Chapter 2: Building Blockchain

By Maitri Hingu

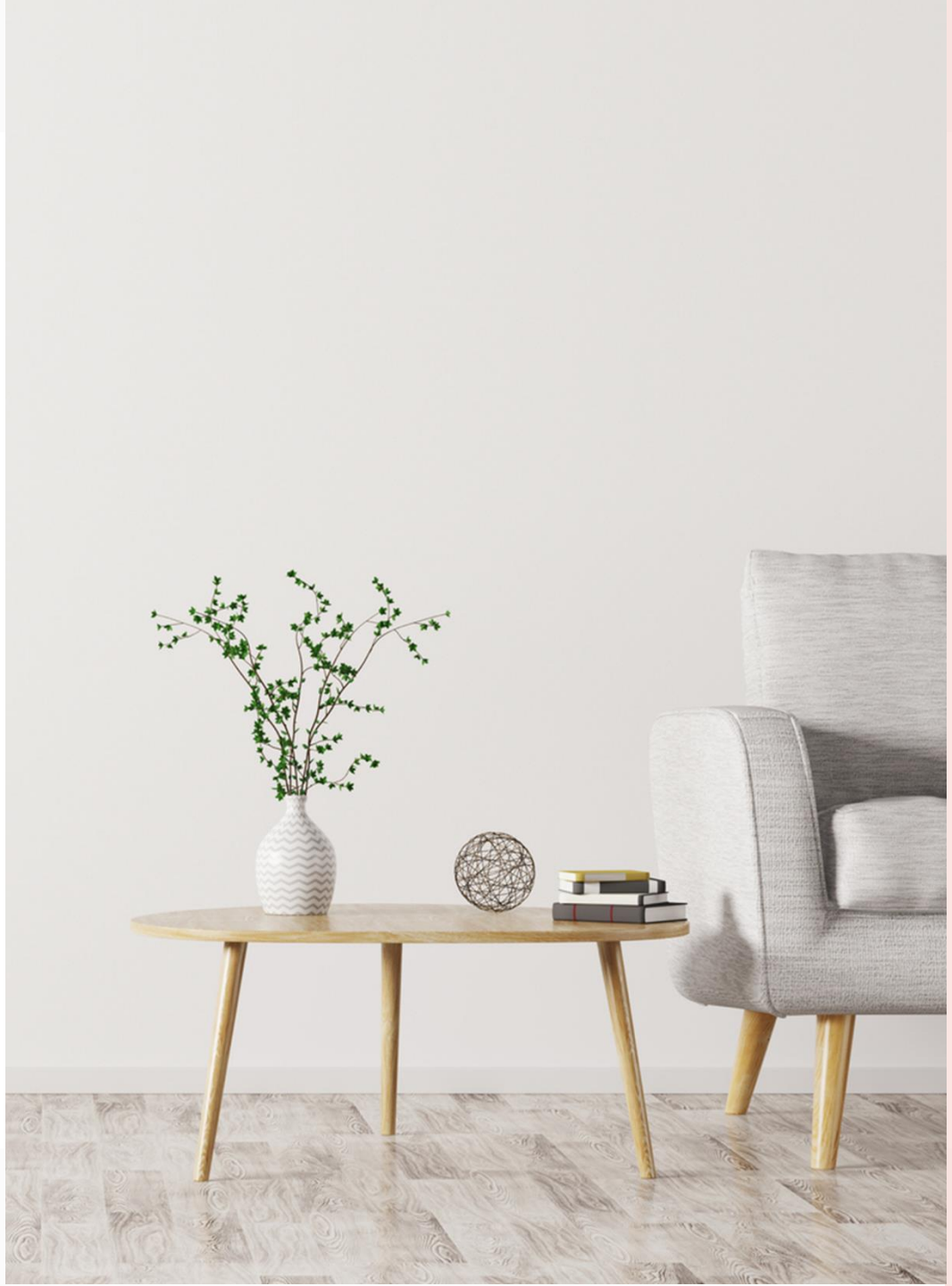
Agenda




- Essentials of Blockchain
- Blockchain Architecture & Generic of Blockchain
- Types of Blockchain
- Consensus Mechanism
- Creating Blocks & Links, Inserting Hashes
- Forking in Blockchain

By Maitri Hingu

Essentials of Blockchain



Essentials of Blockchain

- **Decentralization:**

Blockchain operates on a decentralized network of nodes (computers). There is no central authority or intermediary controlling the entire system. This decentralization enhances security, transparency, and resilience.

- **Distributed Ledger:**

The ledger, which contains a record of all transactions, is distributed across all nodes in the network. Every participant has a copy of the entire blockchain, ensuring that no single point of failure exists.

- **Cryptography:**

Cryptographic techniques, such as hashing and digital signatures, are crucial for securing transactions and ensuring data integrity. Hash functions generate unique identifiers for data, and digital signatures authenticate the origin of transactions.

Essentials of Blockchain

- **Consensus Mechanism:**

Consensus mechanisms are protocols that enable nodes in the network to agree on the state of the blockchain. Common mechanisms include Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS). Consensus ensures that all nodes reach a common understanding of the blockchain's state.

- **Smart Contracts:**

Smart contracts are self-executing contracts with the terms of the agreement written in code. They automatically execute and enforce the terms of an agreement when predefined conditions are met. Ethereum is a popular blockchain platform known for supporting smart contracts.

Essentials of Blockchain

- **Immutability:**

Once a block is added to the blockchain, it is extremely difficult to alter or delete. The cryptographic link between blocks and the proof-of-work or other consensus mechanisms contribute to the immutability of the blockchain.

- **Transparency:**

All transactions on the blockchain are visible to participants in the network. This transparency fosters trust and accountability among users, as they can independently verify the history of transactions.

- **Security:**

The combination of decentralization, cryptography, and consensus mechanisms enhances the security of blockchain. The decentralized nature makes it resistant to attacks, while cryptographic techniques secure data and transactions. Consensus mechanisms ensure agreement on the validity of transactions.

Essentials of Blockchain

- **Tokenization:**

Many blockchain networks involve the creation and use of native tokens. These tokens can represent assets, access rights, or serve as a medium of exchange within the blockchain ecosystem.

- **Interoperability:**

Efforts are made to enable different blockchain networks to communicate and share data seamlessly. Interoperability allows for greater flexibility and collaboration between different blockchain platforms.

