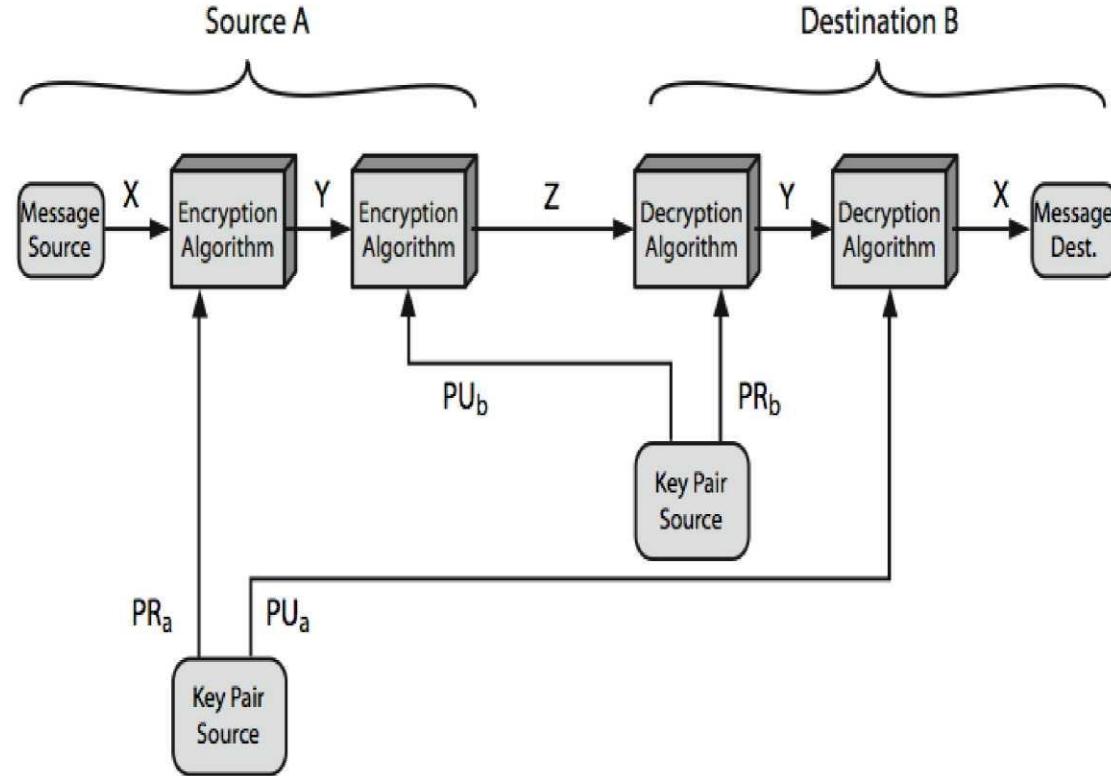
Public-Key Cryptography



Symmetric vs Public-Key

Conventional Encryption	Public-Key Encryption
<p>Needed to Work:</p> <ol style="list-style-type: none">1. The same algorithm with the same key is used for encryption and decryption.2. The sender and receiver must share the algorithm and the key. <p>Needed for Security!</p> <ol style="list-style-type: none">1. The key must be kept secret.2. It must be impossible or at least impractical to decipher a message if no other information is available.3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	<p>Needed to Work:</p> <ol style="list-style-type: none">1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption.2. The sender and receiver must each have one of the matched pair of keys (not the same one). <p>Impeded for Security:</p> <ol style="list-style-type: none">1. One of the two keys must be kept secret.2. It must be impossible or at least impractical to decipher a message if no other information is available.3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Public-Key Cryptosystems





Public-Key Applications

- It can be applied into 3 categories:
 - **encryption/decryption** (provide secrecy)
 - **digital signatures** (provide authentication)
 - **key exchange (of session keys)**
- Some algorithms are suitable for all uses, others are specific to one

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Difftse Heliman	No	No	Yes
DS5	No	Yes	No



Public-Key Requirements

- Public-Key algorithms rely on two keys where:
 - it is computationally infeasible to find decryption key knowing only algorithm & encryption key
 - it is computationally feasible to en/decrypt messages when the relevant (en/decrypt) key is known
 - either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)
- The above are formidable requirements which only a few algorithms have satisfied



Public-Key Requirements

- Need a trapdoor one-way function One-way
- function has the following property:
 - $Y = f(X)$ easy
 - $X = f^{-1}(Y)$ infeasible
- A trap-door one-way function has
 - $Y = f_k(X)$ easy, if k and X are known
 - $X = f_k^{-1}(Y)$ easy, if k and Y are known
 - $X = f_k^{-1}(Y)$ infeasible, if Y known but k not known
- A practical public-key scheme depends on a suitable trap-door one-way function



Security of Public Key Schemes

- Like private key schemes brute force **exhaustive search** attack is always theoretically possible
 - >but keys used are too large (>512bits)
 - ^requires the use of **very large numbers**
 - > hence is **slow** compared to private key schemes



RSA

- Proposed by Rivest, Shamir & Adleman at MIT in 1977
- Best known & widely used public-key scheme
- Uses large integers (eg. 1024 bits)
- Security due to cost of factoring large numbers



RSA En/decryption

- To encrypt a message M , the sender:
 - obtains **public key** of recipient $PU=\{e,n\}$
 - computes: $C = M^e \text{ mod } n$, where $0 < M < n$
- ciphertext C the owner:
 - uses their private key $PR=\{d, n\}$
 - computes: $M = C^d \text{ mod } n$
- Note that the message M must be smaller than the modulus n
(block if needed)



RSA Key Setup

- Each user generates a public/private key pair by selecting **two large primes at random: p, q**
 - computes their system modulus $n=p \cdot q$
 - note $\phi(n)=(p-1)(q-1)$
- **selecting at random the encryption key e**
 - where $1 < e < \phi(n)$, $\gcd(e, \phi(n)) = 1$
- solve following equation to find decryption key d
 - **$e \cdot d \equiv 1 \pmod{\phi(n)}$ and $0 < d < n$**
- publish their public encryption key: PU={e, n} keep secret
- private decryption key: PR={d, n}



Why RSA Works (Mathematics behind RSA)

Because of Euler's Theorem:

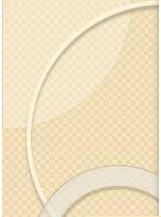
- o $a^{0(n)} \text{mod } n = 1$ where $\text{gcd}(a,n)=1$

In RSA have:

- o $n=p \cdot q$, where p and q are primes
- o $\phi(n) = (p-1)(q-1)$
- o carefully chose e & d to be inverses mod $\phi(n)$
- o hence $e \cdot d = 1 \text{ mod } \phi(n)$

or, $e \cdot d = k \cdot \phi(n) + 1$, for some k Hence, we can write

$$\begin{aligned} C^d \text{ mod } n &= M^{ed} \text{ mod } n \\ &= M^{k * \phi(n) + 1} \text{ mod } n = M \cdot M^{k - \phi(n)} \text{ mod } n \\ &= M \cdot (M^{\phi(n)})^k \text{ mod } n = M \text{ mod } n = M \end{aligned}$$



RSA Example - Key Setup

1. Select primes: $p=17$ & $q=11$
2. Calculate $n = p \cdot q = 17 \times 11 = 187$
3. Calculate $\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e : $\gcd(e, 160)=1$; choose $e=7$
5. Determine d : $de \equiv 1 \pmod{160}$ and $d < 160$ Value is $d=23$ since $23 \times 7 = 161 = 10 \times 160 + 1$
6. Publish public key $PU=\{7, 187\}$
7. Keep secret private key $PR=\{23, 187\}$



RSA Example - En/Decryption

- sample RSA encryption/decryption is:
- given message $M = 88$ (note: $88 < 187$)
- encryption:

$$C = 88^7 \bmod 187 = 11$$

- decryption:

$$M = 11^{23} \bmod 187 = 88$$



Efficient Encryption

- Encryption uses exponentiation to power e
- Hence if e small, this will be faster
 - often choose $e=65537$ ($2^{16}-1$)
 - also see choices of $e=3$ or $e=17$
- But if e too small (eg $e=3$) can be attacked



RSA Key Generation

- **Users of RSA must:**
 - **determine two primes at random - p, q**
 - **select either e or d and compute the other**
- **Primes p,q must not be easily derived from $n=p.q$**
 - **means must be sufficiently large**
 - **typically guess and use probabilistic test**
- **Exponents e, d are inverses, so use Inverse algorithm to compute the other**



Exercises

In a public-key system using RSA, you intercept the ciphertext $C = 10$ sent to a user whose public key is $e = 5$, $n = 35$. What is the plaintext M ?

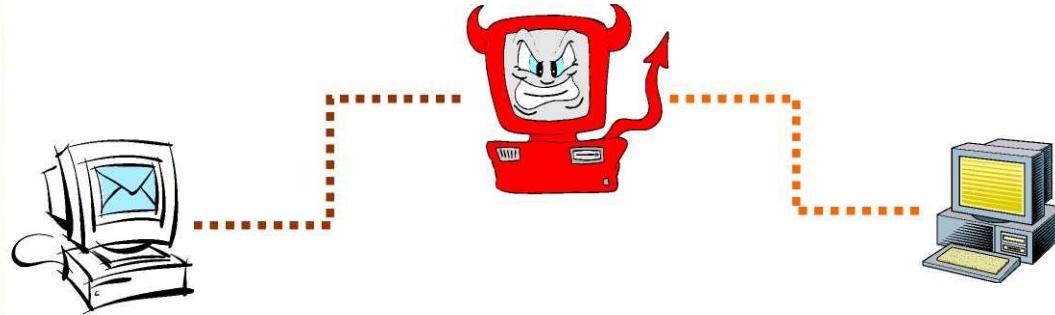
In the RSA public-key encryption scheme, each user has a public key, e , and a private key, d . Suppose Bob leaks his private key. Rather than generating a new modulus, he decides to generate a new public and a new private key. Is this safe?



Hashing

(MACs, hashes, and signatures)

Data Integrity and Source Authentication



- Encryption does not protect data from modification by another party. •
Why?
- Need a way to ensure that data arrives at destination in its original form as sent by the sender and it is coming from an authenticated source.



Message authentication codes (MACs)

- Encryption helps prevent an unauthorized individual from reading a message, **but it does not prevent that individual from tampering with the message.**
- An altered message, even if the alteration results in nothing but nonsense, can have real costs.
- A message authentication code (MAC) helps prevent message tampering.
- **Digital signatures are the public key equivalent of private key message authentication codes (MACs).**
- Although MACs use private keys to enable a message recipient to verify that a message has not been altered during transmission, **signatures use a private/public key pair.**

Source: <https://learn.microsoft.com/en-us/windows/uwp/security/macshashes-and-signatures>



Hashes

- A cryptographic hash function takes an arbitrarily long block of data and returns a fixed-size bit string.
- Hash functions are typically used when signing data. Because most public key signature operations are computationally intensive, it is typically more efficient to sign (encrypt) a message hash than it is to sign the original message.

Source: <https://learn.microsoft.com/en-us/windows/uwp/security/macshashes-and-signatures>



Digital signatures

- Digital signatures are the public key equivalent of private key message authentication codes (MACs).
- Whereas MACs use private keys to enable a message recipient to verify that a message has not been altered during transmission, signatures use a private/public key pair.
- Because most public key signature operations are computationally intensive, however, it is typically more efficient to sign (encrypt) a message hash than it is to sign the original message.
The sender creates a message hash, signs it, and sends both the signature and the (unencrypted) message. The recipient calculates a hash over the message, decrypts the signature, and compares the decrypted signature to the hash value. If they match, the recipient can be fairly certain that the message did, in fact, come from the sender and was not altered during transmission.

Source: <https://learn.microsoft.com/en-us/windows/uwp/security/macshashes-and-signatures>



Hash Functions

Condensed form of arbitrary message to fixed size, i.e.,

$$h = H(M)$$

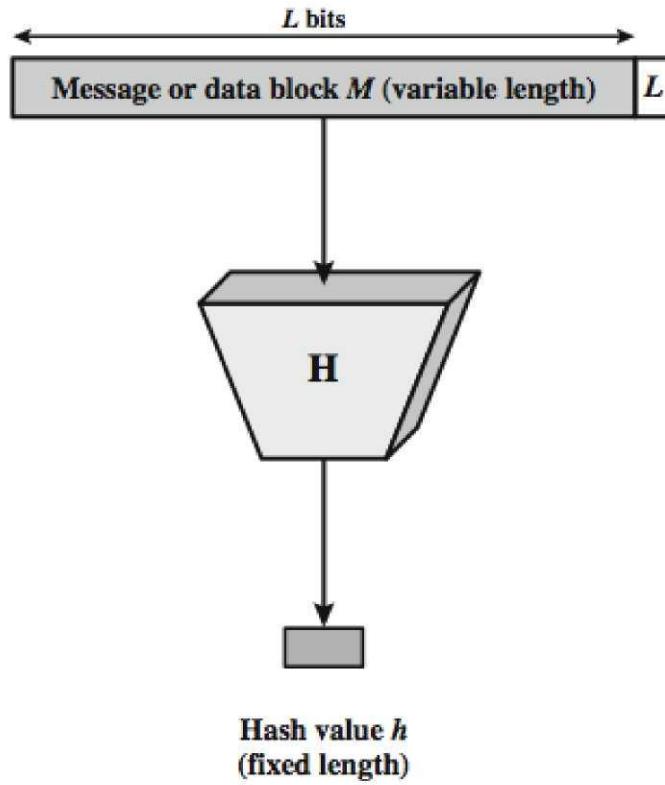
Usually assume hash function is public Hash is used to detect changes to message Objective is to form a

cryptographic hash function

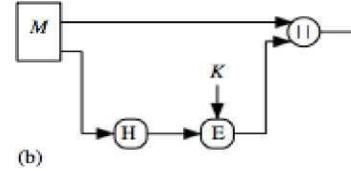
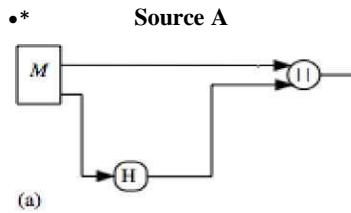
- computationally infeasible to find data mapping to specific hash (one-way property)
- computationally infeasible to find two data to same hash (collision-free property)



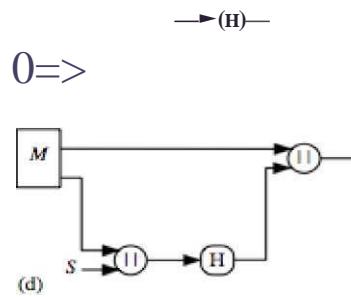
Cryptographic Hash Function

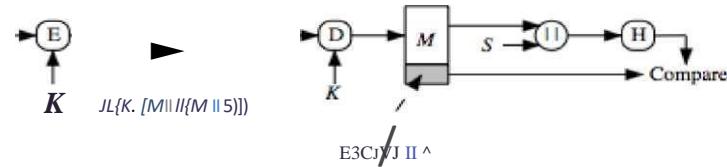
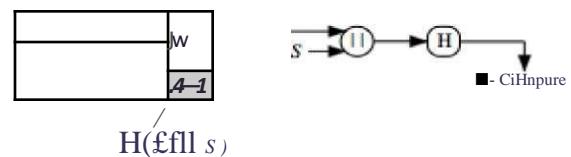
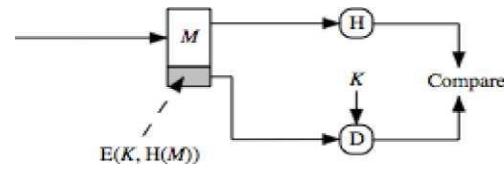
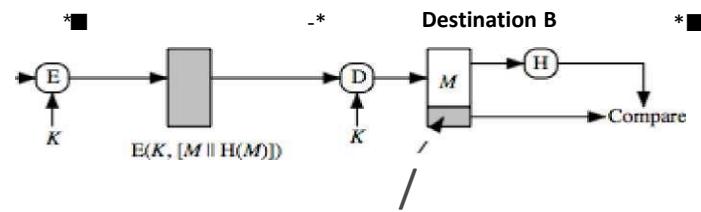


Hash Functions Message Authentication

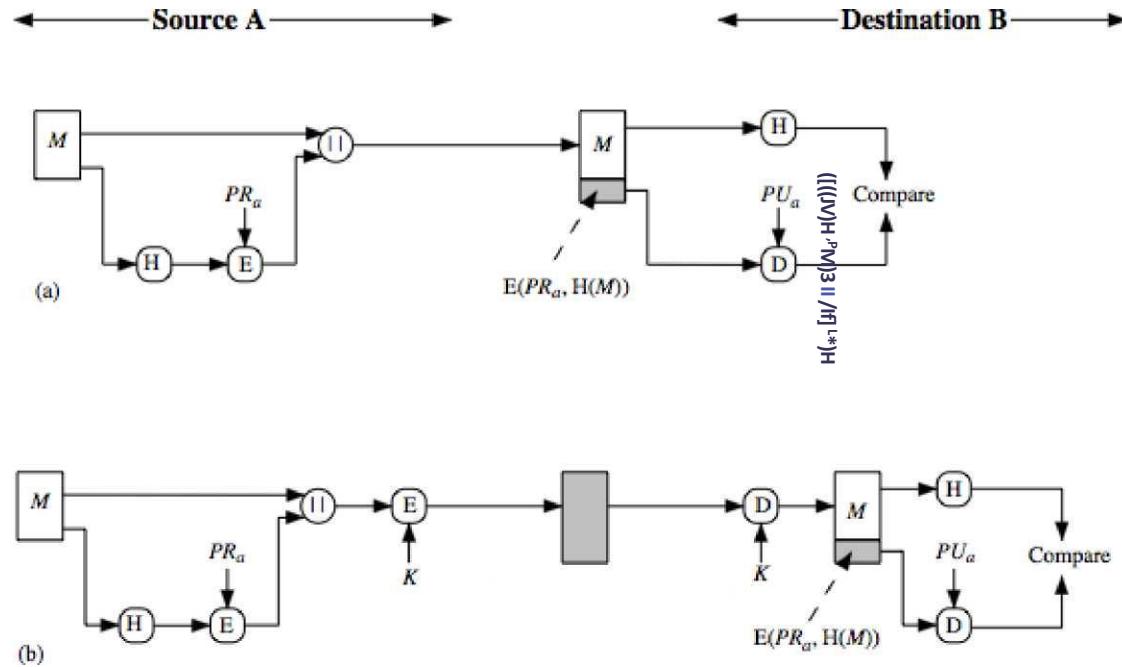


to 50-





Hash Functions & Digital Signatures





Other Hash Function Uses

- To create a one-way password file
 - store hash of password not actual password
- For intrusion detection and virus detection
 - keep & check hash of files on system



Two Simple Insecure Hash Functions

- Consider two simple insecure hash functions
- **bit-by-bit exclusive-OR (XOR) of every block**
 - $C_I = b_{i1} \text{ xor } b_{i2} \text{ xor } \dots \text{ xor } b_{im}$
 - **a longitudinal redundancy check**
 - reasonably effective as data integrity check
- **one-bit circular shift on hash value**
 - Initially set the n-bit hash value to zero
 - for each successive *n-bit* block
 - **rotate current hash value to left by 1 bit and XOR the block**
 - good for data integrity but useless for security



Secure Hash Algorithm

- SHA originally designed by NIST & NSA in 1993 and revised in 1995 as SHA-1.
- SHA-1 produces 160-bit hash values .
- In 2005, a research team described an attack in which two separate messages could be found that deliver the same SHA-1 hash using 2^{69} operation.
- Security of SHA-1 have raised concerns on its use in future applications



Revised Secure Hash Standard

NIST produced a revised version of the standard, and defined three new versions of SHA.

- SHA-256, SHA-384, SHA-512
- Collectively, these hash algorithms are known as SHA-2.
- Structure & detail is similar to SHA-1 and hence analysis should be similar but security levels are rather higher

SHA Versions

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message digest size	160	224	256	384	512
Message size	< 2 ⁶⁴	< 2 ⁶⁴	< 2 ⁶⁴	< 2 ¹²⁸	< 2 ¹²⁸
Block size	512	512	512	1024	1024
Word size	32	32	32	64	64
Number of steps	80	64	64	80	80



SHA-256

<https://sectigostore.com/blog/sha-256-algorithm-explained-by-a-cyber-security-consultant/>



Hash Functions

A hash function maps a message of an arbitrary length to a m-bit output

- output known as the **fingerprint** or the **message digest**

- What is an example of hash functions?
 - Give a hash function that maps Strings to integers in $[0, 2^A \{32\} - 1]$
- Cryptographic hash functions are hash functions with additional security requirements



Hash Family

- A hash family is a four-tuple (X, Y, K, H) , where
 - X is a set of possible messages
 - Y is a finite set of possible message digests
 - K is the keyspace
 - For each $K \in K$, there is a hash function $h_K : X \rightarrow Y$. Each $h_K : X \rightarrow Y$
- Alternatively, one can think of H as a function
$$K \times X \rightarrow Y$$

Key Concepts of Blockchain Technology

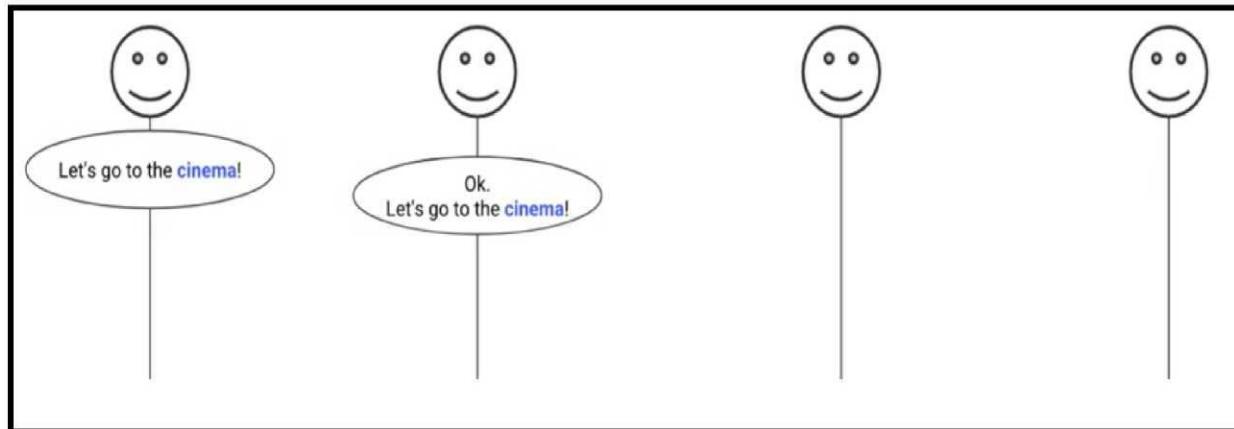
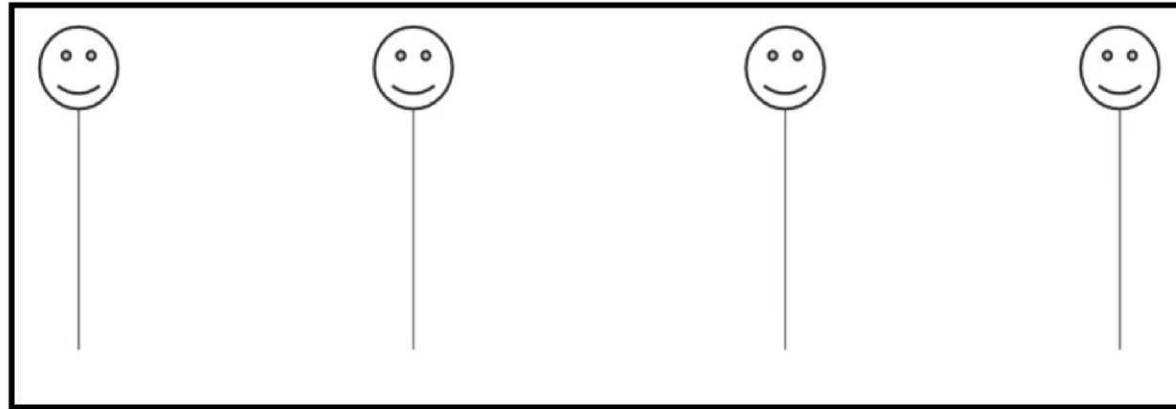
1. Peer-to-peer network
2. Public Key

Cryptography

3. Distributed Consensus

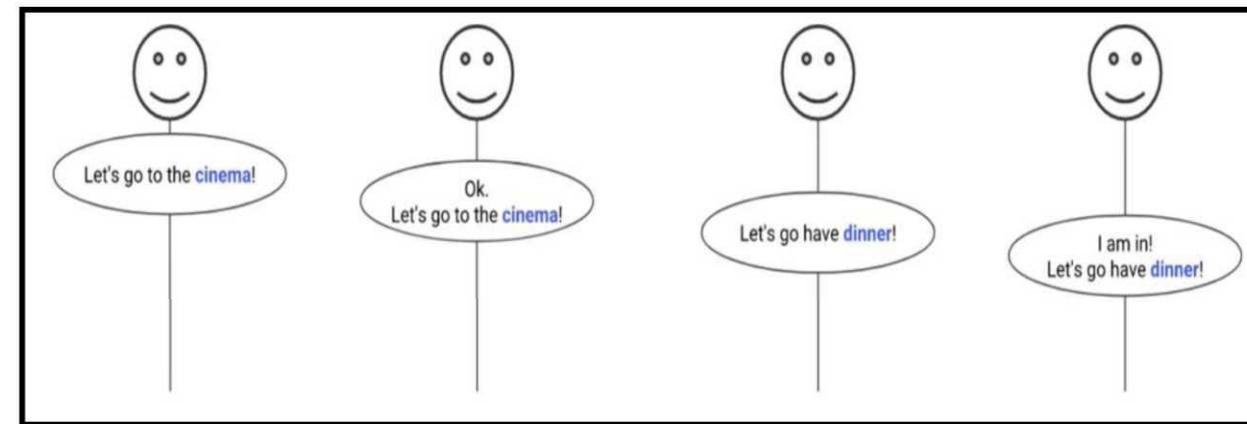
Distributed Consensus

What is to reach a consensus?



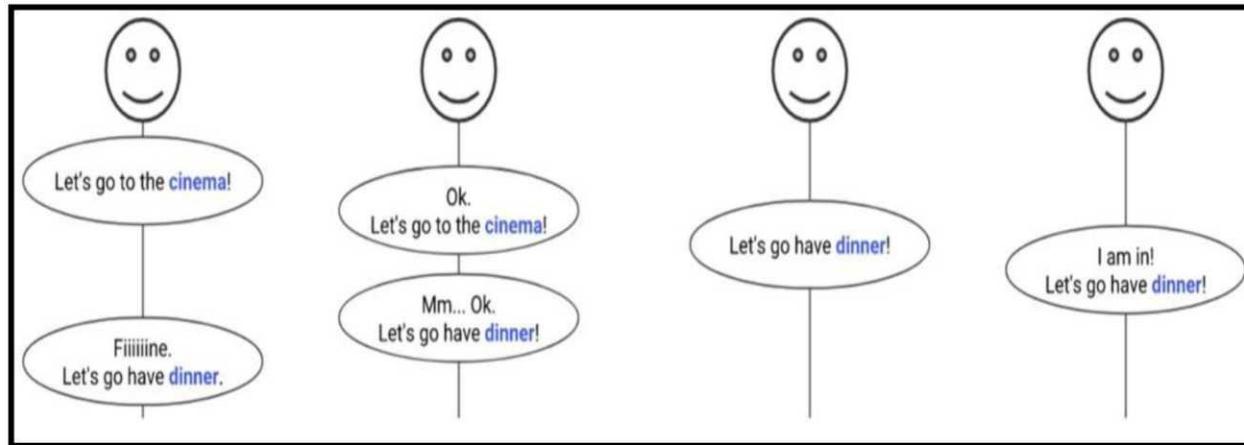
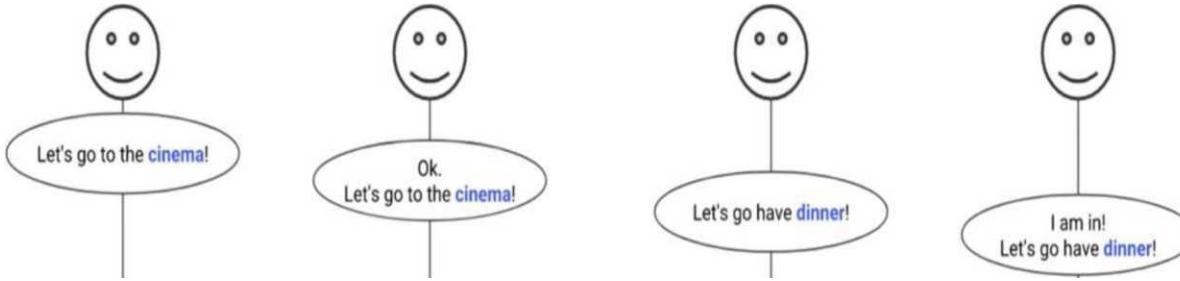
Distributed Consensus

What is to reach a consensus?



Distributed Consensus

What is to reach a consensus?



Distributed Consensus

What is to reach a consensus with Paxos?

Consensus is agreeing on **one** result.

Once a **majority** agrees on a proposal, that is the consensus.

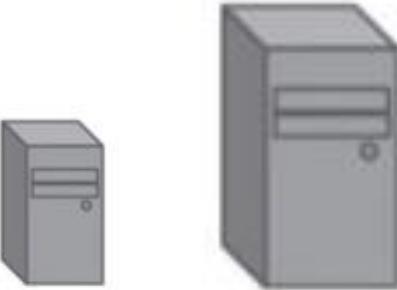
The reached consensus can be **eventually** known by everyone.

The involved parties want to agree on **any** result, not on their proposal.

Communication channels may be **faulty**, i.e., messages can get lost.

Distributed Consensus

Why do systems need to reach a consensus?



Infinitely powerful
and scalable single
computer



Infinitely powerful
and scalable single
computer

Distributed Consensus

Why do systems need to reach a consensus?

- There doesn't exist infinitely powerful and scalable computer.
- Choose a different option
 - Leader-Replicas Schema (Centralized System)
 - Peer-to-Peer Schema

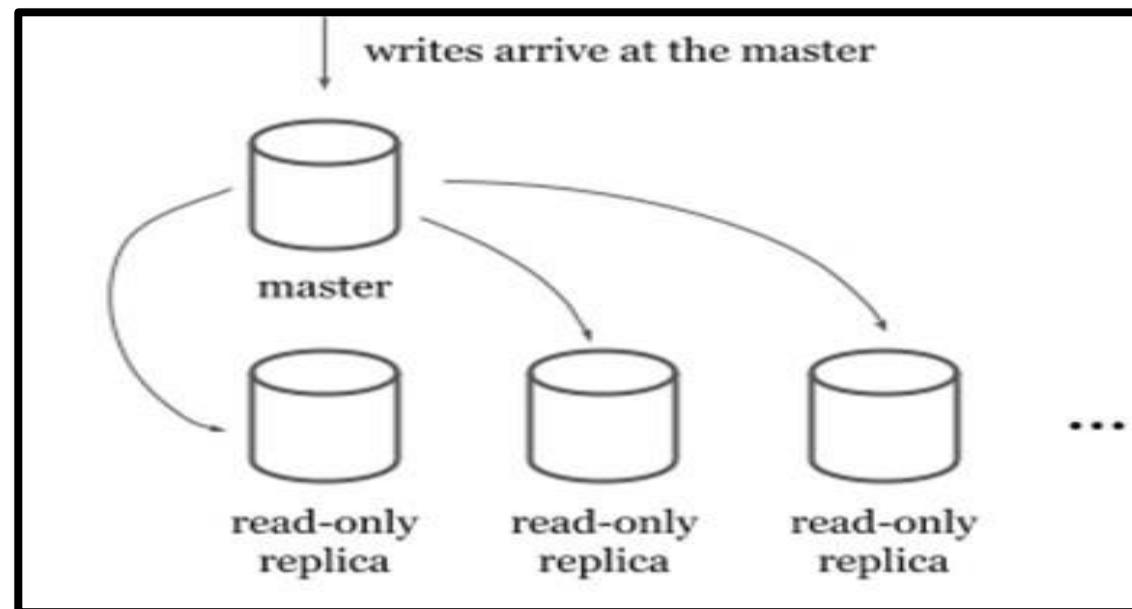
Distributed Consensus

Why do systems need to reach a consensus?

- Leader-Replicas Scheme

- A node wants to update a value, talk to leader.
- The leader will serialize the different proposals.
- The leader sends all replicas to all nodes

- Example-Bank Transactions

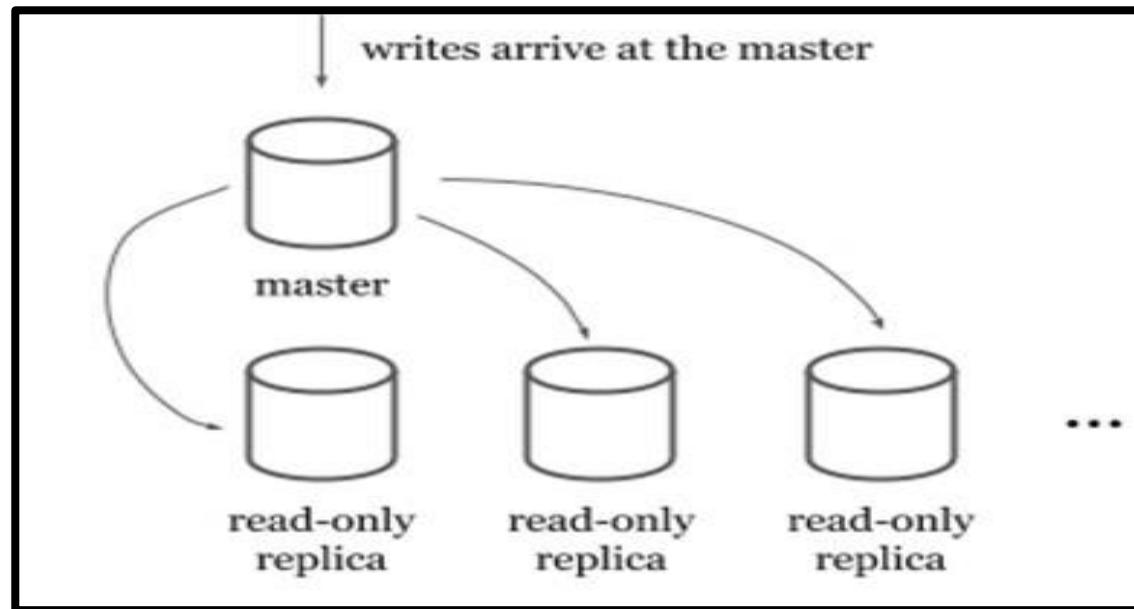


Distributed Consensus

Why do systems need to reach a consensus?

Leader-Replicas Scheme

- Single Point failure.
- If the leader becomes unavailable, nodes must reach a **consensus** to elect a new leader.

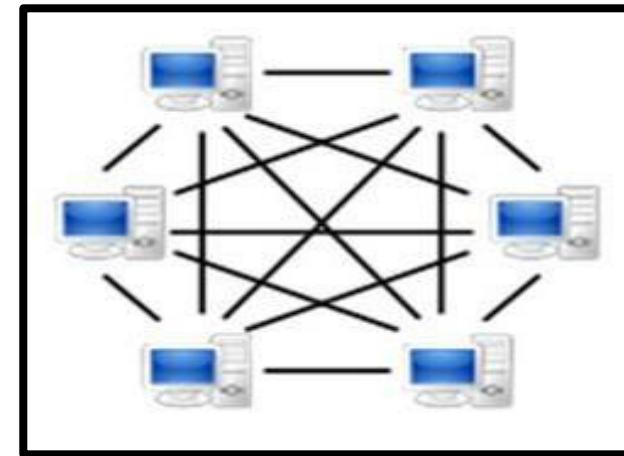


Distributed Consensus

Why do systems need to reach a consensus?

- Peer-to-Peer Schema
 - All nodes are the same and behave the same.
 - If a node wants to update a value, they send a proposal to everyone. A consensus mechanism is required.
- Example-Blockchain Transactions, BitTorrent

- The nodes need to reach consensus continuously so as to guarantee consistency.



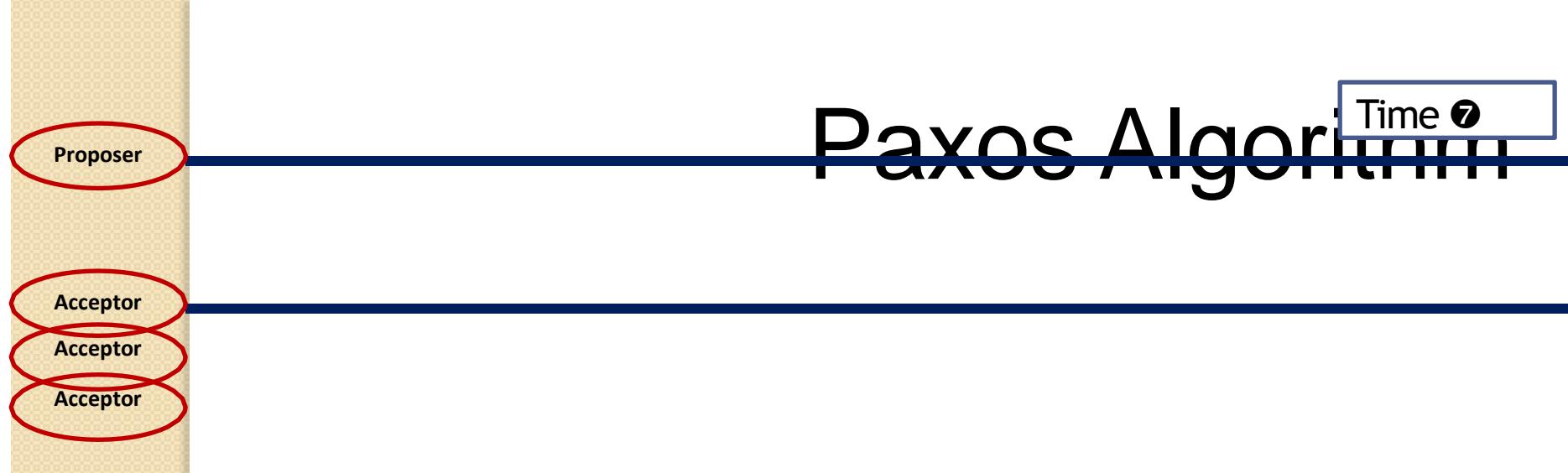
- In both cases (P2P and leader-replica), a consensus mechanism is required.

Algorithm- Basics

- Paxos defines three roles-
 - Proposers— Propose values to reach a consensus on.
 - Acceptors-Contribute to reach the consensus itself.
 - Learners- Learn the agreed-upon value.
- Paxos nodes can take multiple roles, even all of them.
- Paxos nodes must know “how many accepters a majority is”

- Paxos nodes must be persistent. They can’t forget what they accepted.
- Communication channels may be faulty.
- A Paxos run aims at reaching a single consensus.
- Once a consensus is reached, it cannot progress to consensus on another value in the same run.
- In order to reach another consensus, a different Paxos run must happen.

Paxos Algorithm



PROPOSER wants to propose a certain VALUE:

It sends PREPARE ID_p to a majority (or all) of ACCEPTORS. ID_p must be unique, e.g., slotted timestamp in nanoseconds.

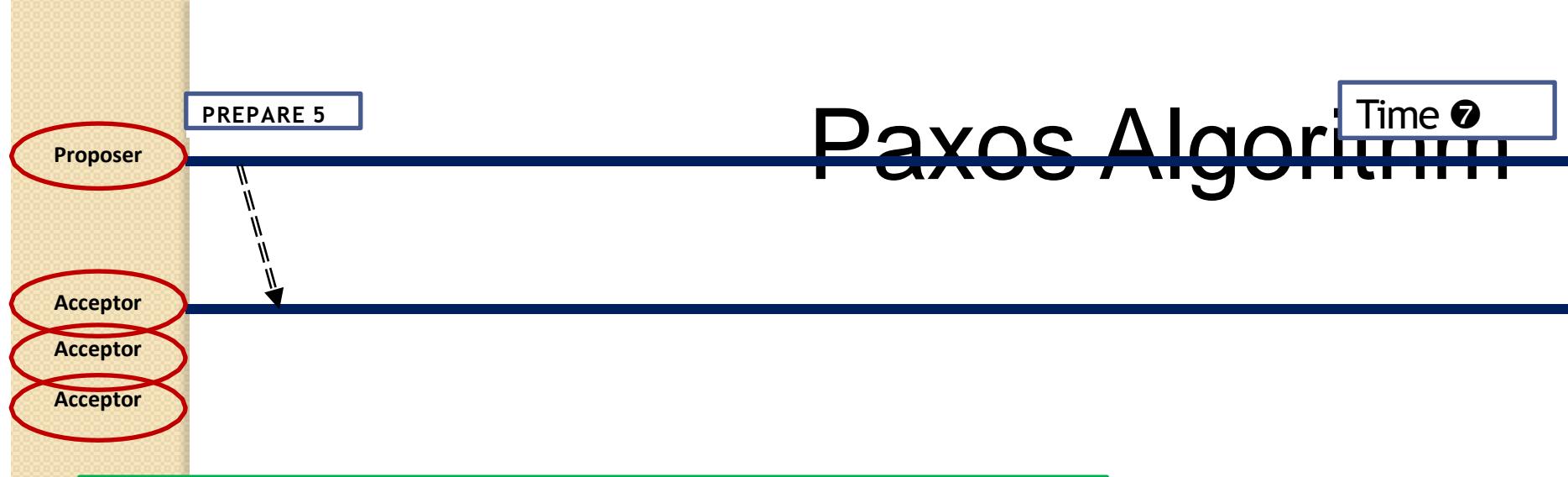
Example- PROPOSER 1 chooses IDs 1, 3, 5,, etc.

PROPOSER 2 chooses IDs 2, 4, 6, etc.

Timeout? Retry with a new (higher) ID_p.

No ID is used twice.
In the case of a time out,
the proposer will try with
new higher ID.

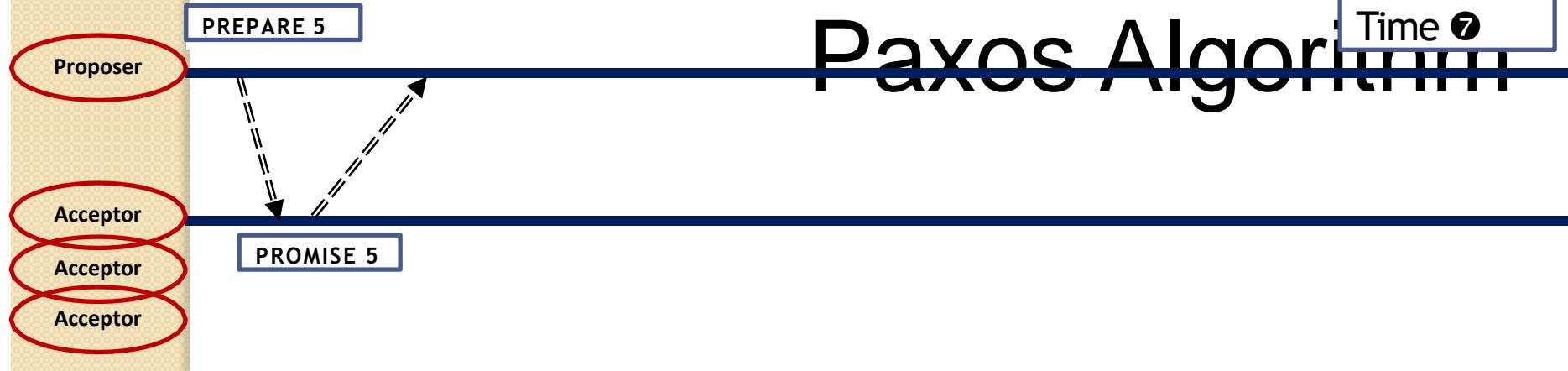
Paxos Algorithm



PROPOSER wants to propose a certain VALUE:
It sends PREPARE ID_p to a majority (or all) of ACCEPTORs. ID_p must be unique, e.g., slotted timestamp in nanoseconds.
Example- PROPOSER 1 chooses IDs 1, 3, 5,, etc.
PROPOSER 2 chooses IDs 2, 4, 6, etc.
Timeout? Retry with a new (higher) ID_p.

No ID is used twice.
In the case of a time out,
the proposer will try with
new higher ID.

Paxos Algorithm



PROPOSER wants to propose a certain VALUE:

It sends PREPARE ID_p to a majority (or all) of ACCEPTORS. ID_p must be unique, e.g., slotted timestamp in nanoseconds.

Example- PROPOSER 1 chooses IDs 1, 3, 5,, etc.

PROPOSER 2 chooses IDs 2, 4, 6, etc.

Timeout? Retry with a new (higher) ID_p.

No ID is used twice.
In the case of a time out,
the proposer will try with
new higher ID.

ACCEPTOR receives a PREPARE message for ID_p.

Did it promise to ignore requests with this ID_p?

Yes ⑦ Ignore

No ⑦ It will promise to ignore any request lower than ID_p.

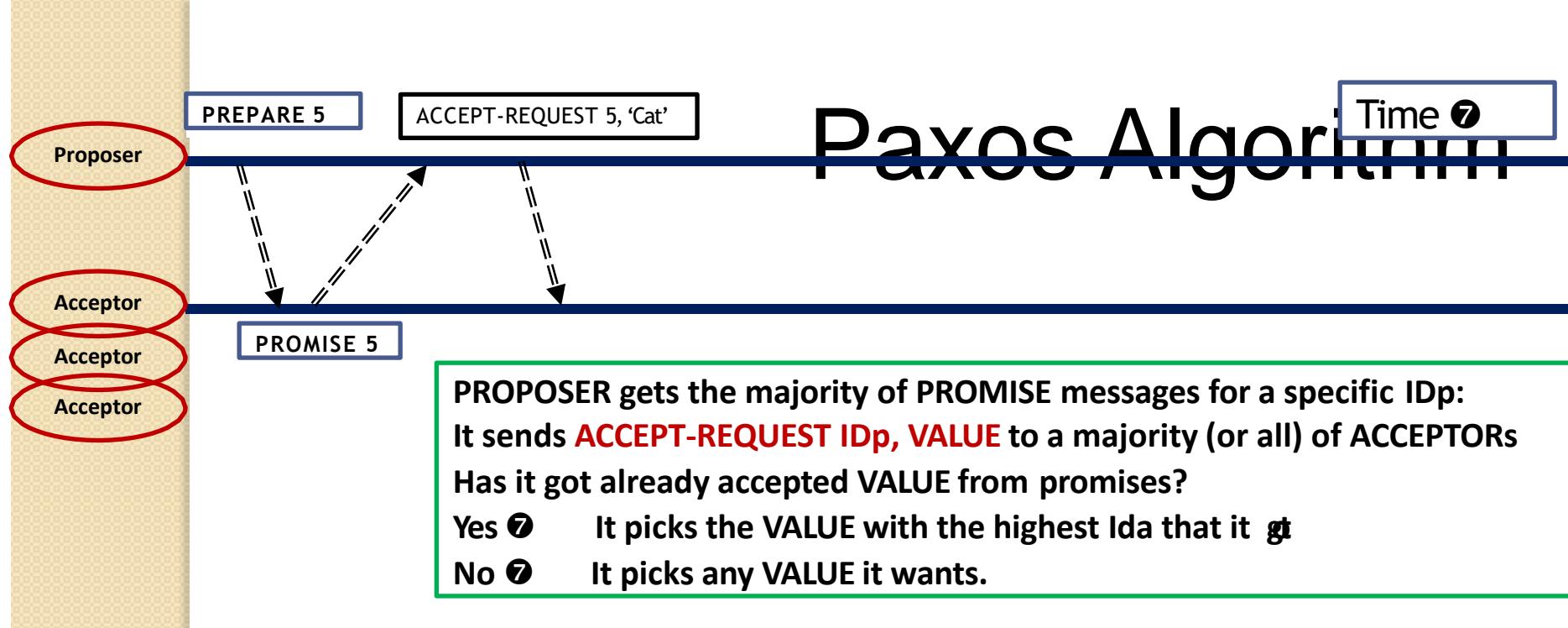
Has it ever accepted anything? (Assume accepted ID = ID_a)

Yes ⑦ Reply with PROMISE ID Accepted ID_a, V_f

No ⑦ Reply with PROMISE ID

If a majority of ACCEPTORS promise, no ID < ID_p can make it through.

Paxos Algorithm

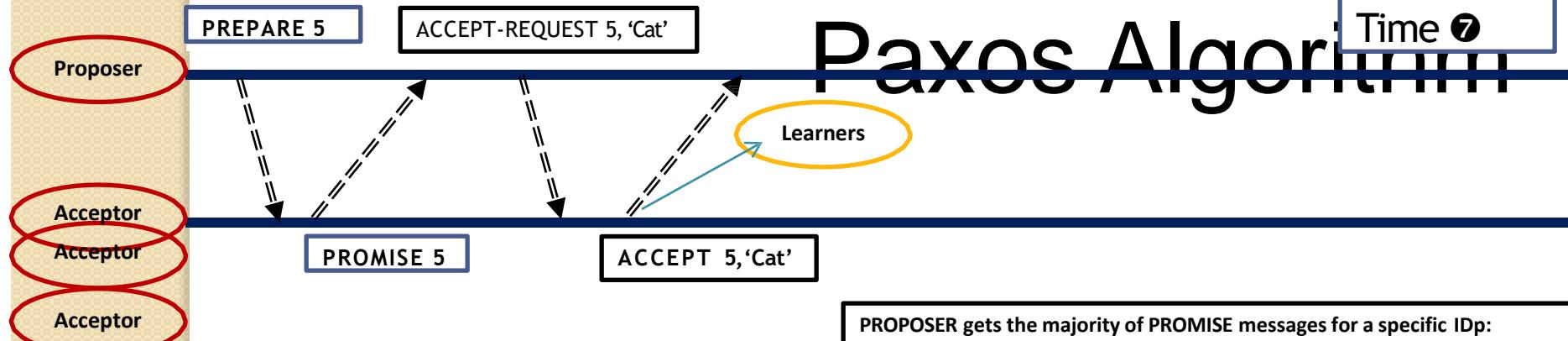


PROPOSER wants to propose a certain **VALUE**:
It sends **PREPARE ID_p** to a majority (or all) of **ACCEPTORs**.
ID_p must be unique, e.g., slotted timestamp in nanoseconds.
Example- PROPOSER 1 chooses IDs 1, 3, 5,, etc.
PROPOSER 2 chooses IDs 2, 4, 6, etc.
Timeout? Retry with a new (higher) **ID_p**.

ACCEPTOR receives a **PREPARE** message for **ID_p**.
Did it promise to ignore requests with this **ID_p**?
Yes ⑦ Ignore
No ⑦ It will promise to ignore any request lower than **ID_p**
Has it ever accepted anything? (Assume accepted **ID** = **ID_a**)
Yes ⑦ Reply with **PROMISE ID Accepted ID_a, VALUE**
No ⑦ Reply with **PROMISE ID**

Paxos Algorithm

Time 7



PROPOSER wants to propose a certain VALUE:

It sends PREPARE ID_p to a majority (or all) of ACCEPTORS.

ID_p must be unique, e.g., slotted timestamp in nanoseconds.

e.g. PROPOSER 1 chooses IDs 1, 3, 5,, etc.

PROPOSER 2 chooses IDs 2, 4, 6,, etc.

Timeout? Retry with a new (higher) ID_p.

PROPOSER gets the majority of PROMISE messages for a specific ID_p:
 It sends ACCEPT-REQUEST ID_p, VALUE to a majority (or all) of ACCEPTORS
 Has it got already accepted VALUE from promises?

Yes 7 It picks the VALUE with the highest ID_a that it got.

No 7 It picks any VALUE it wants.

ACCEPTOR receives a PREPARE message for ID_p.

Did it promise to ignore requests with this ID_p?

Yes 7 Ignore

No 7 It will promise to ignore any request lower than ID_p

Has it ever accepted anything? (Assume accepted ID = ID_a)

Yes 7 Reply with PROMISE ID_p Accepted ID_a, VALUE

No 7 Reply with PROMISE ID_p

If a majority of ACCEPTORS promise, no ID < ID_p can make it through.

ACCEPTOR receives an ACCEPT REQUEST ID_p, VALUE:

Did it promise to ignore requests with this ID_p?

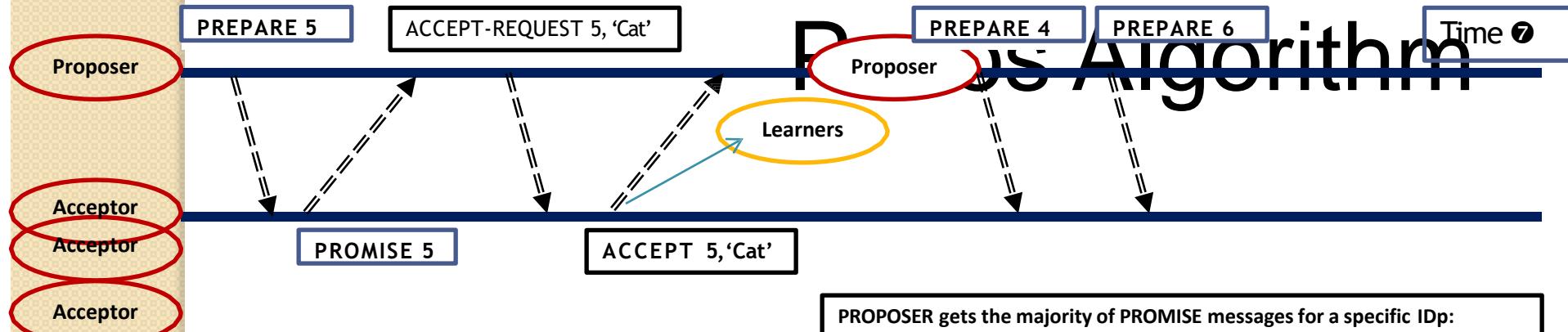
Yes 7 Ignore

No 7 Reply with ACCEPT ID_p, VALUE. Also, send it to all LEARNERS.

If a majority of ACCEPTORS accept ID_p, VALUE, the CONSENSUS is reached. CONSENSUS is and will be on VALUE (not necessarily ID_p).

PROPOSER or LEARNER gets ACCEPT ID_p, VALUE:

If a PROPOSER/LEARNER gets the majority of acceptance for a specific ID_p, they know that CONSENSUS has been reached on VALUE (not ID_p).



PROPOSER wants to propose a certain **VALUE**:
 It sends **PREPARE ID_p** to a majority (or all) of **ACCEPTORs**.
 ID_p must be unique, e.g., slotted timestamp in nanoseconds.
 e.g. PROPOSER 1 chooses IDs 1, 3, 5,, etc.
 PROPOSER 2 chooses IDs 2, 4, 6, etc.
 Timeout? Retry with a new (higher) ID_p.

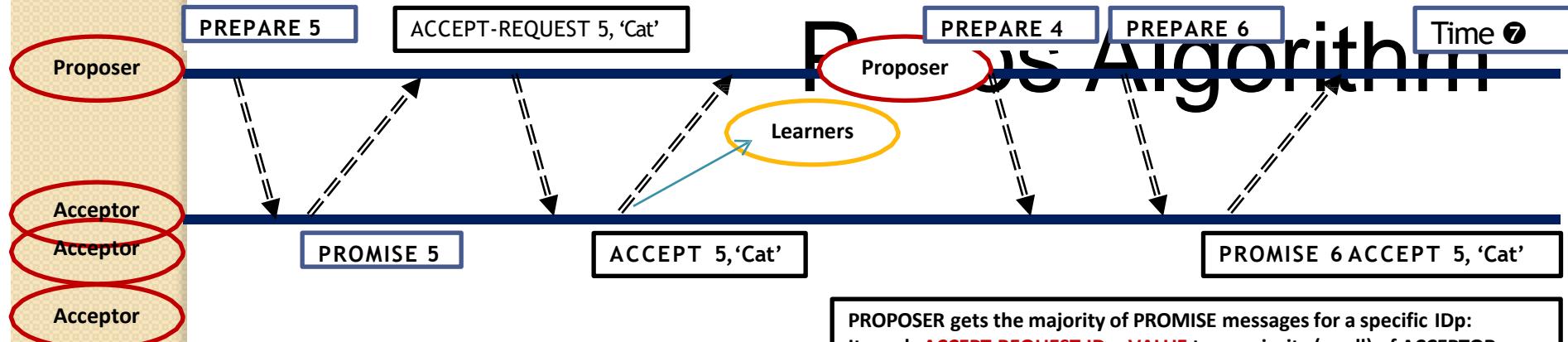
PROPOSER gets the majority of **PROMISE** messages for a specific ID_p:
 It sends **ACCEPT-REQUEST ID_p, VALUE** to a majority (or all) of **ACCEPTORs**
 Has it got already accepted **VALUE** from promises?
 Yes ⑦ It picks the **VALUE** with the highest ID_a that it got.
 No ⑦ It picks any **VALUE** it wants.

ACCEPTOR receives a **PREPARE** message for ID_p.
 Did it promise to ignore requests with this ID_p?
 Yes ⑦ Ignore
 No ⑦ It will promise to ignore any request lower than ID_p
 Has it ever accepted anything? (Assume accepted ID = ID_a)
 Yes ⑦ Reply with **PROMISE ID_p ACCEPTED ID_a, VALUE**
 No ⑦ Reply with **PROMISE ID_p**
If a majority of ACCEPTORs promise, no ID < ID_p can make it through.

ACCEPTOR receives an **ACCEPT-REQUEST** message for ID_p, **VALUE**:
 Did it promise to ignore requests with this ID_p?
 Yes ⑦ Ignore
 No ⑦ Reply with **ACCEPT ID_p, VALUE**. Also, send it to all **LEARNERS**.
If a majority of ACCEPTORs accept ID_p, VALUE, the CONSENSUS is reached. CONSENSUS is and will be on VALUE (not necessarily ID_p).

PROPOSER or **LEARNER** get **ACCEPT** message for ID_p, **VALUE**:
If a PROPOSER/LEARNER gets the majority of acceptance for a specific ID_p, they know that CONSENSUS has been reached on VALUE (not ID_p).

Let's introduce a second proposer who is unaware of any consensus and wants to propose a different value.



PROPOSER wants to propose a certain VALUE:
It sends PREPARE ID_p to a majority (or all) of ACCEPTORS.
 ID_p must be unique, e.g., slotted timestamp in nanoseconds.
e.g. PROPOSER 1 chooses IDs 1, 3, 5,, etc.
PROPOSER 2 chooses IDs 2, 4, 6,, etc.
Timeout? Retry with a new (higher) ID_p .

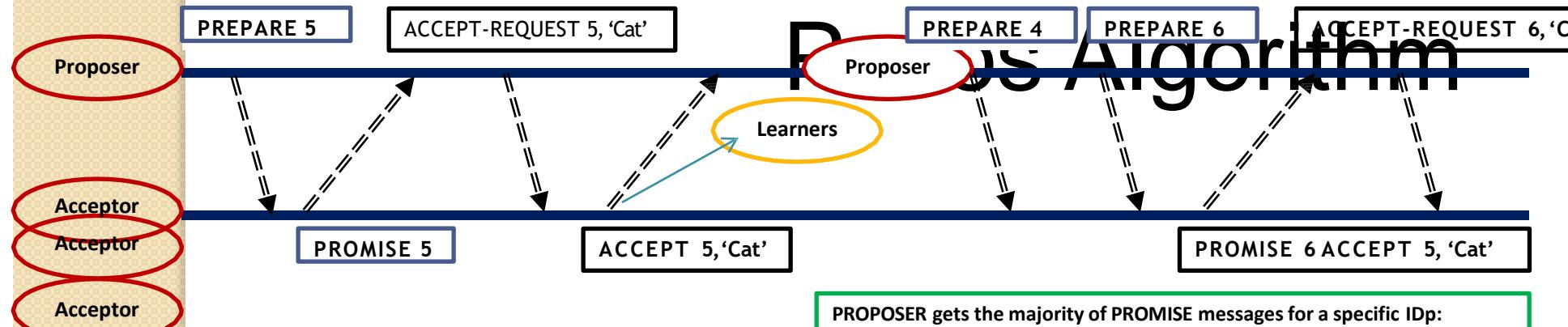
PROPOSER gets the majority of PROMISE messages for a specific ID_p :
It sends ACCEPT-REQUEST ID_p , VALUE to a majority (or all) of ACCEPTORS
Has it got already accepted VALUE from promises?
Yes ⑦ It picks the VALUE with the highest ID_a that it got.
No ⑦ It picks any VALUE it wants.

ACCEPTOR receives a PREPARE message for ID_p .
Did it promise to ignore requests with this ID_p ?
Yes ⑦ Ignore
No ⑦ It will promise to ignore any request lower than ID_p
Has it ever accepted anything? (Assume accepted $ID = ID_a$)
Yes ⑦ Reply with PROMISE ID_p ACCEPTED ID_a , VALUE
No ⑦ Reply with PROMISE ID_p
If a majority of ACCEPTORS promise, no $ID < ID_p$ can make it through.

ACCEPTOR receives an ACCEPT_REQUEST message for ID_p , VALUE:
Did it promise to ignore requests with this ID_p ?
Yes ⑦ Ignore
No ⑦ Reply with ACCEPT ID_p , VALUE. Also, send it to all LEARNERS.
If a majority of ACCEPTORS accept ID_p , VALUE, the CONSENSUS is reached. CONSENSUS is and will be on VALUE (not necessarily ID_p).

PROPOSER or LEARNER get ACCEPT message for ID_p , VALUE:
If a PROPOSER/LEARNER gets the majority of acceptance for a specific ID_p , they know that CONSENSUS has been reached on VALUE (not ID_p).

Let's introduce a second proposer who is unaware of any consensus and wants to propose a different value.



PROPOSER wants to propose a certain VALUE:
It sends PREPARE IDp to a majority (or all) of ACCEPTORS.
IDp must be unique, e.g., slotted timestamp in nanoseconds.
e.g. PROPOSER 1 chooses IDs 1, 3, 5,, etc.
PROPOSER 2 chooses IDs 2, 4, 6, etc.
Timeout? Retry with a new (higher) IDp.

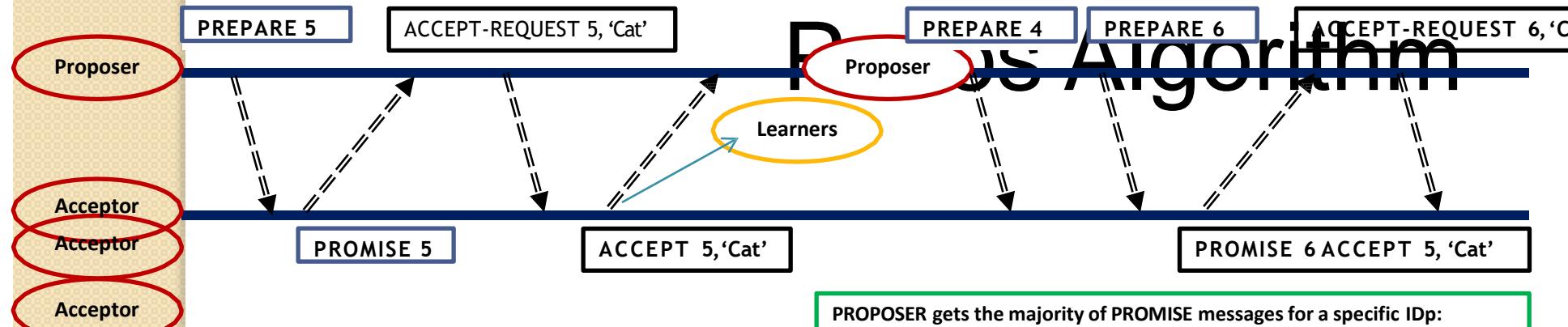
PROPOSER gets the majority of PROMISE messages for a specific IDp:
It sends ACCEPT-REQUEST IDp, VALUE to a majority (or all) of ACCEPTORS
Has it got already accepted VALUE from promises?
Yes ⑦ It picks the VALUE with the highest IDa that it got.
No ⑦ It picks any VALUE it wants.

ACCEPTOR receives a PREPARE message for IDp.
Did it promise to ignore requests with this IDp?
Yes ⑦ Ignore
No ⑦ It will promise to ignore any request lower than IDp
Has it ever accepted anything? (Assume accepted ID = IDa)
Yes ⑦ Reply with PROMISE IDp ACCEPTED IDa, VALUE
No ⑦ Reply with PROMISE IDp
If a majority of ACCEPTORS promise, no ID < IDp can make it through.

ACCEPTOR receives an ACCEPT_REQUEST message for IDp, VALUE:
Did it promise to ignore requests with this IDp?
Yes ⑦ Ignore
No ⑦ Reply with ACCEPT IDp, VALUE. Also, send it to all LEARNERS.
If a majority of ACCEPTORS accept IDp, VALUE, the CONSENSUS is reached. CONSENSUS is and will be on VALUE (not necessarily IDp).

PROPOSER or LEARNER get ACCEPT message for IDp, VALUE:
If a PROPOSER/LEARNER gets the majority of acceptance for a specific IDp, they know that CONSENSUS has been reached on VALUE (not IDp).

Let's introduce a second proposer who is unaware of any consensus and wants to propose a different value.



PROPOSER wants to propose a certain VALUE:

It sends PREPARE **ID_p** to a majority (or all) of ACCEPTORS.

ID_p must be unique, e.g., slotted timestamp in nanoseconds.

e.g. PROPOSER 1 chooses IDs 1, 3, 5,, etc.

PROPOSER 2 chooses IDs 2, 4, 6,, etc.

Timeout? Retry with a new (higher) ID_p.

PROPOSER gets the majority of PROMISE messages for a specific ID_p:
It sends ACCEPT-REQUEST **ID_p, VALUE** to a majority (or all) of ACCEPTORS
Has it got already accepted VALUE from promises?

Yes ⑦ It picks the VALUE with the highest ID_a that it got.

No ⑦ It picks any VALUE it wants.

ACCEPTOR receives a PREPARE message for ID_p.

Did it promise to ignore requests with this ID_p?

Yes ⑦ Ignore

No ⑦ It will promise to ignore any request lower than ID_p

Has it ever accepted anything? (Assume accepted ID = ID_a)

Yes ⑦ Reply with PROMISE ID_p ACCEPTED ID_a, VALUE

No ⑦ Reply with PROMISE ID_p

If a majority of ACCEPTORS promise, no ID < ID_p can make it through.

ACCEPTOR receives an ACCEPT_REQUEST message for ID_p, VALUE:

Did it promise to ignore requests with this ID_p?

Yes ⑦ Ignore

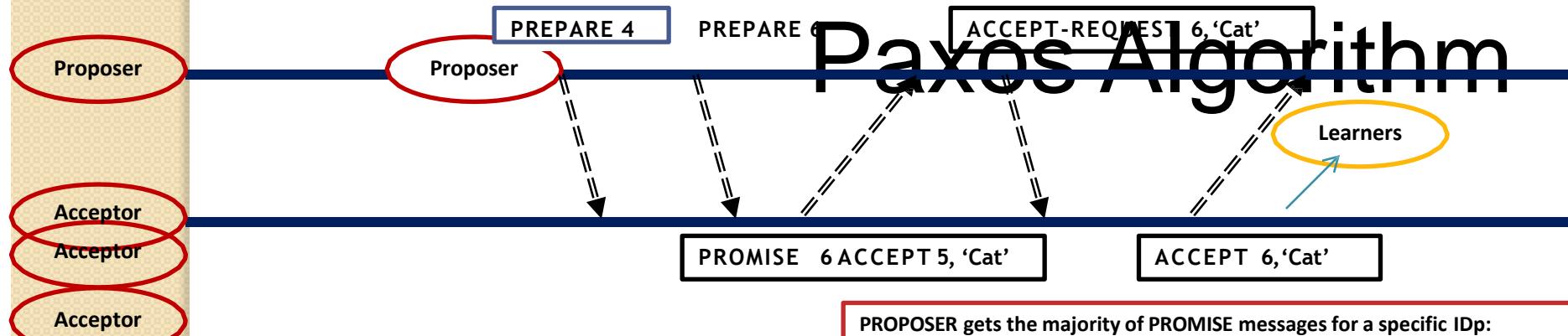
No ⑦ Reply with ACCEPT ID_p, VALUE. Also, send it to all LEARNERS.

If a majority of ACCEPTORS accept ID_p, VALUE, the CONSENSUS is reached. CONSENSUS is and will be on VALUE (not necessarily ID_p).

PROPOSER or LEARNER get ACCEPT message for ID_p, VALUE:

If a PROPOSER/LEARNER gets the majority of acceptance for a specific ID_p, they know that CONSENSUS has been reached on VALUE (not ID_p).

Let's introduce a second proposer who is unaware of any consensus and wants to propose a different value. **A consensus has been reached.**



PROPOSER wants to propose a certain VALUE:

It sends PREPARE ID_p to a majority (or all) of ACCEPTORS.

ID_p must be unique, e.g., slotted timestamp in nanoseconds.

e.g. PROPOSER 1 chooses IDs 1, 3, 5,, etc.

PROPOSER 2 chooses IDs 2, 4, 6,, etc.

Timeout? Retry with a new (higher) ID_p.

PROPOSER gets the majority of PROMISE messages for a specific ID_p:
It sends ACCEPT-REQUEST ID_p, VALUE to a majority (or all) of ACCEPTORS
Has it got already accepted VALUE from promises?

