# Understanding Basics of Blockchain

Patika dev

BNB CHAIN

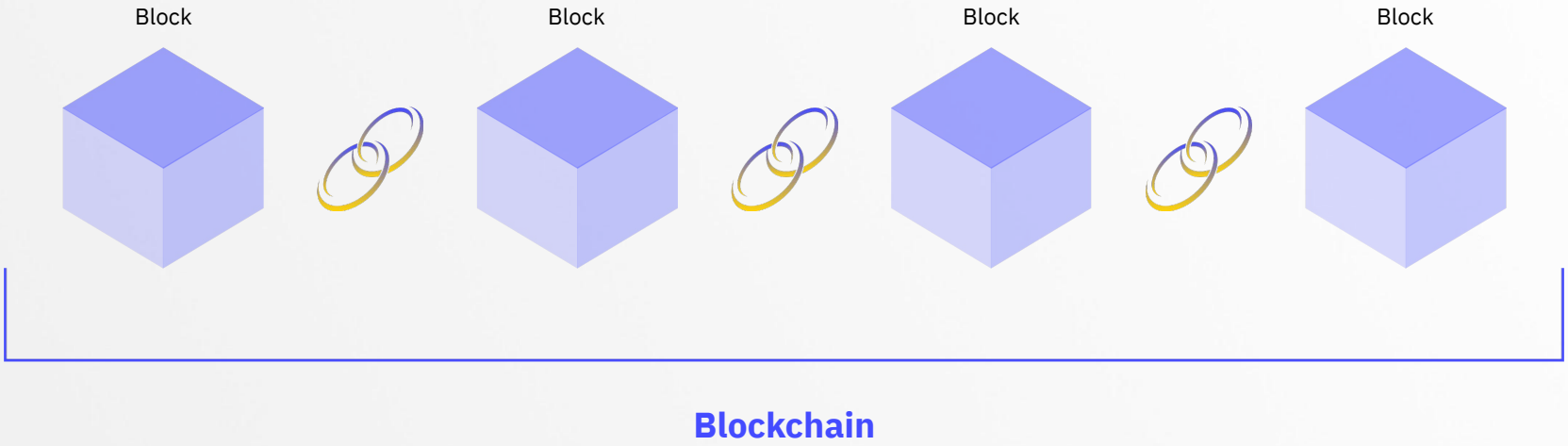
# Content



Patika dev BNB CHAIN

# What is a Block and Its Structure?

In general terms, a blockchain is a giant ledger of transactions, records, and data.
Blocks are the building blocks of blockchains. Transactions are recorded in blocks.



# What is a Block and Its Structure? Block Elements

1. **Block Size**
   The amount of data the block can hold
2. **Block Header**
   Holds the metadata of the block
3. **Transaction Counter**
   Represents the number of transactions are stored in the block
4. **Transactions**
   A list of transactions within the block

- **Version**
  The version number of the block
- **Previous block hash**
  The hash value of the block header of the previous block
- **Timestamp**
  The time at which the block was created
- **Difficulty**
  Difficulty level of finding a hash value
- **Nonce**
  The value that validators try to find to reach a targeted hash
- **Merkle root**
  Value obtained after placing all transactions into a merkle tree

Patika dev BNB CHAIN

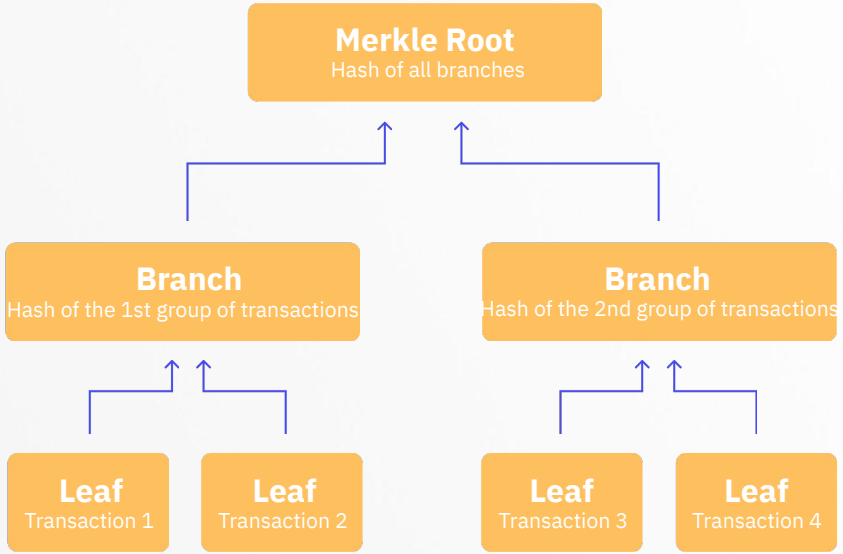
# What is a Block and Its Structure? Merkle Root

The merkle root is a unique representation of all of the transactions in a block.
It is a hash value that is calculated from the transactions.

These hashes are then grouped into pairs and hashed again, and the process is repeated until there is only the **'Root'** left.

A hash is calculated for each pair of transactions using a hash function, creating a **'Branch'**.

The transactions (**'Leaf'**) are first grouped into pairs.




Patika dev BNB CHAIN

# What is a Block and Its Structure? Block Hash

Hash values are generated using hashing algorithms. SHA256 is one of these.
The SHA256 takes various properties of a block as input and produces a block hash as output.

**Version**
The version number of the block

**Previous block hash**
The hash value of the block header of the previous block

**Timestamp**
The time at which the block was created

**Difficulty**
Difficulty level of finding a hash value

**Nonce**
The value that validators try to find to reach a targeted hash

**Merkle root**
Value obtained after placing all transactions into a merkle tree

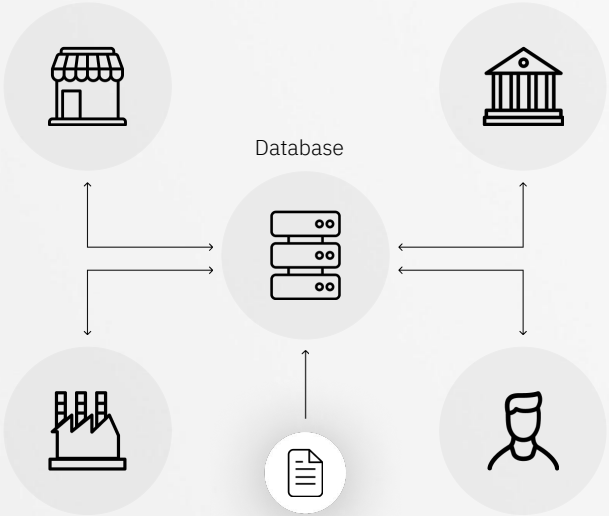

Patika dev
BNB CHAIN

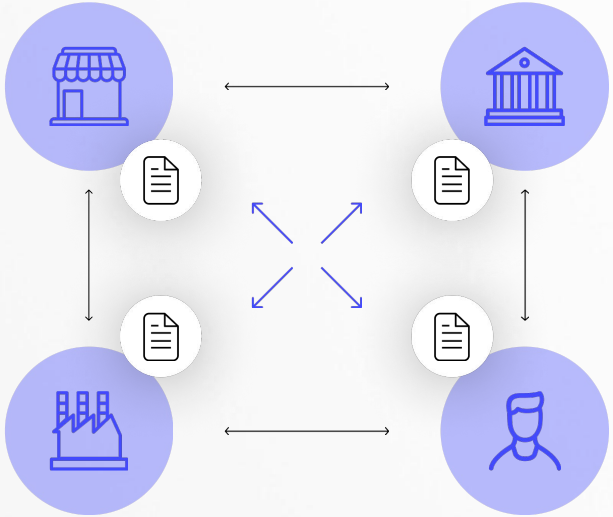
# How are Blocks Chained Together?

A blockchain is made up of a series of blocks that are linked together.
The link between the blocks is created using the hashing algorithm: The hash of previous block is used to calculate the hash of the next block.

# Distributed Ledger & Blockchain Transparency

A distributed ledger is a database that is stored, shared, and synchronized across a network of computers. Blockchain technology allows multiple parties to record transactions and share the records with each other transparently and securely.



Patika dev

BNB CHAIN

# Database vs Blockchain

| Database | Blockchain |
| --- | --- |
| Centralized | Decentralized |
| Fast | Slow |
| Mutable | Immutable |

Patika.dev BNB CHAIN

# Pros and Cons of Blockchain

## Pros

- Security
- Transparency
- Accessibility
- Decentralization

## Cons

- Performance
- Complexity
- Control
- Speed of transactions

Patika dev BNB CHAIN

# Types of Blockchains Permissionless & Permissioned

## Permissionless

A distributed ledger that allows anyone to participate as a validator or a user.

Anyone can read, write, and validate transactions without the need for prior approval.

Not controlled by a single entity.

Privacy focused.

Examples: BNB Chain, Ethereum, Avalanche

## Permissioned

A distributed ledger that restricts access to certain parties.

Only authorized parties are allowed to validate transactions or access the data on the network.

Often has a governing body.

Controlled privacy.