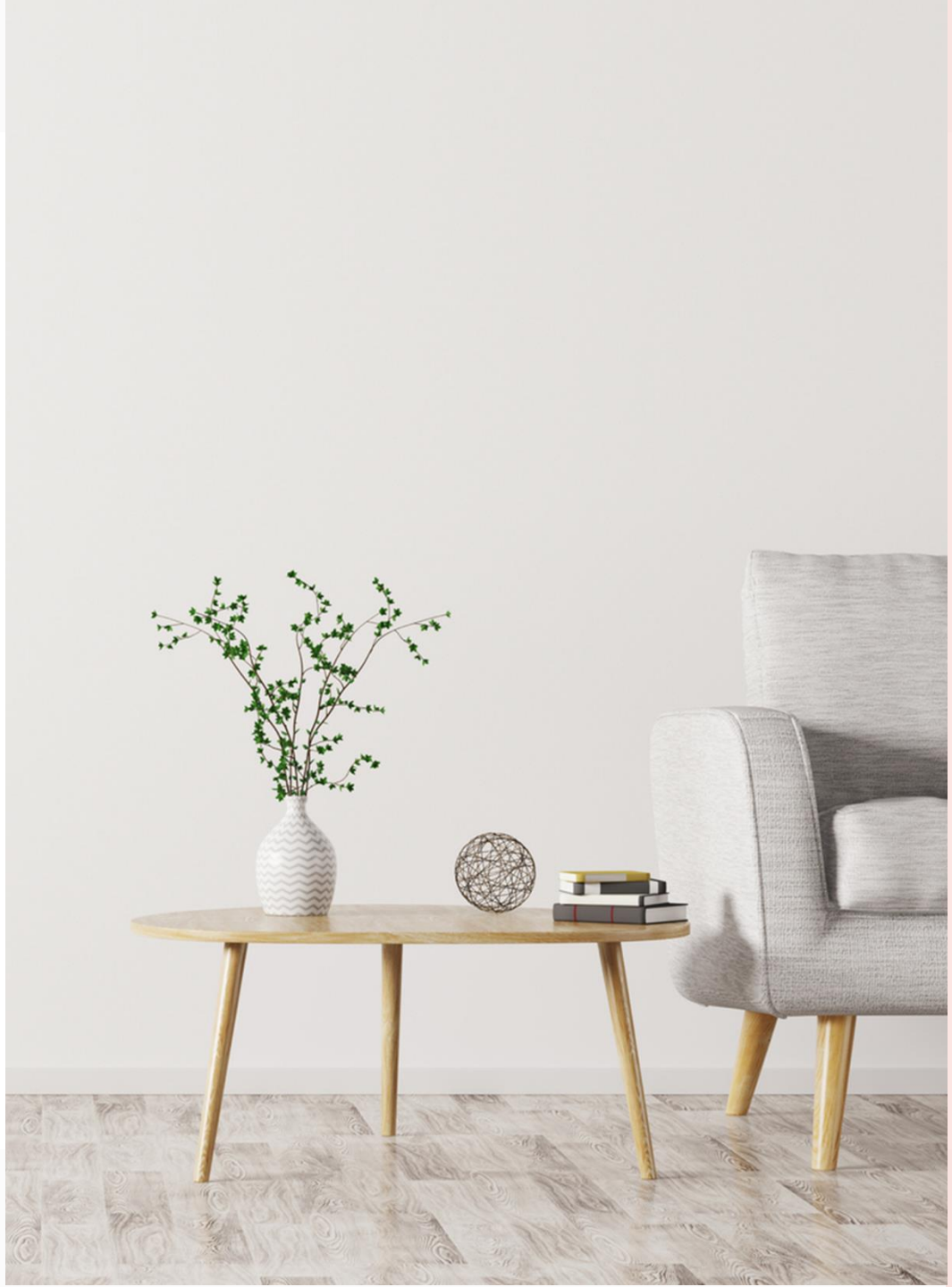
Essentials of Blockchain

- **Permissioned and Permissionless Blockchains:**

Blockchains can be categorized as permissioned (private) or permissionless (public). Permissioned blockchains restrict participation to authorized entities, while permissionless blockchains allow anyone to participate.

Understanding these essentials provides a foundation for grasping the principles and mechanisms that make blockchain a transformative technology in various industries, from finance to supply chain management.

Blockchain Architecture



Blockchain Architecture

The architecture of a blockchain system consists of several components working together to ensure the secure and decentralized nature of the network.

Nodes: Nodes are individual computers or devices that participate in the blockchain network. Each node maintains a copy of the entire blockchain ledger and follows the rules defined by the consensus mechanism. Nodes can be categorized as:

- **Full Nodes:** Store the entire blockchain and participate in transaction validation.
- **Miners/Validators:** Nodes that contribute computational power to validate transactions and create new blocks (in proof-of-work or similar consensus mechanisms).

Blockchain Architecture

Consensus Mechanism: The consensus mechanism is a set of rules or protocols that nodes follow to agree on the state of the blockchain.

Common consensus mechanisms include:

- **Proof of Work (PoW):** Miners solve complex mathematical puzzles to validate transactions and add blocks. The first to solve the puzzle gets the right to add the block.
- **Proof of Stake (PoS):** Validators are chosen to create new blocks based on the amount of cryptocurrency they hold and are willing to "stake" as collateral.
- **Delegated Proof of Stake (DPoS):** Similar to PoS, but a small number of nodes are elected by the community to validate transactions and create blocks.

Blockchain Architecture

Blockchain Protocol: The protocol defines the rules and standards that govern how nodes communicate and validate transactions. Bitcoin and Ethereum are examples of blockchain protocols, each with its own set of rules.

Blocks: Transactions are grouped into blocks, and each block contains a cryptographic hash of the previous block, creating a chain of blocks. A block typically includes:

- A timestamp.
- A reference to the previous block's hash.
- A Merkle tree of transaction hashes.
- A nonce (for PoW) or other consensus-related data.

Blockchain Architecture

Transaction Pool/Mempool: Transactions initiated by users are first stored in the transaction pool (mempool) before being included in a block. Miners select transactions from the pool to include in a new block during the mining process.

Smart Contracts: Smart contracts are self-executing programs with predefined rules encoded on the blockchain. They automatically execute when certain conditions are met. Ethereum is a notable blockchain platform that supports smart contracts.

Wallets: Wallets are software or hardware tools that enable users to store their cryptographic keys (private and public keys) and interact with the blockchain. They allow users to initiate transactions, check balances, and manage their cryptocurrency holdings.

Cryptographic Hash Functions: Cryptographic hash functions (e.g., SHA-256) are used to create unique, fixed-size identifiers for data, ensuring data integrity and security within the blockchain.

