CHAPTER 2

Blockchain Types and Consensus Mechanism

LEARNING OBJECTIVES

This chapter covers the different types of Blockchain and provides a brief on the consensus algorithms in a decentralized network architecture where anyone can be a node and decisions are collectively reached. The consensus mechanism enables the trustless nature of Blockchain. Consensus algorithms form the governance mechanism that prevents malicious actors from tampering with the data or record-sets. Transactions are entered only with the consensus of all the relevant parties after they validate all relevant data.

2.1 INTRODUCTION

As described in the previous chapter on fundamentals, blockchain is a **Digital Ledger** of information that is:

1) **Distributed** – Every participating node has a digital copy of the blockchain database, whereby all can contribute to processing the blockchain.
2) Maintained by **Consensus** – The consensus algorithms are the governance mechanisms to guarantee that the data/records are legitimate and not tampered with. Transactions can be entered only with the agreement of all the relevant parties, on validation of the data.
   Hence, it is ensured that the records are
3) **Immutable** – Once consensus is reached on the validity of a transaction/data and recorded on the blockchain, it cannot be changed or deleted. Any subsequent changes are recorded in a new transaction block (refer Fig. 2.1);
   and
4) **Auditable** – The immutable tracking of a record with timestamp allows for the provenance of the asset at every step.

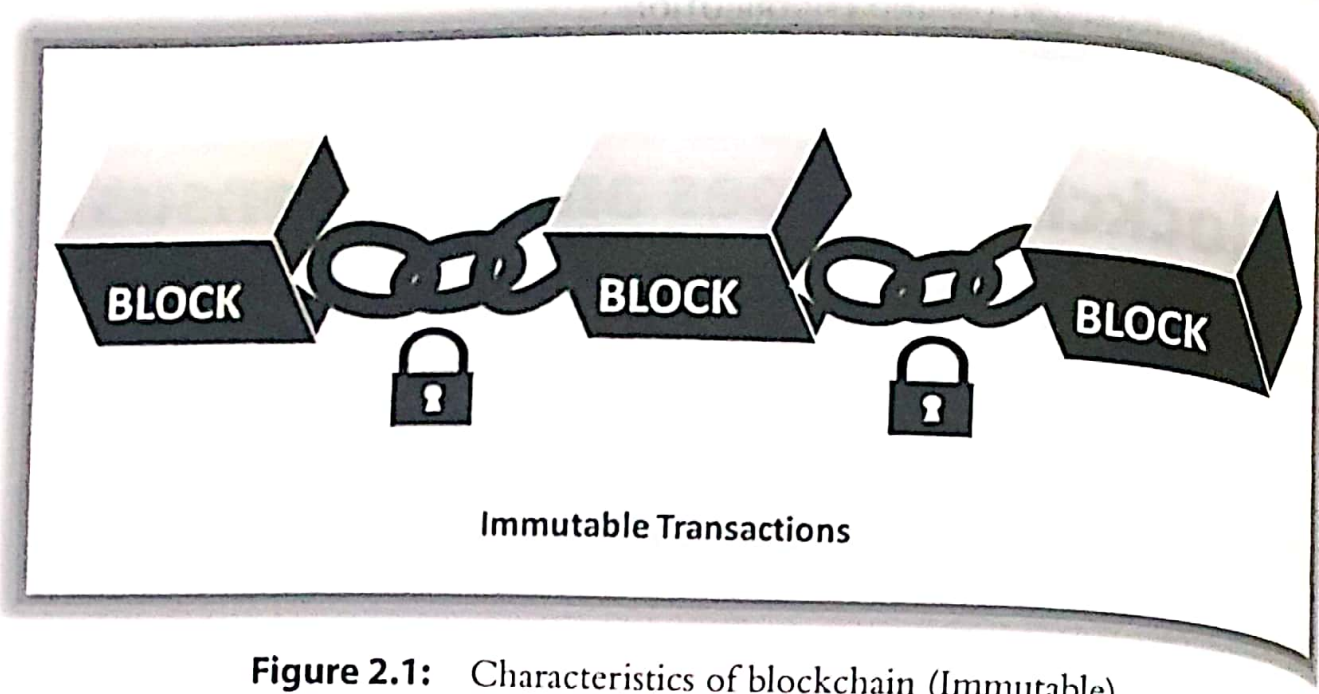


Figure 2.1: Characteristics of blockchain (Immutable)

With blockchain, the dependency of "trust" on third-party or intermediaries (like banks, brokers, lawyers) is made redundant by the direct peer-to-peer handshake within the network of nodes. The following characteristics enable the **Trustless** nature of the Blockchain:

a) **Decentralization** – The data is digital and decentralized, i.e., it can be shared across a network of computers or servers without the need for a central authority for making decisions.
b) **Transparency** – Data is transparent and open to everyone in the P2P network, i.e., anyone can view the transactions.
c) **Privacy** – User identity is kept private by robust cryptography.
d) **Security** – Transactions are cryptographically secure using hash algorithms.

However, even with its rich set of features, it cannot be plugged into any and every industry in its vanilla form. Some industries may consider the full transparency of data to be a security risk, especially for government and research organizations. Other industries find the accessibility to the network a source of concern.

To circumvent these business concerns, technologists have attempted to find ways to strike a balance between the organizational risk appetite and the blockchain's capability. The two main blockchain capabilities taken into consideration to enable fit for service are its decentralization and consensus mechanism. This chapter will explore the different types of Blockchain based on the kind of access and the various consensus mechanisms.

2.2 Decentralization and Distribution

2.2.1 Decentralization

One of the main characteristics of blockchain technology is its decentralization, where transactions are not under the control of any single party.

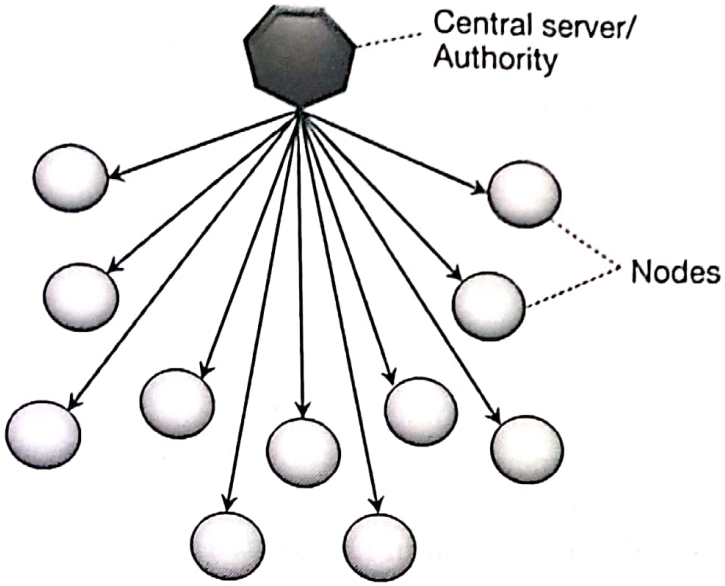


Figure 2.2: Centralized network

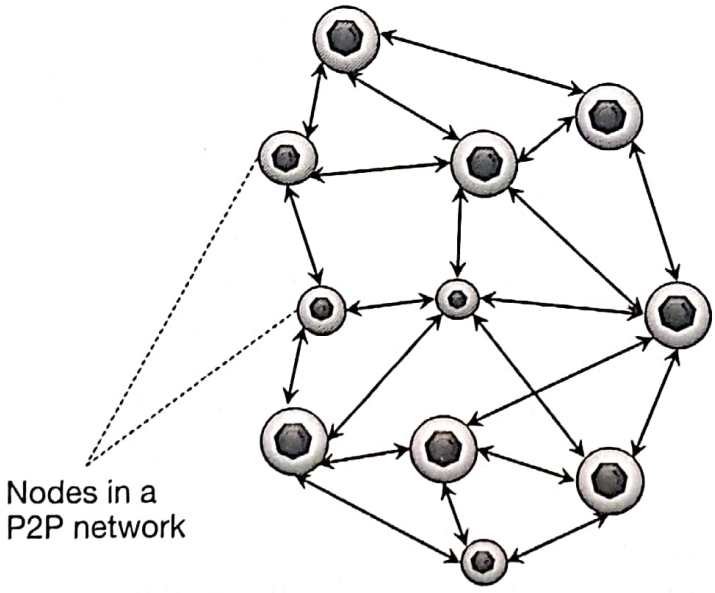


Figure 2.3: Decentralized network

In centralized systems (refer Fig. 2.2), only a central authority or administrator has the power to maintain and update the database. Thus, data flows are controlled and managed by the central authority. The central authority maintains the database by defining the rules and procedures users can have on adding, deleting, or updating the data. All the nodes (computers and devices) connected to the network are subject to access granted by the central authority. In simple words, the central authority makes all the decisions. A failure at the centre means the collapse of the entire system.

However, a decentralized system does not rely on any single authority and is self-regulated (refer Fig. 2.3). Blockchain technology uses a decentralized P2P network architecture wherein anyone can be a node. Every node is equal in the hierarchy with equal access to maintain the database.

While in a centralized environment, an organization such as a bank holds the sole right to read, write or send transactions, in the decentralized environment, anyone can access or write into the ledger. This inclusivity ensures that

a) no single entity has sole control of the network
b) there is no single infrastructural point of failure,
c) there is a collective agreement on the state of the system via consensus, which is explained later in this chapter.

The decentralization that is inherent in Distributed Ledger Technology is the core of Blockchain Technology.