Examples: Corda, Hyperledger

Patika dev BNB CHAIN

# Blockchain Key Terms

## Cryptography in Blockchain

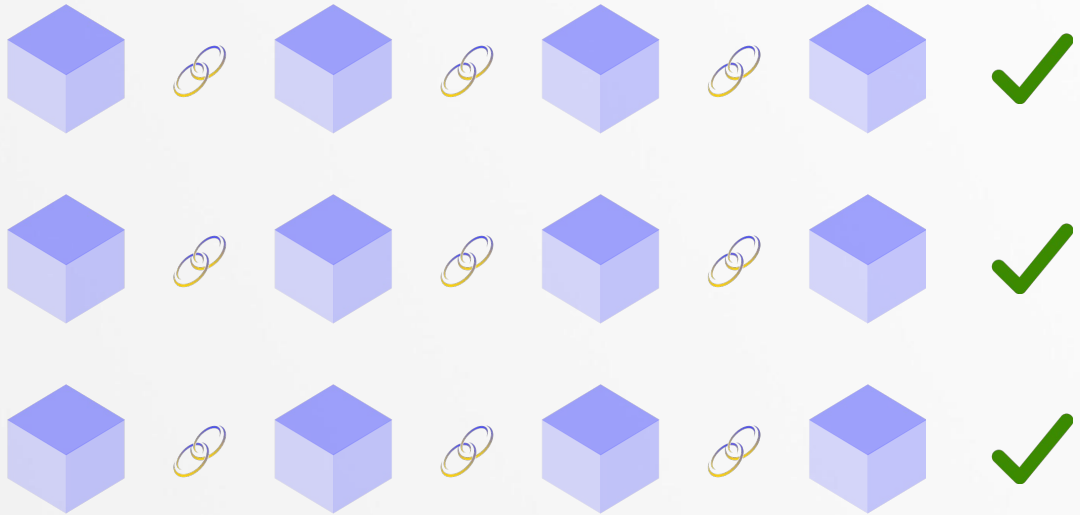
Patika dev

BNB CHAIN

# Content

# What is Meant by the Immutability of Data?

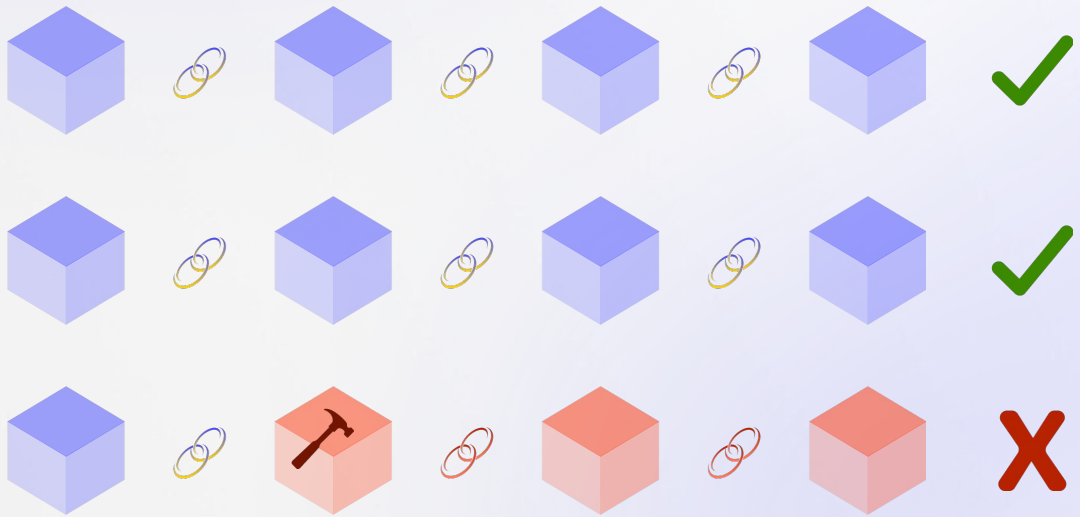
Immutability in a blockchain refers to the fact that once data has been recorded, it cannot be changed or modified.
This feature is accomplished through the use of cryptographic techniques. These allow the integrity of data to be verified and maintained.
Immutability makes a blockchain a suitable platform for storing and sharing important information, and helps to ensure that the data can be trusted as it cannot be altered.



Patika dev BNB CHAIN

# What if Data is Changed on One of the Computers?

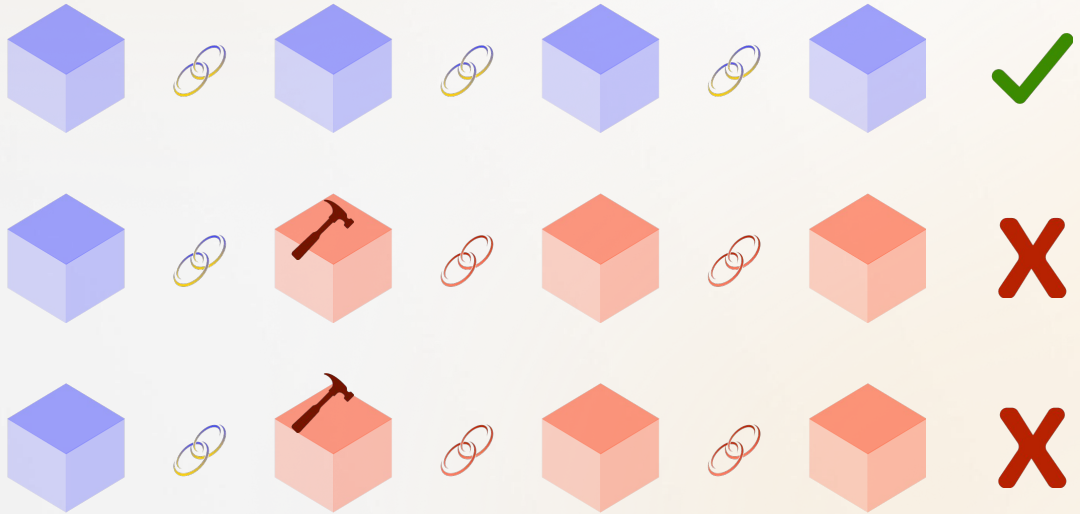
If you tried to change the data on one computers in a blockchain network, the change would not be accepted by the other computers in the network. This is because each computer in the network has a copy of the blockchain, and they all work together to validate new transactions and maintain the integrity of the data.



Patika dev BNB CHAIN

# 51% Attack

If malicious actors were able to take control of the majority of the computers on a blockchain network, they could potentially manipulate the network in various ways.
This is known as a "51% attack," as the attackers would need to control more than 50% of the network's computing power to carry out such an attack.



Patika dev BNB CHAIN

# What is Cryptography?

Cryptography is the practice of creating and using codes and ciphers to protect information from unauthorized access or manipulation.

It involves the use of mathematical algorithms and techniques to encode and decode data, ensuring that it can only be accessed or modified by those who have the proper key or knowledge.

Cryptography is a practice that has a long history, going back to ancient civilizations. It involves the use of codes and ciphers to protect information from unauthorized access or manipulation.

Throughout history, cryptography has been used for many purposes, such as securing military and diplomatic communications, verifying financial transactions, and maintaining privacy in electronic communication.

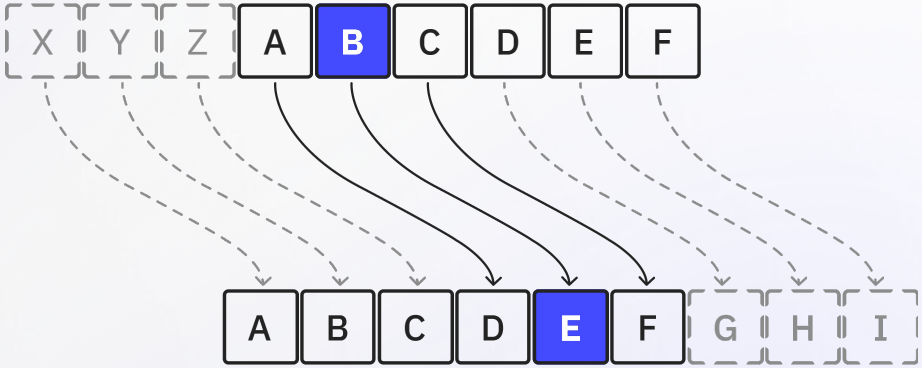
Patika dev BNB CHAIN

# Cryptography Example in History

The Caesar cipher was used by Julius Caesar to encode his communications. It is said that Caesar would send messages to his officials using this cipher, in order to protect the contents of the message from being intercepted and understood by enemies.

To use the Caesar cipher, Caesar would choose a shift value (for example, 3) and then shift each letter in the message by that number of positions. For example, if the shift value was 3, the letter "A" would be replaced with "D," "B" would be replaced with "E," and so on.

To decode the message, the recipient shifts each letter in the encoded message back by the same number of positions. In this way, Caesar could communicate privately with his officials, while protecting the contents of the message from being understood by others.




Patika dev BNB CHAIN

# Cryptography in Blockchain and Its Use

Cryptography is a fundamental aspect of blockchain technology, and is used to secure and protect the data stored on the blockchain.

It involves the application of mathematical algorithms and techniques to encode and decode data, ensuring that it can only be accessed or modified by those who possess the necessary key or knowledge.

Cryptography performs a number of crucial functions in the operation of a blockchain, including verifying the authenticity and integrity of transactions, protecting data from unauthorized access, and preserving the privacy of users.

# Cryptographic Methods Used in Blockchain

There are several cryptographic methods used in blockchain technology to secure and protect the data stored on the blockchain. Some of the most commonly used methods include:


## Hashing

This involves the use of a mathematical function to convert data into a fixed-length string of characters known as a "hash." This hash can be used to verify the integrity of the data, as any change to the data will result in a different hash being generated.

## Digital signatures

These are used to verify the authenticity of transactions on the blockchain and to ensure that they cannot be altered. Digital signatures use a pair keys (a public key and a private key) to create a unique signature that can be used to verify the identity of the sender.

## Public-key cryptography

This is a type of cryptography that uses a pair of keys (a public key and a private key) to secure communication. The public key is used to encrypt the data, and the private key is used to decrypt it.

## Symmetric-key cryptography

This is a type of cryptography that uses a single key to both encrypt and decrypt the data. The key must be kept secret in order to maintain the security of the data.

Patika dev BNB CHAIN

# Blockchain Key Terms

## Cryptographic Hash Functions

Patika dev

BNB CHAIN

# Content

# What are Cryptographic Hash Functions?

Cryptographic hash functions are mathematical functions that take an input and return a fixed-size string of characters, which is unique to the input.
The output is often referred to as the 'hash' or 'message digest'.
Two examples of cryptographic hash functions are MD5 and SHA-256.

It is possible to hash data, images, books, and videos using SHA-256 (or any other cryptographic hash function). Cryptographic hash functions can be applied to any type of data, regardless of its format or content. For example, you could hash a text file, an image file, a PDF document, or a video file using SHA-256. The hash function will take the data as input and produce a fixed-size output (256 bits) that is unique to the input.


Patika dev
BNB CHAIN

# Properties of Cryptographic Hash Functions

## One-way

It is computationally infeasible to determine the original input from the hash value. For example, it is difficult to determine the original message or data from a hash value.


## Deterministic

The same input will always produce the same output. For example, if you hash the message "Hello, world!" multiple times, you will always get the same hash value.

## Fast

It should be efficient to generate a hash value from an input. For example, it should take only a few milliseconds to generate a hash value for a message or data input.


## Collision resistance

It should be difficult to find two inputs that produce the same output. This ensures that each input is unique and distinguishable from others.

## Avalanche effect

A small change in the input should produce a significant change in the output. For example, changing one letter in "Hello, world!" should produce a very different hash value.

Patika dev BNB CHAIN

# Blockchain Key Terms

# Blockchain Workflow

Patika dev

BNB CHAIN

# Content

# Nodes

In a blockchain network, nodes are computers or devices that help maintain the distributed ledger and ensure network integrity. They verify and record transactions and ensure blockchain security.

Nodes help decentralize the blockchain and make it resistant to tampering or censorship by participating in the network and working together to validate and record new transactions, ensuring the accuracy and up-to-date status of the distributed ledger.

### Full Nodes:

Full nodes have a complete copy of the blockchain and are responsible for validating new transactions and blocks. They play a crucial role in maintaining the integrity of the blockchain.