Blockchain Architecture

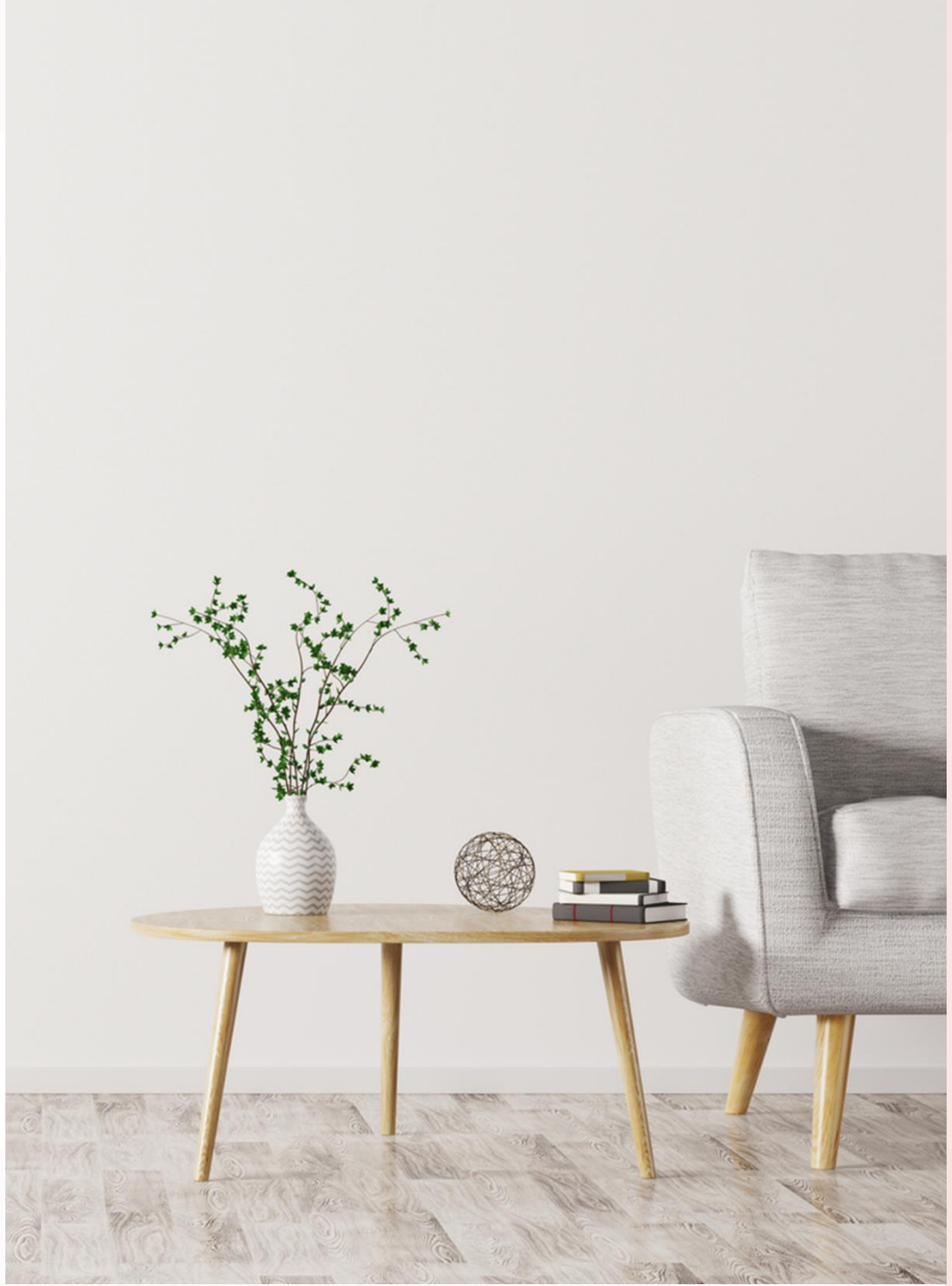
Decentralized Applications (DApps): DApps are applications built on top of blockchain platforms that leverage smart contracts. They often run on a decentralized network and provide functionality similar to traditional applications.

Network and P2P Communication: Nodes communicate with each other through a peer-to-peer (P2P) network. P2P communication ensures that there is no central point of control or failure in the network.

Mining Reward/Transaction Fees: Miners are typically rewarded with newly created cryptocurrency (block reward) and transaction fees for successfully adding a block to the blockchain. This incentivizes miners to participate in the network.

Understanding the architecture of a blockchain system involves grasping the roles and interactions of these components, each contributing to the overall security, transparency, and decentralization of the network. Different blockchain networks may have variations in their architecture, depending on the specific protocol and consensus mechanism employed.

Generic Elements of Blockchain



Generic Elements of Blockchain

Blocks:

- A blockchain consists of a chain of blocks, where each block contains a list of transactions.
- Transactions could include various types of data, not just financial transactions. For example, they could represent the transfer of assets, ownership records, or any information that needs to be securely recorded.

Chain of Blocks:

- The blocks are linked together in a chronological order, forming a chain. Each block contains a reference to the previous block, creating a continuous and unbroken chain of transactions.
- This linking mechanism ensures the integrity of the entire blockchain, as altering one block would require changing all subsequent blocks.

Generic Elements of Blockchain

Decentralization:

- Unlike traditional centralized systems, blockchain operates on a decentralized network of nodes (computers). Each node has a copy of the entire blockchain, and there is no central authority controlling the network.
- This decentralization enhances security, reduces the risk of a single point of failure, and increases transparency.

Consensus Mechanism:

- Consensus mechanisms are protocols used to achieve agreement on the state of the blockchain. They ensure that all nodes in the network have the same copy of the blockchain.
- Common consensus mechanisms include Proof of Work (used in Bitcoin), Proof of Stake, Delegated Proof of Stake, and Practical Byzantine Fault Tolerance.

Generic Elements of Blockchain

Cryptography:

- Cryptography plays a crucial role in securing transactions and maintaining the integrity of the blockchain.
- Each block contains a cryptographic hash of the previous block, creating a link between the blocks. Additionally, cryptographic techniques are used to secure transactions and control access to the blockchain.

Smart Contracts:

- Smart contracts are self-executing contracts with the terms of the agreement directly written into code. They automatically enforce and execute the terms when predefined conditions are met.
- Ethereum is a popular blockchain platform that introduced the concept of smart contracts, enabling developers to create decentralized applications (DApps).

Generic Elements of Blockchain

Immutable Ledger:

- Once a block is added to the blockchain, it is extremely difficult to alter or delete the information within it. This immutability ensures the integrity of the transaction history.
- The use of cryptographic hash functions and the consensus mechanism contribute to the immutability of the blockchain.

Public and Private Keys:

- Participants in a blockchain network have public and private cryptographic keys. The public key is visible to others and serves as an address for receiving transactions, while the private key is kept confidential and is used to sign transactions, providing proof of ownership.