2.2.2 Distributed Ledger Technology

In blockchain technology, a distributed ledger is a decentralized ledger of all the information that is recorded on the blockchain by consensus. One can imagine it to be a database similar to an accounting ledger where financial transactions are recorded, except that it is not restricted to financial data. Here, a transaction refers to any digital data including text, picture, or audio files. Unlike the typical database that is either centrally located in one server or spread out among several select servers, this database is distributed to all parties and locations. It can be accessed by every single member of the blockchain network, thus ensuring incorruptibility as malicious changes cannot be made when everyone has simultaneous access to all records.

In other words, a distributed ledger technology or DLT is defined as a decentralized database that can securely record and share financial, physical, or electronic assets across a geography agnostic network through transparent updates of information.

However, it should be noted that a decentralized system does not necessarily mean a fully distributed one. Figure 2.4 represents a decentralized network where the processing work or control is shared with sub-nodes.

This decentralization is different from the decentralized network of blockchain, where the work is shared between all the nodes (refer Fig. 2.3). In DLT of blockchain, a copy of the database is available on every computer or device of the users in the network as depicted in Fig. 2.5. The database is independent of a central authority and any changes or entries to the ledger have to be "agreed" upon by all users/nodes/parties. This agreement mechanism is referred to as the consensus mechanism.

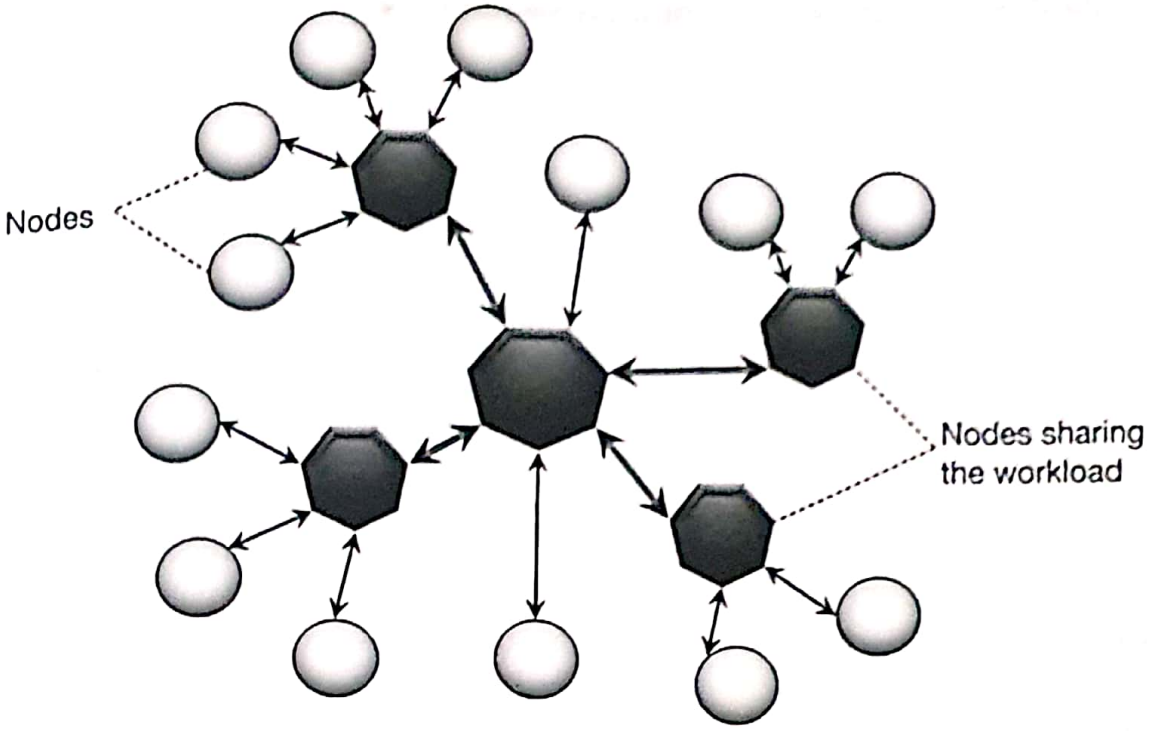


Figure 2.4: Decentralized network

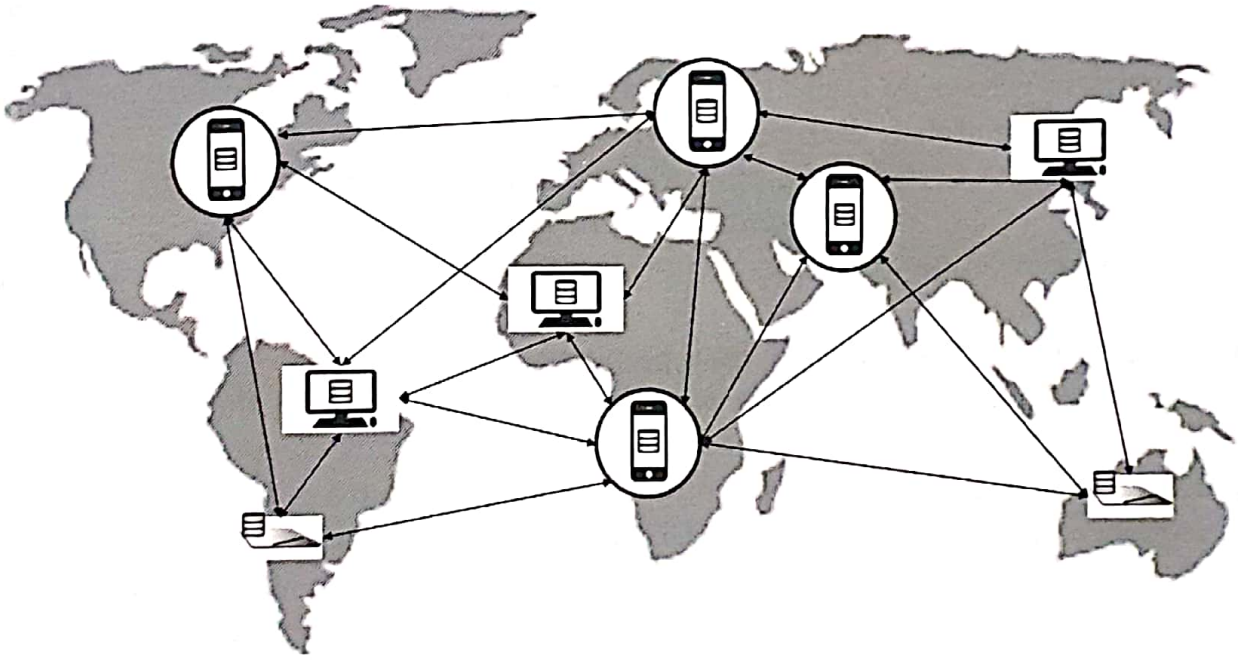


Figure 2.5: Decentralized distributed network (Blockchain)

Once consensus is reached, the database is updated to all the nodes in the network. Thus, at any given time, information is synchronized across all the nodes. Hence, it is also referred to as distributed consensus.

2.2.2.1 PAXOS Consensus in Distributed Systems

"A distributed system is one in which the failure of a computer you didn't even know existed can render your computer unusable."

– Leslie Lamport

Distributed systems are built for high availability and scalability involving a group of computers or a set of distinct processes working together to accomplish a common objective. Emails, web browsers, and many other mainstream software such as Netflix Eureka and Apache Zookeeper, all use distributed system algorithms. Some of the challenges faced in distributed systems are

a) **Clock drift**: Need for complete universal and ordered information to ensure that the message or data being transmitted in consistent and up-to-date.
b) **Concurrency**: Maintaining consistency and avoiding conflicts when multiple operations are taking place on the same data object.
c) **Message transmission**: Ensuring coordinated and in-time communication between computers to avoid duplication and delayed data or messages.
d) **Component failure**: Ensuring that breakage in one node/machine does not impact the function of another node or network.

The above challenges in making the network fault-tolerant can, in most instances, be addressed by implementing consensus algorithms. PAXOS was the first real-world fault-tolerant consensus algorithm introduced by Lynch and Liskov in the 1990s and later mathematically proven by Leslie Lamport and used by internet companies like Google and Amazon to build their distributed services.

The primary PAXOS mechanism works under the principle that if the majority of the nodes agree on a value, then consensus is reached. It has three roles:

1) **Proposer**: A proposer receives client requests called 'values' and sends these proposed values to acceptors.
2) **Acceptor**: Receives messages from proposers and learners. They view the proposed values and inform the proposer whether they accept or reject the proposed value. They also inform the proposer if another value was already accepted.
3) **Learner**: Listens to all the acceptor's decisions and delivers values in an ordered sequence. If any gap is found, the learner should contact the acceptors and repeat the decision. For example, say learners noted IDs 1 to 6, and the next instance delivered is value 8. The learner reverts to acceptors to repeat the procedure.

In practice, a node or server can function in all three roles. There are two phases (refer Fig. 2.6) in the underlying PAXOS protocol.

Phase I: A proposer prepares a unique number **N** and gets acceptors to accept the proposed number **N,** i.e., get their promise to accept the values within a set timeout period. If any acceptor has previously accepted a value, he or she should inform the proposer of the already accepted proposal number and value.