Yes ⑦ It picks the VALUE with the highest ID_a that it got.

No ⑦ It picks any VALUE it wants.

ACCEPTOR receives a PREPARE message for ID_p.

Did it promise to ignore requests with this ID_p?

Yes ⑦ Ignore

No ⑦ It will promise to ignore any request lower than ID_p

Has it ever accepted anything? (Assume accepted ID = ID_a)

Yes ⑦ Reply with PROMISE ID_p ACCEPTED ID_a, VALUE

No ⑦ Reply with PROMISE ID_p

If a majority of ACCEPTORS promise, no ID < ID_p can make it through.

ACCEPTOR receives an ACCEPT_REQUEST message for ID_p, VALUE:

Did it promise to ignore requests with this ID_p?

Yes ⑦ Ignore

No ⑦ Reply with ACCEPT ID_p, VALUE. Also, send it to all LEARNERS.

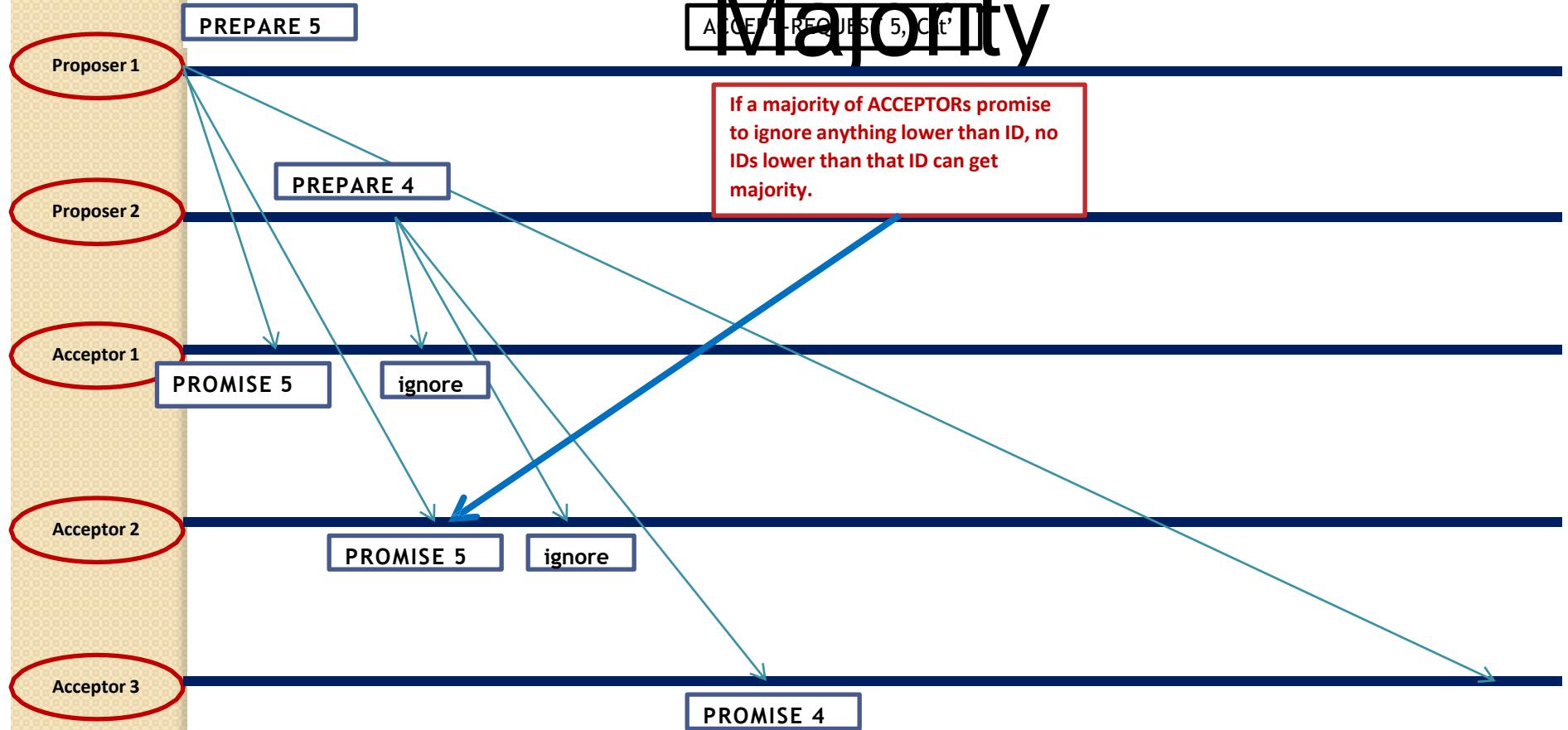
If a majority of ACCEPTORS accept ID_p, VALUE, the CONSENSUS is reached. CONSENSUS is and will be on VALUE (not necessarily ID_p).

PROPOSER or LEARNER get ACCEPT message for ID_p, VALUE:

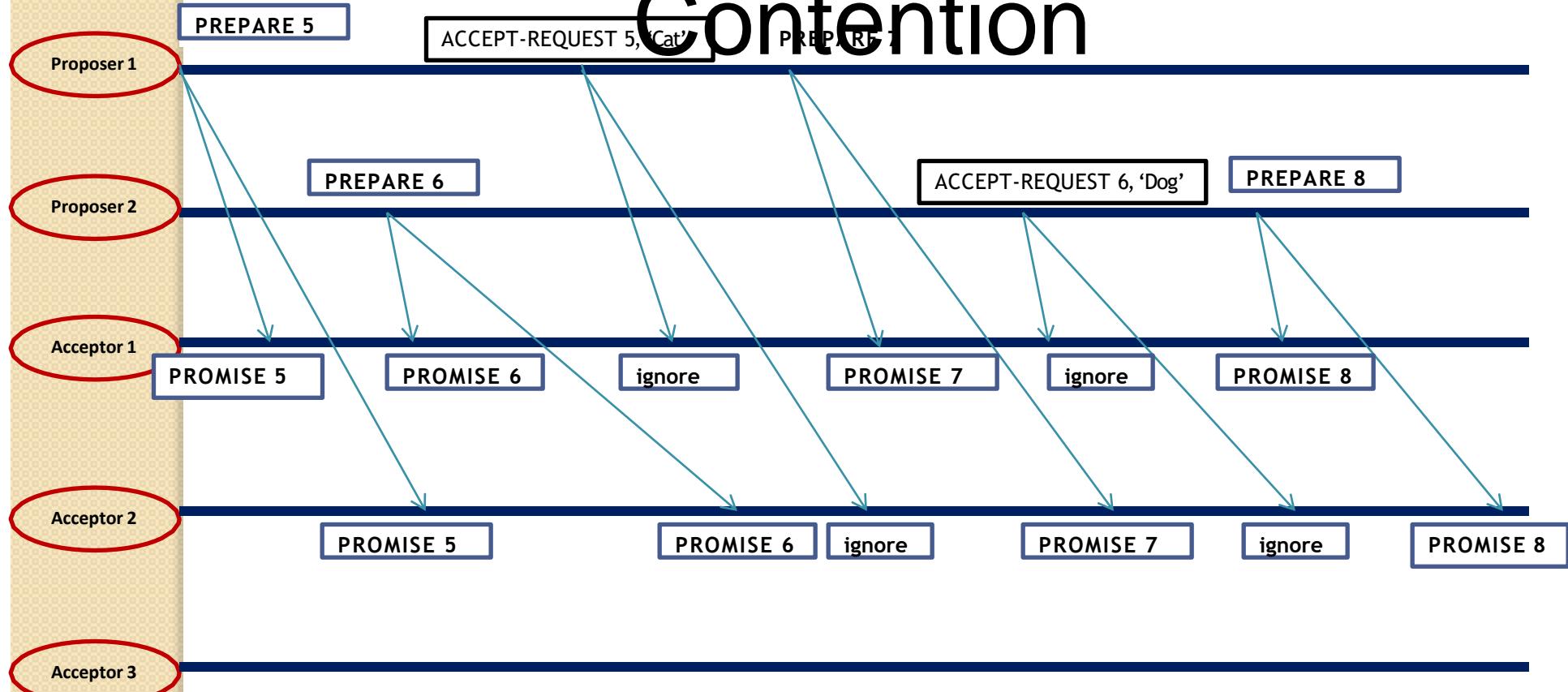
If a PROPOSER/LEARNER gets the majority of acceptance for a specific ID_p, they know that CONSENSUS has been reached on VALUE (not ID_p).

Let's introduce a second proposer who is unaware of any consensus and wants to propose a different value. **A consensus has been reached.**

Review. Majority



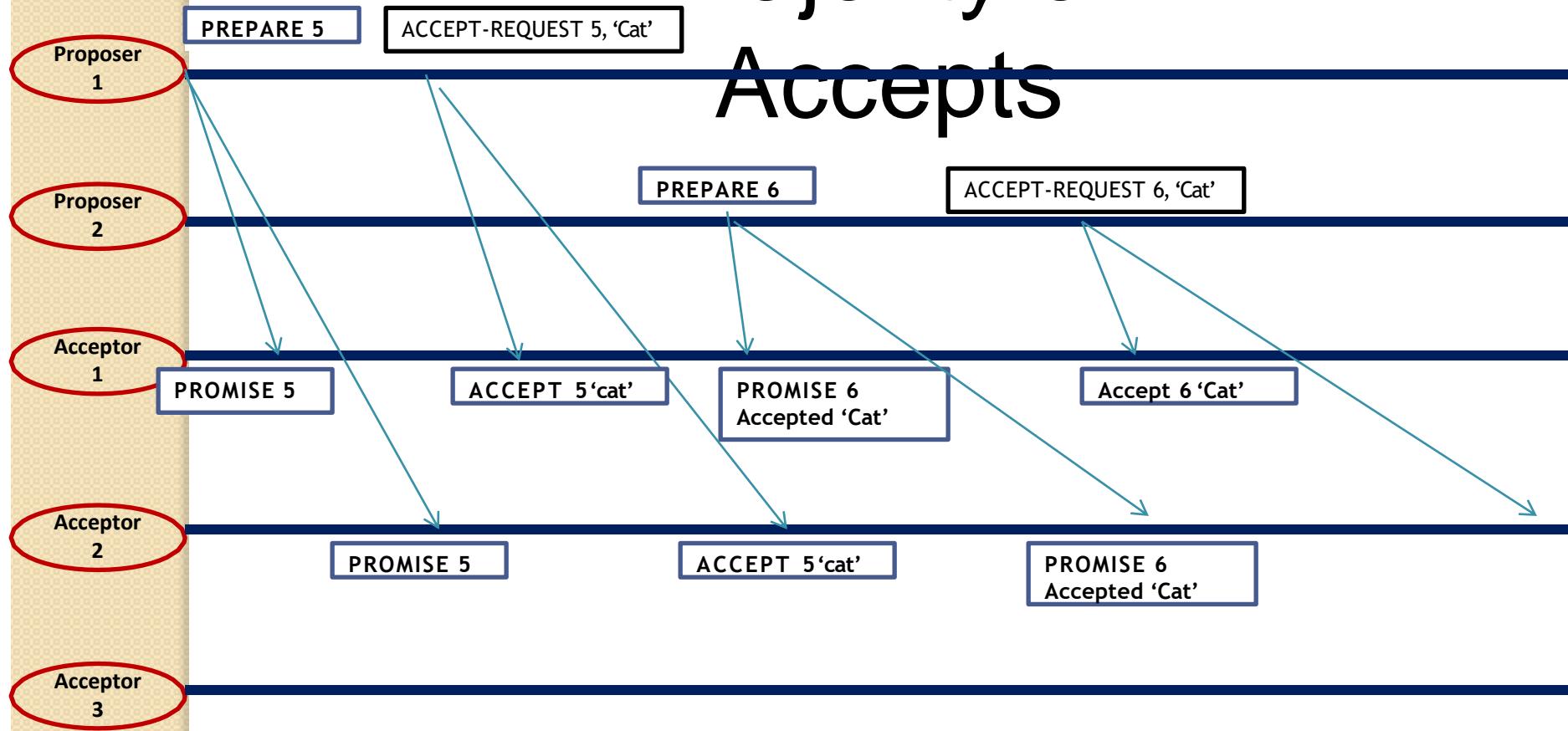
Review. Contention



Set Exponential Backoff

Truncated exponential backoff is a standard error-handling strategy for network applications. In this approach, a node (client) periodically retries a failed request with increasing delays between requests.

Majority of Accepts



Set Exponential Backoff

Truncated exponential backoff is a standard error-handling strategy for network applications. In this approach, a node (client) periodically retries a failed request with increasing delays between requests.

Video Link: https://www.youtube.com/watch?v=d7nAGI_NZPk

Components of Blockchain

1. Node
2. Ledger
3. Wallet
4. Nonce
5. Hash
6. Mining
7. Consensus protocol

Node

- An electronic device (computers, mobile devices, servers, etc.) that is connected to the Blockchain network through the internet.
 - Each node has a copy of the Blockchain ledger.
 - The primary goal is to maintain the integrity of the Blockchain.
- Full Node
 - Maintains a full copy of the transaction history of the Blockchain.
- Helps in mining
 - Accepting transactions/blocks □ Validating the block □ Broadcasting the block to the network
- Partial Node
 - Maintains a partial copy of the transaction history of the Blockchain.
 - Have only block headers
 - Rely on Full nodes

Ledger

- An immutable digital database of information or transactions.
 - Public
 - Any Blockchain node can access, and read the ledger.
 - Any Blockchain node can verify the transactions in the ledger.
 - Distributed-
 - Works based on the replication principle □ Each node has a copy of the Blockchain ledger.
 - Decentralized
 - No central control (No single point of failure)
 - Any node can validate the records.
 - Traditional DBMS works on the CURD principle.
 - Blockchain works on an Append-only principle.

Wallet

- Allows users to manage cryptocurrency.
- A person with Blockchain Wallet can perform the transactions of Cryptocurrency.
- When Person A sends cryptocurrency coins to Person B's wallet.
 - Person A transfers the ownership of the coin to Person B.
 - If the transaction is validated on the Blockchain, Person B's wallet balance increases, and Person A's wallet balance decreases accordingly.
 - New transaction is added in a new block of the Blockchain
- Features
 - Privacy- Private and public keys are created at the time of Wallet Creation.
 - Secured Transactions-Private key is used to send funds and encrypt messages.
 - Ease of Usage-Wallet are available to anyone easily.
 - Currency Conversion-Help to deal with various cryptocurrencies without worrying about currency conversion
 - Balance - Maintains Balance

Nonce

- It is used to create a block (mine the block)
- Nonce is a cryptographic number that is used once in cryptographic communication.
- It makes a block additionally unique.
- It allows miners to participate in the cryptographic puzzle challenge.
- In Bitcoin Blockchain-
 - The miners change the nonce and generate the hash value using SHA-256.
 - The nonce is a 32-bit unsigned integer. The range is 0 to 4294967295.

Hash

- A hash function can take data of any size, perform some operation on it, and returns a fixed-size “hash”.
- Hash is data of Fixed size and it is independent of the input data.
- Hash is unique.
 - Hash is used as identifiers for blocks, transactions, and addresses.
 - Bitcoin uses Secure Hashing Algorithm 256 i.e., SHA-256 generating a 256-bit hash.
 - It is one-directional i.e., the input can not be generated using hash value. □ It has Avalanche effects.
 - It keeps the database small.

Mining

- The nodes in the Blockchain are known as miners, which mine (add) new blocks.
 - Mining is a mechanism to validate new transactions and add them to the Blockchain ledger.
 - Miners compete to solve a complex mathematical problem based on nonce and Hashing.
 - It includes hashing a block, introducing a nonce to the hashing function, and running hash repeatedly.
 - It is the process of searching a Golden Nonce for which the resulting Hash should be less than the “target value” (unique).
- Once the Golden Nonce is found, the new block is added to the network and propagated.
- The winning node receives some reward and transaction fees (optional).

Consensus Protocol

- A set of rules whereby nodes in the network can achieve agreement on the state of the Blockchain network.
- All participants must agree on a single source of truth.
- Three Features
 - Consensus is required for any change □ Spread control between nodes
 - Consensus protocols are costly to keep the network honest.
 - Byzantine General's Problem □ Proof-of-Work is used in bitcoin Blockchain.
 - Proof-of-Stake is used in Ethereum Blockchain.

Block

The block is a record that contains the transaction data details. It comprises of following details:

- Hash of the block - Alphanumeric number to identify the block
- Hash of the previous block.
- Timestamp
- Nonce - the random number used to vary the value of the hash
- Merkle Root - hash of all the hashes of all the transactions in the block
- Transaction data. (Note: This contains details of several transactions)

Block Number (Fixed)

Time Stamp

Nonce: 3021 Data:
A send 50 Bitcoin to B
C send 50 Bitcoin to D

Previous Hash

Hash

Blockchain Transaction

- A node in the Blockchain requests a transaction via a wallet.
- The transaction is broadcasted to all the nodes in the network.
- The transaction is validated/verified by the network using consensus algorithms.
- The transaction is either accepted or rejected.
 - If accepted, the transaction is added in chronological order with other transactions to create a new block of data.
- The transaction is now part of the Blockchain and is permanent and immutable.

Joe send 0.5 BTC to Ann through the Blockchain.



0.5 BTC



Blockchain Transaction

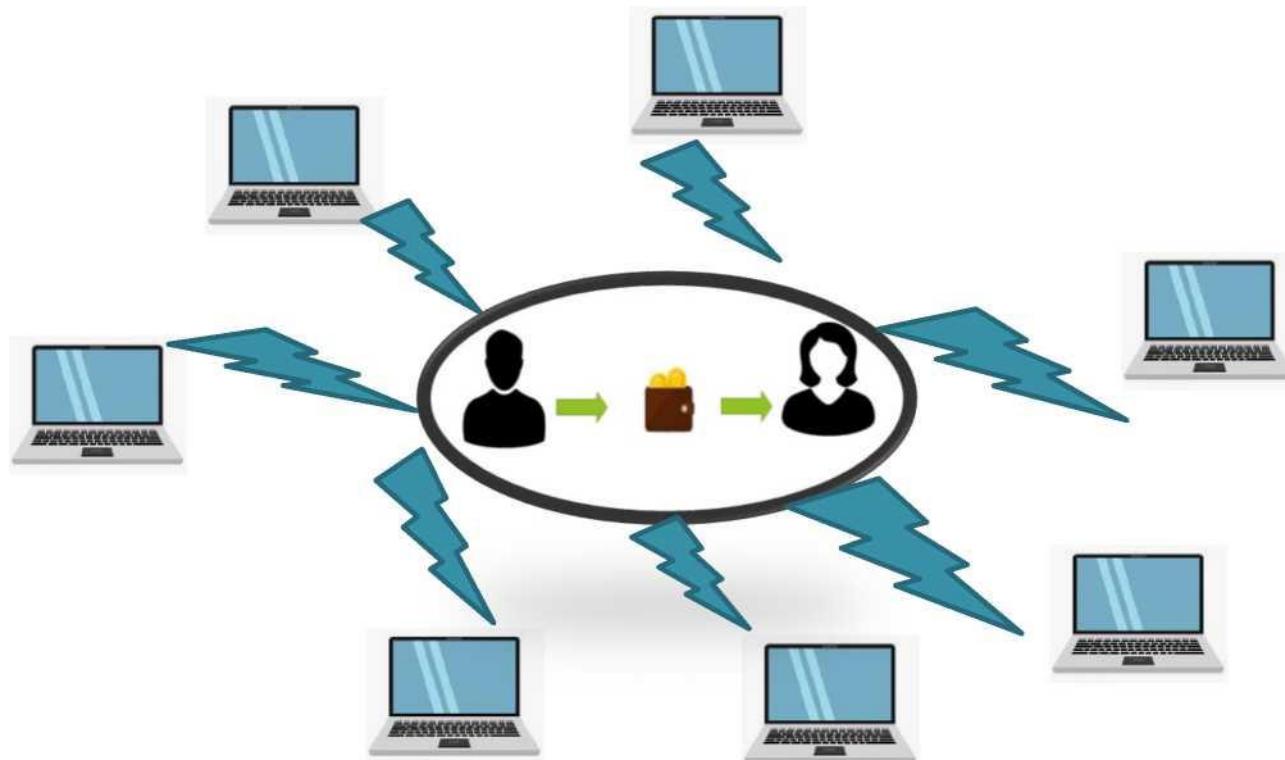
Step1: Joe requests the proposed transaction.

Joe sends 0.5 BTC from his Wallet to Ann through the Blockchain.



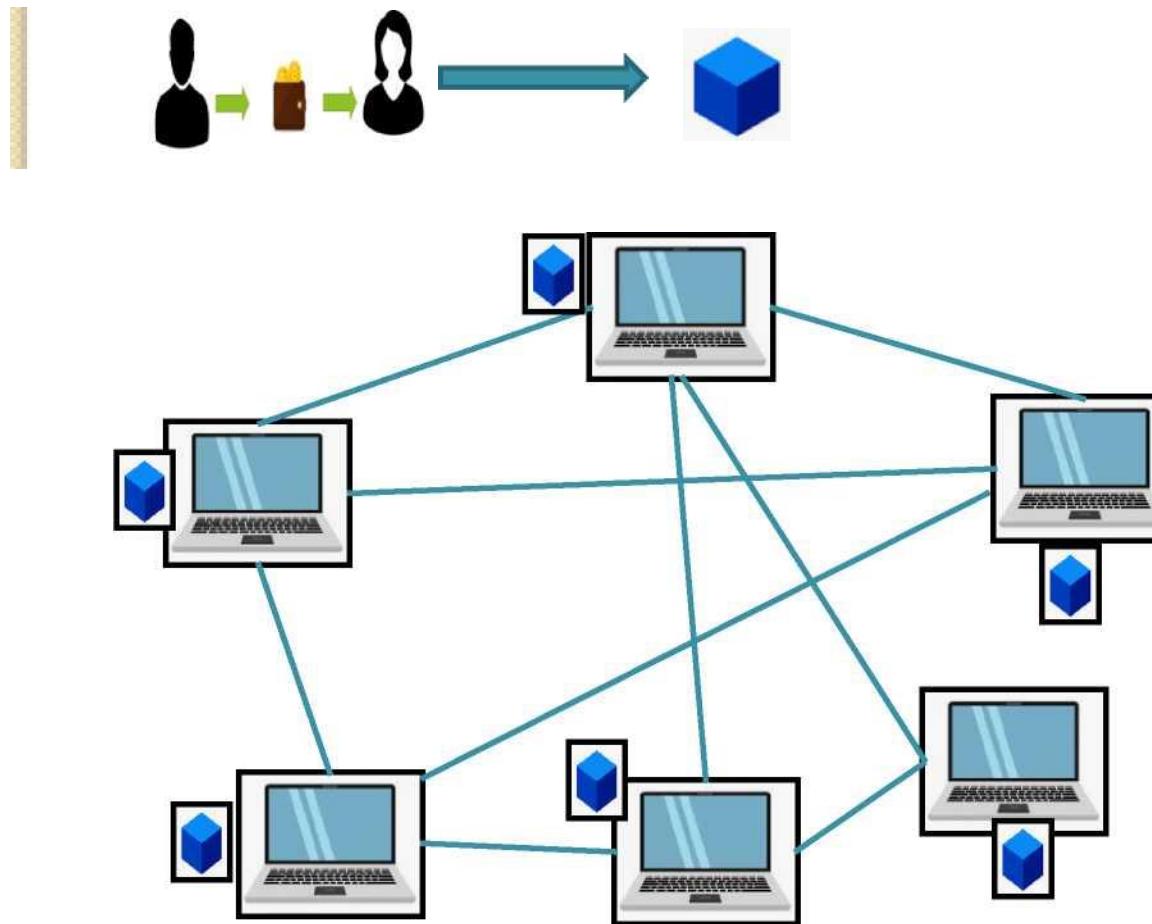
Blockchain Transaction

Step 2: The proposed transaction broadcasted to all the nodes in the network.



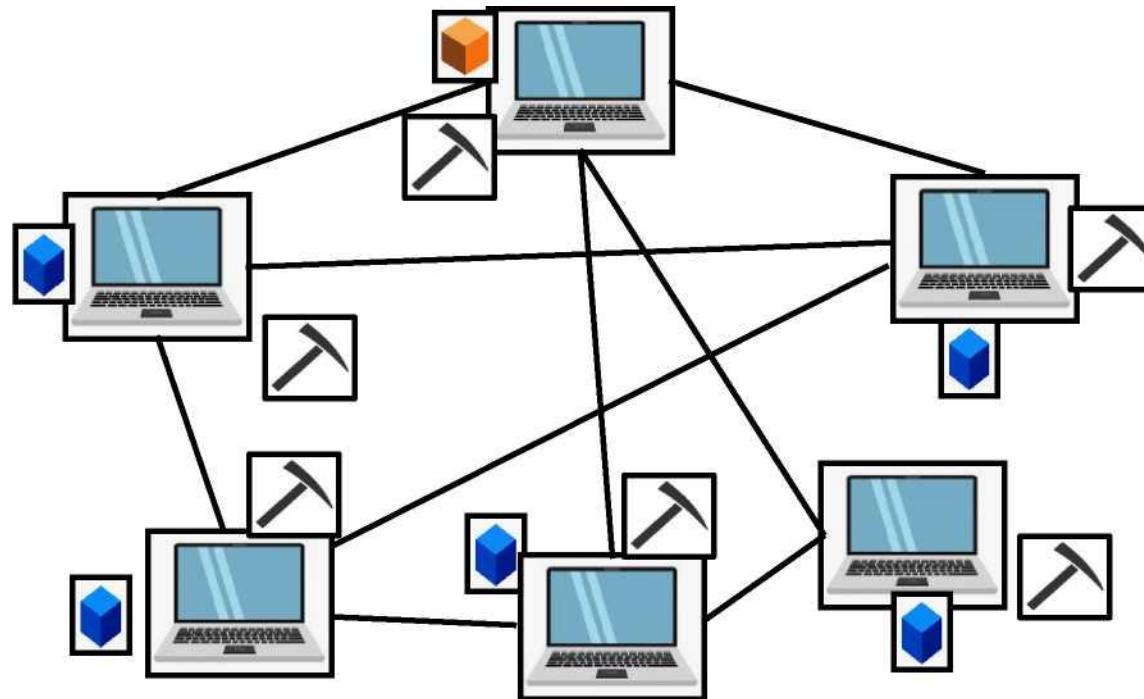
Blockchain Transaction

Step 3: Miners verify the transaction and bundle it into a block with other transactions. Miners will validate the authenticity of the transaction, i.e., the status of Joe, his balance, etc. Miners may include many transactions.



Blockchain Transaction

Step 4: Miners compete to solve the complex mathematical puzzle. The puzzle requires much computational power to solve. It helps to protect the Blockchain against attack.



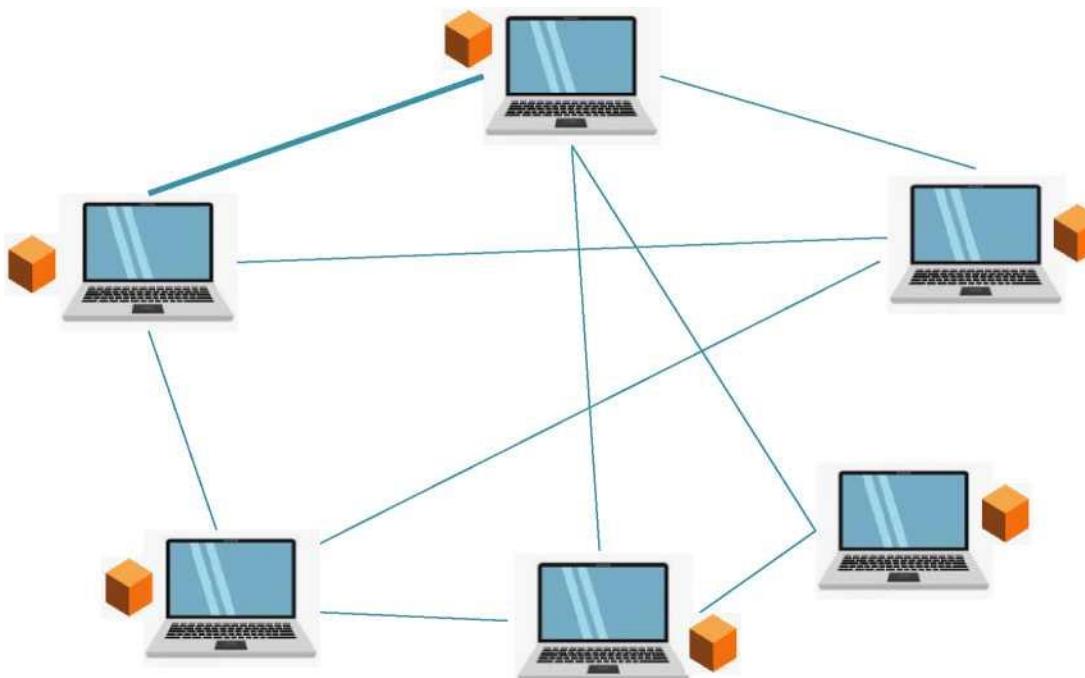
Blockchain Transaction

Step 5: The nodes verify the miner's work.

The miners who find the correct hash broadcast the Block to the network.

The majority of the nodes need to verify the Block.

Once approved, a winning miner can collect his reward and transaction fees (if any).



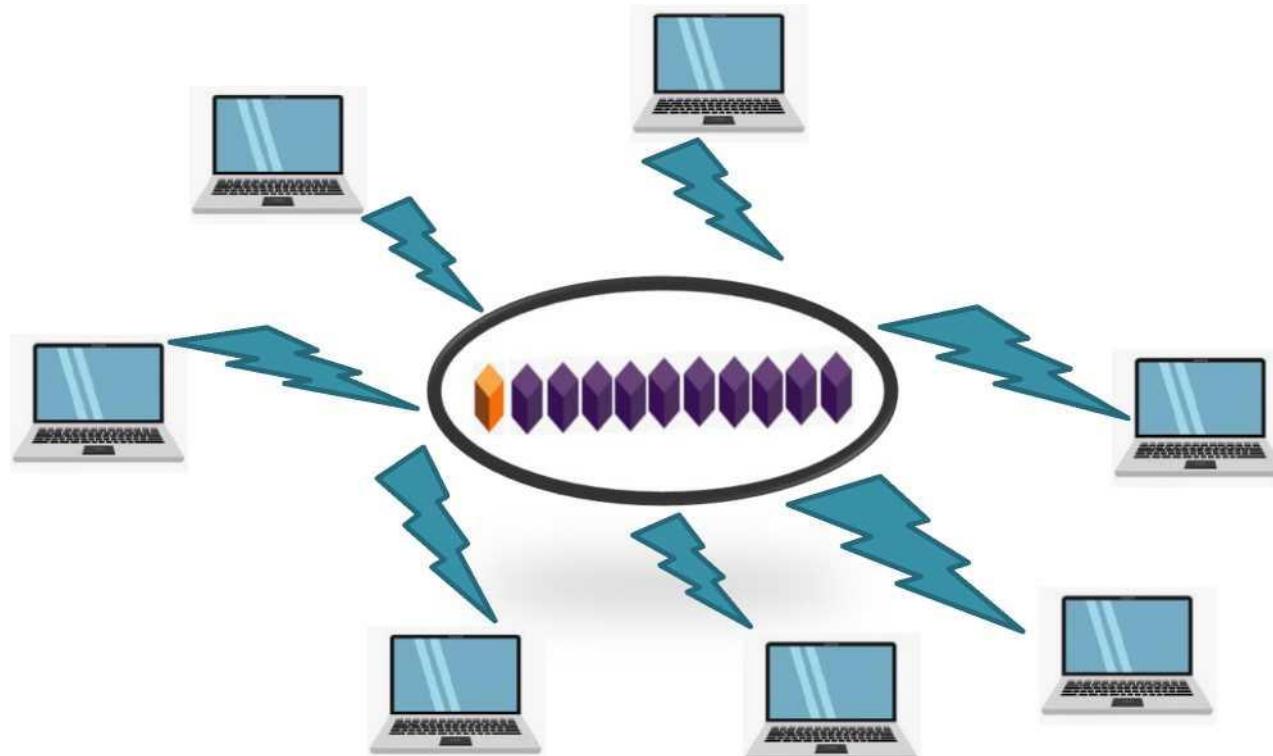
Blockchain Transaction

Step 6: Block is added to the Blockchain.



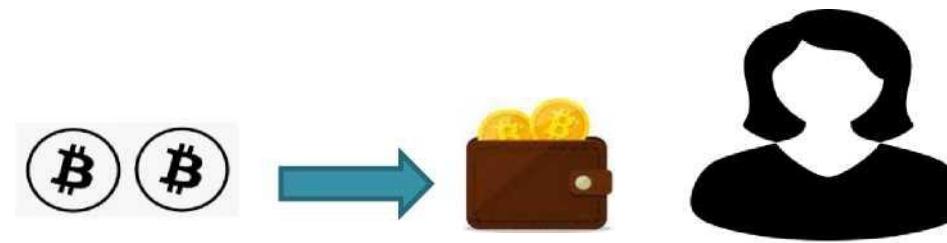
Blockchain Transaction

Step 7: The updated copy of the is circulated throughout the Network.



Blockchain Transaction

Step 8: Transaction Completion. Ann receives 0.5 BTC in her Wallet.



How does Blockchain Technology Work?

Text Book

- *Blockchain Technology by Chandramouli Subramanian, Asha A George, Abhilash K A, and Meena Karthikeyan, Orient Blackswan*



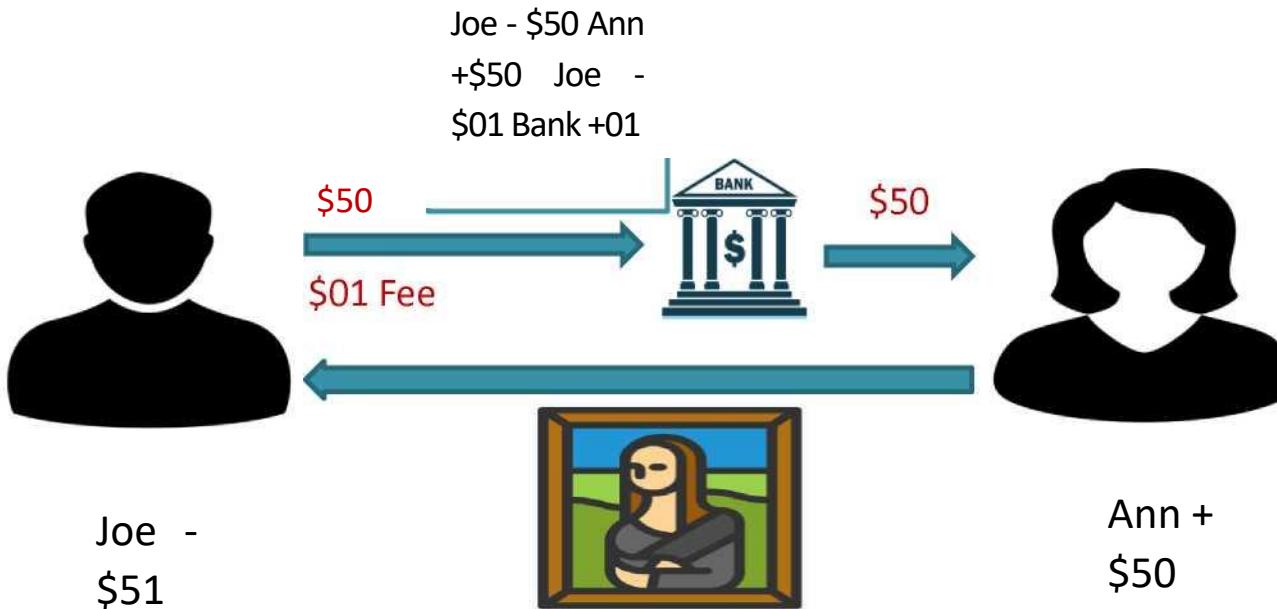
Physical Transaction

- Joe is interested to purchase a painting from Ann for \$50.
- Joe handover a physical \$50 currency note to Ann.
- Ann will hand over the painting to Joe.



Traditional Online Transaction

- Joe and Ann are situated remotely and unable to meet.
- They agree on Online Transactions.
- Bank comes into the picture and acts as the central authority.
- Joe requests to debit his account for \$50 and send this \$50 to Ann's account.
- The bank may charge \$1 for this online transaction.
- The bank debits \$51 from Joe's account and adds \$50 to Ann's account.
- It collects \$1 as a fee from Joe's account.
- Ann acknowledges that after her account is credited, she will send the painting to Joe.
- Ann may also request delivery charges.



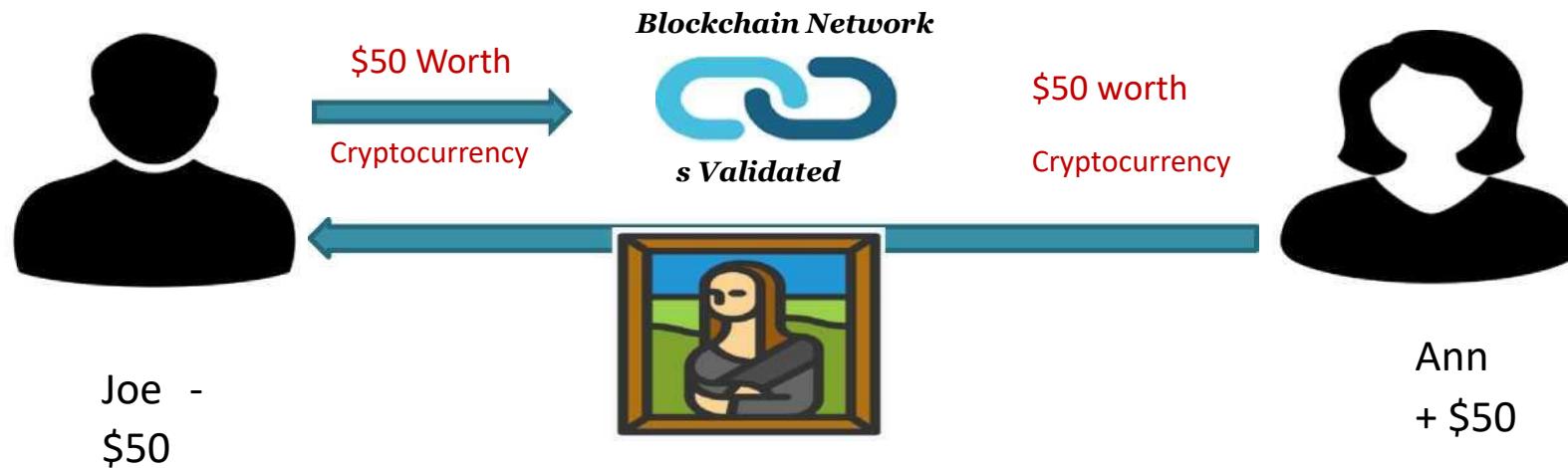
Traditional Online Transaction



- In a real-life scenario, multiple banks or cross-border exchanges involve a large amount of money.
- Financial institutions have become indispensable for verifying ownership, transaction maintenance, and dispute mediation.
- Bank has information regarding both parties and their transaction.
- Bank charges fees to customers.

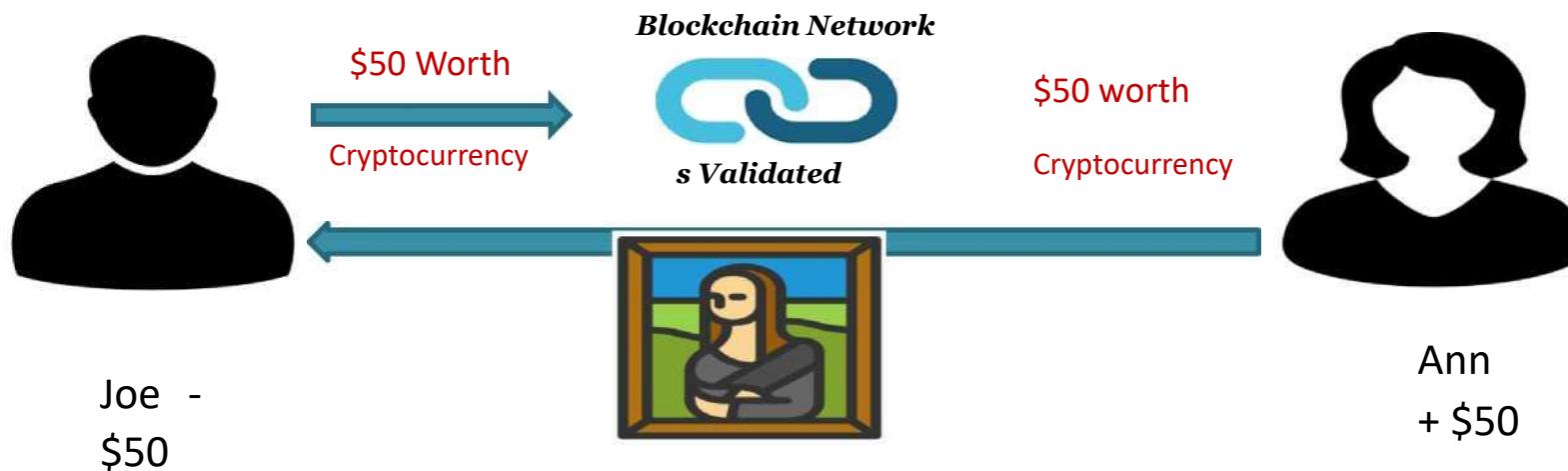
Blockchain Transaction

- ❑ Assume that Joe and Ann are members of the Blockchain.
- ❑ The transaction is performed using cryptocurrency.
- ❑ Joe sends cryptocurrency worth \$50 to Blockchain Network.
- ❑ Once the Blockchain network validates this transaction, Ann receives \$50 equivalent cryptocurrency.
- ❑ A new block is added to the Blockchain. The block is mined by the miners. The winning miner receives some rewards and transaction fees (The fee is insignificant in comparison to fees charged by central banks).
- ❑ Transaction fee is not mandatory



Blockchain Transaction

- Ann acknowledges that after her account is credited, she will send the painting to Joe.
- Ann may also request delivery charges.



How does Blockchain Technology Work?

One of the features of the Blockchain is Digital trust.

Digital trust relates to two questions:

- Authentication (Proving Identity)
 - Are you whom you say you are?
- Authorization (Proving Permissions)
 - Do you have the right to do what you want to do?

How Blockchain Technology Works ?: Authentication

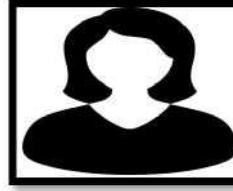
- Authentication is addressed with Cryptography.
- Blockchain utilizes public-key or Asymmetric Cryptography.
 - Public Key and Private keys are generated and stored in the “Wallet”
- Wallet
 - A blockchain wallet is a software program that helps someone exchange funds easily.
 - Stores keys and facilitates cryptocurrency transactions.
 - Transactions are secured, as they are cryptographically signed.
 - The wallet is accessible from web devices, including mobile ones, and the privacy and identity of the user are maintained.
- Public Key is shared publicly.
- Private keys should not be shared with anyone.

How Blockchain Technology Works ?: Authentication



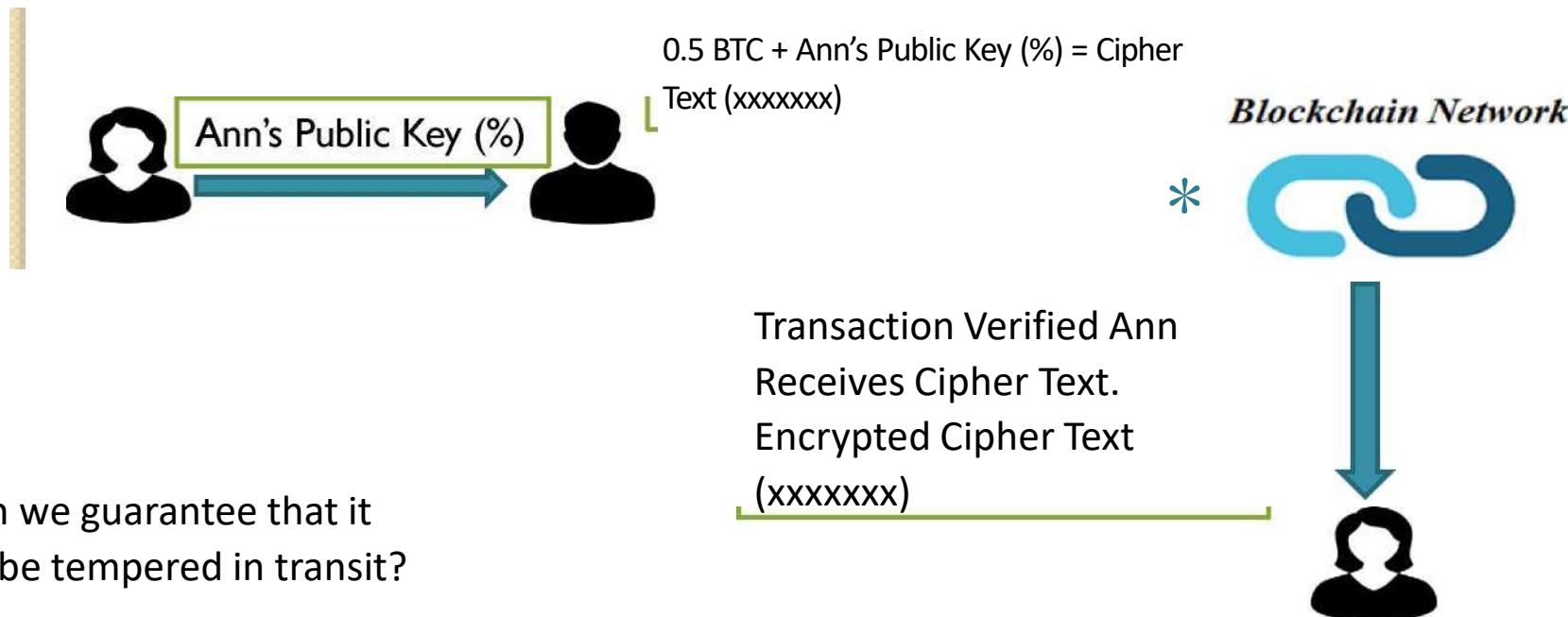
Joe

Public Key (#)
Private Key (@)



Ann

Public Key (%)
Private Key (*)

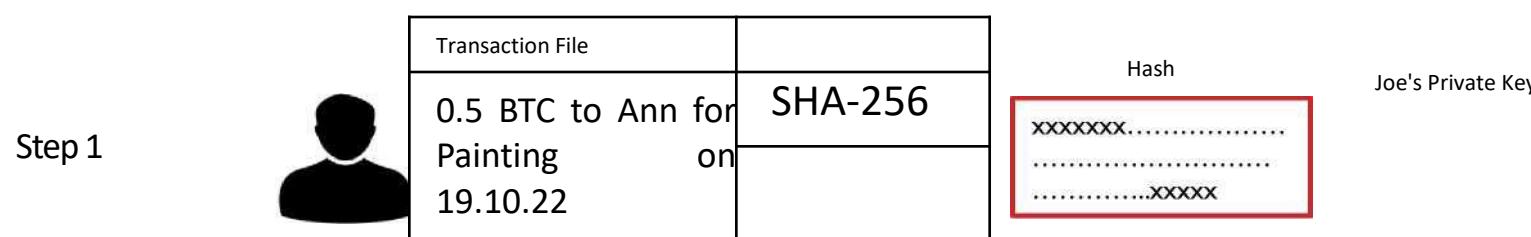
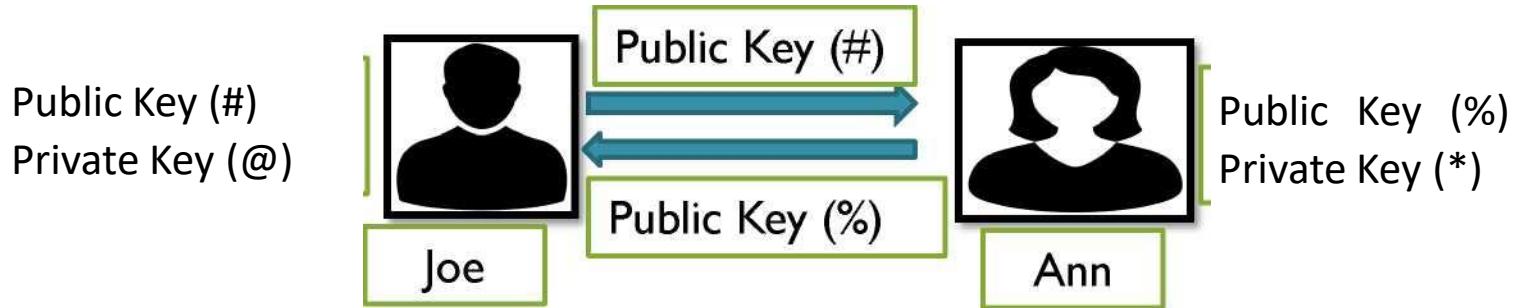


How Blockchain Technology Works ?: Digital Signature

- Digital Signature provides validation and authentication.
- It consists of three parts-
 - Generation of Public and Private keys
 - Public-key cryptography can ensure the security of transactions in the Blockchain.
 - The Transactions are time-stamped, making each transaction unique.
 - Signing Algorithm
 - Verification Algorithm

How Blockchain Technology Works ?: Digital Signature

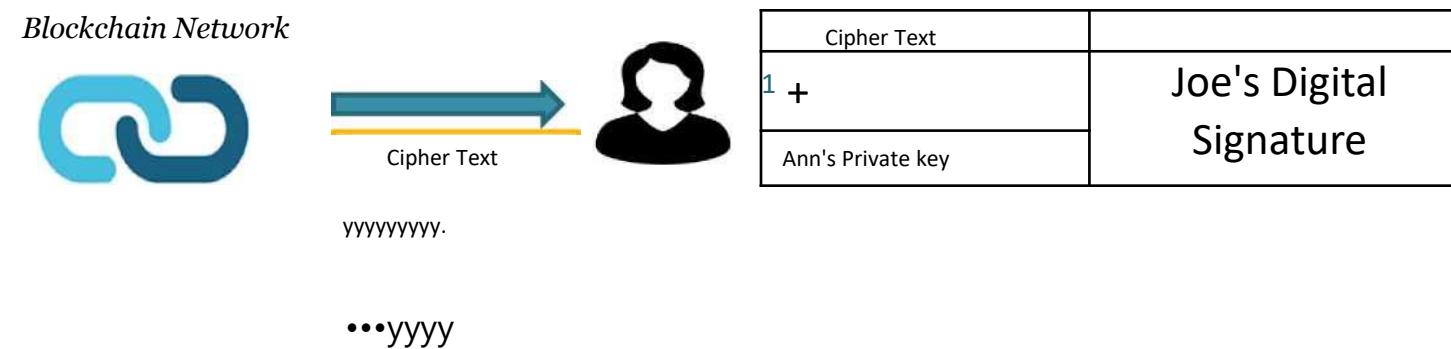
□ Signing Algorithm



&

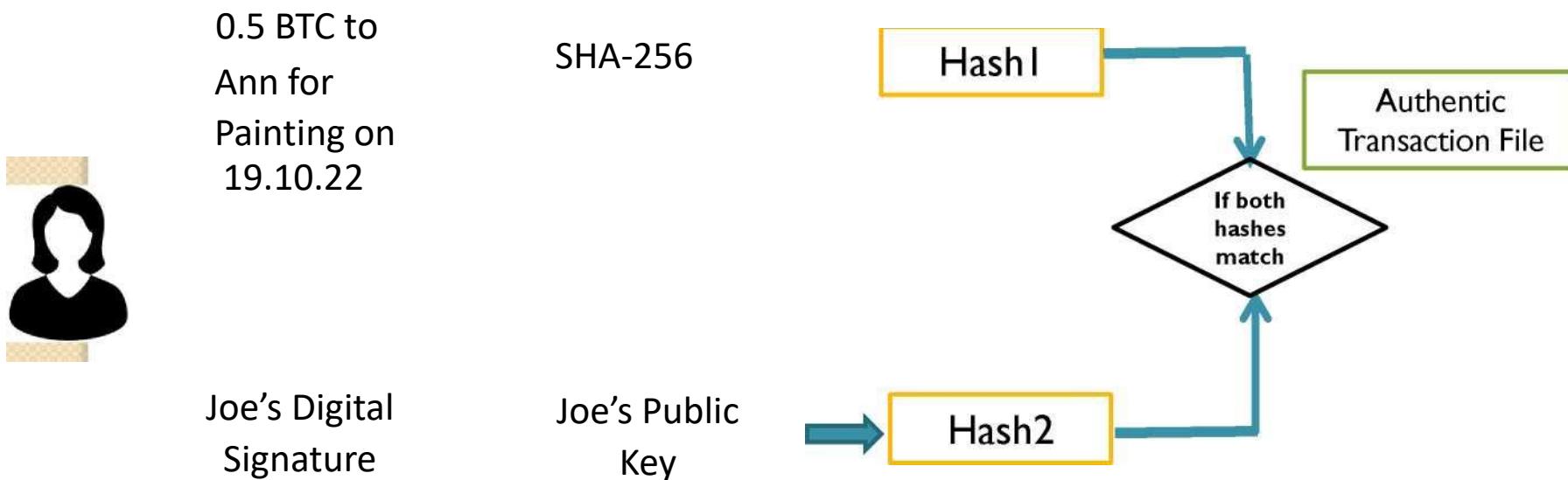
How Blockchain Technology Works ?: Digital Signature

□ Signing Algorithm



How Blockchain Technology Works ?: Digital Signature

□ Verification Algorithm



How Blockchain Technology Works ?

- Digital Signature
 - Helps to maintain the immutability of blocks.
 - Confirms the security, authentication, integrity, and non-repudiation of data or transactions.
 - Helps to address the “Authentication” i.e.,“Are you whom you say you are?”.
- Authorization (Proving Permissions):“Do you have the right to do what you want to do ?”
 - Addressed by distributed consensus on a P2P network on Blockchain.
 - All nodes are peers and consist of an identical copy of the database or ledger.
 - All nodes have the responsibility to maintain the network.

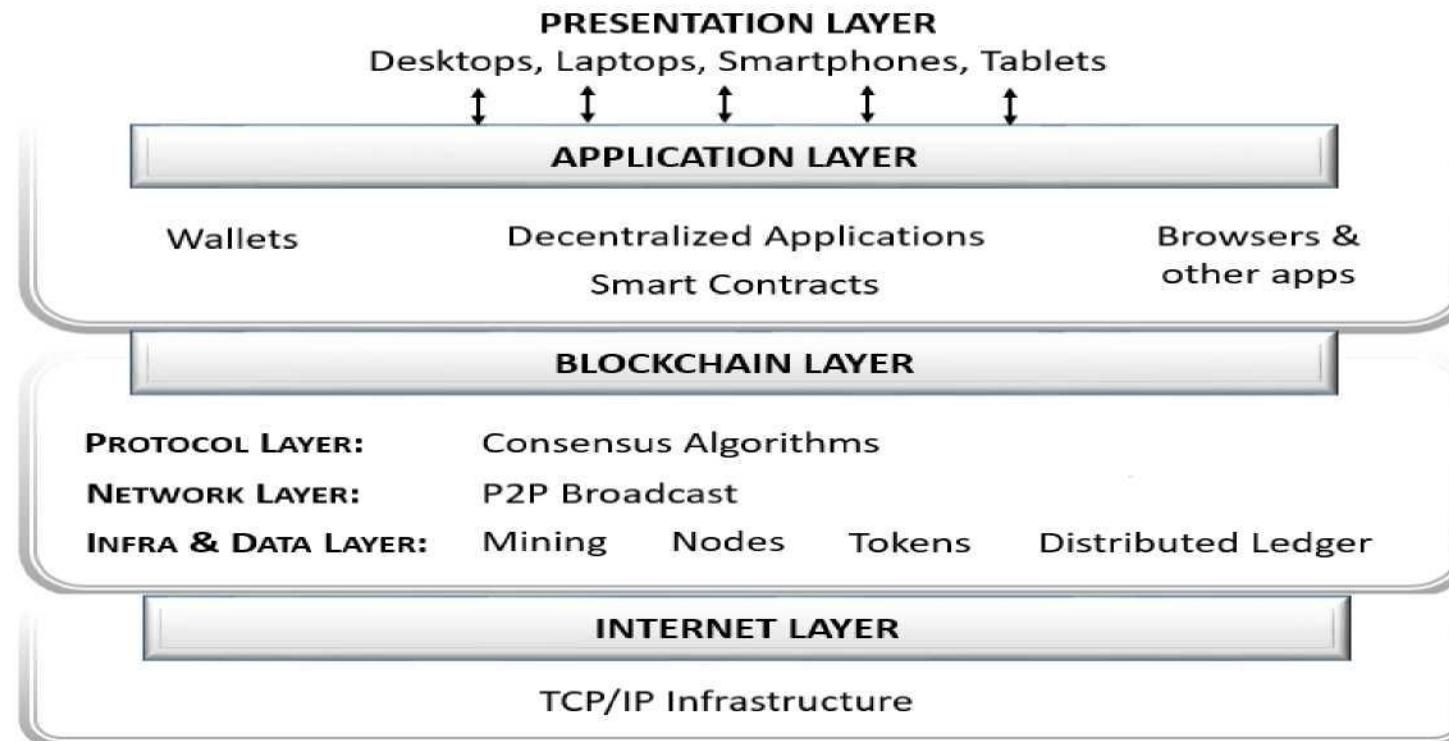
Double-spending

- Double-spending is spending money more than once.
- Blockchain's consensus process and the fundamental temporal structure of how the blocks are chained together prevent the double-spend problem.
- A transaction is verified and added to a block via the consensus mechanism after a considerable amount of computational power and resources are spent.
- Once a transaction is confirmed, it is nearly impossible to double-spend it.
- The more confirmed blocks in the chain, the harder it is to double-spend the crypto.

Blockchain Layers

A simple structure of the Blockchain ecosystem consists of four layers-

- Presentation Layer
- Application Layer
- Blockchain Layer
- Internet Layer



Blockchain Layers

Presentation Layer

- It provides communication between the user and the Blockchain network.
- A door to users to provide access to the Blockchain network.
- User interface (UI) and User Experience (UX) of the App play a major role.
- Security, Ease of usage, and Mobility are important.
- A good UX design will improve customer satisfaction and company reputation.
- A good UI design provides effective visuals leading to an intuitive interface.

Application Layer

- It provides communication between the Blockchain layer and the presentation layer.
- It combines user experience and business logic.
- It consists of Decentralized applications (DApps)
- DApps connect to the Blockchain via Smart contracts.
 - Smart contract is a program consisting of rules that automatically execute if the pre-defined conditions are met.

Blockchain Layers

Blockchain Layer

- It consists of core components of the Blockchain.
- It consists of the consensus algorithms, the medium, and the interface for the P2P network.
- It controls the mining process, consensus protocols, participating nodes, and distributed ledger.

Internet Layer

- Blockchain runs over the internet.
- It helps to connect all nodes (computers, IoT devices, smartphones, etc.)

Decentralized and Distributed

Blockchain technology works on the principle of ledger data distributed to all nodes that are non-hierarchical with no single central control.

Pros

- Removes any single point of failure by replicating the ledger on every node in the network
- Better communication between nodes fosters transparency and faster consensus and synching of data
- It allows for more engagement as everyone is involved in the decision-making process and keeping the network honest.

Cons

- In some cases, the traditional database may be more suited and do the work a lot faster and cheaper
- Specific and trusted third parties exist in some domains that may guarantee more efficient and specialized services using other technologies
- If a time-tested and fully functional database and the operational network are already in place, the benefits of replacing or introducing blockchain may not produce the required return on investment.
- Stronger players (nodes with higher computing power or with pooling) can take control of the network, impacting decentralization.

Trustlessness

In the blockchain, cryptography completely replaces the need for third parties to ensure trust. Also, the complex consensus protocols that are run within the blockchain network to unanimously and securely agree on what should be added and should not be added to the ledger secures it further, thus ensuring its integrity at all times.

Pros

- Allows for multiple entities or key players who do not trust each other (i.e., unknown to each other or across borders) to transact directly with one another
- Ensures valid and accurate data
- Disintermediation (removal of the middleman) reduces the overall cost of transacting.

Cons

- The integrity of data is obtained at the expense of time. Every node needs to run the blockchain to verify transactions and maintain consensus. Currently, blockchain can, on average, process only 5 transactions/sec.
- Significant computing power is expended by miners leading to substantial energy consumption and wastage. Hence it is not suitable for organizations that require instant transaction results within milliseconds.
- Nodes may prioritize transactions with higher rewards.

Immutability

In the blockchain, one cannot modify data or transactions once they are recorded in the Blockchain database. It becomes a permanent record that is close to impossible to change. Any change required can be addressed only by adding a new block of data to the existing chain of blocks in chronological order, ensuring that the database is complete and consistent.

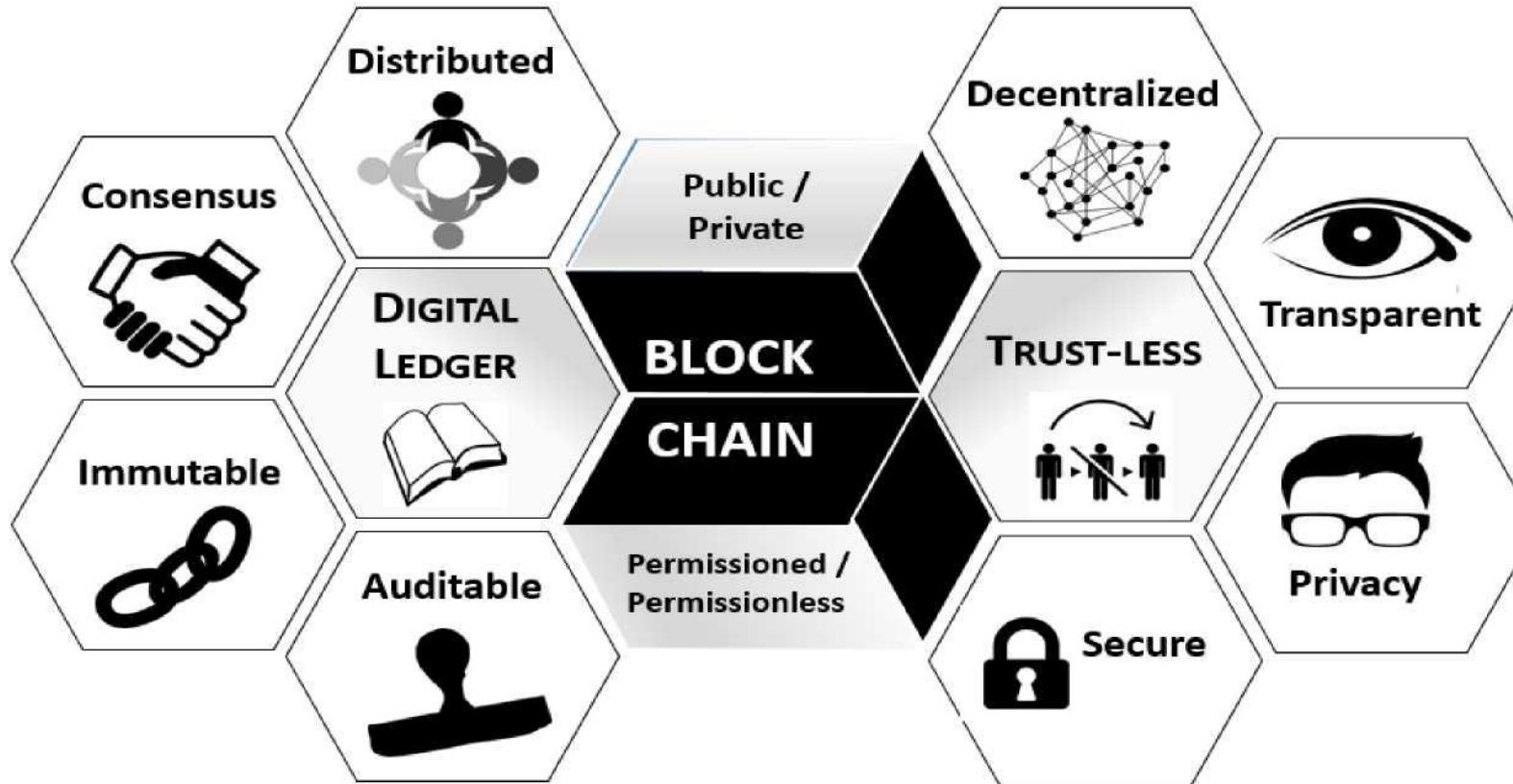
Pros

- Contains a verifiable record of all transactions made that is auditable.
- Consensus algorithms and Blockchain propagation mitigate the risk of double-spending fraud and manipulation of data.
- There is provenance, i.e., the ability to track transaction or product movement across accounts.

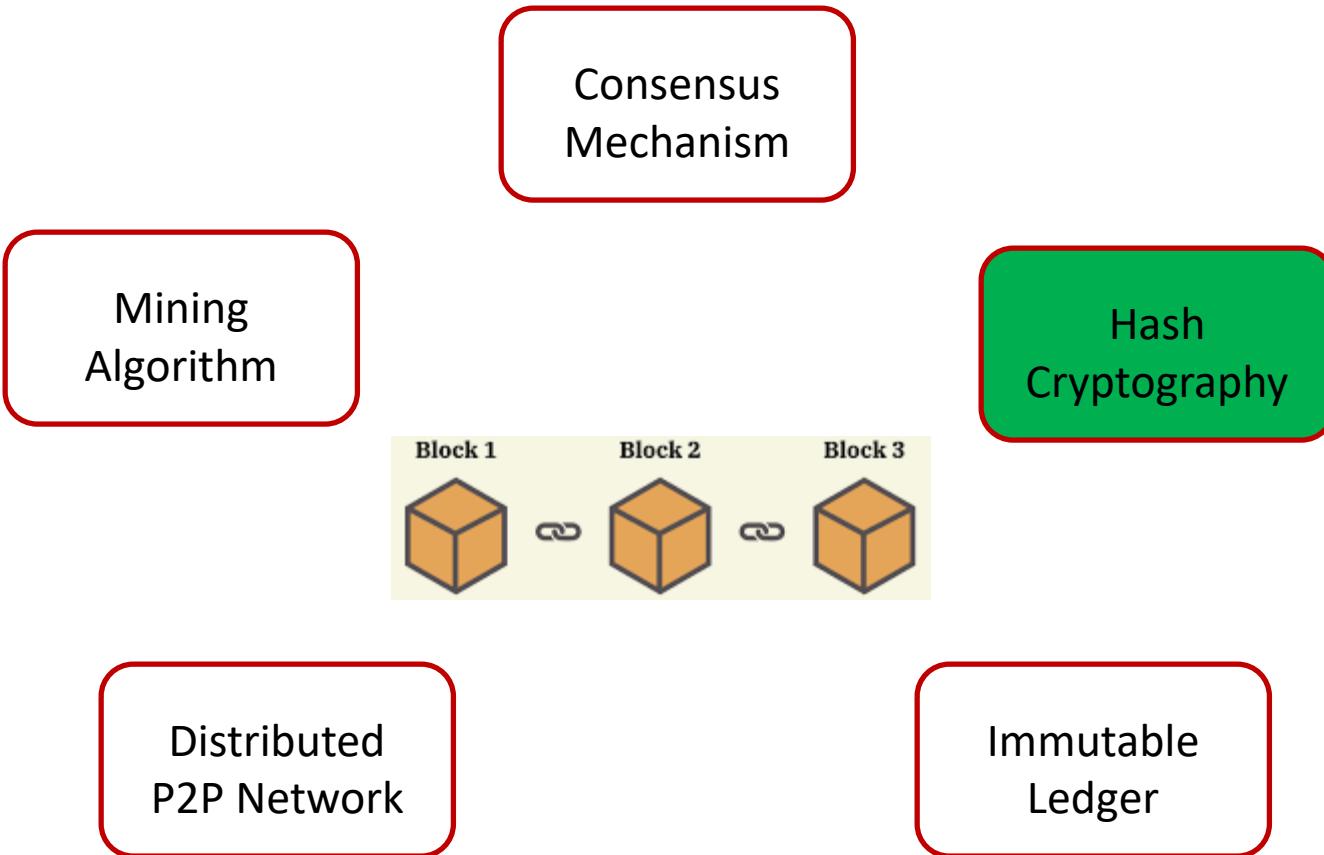
Cons

- Not every node has the capacity to maintain and run a full copy of the Blockchain. This can potentially affect the consensus and immutability.
- In smaller Blockchain, there is a risk of 51% attack. If one or a group of malicious nodes can get 51% of the mining hash rate, they can manipulate the transactions.
- Quantum Computing can potentially break the cryptographic algorithm to reverse engineer public keys of the Blockchain networks to obtain the private keys.