### Lightweight Nodes:

Lightweight nodes, also known as SPV (Simplified Payment Verification) nodes, do not have a complete copy of the blockchain. Instead, they rely on full nodes to provide them with the necessary information to verify transactions.

### Mining Nodes:

In a proof-of-work blockchain, such as Bitcoin, mining nodes are responsible for solving complex mathematical problems in order to create new blocks and earn rewards.

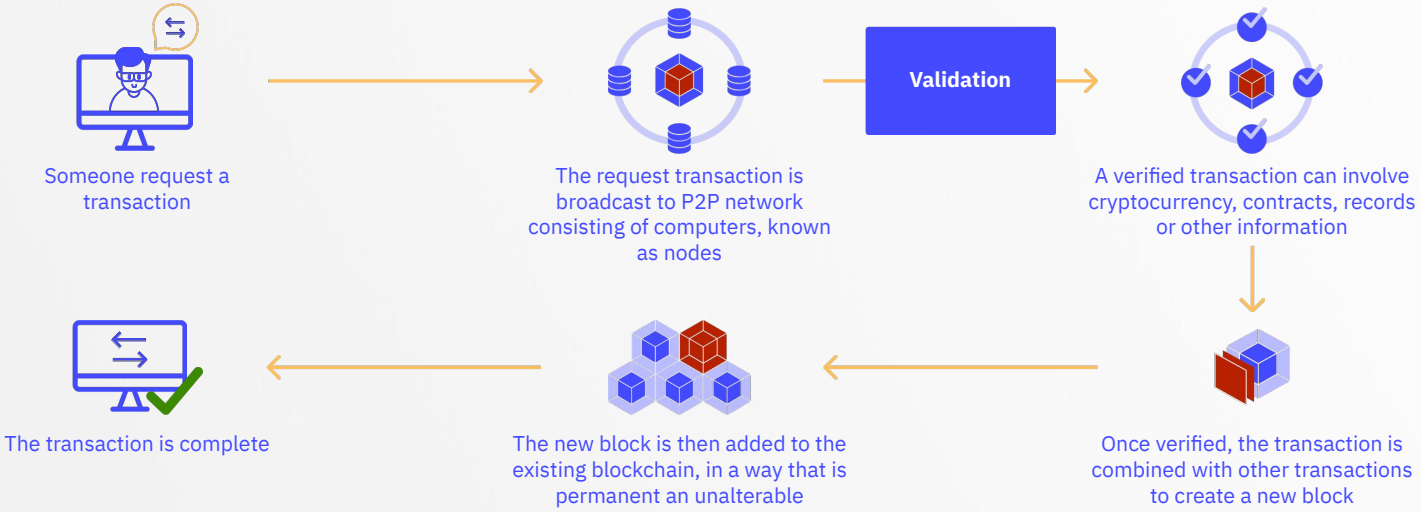
### Validator Nodes:

Validator nodes are responsible for creating new blocks on the Binance Smart Chain. They are selected through a process called "bonding," in which users can stake their tokens to become eligible to participate as a validator.

# Nodes: How are Transactions Executed?

A wallet stores keys and enables the sending and receiving of digital currency. When you send a transaction, your wallet connects to a node and broadcasts the transaction. The transaction is added to the blockchain once it is verified.

The node that a transaction is connected to records the transaction on the blockchain and makes it available to other nodes, helping to ensure the integrity and security of the blockchain.

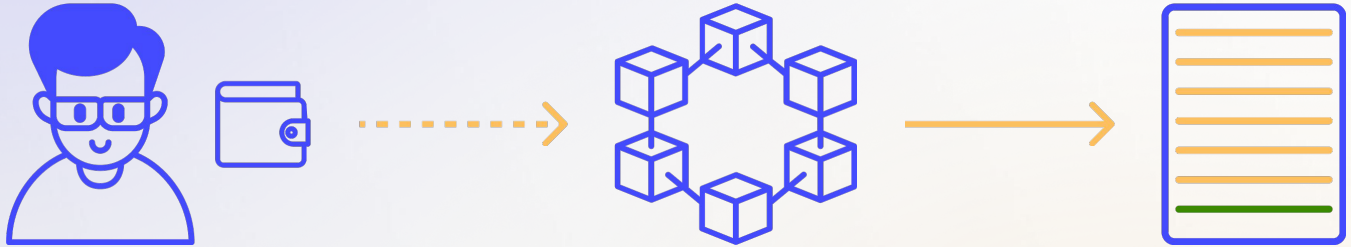



Patika dev BNB CHAIN

# Blockchain As Workflow: Transaction Pool

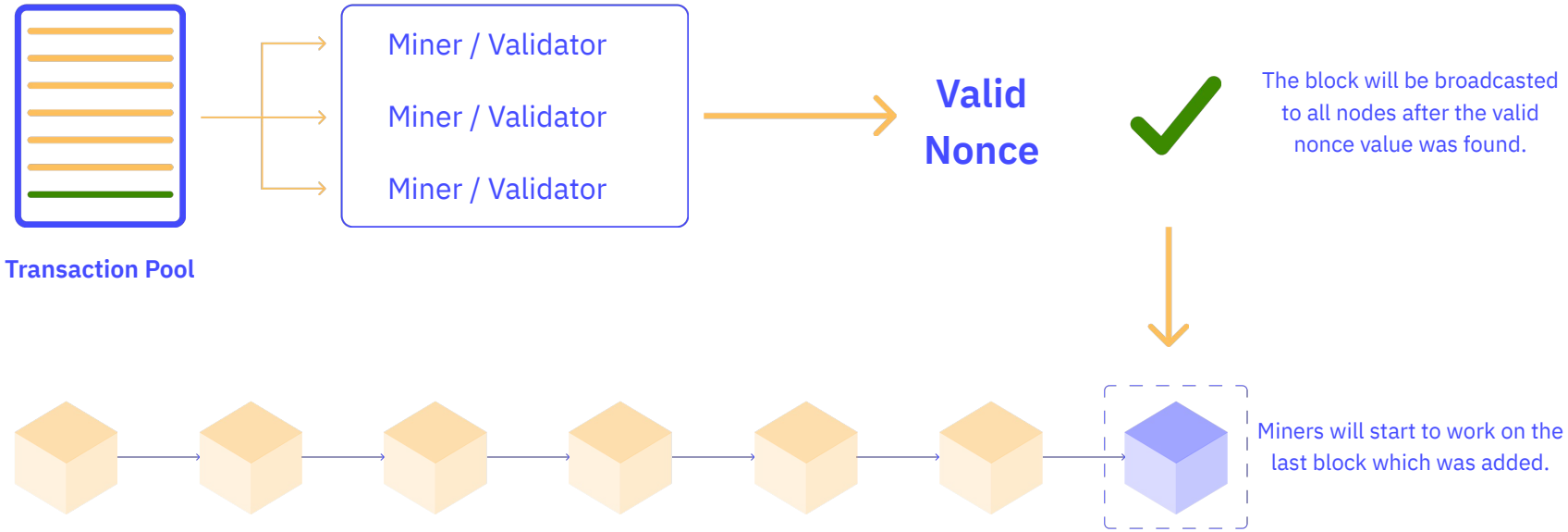
Transactions broadcast to the blockchain are temporarily stored in the transaction pool, also known as the "memory pool" or "mempool." This is a collection of unconfirmed transactions waiting to be included in the next block.

When a transaction is broadcast to the network, nodes receive it and place it in the transaction pool. It will then wait until a miner selects it to be included in the next block. Once a transaction is included in a block and the block is added to the blockchain, the transaction is considered confirmed.



# Blockchain As Workflow: Mining

Transactions in a blockchain network are temporarily stored in the transaction pool until they are added to a block by miners. When a block is successfully mined, it is broadcast to the network and added to the blockchain if it is deemed valid. This helps ensure the integrity and security of the blockchain.

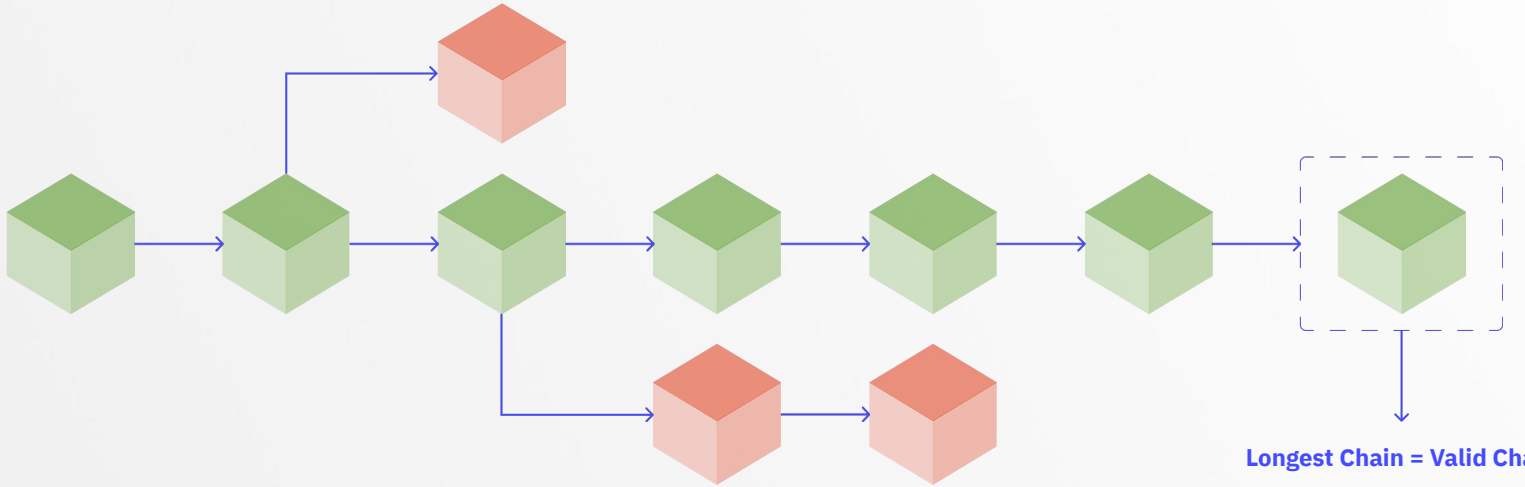



Patika dev BNB CHAIN

# Blockchain Forks: Temporary Fork

A temporary fork may occur when two valid hash values or two similar hash values are obtained simultaneously. This can happen when two miners simultaneously solve the PoW puzzle and create competing blocks.

The network resolves a fork by choosing the longest chain, which is assumed to represent the most work and be the most trustworthy, and discarding the other one. This ensures that only valid blocks are added to the blockchain and helps to maintain its integrity and security.