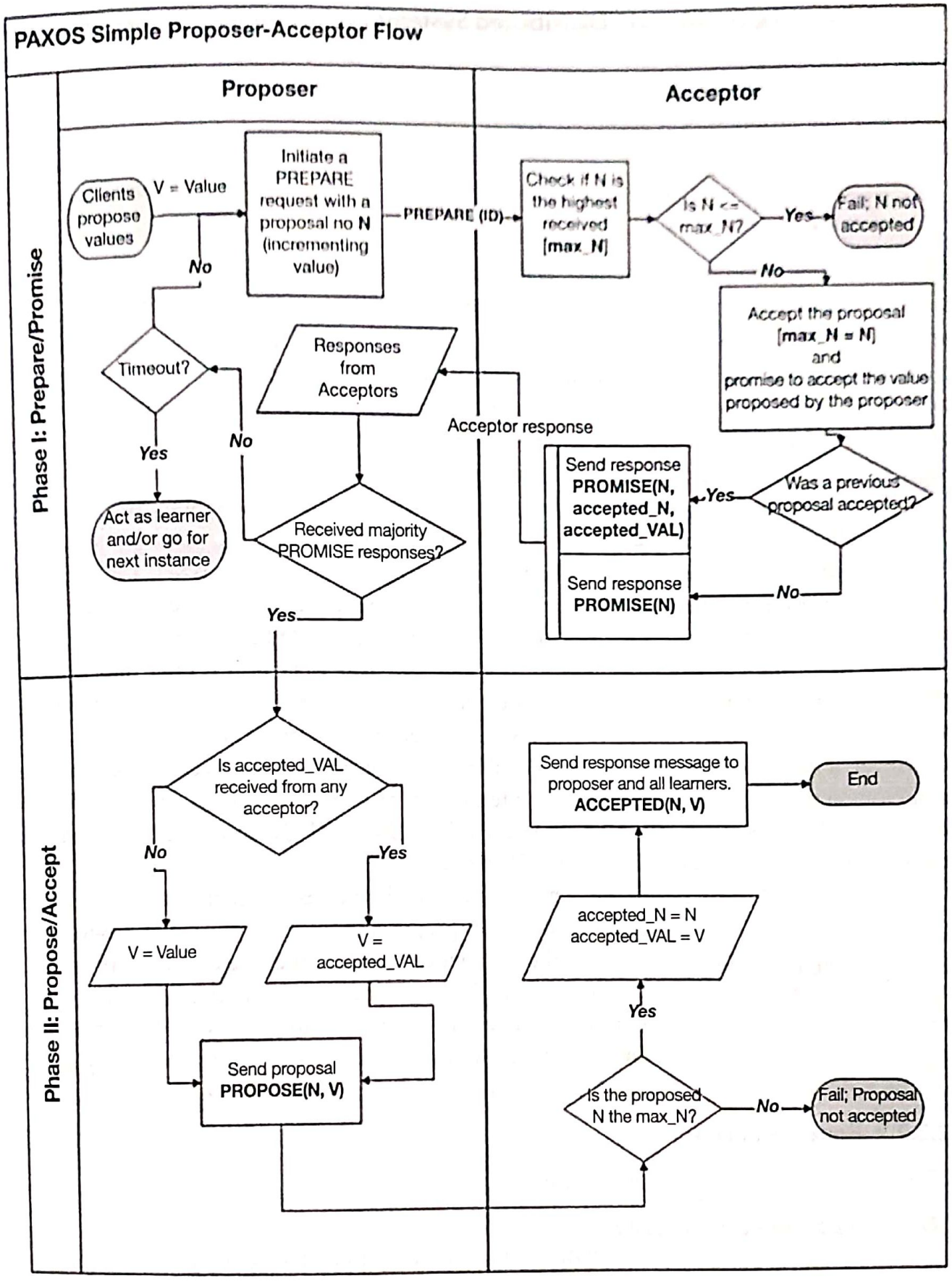


Figure 2.6: PAXOS simple proposer–acceptor flow

The acceptors will accept **N** only if it is higher than the proposal number value they have stored, if any. If it is less or equal to the stored number, they respond with a fail message or not respond at all.

Once the proposal **N** is accepted, the acceptor:

a) does not or cannot accept a proposal less than **N**; **N** is the new proposal number.
b) has to respond with **N** and the proposal with the highest number less than **N** that the acceptor has accepted.

Phase II: Here, the proposers check if they got the majority vote, i.e., whether they can use their proposal or whether they have to use the highest-numbered one received from among all the responses.

The proposer will then send the acceptance request, with a time-out, which the acceptor has to accept/commit if the value is the same as the previously accepted proposal, and the number is the highest sequence number agreed to. If majority of acceptors have ACCEPTED the number and value, it means that consensus is reached on the value.

The acceptors may fail to respond within the timeout period, in which case consensus is not reached, and the next proposer will continue the cycle.

The intricate design of PAXOS allows for the acceptance of values when a majority of nodes agree even if the rest of the nodes deny or ignore a proposed value. This intricacy made PAXOS challenging to understand and implement, paving the way for **RAFT** (refer Chapter 6), a simpler consensus algorithm where the leader election is built directly into the algorithm, making it a less complicated mechanism. However, PAXOS and RAFT address only the fundamental crash failures, i.e., when the component(s) in a network fails. However, it does not solve the Byzantine Fault tolerance problem (refer Section 2.4.1) where malicious actors (nodes) could choose to alter, block, or stop the messages altogether. Thus came the DLS (Depth Limited Search) and pBFT (Practical Byzantine Fault Tolerance) algorithms for systems that could exhibit Byzantine behaviour.

PAXOS and RAFT paved the way for distributed consensus algorithms that are used in cryptocurrencies. For bitcoin, Nakamoto proposed a combination of peer-to-peer gossip (gossip protocol), safety probability (longer the chain, lower the chance of malicious attack), Sybil attack resistance (proof-of-work algorithm), and incentives (block rewards for nodes to use their resources to compute complex puzzles) to address the byzantine and other faults inherent to distributed systems. Many consensus algorithms are at play based on the type of blockchain involved. These are explained in Section 2.4.

2.2.2.2 Benefits of DLT

The benefits of a distributed ledger are:

1) Transparency and security

As the data is shared and visible to all the nodes, it is quite difficult to make any unauthorized changes. Every participating node maintains a copy of the ledger, thus preventing a single

point of failure. Any entry has to be consensually agreed upon by all parties making the distributed ledger secure and almost hack-proof.

2) Decentralization

Every owner or node has control over his/her data, unlike a centrally managed system, where a corporate or central authority has sole control. A centrally managed system can lead to a single point of failure. Decentralization gives the users more decision-making power over what goes into their data record. The consensus protocols help to unanimously and securely agree on what should or should not be added to the distributed ledger and bring about an inherent trust within the system.

3) Speed and efficiency

In today's world, the validation and reconciliation of information between various disparate systems are mostly manual, involving time-consuming procedures and prone to errors. With the trustless and distributed nature of blockchain's DLT, the administrative effort of capturing, validating, and synchronizing individual sets of information by staff and intermediaries can be eliminated. This practically diminishes the chances of human error and improves operational efficiencies.

4) Cost savings

Intermediaries such as banks, brokers, lawyers, and administrative staff are required to foster trust between parties. However, this also leads to delays and costs. It goes without saying that with disintermediated systems, companies can save on bottom-line costs while also realizing near-time transactions and efficiencies.

Blockchain uses the underlying technology of DLT that supports immutable transactions on a peer-to-peer (P2P) network without any centralized coordinating entity. Blockchain systems maintain a distributed database in a decentralized manner using cryptographic signatures and consensus-based validation procedures, making it secure, frictionless, and indisputable.

2.3 Types of Blockchain

2.3.1 Accessibility and Permissions

Organizations, including blockchain protagonists, ask the quintessential question – "Are the people really whom they say they are AND is there a risk in granting them rights to the system?"

A node is any computer or hardware device connected to the blockchain network via the internet. There are people behind these devices. Hence a node is also referred to as a member or actor or participant or party or user in the blockchain network. Blockchains are categorized based on user identity/authentication and user rights/authorization.

1) Public and Private Blockchain

A blockchain is considered to be either public or private, based on whether the network is open or accessible to anyone with an internet connection. In a **public** blockchain, anyone can join the network. They can download a copy of the ledger and initiate, broadcast, or mine blocks. Users are anonymous (refer Fig. 2.7).

In a **private** blockchain, membership or association with the blockchain is restricted. One has to meet certain pre-requisite conditions to be a part of the blockchain network. Users are not anonymous.

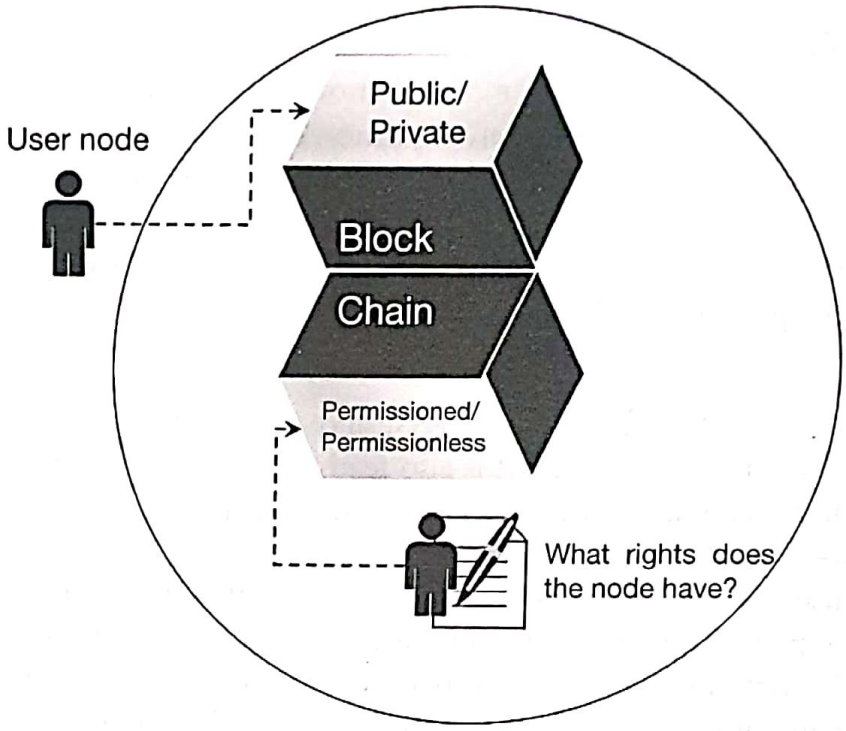


Figure 2.7 Blockchain user anonymity

2) Permissionless and Permissioned Blockchain

This category is based on the type of rights the user or node has within the blockchain network. The rights, if any, are defined by a central entity or group of entities. A Blockchain is considered **permissionless** if no such control entity exists, and all the nodes have equal rights to the network, i.e., they can all read, receive and send transactions and participate in the consensus mechanism.