Characteristics of Blockchain

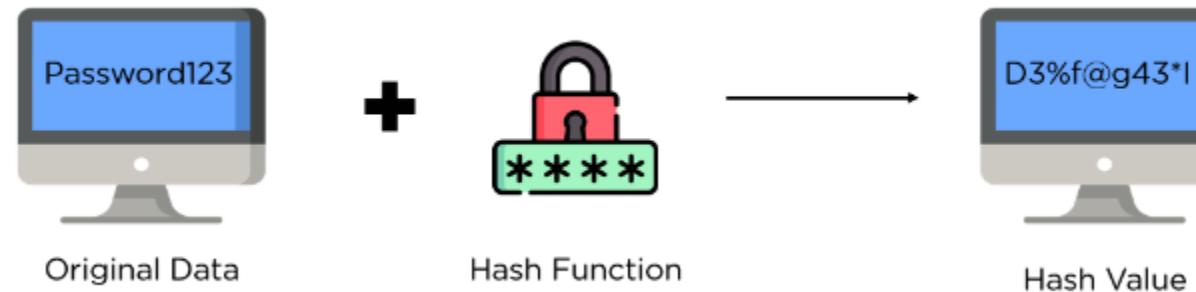


Blockchain Technology: An Intuition



Hash Cryptography

- ❑ Hashing is the process of scrambling raw information to the extent that it cannot reproduce it back to its original form.
- ❑ It takes a piece of information and passes it through a function that performs mathematical operations on the plaintext.
- ❑ This function is called the hash function, and the output is called the hash value.



- ❑ There are two primary applications of hashing:
 - ❑ Password Hashes
 - ❑ Integrity Verification

Hash Cryptography

- ❑ Secure Hash Algorithm (SHA)-256 is a part of the SHA 2 family of algorithms which was published in 2001.
- ❑ It was a joint effort between the NSA and NIST to introduce a successor to the SHA 1 family, which was slowly losing strength against brute force attacks.
- ❑ In SHA-256, the hash value will always be 256 bits irrespective of the size of plaintext/cleartext.

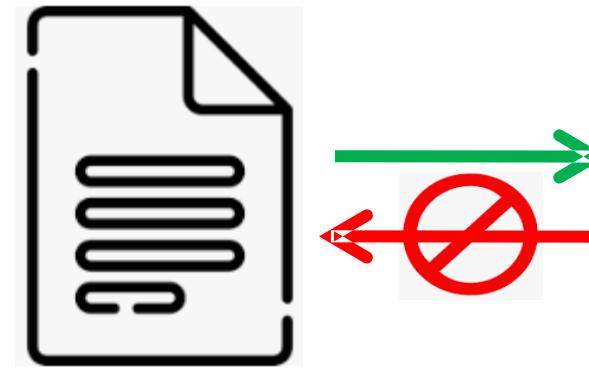
<https://medium.com/biffures/part-5-hashing-with-sha-256-4c2afc191c40>

https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm#what_is_hashing

Hash Cryptography

Five requirements of Hash Algorithms are –

1) One way



2a94608f8d29768c
C20d46eff4e793e5
E5f00d3dfdf9c03e
0b6aa6cdac33e4b8

Hash Cryptography

Five requirements of Hash Algorithms are –

- 1) One way
- 2) Deterministic



2a94608f8d29768c
C20d46eff4e793e5
E5f00d3dfdf9c03e
0b6aa6cdac33e4b8



2a94608f8d29768c
C20d46eff4e793e5
E5f00d3dfdf9c03e
0b6aa6cdac33e4b8

Hash Cryptography

Five requirements of Hash Algorithms are –

- 1) One way
- 2) Deterministic
- 3) Fast Computation



2a94608f8d29768c
C20d46eff4e793e5
E5f00d3dfdf9c03e
0b6aa6cdac33e4b8

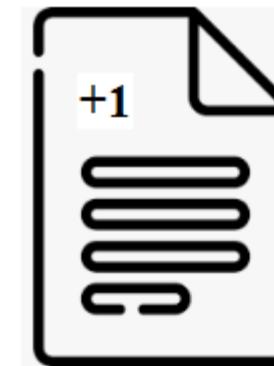
Hash Cryptography

Five requirements of Hash Algorithms are –

- 1) One way
- 2) Deterministic
- 3) Fast Computation
- 4) The Avalanche Effect



2a94608f8d29768c
C20d46eff4e793e5
E5f00d3dfdf9c03e
0b6aa6cdac33e4b8



9f075f5ff2c2c35b
1159ec947098fa93
F996a6b79857c955
2f8bbf8890eb86e1

Hash Cryptography

Five requirements of Hash Algorithms are –

- 1) One way
- 2) Deterministic
- 3) Fast Computation
- 4) The Avalanche Effect
- 5) Must withstand collisions



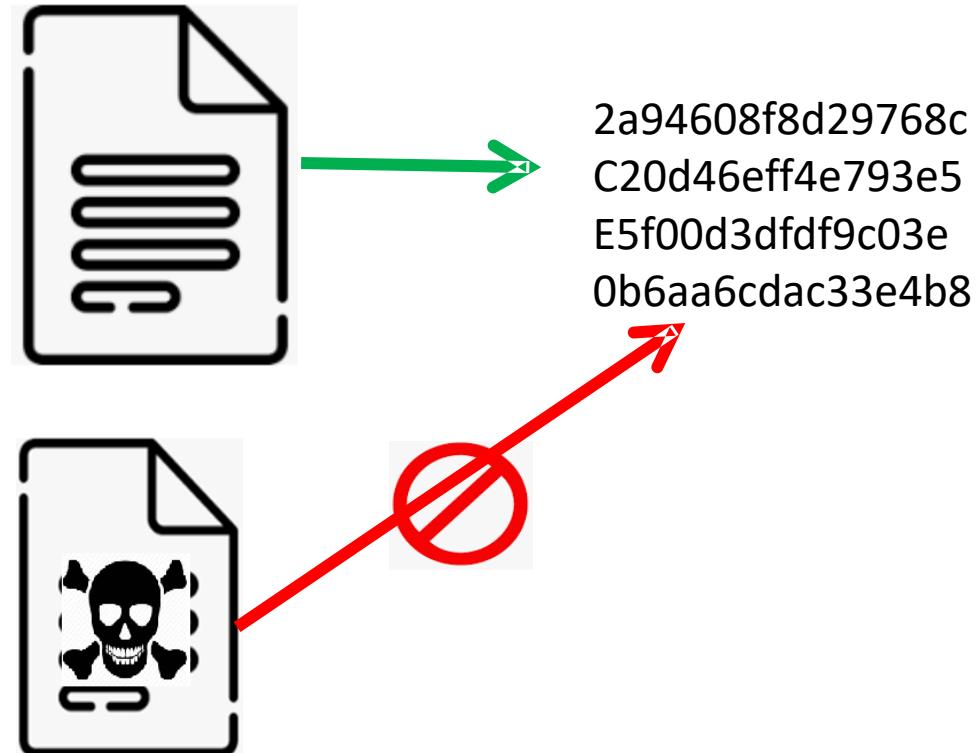
2a94608f8d29768c
C20d46eff4e793e5
E5f00d3dfdf9c03e
0b6aa6cdac33e4b8



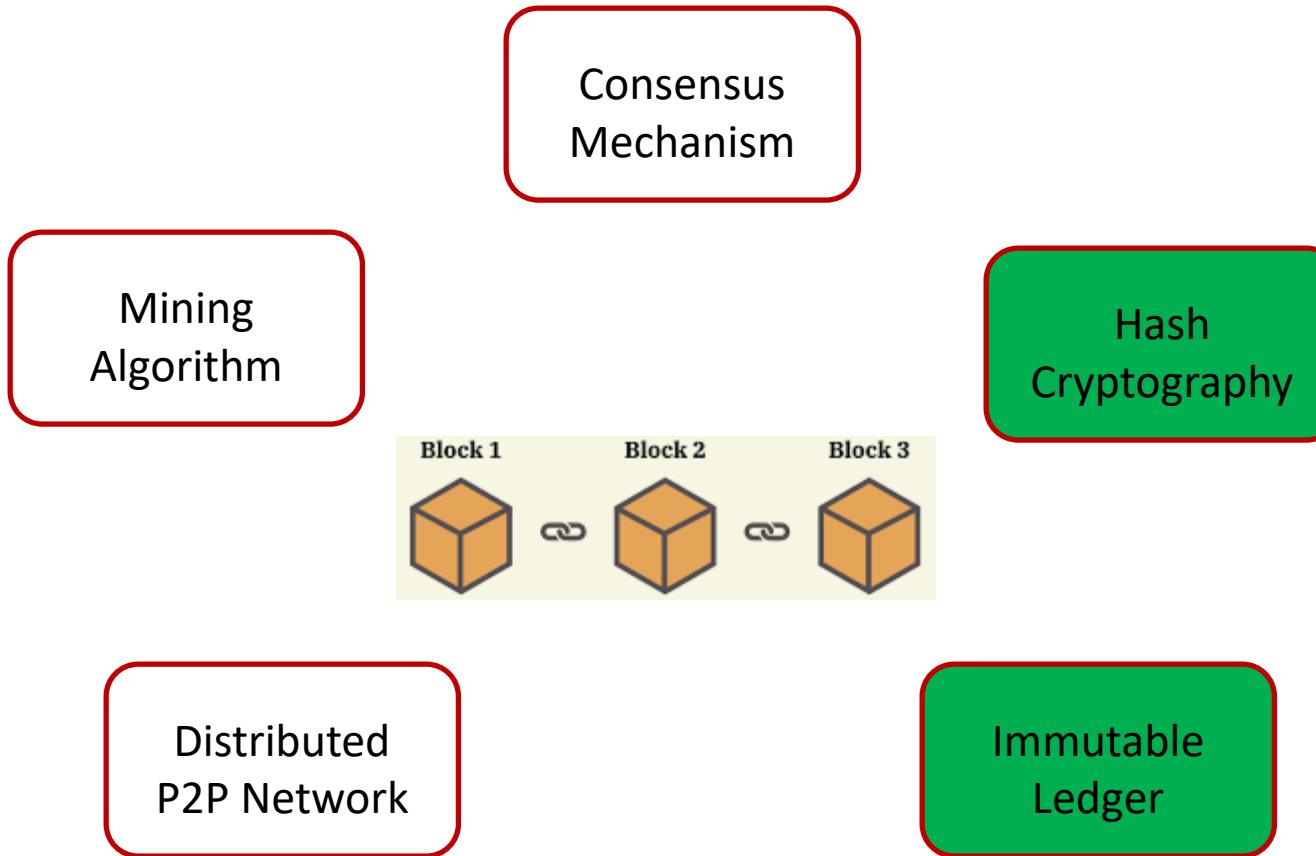
Hash Cryptography

Five requirements of Hash Algorithms are –

- 1) One way
- 2) Deterministic
- 3) Fast Computation
- 4) The Avalanche Effect
- 5) Must withstand collisions



Blockchain Technology: An Intuition



Immutable Ledger

- ❑ The biggest issue in cybersecurity
 - ❑ ensuring that our data has not been manipulated, replaced, or falsified by a company or its employees.
 - ❑ It is possible by methods like private keys and user permissions.
 - ❑ But in reality, we cannot prove — methodically or mathematically.
- ❑ **Immutability**
 - ❑ The ability for a blockchain ledger to remain a permanent, indelible, and unalterable history of transactions.
- ❑ **Immutability is achieved by**
 - ❑ Cryptography + Blockchain Hashing Process = Immutability

Immutable Ledger

Read this sentence-

Each transaction that is verified by the blockchain network is timestamped and embedded into a “block” of information, cryptographically secured by a hashing process that links to and incorporates the hash of the previous block, and joins the chain as the next chronological update.

How immutability is achieved?

The hashing process of a new block always includes meta-data from the previous block’s hash output.

This link in the hashing process makes the chain “unbreakable” — it’s impossible to manipulate or delete data after it has been validated and placed in the Blockchain.

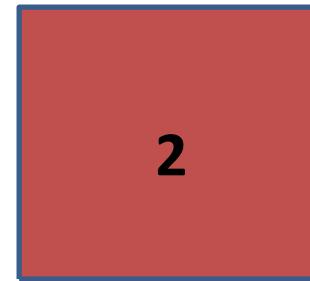
If attempted, the subsequent blocks in the chain would reject the attempted modification (as their hashes wouldn’t be valid).

In other words, if data is tampered with, the blockchain will break, and the reason could be readily identified.

This characteristic is not found in traditional databases, where information can be modified or deleted with ease.

Immutable Ledger

Genesis Block



Data: A

P. Hash: 0...000000

Hash: 0...034256DAB

Data: B

P. Hash: 0...034256DAB

Hash: 00.....1ABDEF230

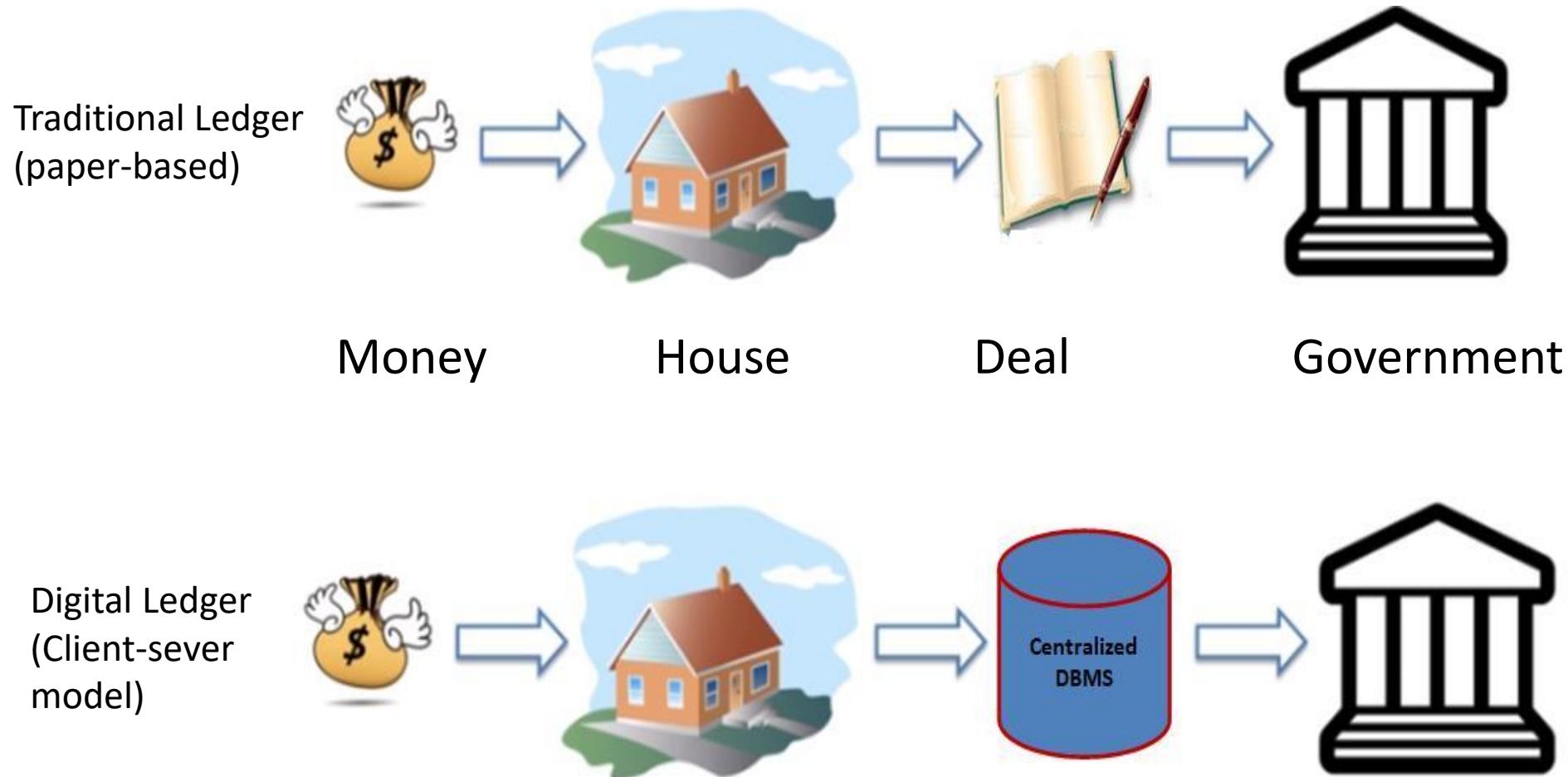
Data : C

P. Hash: 00.....1ABDEF230

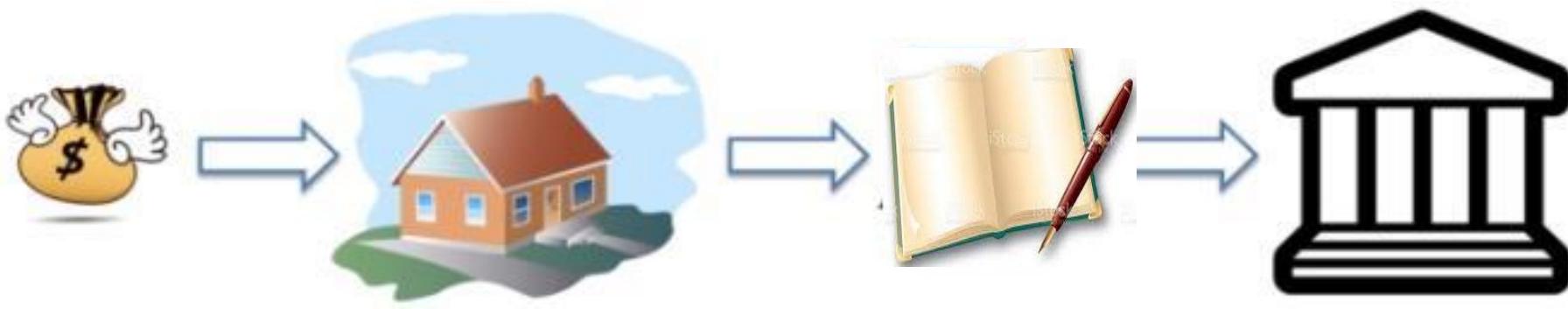
Hash: 00.....1879A23098

Blocks are cryptographically linked together.

Immutable Ledger



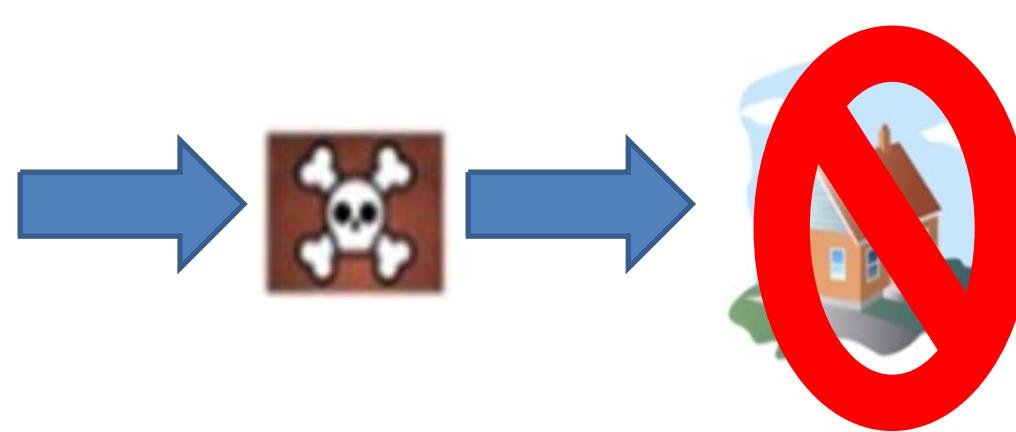
Immutable Ledger



Traditional
Ledger

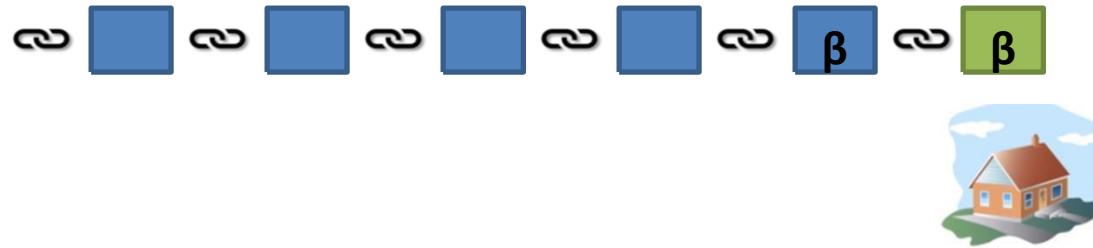


Centralized
DBMS



Immutable Ledger

- Consider a Blockchain-based land registry System in which new block will be added for new transactions.
- When any person purchase a home, It is added in the new Blockchain with transfer of ownership and other details.

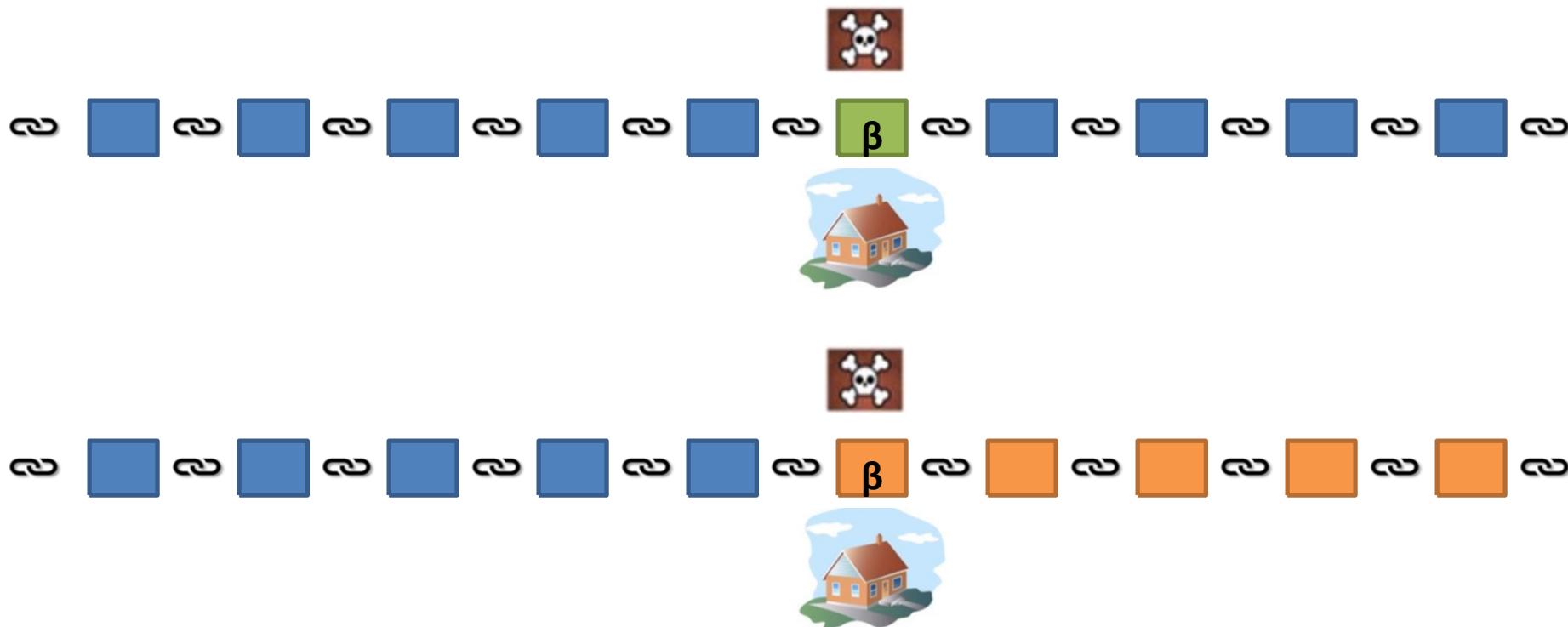


- Many transactions occur on daily basis, thus, many blocks are added to the Blockchain.



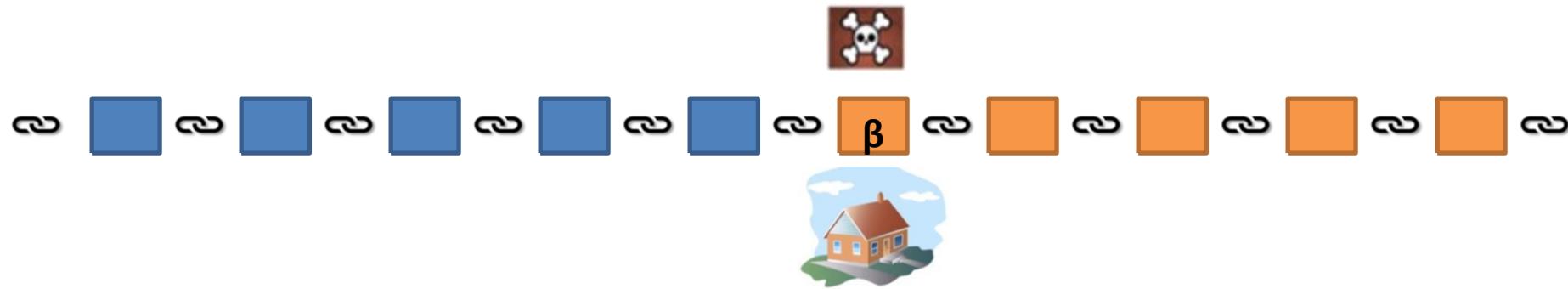
Immutable Ledger

- A malicious person tries to hack the desired block.
- The hash of block β will be changed.
- The cryptographic link between block β and the next block will no longer work.
- After a single change in block β , the next blocks will no longer be valid.
- Thus, the malicious person needs to change the hash of all blocks after block β .

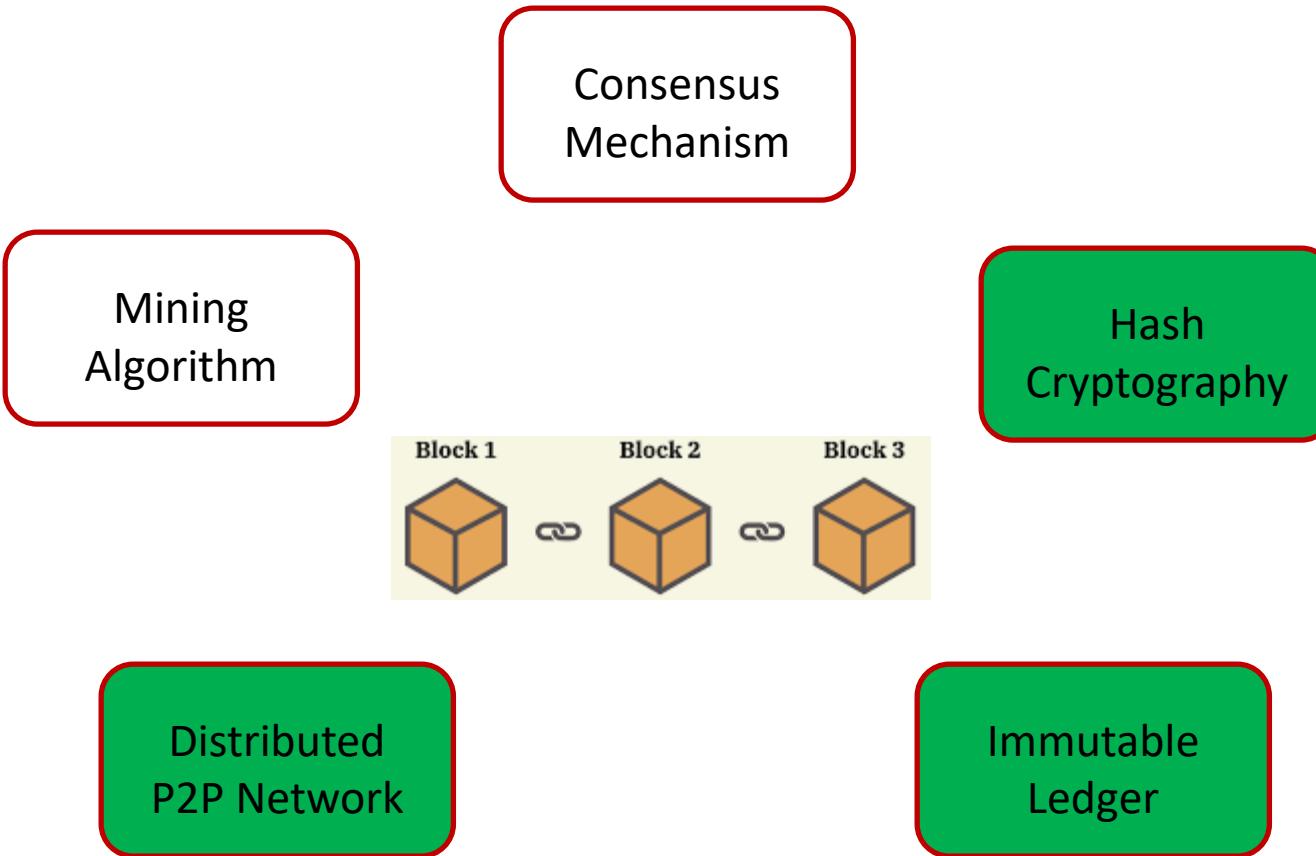


Immutable Ledger

- Average time to add one block in the Bitcoin Blockchain is 10 minutes due to Proof-of-Work Algorithm (The most computationally demanding algorithm).
- It means 52560 ($365*24*6$) blocks will be added in one year.
- Thus, it is required to change the hash in too many blocks to perform a single change in a block.
- Thus, It is practically very difficult to change one block in the Blockchain.
- It shows the immutable property of Blockchain Ledger.



Blockchain Technology: An Intuition



Distributed P2P Network

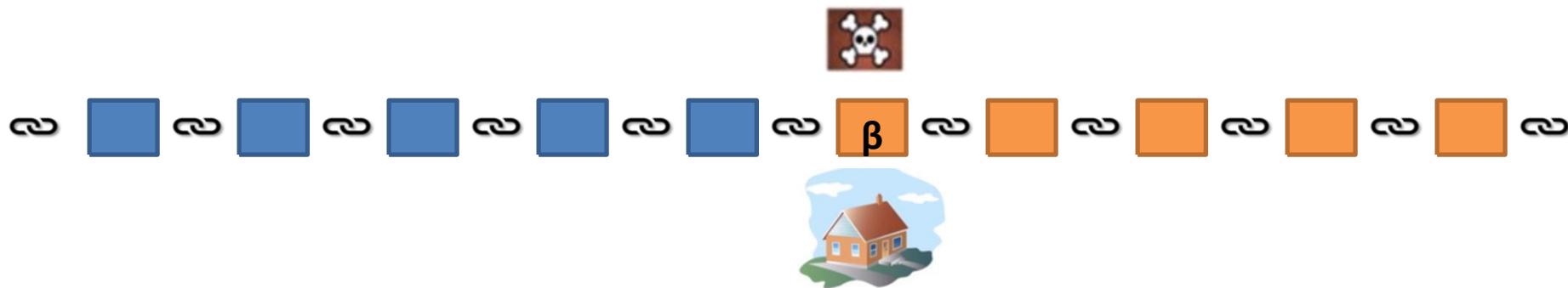
▪ Scenario 1:

- Assume that Property Ledger-based Blockchain is maintained by a single organization. If the malicious person has enough time, he may change the hash of all blocks followed by the block β .

▪ Scenario 2:

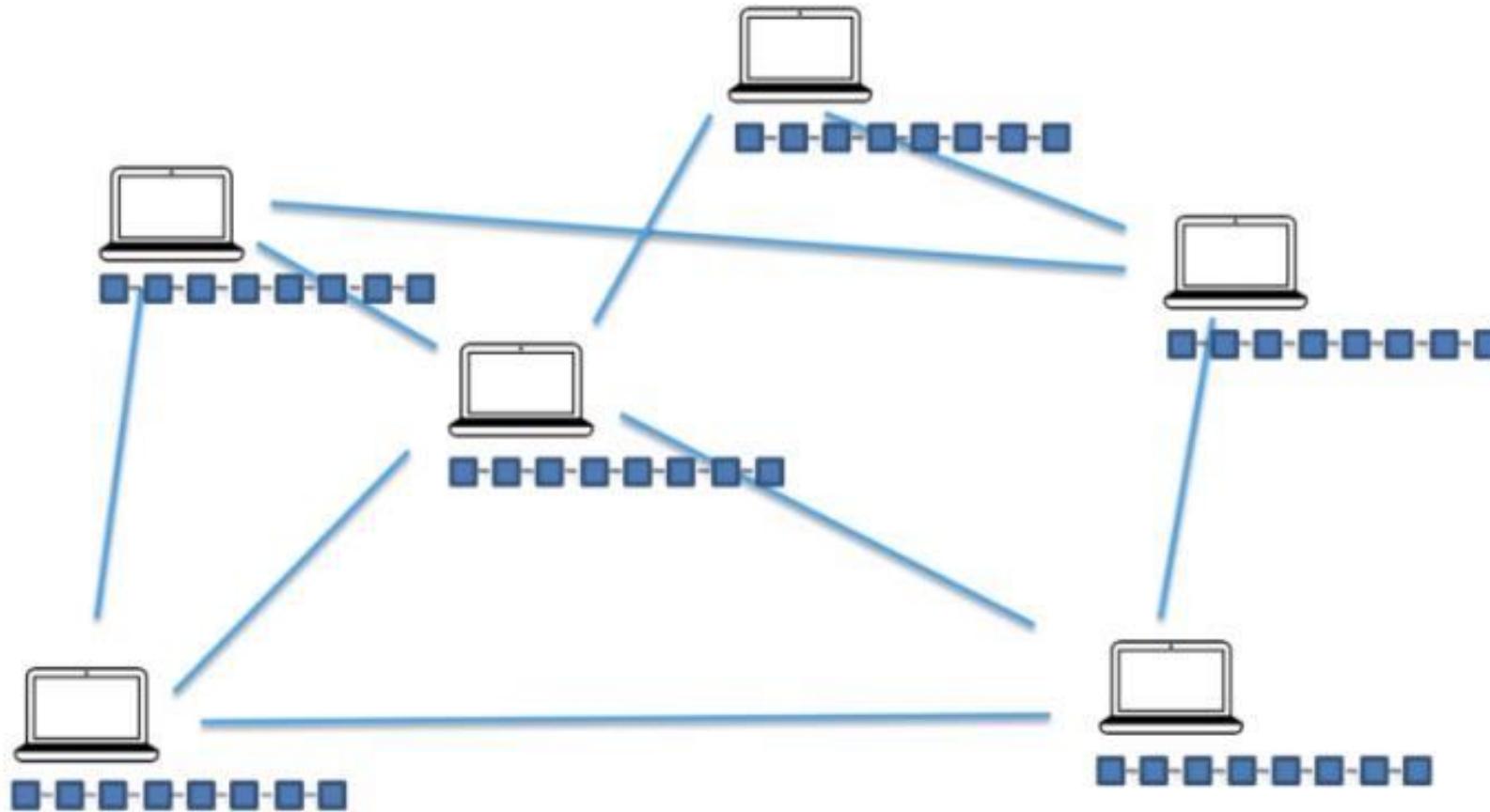
- If the data in block β accidentally has been changed (corrupted), the cryptographic chain will be broken. How to recover data in the block β ?

▪ How Blockchain maintains Immutability?



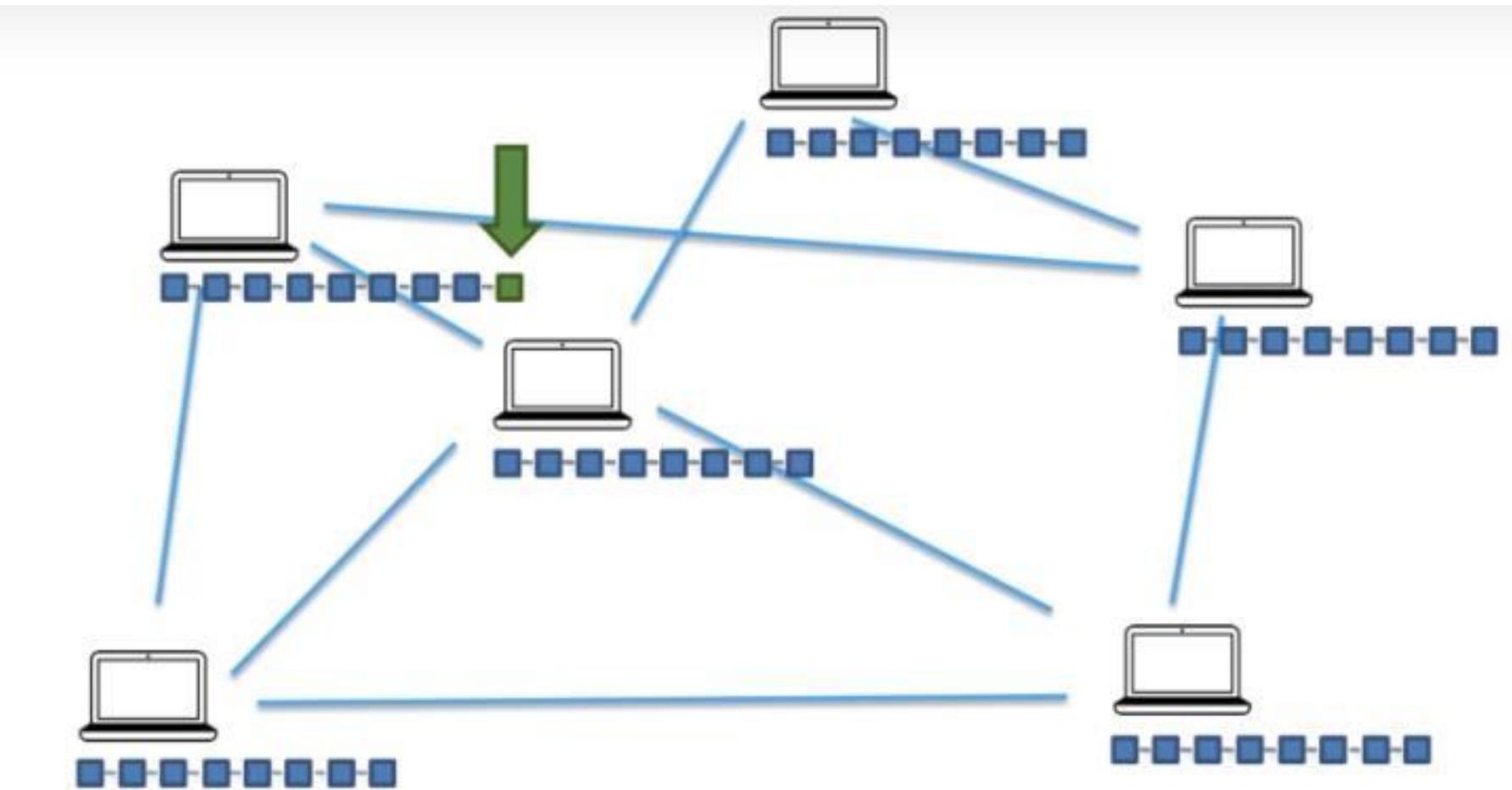
Distributed P2P Network

- There exists distributed P2P network of many nodes (1 million) across the Globe.
- Every node in the distributed P2P network contains a copy of the whole Blockchain.
- A small change needs approval of the majority of nodes.



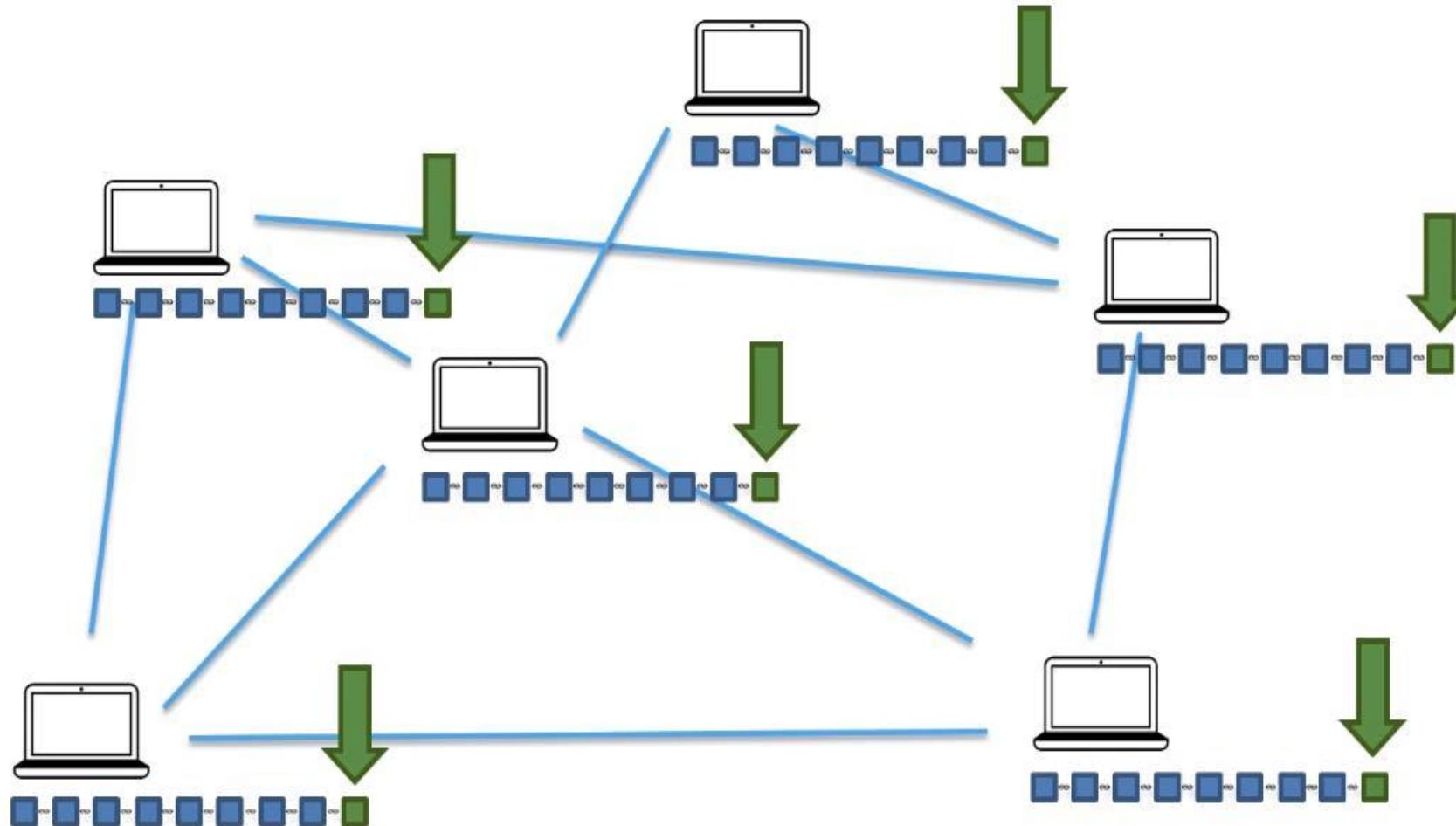
Distributed P2P Network

- Assume a new node is mined by one miner.
- The new block is added to the Blockchain.
- It is also broadcast on the distributed P2P network.



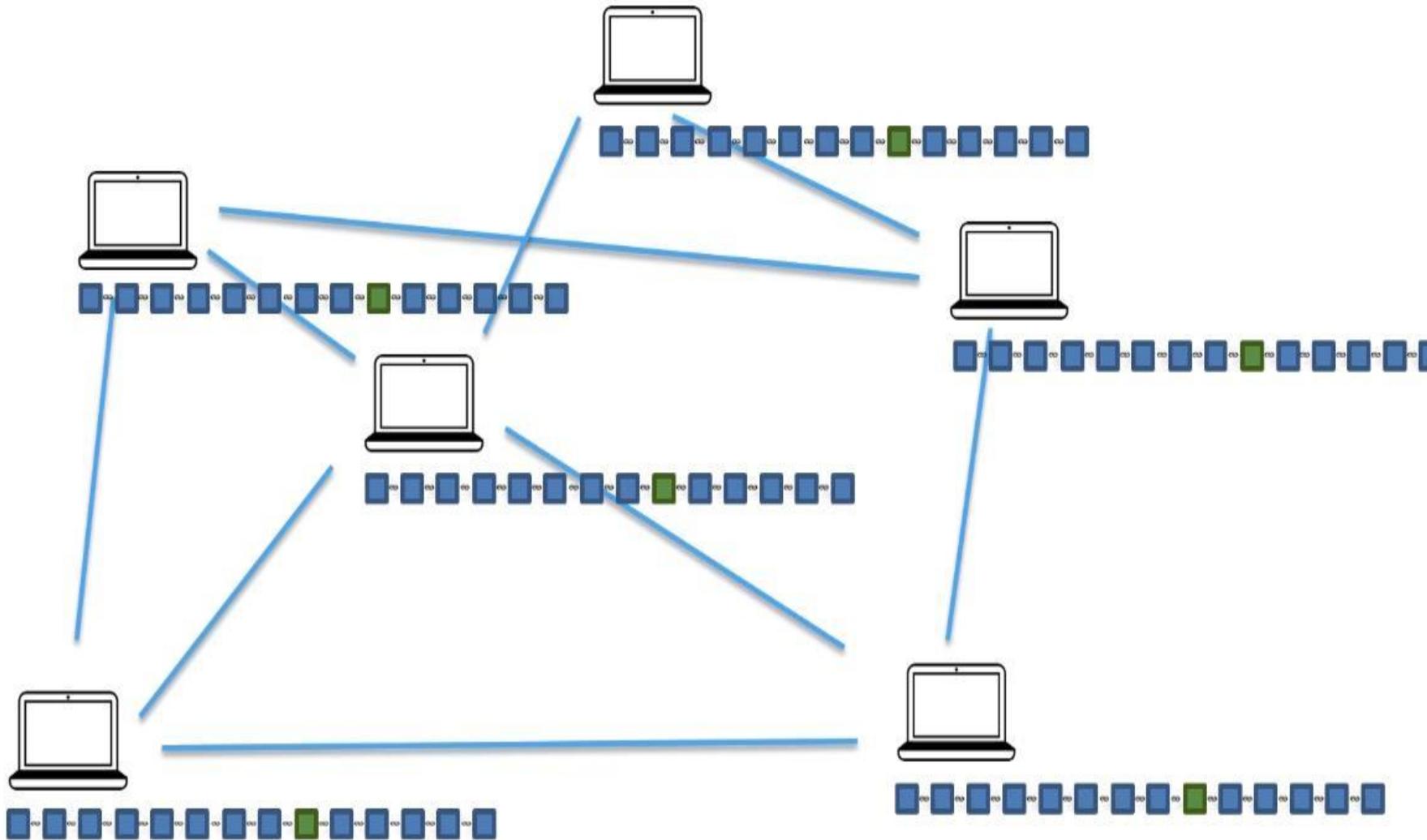
Distributed P2P Network

- Assume a new node is mined at one computer.
- The new block is added to the Blockchain.
- It is also broadcast on the distributed P2P network.
- If consensus is reached, the block will be added to all nodes of the network.



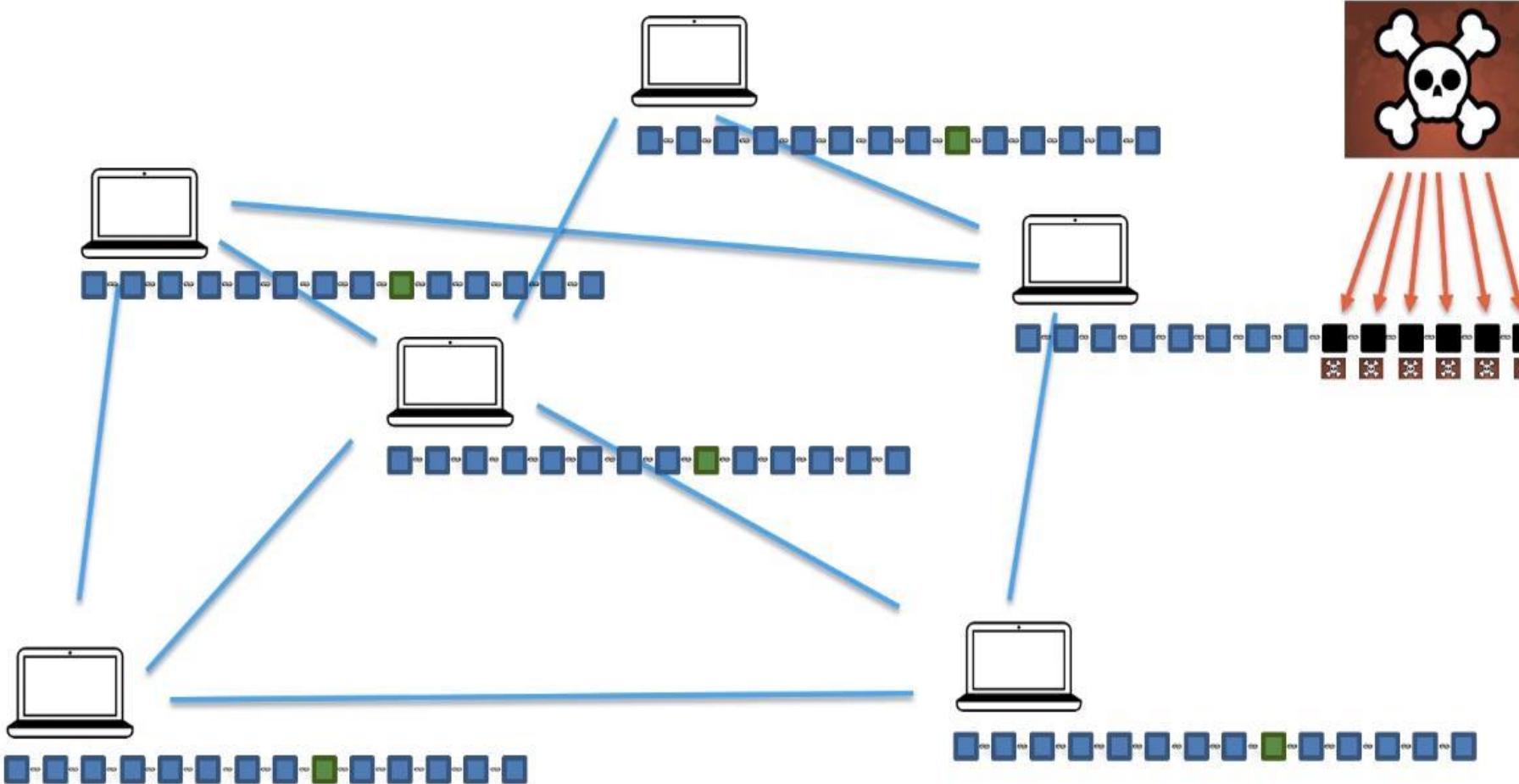
Distributed P2P Network

- Assume that after block β , many nodes have been mined and added to the Blockchain on the distributed P2P network.



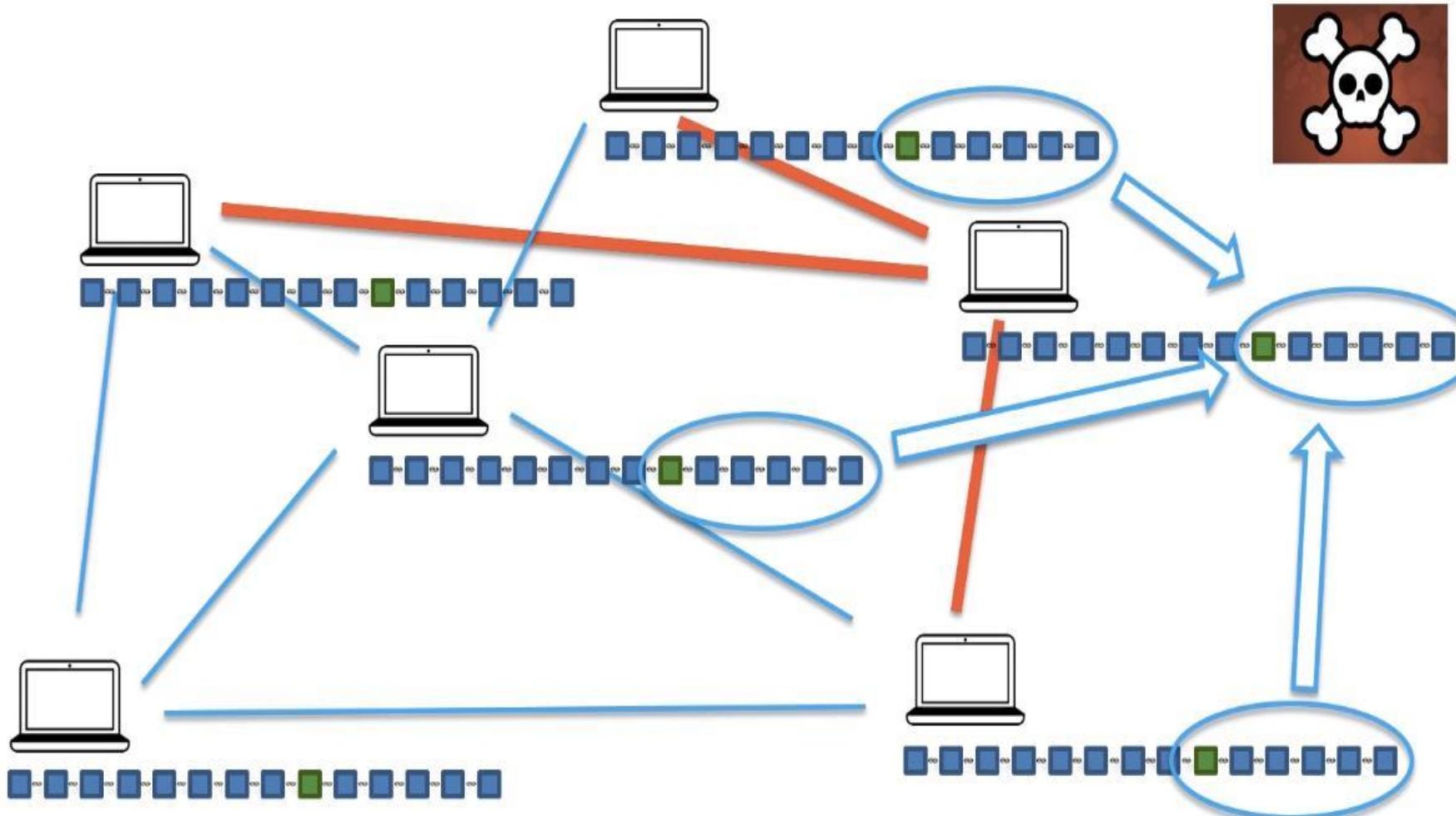
Distributed P2P Network

- If the block β is changed accidentally or intentionally.
- On discovery of a new block or periodically, the system checks the consistency.



Distributed P2P Network

- On discovery of a new block or periodically, the system checks the consistency.
- The inconsistent Blockchain will be copied across the network.



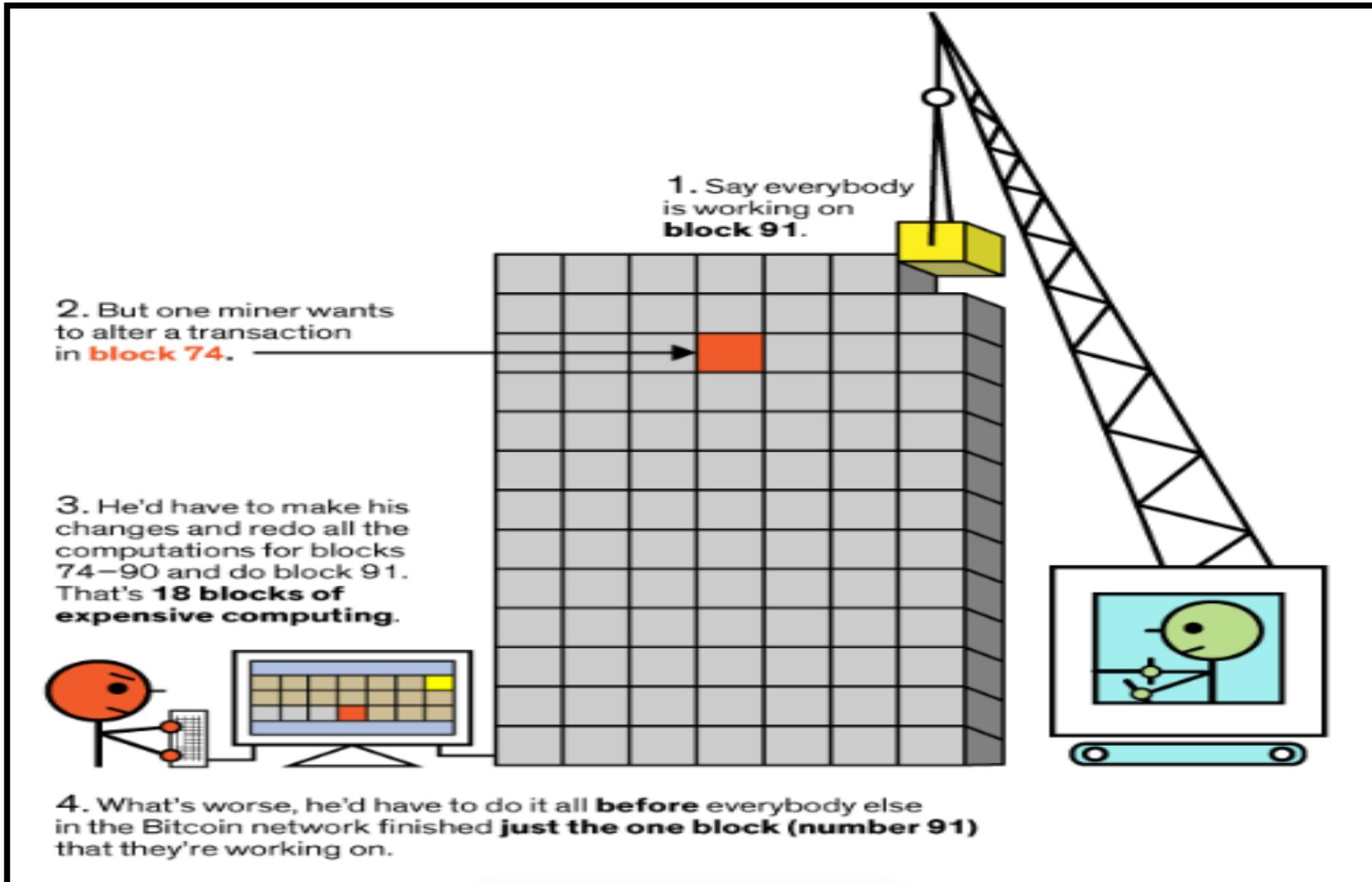
Distributed P2P Network

- It is required to change the Blockchain Ledger across the distributed P2P network for a single data change in the old block.
- The above task must be completed before the discovery of the new block.
- It must be completed in time which is basically the average time to add a block in the Blockchain.
- It is practically impossible.
 - It represents the immutable property of the Blockchain.

<https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274>

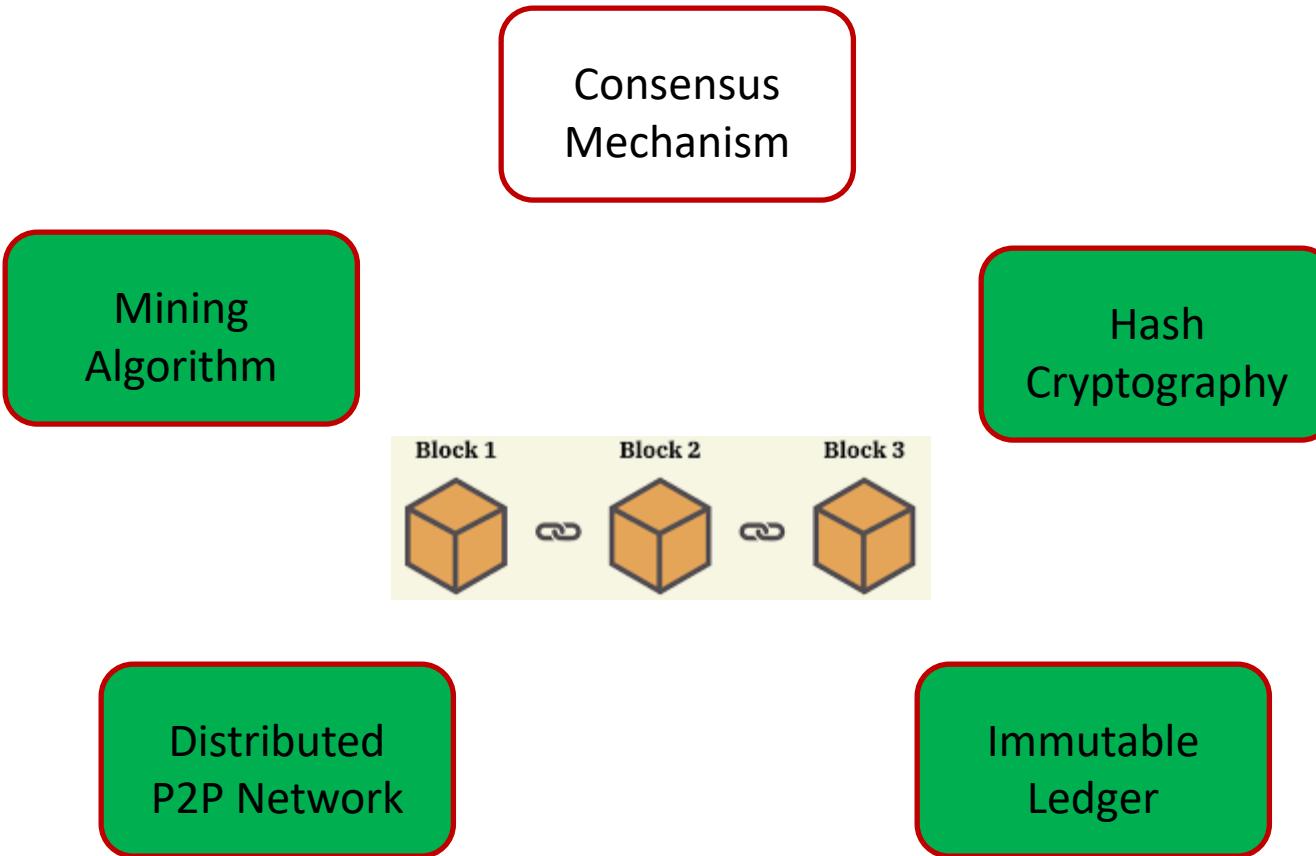
<https://medium.com/fluree/immutability-and-the-enterprise-an-immense-value-proposition-98cd3bf900b1>

Why you can't cheat at Bitcoin ?



<https://medium.com/loom-network/understanding-blockchain-fundamentals-part-2-proof-of-work-proof-of-stake-b6ae907c7edb>

Blockchain Technology: An Intuition

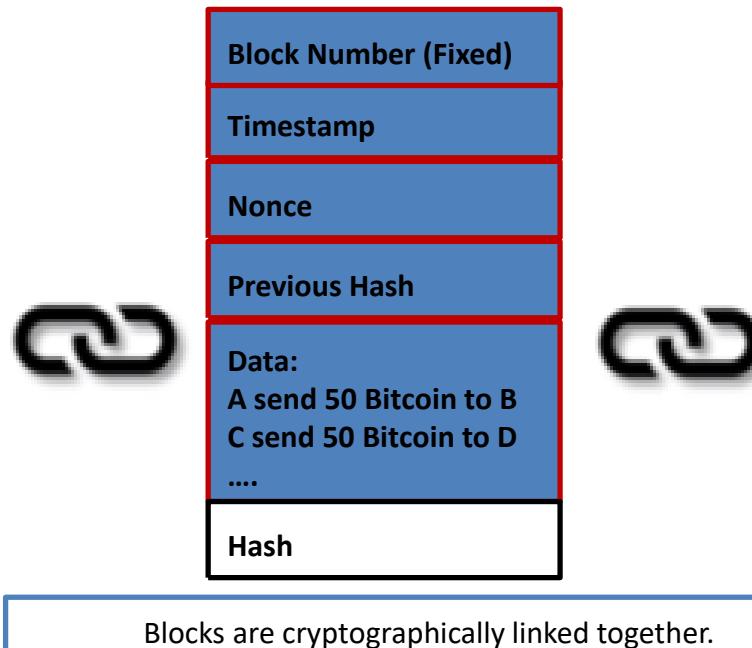


Mining Algorithm

- Blockchain "mining" is the computational work that nodes in the network undertake in hopes of earning new tokens/adding new transactions.
- The miners received rewards/transaction fees for mining the new block.
- They are doing the work of verifying the legitimacy of transactions.
- these rewards/transaction fees are meant to keep blockchain users **honest**.
- By verifying transactions, miners are helping to prevent the "double-spending problem."

Mining Algorithm

- **Block Number:** It depends on the previous block.
- **Previous Hash:** It is the hash value of the previous block.
- **Hash:** It is the hash of the current block which is generated using some algorithm (SHA-256) and follows some threshold value (Leading to four zeros).
- **Nonce:** In cryptography, a nonce is an arbitrary number that can be used just once in a cryptographic communication. It is often a random or pseudo-random number.
- **Data:** It consists of transactions chosen from mempools.
- **Timestamp:** It represents current time taken from https://time.is/Unix_time



Mining Algorithm

▪Nonce

- The "nonce" in a bitcoin block is a 32-bit (4-byte) field whose value is adjusted by miners so that the hash of the block will be less than or equal to the current target of the network.
- The rest of the fields may not be changed, as they have a defined meaning.
- Any change to the block data (such as the nonce) will make the block hash completely different (due to the avalanche effects of SHA-256).
- Since it is believed infeasible to predict which combination of bits will result in the right hash, many different nonce values are tried, and the hash is recomputed for each value until a hash less than or equal to the current target of the network is found.**
- The target required is also represented as the difficulty, where a higher difficulty represents a lower target. As this iterative calculation requires time and resources, the presentation of the block with the correct nonce value constitutes proof of work.
- The value of nonce varies from 0 to $(2^{32}-1)$, i.e., 4, 294, 967, 296.

▪Golden Nonce

- A golden nonce in Bitcoin mining is a nonce that results in a hash value lower than the target.

Mining Algorithm

- Hash

- A hash is a Hexadecimal number that is generated using SHA-256

(FABCDE0123456789FABCDE0123456789FABCDE0123456789FABCDE01234567
89)₁₆

=

(113411911612358349062347011098575490438170457584601795556000509
562573574989705)₁₀

=

(11111010101110011011110000000010010001101000101011001111000100
111111010101111001101111000000001001000110100010101100111100010
0111111010111100110111100000000100100011010001010110011110001
0011111101011110011011110000000010010001101000101011001111000
1001)₂

Mining Algorithm

Largest Hash: FFFF..... FFFF

Smallest Hash: 0000..... 0000

FABCDE0123456789FABCDE0123456789FABCDE0123456789FABCDE0123456789

FABC FABCDE0123456789 FABCDE0123456789 FABCDE0123456789 FABCDE012345

000000000000000000006789DE0123456789FABCDE0123456789FABCDE012345

All Possible Hashes

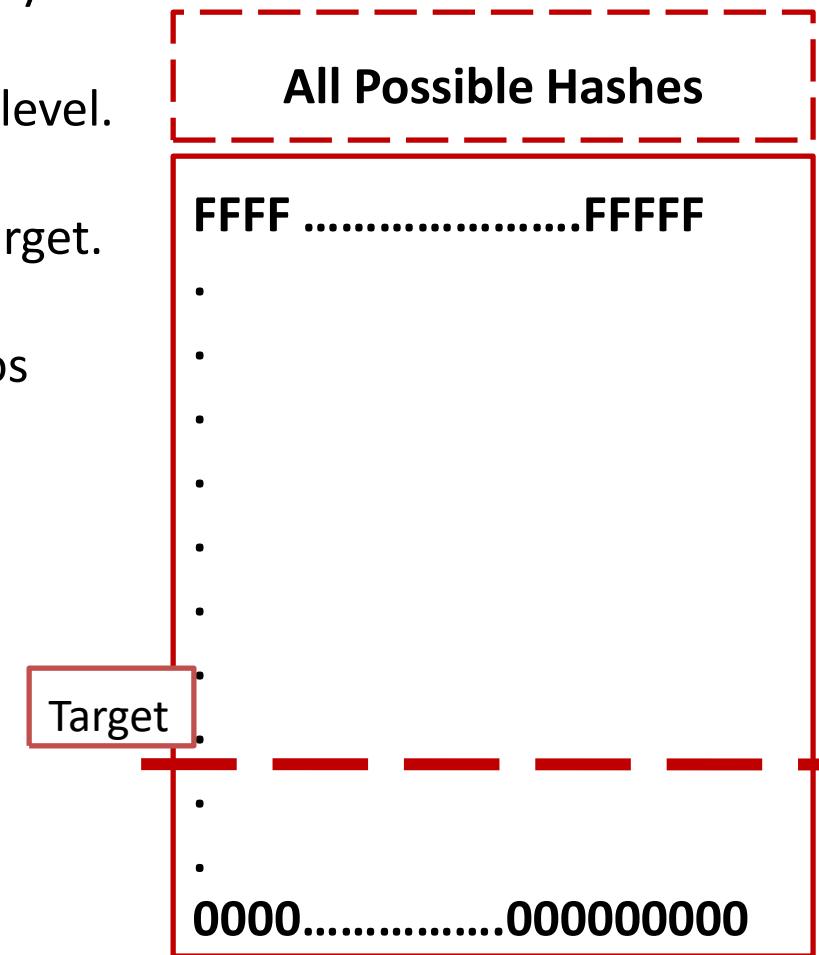
FFFF **FFFFF**

• • • • • • • • •

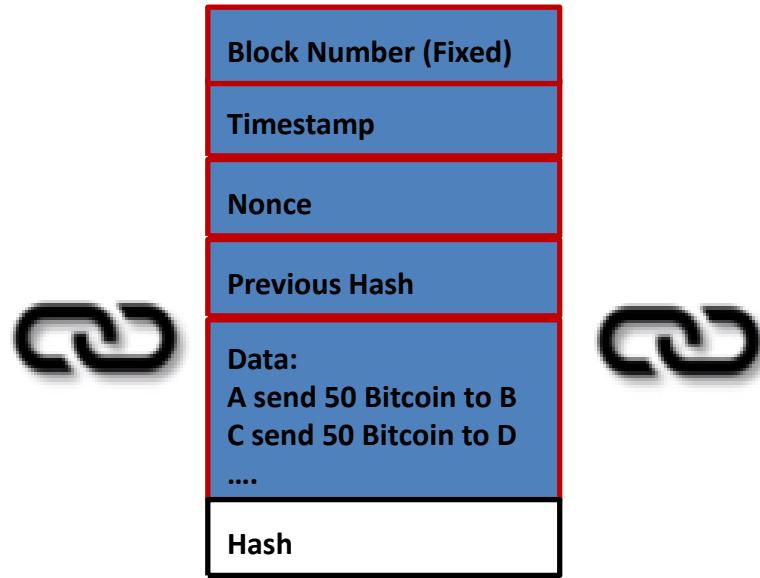
0000.....00000000

Mining Algorithm

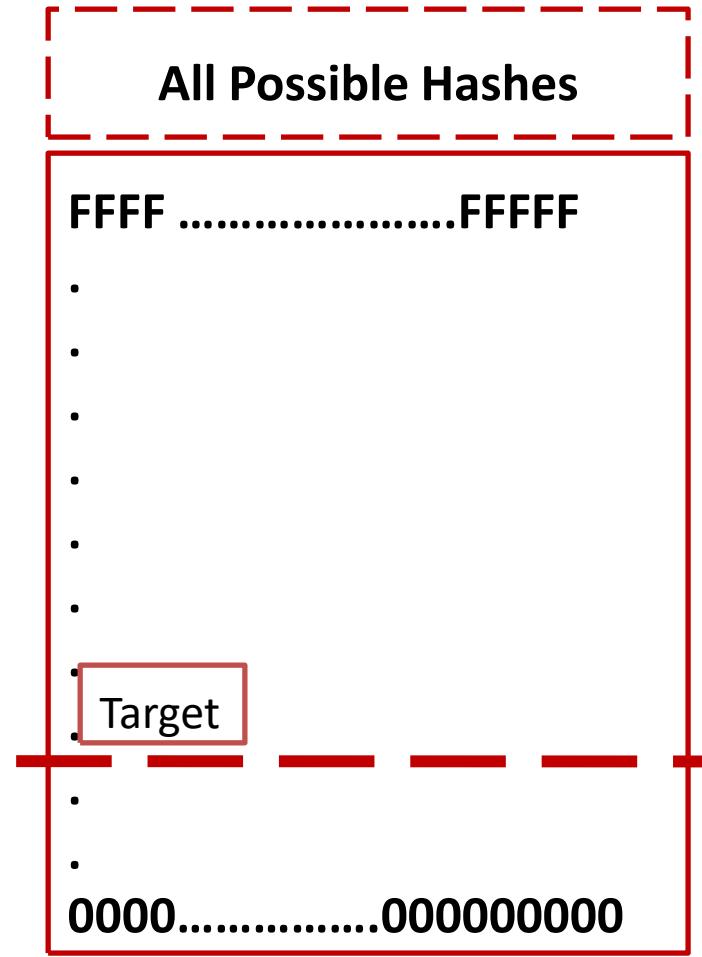
- Cryptographic puzzle:
 - Find a hash value having some leading zeros.
 - Very difficult to solve but very easy to verify.
- Number of leading zeros defines the difficulty level.
 - Fewer zeros represent a very easy target.
 - More leading zeros represent a difficult target.
- If the current target consists of 12 leading zeros
 - 00000123....ABDFE -Invalid
 - 00000000000123905556....12 -Valid
- Why Cryptographic puzzle is a difficult problem ?