Generic Elements of Blockchain

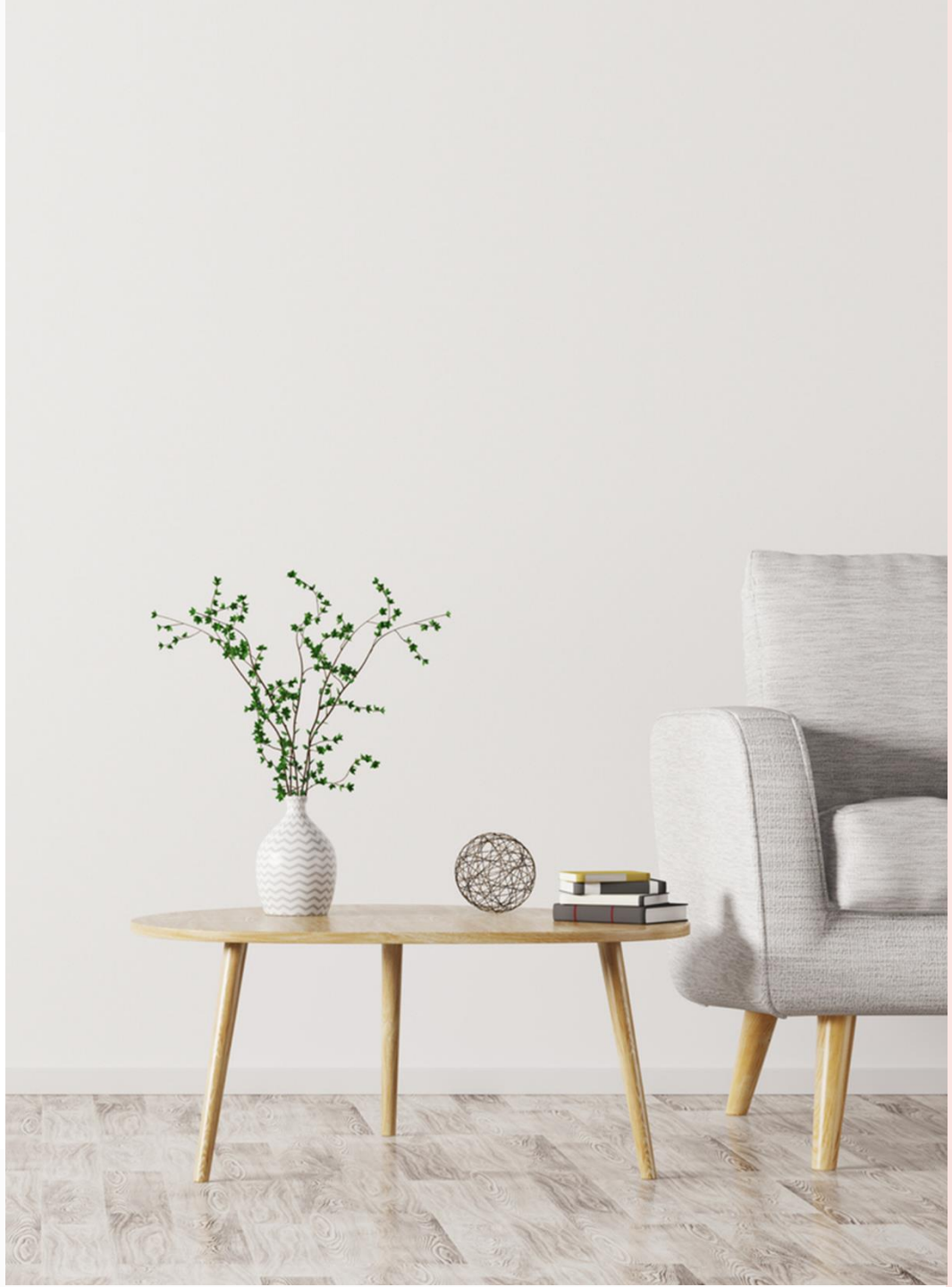
Nodes:

- Nodes are individual computers or devices participating in the blockchain network. They validate transactions, store a copy of the blockchain, and contribute to the consensus process.
- Nodes can be categorized as full nodes (store the entire blockchain) or lightweight nodes (store only part of the blockchain).

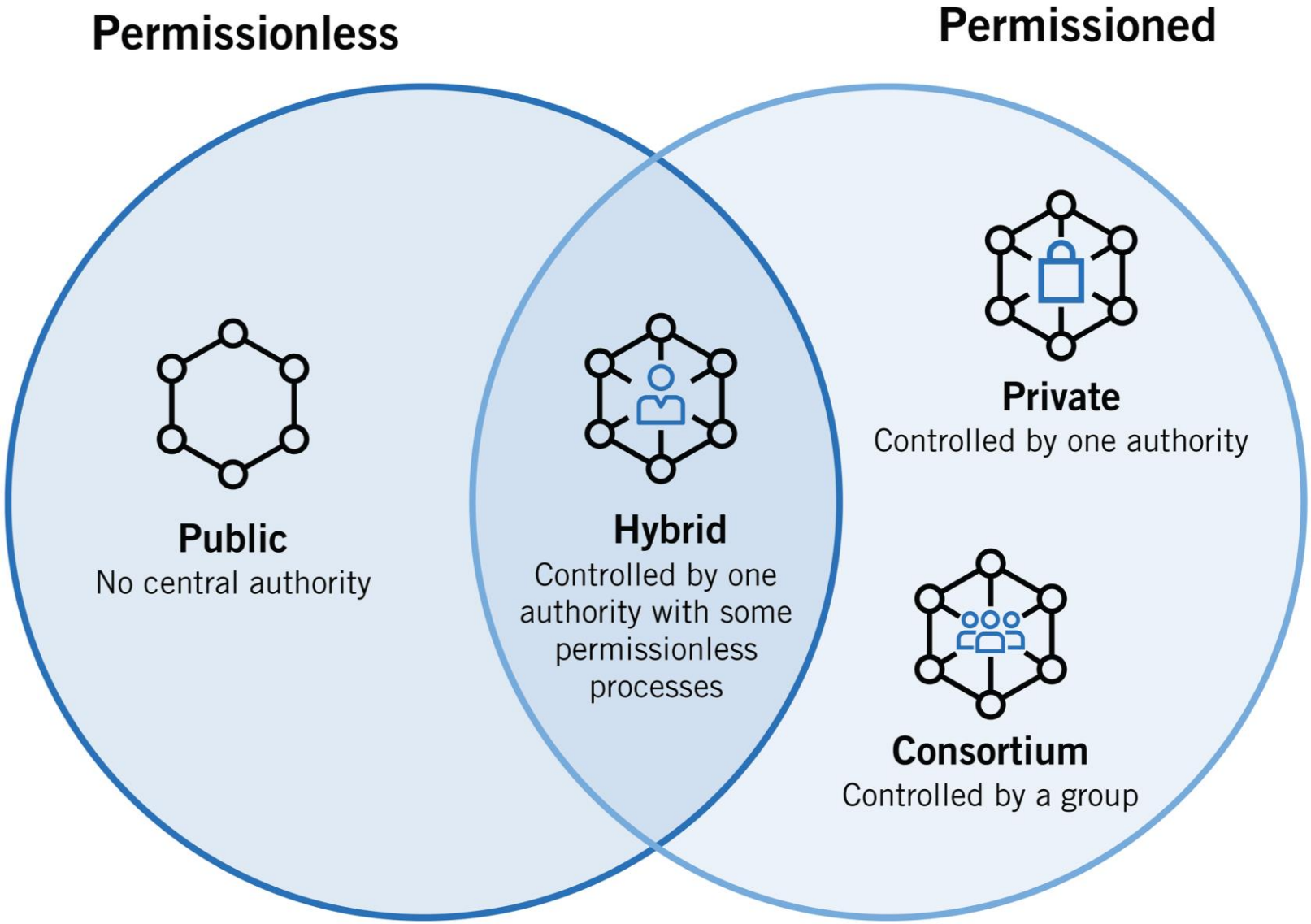
Mining (in Proof of Work systems):

- In Proof of Work blockchain networks like Bitcoin, mining is the process through which new blocks are added to the blockchain. Miners use computational power to solve complex mathematical problems, and the first to solve it gets the right to add a new block and is rewarded with newly created cryptocurrency and transaction fees.

Types of Blockchain



Types of Blockchain



Types of Blockchain

1. Public Blockchain:

Public blockchains are permissionless in nature, allow anyone to join, and are completely decentralized. Public blockchains allow all nodes of the blockchain to have equal rights to access the blockchain, create new blocks of data, and validate blocks of data.

To date, public blockchains are primarily used for exchanging and mining cryptocurrency. You may have heard of popular public blockchains such as Bitcoin, Ethereum, and Litecoin. On these public blockchains, the nodes “mine” for cryptocurrency by creating blocks for the transactions requested on the network by solving cryptographic equations. In return for this hard work, the miner nodes earn a small amount of cryptocurrency. The miners essentially act as new era bank tellers that formulate a transaction and receive (or “mine”) a fee for their efforts.