In a **permissioned** blockchain, the central entity or group restricts the roles that the nodes can play. It can vary from nodes having rights to only initiate transactions to those who validate transactions and to still others that deploy or execute smart contracts. In other words, only selected nodes will participate in the consensus mechanism for permissioned blockchain. In contrast, in a permissionless blockchain, all or majority nodes in the network need to agree on the validity of a record collectively.

Based on the authentication and authorization privileges, Blockchain can be classified as below:

a) Public Blockchain (or Public Permissionless Blockchain)
b) Private Blockchain (or Private Permissioned Blockchain)
c) Consortium Blockchain (or Public/Private Permissioned Blockchain)
d) Hybrid Blockchain (interconnected Public-Private Blockchain)

2.3.2 Public Blockchain

The **public permissionless** blockchain is synonymous with a **public** blockchain. Bitcoin, the first blockchain created, is a public permissionless blockchain.

In a public blockchain, anyone in the world can access the blockchain, download a copy of the code, and run a node. It is a fully decentralized distributed network. One does not need any permission to read/access a transaction, initiate a transaction, or participate in the consensus process (PoW) to create a block. Participants or nodes remain anonymous through high cryptographic protocols. Anonymity, transparency, and immutability are valued over efficiency.

The distinct features of a public blockchain (refer Fig. 2.8) are:

- It is open to the public, as the name suggests. Anyone can join the network and be a participant/node.
- No permissions are required for anyone to read/send transactions
- The standard consensus algorithm used is Proof-of-Work (PoW), where nodes (miners) solve the hash puzzle and submit their resultant block to the rest of the network participants for consensus.

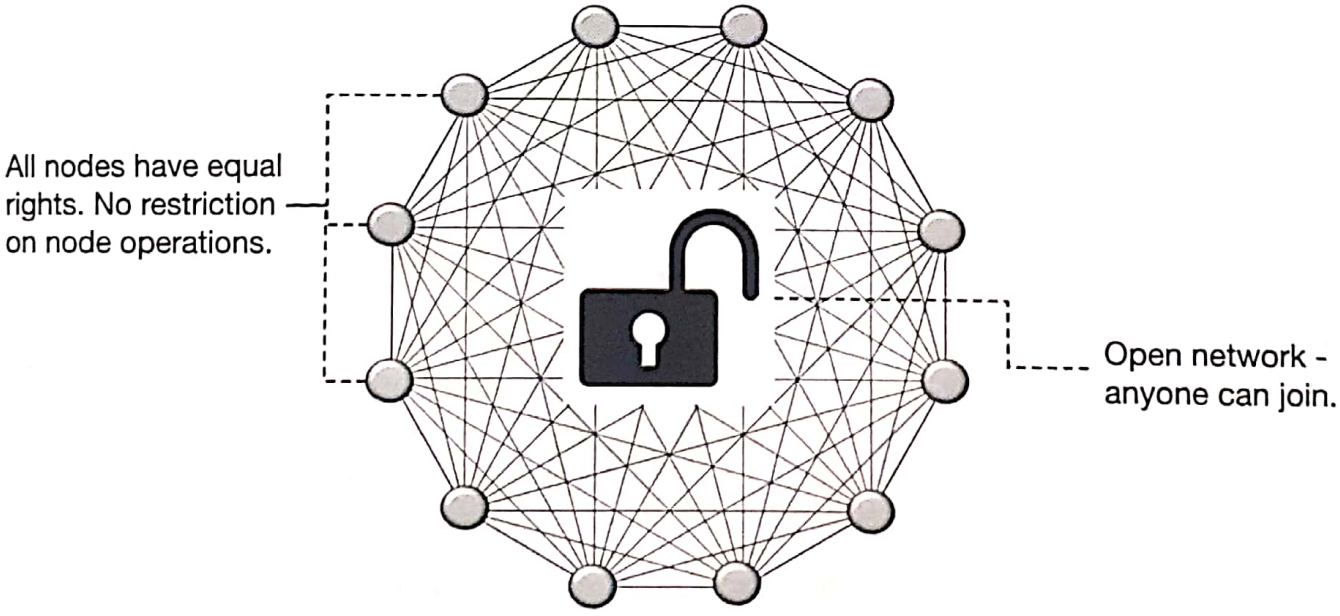


Figure 2.8: Public blockchain

- There is 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.

The downside of a public blockchain is its poor scalability:

- It has low transaction processing speed – ten minutes to create a block.
- Consensus mechanism requires an immense amount of energy and computational power.
- Participants with supercomputers or more powerful ASICs have a better chance of mining than the others, hence the risk of decentralization with mining pools (refer 51% risk in Section 2.4.3.1).

Bitcoin, Litecoin, Ethereum are the most common examples of a public blockchain.

2.3.3 Private Blockchain

The **private permissioned** blockchain, also known as the **private** blockchain (refer Fig. 2.9), differs from the public blockchain in its accessibility and permission. The network is not open to everyone. It leverages the blockchain features of distributed database, immutability, and security. However, the innovative blockchain feature of decentralization and openness is lost as all the permissions are controlled by a few nodes in the organization. Blockchain technology was invented to remove the power of central authority. So in the case of a private blockchain network or organization where the owner has sole control over who can read, write, and validate data, it stands to reason why many may not consider it to be a real blockchain. In public blockchain, efficiency and immutability take precedence over anonymity and transparency.

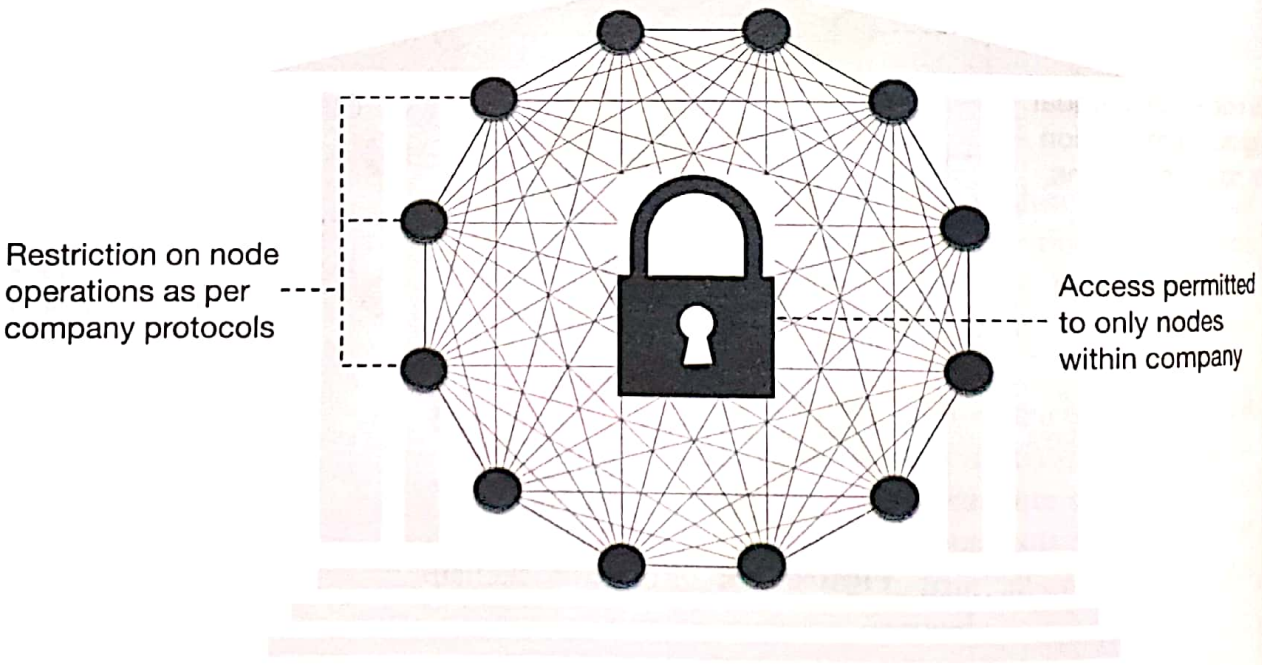


Figure 2.9: Private blockchain

The key features of a private blockchain are:

- It is not open to the public. However, participants are known to each other and hence, trust is assured.
- All participants are pre-approved by the organization. The ledger and access therein are distributed within the network of participants.
- The owner or central authority controls the permission to read, write, or audit the ledger.
- The central authority controls the consensus process.
- High cryptographic methods secure the data.
- It has high transaction processing speed, taking only seconds to create a block.
- Very low energy consumption as supercomputers are not required for processing.

Challenges of a private blockchain:

- They are not decentralized. The trade-off is better scalability and security.
- Blockchain is supposed to function in a trustless environment. If nodes are trusted, it may be cheaper to go with a traditional database
- A central authority means a single point or point of failure, unlike the public blockchain, where there is zero downtime.
- Not considered a legitimate blockchain as it is permissioned, and there is the inherent skepticism on the immutability and trust of transactions, if controlled by a singular authority.
- The organization must agree on who has the highest power to be the central authority.

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.

Examples of private blockchain are Multichain and Monax.