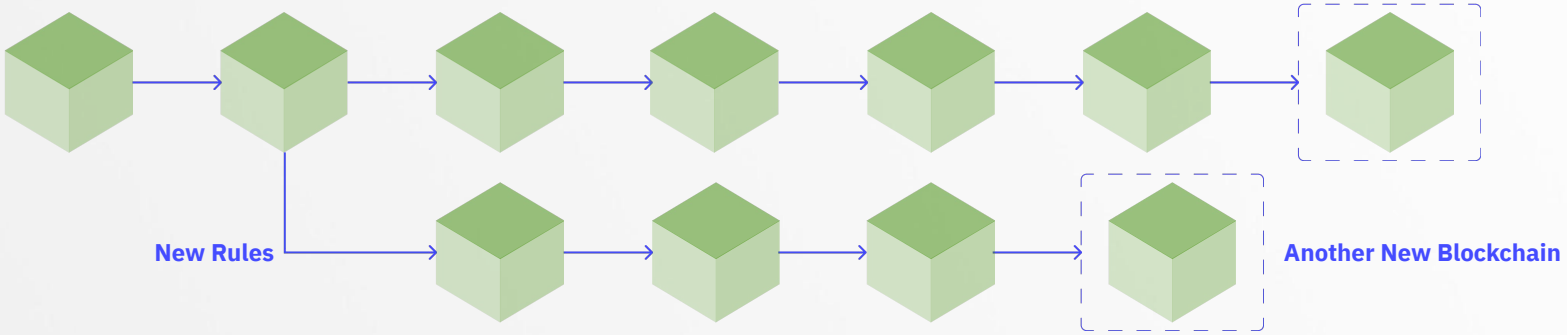



Patika dev BNB CHAIN

# Blockchain Forks: Hard Fork

A hard fork is an upgrade to a blockchain network that creates a new blockchain with different rules from the original. It can happen when there is a disagreement within the network or a desire to introduce new features to the network.

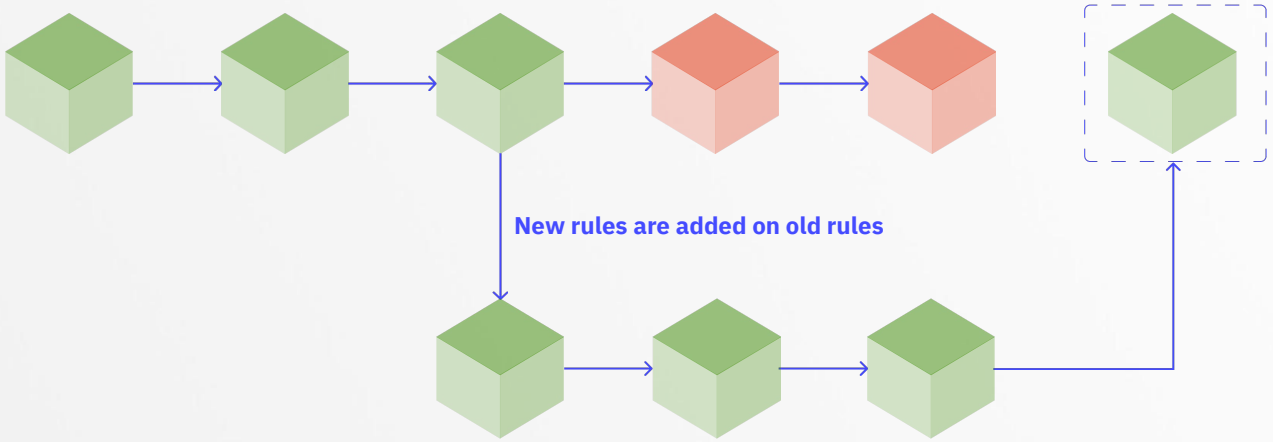
During a hard fork, the network splits into two separate blockchain networks: the original network and the new network. Each network has its own set of rules and operates independently of the other.




Patika dev BNB CHAIN

# Blockchain Forks: Soft Fork

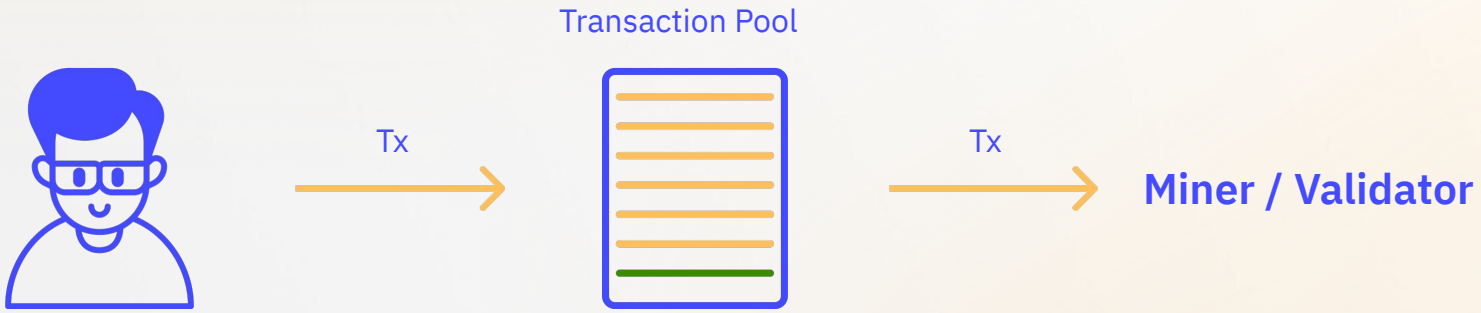
A soft fork is a backwards-compatible update to a blockchain network. It is used to introduce new features or make minor changes to the network. It does not result in the creation of a separate, incompatible blockchain. Nodes that have not yet upgraded can still participate in the network during a soft fork.




Patika dev BNB CHAIN

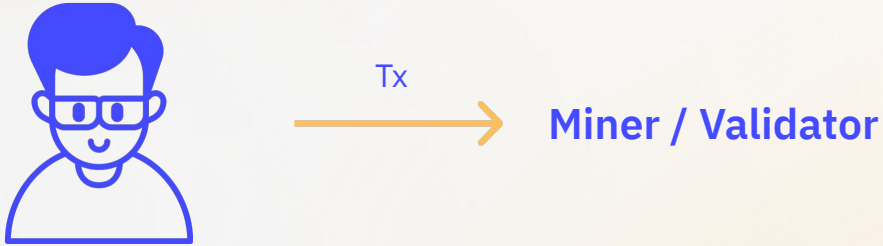
# Type of Blockchain Transactions: Public Transactions

Public transactions are stored in the transactions pool for a while and wait for confirmation.



# Type of Blockchain Transactions: Private Transactions

Private transactions are sent directly to the miner / validator without being stored in the transactions pool.



# Consensus Algorithm and Block Mining

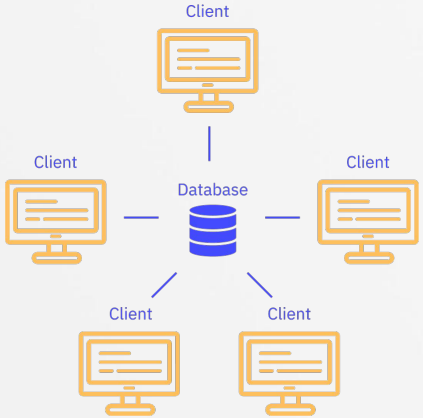
Patika dev BNB CHAIN

# Content

# Peer-to-Peer Network (P2P Network)

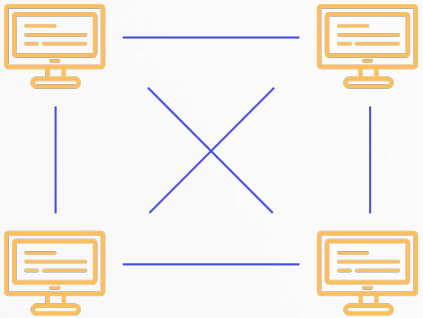
Peer-to-peer (P2P) networks are a type of computer network in which two or more computers are connected and share resources without the need for a central server. Instead of relying on a single central server to manage and distribute data, P2P networks allow computers to communicate and share information directly with each other.

## Client - Server Network




A client-server network is a type of computer network in which one or more computers, called "servers," provide data and resources to other computers, called "clients."

## Peer to Peer Network



P2P network is a type of computer network in which two or more computers are connected and share resources without the need for a central server.

Patika dev BNB CHAIN

# Server-Client Network vs P2P Network

| | Server-Client Networks | P2P Networks |
|---|---|---|
| **Architecture**  | Centralized, with one or more servers providing data and resources to clients. | Decentralized, with no central server. Each peer acts as both a client and a server, allowing users to share files and data directly with each other. |
| **Scability**  | May require additional servers to handle an increase in requests or clients. | Can be more scalable, as each peer can act as both a client and a server, allowing the network to handle more requests without the need for additional servers. |
| **Security**  | May be more vulnerable to security threats, as there is a central point of control that could be targeted by hackers. | May be more secure, as there is no central point of control that could be targeted by hackers. However, P2P networks can also be more vulnerable, as each peer is responsible for securing its own data and resources. |
| **Efficiency**  | Can be efficient, as all data and resources are stored and managed on a central server. | May be less efficient, as data and resources are distributed across multiple peers and may require multiple hops to reach their destination. |

Patika dev BNB CHAIN

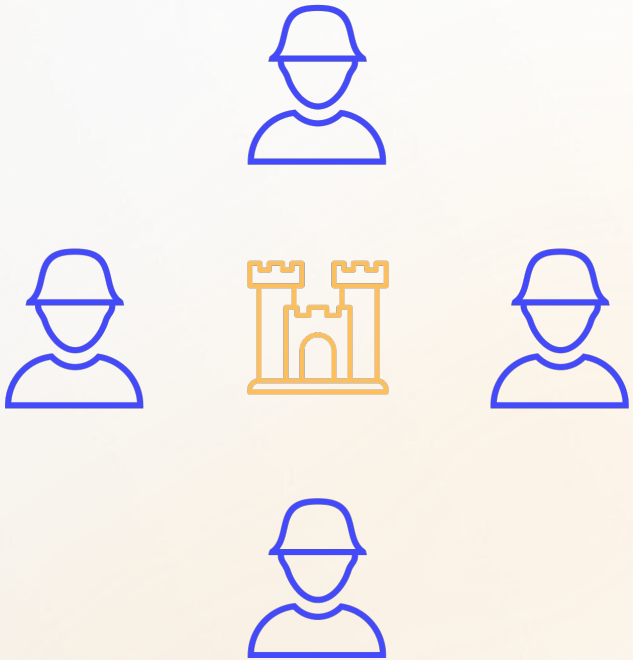
# The Byzantine Generals’ Problem

The Byzantine Generals' Problem, also known as the Byzantine Fault Tolerance (BFT) problem, is a hypothetical scenario that demonstrates the difficulty of achieving consensus in a distributed system when some nodes may not be trustworthy.