Mining Algorithm

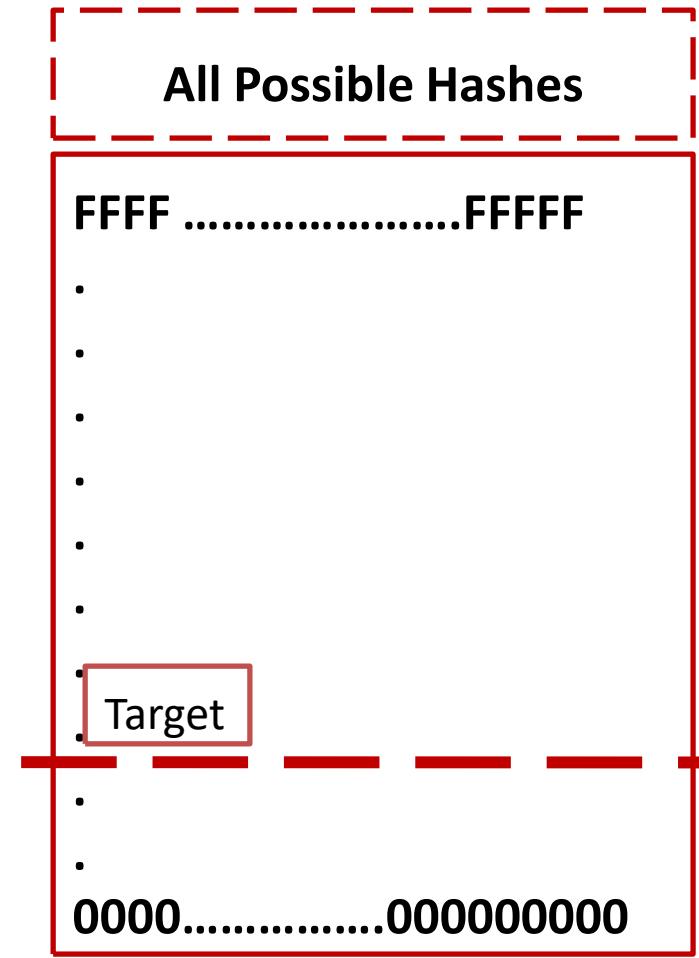
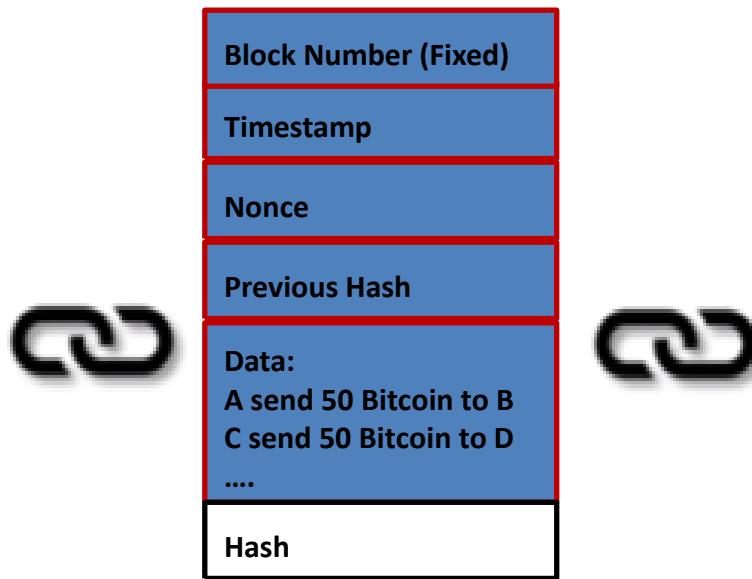


- Only nonce can be changed which is from 0 to $2^{32}-1$.
- Nonce and Hashes are randomly linked and generated using SHA-256



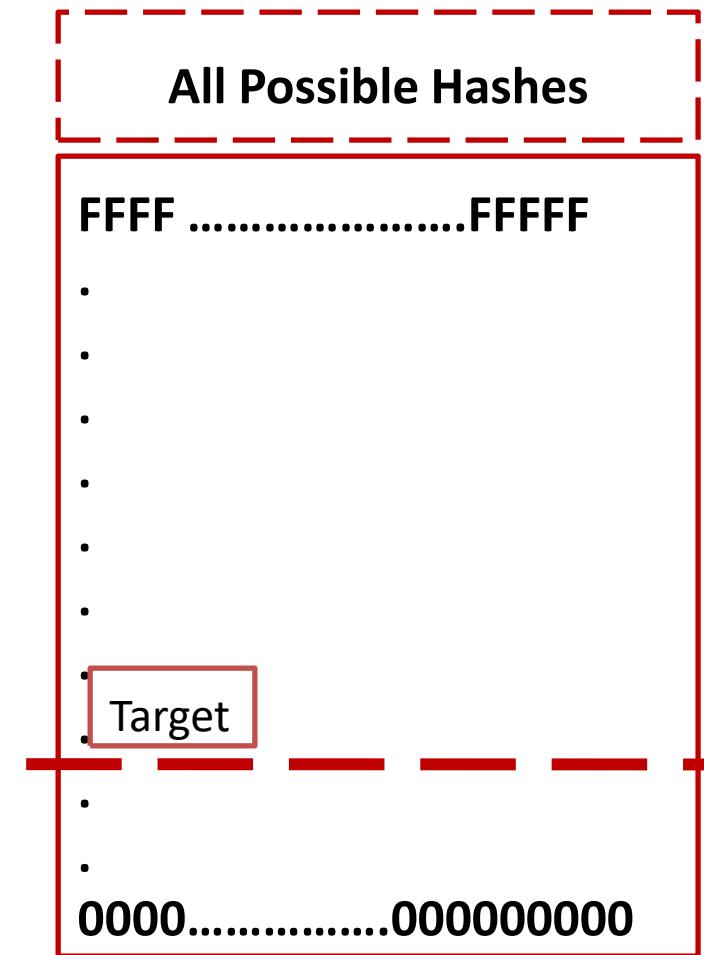
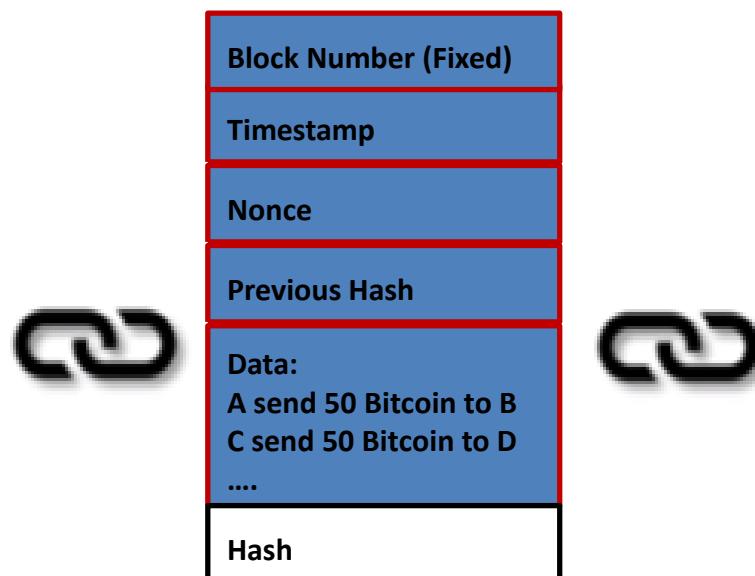
Mining Algorithm

- The value of Nonce will be changed.
- If the hash value generated using SHA-256 is below the current target, then the miner wins.
- The nonce will be known as Golden Nonce.
- This is the proof that he has done enough work (PoW).
- The miner is allowed is add new block in the Blockchain.



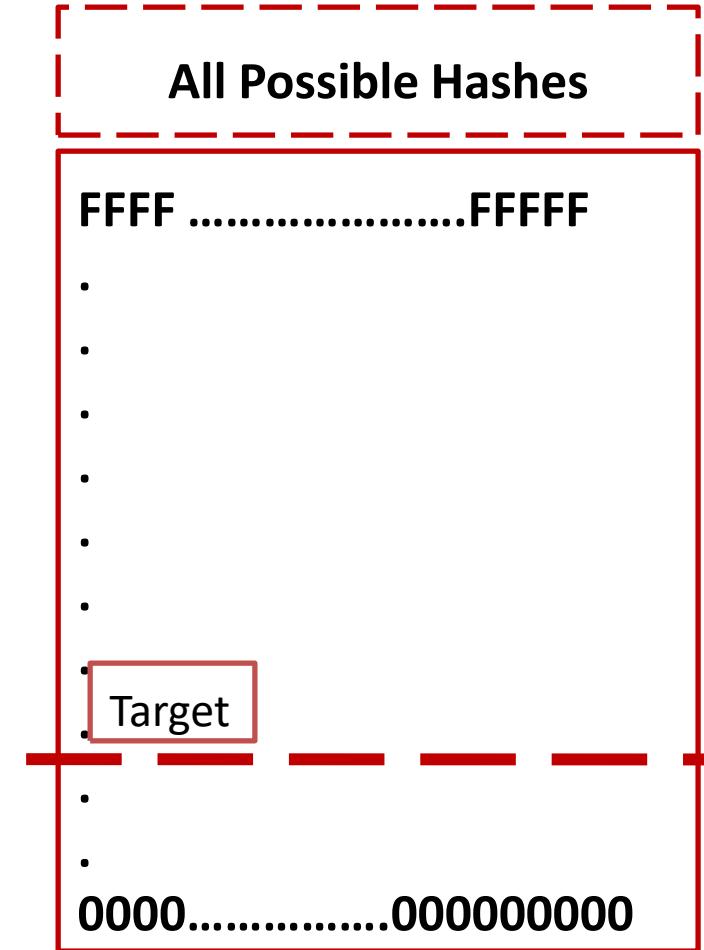
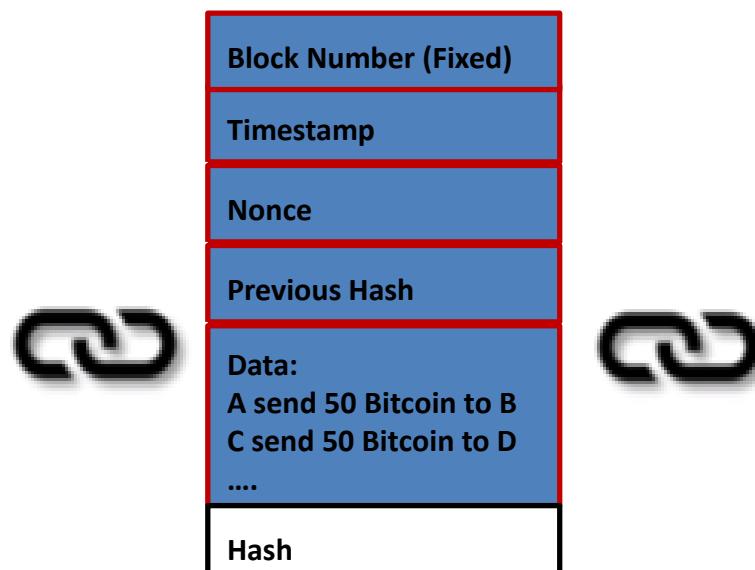
Mining Algorithm

- If the hash value changed from 21 to 22, the SHA-256 value will be very much different due to Avalanche effects.
- It makes the hash value unpredictable and makes the mining process very difficult.

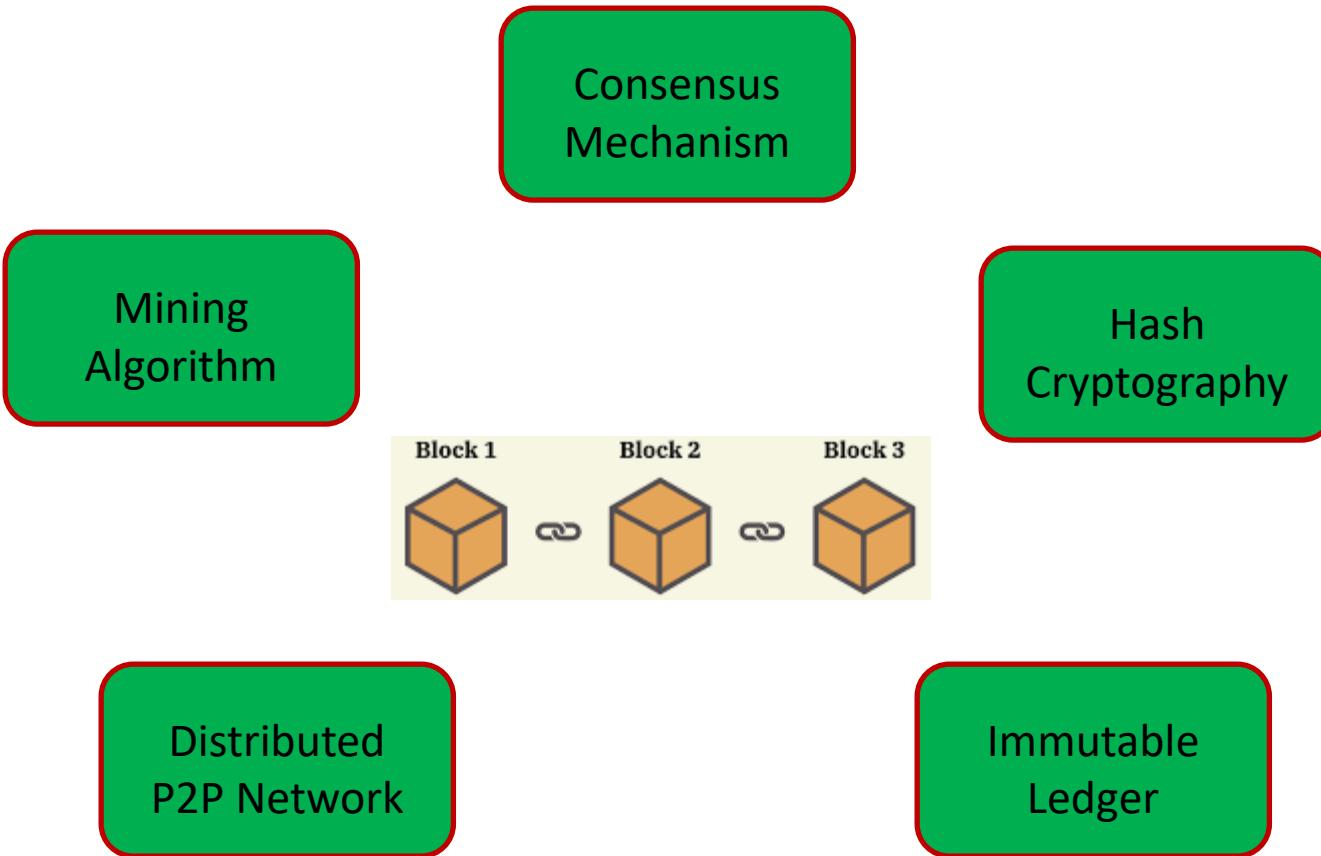


Mining Algorithm

- If the Golden Nonce is not found
 - Change data/time.
 - Start the process again, i.e., solve the cryptographic puzzle again.



Blockchain Technology: An Intuition



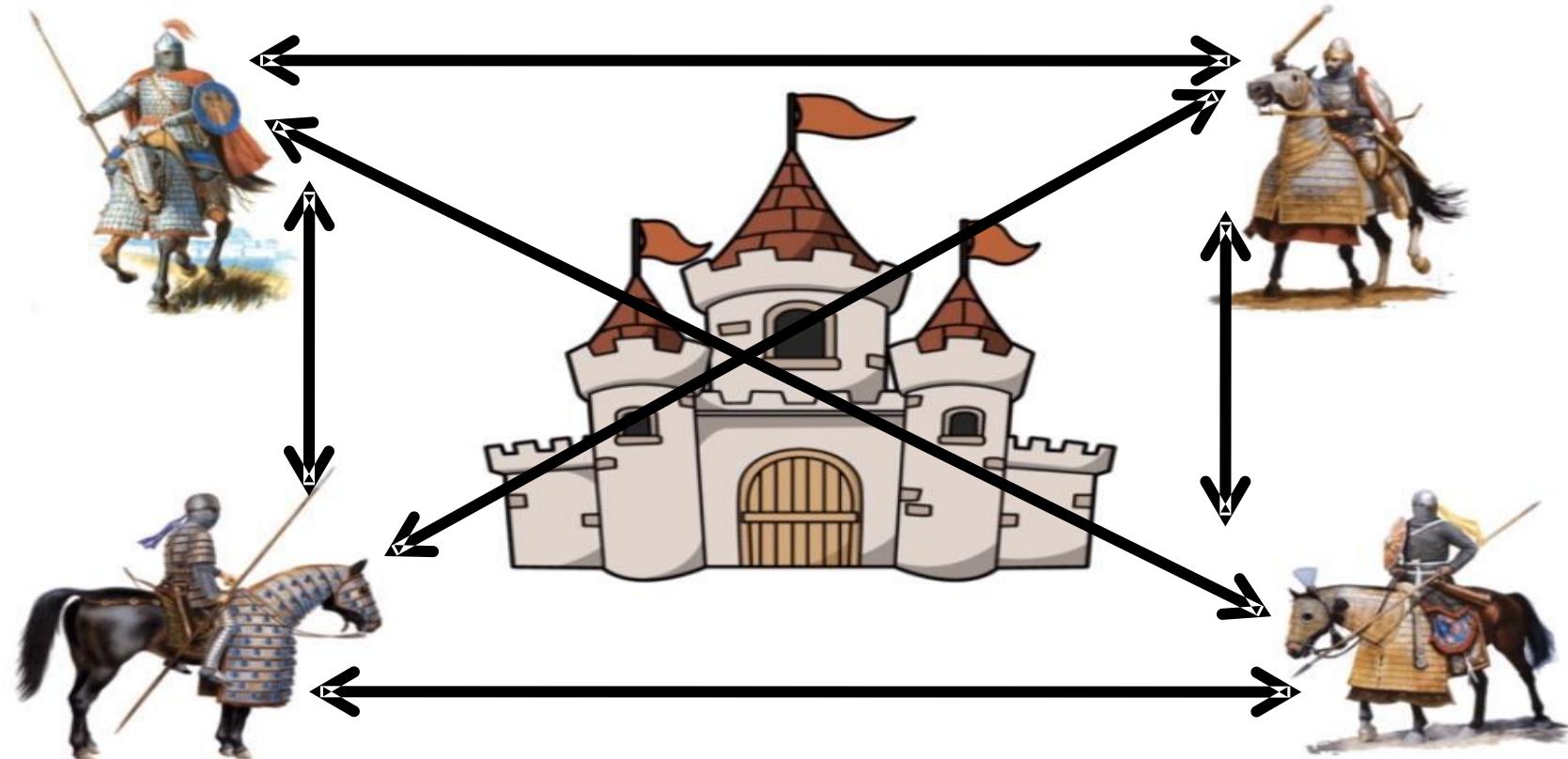
Byzantine Generals Problem

- The Generals of the Byzantine Army are interested to attack a castle.
 - If the majority of Generals are ready to attack, they will win.
 - If the majority of Generals are ready to retreat, they will be all safe.
 - If they don't have a consensus on attack or retreat, they will be destroyed by the enemy.



Byzantine Generals Problem

- One General is the commander of the Byzantine Army.
 - There might be a traitor (The commander or some other Generals may be traitors).
 - They communicate with each other directly.
- How to achieve consensus on their decisions?**

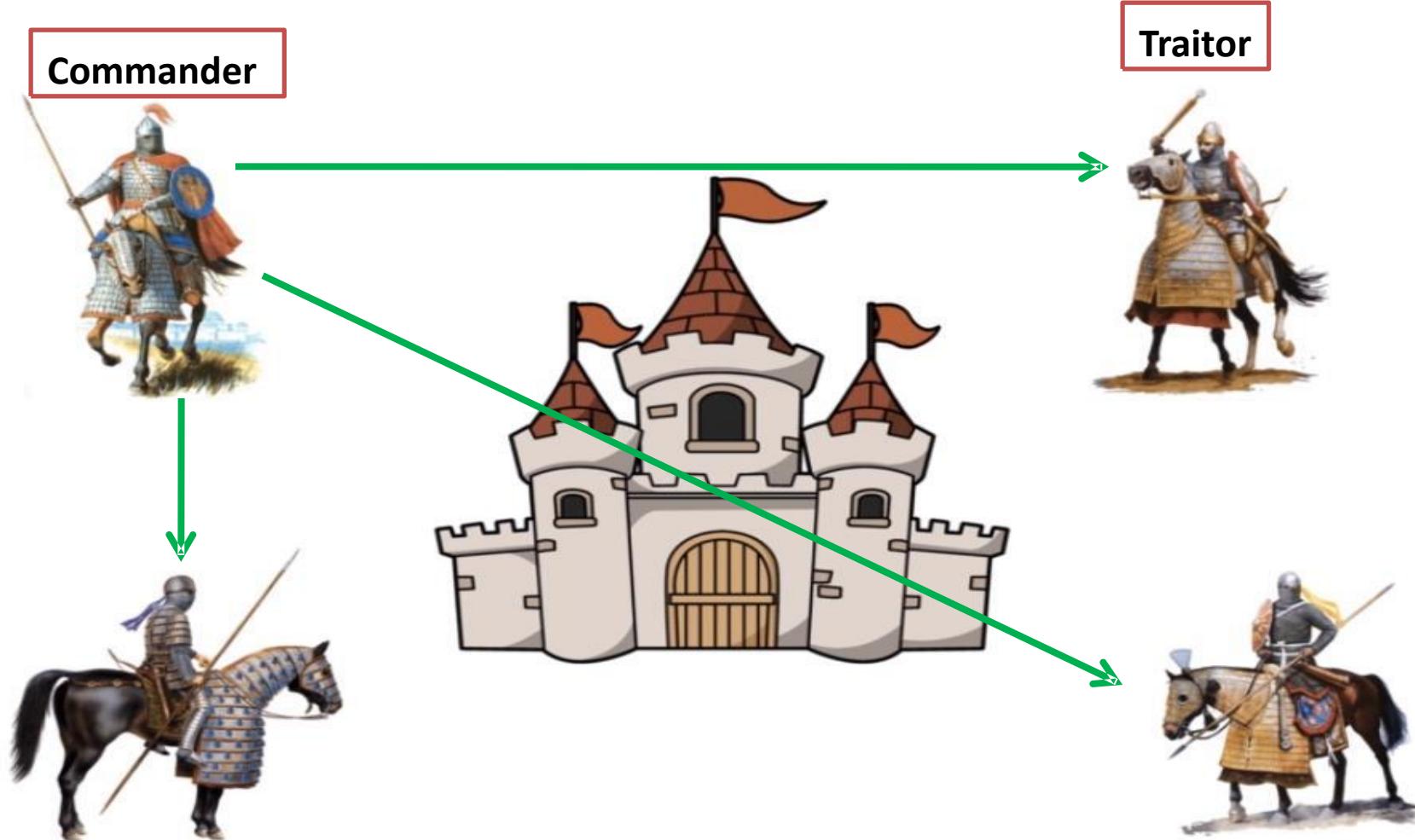


Byzantine Generals Problem

- It is a very complex problem in the distributed environment.

Case 1: The commander is loyal.

Commander decided to attack

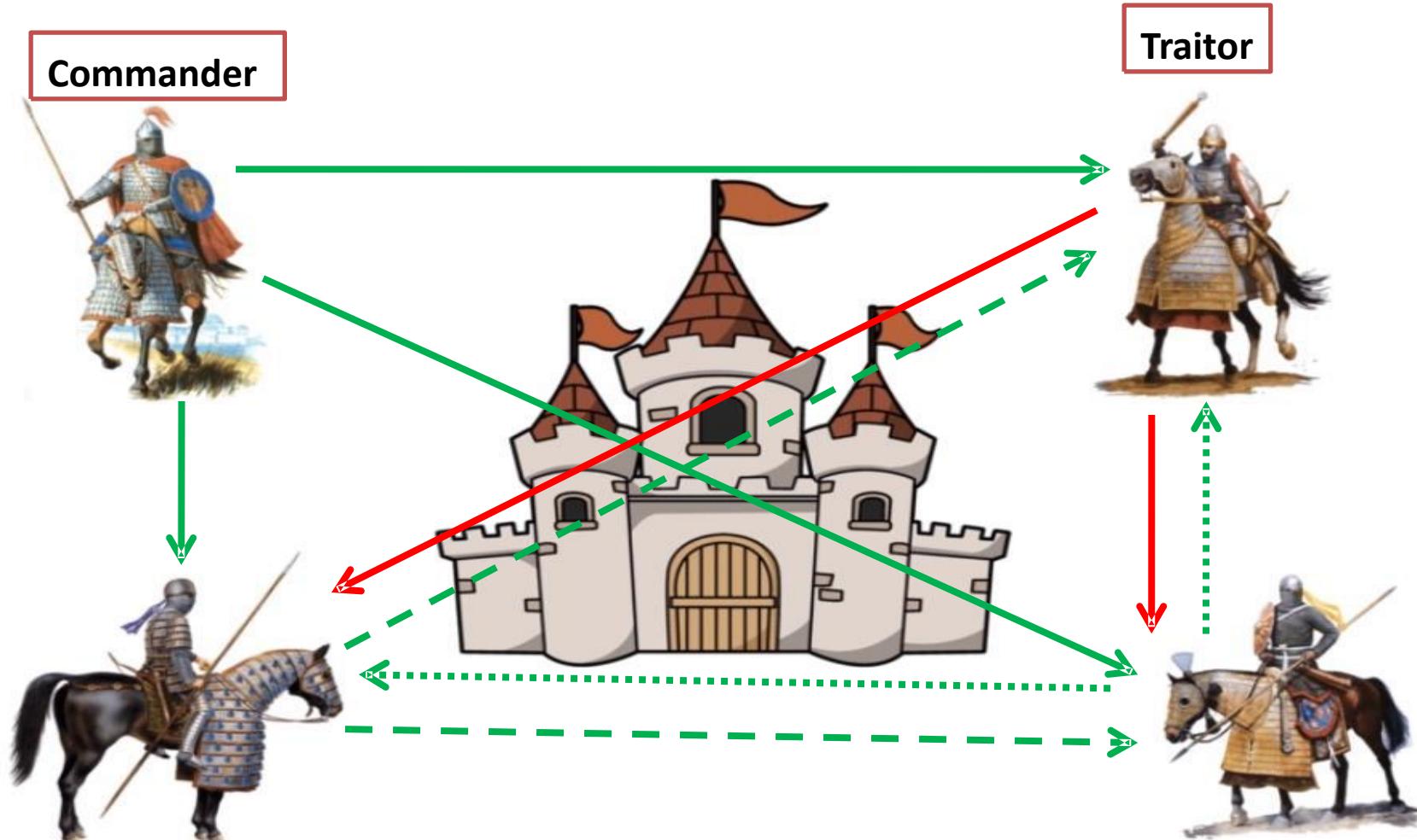


Byzantine Generals Problem

- It is a very complex problem in the distributed environment.

Case 1: The Commander is loyal.

Commander decided to attack



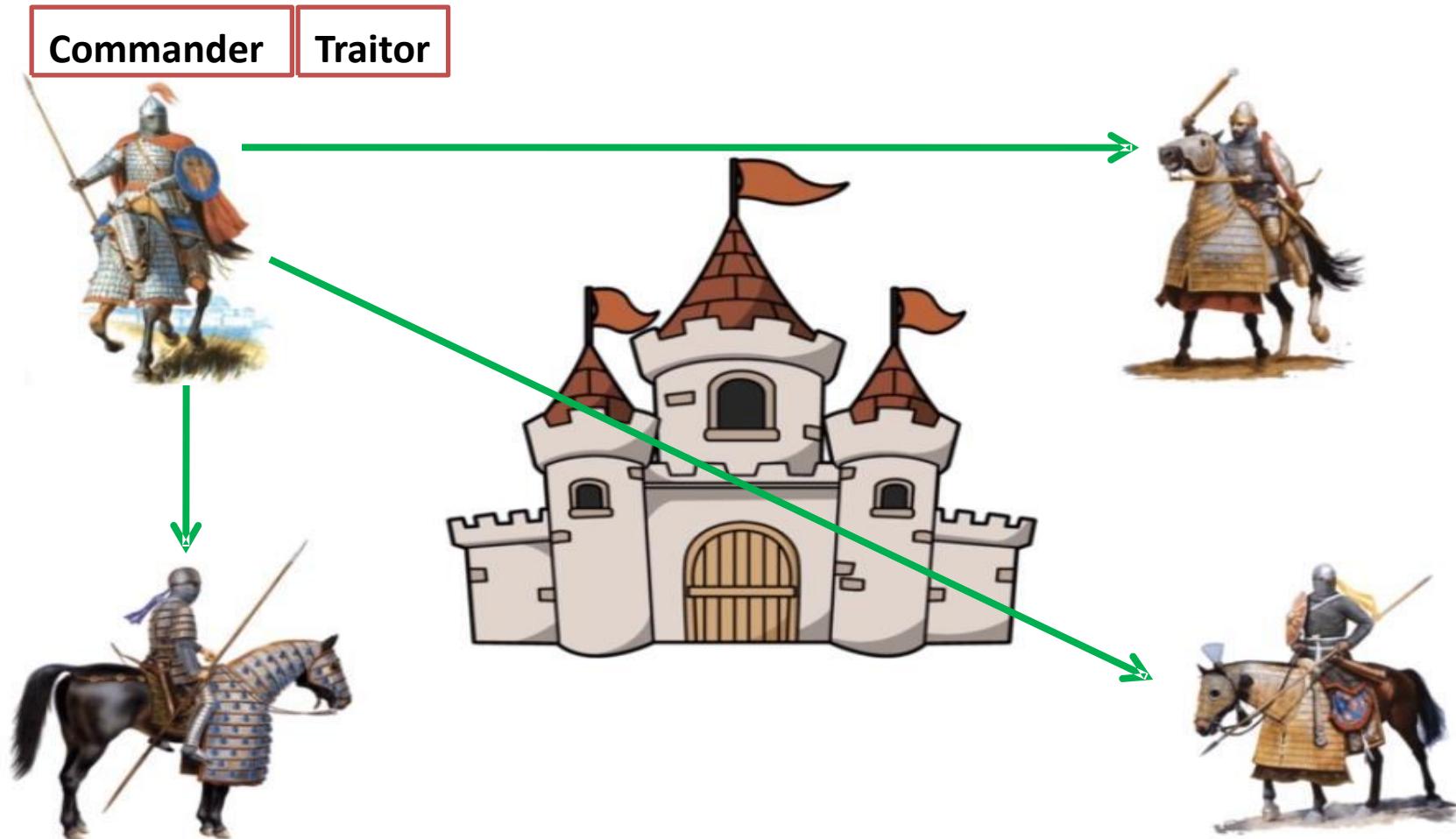
Consensus has been reached on 'Attack'.

Byzantine Generals Problem

- It is a very complex problem in the distributed environment

Case 2: The Commander in traitor.

Will this algorithm work?

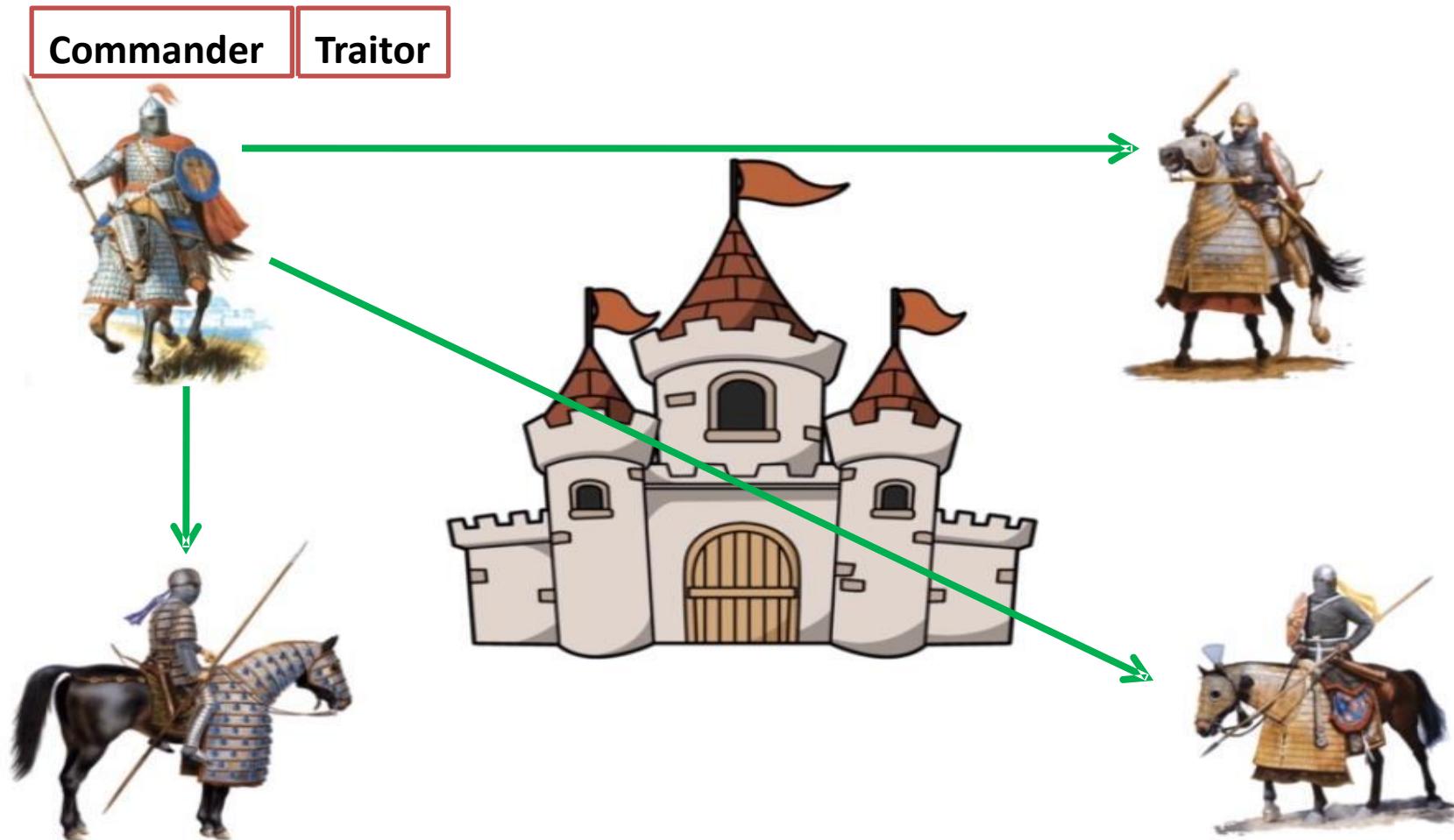


Byzantine Generals Problem

- It is a very complex problem in the distributed environment.

Case 2: The Commander in traitor.

He sends the same message to everyone and not going to do the same.

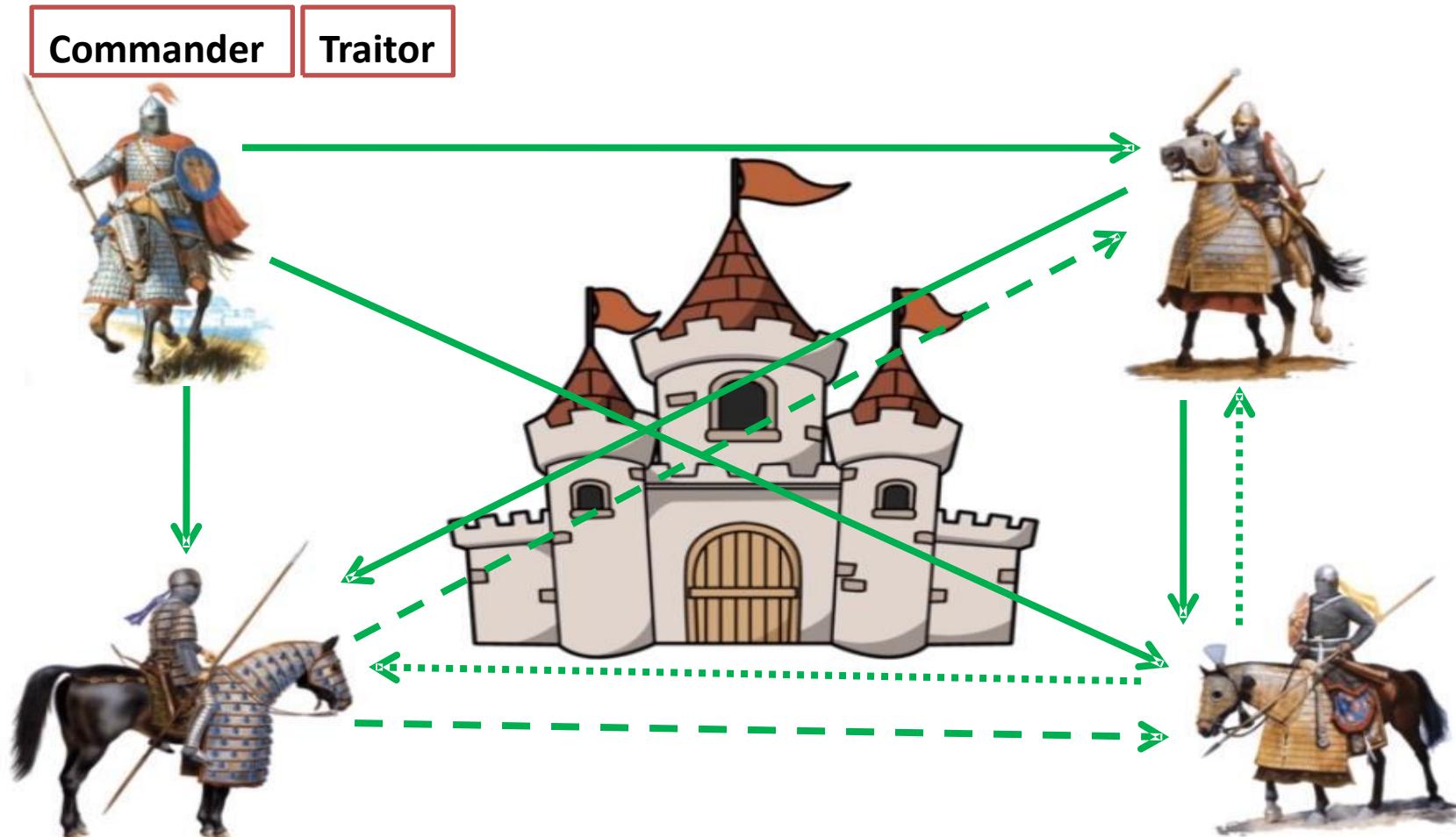


Byzantine Generals Problem

- It is a very complex problem in the distributed environment.

Case 2: The Commander in traitor.

He sends the same message to everyone and not going to do the same.



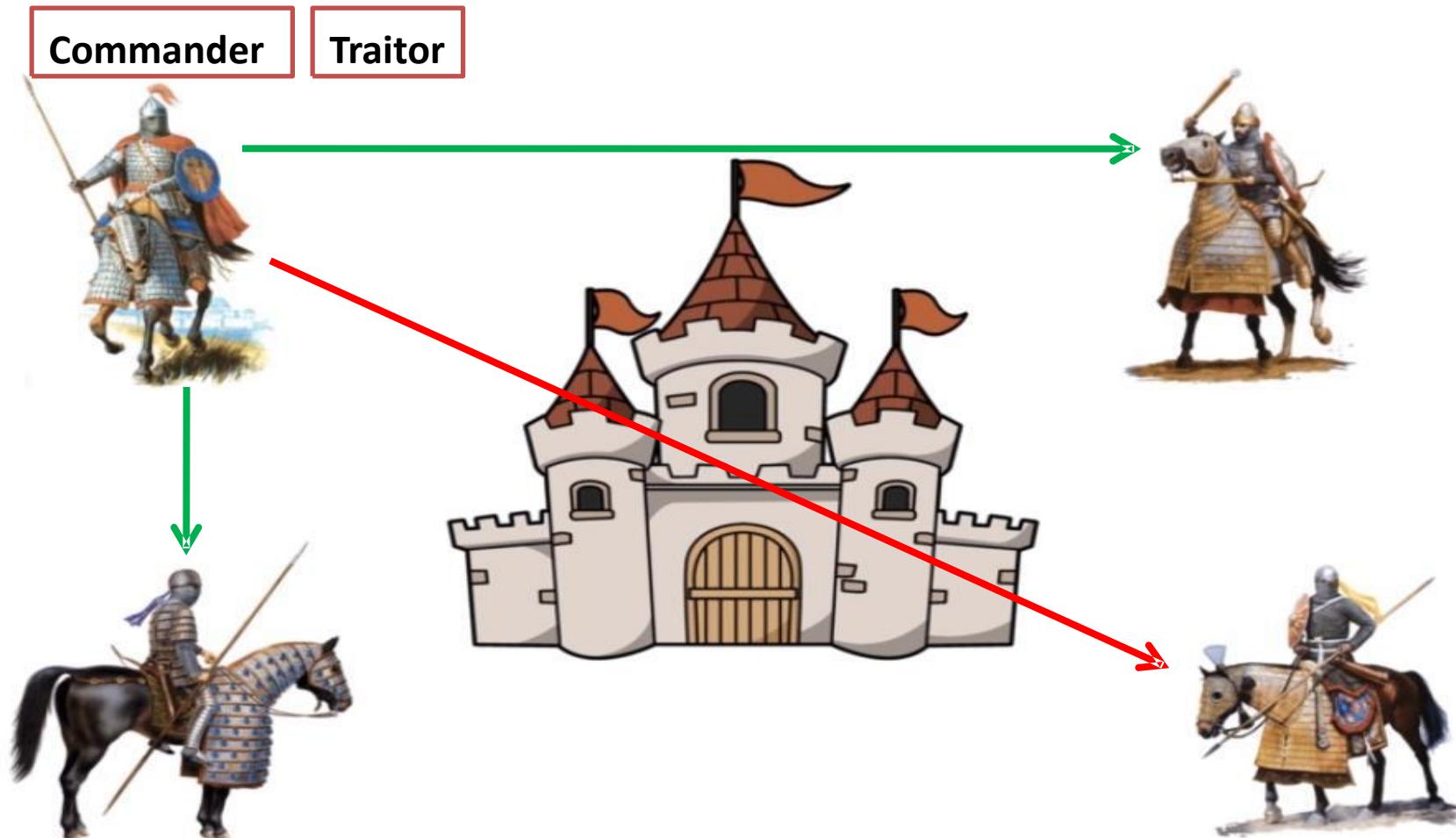
Consensus has been reached on 'Attack'.

Byzantine Generals Problem

- It is very complex problem in the distributed environment ?

Case 2: The Commander in traitor.

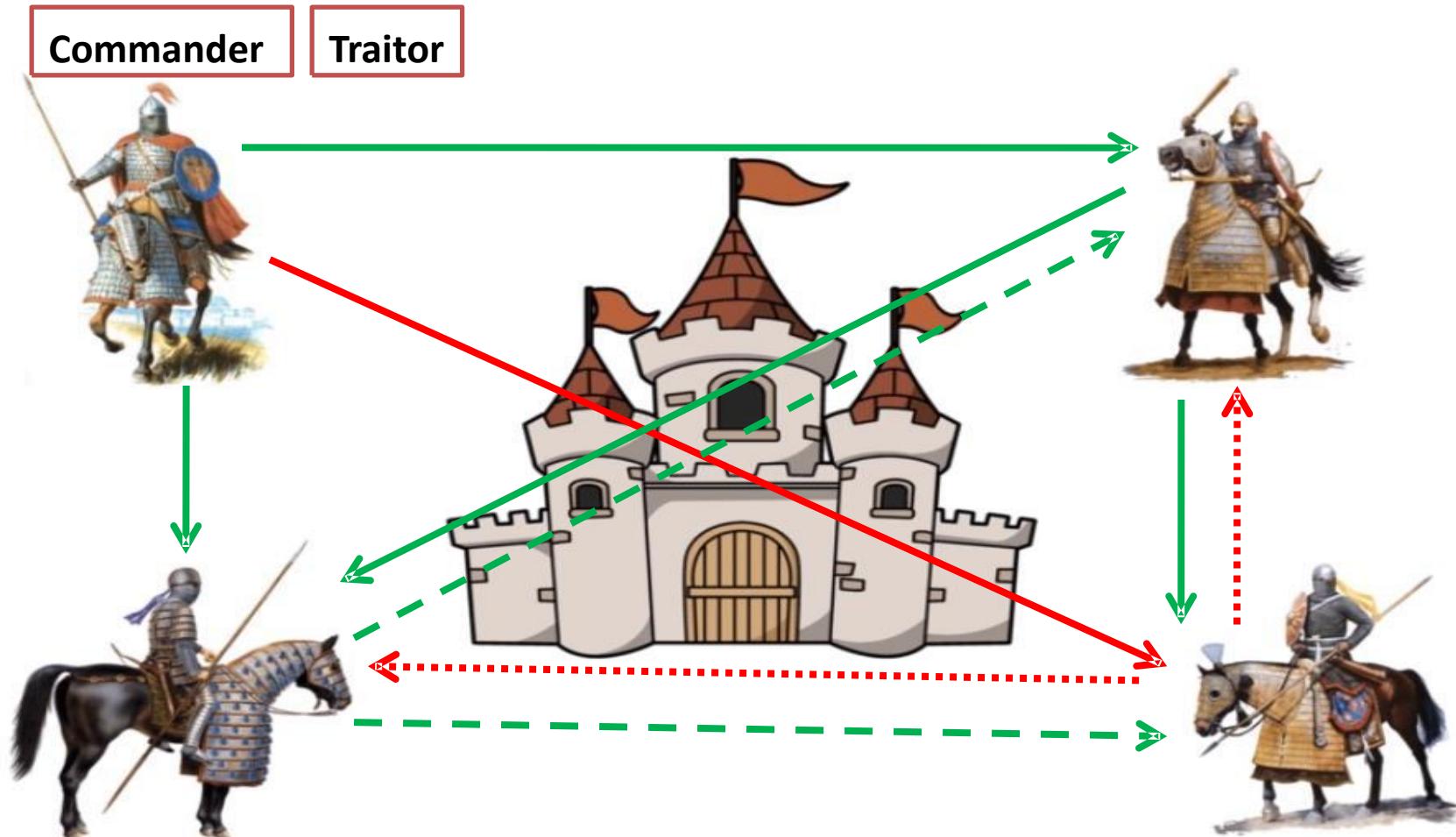
He sends different message to all generals and not going to do the same.



Consensus has been reached on 'Attack'.

Byzantine Generals Problem

- It is a very complex problem in the distributed environment.
- **Case 2: The Commander in traitor.**
He sends different messages to all generals and decides to retreat.



Consensus has been reached on 'Attack'.

Byzantine Generals Problem

- Byzantine Fault Tolerant depends on the majority.

- How much fault it can handle?

Case 3: If they are two traitors, it is not possible to reach a consensus.

Commander



Traitor



Traitor



Byzantine Generals Problem

- **Byzantine Fault Tolerant.**
- How much fault it can handle?
 - If there are 2/3 good nodes, the consensus will be reached.
 - It can handle 33% of malicious nodes.
- A Byzantine fault is a condition of a system, particularly distributed systems, where components may fail and there is imperfect information on whether a component has failed.
- To avoid catastrophic failure of the system, the system's actors must agree on the same strategy, but some of these actors are unreliable.
 - A component (such as a server) can inconsistently appear as both failed and functioning to failure-detection systems, presenting different symptoms to different observers.
 - It is difficult for the other components to declare it failed and shut it out of the network because they need to first reach a consensus regarding which component has failed in the first place.
 - Byzantine fault tolerance (BFT) is the resiliency of a fault-tolerant computer system to such conditions.

Byzantine Generals Problem

- Byzantine Fault Tolerance is a system's ability to continue operating even if some of its nodes fail or act maliciously.
- The objective of Byzantine fault tolerance is to be able to defend against failures of system components with or without symptoms that prevent other components of the system from reaching an agreement among themselves, where such an agreement is needed for the correct operation of the system.
- Applications
 - Airplane System (Assume each general is one component)
 - Nuclear Power Plant
 - Etc.

For more explanation

The Byzantine Generals Problem

Leslie Lamport, Robert Shostak, and Marshall Pease (1982)

How Bitcoin works in the context of the byzantine general's problem?

- Blockchains are decentralized ledgers that are not controlled by a central authority.
- Due to the value stored in these ledgers, bad actors have huge economic incentives to try and cause faults.
- Thus, a solution to the Byzantine Generals' Problem for Blockchains is much needed.

- The big breakthrough was when Bitcoin was invented.
- Proof-of-Work as a probabilistic solution to the Byzantine General's Problem as described in depth by Satoshi Nakamoto in the following-
 - <https://www.mail-archive.com/cryptography@metzdowd.com/msg09997.html>

How Bitcoin works in context of the byzantine generals problem?

- Blockchains use consensus algorithms to elect a leader who will decide the contents of the next block.
 - Proof-of-Work
 - Cryptographic Puzzle (Difficult to solve, but very easy to verify)
- That leader is also responsible for broadcasting the block to the network so that the other peers can verify the validity of its contents.
- **The cryptographic puzzle is hard for the proposing general but easy for other generals to review. (????)**
- All the generals already know the structural configuration of the solution.
- Now, on the battlefield, suppose a general fabricates a plan and wants to send the message to others to attack a particular day and time along with the plan.

How Bitcoin works in context of the byzantine generals problem?

- Now, on the battlefield, suppose a general fabricates a plan and wants to send the message to others to attack a particular day and time along with the plan.
- The steps will be as follows:
 - He will append a nonce to the original plan.
 - He will then hash the plan appended with the nonce and see the result. He will then keep adding the nonce and checking if he got the desired hash.
 - He will then send the messengers to other relevant generals. Even if the messengers get caught, any modification to their message will lead to an entirely different hash, which the other generals can easily identify.
 - All others finally verify the plan and act accordingly.

Georgios Konstantopoulos

Understanding Blockchain Fundamentals, Part 1: Byzantine Fault Tolerance

<https://www.hcltech.com/blogs/byzantine-fault-tolerance-bft-and-its-significance-blockchain-world>

How bitcoin works in context of the byzantine generals problem?

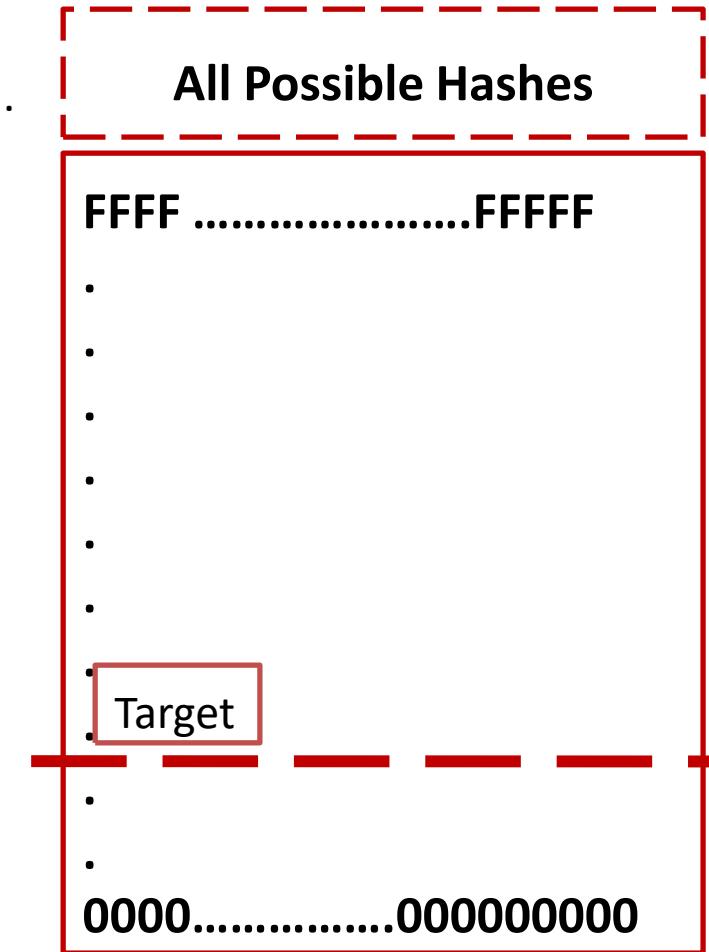
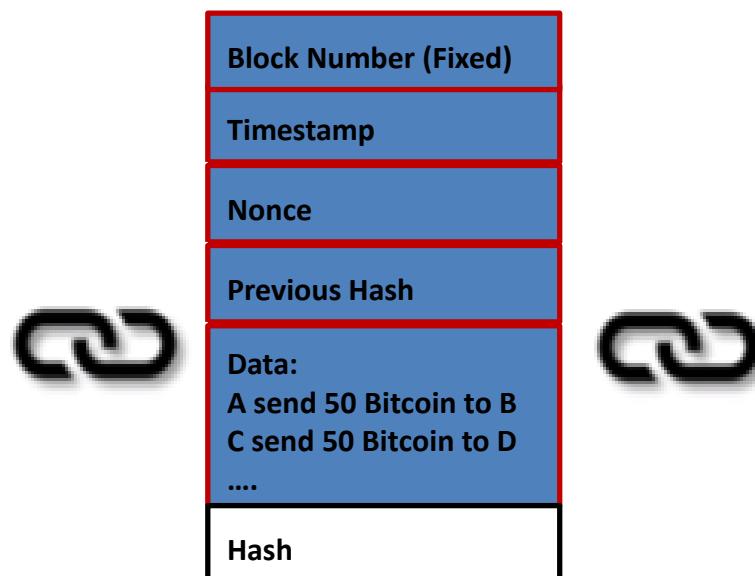
- Byzantine fault tolerance is 50% assuming zero network latency.
- It is around 46% (Ethereum) and 49.5% (Bitcoin) fault-tolerant under actually observed conditions, but it goes down to 33% if network latency is equal to the block time and reduces to zero as network latency approaches infinity.
- The solution could fail only if the malicious party captures 50% of the network power.

Consensus Mechanism

- In Blockchain Technology, the purpose of the Consensus mechanism is to deal with two challenges.
 - Attackers
 1. Attackers may add a malicious block at the end of the chain.
 2. Attackers may try to change the data in the old block.
 - Competing Chains
 - Two or more miners can mine the block at the same time.
 - Nothing malicious in this case.
 - Multiple Blockchain may exist in the network.
 - Longest chain wins.

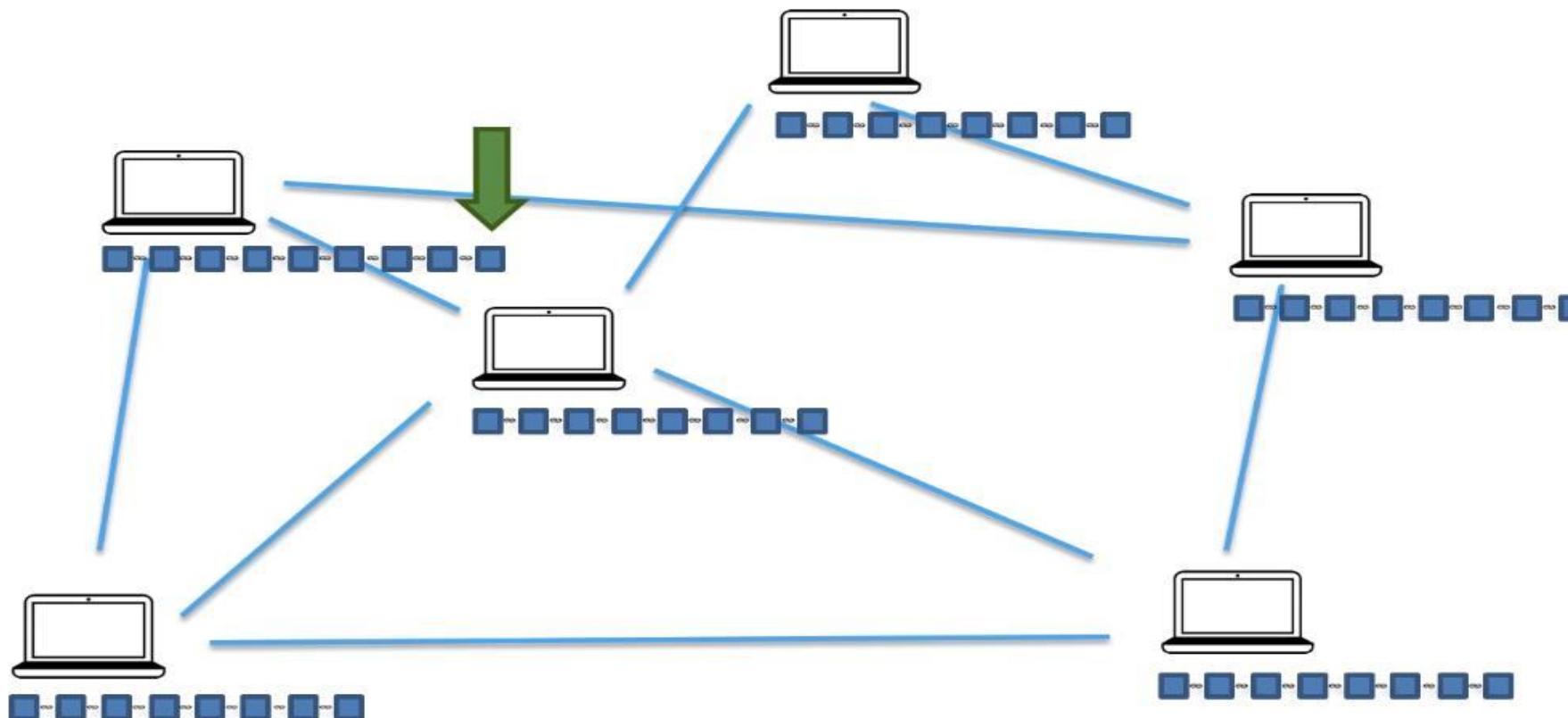
Consensus Mechanism

- The value of Nonce will be varied.
- If the hash value generated using SHA-256 is below the current target, then the miner wins.
- The nonce will be known as Golden Nonce.
- This is proof that he has done enough work (PoW).
- The miner is allowed to add a new block in the Blockchain.



Consensus Mechanism

- After discovering Golden Nonce, the miner is going to add the block to the Blockchain.
- The miner will receive a reward and fee associated with the transactions added to the block.
- Every node in the Distributed P2P network will perform a series of checks before adding the block to the Blockchain.
 - These checks are very rigorous.

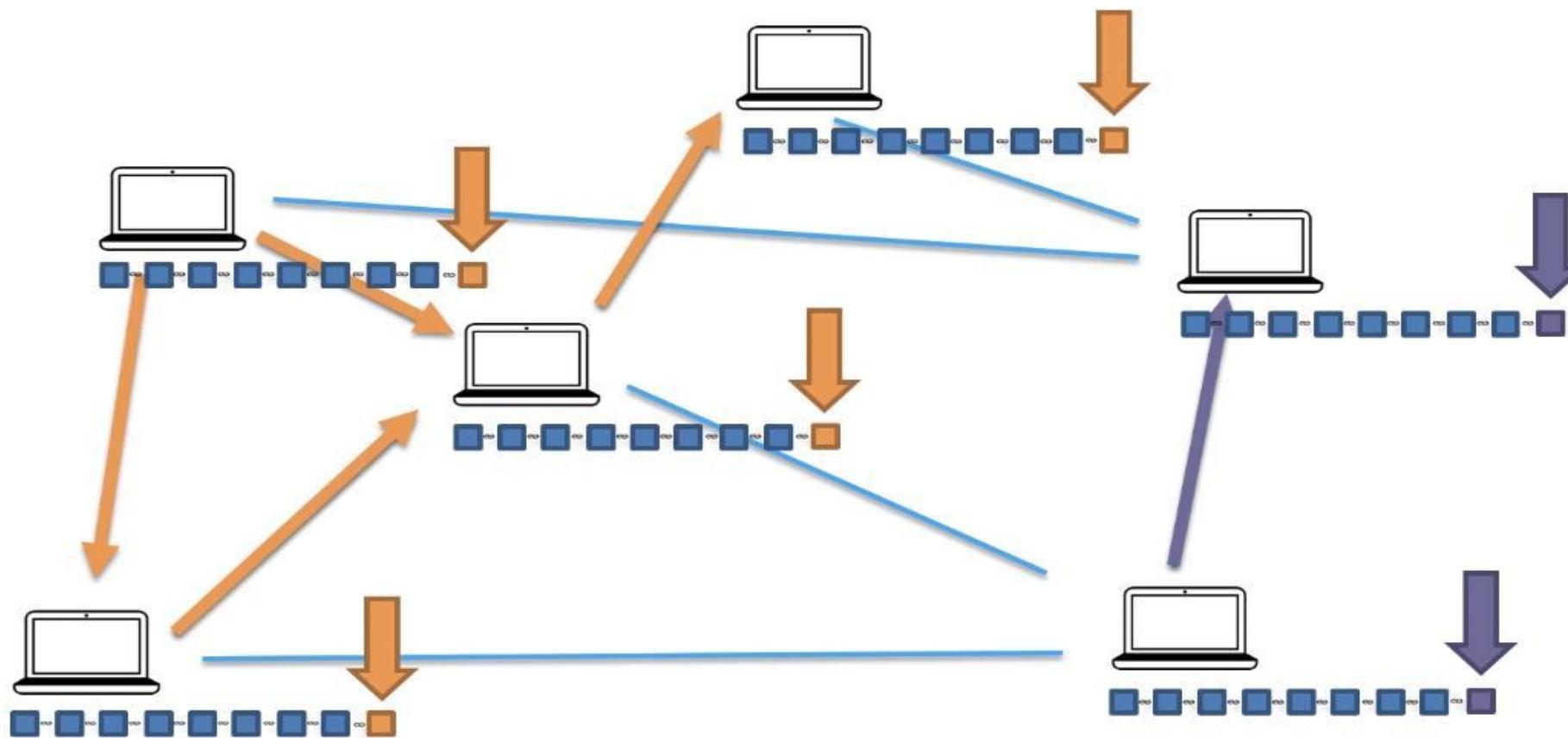


Consensus Mechanism

- Every node in the Distributed P2P network will perform a series of checks before adding the block to the Blockchain.
- These checks are very rigorous. A few of them are-
 - The block data structure is syntactically valid
 - The block header hash is equal to or less than the target (enforces the Proof-of-Work)
 - The block timestamp is less than two hours in the future (allowing for time errors)
 - The block size is within acceptable limits
 - The first transaction (and only the first) is a coinbase transaction
 - All transactions within the block are valid using the transaction checklist.
- If all conditions are fulfilled, the block will be added to the Blockchain, otherwise, the node is rejected.
- It deals with the first challenge (How).
 - There is a list of checks that need to be performed by the majority of nodes.
 - Transactions are verified by the memory pools.

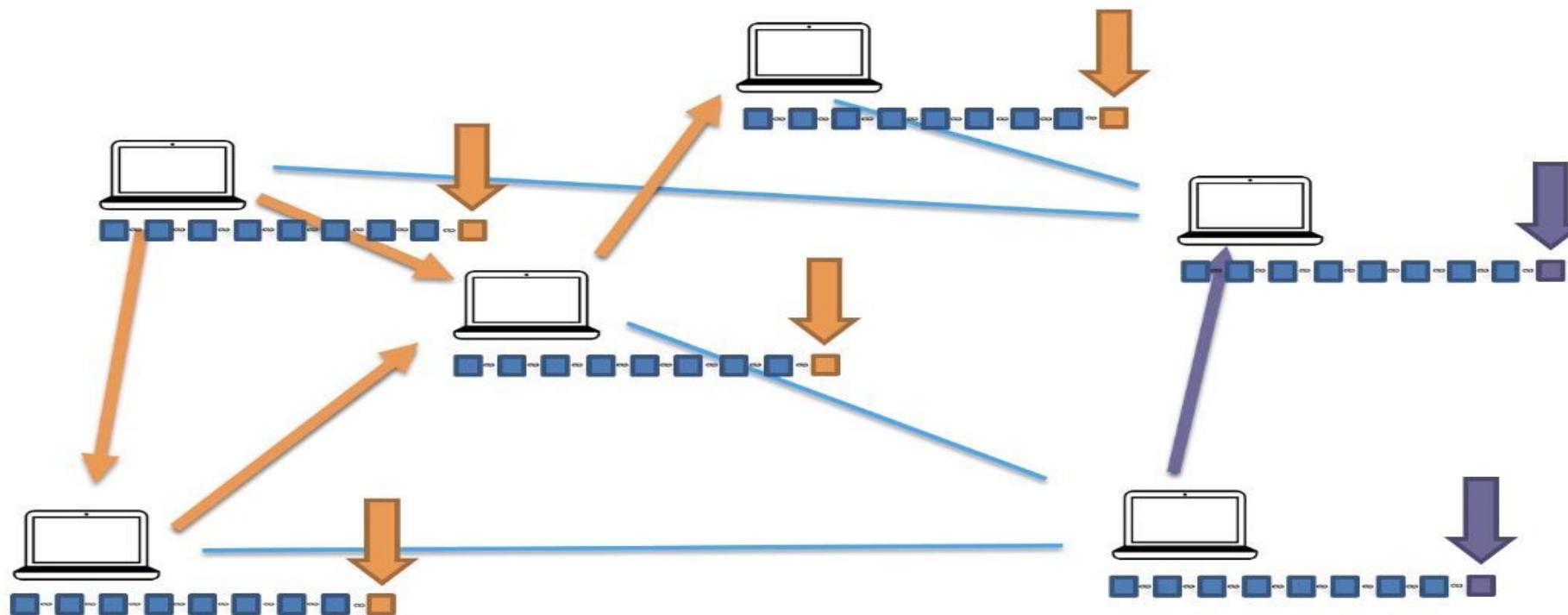
Consensus Mechanism

- Competing Chains
 - Two or more miners can mine the block at the same time.
 - Nothing malicious in this case.
 - Multiple Blockchain may exist in the network
- This is very similar to Byzantine Generals Problem (Two proposals in the network).