2.3.4 Consortium Blockchain

The **consortium** blockchain (refer Fig. 2.10), also known as the **federated** blockchain, is a permissioned blockchain and considered to be a hybrid between public and private blockchain. It is a distributed ledger that anyone can download and access. It has the security features inherent to public blockchains while also maintaining a fair amount of control over the network. Unlike private blockchain, the consensus process is not controlled by one company but by a predetermined consortium of companies or representative individuals. Only the predetermined group has the right to take part in the validation process to create a block. For example, in a supply chain user case, the consortium may be the importer, the exporter, the shipping company, the customs, the participating banks, and inspectors.

Based on the type of enterprise, there may be an overlay of smart contracts and/or other protocols in place that restrict user access. This type of blockchain is best used in government applications.

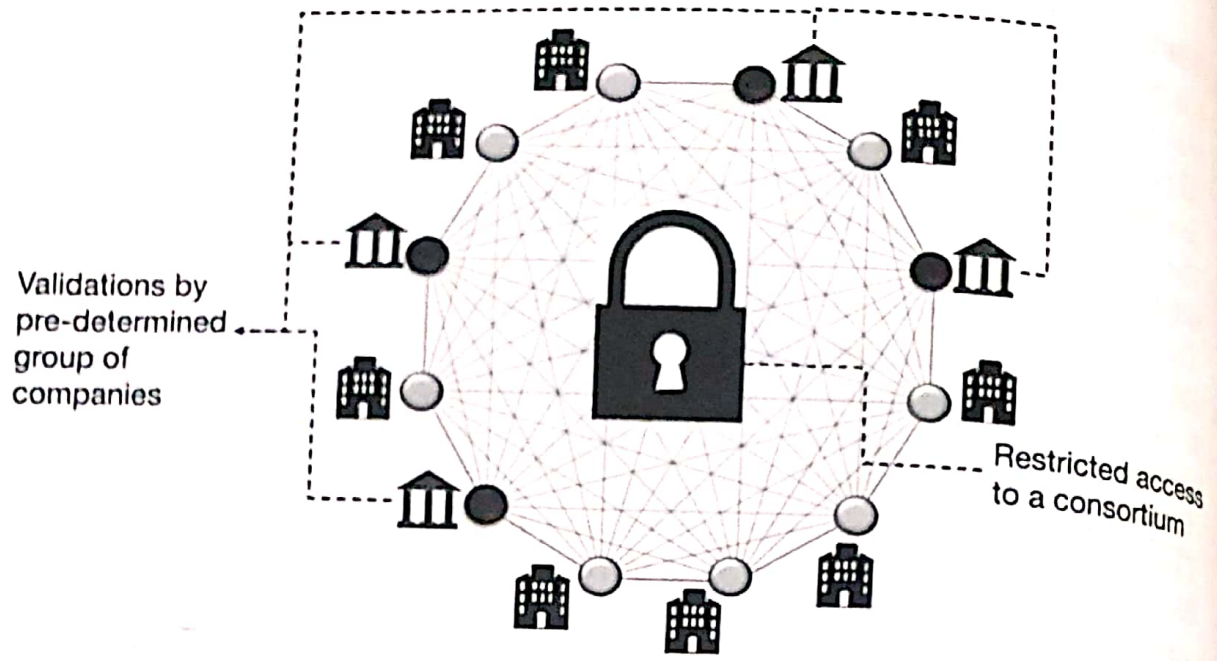


Figure 2.10: Consortium blockchain

> *"So far there has been little emphasis on the distinction between consortium blockchains and fully private blockchains, although it is essential. The former provides a hybrid between the 'low-trust' provided by public blockchains and the 'single highly-trusted entity' model of private blockchains, whereas the latter can be more accurately described as a traditional centralized system with a degree of cryptographic auditability attached."*
>
> – Vitalik Buterin on the consortium blockchain

The key features of a consortium blockchain are:

- Any member node can initiate and receive transactions. However, permission to write or audit the ledger is determined by a group of pre-approved individuals or organizations (consortium).
- The consensus process is done by a group of pre-approved nodes that have full access to the ledger.
- High cryptographic methods secure data.
- They are faster with higher scalability as compared to a public blockchain.
- They have better transaction privacy and traceability.

Challenges of a consortium blockchain:

- They are not fully decentralized. The trade-off is better scalability and security.
- Different organizations have different requirements. Agreement on a standard set of rules may get challenging.

Examples of consortium blockchain platforms are R3 Corda and Hyperledger Fabric.

2.3.5 Hybrid Blockchain

The public blockchain is fully decentralized, tamper-proof, anonymous, transparent, immutable, and open to the public. These features are achieved at the cost of low throughput, poor scalability, expensive hardware, and significant energy consumption. The private blockchain boasts of higher speeds, lower costs, and better scalability while being criticized for its centralization and restricted access. The **hybrid blockchain** (refer Fig. 2.11), as the name suggests, incorporates the best practices of both models. It takes the benefits of both the public and private blockchain, thus attempting to neutralize the negatives.

Hybrid blockchain is best suited for highly regulated enterprises or government organizations that require control over what data is kept private and what can be shared with the public. The hybrid blockchain consists of a public blockchain and a private network that is restricted to only those nodes that are invited by a centralized body. The private network(s) generate the hashed data blocks that are then shared with the public network without compromising the privacy of data. The public blockchain does the verification and time-stamping. Hence, it can be considered as a private network(s) sitting within the main public blockchain. The private network does the creation of transactions, thus maintaining the privacy of data while the public blockchain stores and verifies the blocks, ensuring disintermediation.

Privacy is a concern in public blockchain as all the data are visible to everyone. This risk is mitigated in the hybrid blockchain. If members do not want their transaction data to be accessible without their permission, they can assign exclusive rights for view or modification, in which case it goes to the different members for acceptance and consensus.

The public network uses Delegated Proof-of-Stake (refer Section 2.4.3.4) as its consensus protocol. However, the private network can use any consensus protocol of its choice.

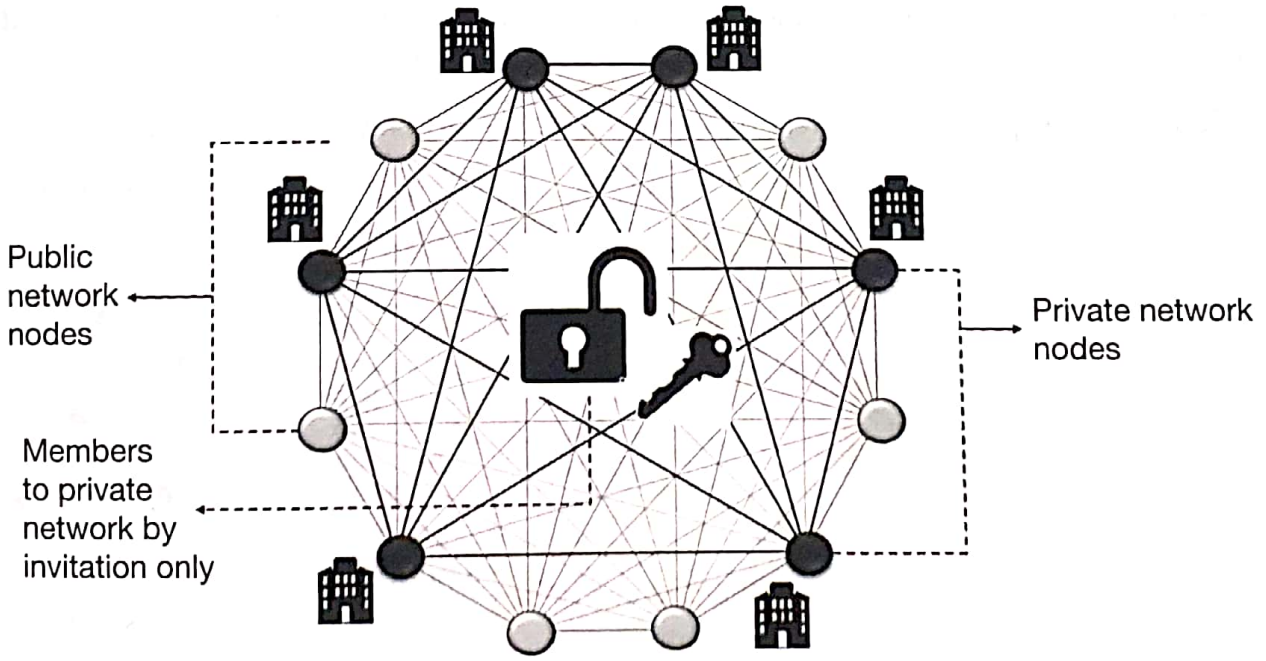


Figure 2.11: Hybrid blockchain

The key features of a hybrid blockchain are:

- It is open to the public; hence the public blockchain is an element where anyone can participate. It also has the private network element that consists of participants invited by a central authority.
- The ledger is distributed within the network of participants. Either the central body or the network members (based on the application need) can decide which transaction data can be public and which needs to be confined to specific members.
- The private network creates the hash of transactions and passes it on to the public network for validation and approval, thus maintaining the trustless nature of blockchain.
- As a combination of a public and private blockchain, some processes are public while others are private. The processes can be changed by the private or central authority to fit the purpose, with the consensus of all the nodes in the network.
- Consensus protocols are available in both public and private networks. The main public network uses the DPoS consensus while the private network can have its own.
- 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.
- Data is auditable. Though the privacy of transactions is maintained, it is open for verifiability as and when required.

Challenges of a hybrid blockchain:

- It is a relatively new ecosystem, with XinFin being the only genuinely functional hybrid blockchain protocol/platform currently available for highly regulated markets.

