Module-1

Module-1

Distributed systems, CAP theorem, Byzantine Generals problem, Consensus. The history of blockchain, Introduction to blockchain, Various technical definitions of blockchains, Generic elements of a blockchain, Features of a blockchain, Applications of blockchain technology, Tiers of blockchain technology, Consensus in blockchain, CAP theorem and blockchain, Benefits and limitations of blockchain.

Chapter 1

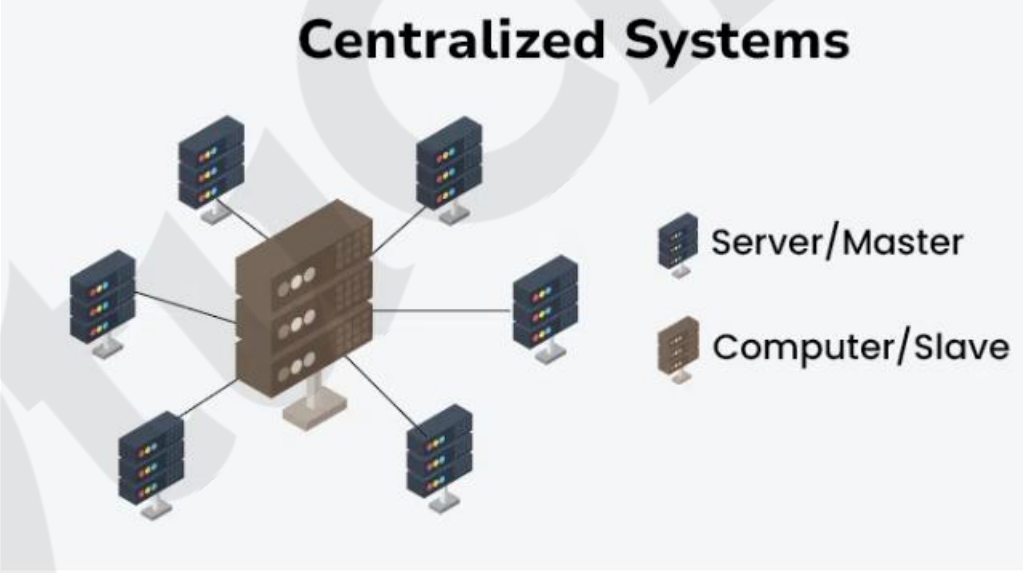
Reference material: Imran Bashir. "Mastering Blockchain", Second & Third Edition, Packt Publications & various e-resources like geekforgeeks etc

Introduction:

before diving to blockchain it is important to understand the concepts of centralised systems, decentralised systems and distributed systems

Centralised systems:

- Centralised systems are a type of computing architecture where all or most of the processing and data storage is done on a single central server or a group of closely connected servers.
- This central server manages all operations, resources, and data, acting as the hub through which all client requests are processed. The clients, or nodes, connected to the central server typically have minimal processing power and rely on the server for most computational tasks.



Centralized Systems

Key Characteristics of Centralized Systems

1. Single Point of Control:

- All data processing and management tasks are handled by the central server.
- Easier to manage and maintain since there is one primary location for administration.

2.Simplicity:

- Simplified architecture with a clear structure where all operations are routed through the central node.
- Easy to deploy and manage due to centralized nature.

3.Efficiency:

- Efficient use of resources as the central server can be optimized for performance.
- Easier to implement security measures and updates centrally.

4. Scalability Issues:

- Limited scalability as the central server can become a bottleneck if the load increases significantly.
- Adding more clients can strain the server's resources, leading to performance degradation.

5.Single Point of Failure:

- If the central server fails, the entire system can become inoperative.
- High availability and redundancy measures are essential to mitigate this risk.
- Users interact with the system as if it were a single entity.

Centralized system use cases

1.Enterprise Resource Planning (ERP) Systems:

Description: Centralized ERP systems manage and integrate core business processes such as finance, HR, and supply chain in a single system.

Benefits: Simplified management, consistent data, and centralized control over business processes.

2.Customer Relationship Management (CRM) Systems:

Description: Centralized CRM systems store and manage customer data, interactions, and sales processes in one location.

Benefits: Improved customer data consistency, streamlined customer service, and centralized reporting.

3.Email Servers:

Description: Centralized email servers manage and store email communications for an organization.

Benefits: Centralized email storage, simplified backup and security measures, and easy management of user accounts.

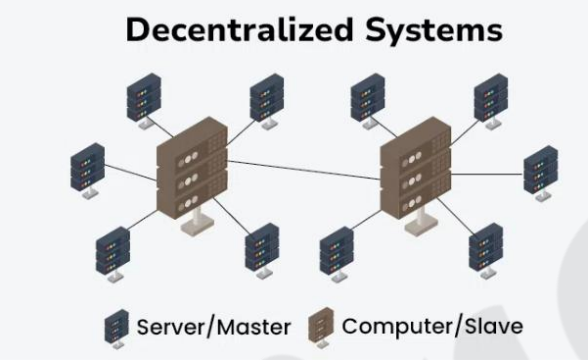
4.Banking Systems:

Description: Centralized banking systems manage customer accounts, transactions, and financial services through a central server.