Consensus Mechanism

- There exist two competing chains in the network.
 - What is the solution?
- The network continues with multiple competing chains.
- The nodes in the network wait for the new blocks to be added to the Blockchain.
 - The longest chain wins.
 - The part of the network with maximum hashing power will generate the longest chain and wins.
 - Hashing power means the number of hashes generated by the network.



Consensus Mechanism

- The nodes in the network wait for the new blocks to be added to the Blockchain.
- The part of the network with maximum hashing power will add more blocks to the Blockchain.
- It means if you have the part of the network that has 50% Hashing power, it can add the blocks or can control the Blockchain.
 - In BZP, 70% of generals need to agree on one decision.
 - In the case of Blockchain, 51% is enough.
- Thus, the consensus protocol in a Blockchain is more powerful than the consensus protocol in the Byzantine general's problem.
- You need 70% in the Byzantine generals problem and 50% in the Blockchain in order to reach a consensus.

Consensus Mechanism

- What happened to the blocks added in smaller chains?
 - They are orphaned blocks.
 - The reward and transactions are also not valid.
- <https://www.coindesk.com/markets/2017/03/04/a-short-guide-to-blockchain-consensus-protocols/>
- Satoshi Nakamoto Emails List

Signatures: Private and Public Keys

A demo for public or private key.

A demo for signature

A demo for transaction

A demo for block

<https://tools.superdatascience.com/blockchain/public-private-keys/keys>

A demo for hash.

A demo for block

A demo for blockchain

A demo for distributed

A demo for Tokens

A demo for coinbase

<https://tools.superdatascience.com/blockchain/public-private-keys/keys>

BLOCKCHAIN TYPES AND CONSENSUS MECHANISMS



Distributed Ledger

Blockchain is a digital ledger of information. It is-

- **Distributed**
 - Every participating node has a digital copy of the Blockchain database.
 - All nodes can contribute to processing the Blockchain.
- **Maintained by Consensus Algorithms**
 - Mechanism to guarantee that the data are legitimate and not tampered with.
 - Allows transactions only with the agreement of all the relevant parties.
- **Immutable**
 - The records on the Blockchain cannot be changed or deleted.
 - Any subsequent changes are recorded in a new block in the Blockchain.
- **Auditable**
 - Immutable tracking of a record with a timestamp allows for the provenance of the asset at every step.

Distributed Ledger: Trustless Nature

- The ‘trust’ dependency on third party or intermediaries are redundant due to direct P2P handshake within the networks of nodes.
- The following characteristics enable the Trustless nature of the Blockchain.
- **Decentralization**
 - Blockchain consists of decentralized digital data.
 - Data can be shared across a network of computers without a central authority.
- **Transparency**
 - Data is transparent and open to everyone in the P2P network.
- **Privacy**
 - User identity is kept private by robust cryptography.
- **Security**
 - Transactions are cryptographically secured using Hash algorithms.

Benefits of Distributed Ledger Technology

▪ Transparency and security

- Data is visible to all participating nodes. It is difficult to perform any unauthorized changes in the data.
- Every node has a copy of the ledger, thus preventing a single point of failure.
- Any change/entry needs consensus by all nodes making the Distributed ledger secure and almost hack-proof.

▪ Decentralization

- Decentralization gives the users more decision-making power over what goes into their data record.
- Consensus algorithms help to unanimously and securely agree on what should or what should not be added to the distributed ledger.
- It leads an inherent trust within the system.

Benefits of Distributed Ledger Technology

▪ Speed and efficiency

- With the trustless and distributed nature of the Blockchain, the administrative effort of capturing, validating, and synchronization of an individual set of information by central authority can be eliminated.
- It diminishes the chances of human error and improves efficiencies (at the cost of speed).

▪ Cost Savings

- Intermediaries such as banks, brokers, lawyers, and administrative staff are required to foster trust between parties.
- It leads to more costs.

Blockchain system maintains a distributed database in a Decentralized manner using cryptographic signatures and consensus-based validation procedures, making it secure, frictionless, and indisputable.

Types of Blockchain

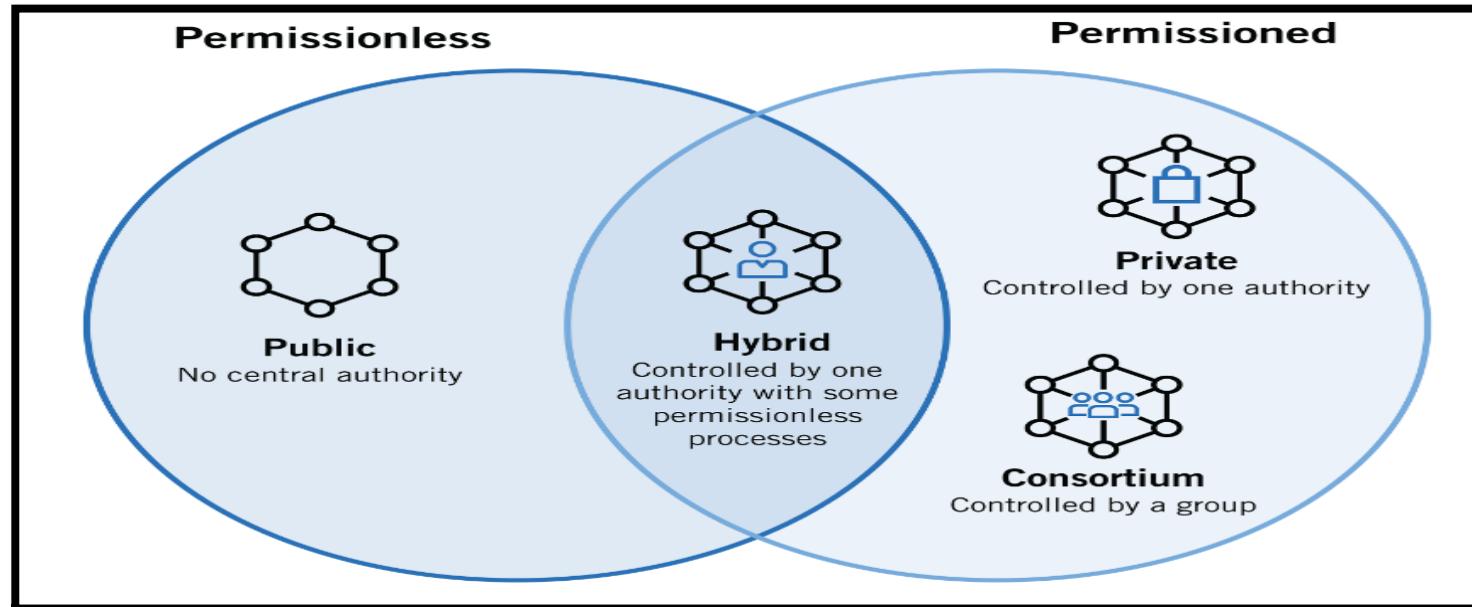
- Blockchains can be categorized based on user identity/authentication and user rights/authorization.

- **Public and Private Blockchain**
 - Depends on accessibility
 - A blockchain is considered either public or private, based on whether the network is open or accessible to anyone with an internet connection.
 - Any node can join the network on the public Blockchain.
 - Only authorized nodes can join the network on a private Blockchain.

- **Permissionless and Permissioned Blockchain**
 - Depends on rights
 - The rights are not restricted in the permissionless Blockchain.
 - The rights and roles are restricted in the permissioned Blockchain.

Types of Blockchain

- The Blockchain can be divided into the following categories based on the authentication and authorization
 - Public Blockchain (Public Permissionless Blockchain)
 - Private Blockchain (Private Permissioned Blockchain)
 - Consortium Blockchain (Public/Private Permissioned Blockchain)
 - Hybrid Blockchain (Interconnected Public-Private Blockchain)



- The scalability of blockchain execution is commonly thought of in terms of transactions per second (TPS).

Public Blockchain (Public Permissionless Blockchain)

- **Anonymity, Transparency, and immutability are valued over efficiency.**
- It is open to the public. Anyone in the world can access the Blockchain.
- No permissions are required.
- Fully decentralized network.
- Anyone can download a copy of the code and run a node.
- Anyone can access/initiate a transaction, and participate in the consensus process to create a block.
- Nodes remain anonymous through high cryptographic protocols.
- No single point of failure (SPOF) as validation (consensus) is done by all the nodes.
- High cryptographic methods are used to secure data.
- It establishes a process of trust (inherent trust).

Public Blockchain (Public Permissionless Blockchain)

- Public Blockchain has poor scalability.
 - The scalability of blockchain execution is commonly thought of in terms of transactions per second (TPS).
- Low transaction speed (10 minutes to create a block in Bitcoin Blockchain).
- Consensus mechanism requires an immense amount of energy and computational power.
- Nodes with powerful computers have a better chance of mining than the others, hence, it may lead to systems controlled by hackers (Centralized systems) (**51% Attack Theoretically possible**).
- Example of Public Blockchain
 - Bitcoin, Litecoin, Avalanche and Ripple
 - Ethereum (?????)

Private Blockchain (Private Permissioned Blockchain)

- It is not open to the public.
 - However, participants are known to each other and hence, trust is assured.
- Only authorized participants can join the Blockchain network.
- Central authority controls the permission to read, write or audit the ledger.
- Central authority controls the consensus mechanism.
- High Cryptographic methods secure the data.
- High transaction speed to create a block (only seconds).
- Low energy consumption as supercomputers are not required.



Private Blockchain (Private Permissioned Blockchain)

- It is not decentralized. It offers better security and scalability.
 - Blockchain is supposed to function in a trustless environment. If nodes are trusted, it may be cheaper to go with a traditional database.
 - Central authority can lead to a single point of failure.
 - Organization decides the central authority.
-
- Example-
 - Multichain
 - Monax
-
- Note:- The private Blockchain is scalable and cryptographically secured from the organization's point of view and hence more cost-effective. It is mostly used by organizations that have strict privacy and compliance requirements.



Consortium Blockchain (Public/Private Permissioned Blockchain)

- Anyone can download and access it.
- Any member node can initiate and receive transactions.
- **A group of authorized individuals or organizations (Consortium) have permission to write or audit the ledger.**
- **A group of authorized individuals or organizations (Consortium) having full access to the ledger can participate in the consensus mechanism.**
- High Cryptographic methods secure the data.
- Block creation is faster compared to a public Blockchain.
- High scalability in comparison to a public Blockchain.
- It has better transaction privacy and traceability.



Consortium Blockchain (Public/Private Permissioned Blockchain)

- Not fully decentralized.
 - However, it offers better scalability and security.
 - Different organizations have different requirements.
 - Consensus is very challenging.
- Examples
 - R3 Corda
 - Hyperledger Fabric



Hybrid Blockchain

- **Public Blockchain**
 - Fully decentralized, tamper-proof, anonymous, transparent, immutable, and open to the public.
 - Low throughput, poor scalability, expensive hardware, and significant energy consumption.
- **Private Blockchain**
 - Higher speeds, lower costs, and better scalability
 - But, centralized and restricted access
- **Hybrid Blockchain incorporates the best practices of both models.**
 - Private Blockchain creates transactions (Block) and maintains the privacy of data.
 - Public Blockchain stores and verifies the blocks, ensuring disintermediation (decentralization).



Hybrid Blockchain

- It has a public network where anyone can participate.
- It also has a private network consisting of participants invited by a central authority.
- The Blockchain is distributed within the network of participants.
- The central authority decides which transaction data can be public and which needs to be confirmed by specific members.
- **Private Blockchain creates transactions (Block) and maintains the privacy of data.**
- **The block is circulated on the public Blockchain for validation and approval, thus, maintaining the trustless nature of the Blockchain**
- Public Blockchain uses Delegated Proof-of-Stake (DPoS) consensus protocol. Private Blockchain may utilize any of the consensus mechanisms.
- Data is immutable and secured by high cryptographic methods.
- It has the highest transaction processing speed.
- Practically hack-proof as no malicious actor can enter either the private network or the robust consensus mechanism of the public network.

Hybrid Blockchain

- Example-
 - XinFin
 - TradeFinex Combining Ethereum and Quorum



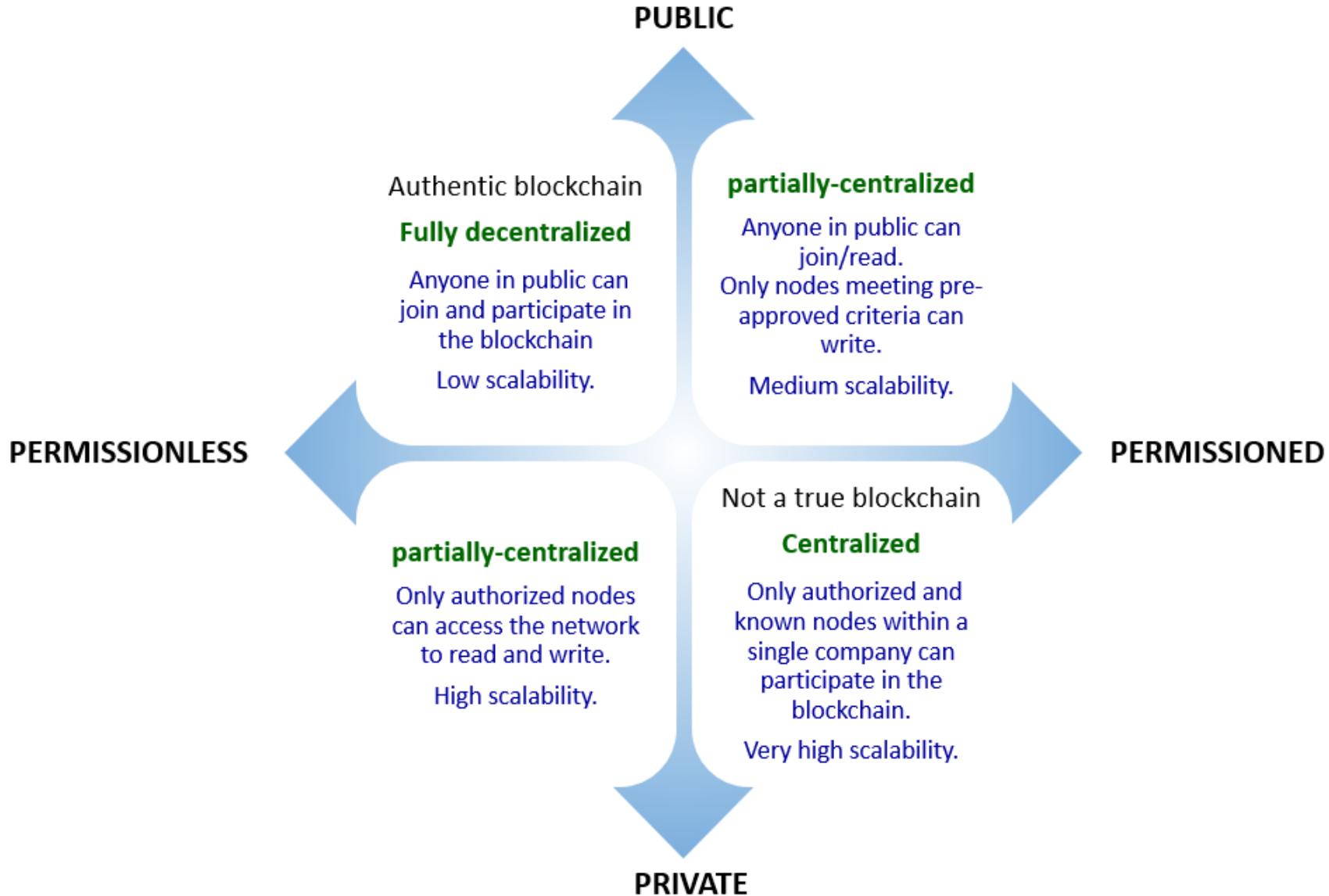
Types and Differences

	PUBLIC BLOCKCHAIN	PRIVATE BLOCKCHAIN	CONSORTIUM BLOCKCHAIN	HYBRID BLOCKCHAIN
ORGANIZATION TYPE	 Public	 Single entity or organization	 Multiple organizations or enterprise	 Highly regulated enterprise
COMMON FEATURES	<ul style="list-style-type: none"> - Chain of blocks - Peer to peer architecture - Public-key cryptography - Immutable - Byzantine fault tolerance - Auditable 			
USERS	Anonymous; but web tracking and cookies pose a risk to privacy	Known & trusted participants	Known & trusted participants	Anonymity for public network members; Private network members, are known within the private network.
ACCESS	open and transparent to all	Access fully restricted	selectively open; relevant transparency provided	Centralized control of providing access, hence privacy and confidentiality maintained
NETWORK TYPE	Decentralized; zero points of failure	Centralized; single point of failure	partially decentralized; multiple points of failure	Zero points of failure
OPERATION	Anyone can read or initiate or receive transactions	Pre-approved participants can read &/or initiate transactions	Pre-approved participants can read &/or initiate transactions	Any combination is possible; Operations are customizable. Central authority decides which transactions can be made public and which are private

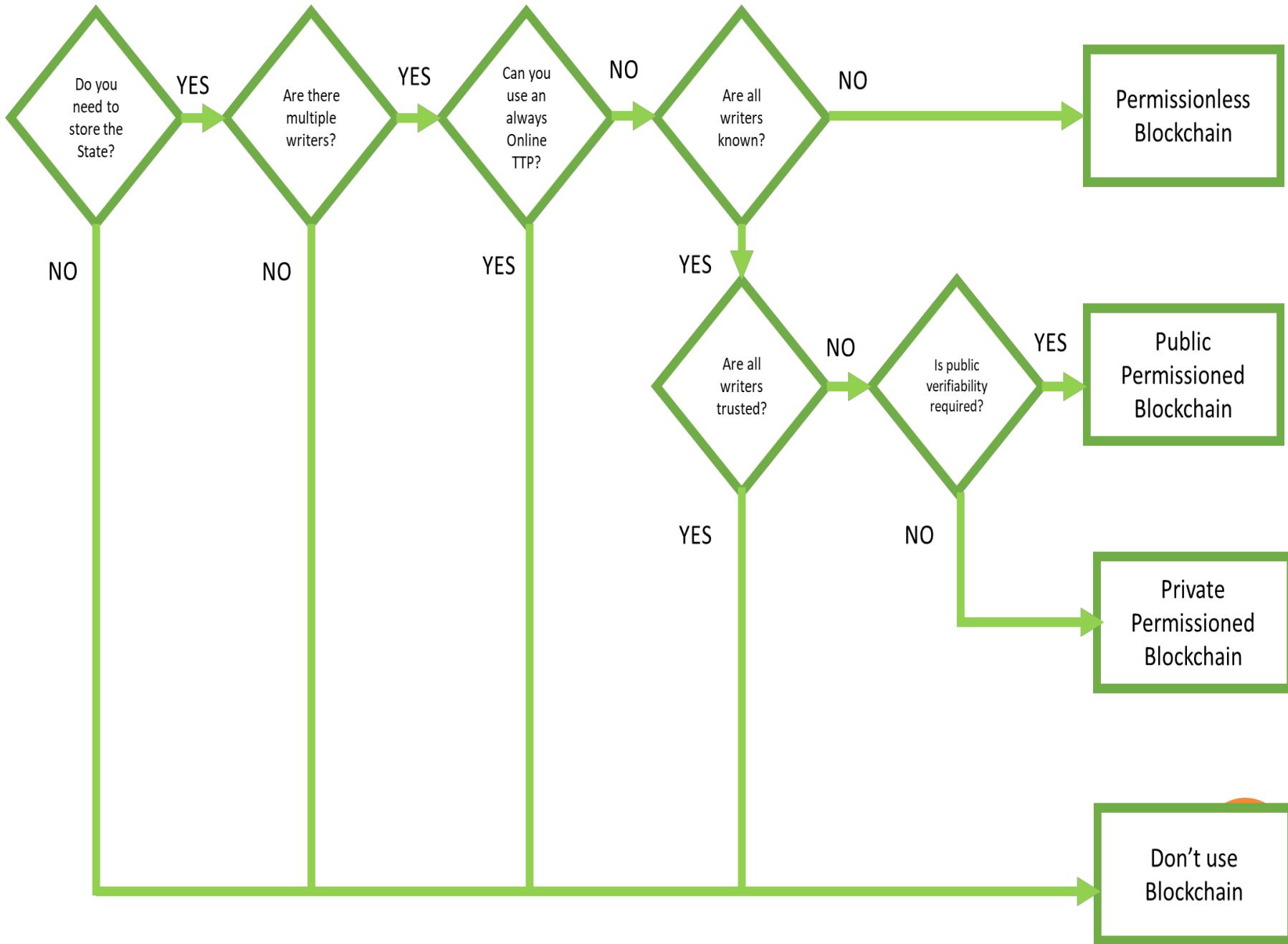
Types and Differences

VERIFICATION	Anyone can be a node and take part in the consensus process to validate transactions and create a block	Single validator node or central authority to create a block	Only privileged members of the consortium can validate and create a block	The public network verifies the block
IMMUTABILITY	Secured by hashing	Secured by distributed consensus	Secured by distributed consensus	Secured by hashing at the private network and secured by distributed consensus by the public blockchain
CONSENSUS MECHANISM	PoW, PoS, etc.	Voting or variations of PoW/PoS consensus algorithms	Voting or variations of PoW/PoS consensus algorithms	DPoS in public and variations in private
INCENTIVIZATION	Incentivizes miners to grow the network	Users limited to within a company; hence incentivization is not relevant	Limited incentivization	Can incentivize users in the main public network
SECURITY	Security based on consensus protocols and hash functions. Higher the security, lower the performance	Security is dependent on the blockchain architecture adopted	Security is dependent on the blockchain architecture adopted	Very high as hackers or unknown parties cannot access the system
TRUST	Trust-free system; trust is enforced via cryptographic proof	Trusted; central control	Trusted; need to trust the majority	Trust-free system; consensus by public blockchain

Types of Blockchain



When to use what type of Blockchain?



Blockchain-as-a-Service

- Similar to Software-as-a-Service
- The upfront investment may discourage many start-ups and SMEs from adopting the blockchain.
- Organizations/Individuals can leverage the BaaS model to build blockchain apps, smart contracts, and other blockchain functions.
- Cloud-based BaaS maintains the infrastructure, storage, bandwidth management, security, and resource allocation.
- Cloud-Based BaaS model enables the company to focus on its core business areas without worrying about infrastructure and performance issues.
- BaaS model must be evaluated for security reasons.
- Example-
 - AWS by Amazon
 - MS Azure by Microsoft
 - IBM Cloud
 - MTBC

Consensus Protocol

- A **consensus protocol** in Blockchain can be defined as a set of rules and procedures for attaining a unified agreement (consensus) between the participating nodes on the status of the network.
- Blockchain Technology utilizes a decentralized network.
- All nodes are equal in the hierarchy and accept decisions collectively for the good of the whole network.
- The consensus protocol aims to overcome the classic problem of a distributed computing system known as the Byzantine Generals Problem.



Byzantine Generals Problem

- Blockchain's key feature of consensus mechanism is seen as a solution to BZP.
- The Consensus mechanism of Blockchain aims to overcome the trust risks attributed to a distributed network system, namely-
 - Authenticity: The message should be easily verifiable to guarantee that it is genuine or not tampered. (SHA-256)
 - Unity: There should be collective agreement by all nodes on the action to be taken. (Majority)
 - Fault-tolerance: A few traitors or hackers should not be able to compromise the process. (Complex Cryptographic Puzzle)

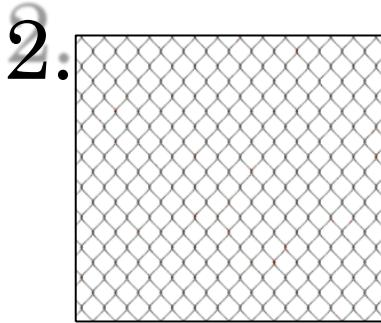


Objectives of Consensus Protocol

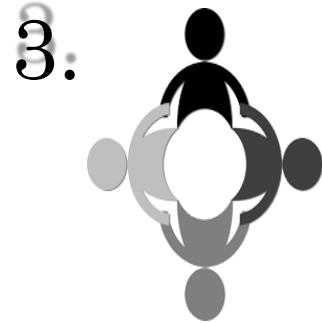
- A consensus mechanism is a fault-tolerant mechanism that is used in Blockchain systems to achieve the necessary agreement amongst members of the network on the transactions that are valid and can be updated on to the ledger.



**Unified
Agreement**



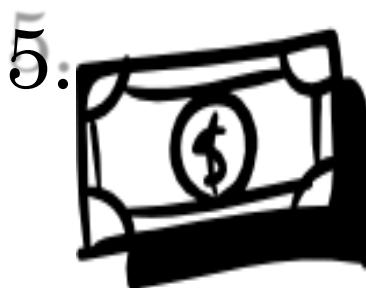
Fault Tolerant



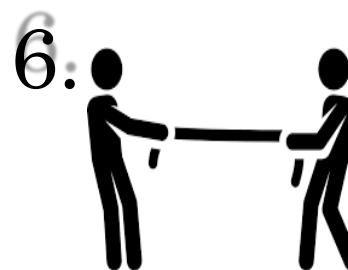
**Collaborative
and
Participatory**



Egalitarian



Incentivisation



Prevent Double-Spend



Objectives of Consensus Protocol

- **Unified Agreement**
 - Agreement should be unified (un-ambiguous) on which data is valid and accurate.
- **Fault-Tolerant**
 - If some of the nodes in the network are unresponsive (faulty), there must be considerable nodes communicating to keep the system functional.
- **Collaborative and participatory**
 - Consensus mechanism should ensure the participation of all nodes in the overall process in the best interests of the group as opposed to the interest of a few nodes.
- **Egalitarian**
 - All nodes are equal and must be treated equally.

Objectives of Consensus Protocol

- **Incentivization**

- Mining is a complex process that requires a vast amount of resources.
- The consensus mechanism must provide some incentives to miners to work for the system to make it more secure.

- **Prevent Double-Spend**

- Double-spending is spending money more than once.
- It is the possibility of digital currency or token being spent more than once by falsification or duplication.
- The double-spend problem is circumvented in blockchain through its consensus mechanism and the basic chronological structure of how the blocks are chained together.
- A transaction is verified and added to a block via the consensus mechanism after a considerable amount of computational power and resources are spent.
- Once a transaction is confirmed, it is nearly impossible to double-spend it

Consensus Algorithms

- | | |
|---|---|
| <ol style="list-style-type: none">1. Proof of Work (PoW)2. Proof of Elapsed Time(PoET)3. Proof of Stake (PoS)4. Delegated Proof of Stake (DPoS)5. Proof of Authority (PoA)6. Practical Byzantine Fault Tolerance7. RAFT | <ol style="list-style-type: none">1. Proof of Stake Anonymous (PoSA)2. Leased Proof of Stake (LPoS):3. Proof of Importance (PoI)4. Proof of Storage5. Proof of Burn6. Proof of Activity7. Proof of Capacity8. Directed Acyclic Graph (DAG) |
|---|---|



Proof of Work (PoW)

- Most well-known mechanism used in Bitcoin Blockchain.
- Proposed by Satoshi Nakamoto.
- Miners compete to solve a complicated mathematical problem based on a cryptographic hash algorithm.
- The solution to the problem is *difficult to produce but easy to verify*.
- The solution (Hash or Golden Nonce) is known as proof of work (PoW).
- PoW makes the process of “adding a new block to Blockchain” very challenging and makes it vulnerable to hackers.
- Mining node releases the PoW in the network for verification to reach consensus.
- The winning miner will get reward and transaction fees (not mandatory) due to the extremely computation-intensive nature of the mining process.
- Example
 - The PoW consensus mechanism is used by Bitcoin, Litecoin, Dash, Monero, and **Ethereum**.

Proof of Work (PoW) : Disadvantages

Time-consuming

- Miners have to iterate over many nonces before finding the Golden Nonce.

High Energy Consumption

- Significant processing power and electricity are consumed in finding the Golden Nonce (Wining Hash).
- As only one miner will be successful, it is wasteful of energy and processing for other miners.

51% Risk

- Miners group the mining pools together to combine their mining resources to counteract the significant energy consumption.
- Mining pools may control the network computing power and validation process. It is known as a 51% attack.
- **It is against the basic principle of distributed and decentralized ledger.**

Proof of Elapsed Time (PoET)

- Developed by Intel Corporation (2016) to enable the permissioned blockchain networks to determine who creates the next block.
- **Follow a lottery system that spreads the chances of winning equally across network participants, giving every node the same chance.**
- A trusted code generates a random wait time for each node in the blockchain network; each node must sleep for that duration.
- The node with the shortest wait time will wake up first and win the block, thus being allowed to commit a new block to the blockchain.

Advantages

- Prevents high resource and energy consumption; it keeps the process more efficient by following a fair lottery system.
- It allows a node to sleep and switch to other tasks for the specified time, thereby increasing network energy efficiency.
- By running a trusted code within a secure environment, the PoET algorithm also enhances transparency by ensuring lottery results (verifiable by external participants).

Proof of Elapsed Time (PoET)

- The PoET consensus mechanism needs to ensure the following crucial factors.
 - Time is indeed random and not a shorter duration which may be chosen by the participants to win.
 - Establishes that the winner has completed the waiting time.
 - Ensures that the trusted code runs within the secure environment and is not alterable by any participant.
 - Ensures that the results are verifiable by participants or other permissioned entities, thereby enhancing the transparency of the network consensus.
- Disadvantages
 - Identity: The identity of the miners is known.
 - Vulnerability: Relies heavily on Trusted Execution Environment. “**Foreshadow**” attack is possible.
- Example
 - Hyperledger Sawtooth architecture uses the PoET consensus mechanism.

Proof of Elapsed Time (PoET)

- Foreshadow and Foreshadow-NG are speculative execution side-channel vulnerabilities.
- Intel CPUs have the Security Guard Extensions (SGX) feature that's designed to protect the privacy and integrity of data and application code from attacks or processes running with high privileges.
- SGX is also supported in cloud infrastructures.
- According to researchers-
 - Foreshadow entails a flaw in SGX's implementation.
 - Successfully exploiting this vulnerability can let attackers access, read, and extract SGX-protected data residing in the CPU's enclaves (that is, SGX-protected memory).



Proof of Stake (PoS)

- Proof-of-stake (POS) was created as an alternative to Proof-of-work (POW).
 - Reduces the amount of computational work needed to verify blocks and transactions.
 - Reduces energy consumption
- Mining nodes are known as “Validators” or “Forgers” or “Delegates”
- Cryptocurrency owners validate block transactions based on the number of coins as validator stakes.



Proof of Stake (PoS)

- The proof-of-stake model allows owners of a cryptocurrency to stake coins and create their own validator nodes.
 - Staking is when you pledge your coins to be used for verifying transactions.
- The owners offer their coins as collateral for the chance to validate blocks.
- Your coins are locked up while you stake them, but you can unstake them if you want to trade them.
- Coin owners with staked coins become "validators."
- Validators are then selected randomly to "mine," or validate the block,, with higher odds being assigned to nodes with larger stake positions.
- This system randomizes who gets to "mine"

- When a block of transactions is ready to be processed, the cryptocurrency's proof-of-stake protocol will choose a validator node to review the block.
- The validator checks if the transactions in the block are accurate.
 - If so, they add the block to the blockchain and receive crypto rewards for their contribution.
- If a validator proposes adding a block with inaccurate information, they lose some of their staked holdings as a penalty.

Proof of Stake (PoS)

- Anyone who owns **Cardano** can stake it and set up their own validator node.
- When Cardano needs to verify blocks of transactions, its Ouroboros protocol selects a validator.
- The validator checks the block, adds it, and receives more Cardano.

Disadvantages

- Cheaper to Attack
 - The perpetrator needs to spend some money on Cryptocurrency.
 - Not required to invest hardware, electricity, time, and other resources.
- Centralization Risk
 - Richest forger can control the consensus mechanism and get even richer.
- Example-
 - Peercoin
 - NXT
 - BlackCoin
 - Cardano

PoW v/s PoS

Proof of Stake	Proof of Work
Block creators are known as validators.	Block creators are known as miners
Participants must buy coins or tokens to become a validator	Participants must buy equipment and energy to become a miner
Energy efficiency	Not energy efficient
Allows for more scalability	Does not allow for more scalability
Network control can be bought	Robust security due to expensive upfront requirement
Validators receive transactions fees as rewards	Miners receive block rewards

The threat of a ~~51% attack~~ still exists on proof-of-stake as it does on proof-of-work, but it's even riskier for the attackers.

Delegated Proof of Stake (DPoS)

- A variation of the PoS consensus mechanism.
- Nodes use their cryptocurrency to vote for the delegates (a.k.a. witness).
- Voting power is directly proportional to the coins.
- Delegates are responsible for validating the transactions and maintaining the blockchain ledger.
 - As it is a democratic system, it is not only the rich, but all users have a chance to be elected as witnesses and earn rewards.
- **Advantages**
 - Voters can easily detect any fraudulent activity by witnesses and can penalize them.
 - All users (nodes) have a chance to be elected as witnesses and earn the rewards.

Delegated Proof of Stake (DPoS)

▪ Disadvantages

▪ 51% Risk

- The nodes (witnesses) responsible for maintaining the blockchain can organize a 51% attack.

▪ Potential Centralized Power

- **The nodes (witnesses) responsible for maintaining the blockchain can organize a 51% attack.**

▪ Example

- Bitshare
- Lisk
- EOS



Proof of Authority (PoA)

- Proposed in 2017.
- A group of pre-selected authorities (validators) secure the Blockchain and produce new blocks.
- The following condition must be fulfilled to identify validators-
 - Validators must have a valid identity in the public domain (physically).
 - Authority must behave uniformly and unbiasedly for all validators.
 - Eligibility criteria for staking identity must be stringent to ensure the trustworthiness of the validator.
- The validators stake their identity rather than coins or tokens.
 - Validators are limited to 25 or fewer to ensure the security and efficiency
 - The identities of the validators are public and verifiable by a reliable third party (like a public notary system).
 - It encourages them to act in the best interest of Blockchain, otherwise, their reputation is ruined.
- Example-
 - VchianThor, Ethereum's Kovan and Rinkeby tesnets
 - Hyperledger and Ripple employs optimized version of PoA.

Proof of Authority (PoA)

- Advantages
 - PoA consumes less power in comparison to PoW.
 - Benefits from zero node-to-node data transfer requirements.
- Issues
 - Semi-centralized
 - Blockchain with PoA leans more towards a centralized system.
 - Works well with private and consortium Blockchain
 - Semi-centralized system offers better scalability
 - Reputational Indifference
 - If the payoff is strong enough, the validators may sacrifice their reputation.
 - The third party may also influence the validators to do some malicious activities and the network may also fail.

Practical Byzantine Fault Tolerance (pBFT)

- Introduced by Miguel Castro and Barbara Liskov at MIT in 1999.
- A potential solution to the Byzantine General's Problem.
- The goal is to decide whether to accept a piece of information submitted to the blockchain or not.
- It is capable to tolerate Byzantine Faults (maximum 1/3 faulty nodes).
- All nodes ("Generals") are considered equal and take their instruction from the **leader** node.
- The leader node is the primary node, and all other nodes are called secondary (backup) nodes.
- The leader is selected at random in a round-robin fashion.



Practical Byzantine Fault Tolerance (pBFT)

- pBFT consensus rounds are broken into 4 phases
 - The client sends a request to the primary(leader) node.
 - The primary(leader) node broadcasts the request to all the secondary(backup) nodes.
 - The nodes(primary and secondary) perform the service requested and then send back a reply to the client.
 - The request is served successfully when the client receives ' $m+1$ ' replies from different nodes in the network with the same result, where m is the maximum number of faulty nodes allowed.
- It works efficiently only when the number of nodes in the distributed network is small due to the high communication overhead
- The communication overheads increase exponentially with every extra node in the network.

Practical Byzantine Fault Tolerance (pBFT)

- Sybil attacks
 - It is susceptible to Sybil attacks, where one entity(party) controls many identities (can assume several identities to manipulate the network).
 - As the number of nodes in the network increase, sybil attacks become increasingly difficult to carry out.
- Scaling
 - pBFT does not scale well because of its communication (with all the other nodes at every step) overhead.
 - As the number of nodes in the network increase(increases as $O(n^k)$, where n is the messages and k is the number of nodes), so does the time taken to respond to the request.



RAFT Algorithm

- RAFT was built as a simpler version of PAXOS.
 - “In Search of an Understandable Consensus Algorithm” by Diego Ongaro, John Ousterhout, Stanford University
- It is similar to pPFT consensus except that-
 - Only the leader node can communicate with other nodes
 - Only the leader node can decide on the transaction
- Nodes have three states: leader, follower, and candidate.
 - The leader candidates’ log must be more up-to-date than the follower logs, otherwise, the candidate will be rejected by the followers.
- RAFT uses the randomized timer to elect the leader for each term.
 - If a leader is not elected in a term, candidates will time out and start the election for the next term.
 - A node starts as a follower expecting a “heartbeat” from a leader.
 - If it does not receive it within the election time, it assumes the leader is dead and takes the candidate state to send out a “RequestVote”.
 - If the candidate node receives the majority of approvals from follower nodes, it transitions to a leader state.

RAFT Algorithm

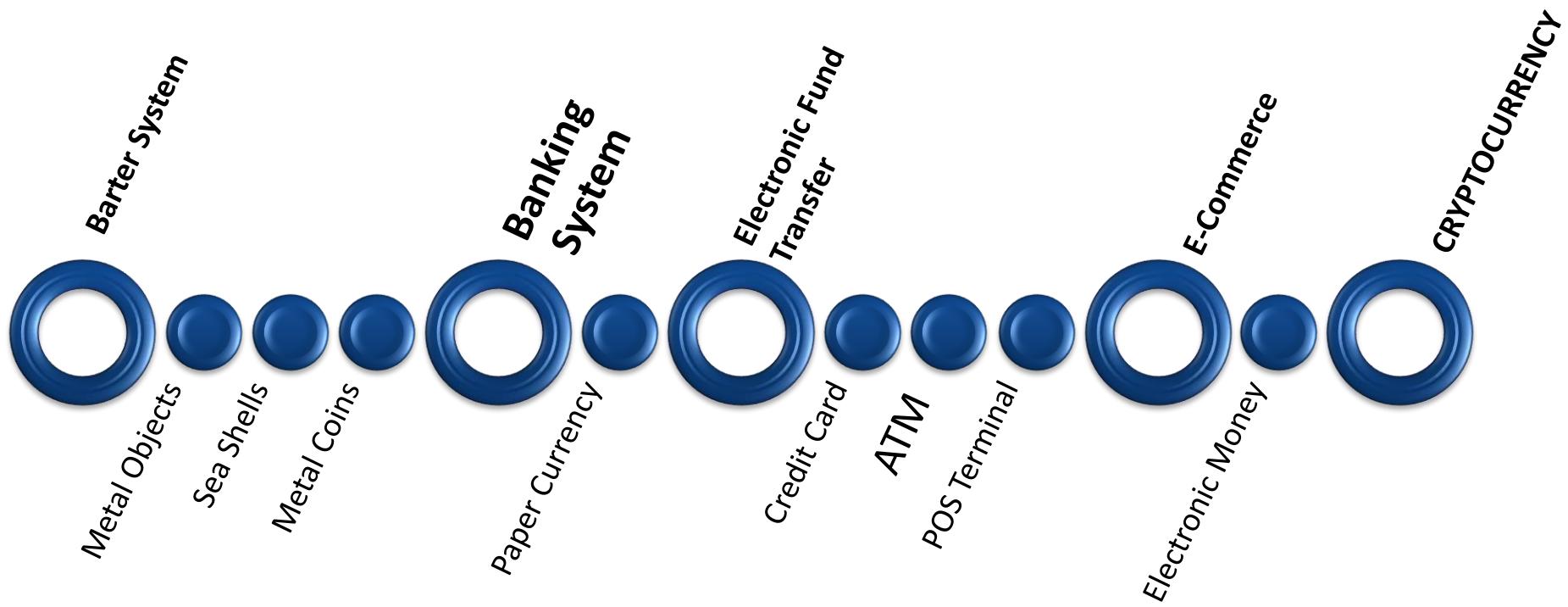
- Only the leader can append log entries based on client requests.
- When the leader node receives a request, it appends the entry to its log as a new entry and sends it to all the follower nodes.
- If the leader node receives the majority of confirmations from follower nodes, the leader commits the message and sends a confirmation message to the client and followers.
- Quorum utilizes RAFT-based consensus for consortium settings.

Cryptocurrency

Bitcoin, Altcoin & Tokens

Evolution of Currency

- Currency and Money
 - Money is a medium of exchange, a unit of account, and a store of value, i.e., Money is defined by the functions it serve
 - Currency is the physical representation of money.



Birth of Bitcoin

- Banks acted as the ultimate gatekeepers of the financial world and charged fees for the services that they provided. **This monopoly, however, had its disadvantages, especially for people in lower-income groups who did not have accounts or IDs or instances where the transaction fees took a toll on their earnings.**
- Financial institutions incur significant costs related to back-office expenses, reconciliations, legalities, secure data storage, prevention measures for security breaches, and potentially fraudulent activities. **These costs are passed down to the end-users as fixed transaction fees irrespective of the size of the transaction.**
- **Transaction fees can also be increased.**
- **There was also the question of transparency.** People deposit money, trusting the banks to keep them safe. However, these deposits are used by banks to find opportunities for additional financial returns like extending mortgages and other loans, and investments.
 - When people defaulted on loan payments and the investments the banks made did not pay off, the banks declared bankruptcy. The result was that while the Government bailed out many of the financial institutions, the depositors lost all the money that they trusted the banks to keep safe, as was seen during the financial crisis of 2008.

Birth of Bitcoin

Three key requirements eventually brought about the birth of Bitcoin. The need was felt for a monetary system

- where one can directly transact with another person without involving a third party, like a bank, to verify and validate the transaction and thus avoiding the cost of mediation.
- that is not backed and controlled by a central authority and can assure the value of the money is maintained
- where there is transparency in transactions, while still maintaining the users' privacy.

What is Cryptocurrency?

- A cryptocurrency is a digital asset that is used as a medium of exchange on the blockchain.
- The most popular cryptocurrency is Bitcoin, followed by Ether of Ethereum and Ripples XRP tokens.
- Since the creation of the first decentralized cryptocurrency Bitcoin, thousands of alternative digital currencies have emerged that are referred to as altcoins or coins and tokens.
 - Altcoins are used for buying or selling products or services
 - Tokens are used as a utility or security.
- According to CoinMarketCap, there are over 3000 active cryptocurrencies as of October 2022 with a market cap 1 Trillion USD.
- <https://coinmarketcap.com/historical/>

What is Cryptocurrency?

- Cryptocurrency first started as a simple peer-to-peer payment focusing more on replacing the traditional cash system.
- It can be many domains-
 - Finance
 - Identity Management
 - Supply chain Management
 - Security
 - Health Management
 - Ownership
 - Records Management
 - Internet of Things
 - Initial Coin Offerings (A form of raising capital for Start-up ventures)

Fork

- **Cryptocurrency and blockchain projects are open-source.**
 - Change the code to build a completely new blockchain(Litecoin)
 - Use existing blockchain to develop applications (Wallets)
- **Creating a cryptocurrency may be easy, but sustaining it is a long-term endeavor.**
- Some aspects of the cryptocurrency that must be decided before the creation
 - Type of usage
 - Total number of coins issues
 - Consensus Mechanism
 - Privacy of transactions
 - Type of Blockchain (private or public)
 - Security and reliability
 - Scalability
 - Etc.
- **The process of duplication from an existing blockchain is known as Forking.**
- A fork creates an alternative version of a blockchain. It is typically done to apply upgrades or new governance rules to a network.
- A fork can be of 2 types:

Fork

- **Hard Fork**
 - A hard fork occurs when the protocol upgrade results in a split in the blockchain that is not backward compatible, i.e., the software validating according to the old protocol will see the new protocol as invalid and vice versa.
 - Hence all clients need to upgrade to the new version if they want to continue participating in the network.
 - The old chain may be completely abandoned or both chains remain active.
 - Bitcoin and Bitcoin-Cash
 - Ethereum and Ethereum Classic
- **Soft Fork**
 - A soft fork is a protocol upgrade that is backward compatible.
 - The old software will recognize the blocks made by the new protocol as valid.
 - Hence it does not require all the nodes in the network to upgrade to maintain consensus as the soft-forked chain follows both the old as well as the new set of protocol rules.
 - However, the majority of miners need to agree on the upgrade so that any block made with the old set of rules is rejected and the old network is eventually abandoned.
 - Bitcoin protocol upgraded to Segregated Witness (SegWit).

Characteristics of Crypto Currency

- **Decentralized**
 - No control of central authority
 - No corruption or inflation
- **Form of Existence**
 - Only digital
- **Limited supply**
 - Maximum supply must be decided at the time of the genesis block creation
- **Global Access**
 - Anyone can access and transact
- **Anonymity & transparency**
 - Transactions must be stored with the person's cryptographic address
 - Blockchain must be available for everyone
- **Impossible to duplicate**
 - Double spending must be impossible
- **Irreversible**
 - Transactions must be irreversible

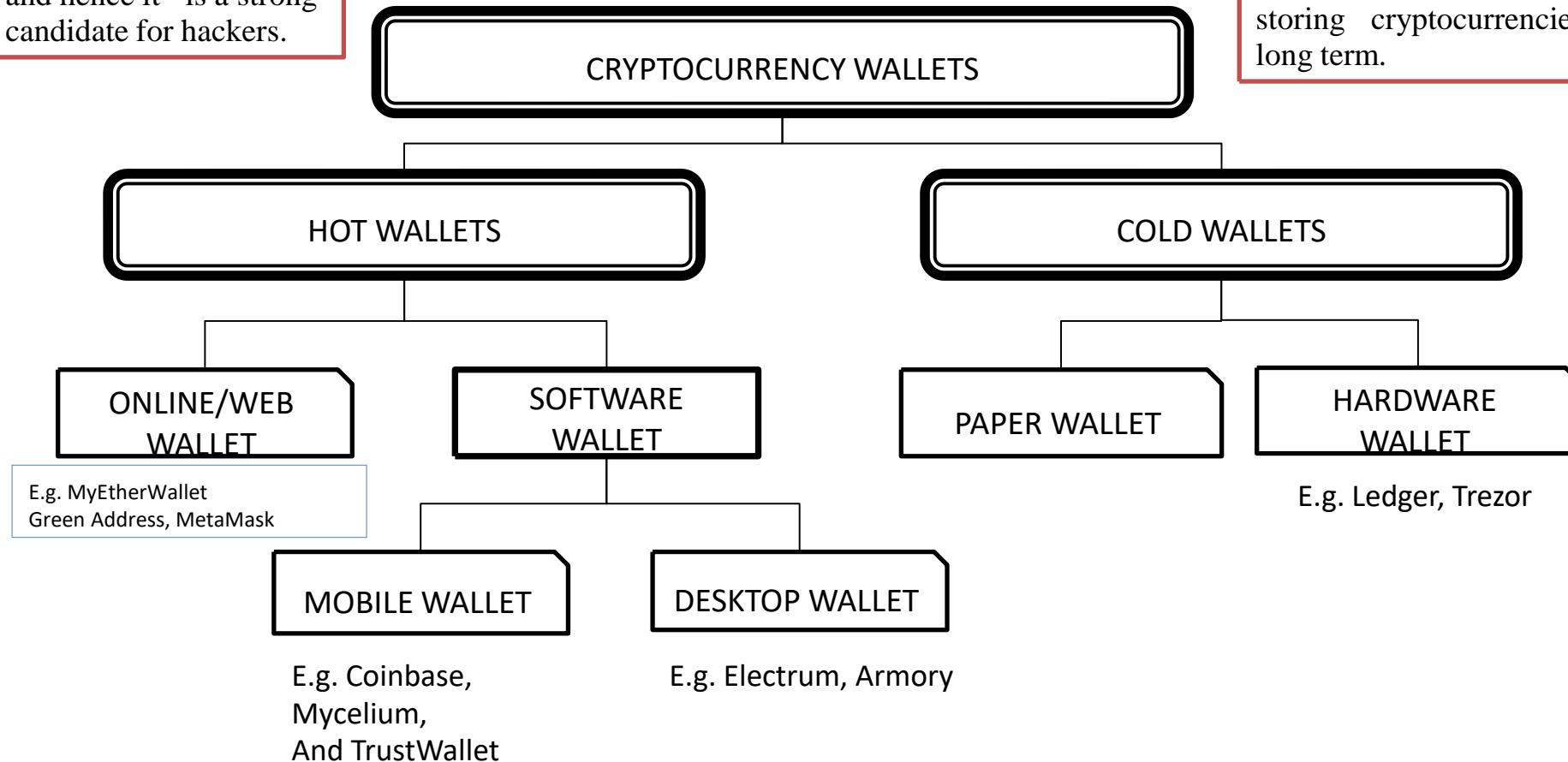
Cryptocurrency Wallets

- Bitcoin or any other cryptocurrency transaction can be done only using a digital wallet.
- A digital wallet or **cryptocurrency wallet** is a software program that stores the user's private and public keys enabling the user to transact crypto assets.
- It is a management system that interacts with various blockchains to enable users to send and receive digital currency and monitor their balance.
- Cryptocurrencies are stored immutably on the blockchain using your public key, i.e., your public key is used by other wallets to send funds to your wallet's address. However, the private key is required if you want to spend cryptocurrency from your address.

Types of Wallets (Storage based)

A hot wallet is designed for online day-to-day transactions. It is always connected to the internet and hence it is a strong candidate for hackers.

A cold wallet is a digital wallet that is not connected to the internet. They are not free. Being offline, they are more secure and used for storing cryptocurrencies long term.



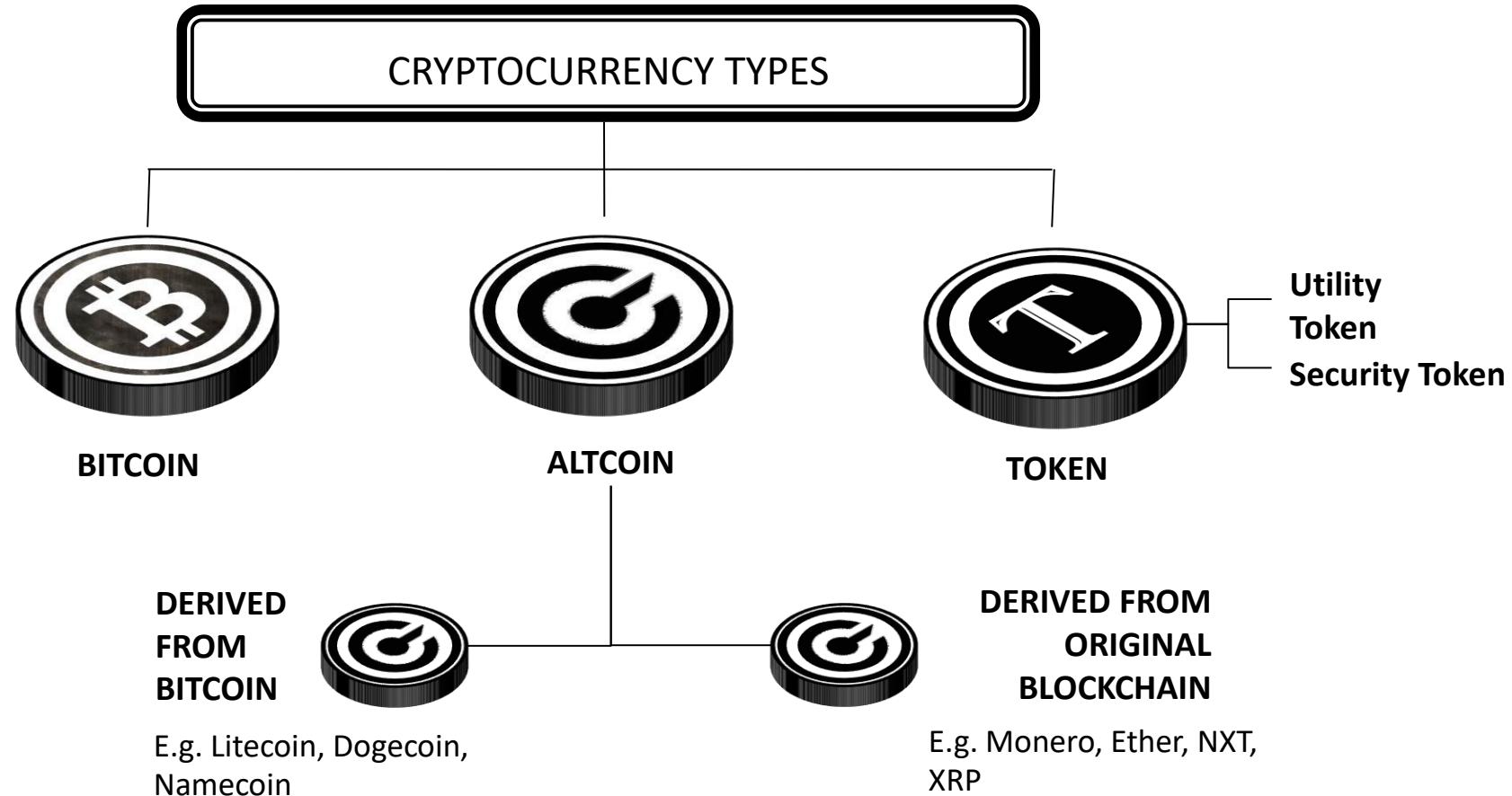
Types of Wallets (Storage based)

- Warm wallets
- Warm wallets combine the transaction speed of hot wallets with an additional level of security. The keys are held online and transactions can be created automatically, but human involvement is needed to sign the transaction and send it to the blockchain.

	Hot	Warm	Cold
Connects to internet	✓	✓	✗
Fully offline	✗	✗	✓
Requires human involvement to sign transactions	✗	✓	✓
Security	✗	✓	✓
Speed and efficiency	✓	✓	✗

<https://www.fireblocks.com/blog/hot-vs-warm-vs-cold-which-crypto-wallet-is-right-for-me/>

Cryptocurrency Types



Alternative Coins

- The term altcoin refers to all cryptocurrencies other than Bitcoin (and for some people, Ethereum).
- There are tens of thousands of altcoins on the market.
- **Altcoins come in several types based on what they were designed for.**
- **The future value of altcoins is impossible to predict, but if the blockchain they were designed for continues to be used and developed, the altcoins will continue to exist.**
- Some altcoins use different consensus mechanisms to validate transactions and open new blocks, or attempt to distinguish themselves from Bitcoin and Ethereum by providing new or additional capabilities or purposes.
- Many are forks—a splitting of a blockchain that is not compatible with the original chain—from Bitcoin and Ethereum. These forks generally have more than one reason for occurring. Most of the time, a group of developers disagree with others and leave to make their own coin.
- **Dogecoin, the popular meme coin, was apparently created as somewhat of a joke. It forked from Litecoin, which itself forked from Bitcoin in 2011. Whatever the intent behind its creation, it was still designed to be a digital payment method.**
- **Altcoins are also referred to as “Currency Tokens” which should not be confused with the broader term of tokens.**

Alternative Coins

- Pros
 - Improve upon another cryptocurrency's weaknesses
 - Higher survivability
 - Thousands to choose from
- Cons
 - Lower popularity and smaller market cap
 - Less liquid than Bitcoin
 - Difficult to determine use cases
 - Many altcoins are scams or lost developer and community interest

Tokens

- Crypto tokens are digital assets that are built on another cryptocurrency's blockchain.
- Every cryptocurrency is built on a blockchain. If a cryptocurrency doesn't have its own blockchain and instead uses another cryptocurrency's blockchain, then it's considered a token.
- Example-
 - Ethereum is a blockchain, and this blockchain's native cryptocurrency is called Ether. Since Ether has its own blockchain, it's considered a crypto coin.
 - Ethereum special was the first programmable blockchain.
 - Because it's programmable, developers can use it to launch their own cryptocurrencies.
 - These cryptocurrencies operate on Ethereum's blockchain instead of their own, which makes them crypto tokens (the official term for tokens built on Ethereum are ERC-20 tokens).
- It's much easier to create a token than a coin.

Tokens: Benefits

- Tokens allow developers to create a cryptocurrency without needing to build a blockchain for that cryptocurrency. That's a big deal because it makes the process of developing cryptocurrencies much faster, simpler, and less expensive.
- Building the blockchain isn't the end of the process either. A new crypto coin needs
 - Validators (miners)
 - Computers in P2P network
 - Security
 - Etc.
- Crypto tokens are assets with value.
- They can typically be created, transferred, traded, bought, and sold, and they're stored in blockchain wallets.
- Transactions with a crypto token are processed on the blockchain that it uses.
 - For example, if it's an ERC-20 token built on Ethereum, then the Ethereum blockchain will handle all transactions for that token.
- Tokens are divided into two categories-
 - Utility Tokens
 - Security Tokens

Tokens: Categories

- **Utility Token**
 - Utility tokens are designed to be used for a specific purpose, for example in a DApp or in a game
 - They offer holders a number of benefits, often access to products and services
 - Utility tokens were often associated with initial coin offerings (ICOs).
 - ICOs serve as a way for developers to raise money for upcoming blockchain projects by offering the project's utility tokens for sale before the project launch.
 - Once the project has gone live, users will already have the tokens they need to begin enjoying its benefits.
 - Utility tokens run on a wide variety of blockchains and ecosystems, but the blockchain in question needs to be smart contract capable.
 - Tokens built on Ethereum are known as ERC-20 tokens.
 - Other examples are
 - Axie Infinity's Small Love Potion, MATIC, and Filecoin

Tokens: Categories

- **Utility Token**



<https://crypto.com/university/utility-tokens-vs-security-tokens#:~:text=Putting%20it%20in%20simple%20terms,stake%20in%20the%20company%20itself.>

Tokens: Categories

- **Security Token**
 - Security tokens (STO) are the digital, web3 version of financial securities
 - They offer ownership in an asset and are regulated by the government just like classic securities
 - tZero and Blockchain Capital are among the two most well-known examples



<https://crypto.com/university/utility-tokens-vs-security-tokens#:~:text=Putting%20it%20in%20simple%20terms,stake%20in%20the%20company%20itself.>

Tokens: Purpose

- Governance Tokens
 - A governance token is a crypto token that gives the holder voting rights in a cryptocurrency project. Token holders are able to make and vote on proposals that help determine the future of that specific cryptocurrency. The more tokens you hold, the more voting power you have.
- Decentralized Finance
 - Decentralized finance (DeFi) refers to alternative financial systems built on blockchain technology.
 - For example, instead of getting a loan from a lender, you could put up crypto tokens as collateral and get one from a DeFi platform.
 - Each DeFi platform has its own token that it uses as its official currency.
- Crypto rewards
 - The previously mentioned DeFi platforms rely on investors who lend their own cryptocurrency funds. In return, investors receive crypto rewards as an incentive. These rewards are usually paid out as crypto tokens.
- Non-fungible tokens
 - A non-fungible token (NFT) is a crypto token that denotes ownership of a digital asset. The ownership information is stored in the cryptocurrency token. NFTs can be used to show who owns a unique digital image, a GIF, or a character in an online game.

Tokens: Example

- Tether (CRYPTO:USDT) and USD Coin (CRYPTO:USDC) are stablecoins pegged to the U.S. dollar. They're designed to maintain a price of \$1, and they're both built on the Ethereum blockchain.
- Shiba Inu (CRYPTO:SHIB) is a controversial meme token that saw its price skyrocket in 2021. That success was primarily due to its popularity, and the token's value has fallen significantly since then. It's also built on the Ethereum blockchain.
- Chainlink (CRYPTO:LINK) is an oracle network that allows smart contracts on a blockchain to receive real-world data. It's built on the Ethereum blockchain as well.
- Uniswap (CRYPTO:UNI) is the token for the decentralized crypto exchange of the same name. The Uniswap exchange offers cryptocurrency trading with no central governing authority, and, like the others on this list, is built on the Ethereum blockchain.

Cryptocurrency v/s Tokens

- Tokens are digital assets that are built to perform a specific function(s) within the project ecosystem.
- Tokens can be used to provide a service or right of product used only for a specific blockchain.
 - A dinner voucher can be used at a Restaurant.
 - The dinner voucher can't be used for watching the cinema.
- Tokens don't have purchasing power.
 - Tokens can be bought with coins but vice-versa is not true.
- Tokens are built to interact with DApps that are executed on the existing Blockchain.

Cryptocurrency Eco-system Players

The strength of any cryptocurrency depends on its community.

- Programmers/Developers: Miners
- Users
- Merchants
- Traders

Crypto Mining

- Miners generate wealth through mining.
- A miner needs to have some level of technical knowledge and expertise in setting up computing software and equipment.
- Blockchains vary in the mining systems that they use. However, they all have some form of a consensus algorithm and an incentive system.
- There are two types of miners:
 - Who go solo are known as solo miners
 - Who collaborates with others referred to as pool miners.
- There are different types of mining based on the processors or equipment used by the miner:
 - CPU Mining
 - GPU Mining
 - ASIC Mining
 - Cloud Mining

Airdrop

- **Airdrop** is a promotional activity aimed at spreading awareness among the blockchain community. It is a distribution event where a blockchain project distributes free coins or tokens to wallet addresses to create a market for the project and create a buzz among investors.
- The various benefits of airdrops to the blockchain enterprise are:
 - Marketing and Hype
 - Rewarding Loyalty Supporters and Investors
 - Wider and Even Distribution of Tokens
 - Lead Database Generation
- Cryptocurrency airdrops are done in two ways
 - A surprise airdrop
 - A planned airdrop

Coin Burning

- Token or Coin Burning is a process of permanently removing coins from circulation to reduce the total supply **to increase the value of the remaining coins.**
- The coin is considered to be burned when it is sent to an unspendable public address, known as an **eater address**, which does not have an operable private key associated with it.
- **Eater Address** does not belong to anyone and has no practical value.

- The key purpose of burning coins:
 - ❖ To Reward Investors
 - ❖ To destroy unsold ICO tokens
 - ❖ To decentralize mining opportunity (Proof-of-Burn (PoB))
 - ❖ In the PoB consensus protocol, miners are granted the right to mine and validate the transactions based on the number of coins they burn.

Cryptocurrency Safety

- Best practice in using Exchanges
 - Choose regulated exchanges that have safety and security measures in place. Binance, Bittrex, and Coinbase are popular crypto exchanges.
- Storing Cryptocurrency
 - Store your crypto in desktop or mobile wallets if short-term and in paper and hardware wallets if long-term.
 - Use wallets from reputable sources.
- Transaction Safety
 - Study the transaction requirements of a cryptocurrency carefully as they may indicate the security precautions to be taken.
- Enable Security Measures
 - Protect wallets and backups with strong passwords.

Blockchain Technology

- Three Layers
 - Blockchain Technology
 - Protocols/Coins
 - Tokens

Blockchain Technology

Technology

Blockchain

Protocol/Coin

Bitcoin

Ethereum

Neo

Ripple

Tokens

X

TRX SNT
REP AE
BNB REP
PPT AHOC

ACA RPX
ONT IAM
QLC TNC
DBC TKY

X

Blockchain Technology

- Blockchain is defined as a **distributed, replicated peer-to-peer network of databases** that allows multiple non-trusting parties to transact **without a trusted intermediary** and maintains an ever-growing, append-only, **tamper-resistant list of time-sequenced records**.
- Blockchain can be considered as a type of **distributed ledger** that sits on the internet for recording transactions and maintaining a permanent and **verifiable record-set of information**.

Protocol

- ❑ A protocol is a set of rules that guides how participants communicate with each other over network
 - ❑ For example, TCP or IP or HTTP.
- ❑ Bitcoin is a protocol which defines a set of rules on how participants of the of the Bitcoin network will communicate with each other and agree on things (mining, consensus, block size, public keys and signatures etc.).
- ❑ There's only one coin attached to every protocol. Every protocol has a coin attached to it.
- ❑ Coin is an asset of the protocol to facilitates the interaction of players, which is used to reward people for mining the Blockchain and adding blocks.
- ❑ It's also used for people to be able to purchase things from each other and so on.

Tokens

- ❑ Tokens rely on smart contracts which are built on top of the different protocols.
 - ❑ Ethereum is the most popular protocol for creating smart contracts and for creating the tokens. There's hundreds of tokens on Ethereum.
 - ❑ Bitcoin has no tokens on it because it does not facilitate that concept of creating smart contracts and therefore nobody creates tokens on Bitcoin.
- ❑ Investing into a coin, means investing into the protocol.
- ❑ Investing a token, means investing into the idea behind what they're building.

Blockchain technology

- In Bitcoin network, a group of people can transact with each other without the need of intermediaries between them.
- The people have never met each other, they can still trust each other because they trust the technology behind it.
- The layer two is about creating a protocol of helping people transact, and inherent component of transacting is exchange of value.
- At the same time, we know that Blockchain can be used for many different things, from health care to media distribution to logistics.
- And that's where the tokens come in.

Bitcoin

- <https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch07.html#:~:text=The%20block%20is%20made%20of,contains%20more%20than%20500%20transactions.>
- <https://medium.com/coinmonks/structure-of-a-bitcoin-block-7f6c4938a5fd>

What is Bitcoin ?

- Nodes



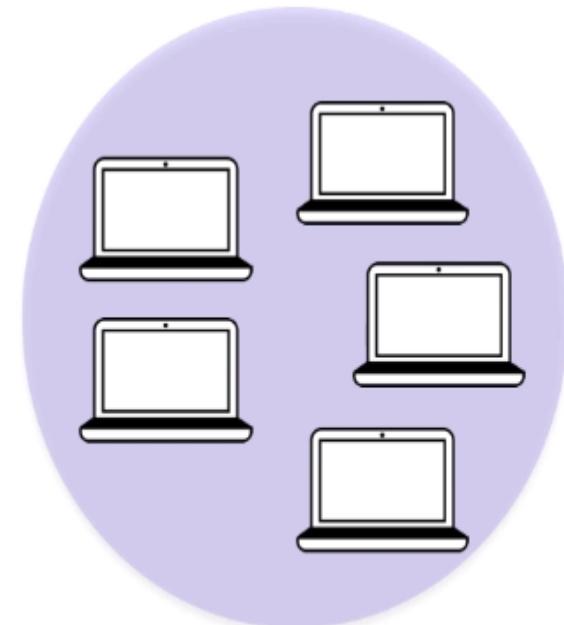
- Large Mines



- Miners



- Mining Pools



What is Bitcoin?

- The Bitcoin protocol includes the following participants.
- And it's not a term for people, it's a term for rather the devices that people use that participate in the network, which are not mining the network.
- So just people who were devices of people who want to transact with each other on this network, there's miners.
- And following the tradition from module one, we're just going to keep representing miners with a laptop.
- And these are participants who actually help the blockchain itself grow by mining it, by adding transactions into blocks and blocks into the blockchain.
- There are large miners.
- These are miners with lots and lots of power, lots and lots of devices and equipment which have a large contribution to the growth of the chain.
- And they are mining pools.
- That's when miners get together to work together on the mining process.

Mining

- Bitcoin mining is the process of discovering new blocks, verifying transactions and adding them to the Bitcoin blockchain.
- Each time a new block is discovered, the successful miner is granted the right to fill that block with new transaction data.
- In return for dedicating time and resources to performing this task, winning miners receive a free amount of newly minted bitcoin known as a “block reward” as well as any fees attached to transactions they store in the new blocks.
- The process of giving successful miners newly minted bitcoin is exclusively how new coins enter circulation.

Bitcoin Monetary Policy

- ❑ Every financial system (central banks of government) has monetary policies being a financial system of its own kind.

- ❑ The Bitcoin Monetary includes two things-
 - ❑ Halving Principle
 - ❑ Block Frequency

Bitcoin Monetary Policy-Halving Principle

Date reached	Block	Reward Era	BTC/block	Year (estimate)	Start BTC	BTC Added	End BTC	BTC Increase	End BTC % of Limit
2009-01-03	0	1	50.00	2009	0	2625000	2625000	infinite	12.500%
2010-04-22	52500	1	50.00	2010	2625000	2625000	5250000	100.00%	25.000%
2011-01-28	105000	1	50.00	2011*	5250000	2625000	7875000	50.00%	37.500%
2011-12-14	157500	1	50.00	2012	7875000	2625000	10500000	33.33%	50.000%
2012-11-28	210000	2	25.00	2013	10500000	1312500	11812500	12.50%	56.250%
2013-10-09	262500	2	25.00	2014	11812500	1312500	13125000	11.11%	62.500%
2014-08-11	315000	2	25.00	2015	13125000	1312500	14437500	10.00%	68.750%
2015-07-29	367500	2	25.00	2016	14437500	1312500	15750000	9.09%	75.000%
2016-07-09	420000	3	12.50	2016	15750000	656250	16406250	4.17%	78.125%
2017-06-23	472500	3	12.50	2018	16406250	656250	17062500	4.00%	81.250%
2018-05-29	525000	3	12.50	2019	17062500	656250	17718750	3.85%	84.375%
2019-05-24	577500	3	12.50	2020	17718750	656250	18375000	3.70%	87.500%
2020-05-11	630000	4	6.25	2021	18375000	328125	18703125	1.79%	89.063%
2021-05-08	682500	4	6.25	2022	18703125	328125	19031250	1.75%	90.625%
2022-05-05	735000	4	6.25	2023	19031250	328125	19359375	1.72%	92.188%
	787500	4	6.25	2024	19359375	328125	19687500	1.69%	93.750%
	840000	5	3.125	2025	19687500	164062.5	19851562.5	0.83%	94.531%
	892500	5	3.125	2026	19851562.5	164062.5	20015625	0.83%	95.313%
	945000	5	3.125	2027	20015625	164062.5	20179687.5	0.82%	96.094%
	997500	5	3.125	2028	20179687.5	164062.5	20343750	0.81%	96.875%
	1050000	6	1.5625	2029	20343750	82031.25	20425781.25	0.40%	97.266%
	1102500	6	1.5625	2030	20425781.25	82031.25	20507812.5	0.40%	97.656%
	1155000	6	1.5625	2031	20507812.5	82031.25	20589843.75	0.40%	98.047%
	1207500	6	1.5625	2032	20589843.75	82031.25	20671875	0.40%	98.438%

https://en.bitcoin.it/wiki/Controlled_supply

Bitcoin Monetary Policy-Halving Principle

- ❑ The halving principle states is that the number of Bitcoins released into the system is halved every single four years.
- ❑ Its actually not just four years, it's 210,000 blocks.
- ❑ Every 210,000 blocks, the number of bitcoins per block is reduced by two.
- ❑ The monetary policy of Bitcoin is entirely controlled by its software i.e., Bitcoin.
- ❑ It is meant to keep things under control, inflation out of control and things.