A group of generals are around an enemy city. They must decide on a strategy for attacking or retreating, and they must communicate to come to a consensus. However, some of the generals may be traitors who try to prevent the loyal generals from reaching an agreement.

The problem gets its name from a historical event in the Byzantine Empire, where a group of Byzantine generals were encamped around an enemy city and had to communicate with each other.

This problem is relevant to consensus algorithms because it highlights the challenges of achieving consensus in a distributed system. Consensus algorithms are designed to solve the Byzantine Generals' Problem by providing a way for nodes to reach agreement on a single value despite untrustworthy nodes.



Patika dev BNB CHAIN

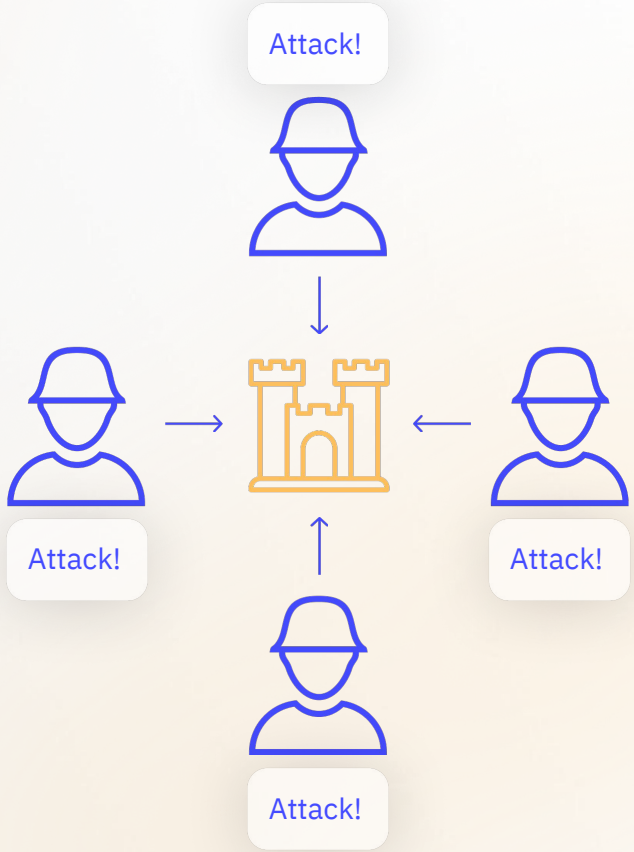
# The Byzantine Generals' Problem

The Byzantine Generals' Problem, also known as the Byzantine Fault Tolerance (BFT) problem, is a hypothetical scenario that demonstrates the difficulty of achieving consensus in a distributed system when some nodes may not be trustworthy.

A group of generals are around an enemy city. They must decide on a strategy for attacking or retreating, and they must communicate to come to a consensus. However, some of the generals may be traitors who try to prevent the loyal generals from reaching an agreement.

The problem gets its name from a historical event in the Byzantine Empire, where a group of Byzantine generals were encamped around an enemy city and had to communicate with each other.

This problem is relevant to consensus algorithms because it highlights the challenges of achieving consensus in a distributed system. Consensus algorithms are designed to solve the Byzantine Generals' Problem by providing a way for nodes to reach agreement on a single value despite untrustworthy nodes.




Patika dev

BNB CHAIN

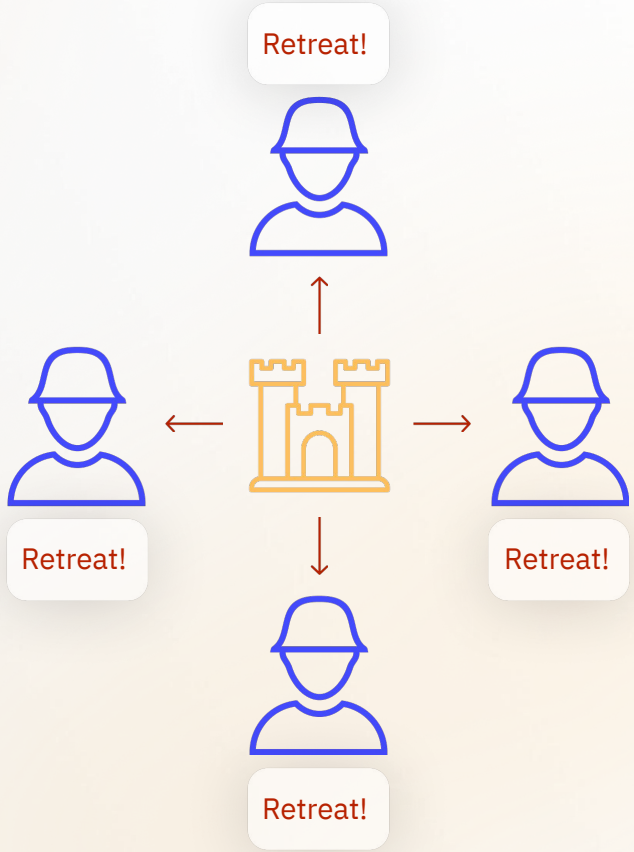
# The Byzantine Generals' Problem

The Byzantine Generals' Problem, also known as the Byzantine Fault Tolerance (BFT) problem, is a hypothetical scenario that demonstrates the difficulty of achieving consensus in a distributed system when some nodes may not be trustworthy.

A group of generals are around an enemy city. They must decide on a strategy for attacking or retreating, and they must communicate to come to a consensus. However, some of the generals may be traitors who try to prevent the loyal generals from reaching an agreement.

The problem gets its name from a historical event in the Byzantine Empire, where a group of Byzantine generals were encamped around an enemy city and had to communicate with each other.

This problem is relevant to consensus algorithms because it highlights the challenges of achieving consensus in a distributed system. Consensus algorithms are designed to solve the Byzantine Generals' Problem by providing a way for nodes to reach agreement on a single value despite untrustworthy nodes.




Patika dev BNB CHAIN

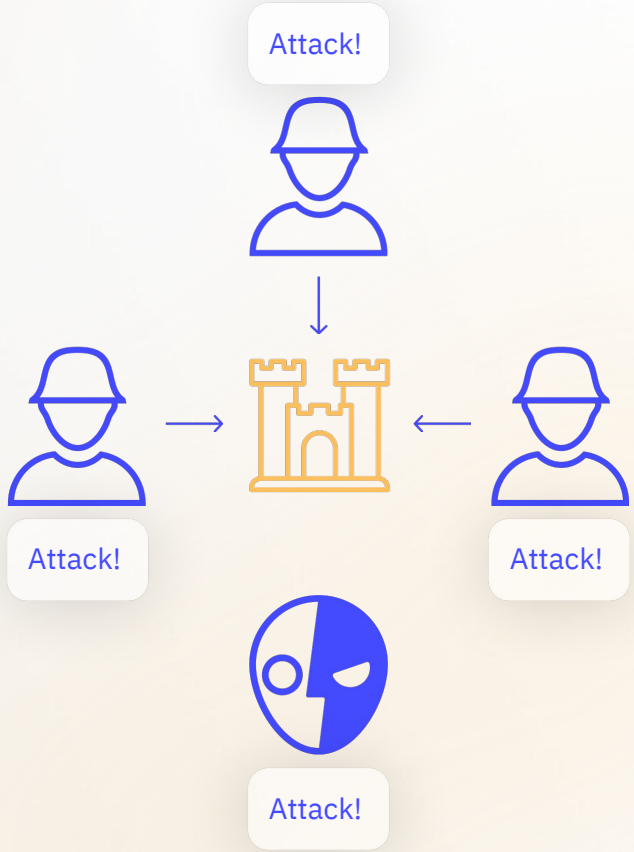
# The Byzantine Generals' Problem

The Byzantine Generals' Problem, also known as the Byzantine Fault Tolerance (BFT) problem, is a hypothetical scenario that demonstrates the difficulty of achieving consensus in a distributed system when some nodes may not be trustworthy.

A group of generals are around an enemy city. They must decide on a strategy for attacking or retreating, and they must communicate to come to a consensus. However, some of the generals may be traitors who try to prevent the loyal generals from reaching an agreement.

The problem gets its name from a historical event in the Byzantine Empire, where a group of Byzantine generals were encamped around an enemy city and had to communicate with each other.

This problem is relevant to consensus algorithms because it highlights the challenges of achieving consensus in a distributed system. Consensus algorithms are designed to solve the Byzantine Generals' Problem by providing a way for nodes to reach agreement on a single value despite untrustworthy nodes.




Patika dev

BNB CHAIN

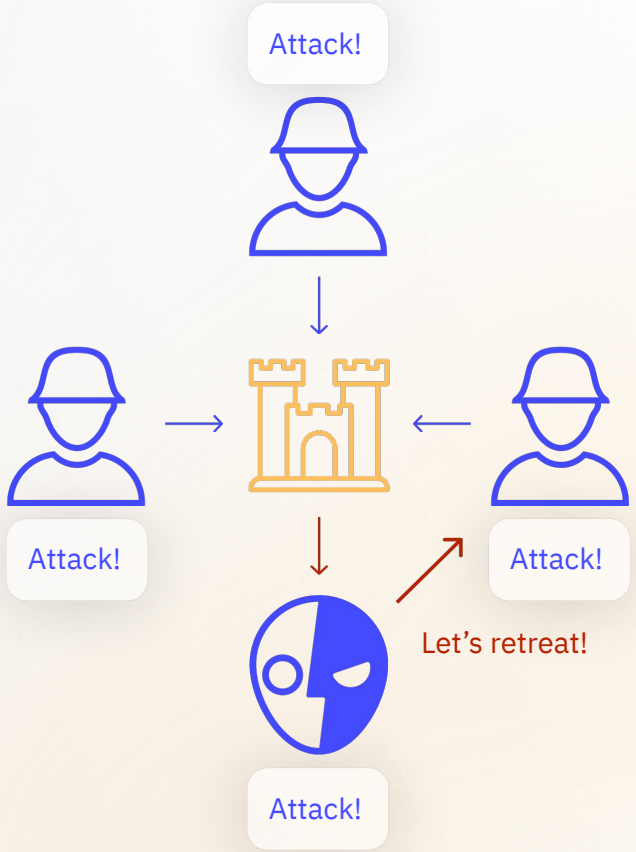
# The Byzantine Generals' Problem

The Byzantine Generals' Problem, also known as the Byzantine Fault Tolerance (BFT) problem, is a hypothetical scenario that demonstrates the difficulty of achieving consensus in a distributed system when some nodes may not be trustworthy.