XinFin, a non-profit organization based in Singapore, with a focus on cross-border trade and finance, has built the first hybrid blockchain platform, TradeFinex, combining Ethereum (for the public blockchain state) and Quorum (for the private blockchain state). Organizations like Ripple, IBM, and other technology companies are exploring hybrid blockchains. It is hoped that we shall see more applications of the technology in the coming years.

Table 2.1 Comparison: Public, Private, Consortium and Hybrid Blockchain

	Public blockchain	Private blockchain	Consortium blockchain	Hybrid blockchain
Organization type				
	Public	Single entity or organization	Multiple organizations or enterprises	Highly regulated enterprise

(Continued)

Table 2.1 (*Continued*)

	Public blockchain	Private blockchain	Consortium blockchain	Hybrid blockchain
Common features	- 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 and trusted participants	Known and 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 and/ or initiate transactions	Pre-approved participants can read and/ or initiate transactions	Any combination is possible; Operations are customizable. Central authority decides which transactions can be made public and which are private

(*Continued*)

Table 2.1 *(Continued)*

	Public blockchain	Private blockchain	Consortium blockchain	Hybrid blockchain
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

(Continued)

Table 2.1 (*Continued*)

	Public blockchain	Private blockchain	Consortium blockchain	Hybrid blockchain
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
Transaction speed	Slow; takes more than 10 minutes for creating a block	High; takes seconds to create a block	Very high; takes seconds to create a block	Highest
Energy consumption	Very high	Low	Low	Low
Scalability	Limited; as the network grows, the node requirements of bandwidth, storage, and computational power exponentially increases.	Better scalability as high storage and computational power is not required.	Better scalability as high storage and computational power is not required.	Highly scalable

2.3.6 Blockchain-as-a-Service

Blockchain-as-a-Service (BaaS) is a concept similar to Software-as-a-Service (SaaS) model. The complexity of the infrastructure and cost of setting up and operating the blockchain may discourage many start-ups and SMEs from adopting blockchain. By adopting the BaaS model, businesses can leverage the cloud-based solution to build their blockchain apps, smart contracts, and other blockchain functions. In contrast, the cloud-based service provider maintains the infrastructure and other back-end operations, including storage, bandwidth management, security, and resource allocation.

For a fee, the service provider takes care of setting up and keeping the blockchain infrastructure up and running on behalf of the company. This will enable the company to focus on its core business areas without worrying about infrastructure and performance-related issues.

It is a lot cheaper to host and run the blockchain on a BaaS solution rather than developing a blockchain in-house. However, security must be assessed thoroughly as mistakes in set-up or configuration/code errors can severely disrupt the blockchain.

All the major technology companies like Amazon (on AWS-Amazon Web Services), Microsoft (by MS Azure), IBM, and Oracle have launched blockchain-as-a-service (BaaS) offerings.

MTBC Inc., a healthcare information technology company based in the US, is the industry's first decentralized integrated suite of web-based solutions, including AI that operates on the BaaS platform. The solution offers interoperability between highly secure blockchain networks for better business/clinical decisions, seamless interoperability between electronic health records (EHRs) giving patients full control over their health records, faster billing cycles, reduced administrative overheads and operating costs.

2.4 Consensus Protocol

As per Webster dictionary, a consensus is a general agreement or opinion shared by all the people in a group. A protocol is a system of standard rules that are acceptable by all parties to control the exchange of information in a network. Thus, 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 uses a decentralized network architecture where anyone can be a node. All nodes are equal in the hierarchy, with no individual node having more access or advantage over the other. Nodes accept decisions collectively for the good of the whole network. The consensus protocol enables this trustless nature of blockchain.

Consensus protocols are the rules that define how the different actors in a distributed ledger authenticate and validate the transactions added to it to prevent different versions of the ledger from being created or previous transactions from being edited.

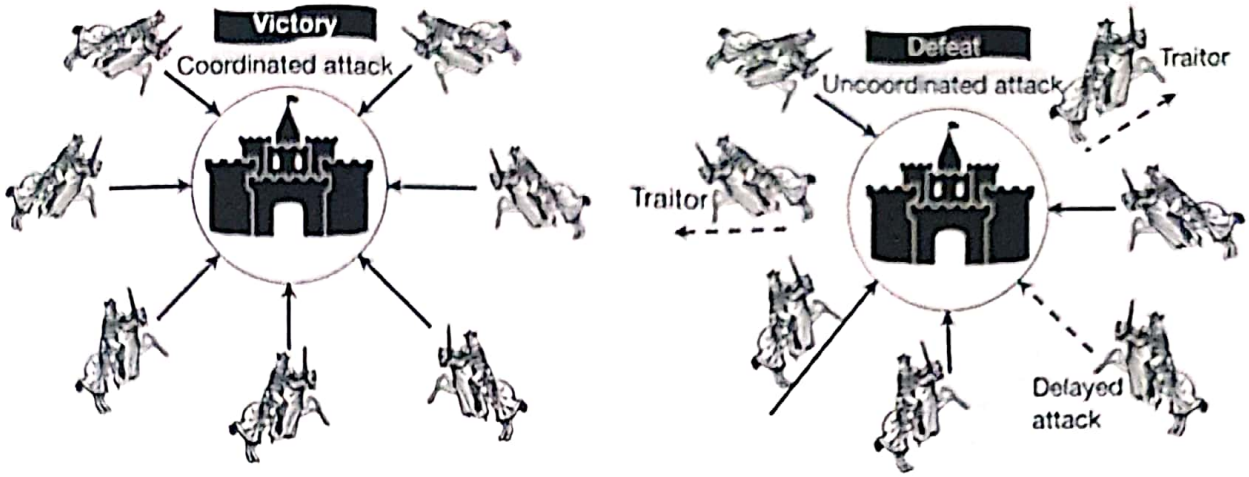
The consensus protocol aims to overcome the classic problem of a distributed computing system known as the Byzantine Generals Problem.

2.4.1 Byzantine Generals Problem

Originally coined as the "Two-generals Problem," the Byzantine Generals Problem is a thought experiment introduced in computer networking classes, especially concerning TCP (Transmission Control Protocol), to highlight that there is no guarantee that failure will not occur when it applies to a two-party communication.

The Problem: Two Byzantine (Roman) armies led by different generals are preparing to attack a fortified city. They are based on either side of the city. The city is strong enough to withstand an individual attack of either army, but not strong enough to defend itself from a coordinated attack by the two armies at the same time. In other words, the two armies must attack the city at the **same time** to win the battle (refer Fig. 2.12).

The two generals, say General A and General B need to agree on the time of the attack. The only way of communication is by sending messengers through the city. The simplest way to send the message will be for one general to take the leadership role and send a messenger through the enemy lines with a proposed day and time. The second general, on receiving the message, sends back the acknowledgment or agreement message back to the first general.



Coordinated attack leading to victory **Uncoordinated attack leading to defeat**

Figure 2.12: Byzantine generals problem

Here are the issues that may hamper victory:

- General A will hesitate to attack at the appointed time if he does not get the acknowledgment from General B.
- The enemies could capture General A's messenger and the intercept the message.
- General A's message could be intercepted and replaced with a fake message. General B may hesitate to attack as he cannot verify the authenticity of the message.
- General B may send an acknowledgment message, but there is no assurance that General B's messenger will not be caught by the enemies and the message intercepted and/or replaced.
- One of the Generals could potentially be a traitor.
- The lack of confidence or doubt between the Generals may trickle down to the soldiers in the army leading to some deserters, thus compromising the strength of the army. The desertion can lead to defeat even if there is a coordinated attack.

Hence, there is no way for either of the generals to guarantee that their counterparts have received their message. Here, the dilemma is between two generals or two participants. In a distributed network, the dilemma is between all the participants or nodes. All the participants need to verify and reach agreements neutralizing corrupt parties and disseminating false and unreliable information.

Blockchain's key feature of consensus mechanism or consensus algorithms is seen as a solution to the Byzantine Generals Problem. The consensus mechanism of blockchain aims to overcome the trust risks attributed to a distributed network system, namely,

a) *Authenticity*: The message should be easily verifiable to guarantee that it is genuine and not tampered with.
b) *Unity*: There should be a collective agreement by all parties (nodes) on action to be taken
c) *Fault-tolerance*: A few traitors or hackers should not be able to compromise the process

2.4.2 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.

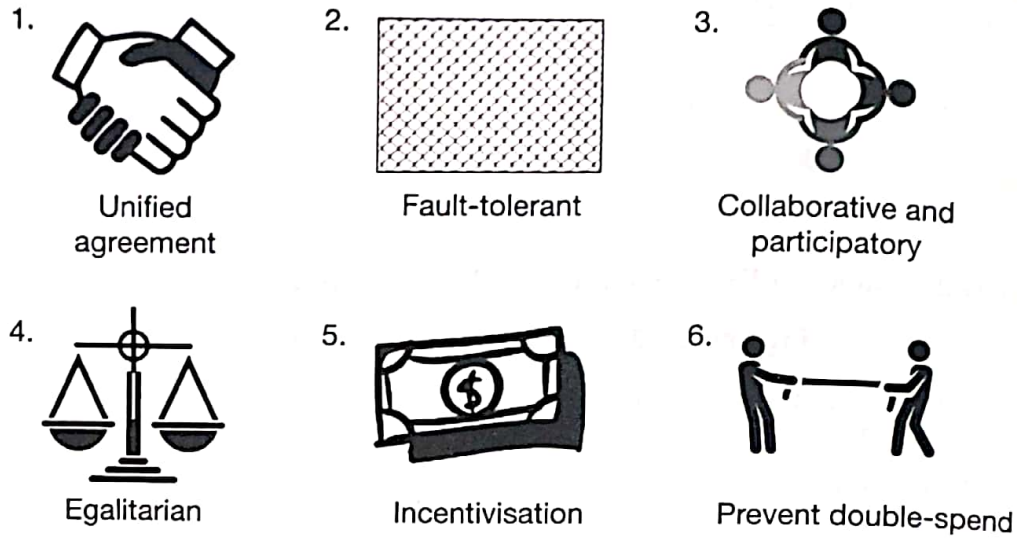


Figure 2.13: Objectives of consensus mechanism

The main objectives of a consensus mechanism (refer Fig. 2.13) are

a) Unified Agreement

There should be a unified agreement on which data is valid and accurate. Protocol rules embedded in the network ensure this.