Types of Blockchain

1. Public Blockchain:

- **Definition:** Public blockchains are open networks that anyone can join and participate in. They are decentralized, permissionless, and transparent. Participants maintain anonymity, and anyone can validate transactions, participate in the consensus process, and create smart contracts.
- **Example:** Bitcoin is a prime example of a public blockchain. It allows anyone to join the network, participate in mining, and transact in a pseudonymous manner. Ethereum is another notable example, known for supporting smart contracts and decentralized applications (DApps).

Types of Blockchain

1. Public Blockchain:

- **Characteristics:**
  - Open to anyone.
  - Decentralized consensus mechanism (Proof of Work in Bitcoin, transitioning to Proof of Stake in Ethereum 2.0).
  - Anonymity for participants.
  - High security due to decentralization.

Types of Blockchain

2. Private Blockchain:

Private blockchains, which may also be referred to as managed blockchains, are permissioned blockchains controlled by a single organization. In a private blockchain, the central authority determines who can be a node. The central authority also does not necessarily grant each node with equal rights to perform functions. Private blockchains are only partially decentralized because public access to these blockchains is restricted. Some examples of private blockchains are the business-to-business virtual currency exchange network Ripple and Hyperledger, an umbrella project of open-source blockchain applications.

Both private and public blockchains have drawbacks – public blockchains tend to have longer validation times for new data than private blockchains, and private blockchains are more vulnerable to fraud and bad actors. To address these drawbacks, consortium and hybrid blockchains were developed.

Types of Blockchain

2. Private Blockchain:

- **Definition:** Private blockchains are restricted networks where only authorized participants have access. They are centralized and often used within organizations or specific groups for collaboration, efficiency, and security. Private blockchains may use different consensus mechanisms and may not require native cryptocurrencies.
- **Example:** Hyperledger Fabric is a popular example of a private blockchain. It is used for enterprise solutions and allows permissioned participants to engage in a shared, private ledger. It is often employed in business settings for supply chain management, financial transactions, and more.

Types of Blockchain

2. Private Blockchain:

- **Characteristics:**
  - Limited access, permissioned.
  - Centralized control by a designated entity or group.
  - Variable consensus mechanisms (not necessarily based on energy-intensive mining).
  - Enhanced privacy and scalability compared to public blockchains.

Types of Blockchain

3. Consortium Blockchain:

Consortium blockchains are permissioned blockchains governed by a group of organizations, rather than one entity, as in the case of the private blockchain. Consortium blockchains, therefore, enjoy more decentralization than private blockchains, resulting in higher levels of security. However, setting up consortiums can be a fraught process as it requires cooperation between a number of organizations, which presents logistical challenges as well as potential antitrust risk . Further, some members of supply chains may not have the needed technology nor the infrastructure to implement blockchain tools, and those that do may decide the upfront costs are too steep a price to pay to digitize their data and connect to other members of the supply chain.

Types of Blockchain

3. Consortium Blockchain:

- **Definition:** Consortium blockchains are a middle ground between public and private blockchains. They are semi-decentralized and involve a group of organizations collaborating to form a shared blockchain network. Consortium blockchains aim to combine the benefits of decentralization and restricted access.
- **Example:** R3 Corda is an example of a consortium blockchain. It is designed for use by financial institutions and facilitates secure and efficient transactions between parties while maintaining privacy and permissioned access. A popular set of consortium blockchain solutions for the financial services industry and beyond has been developed by the enterprise software firm R3. In the supply chain sector, CargoSmart has developed the Global Shipping Business Network Consortium, a not-for-profit blockchain consortium which aims to digitalize the shipping industry and allow maritime industry operators to work more collaboratively.

Types of Blockchain

3. Consortium Blockchain:

- **Characteristics:**
  - Restricted access, permissioned (but shared among a group of organizations).
  - Semi-decentralized control among consortium members.
  - Consensus mechanisms can be tailored to the needs of the consortium.
  - Balances privacy and transparency based on the use case.

Each type of blockchain serves specific purposes based on the desired level of decentralization, security, and accessibility. Public blockchains are suitable for open and trustless environments, private blockchains for internal organizational use, and consortium blockchains for collaborative efforts among trusted entities. The choice of the blockchain type depends on the specific requirements and use cases of the applications being developed.

Types of Blockchain

4. Hybrid Blockchain:

- **Definition:**

A hybrid blockchain combines elements of both public and private blockchains, seeking to leverage the benefits of both models. It allows for the flexibility of public blockchains while addressing the privacy and scalability concerns associated with private blockchains.

- **Characteristics:**
  - **Public and Private Components:** Hybrid blockchains typically consist of two layers - a public layer and a private layer. The public layer is accessible to anyone and provides transparency and decentralization, while the private layer is restricted to authorized participants for confidential transactions.

Types of Blockchain

- **Interoperability:** Hybrid blockchains aim for interoperability between the public and private layers, allowing seamless transfer of assets or data between them. This can be achieved through interoperability protocols or smart contracts that facilitate communication between the two layers.
- **Flexibility and Customization:** Organizations can choose which data or transactions to keep on the public or private layer based on their specific requirements. This flexibility is advantageous for applications that require a balance between transparency and confidentiality.
- **Scalability and Efficiency:** The private layer can handle high-frequency and confidential transactions, providing scalability and efficiency, while the public layer ensures decentralization and transparency.