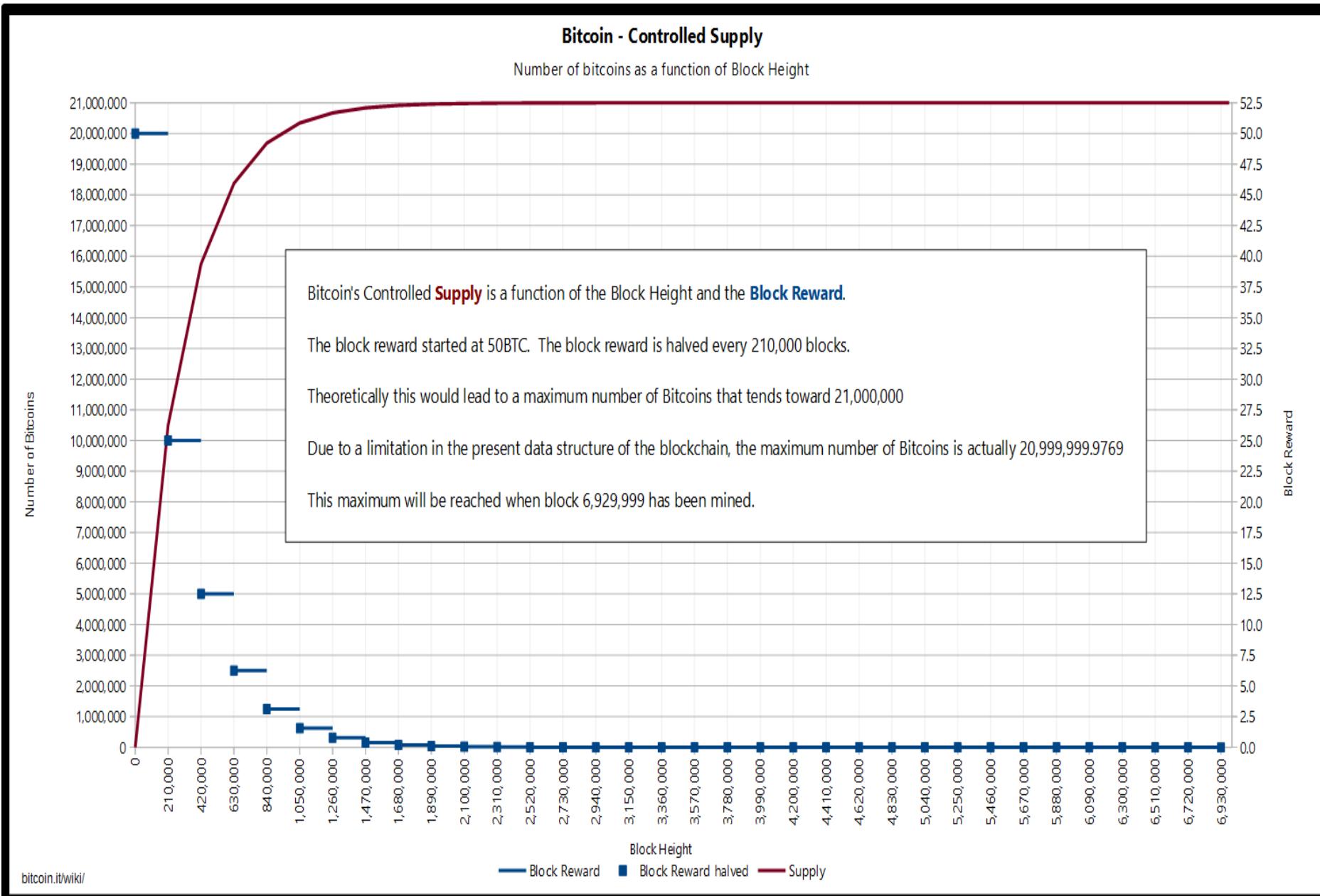
- ❑ Bitcoins are released with every block. Every single time there's a block created, bitcoins are released.

https://en.bitcoin.it/wiki/Controlled_supply

<https://bitsonblocks.net/2015/09/21/a-gentle-introduction-to-bitcoin-mining/>

<https://www.mdpi.com/2078-2489/10/11/335/htm>

Bitcoin Monetary Policy-Halving Principle



Bitcoin Monetary Policy-Halving Principle

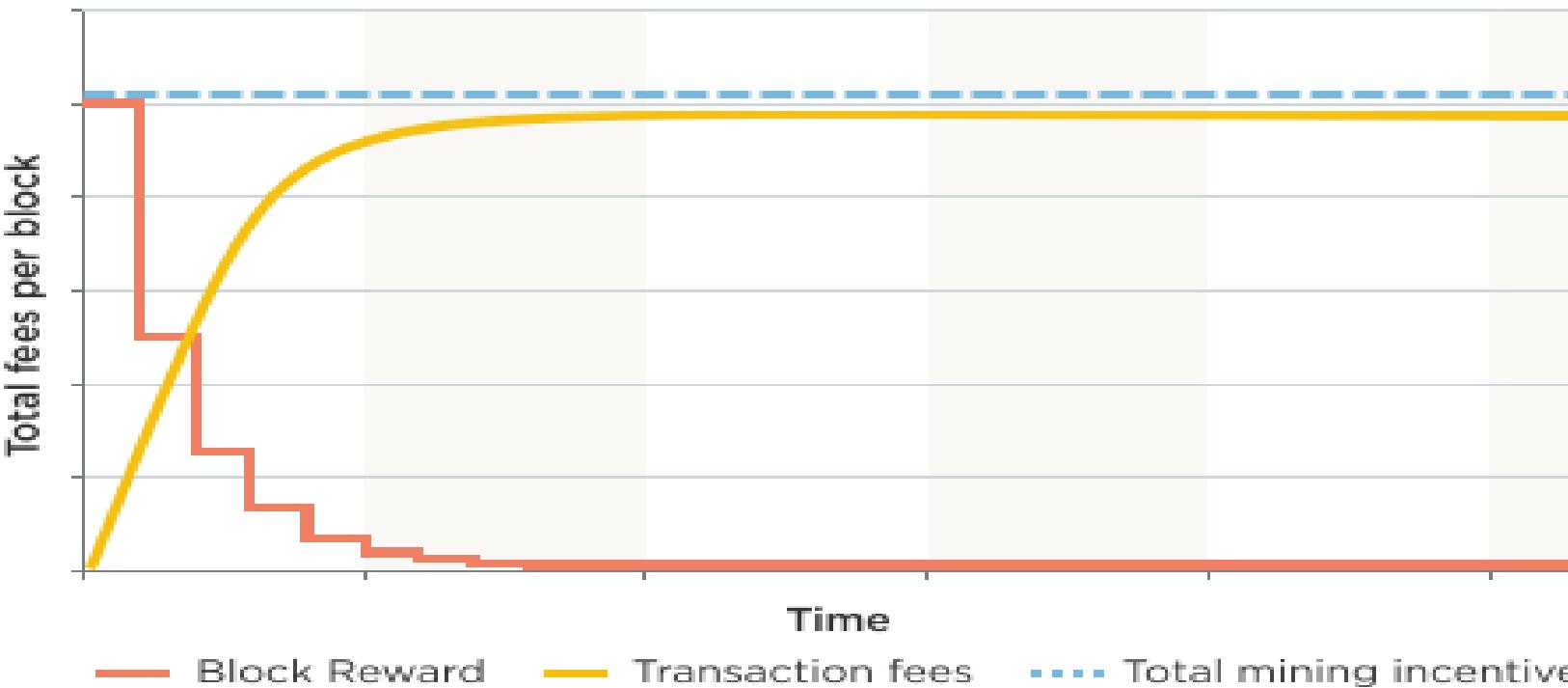
- So every square here is about four years.
- If you think of it that way, you can see that the number of bitcoins is increasing and because of the halving principle.
- So if you take like if you take 50 plus 25 plus 12 and a half plus and so on, and of course,
- you multiply it by the number of blocks, the number of blocks inside this period of four years.
- But in general, you see, because it's halving, it's approximate.
- So it's basically an exponential chart as well and it has an asymptotic value.
- So it goes it will not just keep increasing forever.
- It will converge to a certain amount.
- There is a limitation in the accuracy of how much you can divide a single Bitcoin by because of that limitation, the final bit, the final block or not block, the final bitcoins that will be released as a block reward ever will happen in 2140 at 21 million.
- So theoretically it could keep halving until infinity, but because there is a certain number of decimal points after the comma that are up to the decimal that a bitcoin can handle, the final block will be the final block reward.
- The final Bitcoins released for in the system in this matter will be in 2140, and the total number of bitcoins in circulation ever will be 21 million bitcoins.
- As you can see, it's being halved every four years is going down.

Bitcoin Monetary Policy-Halving Principle

- If the miners are getting less and less and less reward with time, once it gets below one Bitcoin per mined block, and zero after some time, what's the point of mining?
- Well, the hopes are that it won't collapse because the fees should increase because as time passes, more and more people will adopt Bitcoin and more and more people will be using Bitcoin, and therefore more transactions will be required to be processed and therefore miners will be getting more fees and pay and people will be paying more higher fees for their transactions to go through because of the because there will be more demand for transactions to go through. So if you want your transaction to go through, you'll pay more fees and therefore the fees will go up and miners will be compensated.
-
- But in essence, what inflation, as we know is, is like how prices increase in a certain currency. Like if you if a bottle of milk was a dollar, then it becomes \$2. That's because of inflation. Well, in Bitcoin, actually, what you see is over time, there's deflation. This factor is decreasing and that's because the monetary supply is limited and it's the less and less blocks that being released into a system. So Bitcoin is a deflationary type of currency.

Bitcoin Monetary Policy-Halving Principle

**TRANSACTION FEES ARE MEANT TO
REPLACE BLOCK REWARDS**



https://en.bitcoin.it/wiki/Controlled_supply

<https://bitsonblocks.net/2015/09/21/a-gentle-introduction-to-bitcoin-mining/>

<https://www.mdpi.com/2078-2489/10/11/335/htm>

Bitcoin Monetary Policy-Block Frequency

- Block frequency is how often the blocks are mined or added in the Blockchain?
- It depends on the design of the protocol.
- Check mining time of Bitcoin, Ethereum etc.
www.blockchain.com
- How this average time is maintained ?

Cryptocurrency	Average block time
 bitcoin	10 min
 ethereum	15 sec
 ripple	3.5 sec
 litecoin	2.5 min

Bitcoin Monetary Policy-Block Frequency

- I wanted to suggest an interesting article if you'd like to do some additional reading. I found this interesting article by Mark Jevtovic and it's called It's about it's called This Time is different part to what Bitcoin really is. And it's actually about how long is it 15 minutes. Central banks can just do whatever they want. They can release money whenever they like. But in Bitcoin, it's all part of the system, part of a very, very smart system, very well thought through system. So very interesting article. Lots of thought provoking ideas here that will contrast cryptocurrencies to normal fiat currencies, and I highly recommend checking it out for an interesting read.
- This Time is different part 2: what Bitcoin really is by Mark Jevtovic

Mining Difficulty

- Why Mining is Difficult ?
- What is the probability to find a golden nonce ?
- How mining difficulty is adjusted ?

Mining Difficulty

Decimal Number System

XXXXX → 0 to 99999 (1,00,000 possibilities)

0XXXX → 0 to 09999 (10,000 possibilities)

00XXX → 0 to 00999 (1,000 possibilities)

The hash pool size is effectively reduced by 10 times.

Hexadecimal Number System

XXXXX → 0 to FFFFF (10,48,576 possibilities)

0XXXX → 0 to 0FFFF (65,536 possibilities)

00XXX → 0 to 999 (4,096 possibilities)

The hash pool size is effectively reduced by 16 times.

By requesting one leading zero, we're effectively reducing the pool size by 16 times

Every leading zero is reducing the pool size by 16.

Mining Difficulty

SHA-256 HASH Size = 64 Hexadecimal digits

Current Target = 00000000 00000000 009A5DF0 00000000
00000000 00000000 00000000 00000000

Total possible 64-hex digits numbers = 16^{64} =
1157920892373161954235709850086879078532699846656405
64039457584007913129639936 = 1.15×10^{77}

Total valid hashes with 18 leading zeros = 16^{64-18} =
2451992865385422173373355243440494693789982595493763
4816 = 2.45×10^{55}

Probability that a random hash is valid = $1.15 \times 10^{77} / 2.45 \times 10^{55}$
almost = $2.117582368135751 \times 10^{-22}$ = 0. 00000000 00000000
000002

The difficulty can be adjusted

The more number of leading zeros leads to more difficulty

Mining Difficulty Adjustment

- How is mining difficulty calculated and how is it adjusted?
- The difficulty is a measure of how difficult it is to mine a Bitcoin block, or in more technical terms, to find a hash below a given target. A high difficulty means that it will take more computing power to mine the same number of blocks, making the network more secure against attacks. The difficulty adjustment is directly related to the total estimated mining power estimated in the {hashrate} chart.

Mining Difficulty Adjustment

- How is mining difficulty calculated and how is it adjusted?
- It is defined as the ratio of current target and maximum target.
 - $\text{Difficulty} = \text{Current Target} / \text{MaxTarget}$
 - Current Target = 00000000 00000000 009A5DF0 00000000 00000000
00000000 00000000 00000000
 - MaxTarget = 00000000 FFFF0000 00000000 00000000 00000000
00000000 00000000 00000000
- How often is difficulty adjusted?
 - Every block should be released every 10 minutes. Thus, the difficulty is adjusted every 2016 blocks (two weeks).
- Max Target is the target that was in place at the very inception of Bitcoin. They started out with this target. This is not the maximum hexadecimal number.
- Why didn't they start with that?

Mining Difficulty Adjustment

- Why didn't they start with that?
- If they started out with the target at the time being at the very top, then every single
- time a minor would pick an odds, it would be a golden launch.
- So blocks will be mined very quickly with every iteration, and therefore they would mine out all of the bitcoins too quickly even before the first two weeks pass to adjust the default.

- So compared to at the very start of mining Bitcoin, how much longer does it take to mine a bitcoin now?
 - Not longer, because we know it takes 10 minutes. How much more power does it require?
 - Check www.blockchain.com

Mining Difficulty Adjustment

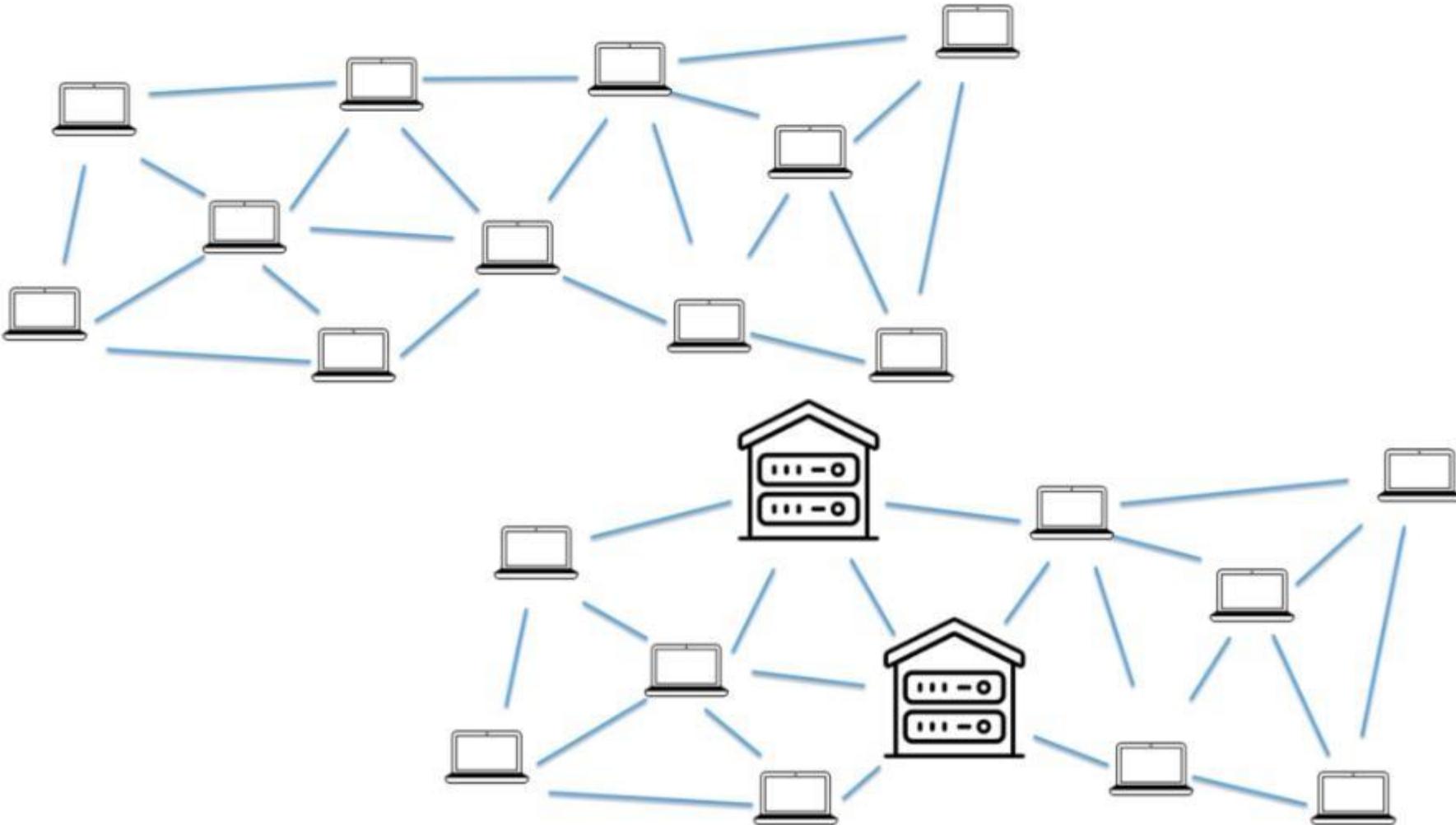
- The important thing to note is that difficulty is not adjusted by any central authority.
- Bitcoin is a decentralized system and difficulty gets adjusted by the nodes themselves.
- If it's more or less than that number, then the difficulty gets adjusted.
- That's how difficult it gets adjusted, every node adjusted.
- And therefore, because the algorithm the same, they all get the same difficulty.
- It's beautiful in its simplicity that all they're doing is they're changing the required number of leading zeros every two weeks. And that is keeping under control this whole swarm of miners and industrial level mining like organizations and mining pools, as we'll see that is all under control by a simple algorithm.

<https://qz.com/1055126/photos-china-has-one-of-worlds-largest-bitcoin-mines/>

Current Target

- The difficulty is defined as the ratio of the current target and the maximum target.
 - Difficulty = Current Target /MaxTarget
 - Current Target = 00000000 00000000 009A5DF0 00000000 00000000 00000000 00000000 00000000
 - MaxTarget = 00000000 FFFF0000 00000000 00000000 00000000 00000000 00000000 00000000
- How to compute the current target?
- The bits field is a little-endian formatted four-byte value interpreted as type int32 that encodes the current target threshold in a compact four-byte field. The resultant block hash must be under the target in order to be considered a valid solution by the network.
- Bits =386,393,970 → Hexadecimal Value is 0x1707E772
- Target = coefficient * $2^{(8 * (\text{index} - 3))}$
- Target = $0x07e772 * 2^{(0x8 * (0x17 - 0x03))} = 0x07e772 * 2^{(0x8 * (0x14))}$
- Target = 0x07e772 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
- Target = 0000 0000 0000 0000 0007 e772 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
- Hash in the Block
- 0000000000000000 0004 ca16a308bd84373b5009199d3d498bd7be9bbe3ce16e

Mining Pools

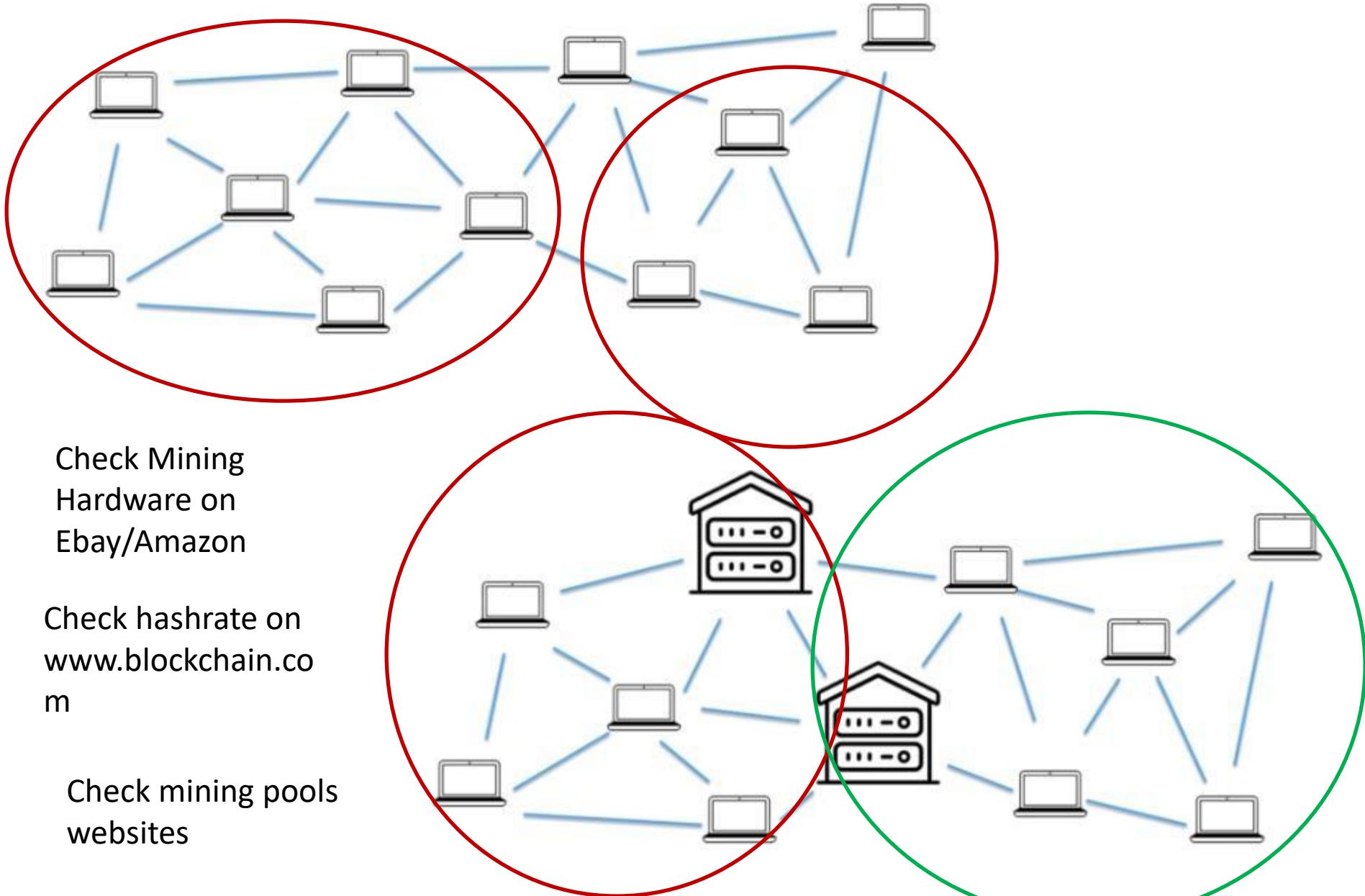


- All miners compete to solve the cryptographic puzzle.
- If the huge industrial size mines join the network, miners with simple laptop can't compete with them. (Chances of solving cryptographic puzzle before them is very rare)

Mining Pools

- Mining pools were invented to provide the fair chance to everyone.
- The miners combine their hashing power, they combine their processing power into this mining pool.
- The mining pool provides a service where the mining pool distributes the work among the miners in such a way that they're not doing double work.
- if they would all be solving like attempting the exact same block configuration of that exact same nonce range, they would be doing double work, then what's the point of them combining their hash rates, their hashing power?
 - The nonce range is divided like from 0 to 1 billion, from 1 billion to 2 billion, from 2 billion to 3 billion, and from 3 billion to 4 billion.
- What about the reward and transactions fee?
 - It will be distributed proportionately to the hashing power that they introduced into the mining pool.
- Mining pools have websites where you can just go and download the software and install it.
 - Anybody can join a mining pool, including these big industrial size rigs.

Mining Pools

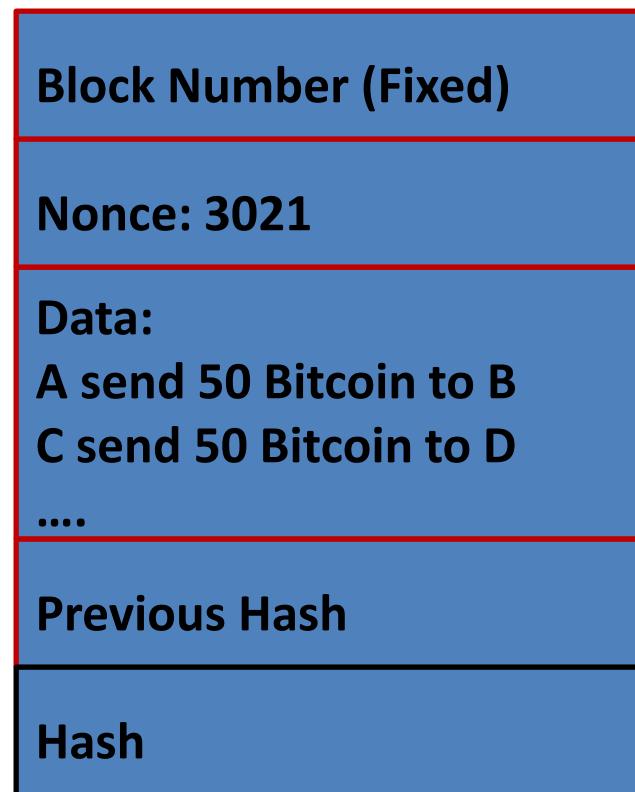


Mining Pools

- A miner just goes to one of mining pools websites. Download a software, and that's when they get started.
- Energy consumption is very high in Bitcoin mining.
- How much energy does mining take?
- The Digiconomist's Bitcoin Energy Consumption Index estimated that one bitcoin transaction takes 1,449 kWh to complete, or the equivalent of approximately 50 days of power for the average US household
 - <https://www.cnet.com/personal-finance/crypto/bitcoin-mining-how-much-electricity-it-takes-and-why-people-are-worried/#:~:text=How%20much%20energy%20does%20mining,for%20the%20average%20US%20household>
 - <https://hbr.org/2021/05/how-much-energy-does-bitcoin-actually-consume>
 - <https://news.climate.columbia.edu/2022/05/04/cryptocurrency-energy/>

Nonce Range

- Nonce is a cryptographic number which is used once in the cryptographic communication.
- It allows miners to participate in the cryptographic puzzle challenge.
- The miners change the nonce and generates the hash value using SHA-256.
- The nonce is a 32-bit unsigned integer. The range is 0 to 4294967295.



Nonce Range

Difficulty

Current Target = 00000000 00000000 009A5DF0 00000000 00000000 00000000 00000000 00000000

Total possible 64-hex digits numbers = $16^{64} =$

11579208923731619542357098500868790785326998466564056403945758400791312963993

$= 1.15 \times 10^{77}$

Total valid hashes with 18 leading zeros = $16^{64-18} =$

24519928653854221733733552434404946937899825954937634816 = 2.45×10^{55}

Probability that a random hash is valid = $1.15 \times 10^{77} / 2.45 \times 10^{55}$ almost = $2.117582368135751 \times 10^{-22}$

= 0. 00000000 00000000 000002 = 2×10^{-22}

Nonce

Maximum value of nonce = 4294967295

Assuming no collisions (one property of SHA-256), total hashes = $2^{32} = 4294967296 = 4 \times 10^9$

Probability that one nonce will be valid = $4 \times 10^9 \times 2 \times 10^{-22} = 8 \times 10^{-13} = 10^{-12}$

= 0.0000 0000 0001

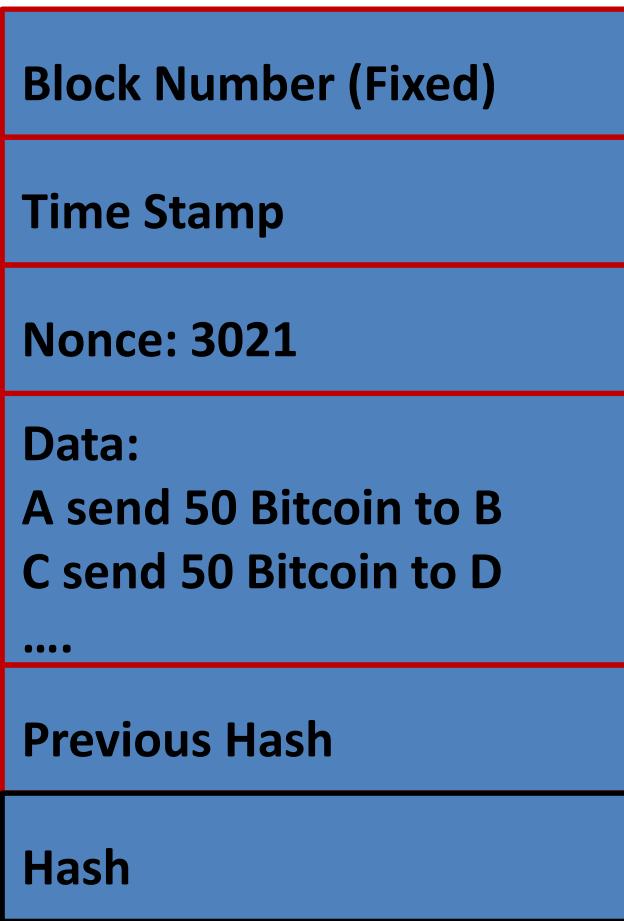
If we generate all hashes, the probability of Golden Nonce is very small.

Is it possible to generate all hashes by a single miner in limited time?

Nonce Range

- Assume a miner generates 100 million (100×10^6) hashes per second.
- The time required to generate all 4 billion hashes = $4 \times 10^9 / 100 \times 10^6 = 40$ Seconds.
- **Time stamp is a field in the block.**
- It is the time stamp of when this block is being mined and the time stamp updates every single second.

- The time is taken from https://time.is/Unix_time
- Unix time is the number of seconds that have passed since 01.01.1970.
- It is universal time.
It effectively provides us infinite number of combinations of timestamp plus nonce range
- The timestamp is updating every second and a miner takes 40 seconds to generate the 4 billion hashes.
- Who is adding blocks in the Blockchain?

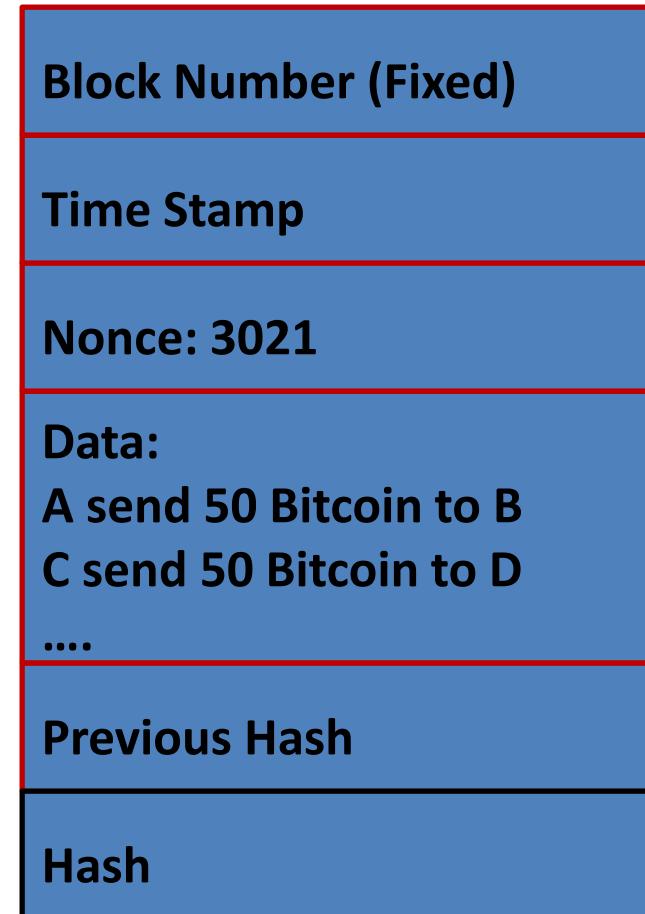


Nonce Range

Mining pool

- Mining pool distributes the nonce range and all miners generate hashes
 - Miners are working together to generate the Golden Nonce .
 - There are competing against each other.
-
- The mining pool smash the nonce range in less than one second.
 - Whenever time updates, they start again.

- What is the current hashrate ?
- www.blockchain.com

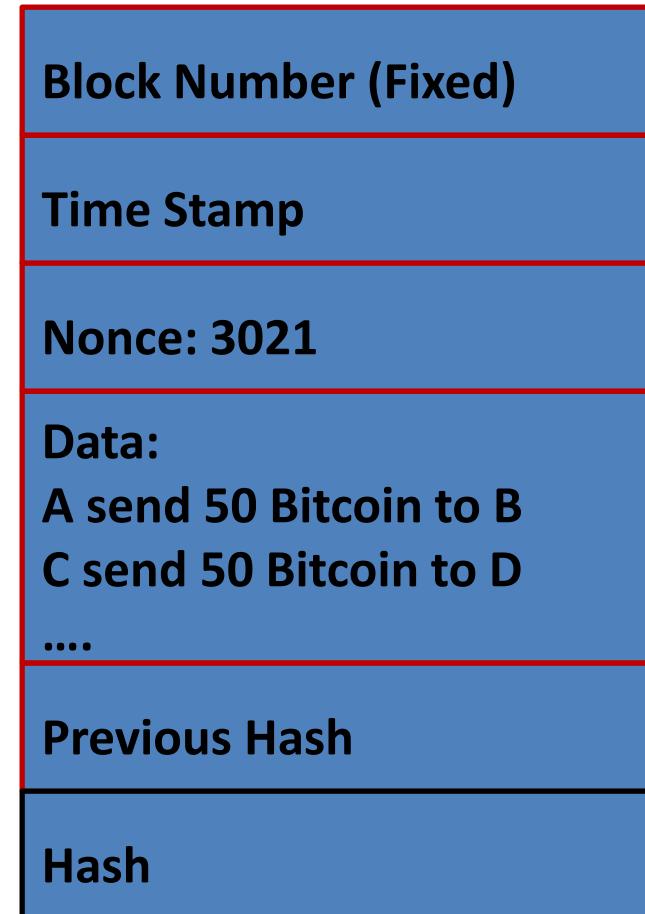


Nonce Range

Mining pool

- Mining pool distributes the nonce range and all miners generate hashes
 - Miners are working together to generate the Golden Nonce .
 - There are competing against each other.
-
- The mining pool smash the nonce range in less than one second.
 - Whenever time updates, they start again.

- What is the current hashrate ?
- www.blockchain.com



Nonce Range

- The mining pool goes through the nonce range from 0 to 4 billion in less than one second.
- They have to wait for time updating.
- This is wastage of so much capacity.
- They just like sitting idly doing nothing.

Transactions Selection

- How transactions are selected and added in the block by the miners?

Miner



Mempool	
1234	T. Fee = 0.00021 BTC
1234	T. Fee = 0.00001 BTC
1234	T. Fee = 0.00004 BTC
1234	T. Fee = 0.00005 BTC
1234	T. Fee = 0.00003 BTC
1234	T. Fee = 0.00015 BTC
1234	T. Fee = 0.00002 BTC
1234	T. Fee = 0.00060 BTC
1234	T. Fee = 0.00070 BTC

Block Number
Time Stamp
Nonce
Data:
Previous Hash
Hash

Transactions Selection

- How transactions are selected and added in the block by the miners?



- Approximately, 2000 transactions are included in a block.
- The blocks are added about every 10 minutes, but transactions happen all the time.
- The transactions are added in mempool before they are included in a block.
- All unconfirmed transactions are stored in mempool (memory pool).
- Mempool is attached to every node, every miner.

Block Number
Time Stamp
Nonce
Data:
Previous Hash
Hash

Transactions Selection

- How transactions are selected and added in the block by the miners?

Miner



Mempool	
1234	T. Fee = 0.00021 BTC
1234	T. Fee = 0.00001 BTC
1234	T. Fee = 0.00004 BTC
1234	T. Fee = 0.00005 BTC
1234	T. Fee = 0.00003 BTC
1234	T. Fee = 0.00015 BTC
1234	T. Fee = 0.00002 BTC
1234	T. Fee = 0.00060 BTC
1234	T. Fee = 0.00070 BTC

Block Number
Time Stamp
Nonce
Data:
1234 T. Fee = 0.00021 BTC
1234 T. Fee = 0.00015 BTC
1234 T. Fee = 0.00060 BTC
1234 T. Fee = 0.00070 BTC
Previous Hash
Hash

- Assume that only four transactions are included in a block (for understanding purpose).
- Initially, the transaction with maximum fee will be selected.
- Transaction fee are not mandatory.

Transactions Selection

- How transactions are selected and added in the block by the miners?

Miner



Mempool	
1234	T. Fee = 0.00021 BTC
1234	T. Fee = 0.00001 BTC
1234	T. Fee = 0.00004 BTC
1234	T. Fee = 0.00005 BTC
1234	T. Fee = 0.00003 BTC
1234	T. Fee = 0.00015 BTC
1234	T. Fee = 0.00002 BTC
1234	T. Fee = 0.00060 BTC
1234	T. Fee = 0.00070 BTC

Block Number
Time Stamp (Variable)
Nonce (Variable)
Data:
1234 T. Fee = 0.00021 BTC
1234 T. Fee = 0.00015 BTC
1234 T. Fee = 0.00060 BTC
1234 T. Fee = 0.00070 BTC
Previous Hash
Hash

- Assume that only four transactions are included in a block (for understanding purpose).
- Initially, the transaction with maximum fee will be selected.
- Transaction fees are not mandatory.

Transactions Selection

- How transactions are selected and added in the block by the miners?

Miner



Mempool	
1234	T. Fee = 0.00021 BTC
1234	T. Fee = 0.00001 BTC
1234	T. Fee = 0.00004 BTC
1234	T. Fee = 0.00005 BTC
1234	T. Fee = 0.00003 BTC
1234	T. Fee = 0.00015 BTC
1234	T. Fee = 0.00002 BTC
1234	T. Fee = 0.00060 BTC
1234	T. Fee = 0.00070 BTC

Block Number
Time Stamp (Variable)
Nonce (Variable)
Data:
1234 T. Fee = 0.00021 BTC
1234 T. Fee = 0.00015 BTC
1234 T. Fee = 0.00060 BTC
1234 T. Fee = 0.00070 BTC
Previous Hash
Hash

- Assume that only four transactions are included in a block (for understanding purpose).
- Initially, the transaction with maximum fee will be selected.
- Transaction fees are not mandatory.

Transactions Selection

- How transactions are selected and added in the block by the miners?

Miner



Mempool	
1234	T. Fee = 0.00021 BTC
1234	T. Fee = 0.00001 BTC
1234	T. Fee = 0.00004 BTC
1234	T. Fee = 0.00005 BTC
1234	T. Fee = 0.00003 BTC
1234	T. Fee = 0.00015 BTC
1234	T. Fee = 0.00002 BTC
1234	T. Fee = 0.00060 BTC
1234	T. Fee = 0.00070 BTC

Block Number

Time Stamp (Variable)

Nonce (Variable)

Data: 1234 T. Fee = 0.00021 BTC

1234 T. Fee = 0.00015 BTC

1234 T. Fee = 0.00060 BTC

1234 T. Fee = 0.00070 BTC

Previous Hash

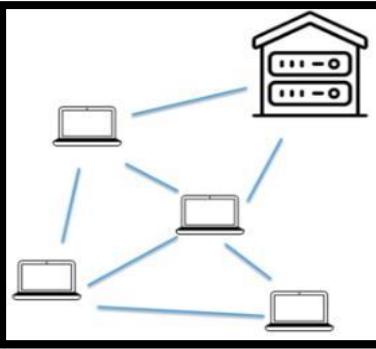
Hash

- ❑ The miner will generate hashes using nonce range from 0 to 4294967295.
- ❑ If the miner generates the 100 MH/s, he may not be able to generate the desired hash (with 18 leading zeros).
- ❑ If he is not able to find the Golden Nonce, the time stamp changes.
- ❑ The miner can repeat the all process again.
- ❑ It represents the reusability of the nonce range.

Transactions Selection

- How transactions are selected and added in the block by the miners?

Mining Pool



Mempool

1234 T. Fee = 0.00021 BTC

1234 T. Fee = 0.00001 BTC

1234 T. Fee = 0.00004 BTC

1234 T. Fee = 0.00005 BTC

1234 T. Fee = 0.00003 BTC

1234 T. Fee = 0.00015 BTC

1234 T. Fee = 0.00002 BTC

1234 T. Fee = 0.00060 BTC

1234 T. Fee = 0.00070 BTC

Block Number

Time Stamp (Variable)

Nonce (Variable)

Data: 1234 T. Fee = 0.00021 BTC

1234 T. Fee = 0.00015 BTC

1234 T. Fee = 0.00060 BTC

1234 T. Fee = 0.00070 BTC

Previous Hash

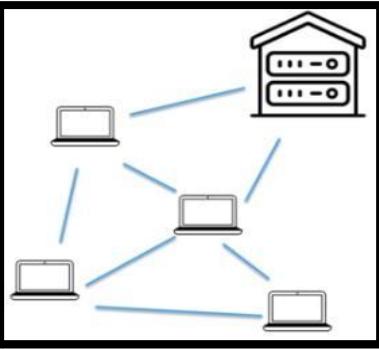
Hash

- The mining pool can generate more than 4 billion hashes in one second.
- The 10% of total hash rate is more than 4 billion. (hashrate on www.blockchain.com)
- Thus, the mining pool can utilize the whole nonce range (0 to 4294967295).
- What do they do after utilizing all nonce range in one second?
- What are they going to do with this idle capacity?
- Either they wait for the next time stamp or they will change the block configuration.

Transactions Selection

- How transactions are selected and added in the block by the miners?

Mining Pool



Mempool	
1234	T. Fee = 0.00021 BTC
1234	T. Fee = 0.00001 BTC
1234	T. Fee = 0.00004 BTC
1234	T. Fee = 0.00005 BTC
1234	T. Fee = 0.00003 BTC
1234	T. Fee = 0.00015 BTC
1234	T. Fee = 0.00002 BTC
1234	T. Fee = 0.00060 BTC
1234	T. Fee = 0.00070 BTC

Block Number

Time Stamp (Variable)

Nonce (Variable)

Data:	1234 T. Fee = 0.00021 BTC
	1234 T. Fee = 0.00015 BTC
	1234 T. Fee = 0.00060 BTC
	1234 T. Fee = 0.00070 BTC

Previous Hash

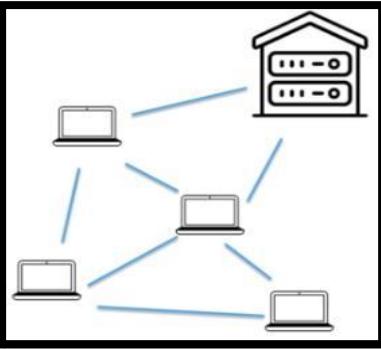
Hash

- ❑ Either they wait for the next time stamp or they will change the block configuration.
- ❑ The miners can change the transactions leading to change in the Block configuration.
- ❑ The miner may remove the transaction with lowest fee in the block and can add the next one.
- ❑ It means the mining pool can again checks all nonce range from 0 to 4 billion.

Transactions Selection

- How transactions are selected and added in the block by the miners?

Mining Pool



Mempool

1234 T. Fee = 0.00021 BTC

1234 T. Fee = 0.00001 BTC

1234 T. Fee = 0.00004 BTC

1234 T. Fee = 0.00005 BTC

1234 T. Fee = 0.00003 BTC

1234 T. Fee = 0.00015 BTC

1234 T. Fee = 0.00002 BTC

1234 T. Fee = 0.00060 BTC

1234 T. Fee = 0.00070 BTC

Block Number

Time Stamp (Variable)

Nonce (Variable)

Data: 1234 T. Fee = 0.00021 BTC

1234 T. Fee = 0.00004 BTC

1234 T. Fee = 0.00060 BTC

1234 T. Fee = 0.00070 BTC

Previous Hash

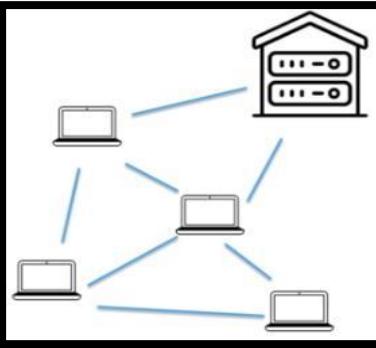
Hash

- ❑ Either they wait for the next time stamp or they will change the block configuration.
- ❑ The miners can change the transactions leading to change in the Block configuration.
- ❑ The miner may remove the transaction with lowest fee in the block and can add the next one.
- ❑ It means the mining pool can again checks all nonce range from 0 to 4 billion.

Transactions Selection

- How transactions are selected and added in the block by the miners?

Mining Pool



Mempool

Mempool	
1234	T. Fee = 0.00021 BTC
1234	T. Fee = 0.00001 BTC
1234	T. Fee = 0.00004 BTC
1234	T. Fee = 0.00005 BTC
1234	T. Fee = 0.00003 BTC
1234	T. Fee = 0.00015 BTC
1234	T. Fee = 0.00002 BTC
1234	T. Fee = 0.00060 BTC
1234	T. Fee = 0.00070 BTC

Block Number

Time Stamp (Variable)

Nonce (Variable)

Data:	1234 T. Fee = 0.00021 BTC
	1234 T. Fee = 0.00015 BTC
	1234 T. Fee = 0.00060 BTC
	1234 T. Fee = 0.00070 BTC

Previous Hash

Hash

If the time stamp updates, this process can start again with transactions having maximum fee.

Transactions Selection

- All of this happens algorithmically, i.e., the transaction selection.
- Algorithm automatically combines the transactions in the best possible way in order to get the maximize the fees, but also in order for the miners not to duplicate their work for different miners in the pool or to duplicate or work.
- If you specify a very low fee in your transaction, like a zero something, it is likely to get stuck in the mempool like no miner will ever use it.
- It might use your transaction might need to switch over, but because it has such a variety, it has like 7000 transactions to choose from.
- They don't even look at transactions below a certain threshold for their allocation.
- And then what happens is your transaction will be released back to you. I think the new standard right now is within 72 hours.
- If it's not picked up by any miner, then it'll be released back to.
- It's like your wallet will usually help you identify what the current fees, average current, average fees are and what you need to input in order for each transaction not to get stuck in the mempool.

Transactions Selection

www.blockchain.com

- Average Transactions per Block
- Average Block Size
- Total Transaction Fee
- Transaction Rate per second
- Etc.

Check the followings-

- <https://btcnitro.com/>
- <https://www.bitcoin.com/get-started/how-bitcoin-transactions-work/>
- <https://www.coindesk.com/learn/how-bitcoin-mining-works-2/>

CPU, GPU, FPGA, and ASIC

Central Processing Unit (CPU)

- The Central Processing Unit(CPU) is the main chip in your computer, phone, tv, etc., that is responsible for distributing instructions throughout the components on the motherboard.
- The CPU is considered to be the "brain" of the computer and is the most versatile of the chips we are covering. However, this versatility does come at a cost, and it will consume more power and be slower at some functions over the more specialized chips.
- Graphics Processing Unit (GPU)
 - The General-Purpose Graphics Processing Unit (GPU) is a type of graphics processing unit that uses a graphics processing unit that processes graphics tasks to compute general-purpose computing tasks that are otherwise processed by the central processing unit.
 - Similar to CPUs, GPUs are general-purpose processors. However, GPUs have more cores and therefore have stronger floating-point computing capabilities and high concurrent processing capabilities.
<https://forum.huawei.com/enterprise/en/differences-between-cpu-gpu-fpga-and-asic/thread/966389-895>

CPU, GPU, FPGA, and ASIC

- **Field Programmable Gate Array (FPGA)**
 - The Field Programmable Gate Array (FPGA) is also a silicon-based semiconductor, but it is based on a matrix of configurable logic blocks (CLBs), which are connected through programmable interconnects.
 - An FPGA can be programmed to implement certain specific applications or functions. Supports hardware language programming and C/C++ programming, featuring high energy efficiency ratio and low latency. However, whenever a function needs to be changed, it must be reprogrammed.
- **Application Specific Integrated Circuit (ASIC)**
 - Application Specific Integrated Circuit (ASIC) is a kind of silicon chip designed for a specific logic function. It is a special chip that realizes a specific function from hardware design. The computing capability and efficiency of the ASIC chip can be customized based on algorithm requirements, but cannot be changed.
 - Because of this, ASIC chips have the characteristics of small size, low power consumption, and high specific computing performance. Once customized, the application scope cannot be changed, and the cost is high.

<https://forum.huawei.com/enterprise/en/differences-between-cpu-gpu-fpga-and-asic/thread/966389-895>

CPU, GPU, FPGA, and ASIC: Computing Performance

- CPU: The CPU uses the von Neumann architecture and its core stores programs and data. The serial computing and parallel computing capabilities depend on the number of CPU cores. Therefore, the computing performance of the CPU is relatively low.
- GPU: The GPU also uses the von Neumann architecture. However, more than 80% of the GPU chip space is occupied by ALU computing units, while less than 20% of the CPU space is occupied by ALU computing units. Therefore, the GPU computing performance is improved, especially in the floating-point computing capability.
- FPGA: The FPGA uses the no-instruction and no-memory sharing mechanism. It implements software algorithms through hardware programming to improve computing efficiency. Therefore, the computing efficiency is an order of magnitude higher than that of both the CPU and GPU. Because of the no-memory mechanism, the computing workload is relatively small, and the CPU needs to be coordinated to process some complex computations.
- ASIC: Currently, ASIC chips are implemented by combining CPLD and FPGA. Therefore, they have the same computing performance as FPGA. It can be understood that ASIC is an FPGA chip whose algorithm logic has been compiled based on a specific scenario.
- Overall, in terms of computing performance, $\text{CPU} < \text{GPU} < \text{FPGA} \approx \text{ASIC}$.

CPU, GPU, FPGA, and ASIC: Power Consumption

- CPU: When processing a large number of complex logic operations, the CPU needs to invoke different instruction sets and store data in the cache. Therefore, the CPU is destined to be large in size and high in power consumption.
- GPU: Similar to the CPU, the GPU still needs to access the cache during processing. In addition, because the GPU has more cores, a higher parallel computing capability, and a higher speed, the GPU has a larger size and higher power consumption than the CPU. Generally, the GPU needs to install an independent fan to ensure heat dissipation.
- FPGA: Because the FPGA has no instructions and does not need a shared memory system, the FPGA chip is small in size and power consumption.
- ASIC: Similar to the FPGA, the ASIC has only a specific logic algorithm and does not require a shared memory architecture. Therefore, the ASIC consumes less power.
- Overall, in terms of power consumption, $\text{ASIC} < \text{FPGA} < \text{CPU} < \text{GPU}$.

CPU, GPU, FPGA, and ASIC: Flexibility

- CPU: The CPU itself is positioned to perform general computing, so the CPU has good flexibility. CPU is also the mainstream processor.
- GPU: Initially, GPUs are mainly used for graphics computing. However, with each generation of GPUs, GPUs are not limited to graphics processing, but can also process complex logic computing. The current CPU ecosystem is mature and is a mainstream heterogeneous computing component with good flexibility.
- FPGA: FPGA is a kind of semi-customized chip, which needs to be developed again for normal use. Therefore, FPGA has bad flexibility.
- ASIC: As mentioned earlier, ASIC can be understood as an FPGA chip on which logic algorithms have been developed. Therefore, ASIC is a customized chip and can only be applied to specific scenarios. Therefore, ASIC has poor flexibility.
- Overall, in terms of flexibility, $\text{ASIC} < \text{FPGA} < \text{GPU} \approx \text{CPU}$.

CPU, GPU, FPGA, and ASIC: Latency

- CPU: The CPU needs to invoke a large number of instruction sets and cache data during calculation. Therefore, the latency is high.
- GPU: Similar to the CPU, the GPU needs to invoke a large number of instruction sets and cache data during computing. However, because the GPU has a better cache access speed, the latency of the GPU is lower than that of the CPU.
- FPGA: The FPGA does not need to invoke the instruction set or cache data when performing operations. Therefore, the latency of the FPGA increases by an order of magnitude compared with that of the CPU.
- ASIC: ASICs are similar to FPGAs, do not need to invoke the instruction set or cache data when performing operations. Therefore, the latency of the FPGA increases by an order of magnitude compared with that of the CPU.
- Overall, in terms of latency, $\text{ASIC} \approx \text{FPGA} < \text{GPU} < \text{CPU}$.

CPU, GPU, FPGA, and ASIC: Application Scenarios

- CPU: The CPU processes complex logical operations and is mainly applied to traditional data center servers.
- GPU: GPUs have excellent graphics processing capabilities and are mainly used in image classification, Safe City, and autonomous driving.
- FPGA: FPGAs can implement computing in specific scenarios based on algorithm logic. Therefore, FPGAs are mainly used in scenarios such as deep learning and big data analysis.
- ASIC: ASIC is mainly used for data inference, assisted driving, and AI synthesis.

CPU, GPU, FPGA, and ASIC

Type	CPU	GPU	FPGA	ASIC
Flexibility	Generally	Generally	Semi-customized	Customization
Computing Performance	Low	Middle	High	High
Power Consumption	Middle	High	Low	Low
Latency	High	Middle	Low	Low
Cost	High	High	Middle	Low
Application Scenarios	Wide application scope •Traditional data centers •PCs	Wide application scope •Image classification •Safe City •Autopilot	Middle application scope •Deep Learning •Big Data Analytics	Low application scope •Data inference •Assisted driving •AI synthesis

<https://forum.huawei.com/enterprise/en/differences-between-cpu-gpu-fpga-and-asic/thread/966389-895>

CPU, GPU, ASIC & Cloud Mining

Central Processing Unit (CPU)	General	< 10 MH/s
Graphics Processing Unit (GPU)	Specialized	< 1 GH/s
Application Specific Integrated Circuit (ASIC)	Totally Specialized	< 1,000 GH/s
Cloud Computing Mining	Depends	Depends

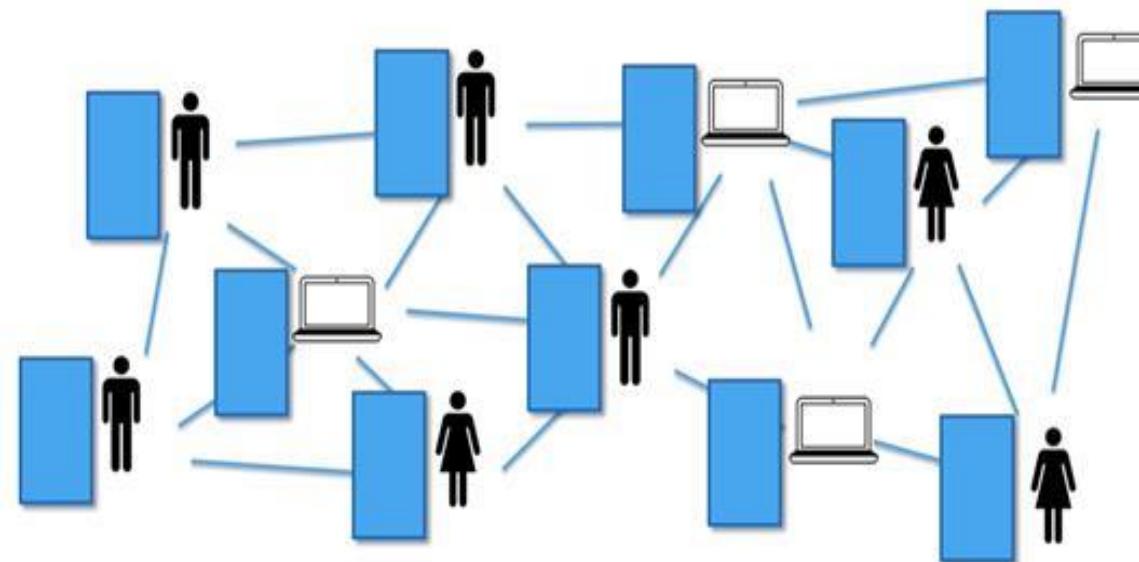
<https://www.vijaypradeep.com/blog/2017-04-28-ethereums-memory-hardness-explained>

CPU, GPU, ASIC & Cloud Mining: Summary

- CPUs are generalized to perform a lots of different things. It can do very sophisticated things. It can definitely solve the SHA 256 hash. It can calculate about ten mega hashes per second.
- A GPU is a video card (Graphics card) in our laptops or computers. It is designed to work with graphics and the operations that is specialized for our matrix operations. The matrix operations are those are required in order for for graphics to appear on. GPU is specifically tailored for to solve matrix operations, matrix multiplications and other matrix operations. It can also compute hashes i.e., It is more efficient and it can compute 1 billion hashes per second.
- An ASIC is totally specialized for one thing and one thing only, and that is to calculate the 256 hash. It is integrated circuit designed for specific purpose. The electricity runs through the device, the inputs are applied, the calculations are performed all that kind of physical level rather than on a logical level. It computes 1 trillion hashes per second.
- In cloud mining, the miners rent equipment off premises somewhere far away and they pay a fee in order for that equipment to participate in mining on their behalf.

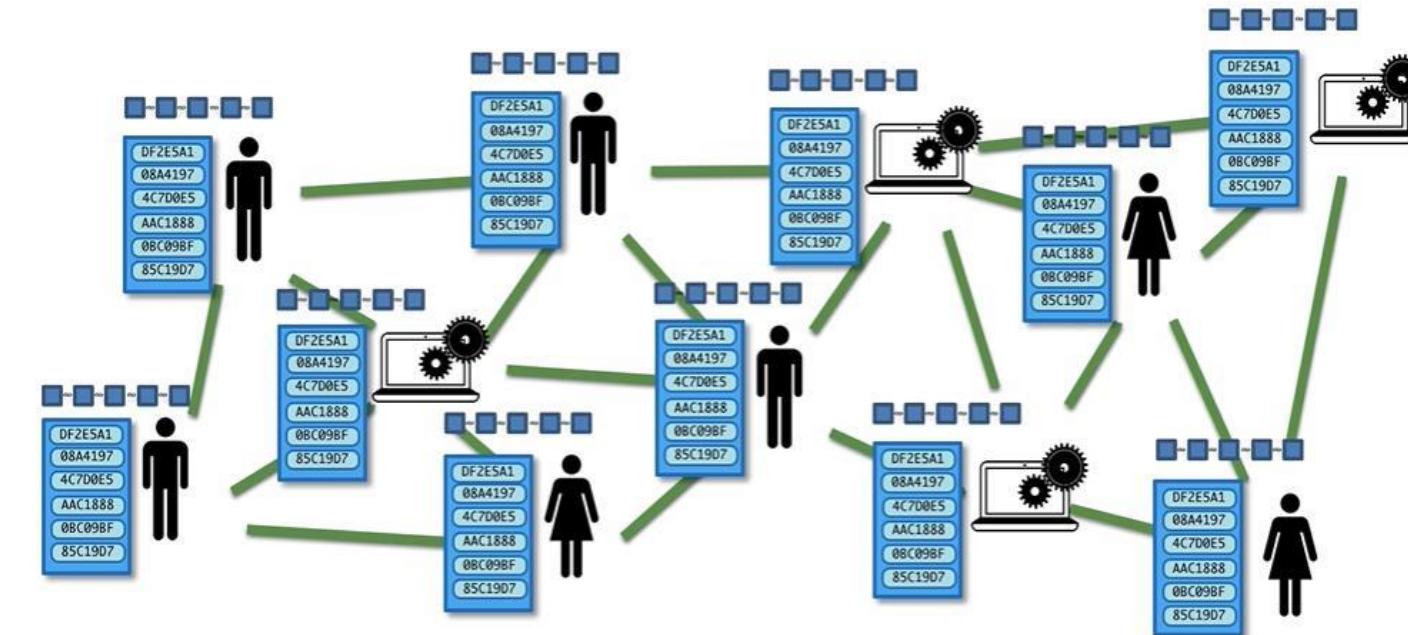
Mempools Functioning

- How mempools work in the distributed peer-to-peer system?
- The Blockchain network consists of nodes (miners and participants).
- A mempool is attached to each one of the nodes.
- The mempool is a staging area for transactions.
- The blocks are added to the blockchain at a certain regularity.
- The transactions occur very frequently.



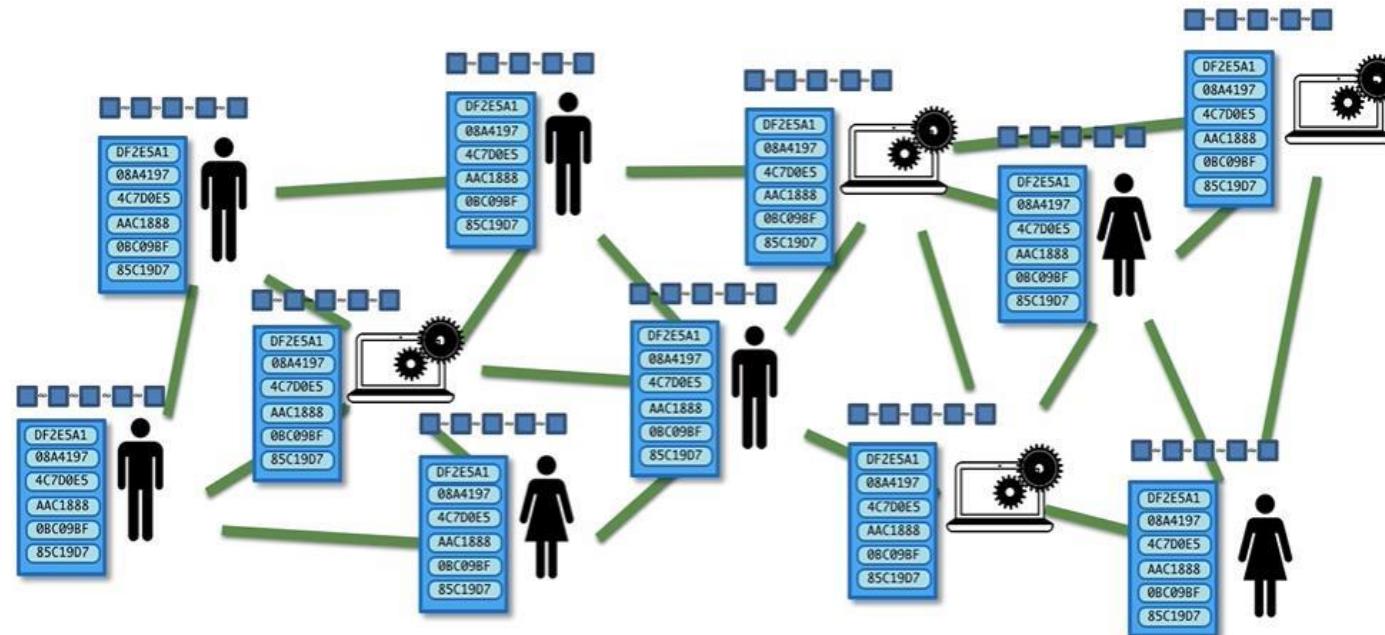
Mempools Functioning

- How mempools work in the distributed peer-to-peer system?
- The Blockchain network consists of nodes (miners and participants).
- A mempool is attached to each one of the nodes.
- The mempool is a staging area for transactions.
- The blocks are added to the blockchain at a certain regularity.
- The transactions occur very frequently.

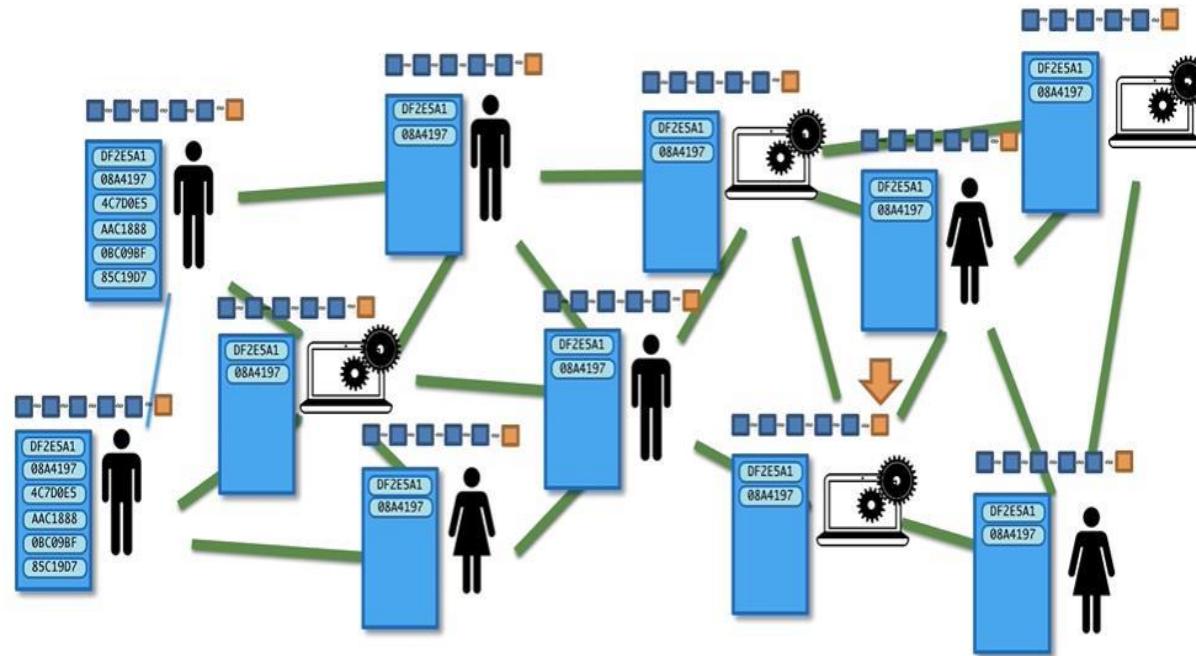
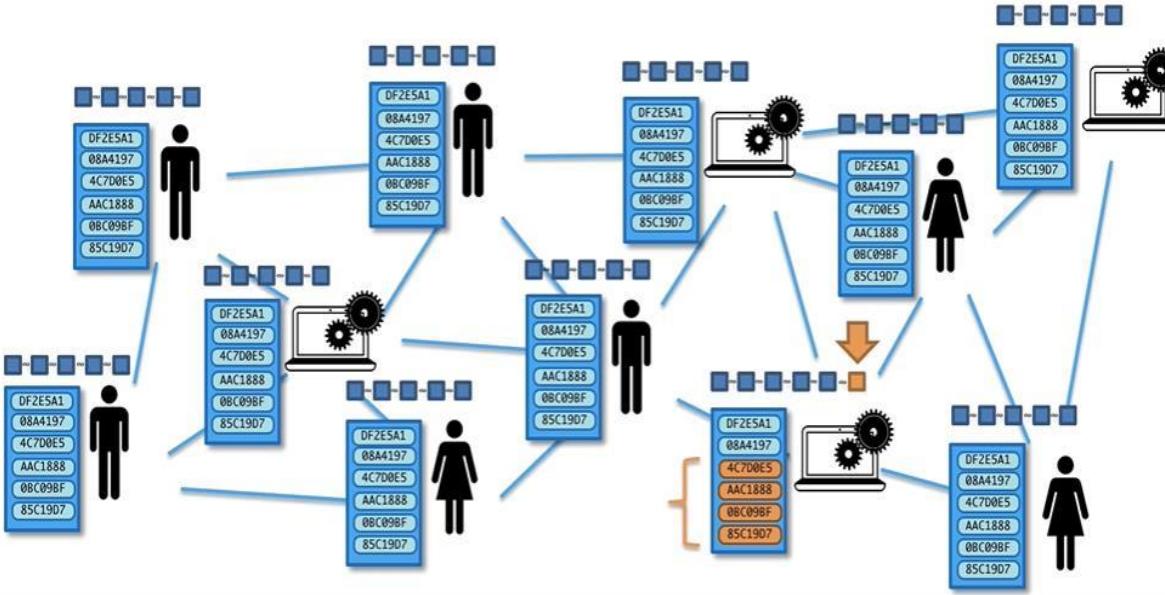


Mempools Functioning

- How mempools work in the distributed peer-to-peer system?
- The Blockchain network consists of nodes (miners and participants).
- A mempool is attached to each one of the nodes.
- The mempool is a staging area for transactions.
- The blocks are added to the blockchain at a certain regularity.
- The transactions occur very frequently.
- All nodes have a copy of the complete Blockchain.