A group of generals are around an enemy city. They must decide on a strategy for attacking or retreating, and they must communicate to come to a consensus. However, some of the generals may be traitors who try to prevent the loyal generals from reaching an agreement.

The problem gets its name from a historical event in the Byzantine Empire, where a group of Byzantine generals were encamped around an enemy city and had to communicate with each other.

This problem is relevant to consensus algorithms because it highlights the challenges of achieving consensus in a distributed system. Consensus algorithms are designed to solve the Byzantine Generals' Problem by providing a way for nodes to reach agreement on a single value despite untrustworthy nodes.




Patika dev BNB CHAIN

# What is Consensus Algorithm?

Consensus algorithms are protocols or processes that are used to achieve agreement on a single value among a group of distributed processes or systems. These algorithms can be evaluated based on several attributes, including fault tolerance, scalability, security, and decentralization.

### Fault Tolerance

A fault-tolerant consensus algorithm can reach consensus even when some nodes fail or behave incorrectly. For example, it may be able to function with unavailable or conflicting nodes.

### Scalability

Scalability refers to the ability of a consensus algorithm to handle an increase in the number of nodes or the volume of transactions. A scalable consensus algorithm should be able to continue reaching consensus efficiently as the size of the network increases.

### Security

The security of a consensus algorithm refers to its ability to protect against attacks or other threats. For example, a secure consensus algorithm might use cryptographic techniques to ensure that transactions are secure and cannot be altered.

### Decentralization

Decentralization refers to the distribution of control and decision-making power among the participating nodes in a network. A decentralized consensus algorithm relies on multiple nodes to reach consensus, rather than a single central authority.

# Proof of Work Consensus Algorithm

The proof of work (PoW) mechanism is a type of consensus algorithm that is used to achieve distributed consensus in a distributed system.

## Key points about the proof of work mechanism:

- It is used to achieve distributed consensus in a distributed system.
- It requires miners to solve a computational puzzle in order to create new blocks.
- The first miner to solve the puzzle is allowed to create a new block and is rewarded with tokens or a block reward.
- The puzzle is designed to be difficult to solve, but easy to verify once it has been solved.
- It helps secure the network by requiring a significant amount of computational power to solve the puzzle and by making it easy for other nodes to verify the solution.

# Proof of Work Consensus Algorithm

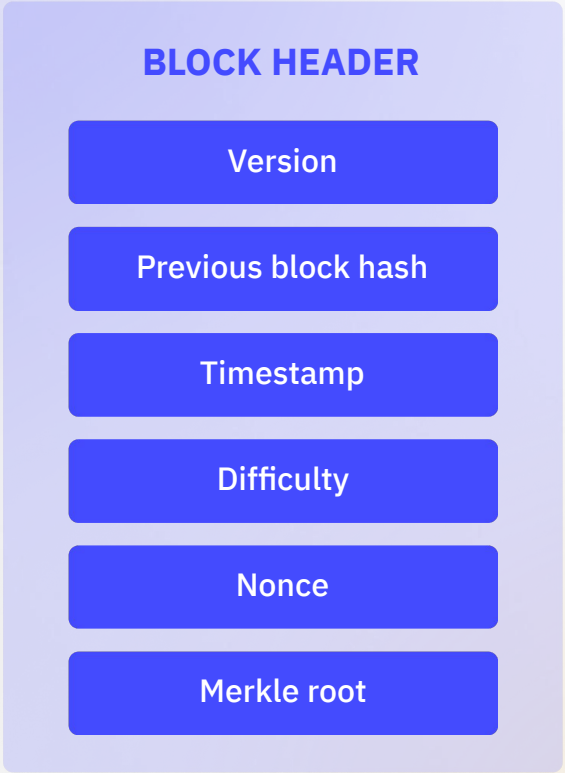
To keep the block production rate limited, miners must solve a puzzle for each block. This puzzle is about the block hash being in a certain format.

**For example, miners should get a hash value whose first 4 digits are 0.**

Example: 000012fa9b916eb9078f8d98a7864e697ae83ed54f5146bd84452cdafd043c19

In this way, malicious miners are prevented from producing blocks one after the other.