Types of Blockchain

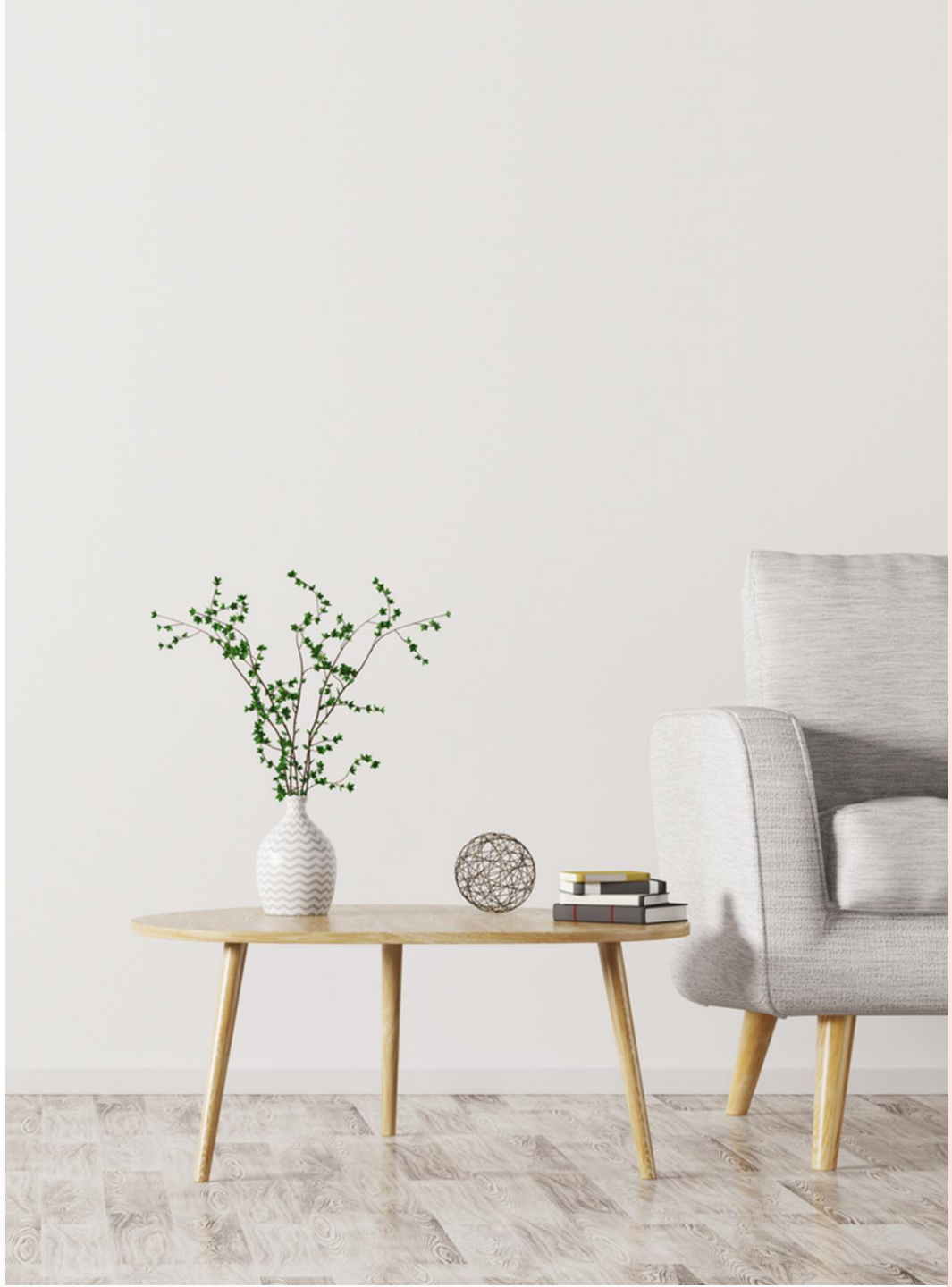
- **Use Cases:** Hybrid blockchains are often suitable for industries like finance, supply chain, or healthcare, where some data needs to be transparent (e.g., for regulatory compliance) while other data must remain private.

In summary, consortium blockchains focus on a group of known and trusted participants, offering controlled decentralization, while hybrid blockchains combine aspects of both public and private blockchains, providing a flexible solution that addresses specific needs of various industries.

Feature	Public Blockchain	Private Blockchain	Hybrid Blockchain	Consortium Blockchain
Access Control	Permissionless, open to anyone	Permissioned, restricted to approved participants	Can be permissioned or permissionless	Permissioned, restricted to members of the consortium
Decentralization	Fully decentralized	Centralized or semi-decentralized	Can be decentralized or centralized based on design	Semi-decentralized, distributed among consortium members
Governance	Decentralized	Centralized	Varies, can be decentralized or centralized	Shared governance among consortium members

Feature	Public Blockchain	Private Blockchain	Hybrid Blockchain	Consortium Blockchain
Transparency	Fully transparent	Can be private or transparent based on design	Can be transparent or private based on design	Varies, often transparent within the consortium
Speed	Slower due to larger number of nodes	Faster due to smaller number of nodes	Varies, can be optimized for speed	Faster than public, but may vary based on consortium size
Immutability	Immutable	Immutable	Immutable	Immutable
Examples	Bitcoin, Ethereum	Hyperledger Fabric, R3 Corda	Dragonchain	B3i consortium

Consensus Mechanism



1. PoW: Proof of Work

Objective: The primary goal of PoW is to secure a blockchain network by making it computationally expensive and time-consuming to add new blocks to the blockchain.

Process: Participants, known as miners, compete to solve complex mathematical puzzles. The first one to solve the puzzle gets the right to add a new block to the blockchain and is rewarded with newly created cryptocurrency coins (e.g., Bitcoin).

Mining Process:

- Miners use their computational power to attempt to find a specific value (called a nonce) that, when combined with the data in the block and hashed, produces a hash that meets certain criteria (e.g., starts with a certain number of leading zeros).
- The difficulty of these puzzles is adjustable and is designed to ensure that new blocks are added roughly every 10 minutes in the case of Bitcoin.

1. PoW: Proof of Work

Consensus Mechanism:

- The consensus is achieved through the agreement of the network on the validity of the longest chain. The longest chain is the one with the most cumulative computational work invested in it.
- Nodes on the network accept the longest valid chain as the legitimate version of the blockchain.

Security:

- The security of PoW comes from the fact that it requires a significant amount of computational power to alter past blocks. Changing the data in a block would require redoing all the work done in the subsequent blocks, making it impractical and expensive.

1. PoW: Proof of Work

51% Attack:

- A potential vulnerability is the 51% attack, where an entity controls more than half of the network's computational power. In such a scenario, the entity could potentially manipulate transactions or double-spend coins.

Examples:

- Bitcoin, the first and most well-known cryptocurrency, uses PoW.
- Litecoin, Ethereum (until it transitions to Ethereum 2.0 with Proof of Stake), and many other cryptocurrencies also use PoW.

It's worth noting that while PoW has been successful in securing blockchain networks, it has faced criticism due to its energy consumption. Critics argue that the computational power and electricity required for mining are environmentally unsustainable. This concern has led to the development and adoption of alternative consensus mechanisms like Proof of Stake (PoS) and others.

2. PoS: Proof of Stake

Proof of Stake (PoS) is a consensus mechanism used in blockchain networks to achieve agreement on the state of the blockchain. Unlike Proof of Work (PoW), which relies on computational work and energy consumption, PoS selects the creator of a new block based on the amount of cryptocurrency they hold and are willing to "stake" as collateral.

Staking:

- In a PoS system, participants (validators or forgers) are required to lock up a certain amount of cryptocurrency as collateral, or "stake," in order to be eligible to validate transactions and create new blocks.
- The more cryptocurrency a participant stakes, the higher the chance they have of being chosen to create the next block.

2. PoS: Proof of Stake

Block Creation:

- Validators take turns proposing and validating new blocks. The probability of being chosen to create a new block is typically proportional to the amount of cryptocurrency a participant has staked.

Selection Process:

- The selection process can be deterministic or pseudorandom, depending on the specific PoS algorithm. Some PoS systems use a combination of factors such as the participant's stake, the age of the staked coins, or a randomization process to select the next validator.

2. PoS: Proof of Stake

Consensus:

- Consensus is reached when a supermajority of validators agree on the validity of a block. This is achieved without the need for resource-intensive calculations like those in PoW.

Security:

- PoS is designed to be secure by making it economically unfeasible for validators to act maliciously. Validators have a financial stake in the network, and if they validate fraudulent transactions or attempt a double spend, they risk losing their staked coins.

2. PoS: Proof of Stake

Advantages:

- Energy Efficiency: PoS is generally considered more environmentally friendly than PoW because it doesn't require the massive computational power associated with mining.
- Decentralization: PoS can promote decentralization, as validators are not required to compete in solving complex puzzles. This can lead to a wider distribution of nodes across the network.

2. PoS: Proof of Stake

Challenges:

- **Nothing at Stake Problem:** This is a theoretical problem where validators might be incentivized to validate multiple conflicting chains simultaneously. Various solutions and mechanisms, such as slashing (penalizing validators for malicious behavior), are implemented to mitigate this risk.
- **Initial Distribution:** The distribution of the cryptocurrency initially can influence the level of decentralization. If a small number of entities hold a large portion of the cryptocurrency, the system may become more centralized.