b) Fault-tolerant

In a blockchain, every node acts as both a client and a server. In a blockchain network, there are thousands of such nodes that lead to high fault tolerance. Even if some of the nodes are unresponsive, there are still a considerable number of nodes communicating to keep the system functional.

c) Collaborative and Participatory

The consensus mechanism should ensure that all nodes participate in the overall process, in the best interests of the group as opposed to the interest of a single or a few nodes. The

features of blockchain P2P network architecture enable constant communication between nodes. It also allows for any node to inspect and verify that the underlying process is fair to all participants in the network.

d) Egalitarian

There should be no discrimination between nodes. Blockchain runs on a peer-to-peer network architecture where every node should be considered equal to every other node. Every node should have equal weightage.

e) Incentivization

Validating new transactions and securing them on the blockchain requires "miners" to solve complex mathematical problems based on cryptographic hash algorithms requiring vast amounts of computational resources, including electricity. Rules are built into the consensus mechanism to incentivize the miners to work for the system, making it more secure.

f) Prevent Double-spend

Double-spending refers to the possibility of digital currency or token being spent more than once by falsification or duplication. Protocol rules embedded in the blockchain consensus mechanism ensure valid and authentic transactions. Additionally, substantial computational resources are extended by miners to secure transactions, making it difficult to double-spend or alter transactions.

There are different kinds of consensus algorithms that work on different principles. Proof-of-Work (PoW) and Proof-of-Stake (PoS) are the most commonly used consensus mechanisms in blockchain for public distributed ledgers. In contrast, Proof-of-Authority (PoA) and Proof-of-Elapsed Time (PoET) algorithms are applied in private blockchains.

2.4.3 Consensus Algorithms

2.4.3.1 Proof of Work

Founded by Satoshi Nakamoto, Proof-of-Work is the most well-known consensus mechanism used by the first blockchain **Bitcoin** in 2009. Here, several nodes of the distributed ledger called **miners** compete to solve a complicated mathematical problem based on a cryptographic hash algorithm. The solution found is called Proof of Work or **PoW**. Without proof of work, adding blocks to the blockchain would be too easy and could make it vulnerable to hackers.

The mining node releases the proof of work to the other nodes for verification to reach consensus.

The solution to the problem is difficult to produce but easy for the network to verify. The process of mining is extremely computation-intensive. So the first miner who manages to produce the PoW will be rewarded either in the form of bitcoins or digital currency.

Disadvantages of the PoW consensus mechanism are

1) **Time-consuming**: Miners have to iterate over many nonces before finding the right solution, which is a time-consuming process (refer Section 1.4.6).
2) **High energy consumption**: Miners conduct significant work in terms of processing power and electricity to find the nonce for creating the winning hash. As only one miner can be successful, for all other miners who competed, it is wasted energy.
3) **51% risk**: To counteract the high time and energy consumption in transaction validation, some miners group the mining pools together to combine their mining resources for more efficiency and savings. Mining pool goes against the basic principle of distributed ledgers as a person or group gaining control of over 50% of the network's computing power can control the validation process. This is usually referred to as a 51% attack.

Bitcoin, Litecoin, Dash, Monero, and Ethereum use PoW as the underlying consensus mechanism.

2.4.3.2 Proof of Elapsed Time

Proof of elapsed time (**PoET**) was conceived in 2016 by Intel. It is commonly used in permissioned blockchain networks to decide on the mining rights or the block winners on the network.

PoET mechanism is based on the principle of a fair lottery system where every single node is equally likely to be a winner. Each miner node in the blockchain network is provided with a randomized timer object from a trusted code that generates a random wait time. This method of randomization aims to circumvent any attempt by a miner to get a timer with a shorter period. The miner who completes the designated waiting time commits a new block to the blockchain and broadcasts the relevant information across the blockchain network. The process is then repeated for the discovery of the next block.

The PoET mechanism is similar to the PoW consensus mechanism except that instead of being resource-intensive, it allows a miner's processor to sleep and switch to other tasks for the specified time, thereby increasing its efficiency and reducing power consumption. Also, in PoET, the identity of the miners is known unlike in PoW, where it remains anonymous.

Disadvantage attributed to PoET consensus mechanism:

1) **Vulnerability** – It relies heavily on the use of a Trusted Execution Environment (TEEs), i.e., Intel SGX-enabled CPUs. Though the protocol prevents nodes from running multiple instances of "wait time" to boost their chances of success, it is vulnerable to various other security attacks such as "Foreshadow", which attacks the secure enclave of SGX.

Hyperledger Sawtooth architecture, developed by Intel, uses PoET consensus.

2.4.3.3 Proof of Stake

The Proof of Stake (**PoS**) was implemented as a consensus algorithm for Peercoin in 2012. The more stake one has in the validating node, the less chance one will be tempted

to corrupt the validating process. In other words, the users with the highest stake in a cryptocurrency will have the most interest in maintaining and securing the network because any attacks would diminish the reputation and price of the cryptocurrency that they hold.

In PoS, the mining nodes are called **validators** or **forgers** or **delegates**. A forger has to commit some of his/her stake (cryptocurrency) in the network as collateral to be in the running for a chance to validate the transaction. An algorithm will randomly select a forger based on the percentage stake or collateral he or she has put forward. Validating nodes can forge or create new blocks proportional to the amount they have staked; i.e., a node with a 10% stake in the network can validate 10% of transactions.

Energy consumption is less here as compared to PoW consensus. Also, the forgers are paid a transaction fee as against the block reward of PoW consensus.

PoS addresses all the disadvantages of PoW with low time and energy consumption and a reduced threat of 51% attack. It also incentivizes forgers to validate legitimately as their staked amount will be forfeited in case of fraudulent behaviour.

Disadvantages of the PoS consensus mechanism are:

1) ***Cheaper to attack:*** A PoS based network is cheaper to attack as the perpetrator would just need to spend some money and not invest in the combined set of money, time, hardware, electricity, and other resources.

2) ***Centralization risk:*** The richest forger can control the consensus mechanism and get even richer.

Examples of Blockchain using the PoS mechanism are Peercoin, NXT, and BlackCoin. Ethereum uses PoS over an existing PoW blockchain, resulting in a hybrid PoW/PoS system called Casper Friendly Finality Gadget (FFG).

2.4.3.4 Delegated Proof of Stake

Delegated Proof of Stake (**DPoS**) is a variation of the PoS consensus mechanism. Here, the network participants or nodes use their cryptocurrency or tokens to vote for the delegates. Just as in PoS, the delegates are responsible for validating transactions and maintaining the blockchain ledger. These elected delegates are called **witnesses**. The more the crypto-coins or tokens, the more the voting power.

In addition to the PoS benefits, DPoS enables better security and even distribution of wealth. Any fraudulent activity by the witnesses can be easily detected by the voters and penalized. 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. This makes DPoS more decentralized than either PoS or PoW.

Disadvantages of DPoS consensus mechanism are:

1) **51% risk**: Since fewer people are in charge of maintaining the network, it is easier to organize a 51% attack
2) **Potential centralized power**: Sufficient decentralization cannot be achieved without compromising the scalability of the network. The more the validators, the

more is the risk of slowing the network down. Hence there is a risk of power getting concentrated in the hands of a few.

Bitshare, Lisk and EOS are examples of blockchains that use the DPoS consensus mechanism.

2.4.3.5 Proof of Authority

Proof of Authority (**PoA**) consensus mechanism proposed in 2017 is used in private blockchains. It is similar to PoS and DPoS in the sense that only a group of pre-selected authorities called **validators** secure the blockchain and can produce new blocks. However, instead of staking coins or tokens, the validators stake their identity.

The identities of the validators are public and verifiable by a reliable third party, such as a public notary database. This incentivizes the validators to act in the best interest of the network, for otherwise, their reputation is ruined. The validators are ideally limited to 25 or less to ensure the efficiency and security of the network. In addition to low energy consumption, PoA also benefits from zero node-to-node data transfer requirements. Once the nodes are verified and approved as validators, re-verification is not done unless required.

The following conditions must be met to identify validators:

- Validators must have a valid identity in the public domain that must match the records found in the public notary database.
- The authority needs to be uniform and unbiased for all validators.
- Eligibility criteria for staking identity must be stringent to ensure the trustworthiness of the validator.

Some of the issues attributed to PoA are:

1) **Semi-centralized**: Blockchains with PoA consensus mechanism lean more towards a centralized system in the form of an authority node as the validators are predetermined. However, this mechanism works well with private or consortium blockchain, enabling better scalability, such as a network of banks where each bank acts as a validator for the others.
2) **Reputational indifference**: If the payoff is strong enough, validator(s) may sacrifice their reputation. However, this issue is a high risk only if the validators are limited in number. If they fall to the influence of third parties with malicious interests, the network could fail.

VChainThor blockchain, Ethereums' Kovan and Rinkeby testnets use the PoA consensus mechanism algorithm. Hyperledger and Ripple also use optimized versions of PoA.