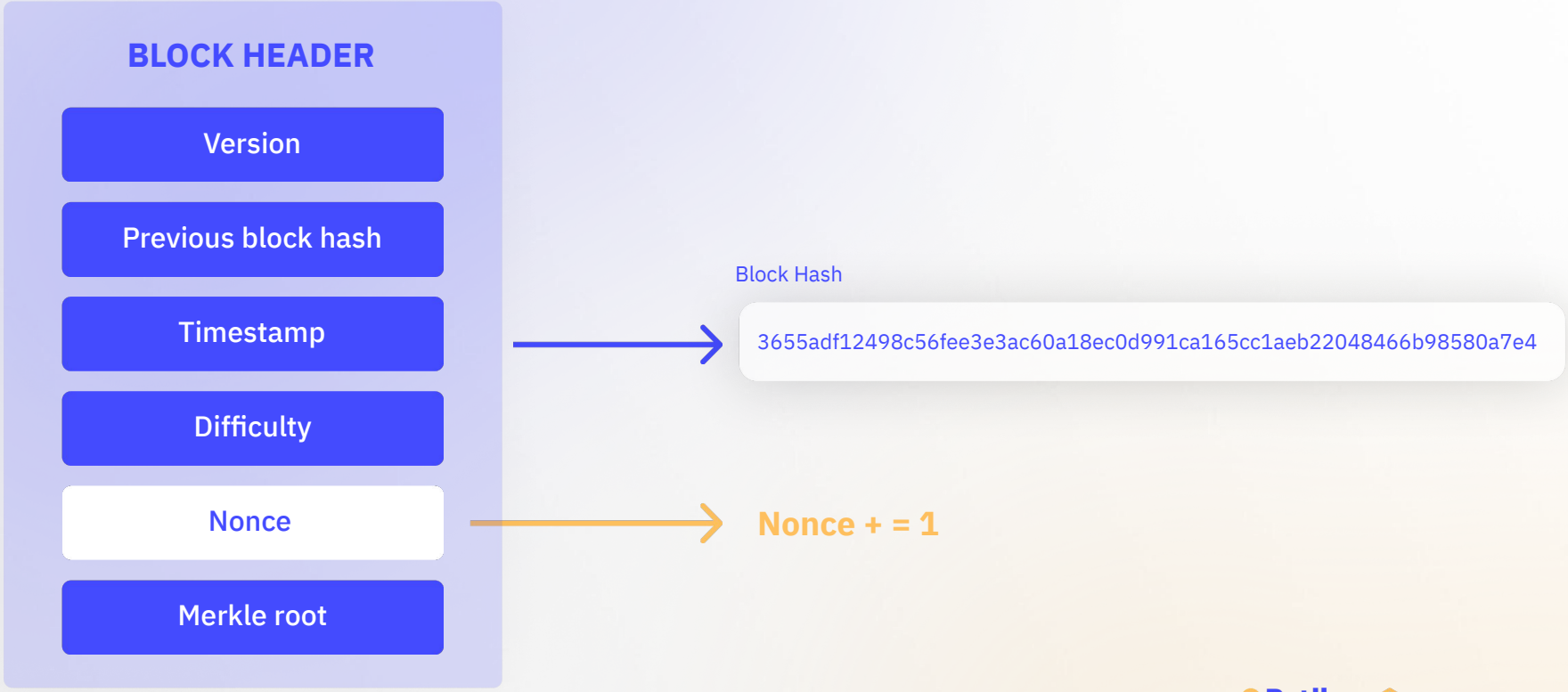
# Proof of Work Consensus Algorithm




Patika dev

BNB CHAIN

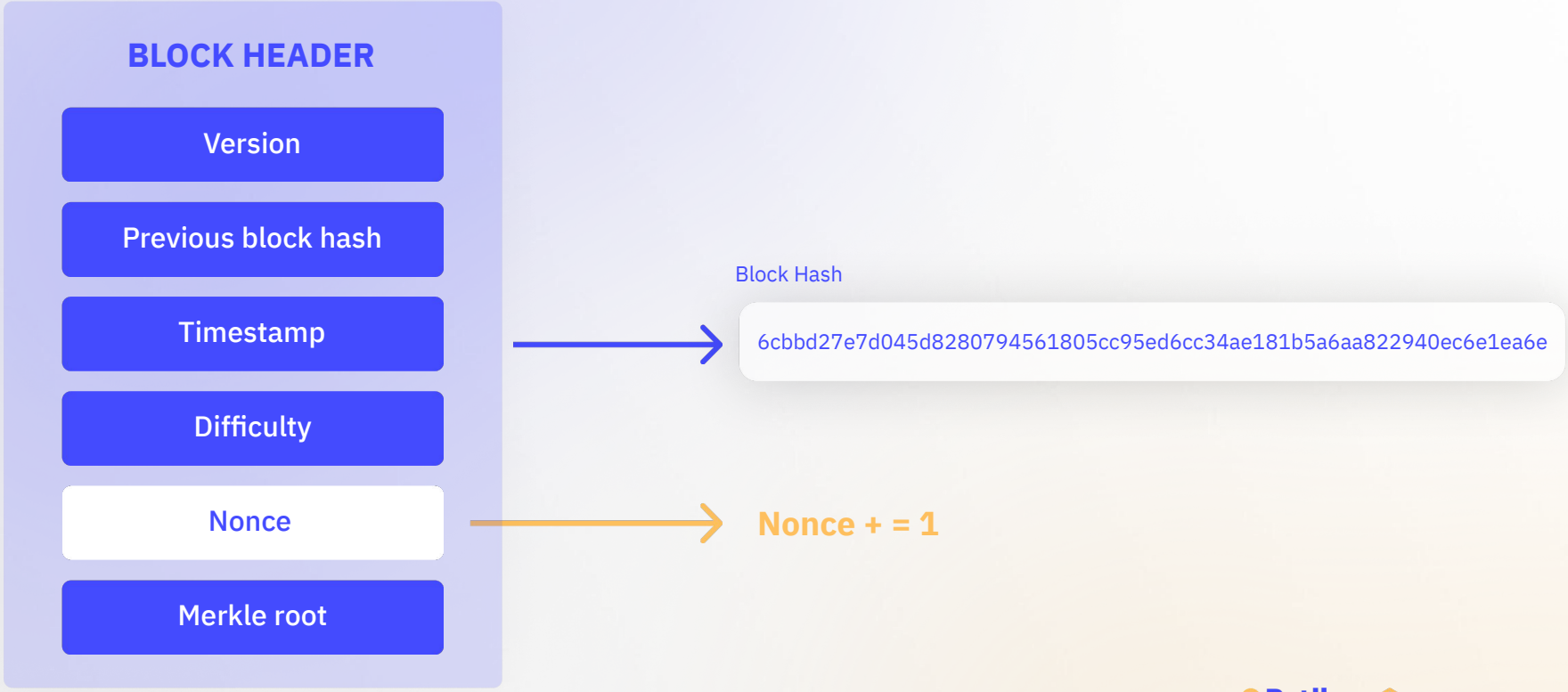
# Proof of Work Consensus Algorithm




Patika dev

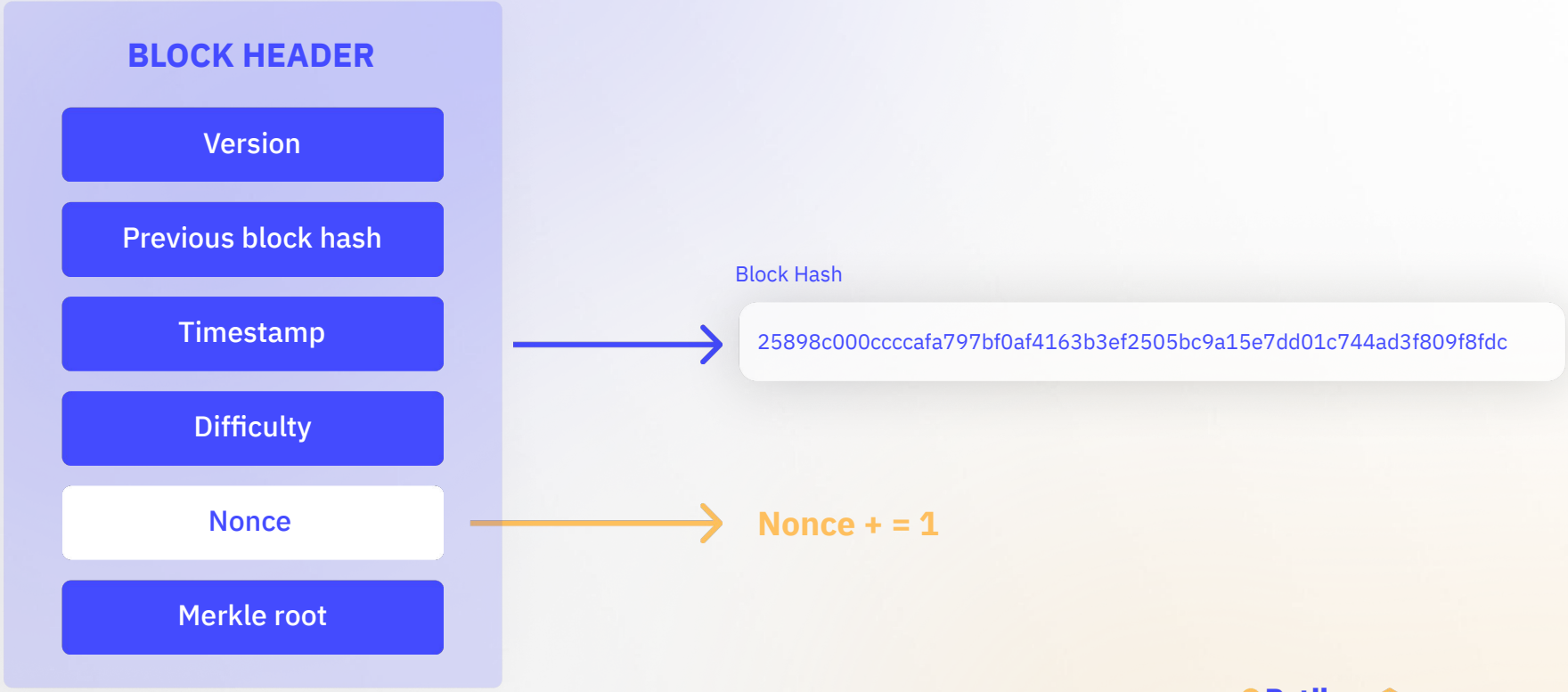
BNB CHAIN

# Proof of Work Consensus Algorithm




Patika dev BNB CHAIN

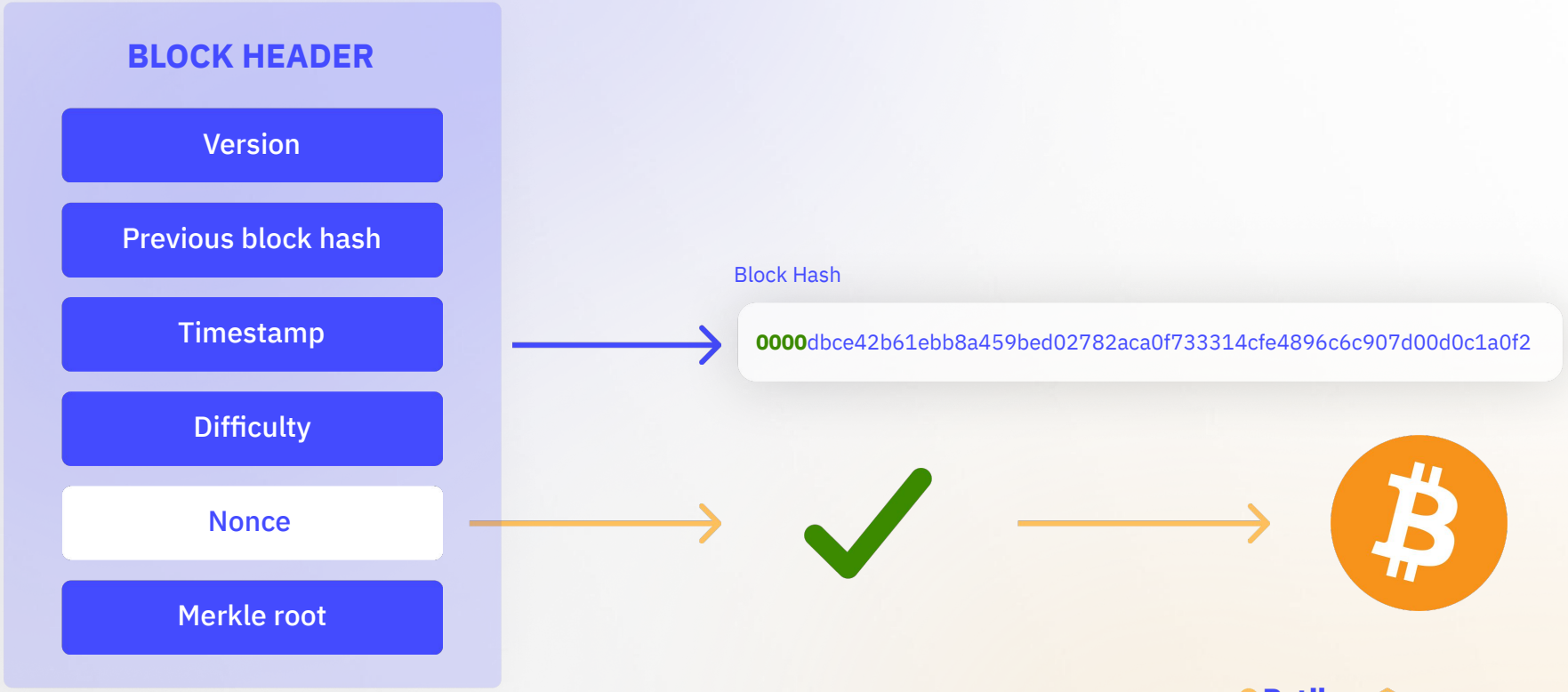
# Proof of Work Consensus Algorithm




Patika dev

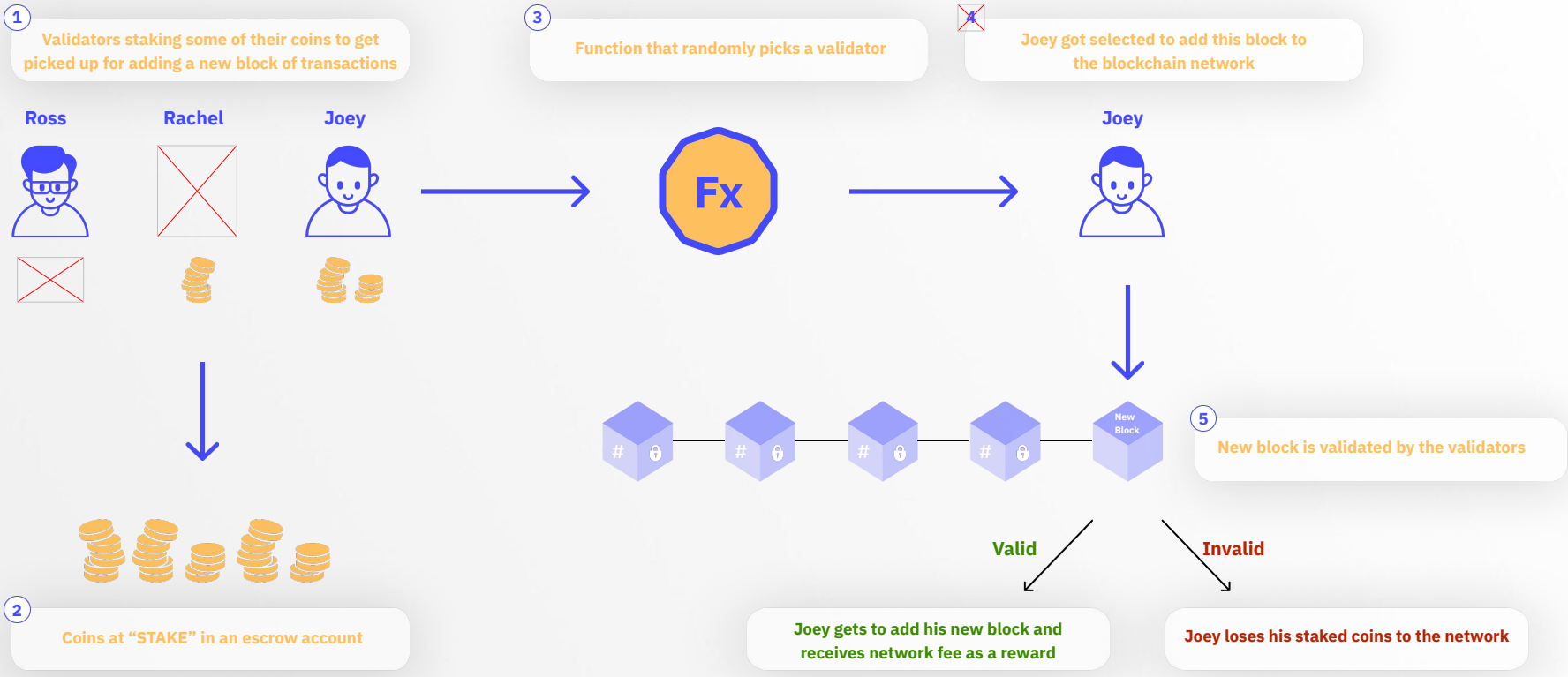
BNB CHAIN

# Proof of Work Consensus Algorithm




Patika dev BNB CHAIN

# Proof of Stake (PoS) Consensus Algorithm




Patika dev

BNB CHAIN

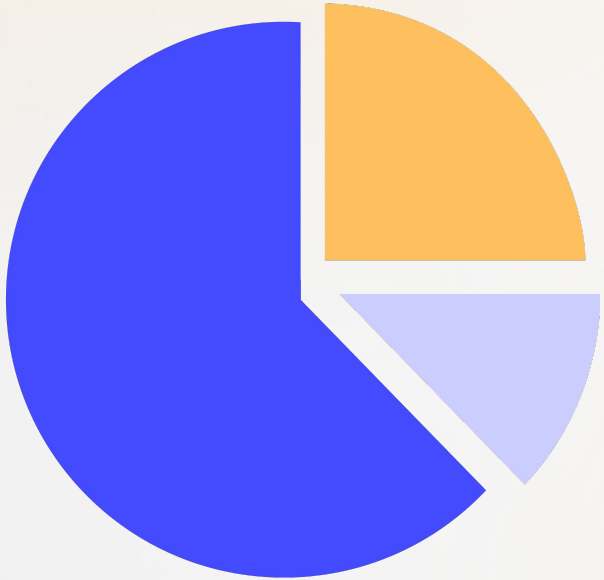
# Proof of Stake (PoS) Consensus Algorithm

A certain amount of coins must be staked to become a validator.

The block generator is randomly selected, considering parameters such as the stake duration and the amount of coins staked.

Compared to PoW, energy consumption is very low.

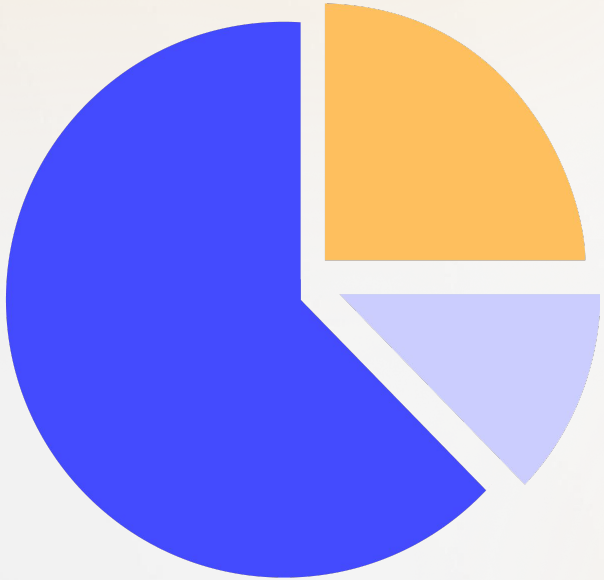
# Delegated Proof of Stake (DPoS) Algorithm - SUI



It is a variation of the Proof of Stake (PoS) Algorithm.

You stake coins in escrow to a delegate. If the delegate wins the right to create a block and earns coins, you receive a share of the reward based on your deposit.

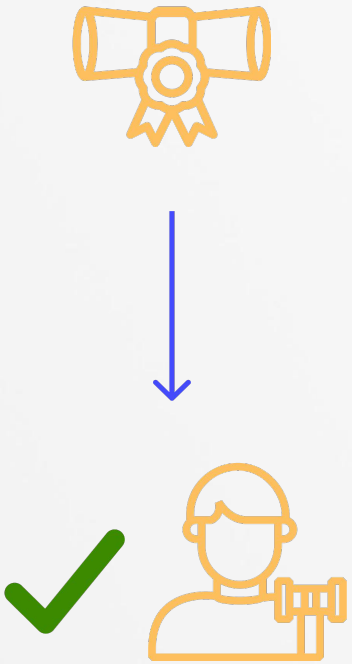
# Delegated Proof of Stake (DPoS) Algorithm - SUI



Delegate selection process is similar to PoS.

The chosen delegates are then responsible for validating transactions and adding them to the blockchain.

# Proof of Authority (PoA) Consensus Algorithm



- Validators are limited.
- Scalability is very high.
- Decentralization is compromised.
- It is based on trust.
- It is generally preferred in private and consortium blockchains.

Patika dev BNB CHAIN

# Proof of Stake Authority (PoSA) Consensus Algorithm

PoSA is a combination of PoA and DPoS. Blocks are produced by a limited set of validators, they are elected in and out based on staking-based governance.

DPoS + PoA = PoSA

Fast transactions, shorter block times.

Cheaper transactions, reduced costs.

Patika dev BNB CHAIN

# Other Variations of Consensus Algorithms

**Proof-of-Capacity:**

A proof type that requires a participant to show proof that they have dedicated a certain amount of disk space to the network.

**Proof-of-Elapsed Time:**

A proof type that requires a participant to show proof that a certain amount of time has elapsed in order to participate in the network.

**Proof-of-Identity:**

A proof type that requires a participant to show proof of their identity in order to participate in the network.

**Proof-of-Activity:**

A proof type that requires a participant to show proof of their activity, such as by completing a task or contributing to the network, in order to participate.

# Understanding Smart Contracts and Cryptocurrency

Patika dev

BNB CHAIN

# Content



Patika dev BNB CHAIN

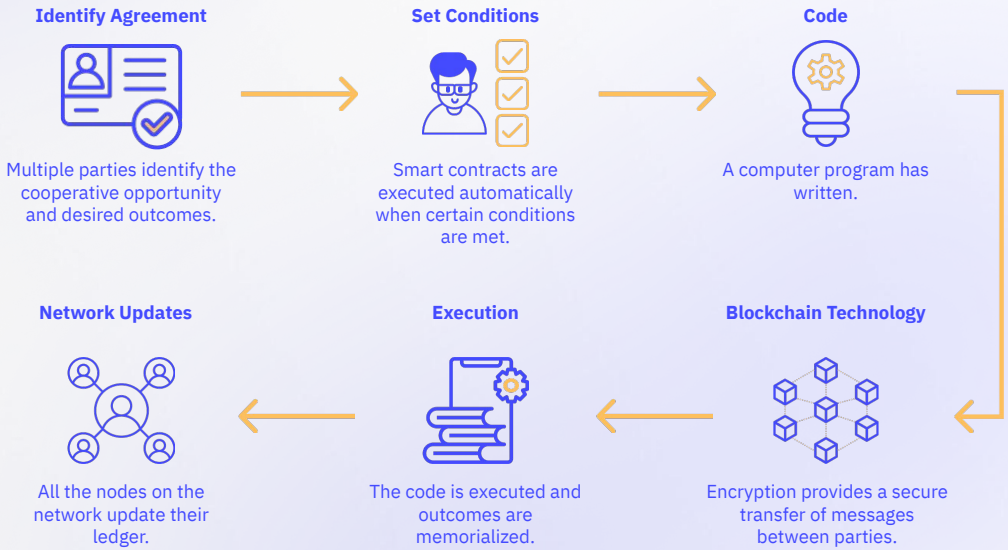
# Introduction to Smart Contracts

A smart contract is a self-executing contract with the terms of the agreement between buyer and seller written into lines of code and stored on the blockchain network.

Smart contracts automate processes and reduce the need for intermediaries and manual processing. They are often used to facilitate, verify, and enforce the negotiation or performance of a contract.

A traditional contract is a legal agreement between parties that specifies the terms of a transaction or relationship. It outlines the rights and obligations of each party and may include clauses outlining what should happen if one party fails to fulfill their obligations.