2. PoS: Proof of Stake

Examples:

- Ethereum is transitioning from PoW to PoS with Ethereum 2.0.
- Cryptocurrencies like Cardano, Algorand, and Tezos use variations of PoS.

It's important to note that there are different PoS variations and consensus algorithms within the broader category of Proof of Stake, and each may have its own unique features and mechanisms.

3. PBFT: Practical Byzantine Fault Tolerance

First we need to understand what is the Byzantine Generals Problem. The Byzantine Generals' Problem is a classic challenge in distributed computing, where a group of generals must coordinate attack or retreat plans despite the presence of traitorous generals. This scenario, first formulated in a 1982 paper, represents consensus challenges in distributed systems. The problem emphasizes the need for a consensus protocol that can tolerate malicious behavior. In blockchain, where nodes need to agree despite potential malicious nodes, the Byzantine Generals' Problem is crucial. It has inspired consensus algorithms like Practical Byzantine Fault Tolerance (PBFT) to address these challenges.

3. PBFT: Practical Byzantine Fault Tolerance

1. Byzantine Fault Tolerance:

- PBFT is a type of Byzantine Fault Tolerance (BFT) algorithm, designed to handle the Byzantine Generals Problem. This problem arises when nodes in a distributed system need to agree on a common decision despite the potential presence of malicious nodes or nodes that may fail in arbitrary ways.

2. Key Goals:

- **Safety:** Ensures that all honest nodes in the network agree on the same set of transactions or state.
- **Liveness:** Ensures that the system continues to make progress, even in the presence of faults, as long as a sufficient number of nodes are honest and operational.

3. PBFT: Practical Byzantine Fault Tolerance

3. Basic Workflow:

- PBFT operates with a set of nodes, often referred to as replicas, that collectively make decisions about the state of the system.
- The algorithm proceeds through a series of communication rounds, with designated roles for nodes, including a leader and non-leader replicas.

Note: People who uses PBFT as Consensus Mechanism they refers to the Nodes as Replicas.

3. PBFT: Practical Byzantine Fault Tolerance

4. Phases of PBFT:

- **Pre-Prepare:** The leader proposes a block of transactions and sends a pre-prepare message to other replicas.
- **Prepare:** Upon receiving a pre-prepare message, each replica broadcasts a prepare message to signify that it has accepted the proposed block.
- **Commit:** Once a replica receives enough prepare messages from others, it broadcasts a commit message, indicating that it is ready to add the block to the blockchain.
- **Execute:** After receiving a sufficient number of commit messages, a replica executes the transactions in the block and broadcasts the result to the network.
- **Reply:** Replicas send replies to the client to acknowledge the completion of the consensus process.

3. PBFT: Practical Byzantine Fault Tolerance

5. Quorum:

- PBFT relies on the concept of quorums, which are subsets of the nodes in the network. For each phase (pre-prepare, prepare, commit), nodes must receive messages from a two-thirds majority of the network to progress to the next phase.

6. Fault Tolerance:

- PBFT is designed to tolerate up to one-third of the nodes behaving maliciously or failing. As long as more than two-thirds of the nodes are honest and operational, the algorithm guarantees safety and liveness.

3. PBFT: Practical Byzantine Fault Tolerance

7. Advantages:

- **Efficiency:** PBFT is known for its efficiency in terms of transaction throughput and low latency.
- **Finality:** Once a block is committed, it is considered final, providing a high level of security.

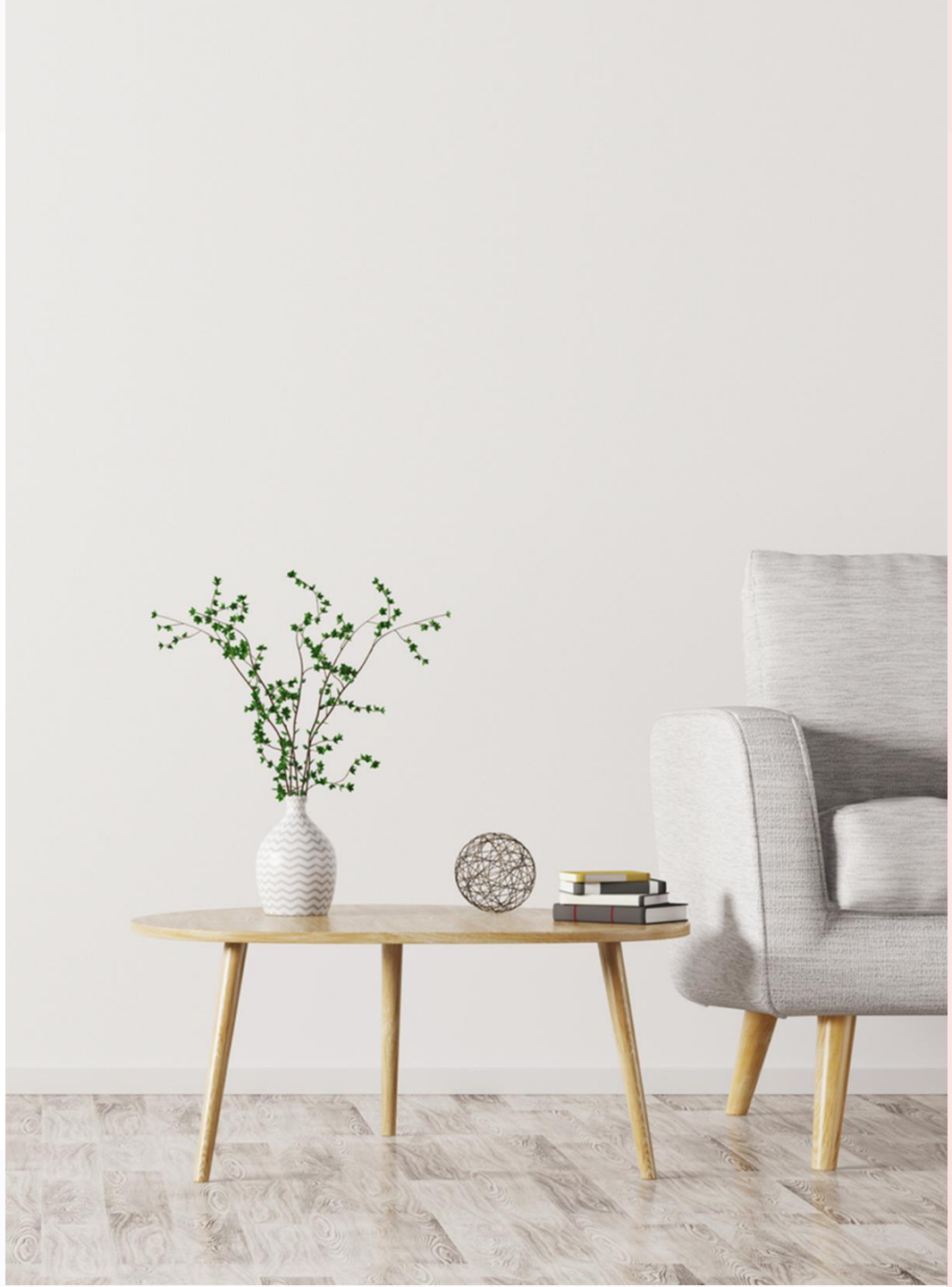
8. Use Cases:

- PBFT is commonly used in permissioned blockchain networks, where the identity of participants is known and trusted, such as in enterprise blockchain solutions.