2.4.3.6 Practical Byzantine Fault Tolerance

Practical Byzantine Fault Tolerance (**pBFT**) was introduced by Miguel Castro and Barbara Liskov at the MIT Laboratory for Computer Science in 1999. It is considered as one of the potential solutions to the Byzantine Generals problem. Here, the goal is to decide

whether to accept a piece of information submitted to the blockchain or not. It tolerates "Byzantine faults" based on the assumption that the number of malicious nodes in the network cannot simultaneously be equal to or exceed one-third of the overall nodes in the system in a given window of vulnerability.

It works on the format of the Byzantine Generals Problem, where all "generals" (nodes) are considered equal and take their work instruction from the **leader** node. The leader node is the primary node, and all other nodes are called secondary or **backup** nodes. The leader is selected at random in a round-robin fashion. A node **client** sends a transaction request to the leader who then broadcasts it to all the backup nodes. The leader and backup nodes will use the message with their internal state to run computation and transmit the decision result to all the client nodes. The final decision is arrived at based on the agreement of the majority.

A high hash rate is not required as pBFT relies on the minimum number of backup nodes to confirm trust, namely $(f+1)$, where f represents the maximum number of faulty nodes. Hence, it is not computationally intensive and as a result, there is substantial energy saving.

The disadvantages of the pBFT protocol are:

1) **Small group sizes** – Due to the amount of communication required between all the nodes, this model works best with small-group networks for better response times.
2) **Sybil Attacks** – A single party can assume several identities or nodes and manipulate the network. This is mitigated with larger network sizes, but scalability and throughput will be compromised.

Stellar, Ripple, and Hyperledger Iroha are some blockchains that use variants of the pBFT consensus mechanism algorithm.

2.4.3.7 RAFT Algorithm

RAFT was built as a simpler version of PAXOS consensus. The principle of the RAFT consensus mechanism is similar to that of pBFT consensus except that only the leader node can communicate with the other nodes and decide on the state of the transaction. Nodes can have three states: leader, follower, and candidate. RAFT uses randomized timers 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. The leader candidates log must be more up-to-date than the follower logs. If a candidate's log is less up-to-date than a potential follower, then the candidate is rejected by the follower.

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 majority approvals from follower nodes, it transitions to a **leader** state.

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. After receiving the confirmation from the majority of the followers,

the leader, in turn, commits the message and sends a confirmation (heartbeat) message to the client and followers.

RAFT based consensus is used in Quorum for consortium settings.

2.4.4 Other Consensus Mechanisms

Some other variations and evolving consensus mechanisms are listed below:

a. Proof of Stake Anonymous (PoSA)

A variation of PoS consensus mechanism, PoSA was first introduced in Cloakcoin in 2014. Here, nodes are incentivized for "cloaking" the transaction. There are no master nodes, making it a truly decentralized and secure network.

The cloaking nodes provide the transaction with inputs and outputs, rendering it close-to-impossible to establish the identity of the receiver or the sender of a transaction and ensuring anonymity.

b. Leased Proof of Stake (LPoS)

LPoS is another variation of the PoS mechanism. In PoS, one needs a large stake to get a chance to validate a block. Hence many users with low balances do not get a chance to generate a block. The LPoS mechanism enables users to sublet their balances to staking nodes. This allows for small holders also to forge a block of transaction in the blockchain. Any reward received is shared proportionally.

c. Proof of Importance (PoI)

First established with the NEM cryptocurrency platform, the PoI consensus mechanism works on the principle that users with the highest balance as also users who provide maximum value to the network should be incentivized. Thus, the chance of forging a block depends on many factors, including coin balance and authority.

d. Proof of Storage

Proof-of-Storage consensus mechanism is implemented in **the Storj** system. Here, the network uses a block tree. Instead of going through every single transaction listed on the blockchain, the user can only see the transactions that are of particular importance to him.

e. Proof of Burn

Iain Stewart created a Proof-of-Burn consensus. When coins are destroyed on the blockchain, it is referred to as being burned. Technically, the coins in circulation are sent to an unspendable address, known as an **eater address**. Just like in PoW consensus where the more that is invested in supercomputers and electricity, the more the chances of mining, in Proof of Burn, more the coins one burns, the more chance one gets to mine blocks. Proof of Burn is used in Counterparty and Slimcoin.

f. Proof of Activity

It is a hybrid of PoW and PoS consensus mechanisms. It starts with miners vying to be the first to solve the cryptographic puzzle and claim their reward. However, the blocks being mined are not transactions but templates with header information and the mining reward address. Once the template block is mined, the PoS selects a random group of validators to sign the block. Once all validators sign the block, it becomes part of the blockchain. If the block remains unsigned by a few, it is discarded, and the next winning block template is used. Proof of Activity reduces the risk of a 51% attack to zero. However, the energy consumption issue is not eased.

g. Proof of Capacity (PoC)

PoC consensus algorithm is currently used only in **Burstcoin**. It was built to circumvent the high energy consumption of PoW and coin hoarding risks of PoS. Here, mining nodes can use the space available on their hard drive to mine crypto-coins instead of using the mining device's computing power.

In the PoC mechanism, the miners will first "plot" their hard drives, i.e., they will create a list of all possible nonce values through repeated hashing of data, including a miner's account. In other words, the miners will compute the solutions and store them ahead of time. Once the actual mining starts, the miner with the fastest solution wins the block.

Using hard drives is said to be 30 times more energy-efficient than ASIC (application-specific integrated circuit)-based mining. It is also more decentralized as anyone can own a basic hard drive.

h. Directed Acyclic Graph (DAG)

DAG was created to circumvent the inefficiencies of PoW. In PoW consensus, it takes around 10 minutes or more to create a block, and blocks cannot be created simultaneously. With DAG, transactions can run on different chains simultaneously. ITC (IoT Chain), built on DAG consensus protocol, is said to process over 10,000 transactions per second.

Consensus protocol algorithms are continuously evolving with multiple variants of PoW and PoS and hybrids.

Summary

Though the distributed ledger technology (DLT) is not complex, it enables complex solutions with blockchain, thus enabling the transformation from "the internet of information" to "the internet of value."

Blockchain is still an evolving technology (refer Fig. 2.14). Though Bitcoin was introduced to replace government-controlled currency with a digital decentralized and distributed alternative, it is yet to be seen whether cryptocurrency is a viable and lasting option.

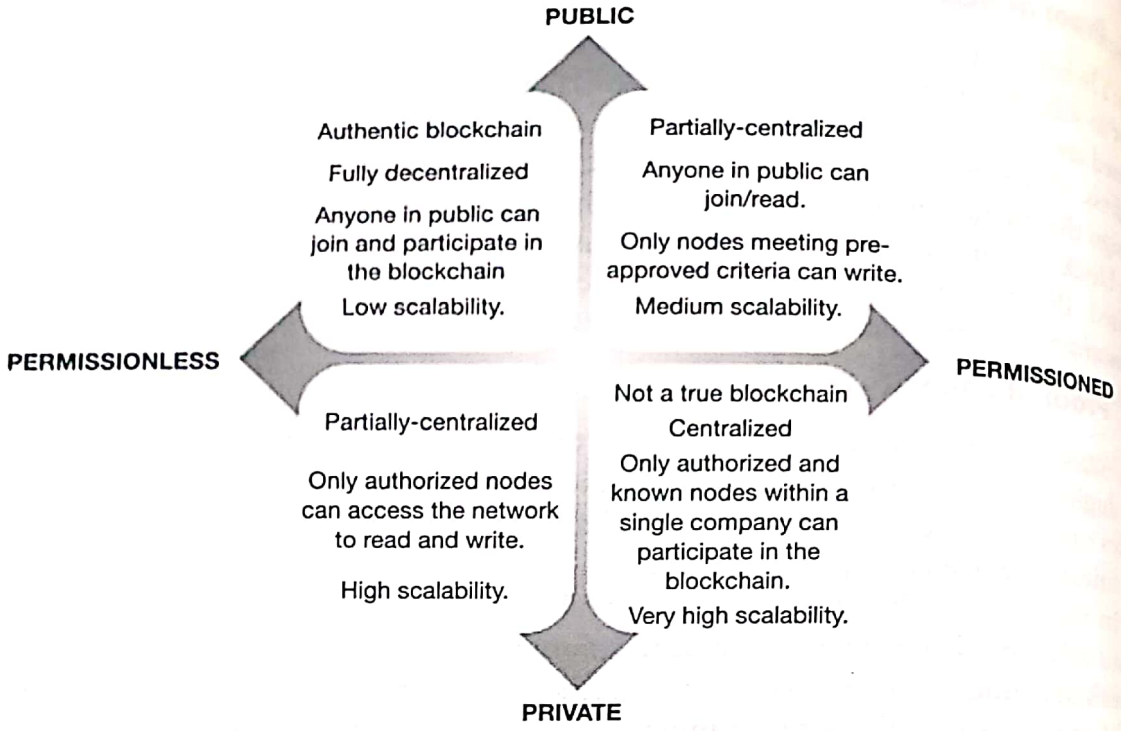


Figure 2.14 Evolution of blockchain

However, one cannot ignore the revolutionary features of blockchain and the opportunities that it provides to financial and various other industrial sectors. With consortium and hybrid blockchain, FinTech and other large technology companies have gone beyond the use-cases of the public and private blockchain.

The consensus mechanisms are continually transforming to cater to various industry needs and mitigating the risks identified with each blockchain type. PoW and PoS were the most popular consensus mechanisms initially used for public blockchains. However, many variants and hybrids like DPoS, PoET, PoA, and others have since been introduced to either circumvent the limitations or improve upon the existing consensus algorithms.

This chapter skims the surface of blockchain types and their underlying protocols.

EXERCISES

Multiple Choice Questions

1. **One of the main characteristics of blockchain technology is its _____, where transactions are not under the control of any single party.**
   A. Decentralization
   B. Centralization
   C. Privatization
   D. Immutability