Smart contracts are similar to traditional contracts in outlining the terms of an agreement, but they are written in code and stored on a blockchain. This makes them easier to enforce and execute, as the terms are automatically enforced by the code.




Patika dev BNB CHAIN

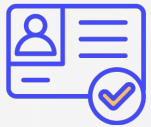
# Introduction to Decentralized Applications (DApps)

DApps are software programs that operate on a decentralized network and are built on a decentralized platform like a blockchain. They are distributed, transparent, and resistant to tampering, and are not controlled by a single entity, making them immune to censorship and ensuring their openness and transparency.

DApps usually have 1) a **front-end interface** for users to interact with, 2) a **back-end codebase** that runs on a decentralized platform. The back-end code is open source and designed to be transparent and verifiable.

## There are various types of DApps, including:

**Financial DApps** used for financial transactions like cryptocurrency exchanges and lending platforms



**Identity DApps** for identity verification and management like identity systems and reputation systems

**Supply chain DApps** for supply chain management like tracking the movement of goods and materials

**Voting DApps** for online voting and governance like decision-making platforms and election systems

Patika dev BNB CHAIN

# How Do Major Blockchain Platforms Work?

| | |
|---|---|
| Only supports money transfers |  |
| EVM Compatible |  |
| Non-EVM |  |
| Cannot run smart contracts |  |

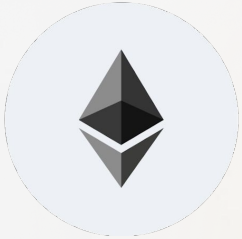
Patika dev

BNB CHAIN

# What is Cryptocurrency?

Cryptocurrency is a digital or virtual currency that uses cryptography for security. It is decentralized, meaning it is not controlled by a central authority like a central bank or government. Cryptocurrencies are based on distributed ledger technology, such as blockchain, and are secured through a process called mining.

Cryptocurrencies can be used as a medium of exchange and traded on online exchanges or used to purchase goods and services from merchants who accept them. They can also act as a store of value, similar to traditional fiat currencies like the US dollar or the euro.

Patika dev BNB CHAIN

# What is the Role of Blockchain?

Cryptocurrencies can be examined in two categories: Coin and token.

Coin

Token

Patika dev

BNB CHAIN

# What is the Role of Blockchain?

Coins are the current cryptocurrency of a blockchain. They can be used for value transfer, payment transactions, payment of gas fees, representing the voting weight in the decision-making process on the blockchain, and other similar purposes.

Coin

Token

Patika dev

BNB CHAIN

# What is the Role of Blockchain?

Tokens, on the other hand, are digital assets that represent a specific service, attribute, asset running on existing blockchains.

Coin

Token

Patika dev

BNB CHAIN

# How Cryptocurrencies Gain Value

The price of cryptocurrencies is determined by the supply and demand balance. When more people want to buy a cryptocurrency, its value increases. Similarly, its value decreases when demand decreases.



## Some factors affecting supply and demand:

- the use of money
- the benefits it provides
- the need for that cryptocurrency
- the amount of supply in circulation

Patika dev BNB CHAIN