Conclusion:

Practical Byzantine Fault Tolerance (PBFT) is a robust consensus algorithm designed to ensure agreement in distributed systems, even when faced with malicious or faulty nodes. It has been influential in the development of consensus mechanisms for permissioned blockchains, providing a practical solution to the challenges posed by Byzantine faults.

Creating Blocks & Links, Inserting Hashes

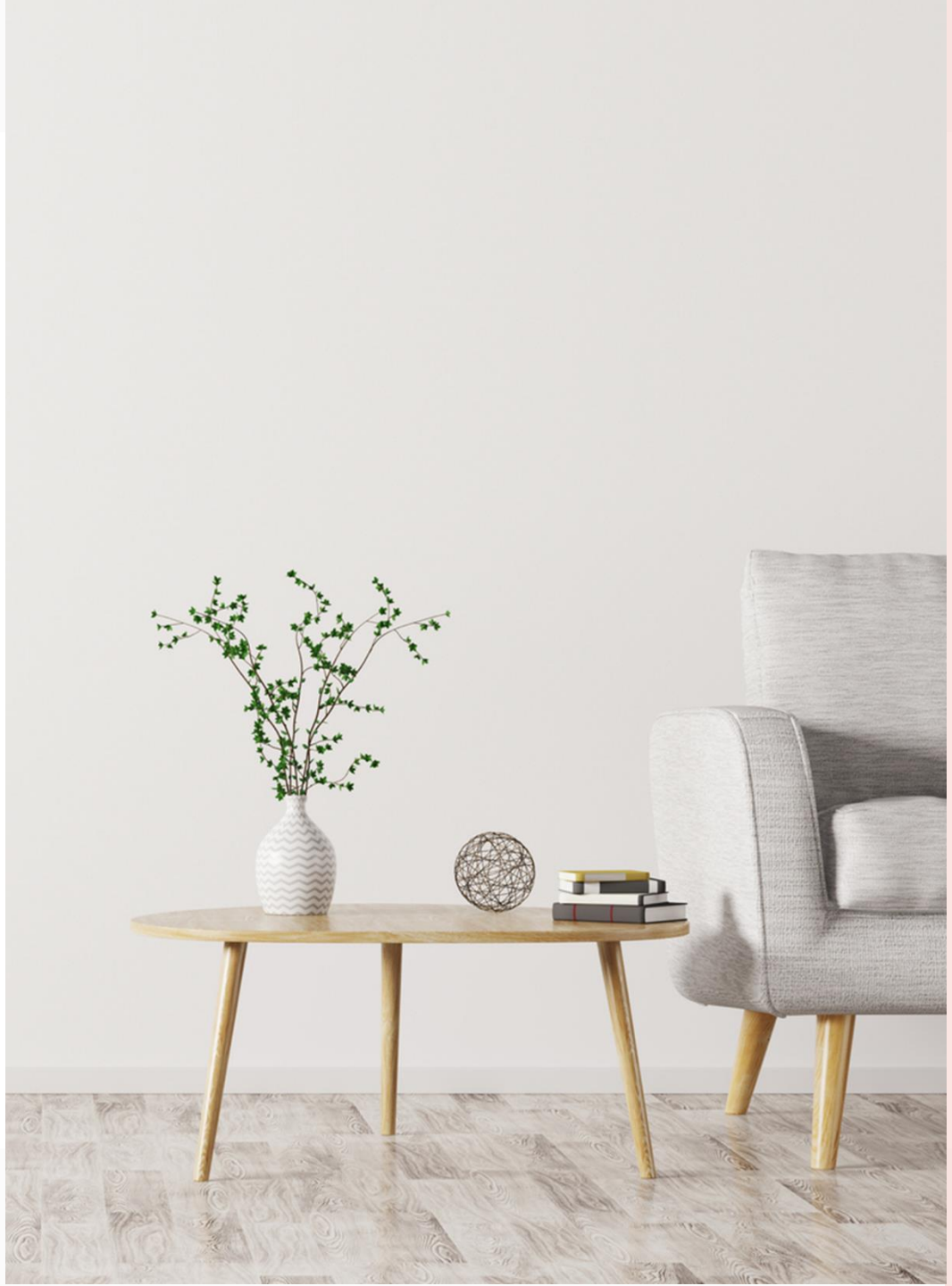


Creating Blocks & Links, Inserting Hashes

https://tools.superdatascience.com/blockchain/hash/

Above link is for representation and visualization purpose only.

Forking in Blockchain



Forking in Blockchain

Forking in blockchain refers to the divergence in the blockchain's protocol, resulting in two separate chains with a shared history up to a certain point. Forks can occur for various reasons, including updates, changes in consensus rules, and disagreements within the community.

There are two main types of forks: Soft Forks & Hard Forks.

Forking in Blockchain

1. Soft Forks:

- **Definition:** A soft fork is a backward-compatible upgrade to the blockchain protocol. It tightens the rules of the blockchain, making previously valid blocks invalid. Despite being a form of upgrade, it allows nodes that have not upgraded to still interact with the upgraded nodes without causing a complete split in the network.
- **Implementation:** Soft forks are implemented by introducing new rules that are more restrictive than the existing ones. Nodes that have not upgraded will still consider the new blocks valid, as they adhere to the old rules as well.
- **Reversibility:** Since soft forks are backward-compatible, they are reversible. If a majority of the network decides to revert to the old rules, the fork can be undone.

Forking in Blockchain

2. Hard Forks:

- **Definition:** A hard fork is a non-backward-compatible upgrade to the blockchain protocol. It introduces new rules that are incompatible with the old ones, leading to a permanent divergence in the blockchain. Nodes that have not upgraded will not be able to validate or relay new blocks created by nodes that have upgraded.
- **Implementation:** Hard forks are typically implemented by introducing new features, changes, or rules that are not compatible with the previous protocol. This can include changes to the consensus algorithm, block size, or other fundamental aspects of the blockchain.
- **Irreversibility:** Hard forks are irreversible. Once the network splits, the two chains continue independently. Participants need to choose which chain to follow based on their agreement with the new rules.

Forking in Blockchain

3. Contentious Forks:

- **Definition:** A contentious fork occurs when there is a significant disagreement within the community regarding the proposed changes. It can lead to a permanent split in the network if both factions continue to support their respective chains.
- **Impact:** Contentious forks often result in the creation of two separate cryptocurrencies, each with its own set of rules and community support.

Forking in Blockchain

4. Non-Contentious Forks:

- **Definition:** A non-contentious fork happens when the majority of the community agrees on the proposed changes, and there is a smooth transition from the old protocol to the new one.
- **Impact:** Non-contentious forks are less likely to result in a permanent split, as the majority of the network adopts the upgrade, and the old chain loses support over time.

In summary, forking in blockchain can be either a soft fork or a hard fork, depending on the backward compatibility of the changes. Additionally, forks can be classified as contentious or non-contentious based on the level of agreement within the community.

Thank You

Maitri Hingu

mkhingu@vnsgu.ac.in