Mempools Functioning



Mempools Functioning

- One node (miner/participant) wants to send some money to somebody.
- The transaction is added to the local mempool.
- The transaction gets broadcasted or relayed across the network.
- This transaction is added to the mempool of each node.
- The nodes ensure the validity of the transaction before including it in the mempool.
- Mempool has up to 10k or more transactions in a single day.
- The block contains 2000 or more transactions depending on the maximum size.
- The miners pick the transactions and add the transactions to the block.
- The block gets broadcasted across the network and it is added to the blockchain on every node.
- The transactions, which are included in the block, are removed from the mempool

Check www.blockchain.com for Mempool size, transaction count, growth, and unconfirmed transactions.

<https://mempool.space/>

<https://jochen-hoenicke.de/queue/#BTC,24h,count>

<https://blog.kaiko.com/an-in-depth-guide-into-how-the-mempool-works-c758b781c608>

Orphaned Blocks

- The total number of blocks mined but ultimately not attached to the main Bitcoin Blockchain.
- An orphan block is a block that has been mined within the Blockchain network but was not accepted by the network.
- There can be two miners who mine valid blocks simultaneously. The network uses both blocks until one chain has more verified blocks. Then, the blocks in the shorter chain are orphaned.
- Orphan blocks are a regular occurrence in a distributed Blockchain such as Bitcoin.
- The series of blocks that create a blockchain are related in that they receive information from the blocks that preceded them. When a block is closed, its data is encoded and passed on to the next block. These two blocks are a parent and child block.
- If two blocks are opened from the same parent block simultaneously, there are two child blocks. Only one of them can be integrated into the chain.

<https://www.blockchain.com/explorer/charts/n-orphaned-blocks>

https://bitcoinchain.com/block_explorer/orphaned

<https://www.investopedia.com/terms/o/orphan-block-cryptocurrency.asp>

Orphaned Blocks

- Orphan blocks are not an issue in blockchains that use proof-of-stake consensus because only one block at a time is proposed.
- The network nodes, which validate blocks, decide which block to use by allowing a small fork between the two child blocks. Then, the nodes determine what block they want to accept by reaching a validation consensus.
- Each block will have subsequent blocks created, initiating a race to verify the most blocks. The fork with more verified blocks—through proof of work (PoW)—gets accepted into the blockchain. Any verified blocks within the shorter chain are discarded (Longest Chain wins)
- The discarded block is called an orphan block (in technical documents, it's called a stale block). The transactions from the orphaned block go back to the memory pool to be validated and added to the new chain.
- An orphan block would technically be a block with unknown parent blocks.

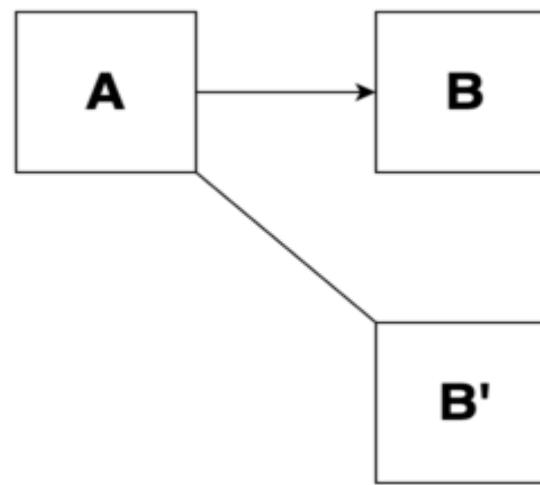
<https://www.blockchain.com/explorer/charts/n-orphaned-blocks>

https://bitcoinchain.com/block_explorer/orphaned

<https://www.investopedia.com/terms/o/orphan-block-cryptocurrency.asp>

Orphaned Blocks

- In blockchains using the PoW algorithm, unintentional forks happen when two miners on the network solve the PoW puzzle at a similar time. When these miners announce they have a “winning” block to the rest of the network (propagation), it is not instantaneous, due to latency.
- It will always take some amount of time before all nodes can become aware of a newly mined block.
- The nodes will keep the block they saw first at the head of their chain, but will also link the block they saw second to their parent as a reference.



<https://www.blockchain.com/explorer/charts/n-orphaned-blocks>

https://bitcoinchain.com/block_explorer/orphaned

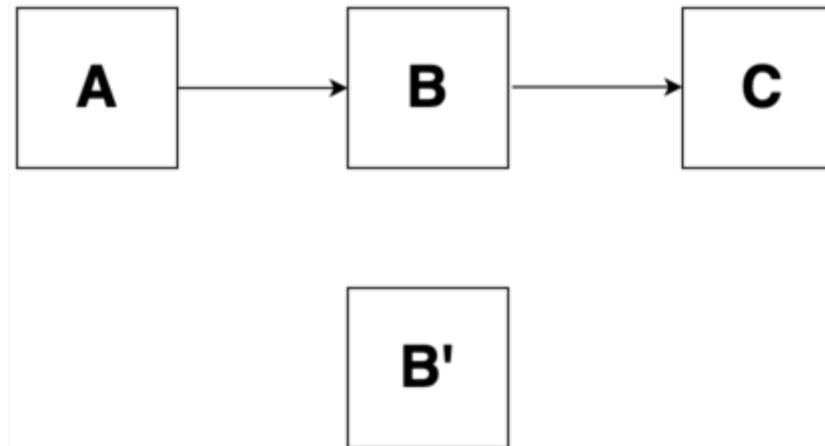
<https://www.investopedia.com/terms/o/orphan-block-cryptocurrency.asp>

Orphaned Blocks

- The Miners will accept the longest chain.
- B' transactions are then returned to the mempool, B' is discarded. It forces it an orphan block.

The miners of B' lose their reward for mining the block.

It is important to wait for at least six confirmations before considering your transaction to be successful.



<https://www.blockchain.com/explorer/charts/n-orphaned-blocks>

https://bitcoinchain.com/block_explorer/orphaned

<https://www.investopedia.com/terms/o/orphan-block-cryptocurrency.asp>

Orphaned Blocks

- If you received money from somebody to sell your bicycle, they sent you money and you give them the bicycle.
- It turns out that your block was not valid i.e., orphan.
- The transaction gets released back into the mempool and they get their money back and they keep it the bicycle because now you can't get hold of them or something like that.
- Thus is also known as double spend problem.
- It also leads to 51% attack.

<https://www.blockchain.com/explorer/charts/n-orphaned-blocks>

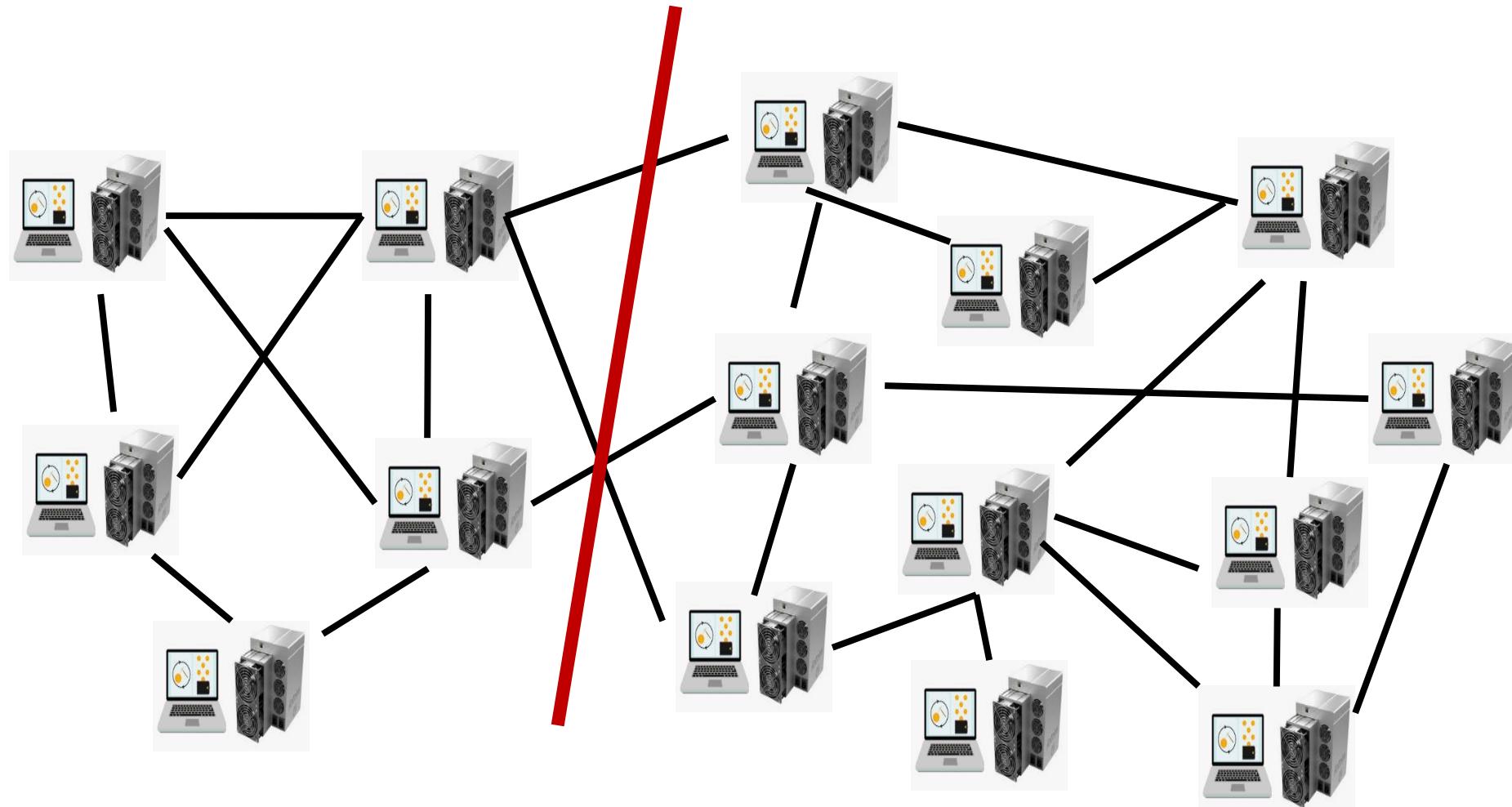
https://bitcoinchain.com/block_explorer/orphaned

<https://www.investopedia.com/terms/o/orphan-block-cryptocurrency.asp>

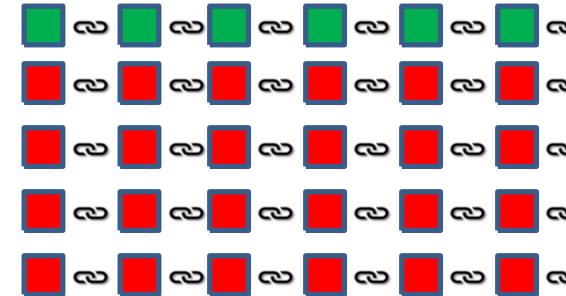
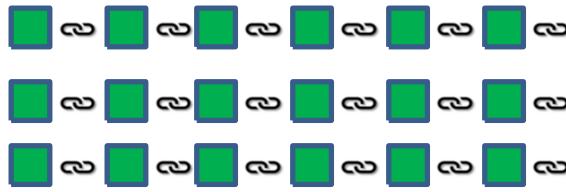
51% Attack

- The computational power of a miner is called hashing power and is measured in hashes per second (H/s).
 - All miners are competing with each other in a race where the winner gets to add the new block to the chain and receives in return a remuneration.
 - The higher the hashing power of a miner the higher his chances to win this competition.
-
- **51% attack occurs when a miner or group of miners(mining pool) manages to gain more than 51% of the network's mining power. Then the respective entity will have the power to control the entire network. If an attacker gets this power, he will be able to:**
 - **Double-spend his money. He can pay with the same cryptocurrency twice or even more.**
 - **Prevent transactions from being confirmed.**
 - **Prevent the generation of new bitcoins.**
-
- It will significantly reduce the trust and drops the price of cryptocurrency.
 - The 51% attack is easy on small network compared to large network.

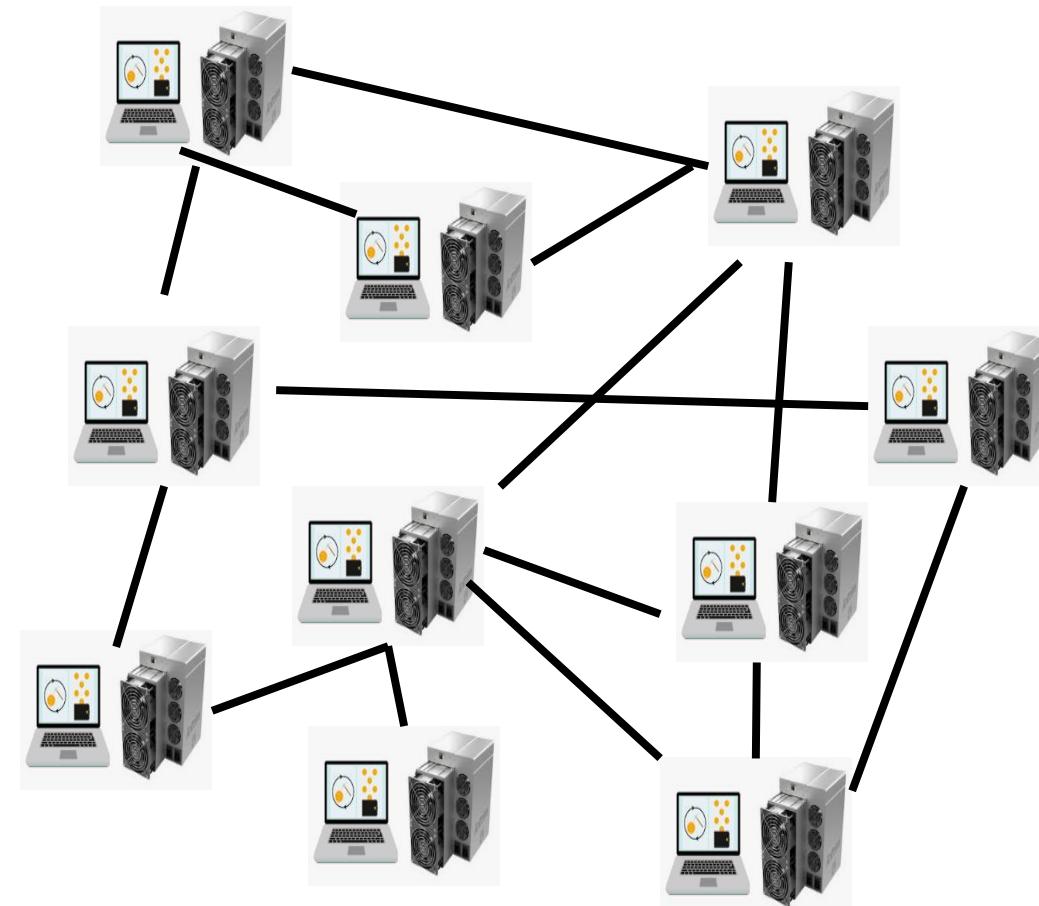
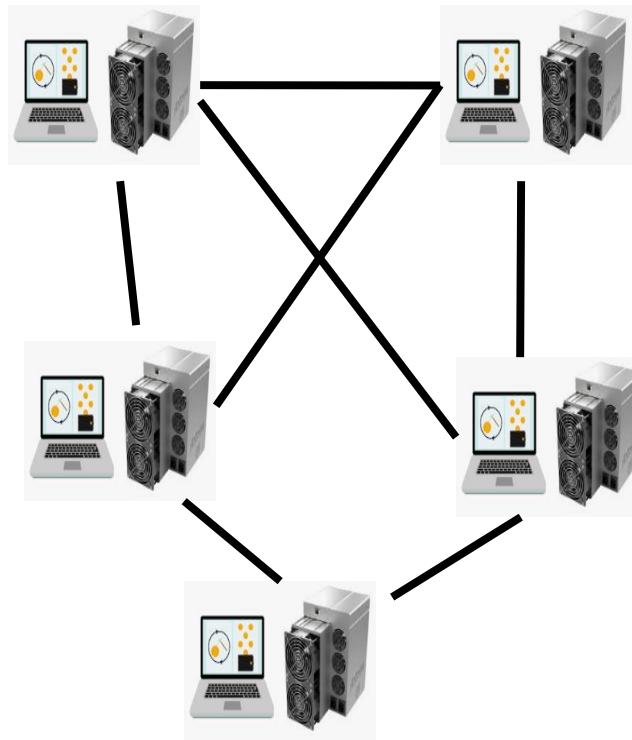
51% Attack



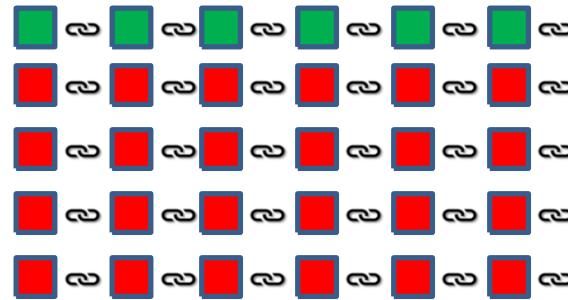
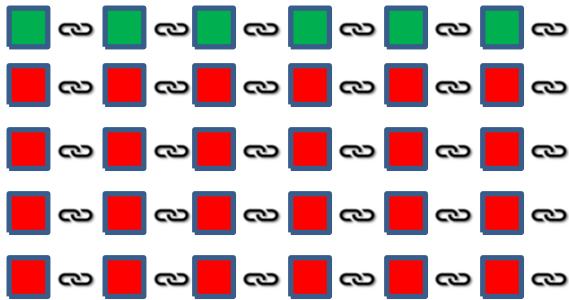
51% Attack



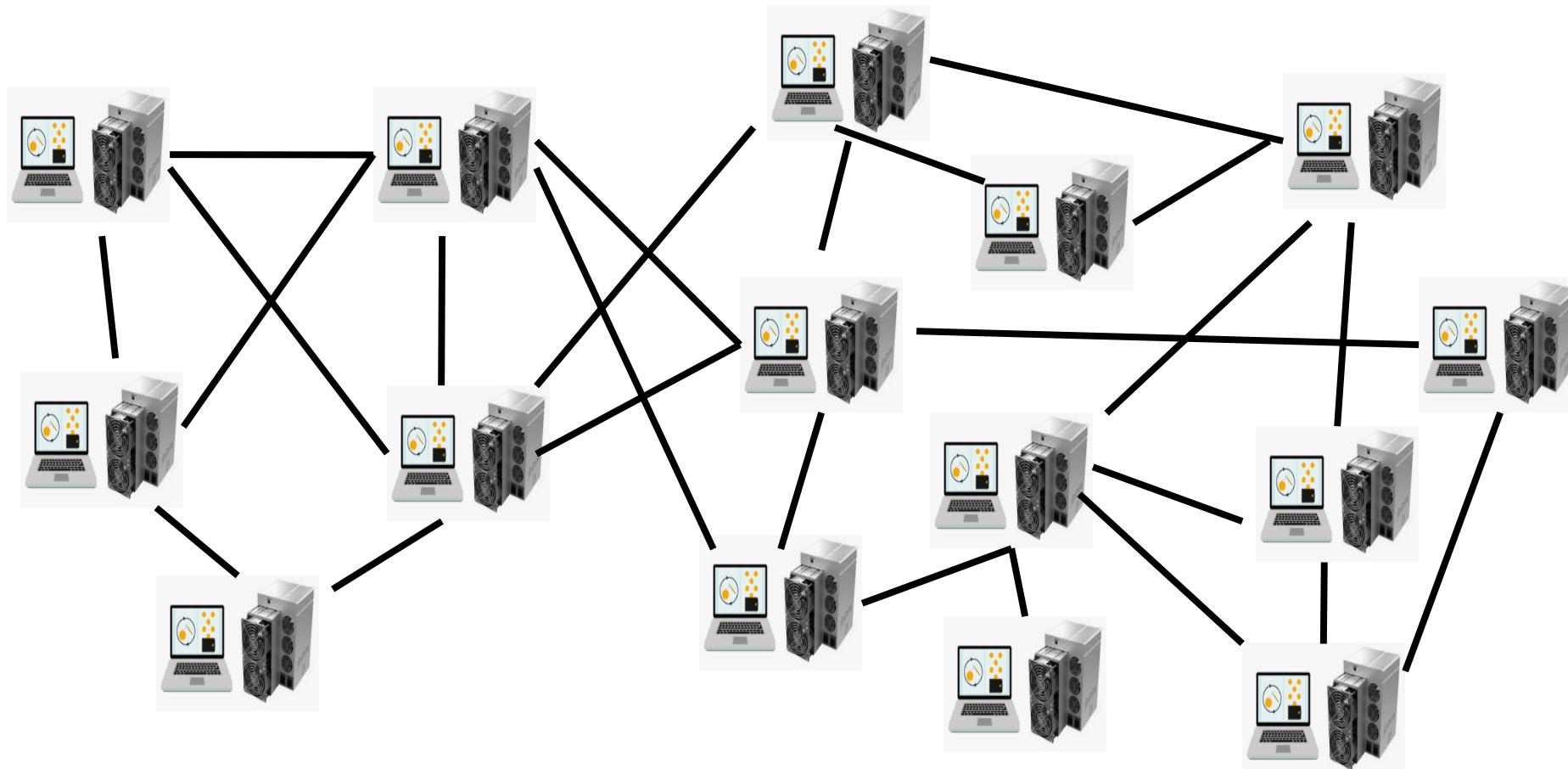
Disconnected Network
Longest chain is king
Longest chain wins



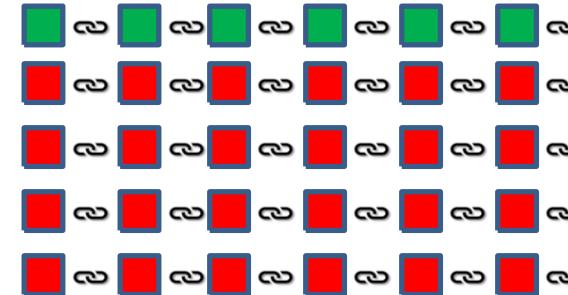
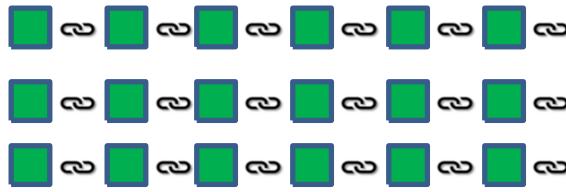
51% Attack



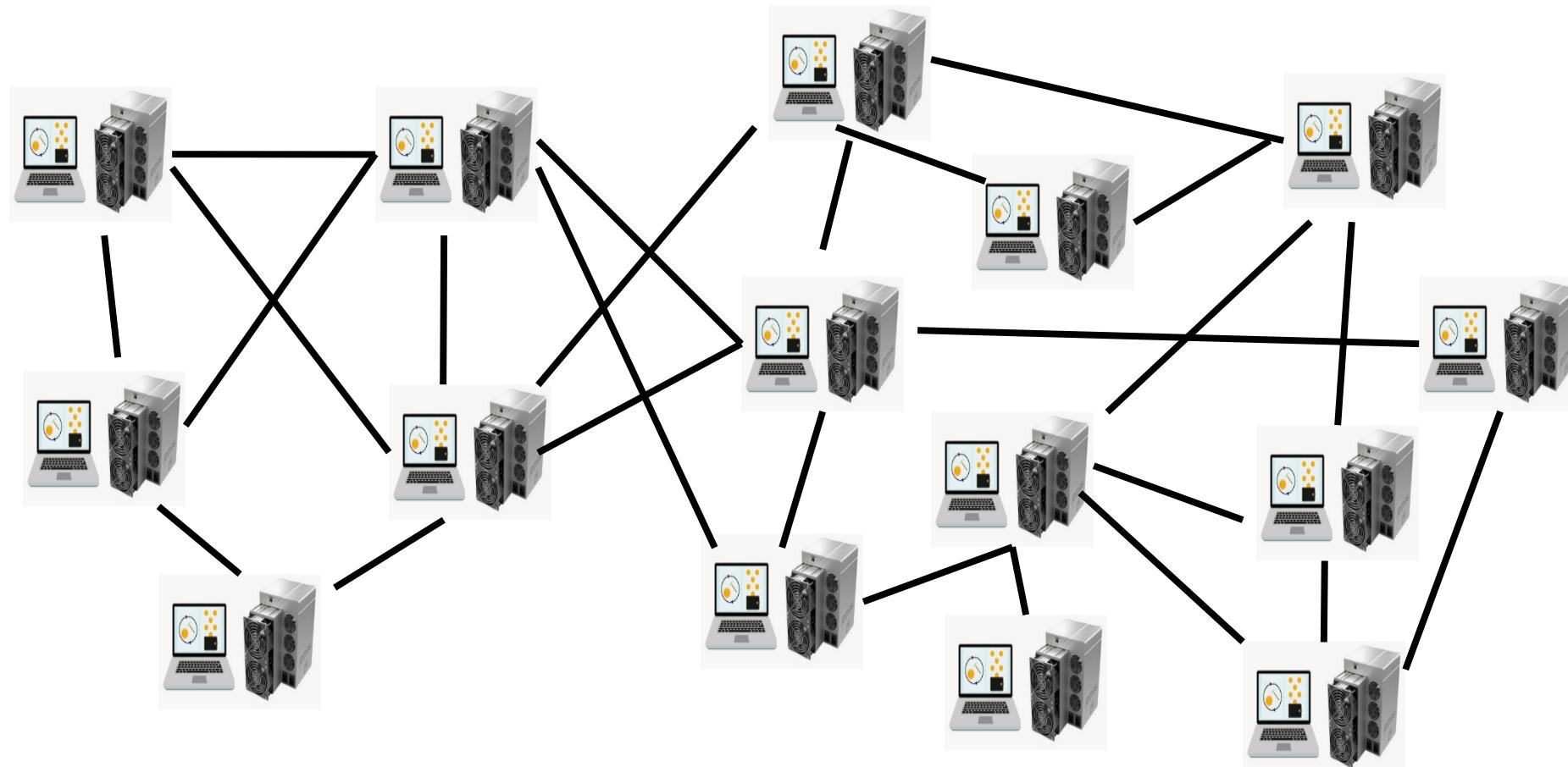
Disconnected Network
Longest chain is king
Longest chain wins



51% Attack



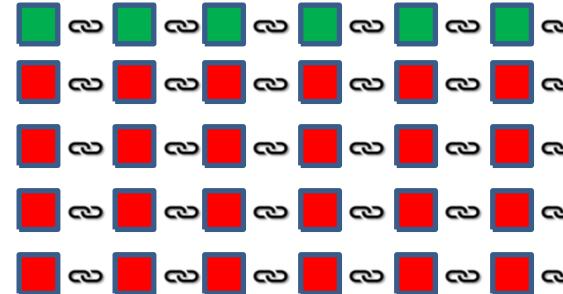
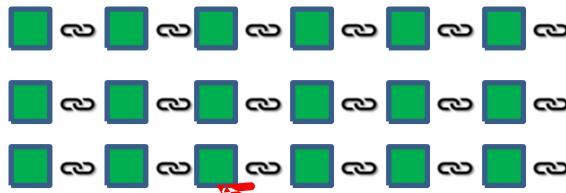
Disconnected Network
Longest chain is king
Longest chain wins



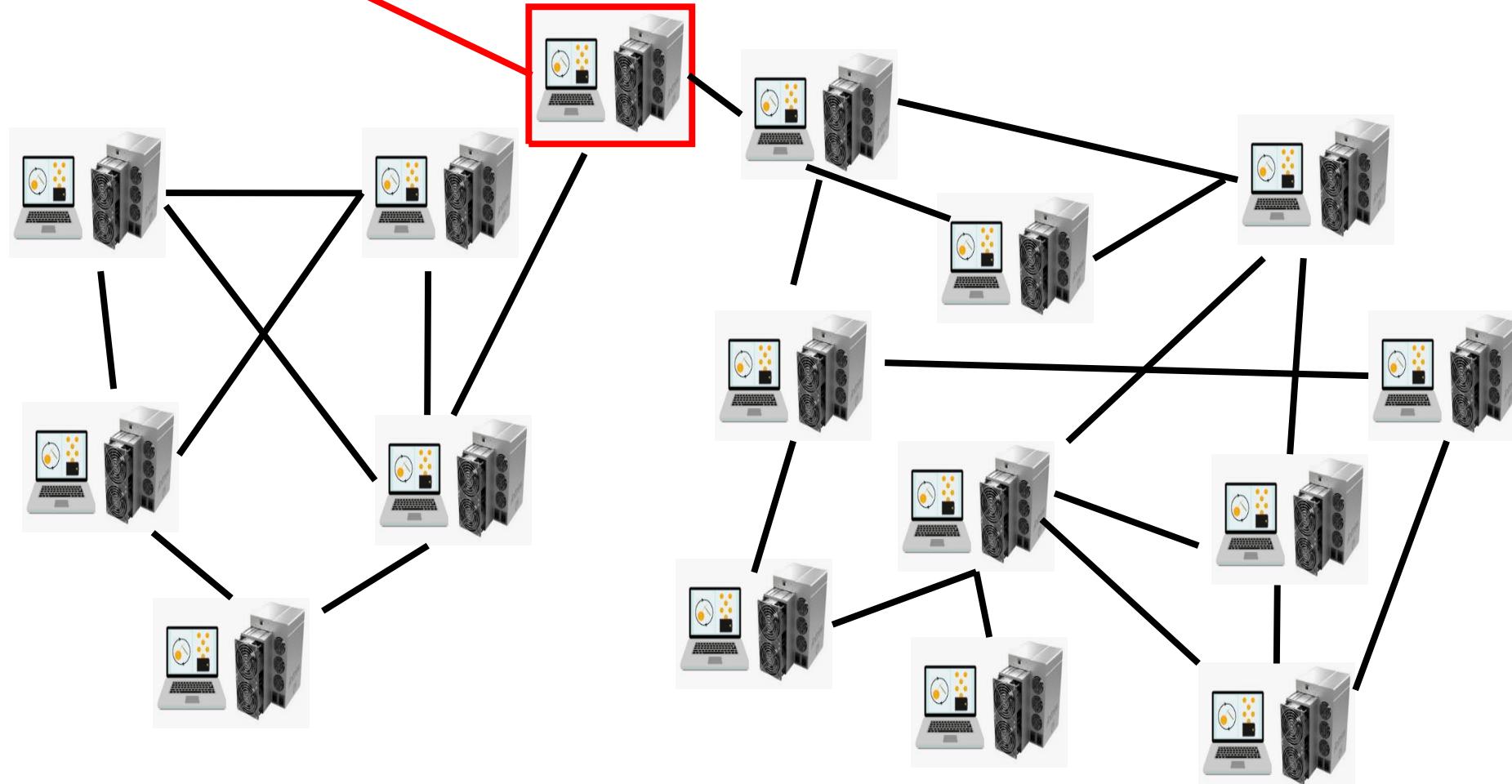
51% Attack

- The 51% attack does not try to disrupt or interfere with the consensus protocol.
- It follows the protocol's rules to attain the effect of changing the blockchain's content to take benefits.
- A group of miners that amount to more than 51% of the hashing power of the entire Bitcoin network decide to separate themselves from the rest of the network, while keeping the communication open within their own group.
- The disconnect network will join and will broadcast their chain through the entire network
- As per rule, the nodes will keep the longer version and delete the short one.
- **What happened to the blocks and transactions in the smaller chain ?**
 - The blocks mined by the remaining group, from the time of the separation to the time of the reunion, will be orphaned, and their transactions (or at least part of them) will be released back into the Mempool.

51% Attack



Disconnected Network
Longest chain is king
Longest chain wins



51% Attack

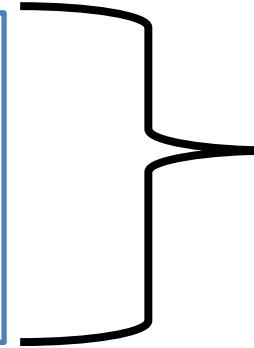
- This scenario opens the door to the possibility of double spending, in case the transactions released back into the Mempool were used to buy goods.
- The group of miners that originally broke up from the network can prevent those transactions from being picked and included in new blocks as they still have 51% (or more) of the total hashing power.
- If these transactions remain in the Mempool for more than a given time limit (for Bitcoin is 72 hours) then they are cancelled and the coins return to their original recipients (the buyer's wallet).
- This way the buyer will keep both the coins and the goods originally purchased with those coins.
- If the 51% attack is carried out in a planned and concerted way, the attackers could take advantage of this unresolved double spending and end up owning goods that they eventually didn't pay for.

Recommended article on Medium

- ❑ the 51% attack what is it
- ❑ Mining difficulty, Hash Power,Nonce Range and the like. An Introduction to Bitcoin Mining.
- ❑ Architecting a Digital Fortress. The Bitcoin's Consensus Protocol.

Transactions & UTXO's

Kamran	→	me	0.5 BTC
Rahul	→	me	0.9 BTC
Yasser	→	me	0.7 BTC
Rahul	→	me	0.6 BTC



Unspent Transaction
Output (UTXO)

I am interested to purchase a car.
The car price is 0.8 BTC

Transactions & UTXO's

Kamran	→	me	0.5 BTC
Rahul	→	me	0.9 BTC
Yasser	→	me	0.7 BTC
Rahul	→	me	0.6 BTC

Unspent Transaction Output (UTXO)

I am interested to purchase a car.
The car price is 0.8 BTC

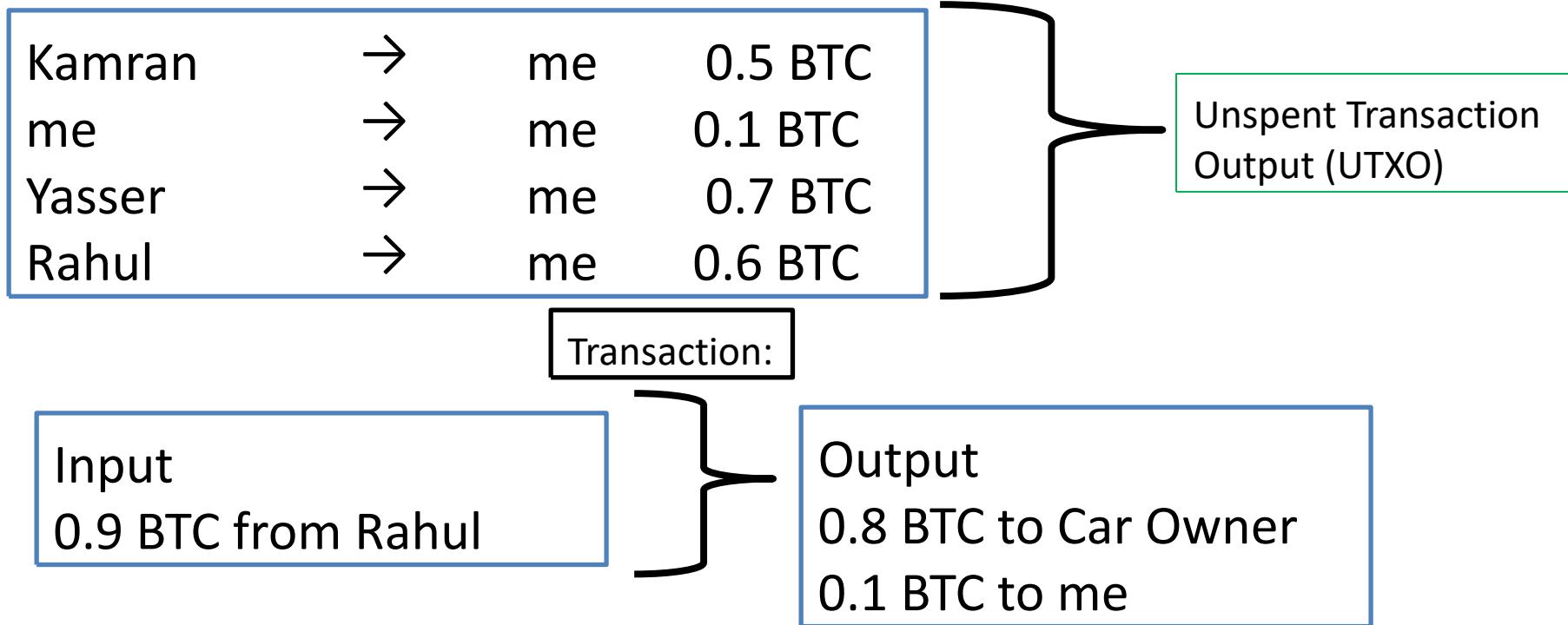
Transaction:

Input
0.9 BTC from Rahul

Output
0.8 BTC to Car Owner
0.1 BTC to me



Transactions & UTXO's



- I create a transaction that essentially says to the network:
- Take my 0.9 BTC UTXO as an input, break it up, send 0.8 BTC of it to Car owner address and return the 0.1 BTC to my address.
- The 0.9 BTC is now a spent output, and can't be reused.
- Meanwhile, two new UTXOs have been created (0.8 BTC and 0.1 BTC).

Transactions & UTXO's

Kamran	→	me	0.5 BTC
Me	→	me	0.1 BTC
Yasser	→	me	0.7 BTC
Rahul	→	me	0.6 BTC

Unspent Transaction Output (UTXO)

I am interested to purchase one more car.
The car price is 0.8 BTC

Transaction:

Input
0.7 BTC from Yasser
0.1 BTC from me

Output
0.8 BTC to Car Owner



Transactions & UTXO's

Kamran	→	me	0.5 BTC	Unspent Transaction Output (UTXO)
Rahul	→	me	0.6 BTC	



Transactions & UTXO's

- The UTXOs are small, unspent chunks of cryptocurrency leftover from transactions. They are recorded in the UTXO database and used in later transactions.
- UTXOs are processed continuously and are part of the beginning and end of each transaction.
- UTXO can be used as input in a new transaction.
- An UTXOs defines where each Blockchain transaction starts and finishes.
- When a transaction is completed, any unspent outputs are recorded into a database as inputs that can be used later for a new transaction.

Transactions & UTXO's

- The cryptocurrency transactions are made of inputs and outputs.
- Anytime a transaction is made, a user takes one or more UTXOs to serve as the input(s).
- Next, the user provides their digital signature to confirm ownership over the inputs, which finally result in outputs.
- The UTXOs consumed are now considered "spent," and can no longer be used.
- The outputs from the transaction become new UTXOs – which can be spent in a new transaction later.

Transactions & UTXO's : Goals

- The UTXO model is used in many cryptocurrencies because it allows users to track ownership of all portions of that cryptocurrency.
- The cryptocurrencies were created with anonymity in mind, UTXOs are associated with the public addresses visible to the entire network.
- Users cannot be identified from their ownership—unless they advertise their address—but the model allows for transparency through the addresses.

Transaction Fees

- The transaction fees are implemented in order to prevent spam transactions that could slow down and clog the network.
- Transaction fees incentivize miners to validate transactions and subsidize the diminishing block subsidy, helping support network security by keeping miners profitable.
- Mathematically, transaction fees are the difference between the amount of bitcoin sent and the amount received.
- The transaction fees are a reflection of the speed with which a user wants their transaction validated on the blockchain.
- When a miner validates a new block in the blockchain, they also validate all of the transactions within the block.
- Once a miner has validated a new block, they receive the transaction fees and block subsidy associated with that block. The sum of the transaction fees and block subsidy is the block reward.

Transaction Fees

Kamran	→	me	0.5 BTC
Rahul	→	me	0.9 BTC
Yasser	→	me	0.6 BTC
Rahul	→	me	0.6 BTC

Unspent Transaction Output (UTXO)

I am interested to purchase a watch and shirt.
The watch price is 0.8 BTC and shirt price is 0.5

Transaction:

Input

0.5 BTC from Kamran
0.9 BTC from Rahul

Output

0.8 BTC for Watch
0.5 BTC for Shirt
0.08 BTC to me

Transaction Fees

Me	→	me	0.08 BTC
Rahul	→	me	0.9 BTC
Yasser	→	me	0.6 BTC
Rahul	→	me	0.6 BTC

Unspent Transaction Output (UTXO)

I am interested to purchase a watch and shirt.
The watch price is 0.8 BTC and shirt price is 0.5

Transaction:

Input

0.5 BTC from Kamran
0.9 BTC from Rahul

Output

0.8 BTC for Watch
0.5 BTC for Shirt
0.08 BTC to me

The difference of input and output is 0.02.
It represents the fee collected by the bitcoin network

Transaction Fees

- Why Transaction fees ?
 - With each Bitcoin halving, the hashrate falls.
 - A falling hashrate simultaneously increases the cost of mining new blocks while decreasing the block rewards.
 - Validating new blocks takes significant computing power and energy, so rising transaction fees incentivize miners to continue validating new blocks.
 - Keeping miners in the market is essential to maintaining network security, and transaction fees play a significant role.
-
- If you wish to have your transaction confirmed immediately, your optimal fee rate may vary significantly. However, if you do not mind waiting, paying 2 sats/vByte will usually allow your transaction to be confirmed within a day or a week.

Transaction Fees

- Bitcoin fees are not dependent on the amount of cryptocurrency within a transaction but are based on the transaction size (in bytes).
- More complicated transactions, ones that have more inputs and outputs, will involve more data and therefore will be more expensive.
- The fees are measured in satoshis—the smallest divisible unit of bitcoin. A satoshi (sat) is 100 millionth of a bitcoin shown as 0.00000001 BTC.
- Bitcoin fees are shown as sats/vByte meaning satoshi per unit of data the transaction will consume. If a transaction is 400 bytes, and the average transaction fee is 80 satoshis per byte, you would pay 32,000 satoshis (or 0.00032 BTC) to have your transaction added to the next block.
- **Transaction fees are implied, as the excess of inputs minus outputs:**
- Fees = Sum(Inputs) – Sum(Outputs)

Transaction Fees

- Bitcoin Transaction Speed
- Transaction fees also reflect the speed with which the user wants to have a transaction validated. When a user initiates a bitcoin transaction, it goes into the mempool. Upon validation, it is included in the block. Miners choose which transactions to validate and include in the block. When there is a backlog of transactions waiting to be validated, it creates an incentive for miners to process transactions with higher fee rates first. Most miners target transactions with high fee to byte ratios. When network transactions begin to reduce, transaction fees will fall.
- You can check fee here (<https://jochen-hoenicke.de/queue/#BTC,24h,fee>

Transaction Fees

- Institutional Transaction Fees
- Bitcoin exchanges, which facilitate matching buyers and sellers, calculate their fee in two ways:
 - A flat fee per transaction, or as a percentage of 30-day total transaction volume.
 - In both cases, exchanges implement a tiered fee structure based on the total dollar volume traded.

Transaction Fees

- Bitcoin fees are not dependent on the amount of cryptocurrency within a transaction but are based on the transaction size (in bytes).
- More complicated transactions, ones that have more inputs and outputs, will involve more data and therefore will be more expensive.
- The fees are measured in satoshis— the smallest divisible unit of bitcoin. A satoshi (sat) is 100 millionth of a bitcoin shown as 0.00000001 BTC.
- Bitcoin fees are shown as sats/vByte meaning satoshi per unit of data the transaction will consume. If a transaction is 400 bytes, and the average transaction fee is 80 satoshis per byte, you would pay 32,000 satoshis (or 0.00032 BTC) to have your transaction added to the next block.

Cryptocurrency Wallets

- ❑ A cryptocurrency wallet is a device or program that stores your cryptocurrency keys and allows you to access your coins.
- ❑ Wallets contain a public key (the wallet address) and your private keys needed to sign cryptocurrency transactions.
- ❑ Anyone who knows the private key can control the coins associated with that address.
- ❑ There are several different types of wallets, each with its own features and levels of security.
- ❑ Many cryptocurrency wallets can be used to store key for different cryptocurrencies.

- ❑ Bitcoin's developer, Satoshi Nakamoto used first wallet.
- ❑ The second wallet belonged to Hal Finney, who corresponded with Nakamoto and reportedly was the first to run the Bitcoin client software wallet.
- ❑ Nakamoto sent him 10 bitcoin as a test, and the cryptocurrency craze began.

Cryptocurrency Wallets

- ❑ Cryptocurrencies are not "stored" anywhere.
- ❑ They are bits of data stored in a database.
- ❑ These bits of data are scattered all over the database;
- ❑ **The wallet finds all of the bits associated with your public address and sums up the amount for you in the app's interface.**

- ❑ Sending and receiving cryptocurrency is very easy using these applications.
 - ❑ Enter the recipient's wallet address
 - ❑ Choose an amount to send
 - ❑ Sign the transaction using your private key
 - ❑ Add an amount to pay the transaction fee
 - ❑ Send it.

How Cryptocurrency Wallets work ?

- ❑ When you log into your bank account, there's a number and that is actually the balance of your account. (For example, INR 10,000)
- ❑ In Blockchain technology, there is no concept of "balance"
 - ❑ It doesn't actually exist.
 - ❑ Cryptocurrencies are not "stored" anywhere.
 - ❑ The balance of your cryptocurrency is the combination of the UTXOs scattered all over the database;

How Cryptocurrency Wallets work ?

I am interested to purchase a car.
The car price is 0.8 BTC

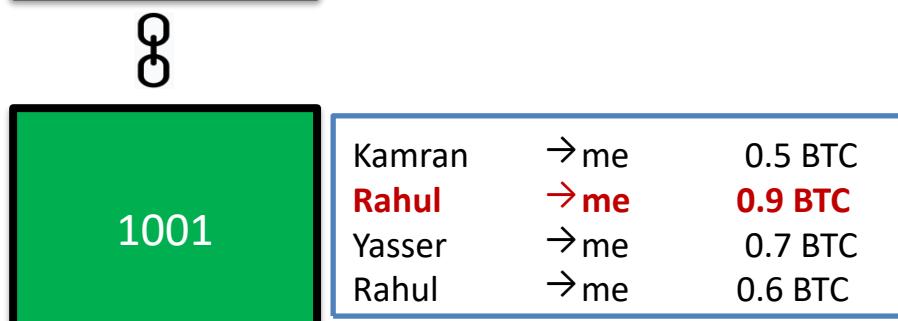
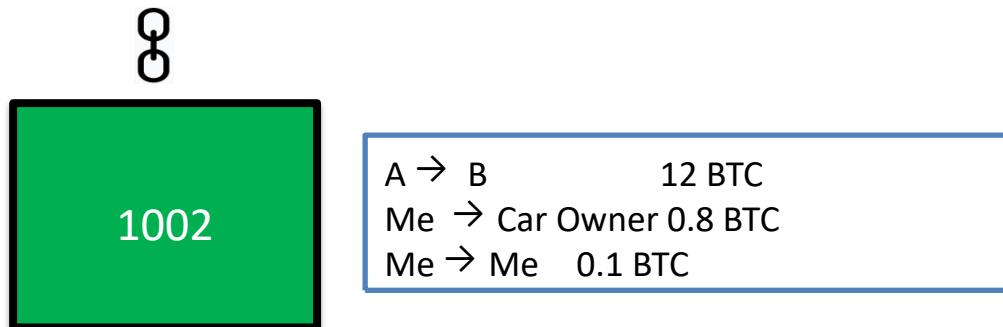
Q

1001

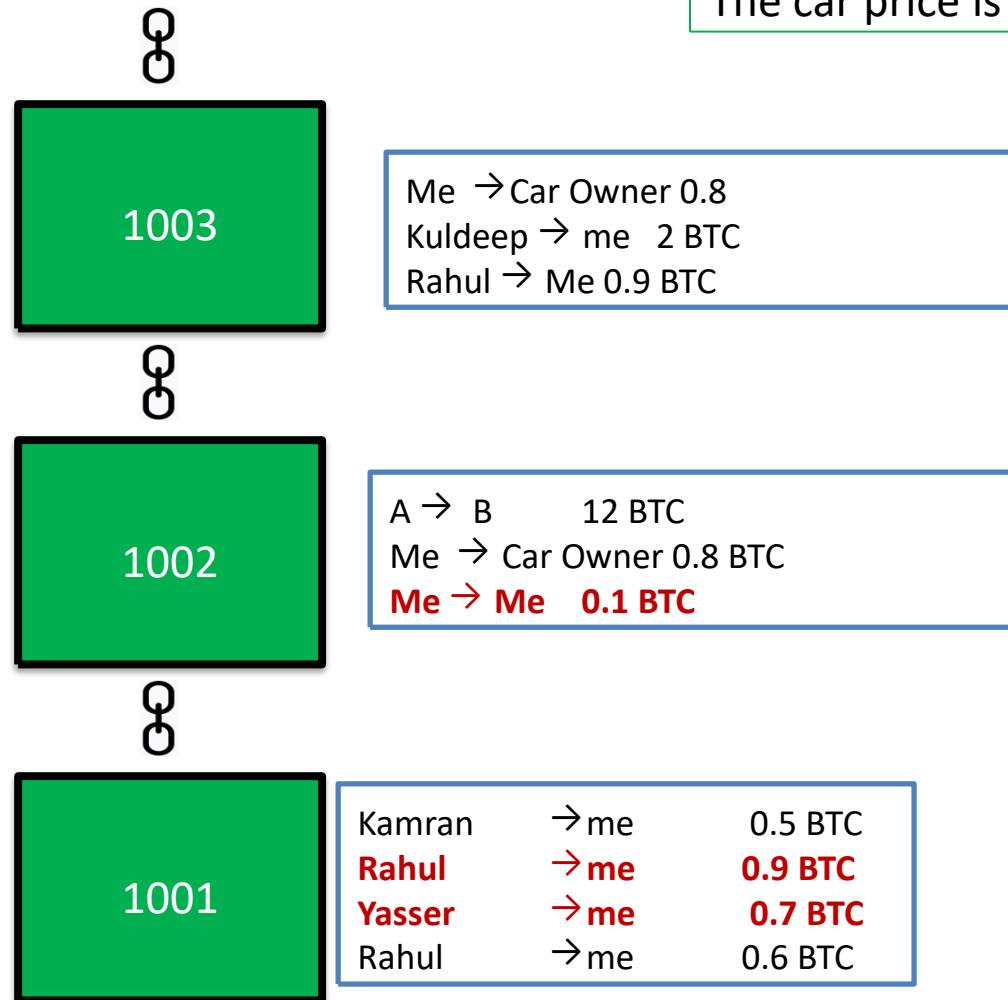
Kamran	→ me	0.5 BTC
Rahul	→ me	0.9 BTC
Yasser	→ me	0.7 BTC
Rahul	→ me	0.6 BTC

How Cryptocurrency Wallets work ?

I am interested to purchase one more car.
The car price is 0.8 BTC



How Cryptocurrency Wallets work ?



I am interested to purchase one more car.
The car price is 0.8 BTC

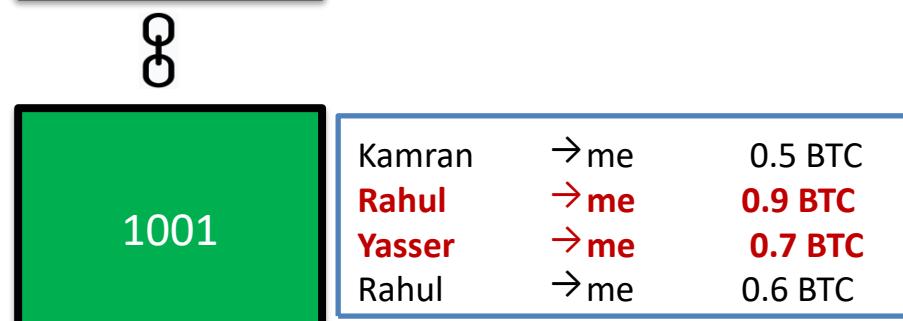
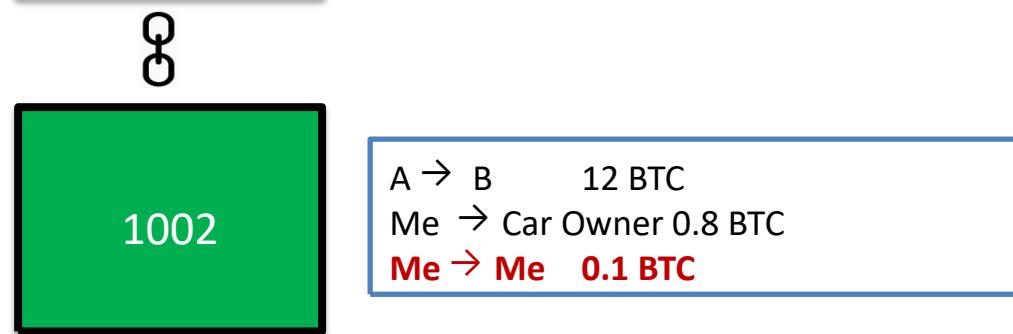
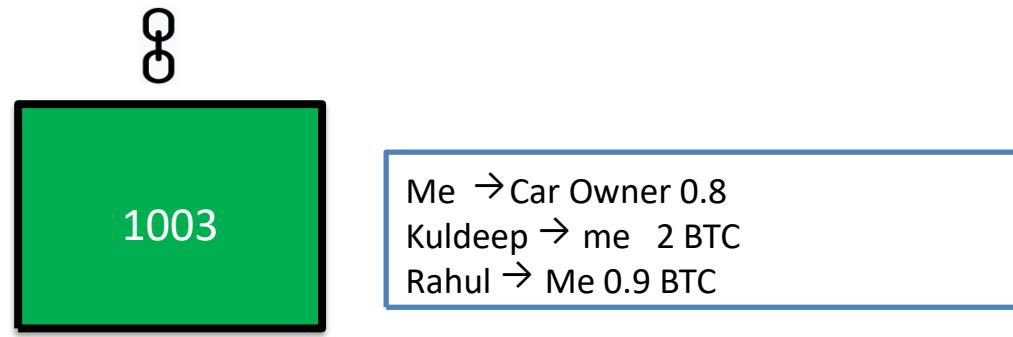
Me → Car Owner 0.8
Kuldeep → me 2 BTC
Rahul → Me 0.9 BTC

A → B 12 BTC
Me → Car Owner 0.8 BTC
Me → Me 0.1 BTC

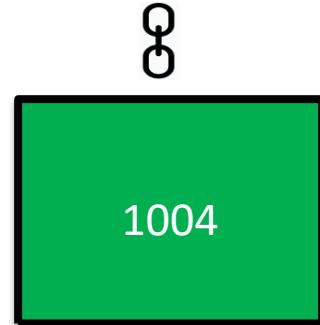
Kamran → me 0.5 BTC
Rahul → me 0.9 BTC
Yasser → me 0.7 BTC
Rahul → me 0.6 BTC

How Cryptocurrency Wallets work ?

I am interested to purchase a watch and shirt.
The watch price is 0.8 BTC and shirt price is 0.5



How Cryptocurrency Wallets work ?



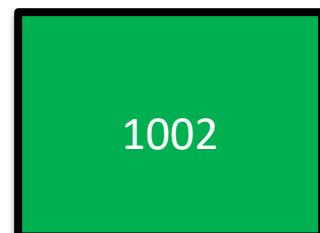
Me → Watch Shop 0.8 BTC
Me → Shirt Shop 0.5 BTC
Me → me 0.08 BTC

I am interested to purchase a watch and shirt.
The watch price is 0.8 BTC and shirt price is 0.5

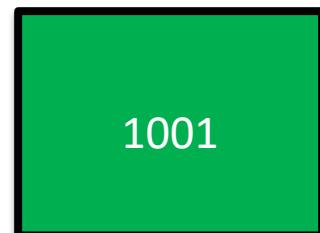


Me → Car Owner 0.8
Kuldeep → me 2 BTC
Rahul → Me 0.9 BTC

How balance is computed by the wallet ?

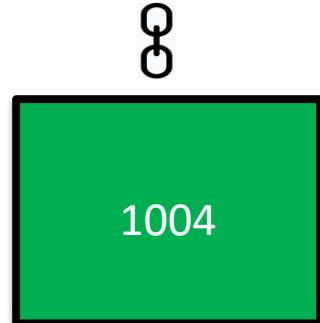


A → B 12 BTC
Me → Car Owner 0.8 BTC
Me → Me 0.1 BTC



Kamran	→ me	0.5 BTC
Rahul	→ me	0.9 BTC
Yasser	→ me	0.7 BTC
Rahul	→ me	0.6 BTC

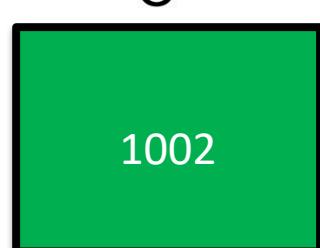
How Cryptocurrency Wallets work ?



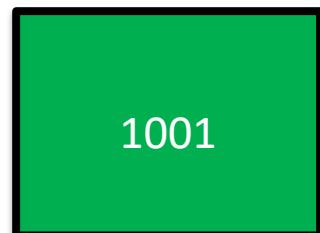
Me → Watch Shop 0.8 BTC
Me → Shirt Shop 0.5 BTC
Me → me 0.08 BTC



Me → Car Owner 0.8
Kuldeep → me 2 BTC
Rahul → Me 0.9 BTC



A → B 12 BTC
Me → Car Owner 0.8 BTC
Me → Me 0.1 BTC



Kamran	→ me	0.5 BTC
Rahul	→ me	0.9 BTC
Yasser	→ me	0.7 BTC
Rahul	→ me	0.6 BTC

The wallet calculates total UTXOs that are available in the Blockchain.

Wallet

- Identifies all transaction which are inputs to some transactions
- Checks all transactions which still have unspent transaction outputs.
- Add up these amounts
- Shows the balance

My balance is $0.6 + 2 + 0.08 = 2.68$ BTC

<https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch05.html#:~:text=The%20concept%20of%20a%20user's,UTXO%20belonging%20to%20that%20user.>

Segregated Witness (SegWit)

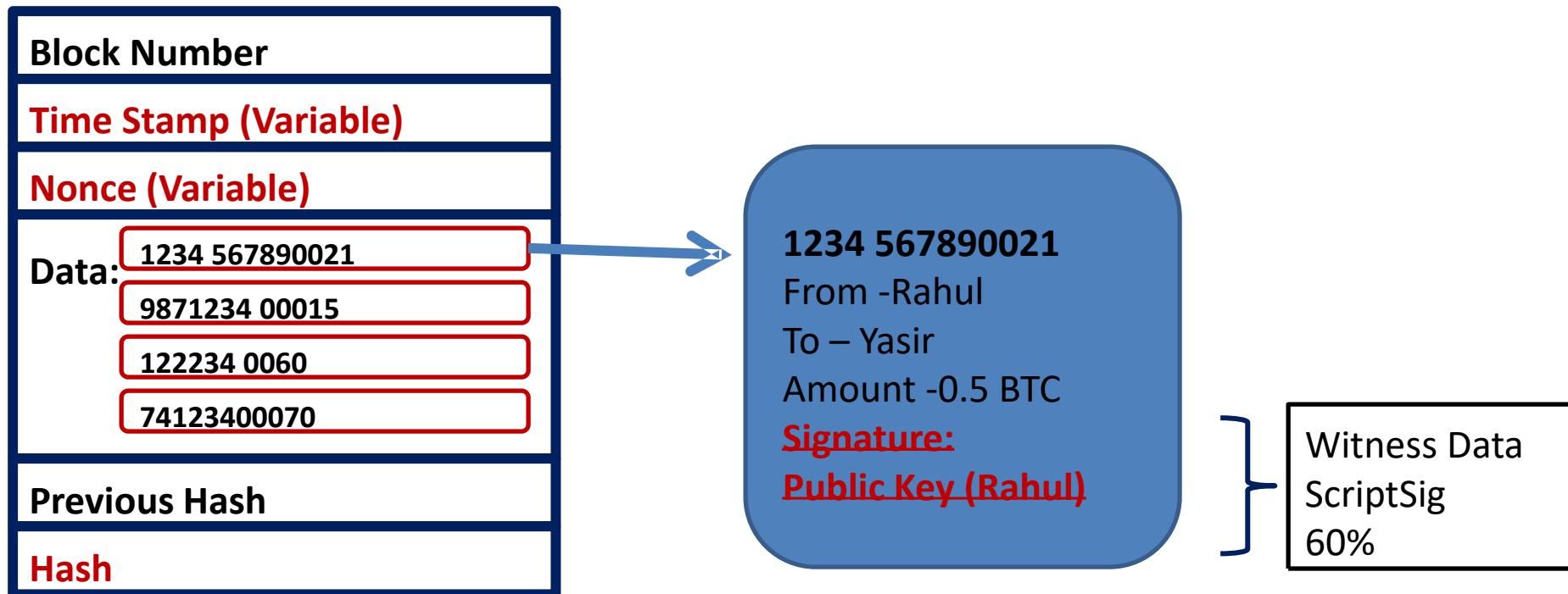
- ❑ There is limit of one megabyte for the block size. (It was included in the original design).
- ❑ Small Size (< 1 MB)
 - ❑ Very few transactions can be included. People have to wait too long for their transactions to confirm.
- ❑ Large Size (> 1 MB)
 - ❑ The network will be slow because of consensus and other communications.
 - ❑ After mining a new block, it will take too much time to copy the block on every node.
 - ❑ More chances of orphaned blocks, more chances of competing chains, and more chances for attackers to take advantage of the Blockchain and do something because information is propagating so slowly.
- ❑ **People noticed that the transaction confirmation require much time. ????**

Segregated Witness (SegWit)

- ❑ People noticed that the transaction confirmation require much time. ????
 - ❑ A fork was proposed.
 - ❑ Fork is an upgrade to “How the Blockchain works, which is not mandatory for everybody”
 - ❑ It can propagate over the network with time and people can accept it slowly.
 - ❑ Two options-
 - ❑ Hard fork
 - ❑ Forward compatible
 - ❑ Soft fork
 - ❑ Backward compatible
- ❑ Segregated Witness (SegWit)
 - ❑ Soft Form
 - ❑ It was adopted by the Bitcoin Blockchain in August 2017.

Segregated Witness (SegWit)

- The maximum block size in the main protocol is 1MB, which restricts the number of transactions Bitcoin can process to approximately 7 per second.
- This was going to limit Bitcoin's potential growth, and prevent it from becoming a usable high-volume payment system.
- This upgrade does enable a greater number of transactions in Bitcoin's blocks.



Segregated Witness (SegWit)

- ❑ It is necessary to include the signature and public key to the transaction.
- ❑ It enables people to verify who has signed the transaction.
- ❑ The issue is that this signature and public, are huge numbers.
- ❑ Transactions are small compared to witness data.
- ❑ They take up to 60% of the whole transaction size, and yet they're not the main purpose of the transaction.
- ❑
- ❑ SegWit
 - ❑ It was proposed to strip the heavy part (script sig) from the transaction.
 - ❑ The script sig will go through the network separately
 - ❑ It reduces transaction size and more transactions can be included.
- ❑ Why is it known SegWit?
 - ❑ The public key and signature data is known as witness.
 - ❑ Segregated witness means that miners are segregating the witness

https://medium.com/@akshay_111meher/segwit-block-size-and-block-weights-f5864a6133fc

Segregated Witness (SegWit)

- A protocol upgrade that changes the way data is stored.
- It was activated on litecoin on May 10, 2017, and on bitcoin in August, 2017.
- SegWit's initial intention was to fix a bug in the Bitcoin code called transaction malleability. This flaw allowed anyone to change small details that modified the transaction id (and the subsequent hash) but not the content.
- SegWit fixed transaction malleability by removing the signature information (otherwise known as the “witness” information) and storing it outside the base transaction block.
- SegWit introduced a new concept called “block weight.”
- This is a mashup of the block size with and without the signature data, and is capped at 4MB, while the block size limit for the base transactions remains at 1MB.
- This means that the SegWit upgrade is compatible with the previous protocol, and avoids the need for a hard fork.

https://medium.com/@akshay_111meher/segwit-block-size-and-block-weights-f5864a6133fc
<https://www.investopedia.com/terms/s/segwit-segregated-witness.asp>

Block Weight

- ❑ A new concept of weights was introduced with the segwit.
- ❑ A segwit transaction was divided into two parts:
 - ❑ Segwit Part of Transaction
 - ❑ The witness of a transaction is classified as segwit part of a transaction.
 - ❑ Non-Segwit Part of Transaction
 - ❑ All the other parts of transactions except witness are classified as Non-Segwit part of a transaction.
 - ❑ A Segwit Transaction has two parts, i.e., (Segwit part and Non-Segwit part)
 - ❑ A non-Segwit transaction has only one part (Non-Segwit part)
 - ❑ The weight of any transaction is defined by a simple formula:
 $3*(\text{non-segwit-part-of-transaction}) + 1*(\text{segwit-part-of-transaction})$
- ❑ **A block weight is defined as the sum of weights of all transactions in the block.**

https://medium.com/@akshay_111meher/segwit-block-size-and-block-weights-f5864a6133fc
<https://www.investopedia.com/terms/s/segwit-segregated-witness.asp>

Block Weight

- ❑ Block Size → Sum of sizes of all non-segwit data in a block + headers.
- ❑ Segwit Block Size → Sum of sizes of all non-segwit data in block + Sum of all segwit data in block + headers
- ❑ Block weight → Sum of weights of all transactions in a block.
- ❑ Bitcoin imposes a rule of 1MB max size on Block Size, not Segwit Block Size.
- ❑ Check the block size and weight in www.blockchain.com
- ❑ The block may be greater than 1 MB, but, the block weight must be less or equal to 4MBU.
- ❑ After segwit data is removed from this block, the Block Size will be less than 1Mb.

https://medium.com/@akshay_111meher/segwit-block-size-and-block-weights-f5864a6133fc
<https://www.investopedia.com/terms/s/segwit-segregated-witness.asp>

Block Weight

- ❑ Legacy nodes only receive the input and output data of segwit transactions.
- ❑ $\text{legacy_block_size} = \sum(\text{size_of_non-segwit-data_of_each_transaction})$
- ❑ $\text{segwit_block_size} = \text{legacy_block_size} + \sum(\text{size_of_segwit-data_of_each_transaction})$

- ❑ For a block, the non_segwit_weight and segwit_weight is defined as
 - $\text{non_segwit_weight} = 3 * \sum(\text{size_of_non-segwit-data_of_each_transaction})$
 - $\text{segwit_weight} = 1 * \sum(\text{size_of_segwit-data_of_each_transaction})$
 - $\text{block_weight} = \text{non_segwit_weight} + \text{segwit_weight}$
- ❑ For a block to be valid
 - ❑ $\text{legacy_block_size} \leq 1 \text{ MB}$
 - ❑ $\text{block_weight} \leq 4 \text{ MBU}$

https://medium.com/@akshay_111meher/segwit-block-size-and-block-weights-f5864a6133fc
<https://www.investopedia.com/terms/s/segwit-segregated-witness.asp>

Block Weight

- ❑ If you run a Segwit node, you will receive the segwit blocks which may be more than 1MB, whereas if you run a non-segwit node all the witnesses from the segwit block will be trimmed off and the block becomes a legacy block. This legacy block will be less than 1MB.
- ❑ However, this was not universally accepted as some among the community were not comfortable with Segwit, as some of the nodes won't have witnesses/signatures. This resulted in chain splitting and created BCH (Bitcoin Cash). BCH now has an adjustable block size.

https://medium.com/@akshay_111meher/segwit-block-size-and-block-weights-f5864a6133fc
<https://www.investopedia.com/terms/s/segwit-segregated-witness.asp>

Bitcoin Address

- ❑ Ownership of bitcoin is established through digital keys, bitcoin addresses, and digital signatures.
- ❑ The digital keys are not actually stored in the network, but are instead created and stored by users in a file, or simple database, called a wallet.
- ❑ The digital keys in a user's wallet are completely independent of the bitcoin protocol and can be generated and managed by the user's wallet software without reference to the blockchain or access to the Internet.
- ❑ Every bitcoin transaction requires a valid signature to be included in the blockchain, which can only be generated with valid digital keys; therefore, anyone with a copy of those keys has control of the bitcoin in that account.
- ❑ Keys come in pairs consisting of a private (secret) key and a public key.
- ❑ The public key allows you to receive transactions, while the private key is necessary to send transactions.
- ❑ The two keys are connected to each other with the clever use of mathematics.
- ❑ The unique public key has its origins in the private key. This connection allows users to create unforgeable signatures, which can only be validated by other participants of the network who have knowledge of the corresponding public key.

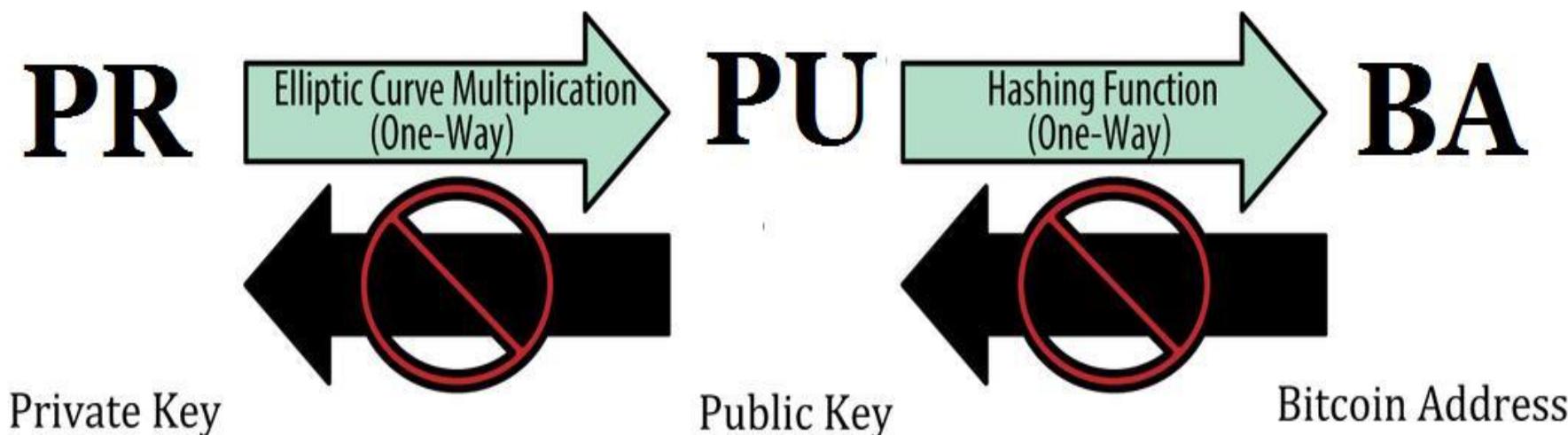
Bitcoin Address

- ❑ In the payment portion of a bitcoin transaction, the recipient's public key is represented by its digital fingerprint, called a bitcoin address, which is used in the same way as the beneficiary name on a check (i.e., "Pay to the order of").
- ❑ In most cases, a bitcoin address is generated from and corresponds to a public key.
- ❑ The bitcoin address is the only representation of the keys that users will routinely see, because this is the part they need to share with the world.
- ❑ When spending bitcoins, the current bitcoin owner presents her public key and a signature (different each time, but created from the same private key) in a transaction to spend those bitcoins. Through the presentation of the public key and signature, everyone in the bitcoin network can verify and accept the transaction as valid, confirming that the person transferring the bitcoins owned them at the time of the transfer.

<https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch04.html#:~:text=The%20bitcoin%20address%20is%20derived,of%20an%20arbitrary%2Dsized%20input.>

Bitcoin Address

- ❑ A bitcoin wallet contains a collection of key pairs, each consisting of a private key and a public key. The private key (k) is a number, usually picked at random.
- ❑ From the private key, we use elliptic curve multiplication, a one-way cryptographic function, to generate a public key (K).
- ❑ From the public key (K), we use a one-way cryptographic hash function to generate a bitcoin address (A).



<https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch04.html#:~:text=The%20bitcoin%20address%20is%20derived,of%20an%20arbitrary%20size%20input.>

Bitcoin Address

- A bitcoin address is a string of digits and characters that can be shared with anyone who wants to send you money.
- Addresses produced from public keys consist of a string of numbers and letters, beginning with the digit “1”. Here’s an example of a bitcoin address:
- Here is the Bitcoin genesis address - the first Bitcoin address ever:
1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa.
- The bitcoin address is what appears most commonly in a transaction as the “recipient” of the funds. If we were to compare a bitcoin transaction to a paper check, the bitcoin address is the beneficiary, which is what we write on the line after “Pay to the order of.” On a paper check, that beneficiary can sometimes be the name of a bank account holder, but can also include corporations, institutions, or even cash. Because paper checks do not need to specify an account, but rather use an abstract name as the recipient of funds, that makes paper checks very flexible as payment instruments.

https://medium.com/@akshay_111meher/segwit-block-size-and-block-weights-f5864a6133fc
<https://www.investopedia.com/terms/s/segwit-segregated-witness.asp>

Bitcoin Address

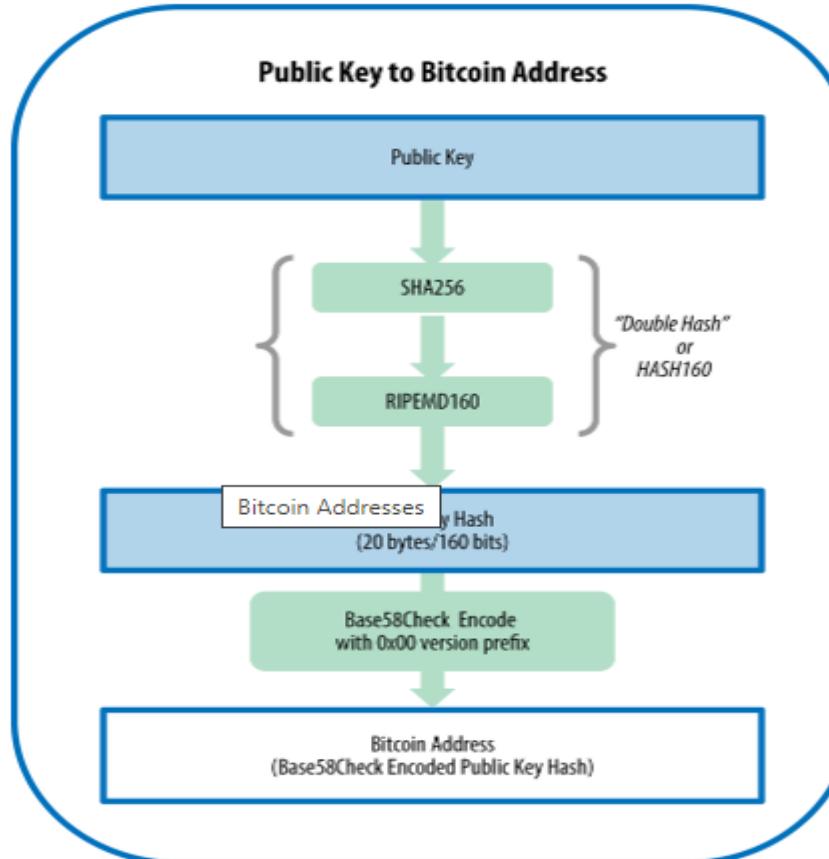
- ❑ Bitcoin transactions use a similar abstraction, the bitcoin address, to make them very flexible. A bitcoin address can represent the owner of a private/public key pair, or it can represent something else, such as a payment script, as we will see in Pay-to-Script-Hash (P2SH). For now, let's examine the simple case, a bitcoin address that represents, and is derived from, a public key.
- ❑ The bitcoin address is derived from the public key through the use of one-way cryptographic hashing. A “hashing algorithm” or simply “hash algorithm” is a one-way function that produces a fingerprint or “hash” of an arbitrary-sized input. Cryptographic hash functions are used extensively in bitcoin: in bitcoin addresses, in script addresses, and in the mining proof-of-work algorithm. The algorithms used to make a bitcoin address from a public key are the Secure Hash Algorithm (SHA) and the RACE Integrity Primitives Evaluation Message Digest (RIPEMD), specifically SHA256 and RIPEMD160.

https://medium.com/@akshay_111meher/segwit-block-size-and-block-weights-f5864a6133fc
<https://www.investopedia.com/terms/s/segwit-segregated-witness.asp>

Bitcoin Address

- ❑ Starting with the public key K, we compute the SHA256 hash and then compute the RIPEMD160 hash of the result, producing a 160-bit (20-byte) number:
- ❑ Bitcoin Addresses

$$A = \text{RIPEMD160}(\text{SHA256}(K))$$



https://medium.com/@akshay_111meher/segwit-block-size-and-block-weights-f5864a6133fc
<https://www.investopedia.com/terms/s/segwit-segregated-witness.asp>

Bitcoin Address: Summary

- Anyone can send Bitcoin using either Bitcoin address or public key.
- The reason is that it is better to keep the public key also safe.
- It is mandatory to expose the public key to send money for the verification purpose.
- We can avoid exposing it when we receive money, we should do that and that's why we use the address.
- Reddit to a question, what's the difference between the public and public address?

https://medium.com/@akshay_111meher/segwit-block-size-and-block-weights-f5864a6133fc

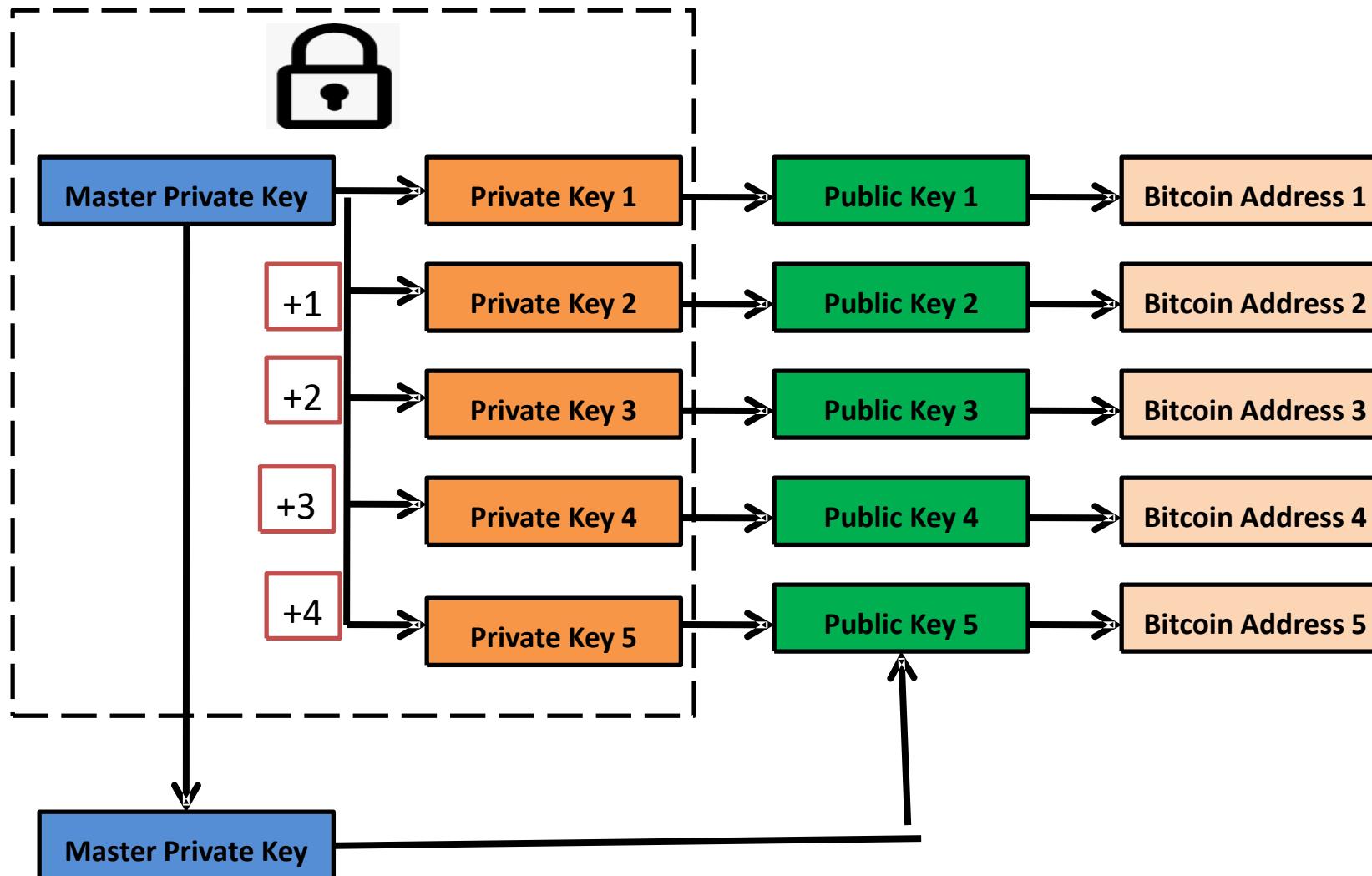
<https://www.investopedia.com/terms/s/segwit-segregated-witness.asp>

Hierarchically Deterministic (HD) Wallets

- ❑ The question is about privacy.
- ❑ Cryptocurrencies are meant to be anonymous or pseudonymous.
- ❑ Using single public key, anyone can monitor the person based on transactions patterns.
- ❑ Every time you want somebody to send you money, you have to give them your address.
- ❑ Somebody can just go through the whole blockchain network and pick up every time your address or public key pops up and they can like start observing things.

- ❑ What is the solution?
- ❑ To improve privacy
 - ❑ Users may create multiple private keys and generate public keys from each one and an address from each one.
 - ❑ One pair of private, public key and address can be used for certain things.
 - ❑ Bitcoin introduced a improvement known as Bitcoin improvement proposal-32.
 - ❑ The hierarchically deterministic wallets were proposed.

Hierarchically Deterministic (HD) Wallets



Hierarchically Deterministic (HD) Wallets

- ❑ A hierarchical deterministic wallet is a digital wallet commonly used to store the keys for holders of cryptocurrencies such as Bitcoin and Ethereum.
- ❑ To prevent hacking these keys must be randomly generated and backed up in the wallet.
- ❑ HD wallets enable a series of key pairs to be created from one random seed (Master Private Key), providing convenience and manageability as well as high-level security.

- ❑ Deterministic wallets were created to offer a solution, one in which all keys can be traced back to an original random seed, usually a set of random words, and a hash function.
- ❑ With a deterministic wallet, the original seed is enough to recover all private and public keys, therefore requiring only a single backup at the time of creation.

Hierarchically Deterministic (HD) Wallets

- ❑ HD Wallets are the most advanced type of deterministic wallet.
- ❑ They contain keys in a tree structure, in which parent keys can produce children keys, which can produce grandchildren keys, and so on, infinitely.
- ❑ The cryptocurrency holder can use the tree structure to organize transactions by type of transaction or by entity involved, such as departments or subsidiaries.

- ❑ Like simple deterministic wallets, all HD wallets are created from a single master root seed, usually represented by a mnemonic word sequence, which makes it easier for account holders to transcribe and store.
- ❑ But HD wallets also offer the option of creating public keys without having to access the corresponding private keys. This means they can be used on insecure servers or in a receive-only mode

Deterministic Wallets, Their Advantages And Their Understated Flaws By Vitalik Buterin

<https://medium.com/geekculture/what-is-bitcoin-improvement-proposal-32-bip-32-586a3f36a95c>

<https://www.gemini.com/cryptopedia/hd-crypto-wallets-hierachichal-deterministic>

<https://www.ledger.com/academy/crypto/what-are-hierarchical-deterministic-hd-wallets>

ETHEREUM

Official Homepage

<https://ethereum.org/en/>

[https://ethereum.org/en/developers
/docs/intro
-to-ethereum/](https://ethereum.org/en/developers/docs/intro-to-ethereum/)

Bitcoin Vs. Ethereum

	BITCOIN	ETHEREUM
Founder	Satoshi Nakamoto(Unknown)	Vitalik Buterin and Team
Purpose	Only Crypto Currency	Decentralized Network Software
Release Date	January 2009	July 2015
Scripting Language	Turing Incomplete	Turing Complete
Coin Release Method	Early Mining	Thru ICO
Average Block Time	Approx. 10 minutes	Approx. 10-12 seconds
Transaction Model	UTXO	Account
Coin Symbol	BTC	ETH
Tokens	Not available	Available
Monetary Policy	Hard Coded	Not Hard Coded
Emission Rate	Halving policy followed	Occasional
Block Limitation	1 MB per block	No limit

Bitcoin Vs. Ethereum

- ❑ Both allows to use digital money without payment providers or banks.
- ❑ Bitcoin is only a peer-to-peer payment network.
- ❑ But Ethereum is programmable, so you can also build and deploy decentralized applications on its network.
- ❑ Ethereum being programmable means that you can build apps that use the Blockchain to store data or control what your app can do.
- ❑ It results in a general purpose Blockchain that can be programmed to do anything.
- ❑ Ethereum is more like a marketplace of financial services, games, social networks and other apps that respect your privacy and cannot censor you.

TERMINOLOGY

- ❑ **Blockchain**

- The sequence of all blocks that have been committed to the Ethereum network in the history of the network.

- ❑ **ETH**

- Ether (ETH) is the native cryptocurrency of Ethereum. Users pay ETH to other users to execute their codes.

- ❑ **EVM**

- The Ethereum Virtual Machine is the global virtual computer whose state every participant on the Ethereum network stores and agrees on. Any participant can request the execution of arbitrary code on the EVM; code execution changes the state of the EVM.

- ❑ **Nodes**

- The real-life machines which are storing the EVM state.

- Nodes communicate with each other to propagate information about the EVM state and new state changes.

- ❑ **Accounts**

- Where ETH is stored.

- Users can create accounts, deposit ETH into the accounts, and transfer ETH from their accounts to other users.

- Accounts and account balances are stored in a big table in the EVM; they are a part of the overall EVM state.

- Two types of accounts.

TERMINOLOGY

- ❑ **Transactions**

- ❑ A "transaction request" is the formal term for a request for code execution on the EVM, and a "transaction" is a fulfilled transaction request and the associated change in the EVM state.
- ❑ Examples of transactions:
 - ❑ Send X ETH from my account to Alice's account.
 - ❑ Publish some smart contract code into EVM state.
 - ❑ Execute the code of the smart contract at address X in the EVM, with arguments Y.

- ❑ **Blocks**

- ❑ The volume of transactions is very high, so transactions are "committed" in batches, or blocks. Blocks generally contain dozens to hundreds of transactions.

- ❑ **Smart contracts**

- ❑ A reusable snippet of code (a program) which a developer publishes into EVM state.
- ❑ Anyone can request that the smart contract code be executed by making a transaction request.
- ❑ Because developers can write arbitrary executable applications into the EVM (games, marketplaces, financial instruments, etc.) by publishing smart contracts, these are often also called dapps, or Decentralized Apps.

WHAT IS ETHEREUM?

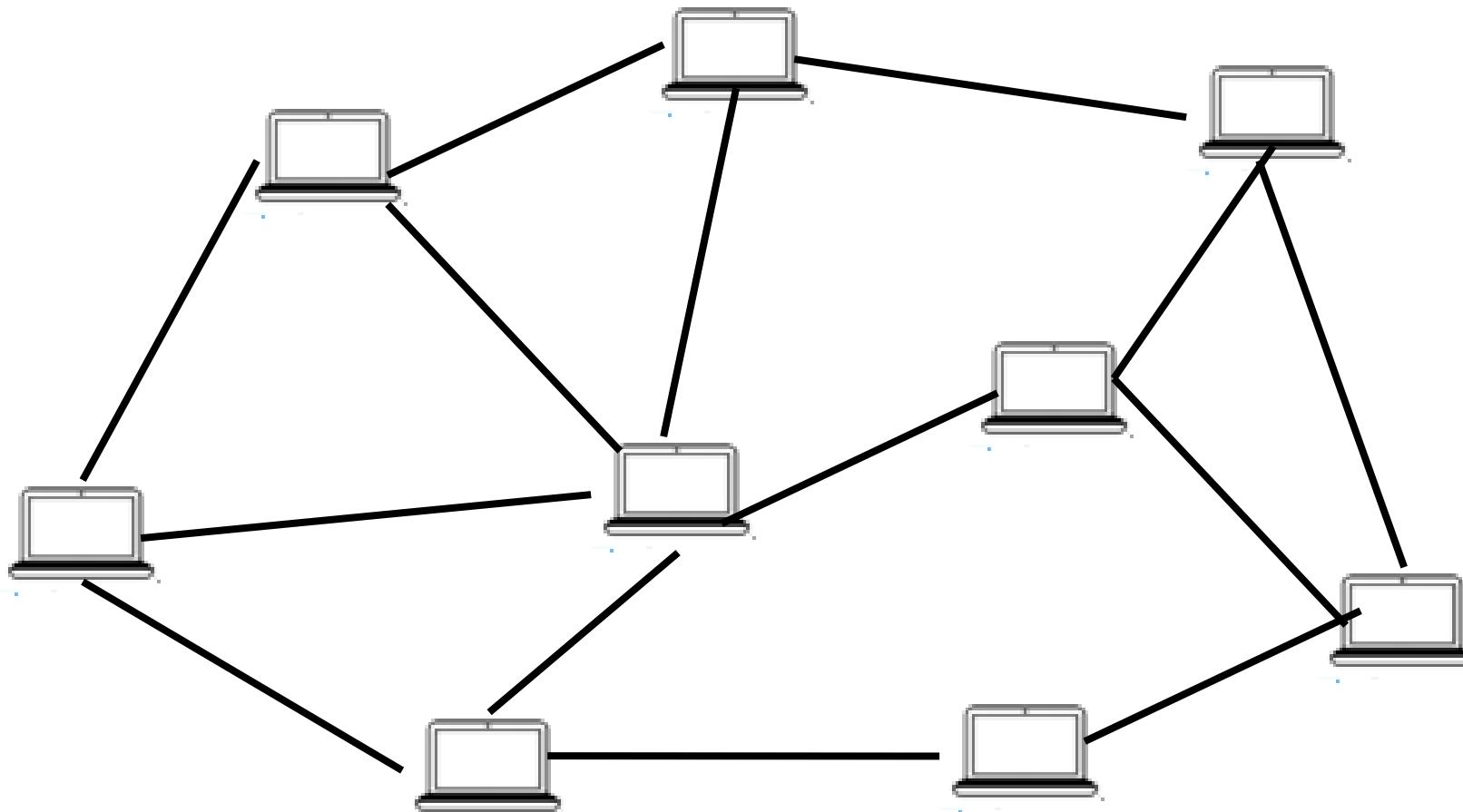
- ❖ Ethereum provides a platform that not only allows us to store transactional data, but actually allows us to store programs that facilitates the execution of programs on decentralize network.
- ❖ It allows to run applications on decentralized network (For exmpale, Alien Worlds DApp).

- ❖ The core idea here is to build a world supercomputer in a distributed manner and use blockchain to facilitate that.
- ❖ Web 3.0 is still evolving and being defined, and as such, there isn't a canonical, universally accepted definition.
- ❖ Web 3.0 will have a strong emphasis on decentralized applications and make extensive use of blockchain-based technologies. Web 3.0 will also make use of machine learning and artificial intelligence (AI) to help empower more intelligent and adaptive applications.

- ❖ Ethereum Definition
- ❖ Ethereum is a Blockchain with a computer embedded in it.
- ❖ It is the foundation for building apps and organizations in a decentralized, permissionless, censorship-resistant way.

WHAT IS ETHEREUM?

- ❑ In the Ethereum universe, there is a single, canonical computer (called the Ethereum Virtual Machine, or EVM) whose state everyone on the Ethereum network agrees on.
- ❑ Everyone who participates in the Ethereum network (every Ethereum node) keeps a copy of the state of this computer.
- ❑ Any participant can broadcast a request for this computer to perform arbitrary computation.
- ❑ Whenever such a request is broadcast, other participants on the network verify, validate, and carry out ("execute") the computation.
- ❑ This execution causes a state change in the EVM, which is committed and propagated throughout the entire network.
- ❑ Requests for computation are called transaction requests; the record of all transactions and the EVM's present state gets stored on the Blockchain, which in turn is stored and agreed upon by all nodes.
- ❑ Cryptographic mechanisms ensure that once transactions are verified as valid and added to the blockchain, they can't be tampered with later.
- ❑ The same mechanisms also ensure that all transactions are signed and executed with appropriate "permissions" (no one should be able to send digital assets from Alice's account, except for Alice herself).



WHAT IS ETHER?

- ❑ Ether (ETH) is the native cryptocurrency of Ethereum.
 - ❑ The purpose of ETH is to allow for a market for computation.
 - ❑ This market provides an economic incentive for participants to verify and execute transaction requests and provide computational resources to the network.
-
- ❑ Any participant who broadcasts a transaction request must also offer some amount of ETH to the network as a bounty.
 - ❑ The network will award this bounty to whoever eventually does the work of verification and broadcasting.
 - ❑ The amount of ETH paid corresponds to the resources required to do the computation.
 - ❑ These bounties also prevent malicious participants from intentionally clogging the network by requesting the execution of infinite computation or other resource-intensive scripts, as these participants must pay for computation resources.
-
- ❑ ETH is also used to provide crypto-economic security to the network in three main ways:
 - ❑ It is used as a means to reward validators who propose blocks or call out dishonest behavior by other validators;
 - ❑ It is staked by validators, acting as collateral against dishonest behavior—if validators attempt to misbehave their ETH can be destroyed;
 - ❑ It is used to weight 'votes' for newly proposed blocks, feeding into the fork-choice part of the consensus mechanism.

SMART CONTRACT

- ❑ Smart Contracts are actually programs which are executing on the blockchain.
 - ❑ A contract is a set of rules that or a set of clauses that parties agree on that govern the relationship between them.
 - ❑ The contracts written in programming language are known as smart contract
- ❑ In practice, participants don't write new code every time they want to request a computation on the EVM.
- ❑ Rather, application developers upload programs (reusable snippets of code) into EVM state, and users make requests to execute these code snippets with varying parameters. We call the programs uploaded to and executed by the network smart contracts.
- ❑ At a very basic level, you can think of a smart contract like a sort of vending machine:
- ❑ Thus, with smart contracts, developers can build and deploy arbitrarily complex user-facing apps and services such as: marketplaces, financial instruments, games, etc.

TURING COMPLETENESS

- ❑ What is Turing completeness of a language ?
 - ❑ It means if a programming language is Turing complete, that means you can code absolutely any logic into that Turing into that language.
 - ❑ How long it will take for that code to run is a different question ?
 - ❑ Few minutes, days, years or more.
-
- ❑ Why is bitcoin script is not Turing complete?
 - ❑ In Bitcoin script, there is one component that is missing a very important component (loops).
 - ❑ It's intentional, the Blockchain is executed on peer-to-peer network.
 - ❑ The loop really slow down the chain.
 - ❑ If somebody intentionally or non intentionally creates a program that has, for instance, an infinite loop, a loop that doesn't have an end, it can destroy the whole blockchain or it can hang the whole blockchain or slow everything down significantly or even in order.
-
- ❑ Solidity, on the other hand, found a way to include loops.
 - ❑ There is story behind it (Vitalik Buterin was actually advocating for a bitcoin script to include loops)

SMART CONTRACT

- ❑ A smart contract or program is copied on every node in the peer-to-peer network. It consists of-
 - ❑ History of all transactions
 - ❑ **History of all smart contracts.**
 - ❑ **Current state of all smart contracts.**



<https://www.freecodecamp.org/news/smart-contracts-for-dummies-a1ba1e0b9575/>

<https://www.provenance.org/whitepaper>

<https://medium.com/@polysvote/smart-contracts-for-dummies-e8f332275c56>

CENTRALIZED v/S DECENTRALIZED APPS

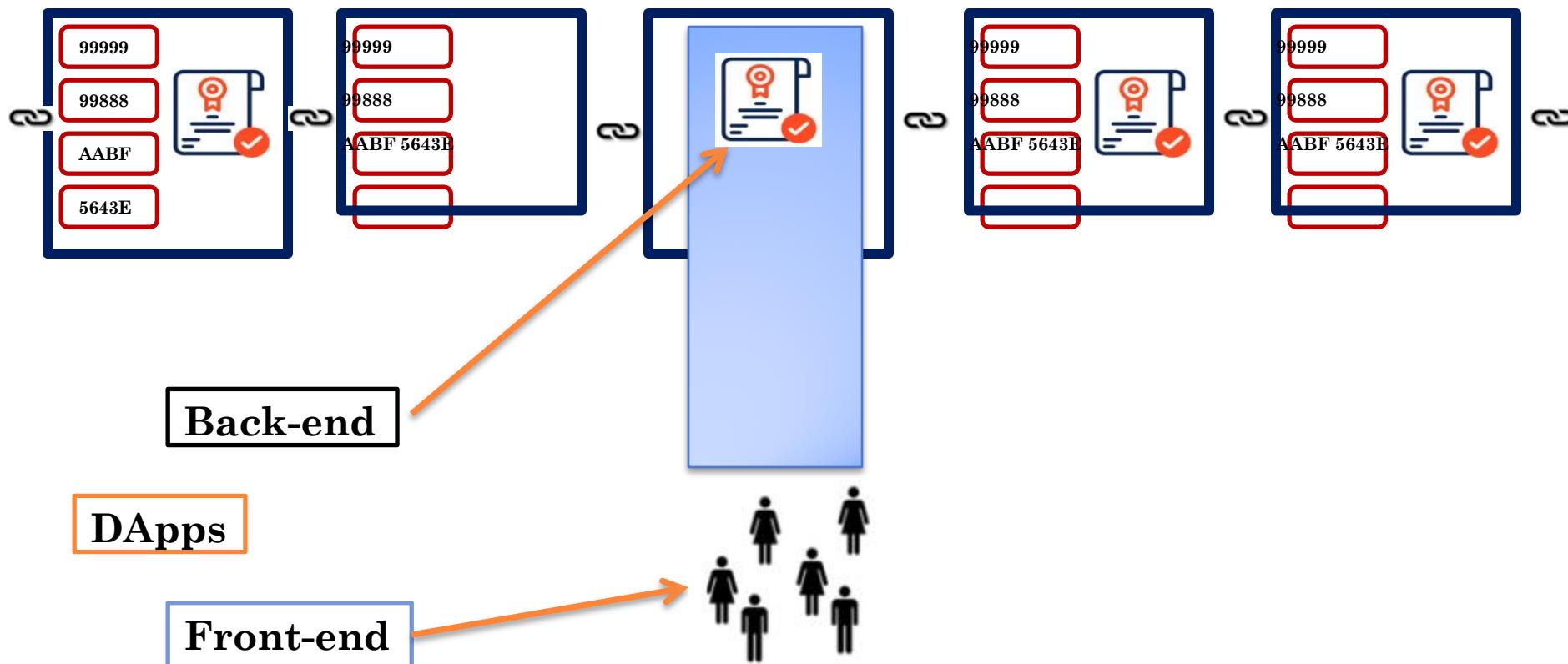
- A centralized app is owned by a single company. The application software for a centralized app resides on one or more servers controlled by the company.
- As a user, you'll interact with the app by downloading a copy of the app and then sending and receiving data back and forth from the company's server.
- Twitter, Facebook, Instagram, and Netflix. Banks and other financial institutions use centralized apps to allow their customers online access to their accounts.
- A decentralized app (also known as a DApp or dapp) operates on a Blockchain or peer-to-peer network of computers.
- It enables users to engage in transactions directly with one another as opposed to relying on a central authority.
- The user of a dApp will pay the developer an amount of cryptocurrency to download and use the program's source code.
- The source code is known as a smart contract, which allows users to complete transactions without revealing personal information.
- Peepeth, a social network alternative to Twitter.
- Cryptokitties is a dApp game that allows users to buy and sell virtual cats.

DECENTRALIZED

APPLICATIONS

- Decentralized applications - also known as “DApps” --- are digital applications that run on a Blockchain network of computers instead of relying on a single computer (Centralized).
- DApps are decentralized, they are free from the control and interference of a single authority.
- Some features are-
- **Decentralized** - DApps operate on Ethereum, an open public decentralized platform where no one person or group has control
- **Deterministic** - DApps perform the same function irrespective of the environment in which they get executed
- **Turing complete** - DApps can perform any action given the required resources
- **Isolated** - DApps are executed in a virtual environment known as Ethereum Virtual Machine so that if the smart contract has a bug, it won’t hamper the normal functioning of the Blockchain network

DECENTRALIZED APPLICATIONS



DECENTRALIZED APPLICATIONS

- ❑ A standard web app, such as Uber or Twitter, runs on a computer system that is owned and operated by an organization, giving it full authority over the app and its workings.
 - ❑ There may be multiple users on one side, but the backend is controlled by a single organization.
-
- ❑ DApps can run on a P2P network or a blockchain network.
 - ❑ Steemit, BitTorrent, Tor, and Popcorn Time.
 - ❑ Multiple participants are consuming content, feeding or seeding content, or simultaneously performing both functions.
-
- ❑ In the context of cryptocurrencies, DApps run on a Blockchain network in a public, open-source, decentralized environment and are free from control and interference by any single authority.
 - ❑ For example,
 - ❑ A developer can create a Twitter-like DApp and put it on a Blockchain where any user can publish messages.
 - ❑ Once posted, no one—including the app creators—can delete the messages.

DECENTRALIZED APPLICATIONS : BENEFITS

- **Zero Downtime**

- Once the smart contract is deployed on the Blockchain, the network as a whole will always be able to serve clients looking to interact with the contract. Malicious actors, therefore, cannot launch denial-of-service attacks targeted towards individual DApps.

- **Privacy**

- You don't need to provide real-world identity to deploy or interact with a DApp.

- **Resistance to censorship**

- No single entity on the network can block users from submitting transactions, deploying DApps, or reading data from the Blockchain.

- **Complete data integrity**

- Data stored on the Blockchain is immutable and indisputable, due to cryptographic primitives. Malicious actors cannot forge transactions or other data that has already been made public.

- **Trustless computation/verifiable behavior**

- Smart contracts can be analyzed and are guaranteed to execute in predictable ways, without the need to trust a central authority. This is not true in traditional models; for example, when we use online banking systems, we must trust that financial institutions will not misuse our financial data, tamper with records, or get hacked.

DECENTRALIZED APPLICATIONS : DRAWBACKS

- **Maintenance**

- DApps can be harder to maintain because the code and data published (deployed) to the Blockchain are harder to modify (even if bugs or security risks are identified in an old version).

- **Performance overhead**

- There is a huge performance overhead, and scaling is really hard. To achieve the level of security, integrity, transparency, and reliability that Ethereum aspires to, **every node runs and stores every transaction. On top of this, proof-of-stake consensus takes time as well.**

- **Network congestion**

- When one DApp uses too many computational resources, the entire network gets backed up. Currently, the network can only process about 10-15 transactions per second; **if transactions are being sent in faster than this, the pool of unconfirmed transactions can quickly balloon.**

- **User experience**

- It may be harder to engineer user-friendly experiences because the average end-user might find it too difficult to set up a tool stack necessary to interact with the Blockchain in a truly secure fashion.

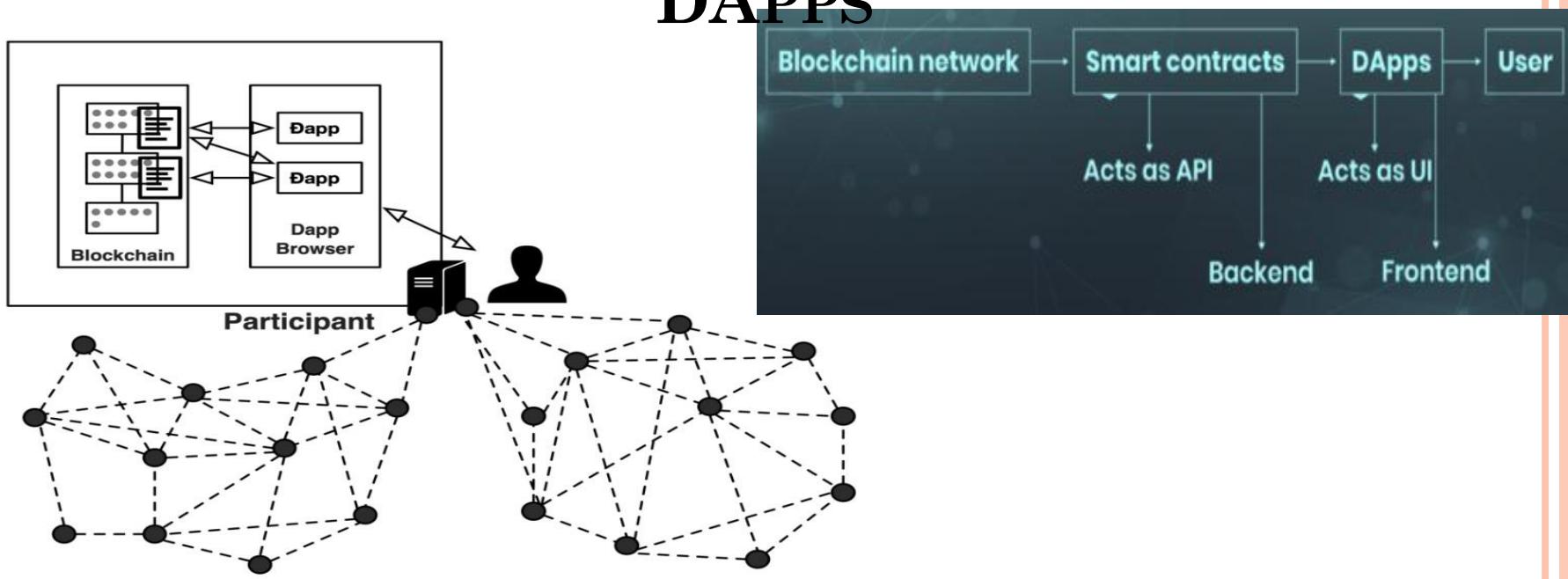
DECENTRALIZED APPLICATIONS : DRAWBACKS

- *Centralization*
 - User-friendly and developer-friendly solutions built on top of the base layer of Ethereum might end up looking like centralized services anyways.
 - For example, such services may store keys or other sensitive information server-side, serve a frontend using a centralized server, or run important business logic on a centralized server before writing to the blockchain.
 - Centralization eliminates many (if not all) of the advantages of blockchain over the traditional model.

SMART CONTRACTS VS DAPPS

- ❑ Decentralized Applications (DApps) are blockchain-based applications that allow users to interact with smart contracts deployed on the blockchain.
- ❑ Users interact with smart contracts by sending transactions to invoke their functions.
- ❑ Users could understand how to interact with a smart contract by analyzing its on-chain or off-chain code (smart contracts are typically shared among all stakeholders as they need to agree on its contractual semantics).
- ❑ The Application Binary Interface (ABI) of the smart contract is also publicly accessible so that users can send transactions to it.
- ❑ **Problem**
 - ❑ Users need a strong technical understanding of blockchain and smart contracts to be able to generate transactions calling smart contracts.
 - ❑ Such a process is error-prone and results in a bad user experience.
 - ❑ How to call a smart contract in a trustless environment?
- ❑ **Solution**
 - ❑ Develop a front-end interface for users to easily interact with smart contracts.
 - ❑ The front end can use the same technology as conventional web or mobile applications to render the interface.
 - ❑ As this interface is a gateway to an application running on a decentralized system (i.e., blockchain), it is referred to as a Decentralized Application (DApp).
 - ❑ The transactions calling smart contracts are generated by DApps and presented to the users for further verification before being sent to the blockchain.
 - ❑ Compared to a conventional application, a DApp is hosted on decentralized storage services like [IPFS](#).
 - ❑ Further, they may be rendered by DApp browsers like [Ethereum Mist](#) or could be implemented as a plug-in to a regular web browser like [MetaMask](#).

SMART CONTRACTS VS DAPPS



- DApps are applications running on a blockchain network, smart contracts are the source of power for these DApps. Smart contracts act as an interface between DApps and blockchain networks.
- Smart contracts are simply a piece of code that acts as a backend mechanism. Whereas, DApp are like user interfaces that interact directly with the user.
- DApps run on smart contracts and connect members with providers directly. Basically, smart contracts are the entities that make a DApp work based on predetermined rules.

ETHEREUM VIRTUAL MACHINE AND GAS.

- ❑ Bitcoin script never included loops because somebody might maliciously or by accident include an infinite loop or a very heavy computation that would really slow down the computations on all of these nodes and slow down the network.

- ❑ Ethereum included loops. How the problem is solved ?
- ❑ Smart contracts run on every single node, on every single machine in the network.
- ❑ How convenient it is for a virus?
- ❑ If somebody writes a virus (smart contract) and sends it out on a Blockchain, it gets copied to all the computers. Everybody's machine will be infected
- ❑ What if smart contracts (programs/virus) gain access to private files?

- ❑ Two main security threats
- ❑ Viruses and access to private files → Ethereum Virtual Machine
- ❑ Infinite loops (heavy computations) → Ethereum Gas

ETHEREUM VIRTUAL MACHINE AND GAS.

- ❑ Viruses and Access to Private Files
- ❑ When you participate in the Ethereum network, you get an Ethereum virtual machine.
- ❑ It's a virtual machine that is running on your computer and therefore it completely encapsulates everything that runs there.
- ❑ The smart contract runs on EVM which runs inside your computer, and nothing can get out of that virtual machine.
- ❑ Virtual machine doesn't know that anything outside of that virtual machine exists.
- ❑ It is guaranteed that anything that happens on the virtual machine stays on the virtual machine.
- ❑ It would never actually spread on to your the rest of your hard drive and all the everything else that you have on your machine.
- ❑ It means programs (smart contract) will never have access to your files to your private files, to other things that you have on your computer.

<https://ethereum.org/en/developers/docs/evm/>

ETHEREUM VIRTUAL MACHINE AND GAS.

- ❑ Infinite loops (heavy computations) → Ethereum Gas
- ❑ For any computation that's run on the Ethereum Blockchain, the developers of the smart contract need to pay well.
- ❑ The code on solidity will be converted to computer code (low level code) and then these prices are applied.
- ❑ It can solve the problem of computationally heavy program (An infinite loop), you will very quickly run out of gas and it'll stop working.
- ❑ Three Benefits
 - ❑ The developers will be penalized for writing in-efficient code (Infinite loops/heavy computations).
 - ❑ Blockchain network will be safe as it will very quickly stop.
 - ❑ It will encourage developers to write good code.

ETHEREUM VIRTUAL MACHINE AND GAS.

- ❑ It is one difference between ether and Bitcoin.
- ❑ Bitcoin
 - ❑ Bitcoin is to create a cryptocurrency, something that people can transact with and exchange value and pay for services and something that's borderless and transparent and stateless and have no restrictions.
- ❑ ether
- ❑ Ether
 - ❑ Ethereum can be utilized to transact/investment as Bitcoin.
 - ❑ Ethereum can be used to create and run code (Dapps) on the Blockchain and pay for it.
 - ❑ The payment for those applications is absolutely necessary because otherwise people are going to abuse the blockchain.
 - ❑ In order to have Turing completeness, you have to have a cost for these operations.

ETHEREUM VIRTUAL MACHINE AND GAS.

- ❑ It is one difference between ether and Bitcoin.
- ❑ Bitcoin
 - ❑ Bitcoin is to create a cryptocurrency, something that people can transact with and exchange value and pay for services and something that's borderless and transparent and stateless and have no restrictions.
- ❑ ether
- ❑ Ether
 - ❑ Ethereum can be utilized to transact/investment as Bitcoin.
 - ❑ Ethereum can be used to create and run code (Dapps) on the Blockchain and pay for it.
 - ❑ The payment for those applications is absolutely necessary because otherwise people are going to abuse the blockchain.
 - ❑ In order to have Turing completeness, you have to have a cost for these operations.

GAS & GAS

PRICE

- ❑ Any operation that involves writing or modifying data on Ethereum requires gas.
- ❑ Operations include transferring ether (or any other ERC20 token), minting and transferring NFTs, deploying smart contracts, etc.

- ❑ Let's compute the cost of deploying a smart contract first. Traditionally, creating smart contracts has always been an expensive process.
- ❑ Go to the Etherscan page of your deployed contract on Rinkeby and check the very first transaction (which will be marked as Create: NFTCollectible)

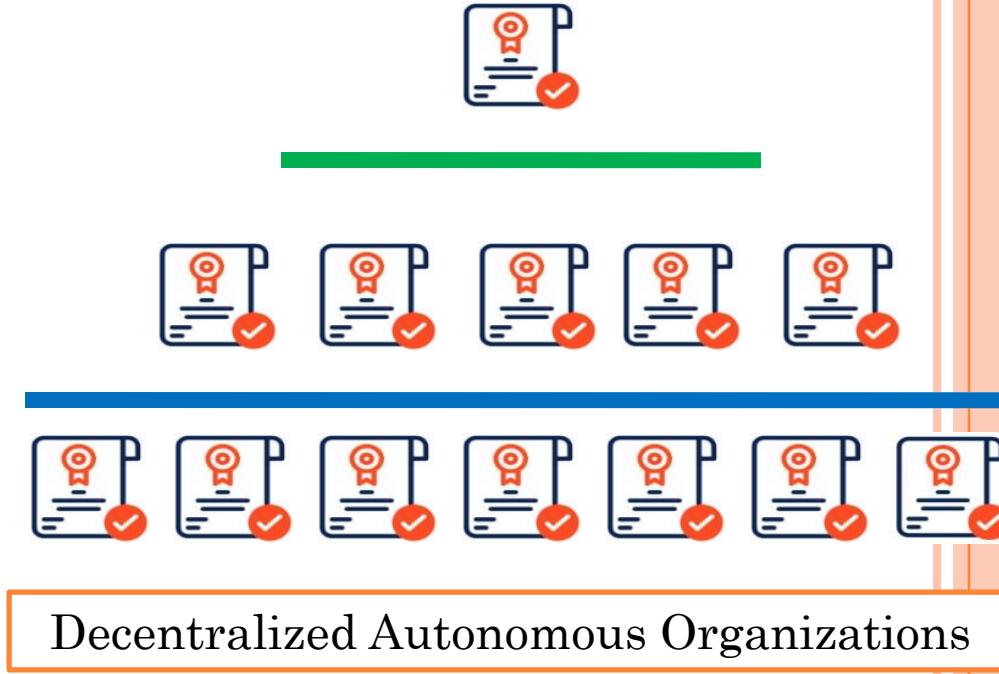
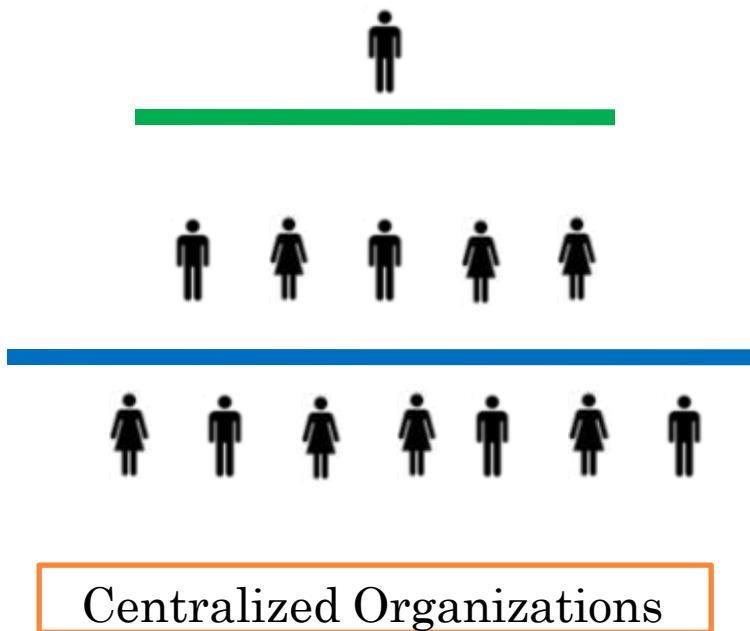
<https://medium.com/scrappy-squirrels/estimating-smart-contract-costs-f65acf818c26>

<https://etherscan.io/gastracker>

<https://rinkeby.etherscan.io/>

DECENTRALIZED AUTONOMOUS ORGANIZATIONS (DAOs)

- ❑ Member-owned communities without centralized leadership.
- ❑ A safe way to collaborate with internet strangers.
- ❑ A safe place to commit funds to a specific cause.



DAOs (WHAT)

- ❑ A DAO is a collectively-owned, Blockchain-governed organization working towards a shared mission.
- ❑ DAOs allow us to work with like-minded folks around the globe without trusting a benevolent leader to manage the funds or operations.
- ❑ There is no CEO who can spend funds on a whim or CFO who can manipulate the books.
- ❑ Instead, Blockchain-based rules baked into the code define how the organization works and how funds are spent.
- ❑ They have built-in treasuries that no one has the authority to access without the approval of the group.
- ❑ Decisions are governed by proposals and voting to ensure everyone in the organization has a voice, and everything happens transparently on-chain.

DAOs (WHY)

- ❑ Starting an organization with someone that involves funding and money requires a lot of trust in the people you're working with.
- ❑ But it's hard to trust someone you've only ever interacted with on the internet.
- ❑ With DAOs you don't need to trust anyone else in the group, just the DAO's code, which is 100% transparent and verifiable by anyone.
- ❑ This opens up so many new opportunities for global collaboration and coordination.

DAOs v/s CENTRALIZED ORGANIZATIONS

Decentralized Autonomous Organizations	Centralized Organizations
Usually flat, and fully democratized.	Usually hierarchical.
Voting required by members for any changes to be implemented.	Depending on structure, changes can be demanded from a sole party, or voting may be offered.
Votes tallied, and outcome implemented automatically without trusted intermediary.	If voting allowed, votes are tallied internally, and outcome of voting must be handled manually.
Services offered are handled automatically in a decentralized manner (for example distribution of philanthropic funds).	Requires human handling, or centrally controlled automation, prone to manipulation.
All activity is transparent and fully public.	Activity is typically private, and limited to the public.

How DAOs WORK ?

- ❑ The backbone of a DAO is its smart contract, which defines the rules of the organization and holds the group's treasury.
- ❑ Once the contract is live on the Blockchain, no one can change the rules except by a vote.
- ❑ If anyone tries to do something that's not covered by the rules and logic in the code, it will fail.
- ❑ Treasury is defined by the smart contract too that means no one can spend the money without the group's approval either.
- ❑ It means group makes decisions collectively, and payments are automatically authorized when votes pass.

- ❑ Ethereum is the perfect foundation for DAOs for a number of reasons:
- ❑ Consensus mechanism is distributed and established enough for organizations to trust the network.
- ❑ Smart contract code can't be modified once live, even by its owners. This allows the DAO to run by the rules it was programmed with.
- ❑ Smart contracts can send/receive funds.
- ❑ The Ethereum community has proven to be more collaborative than competitive.

How DAOs WORK ?

- There are many considerations when governing a DAO, such as how voting and proposals work.
- Voting rights are decided based on memberships-
 - Token-based Membership
 - Share-based Membership
 - Reputation -based Membership

For more information on Decentralized autonomous organizations (DAOs)
<https://ethereum.org/en/dao/>

DAO

ATTACK

- DAOs would help with the development of decentralized applications to run on the Ethereum Blockchain.
- DAO organization was stateless, i.e., didn't belong to any country/ individual/ organization
- DAOs was crowdfunded through a token sale by May 2016 and raised \$150 million.
- That was the most successful crowdfunding campaign in the history of the world \$150 Million.
-
- However, unfortunately, there was an error in the code of the DAO, i.e., there was an error in the smart contracts that were coded for this autonomous organization.
- DAO was hacked on June 2016 for \$50 Million. The attacker didn't actually do anything illegal.
- They just exploit the flaw in the code and they used it to leach money out of the DAO account into their own account.
- Nobody could do anything because the DAO is autonomous, i.e., governed by its own algorithms, its own code.
- The contract cannot changed because it's on the Blockchain.

DAO

ATTACK

- ❑ However, the good news was that the funds were again, once again, according to the way the deal was coded, there was a failsafe mechanism which meant that funds cannot be taken out entirely, like even though the attacker moved them to another account that was a child company of the Dao, and then they could move them out later on to their own account after 30 days.
- ❑ It means the whole community had opportunity to think and decide what to do.
- ❑ There were groups-
 - ❑ One to support the integrity of Blockchain (i.e., to support the hacker)
 - ❑ Another to support the money of people must be returned (i.e., the rule of the contract will basically be revert so that we can pull the money back and give it back to the owners).
- ❑ Thus, the hard fork was decided and initiated by Vitalik.
 - ❑ The hard fork split Ethereum into Ethereum and Ethereum classic.
 - ❑ In the Ethereum classic, the hacker's money stayed in Ethereum Classic.
 - ❑ It's like it's like two parallel worlds after a hard fork.

DAO

ATTACK

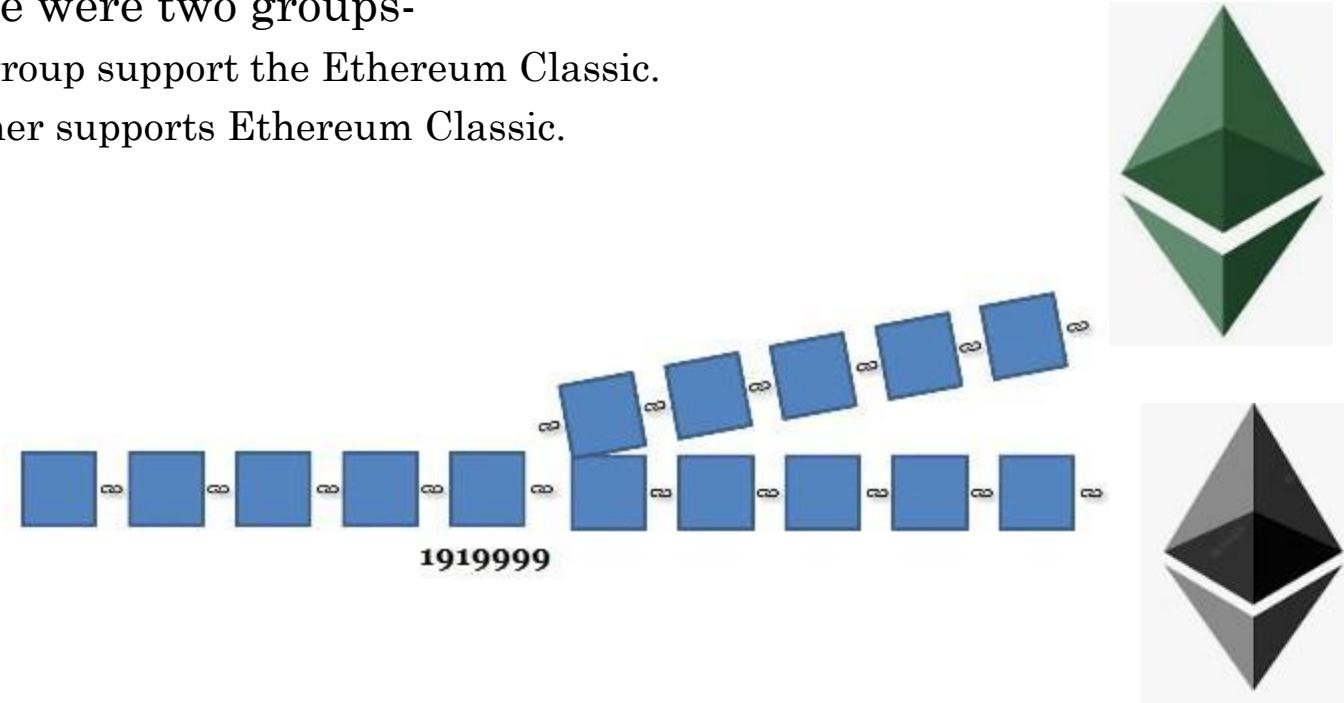
- The important thing to note here, that the problem was in the DAO Code, not in Ethereum itself, it was a problem in how the Smart contract was coded, not the Ethereum platform.

2016
On Ethereum
Investor-directed venture capital fund
Stateless
May 2016 Crowdfunded ~\$150,000,000
June 2016 Hacked for ~\$50,000,000
Dilemma: "Code Is Law?"
Hard fork
Ethereum split into ETH and ETC
Hacker walked away with ~\$67,000,000 in ETC
Problem in DAO code not Ethereum

<https://www.coindesk.com/learn/2016/06/25/understanding-the-dao-attack/>
<https://www.bloomberg.com/features/2017-the-ether-thief/>

HARD FORK-EXAMPLE

- In July 2016, Ethereum at block number 1919999 had fork
 - They introduced new rules that would allow reversing the logic in the contract that allowed the attacker to steal the ether from the Dao.
 - After that, Ethereum chain continued and the reverse the funds and returned them back to the people that contributed them and closed down the doubt.
 - There were two groups-
 - One group support the Ethereum Classic.
 - Another supports Ethereum Classic.



SOFT & HARD FORKS -

EXAMPLE

- On 20th July 2017, Bitcoin accepted segregated witness that happened on block number 476,768.
- Bitcoin accepted the fork for segregated witness. It was soft fork.
- A group of people favors the block size to eight megabytes and a hard fork was proposed.
- On 1st of August, 2017, Hard fork occurs at block no. 478558.
- The Bitcoin split into two Bitcoin and Bitcoin Cash



INITIAL COIN OFFERINGS (ICO)

Technology

Blockchain

Protocol/Coin

Bitcoin

Ethereum

Neo

Ripple

Tokens

X

TRX SNT
REP AE
BNB REP
PPT AHOC

ACA RPX
ONT IAM
QLC TNC
DBC TKY

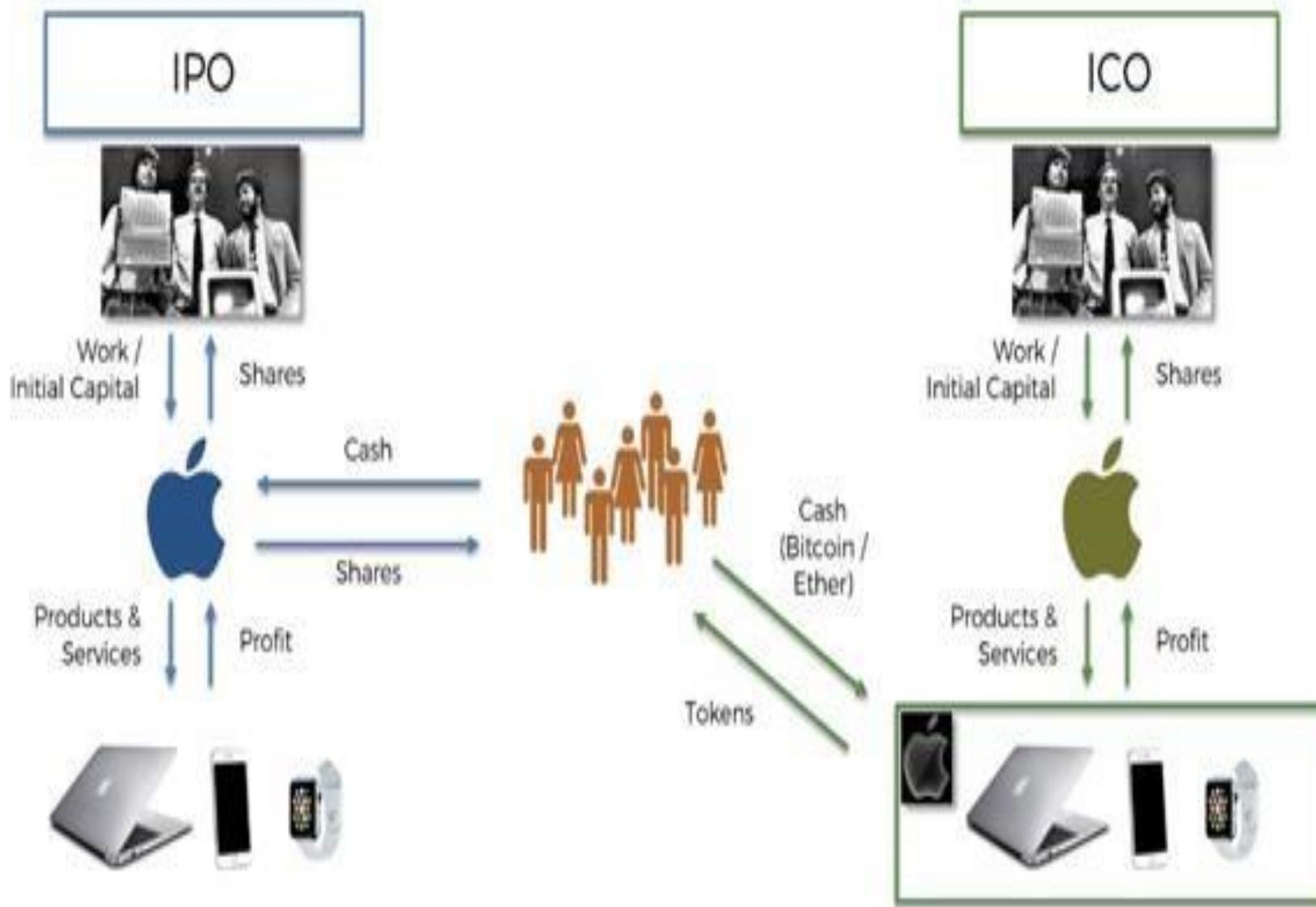
X



INITIAL COIN OFFERINGS (ICOs): GENERAL IDEA



INITIAL COIN OFFERINGS (ICOs): GENERAL IDEA



INITIAL COIN OFFERINGS (ICOs): WHAT

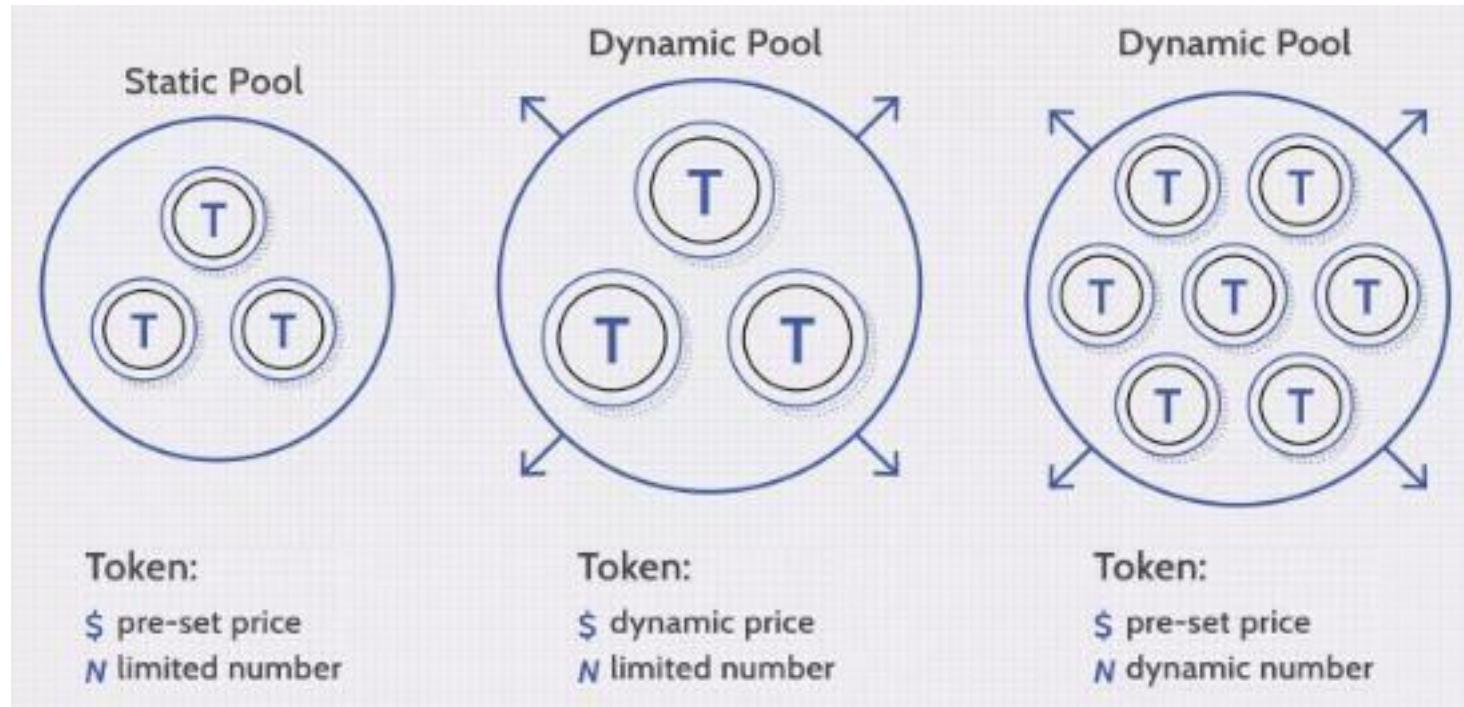
- ❑ An initial coin offering (ICO) is a type of capital-raising activity in the cryptocurrency and blockchain environment.
- ❑ A company seeking to raise money to create a new coin, app, or service can launch an ICO as a way to raise funds.
- ❑ The ICO can be viewed as an initial public offering (IPO) that uses cryptocurrencies.
- ❑ To participate in an ICO, you usually need to first purchase a more established digital currency, plus have a basic understanding of cryptocurrency wallets and exchanges.
- ❑ Interested investors can buy into an initial coin offering to receive a new cryptocurrency token issued by the company.
- ❑ This token may have some utility related to the product or service that the company is offering or represent a stake in the company or project.

- ❑ A few ICOs have yielded returns for investors. Numerous others have turned out to be fraudulent or have performed poorly.

INITIAL COIN OFFERINGS (ICOs): HOW

- ❑ When a cryptocurrency project wants to raise money through an ICO, the project organizers' first step is determining how they will structure the coin. ICOs can be structured in a few different ways, including:
 - ❑ Static supply and static price
 - ❑ A company can set a specific funding goal or limit, which means that each token sold in the ICO has a preset price, and the total token supply is fixed.
 - ❑ Static supply and dynamic price
 - ❑ An ICO can have a static supply of tokens and a dynamic funding goal— this means that the amount of funds received in the ICO determines the overall price per token.
 - ❑ Dynamic supply and static price
 - ❑ Some ICOs have a dynamic token supply but a static price, meaning that the amount of funding received determines the supply.

INITIAL COIN OFFERINGS (ICOs): How



INITIAL COIN OFFERINGS (ICOs): How

- White Paper Release
- Alongside structuring the ICO, the crypto project usually releases a white paper in the crypto industry.
- It makes available to potential investors via a new website dedicated to the token.
- The promoters of the project use their white paper to explain important information related to the ICO:
 - What the project is about
 - The need that the project would fulfill upon completion
 - How much money the project needs
 - How many of the virtual tokens the founders will keep
 - What type of payment (which currencies) will be accepted
 - How long the ICO campaign will run
- The project releases the white paper as part of its ICO campaign, which it designs to encourage enthusiasts and supporters to buy some of the project's tokens.

INITIAL COIN OFFERINGS (ICOs): How

- What Happens to the Funds?
 - If the money raised in an ICO is less than the minimum amount required by the ICO's criteria, the funds may be returned to the project's investors. The ICO would then be deemed unsuccessful.
 - If the funding requirements are met within the specified period, the money raised is spent in pursuit of the project's goals.
- Who Can Launch an ICO?
 - Anyone can launch an ICO.
 - With very little regulation of ICOs in the U.S. currently, anyone who can access the proper technology is free to launch a new cryptocurrency.
 - But this lack of regulation also means that someone might do whatever it takes to make you believe they have a legitimate ICO and abscond with the money. Of all the possible funding avenues, an ICO is probably one of the easiest to set up as a scam.

INITIAL COIN OFFERINGS (ICOs): EXAMPLE

- Ethereum's ICO in 2014 is an early, prominent example of an initial coin offering. The Ethereum ICO raised \$18 million over a period of 42 days.
- In 2015, a two-phase ICO began for a company called Antshares, which later rebranded as Neo. The first phase of this ICO ended in October 2015, and the second continued until September 2016. During this time, Neo generated about \$4.5 million.
- In another example, during a one-month ICO ending in March 2018, Dragon Coin raised about \$320 million.
- Also in 2018, the company behind the EOS platform shattered Dragon Coin's record by raising a whopping \$4 billion during a yearlong ICO.
- The first instance of the SEC cracking down on an ICO occurred on Dec. 11, 2017, when the agency halted an ICO by Munchee, a California company with a food review app. Munchee was attempting to raise money to create a cryptocurrency that would work within the app to order food. The SEC issued a cease-and-desist letter, treating the ICO as an unregistered securities offering.
- In 2017, more than \$7 billion was raised using ICOs.
- In 2018, the figure almost doubled.

INITIAL COIN OFFERINGS (ICOs): TYPES

- Private ICO

- In private initial coin offerings, only a limited number of investors can participate in the process.
- Only accredited investors (financial institutions and high net-worth individuals) can participate in private ICOs, and a company can choose to set a minimum investment amount.

- Public ICOs

- Public initial coin offerings are a form of crowdfunding that targets the general public.
- The public offering is a democratized form of investing because almost anyone can become an investor.
- However, due to regulatory concerns, private ICOs are becoming a more viable option relative to public offerings.
- The largest ICO to date was executed by Telegram, an instant messaging services provider.
- During a private ICO, the UK-registered company raised over \$1.7 billion.

ETHEREUM ACCOUNTS

- An Ethereum account is similar to a bank account, but for ethers or ETH, where Ethereum can be held, transferred to other accounts, and can also be used to execute smart contracts.
- An Ethereum account is an entity that is composed of an Ethereum address along with a private key. The first 20 bytes of the SHA3 hashed public key is the Ethereum address.

- Types-
- Externally Owned Account
- Contract-Based Account

<https://www.geeksforgeeks.org/what-are-ethereum-accounts/>

ETHEREUM ACCOUNTS

- Externally Owned Account
 - This is the most basic type of Ethereum account, it functions similarly to a Bitcoin account.
 - A private key controls the Ethereum address for EOAs. A person can open as many EOAs as they require.
 - It is created whenever a wallet is created, and it is made with a private key that is required to access EOAs, check balances, send and receive transactions, and establish smart contracts.
- Transactions from an external account to a contract account can trigger code that can execute many different actions, such as transferring tokens or even creating a new contract.
- Externally Owned Accounts cannot list incoming transactions.

ETHEREUM ACCOUNTS

- Contract-Based Accounts
 - Contract-based accounts can perform all of the functions of an externally owned account, but unlike EOAs, they are formed when a contract code is deployed, are governed by contract codes, and are accessed using a unique address.
 - When one party accepts a contract, a unique account is formed which contains all of the charges associated with that contract.
 - Each contract is granted a distinct serial number, which is referred to as a contract account.
 - A contract account can list incoming transactions.
 - Creating contract accounts costs gas because they use the valuable computational and storage resource of the network.
 - Contract accounts can't initiate new transactions on their own. Instead, contract accounts can only fire transactions in response to other transactions they have received either from an externally owned account or from another contract account.

- <https://docs.soliditylang.org/en/v0.8.17/>
- <https://remix-project.org/>