Benefits: Enhanced security, centralized transaction processing, and consistent financial records

What are Decentralized Systems?

Decentralized systems are computing architectures where multiple nodes, often spread across different locations, share control and processing power without a single central authority. Each node in a decentralized system operates independently but collaborates with others to achieve common goals. This structure enhances fault tolerance, scalability, and resilience compared to centralized systems.



Key Characteristics of Decentralized Systems:

1. Distributed Control:

- No single point of control or failure.
- Each node operates independently, contributing to the overall system's functionality.

2. Fault Tolerance:

- If one node fails, the system can continue to function with the remaining nodes.
- Enhanced resilience against failures and attacks.

3.Scalability:

- Easier to scale by adding more nodes without overwhelming a central point.
- Load distribution across multiple nodes improves performance and resource utilization.

4.Coordination and Communication:

- Nodes must communicate and coordinate to maintain system integrity and consistency.
- Complex algorithms and protocols often manage this coordination.

5.Autonomy and Redundancy:

- Each node can operate autonomously, contributing to redundancy and reducing single points of failure.
- Data and services are often replicated across multiple nodes for reliability.

Decentralized system use cases

1.Blockchain and Cryptocurrencies: Decentralized ledgers that record transactions across multiple nodes without a central authority.

Benefits: Enhanced security, transparency, and resistance to fraud and censorship.

Examples: Bitcoin, Ethereum.

2. Peer-to-Peer (P2P) File Sharing: Networks where users share files directly with each other without a central server.

Benefits: Increased resilience, reduced central bottlenecks, and distributed resource sharing.

Examples: BitTorrent, Gnutella.

3.Decentralized Finance (DeFi) Platforms: Financial services built on blockchain technology, offering services like lending, borrowing, and trading without intermediaries.

Benefits: Greater accessibility, reduced fees, and increased transparency.

Examples: Uniswap, Compound.

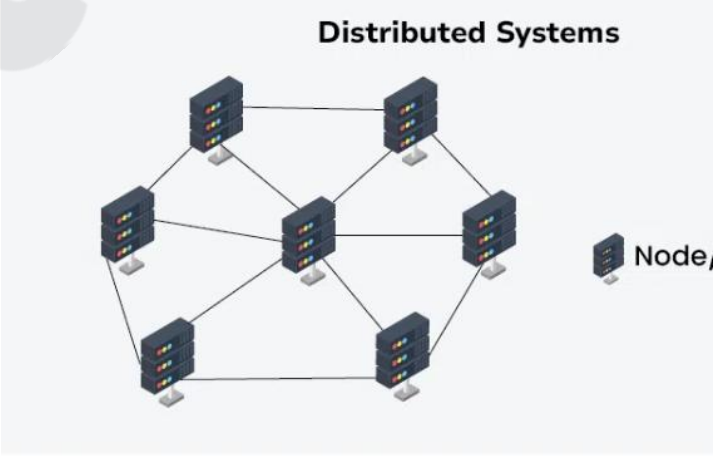
4.Mesh Networks: Networks where each node relays data for the network, providing a decentralized approach to internet connectivity.

Benefits: Increased network resilience, scalability, and coverage in remote areas.

Examples: Community-based Wi-Fi networks, disaster recovery networks.

What are Distributed Systems?

Distributed systems are computing architectures where multiple independent nodes or computers work together to achieve a common goal. These nodes communicate and coordinate with each other over a network, appearing as a single coherent system to the end user. Distributed systems aim to improve performance, reliability, scalability, and resource sharing by leveraging the collective power of interconnected devices.



Distributed Systems

Key Characteristics of Distributed Systems

1.Geographical Distribution:

- Nodes are spread across different physical locations.
- They communicate via a network, such as a local area network (LAN) or the internet.

2.Resource Sharing:

- Nodes share resources such as processing power, storage, and data.
- This enables more efficient utilization of resources.

3.Concurrency:

- Multiple nodes operate concurrently, performing tasks simultaneously.
- This parallelism enhances the system's overall performance and throughput.

4.Scalability:

- Easy to scale by adding more nodes to the system.
- System capacity and performance improve with the addition of resources.

5.Fault Tolerance:

- Designed to handle failures gracefully.
- Redundancy and replication ensure the system remains operational even if some nodes fail.

6.Transparency:

- The complexity of the distributed system is hidden from users.

Distributed system use cases

1.Cloud Computing Platforms:

Description: Cloud services provide scalable and flexible computing resources over the internet.

Benefits: On-demand resource allocation, high availability, and fault tolerance.

Examples: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform.

2.Content Delivery Networks (CDNs):

Description: Networks of distributed servers that deliver web content based on users' geographic locations.

Benefits: Reduced latency, increased content delivery speed, and load balancing.

Examples: Akamai, Cloudflare, Amazon CloudFront.

3.Distributed Databases:

Description: Databases that store data across multiple servers to improve performance and reliability.

Benefits: High availability, scalability, and fault tolerance.

Examples: Google Spanner, Amazon DynamoDB, Apache Cassandra.

4.Microservices Architectures:

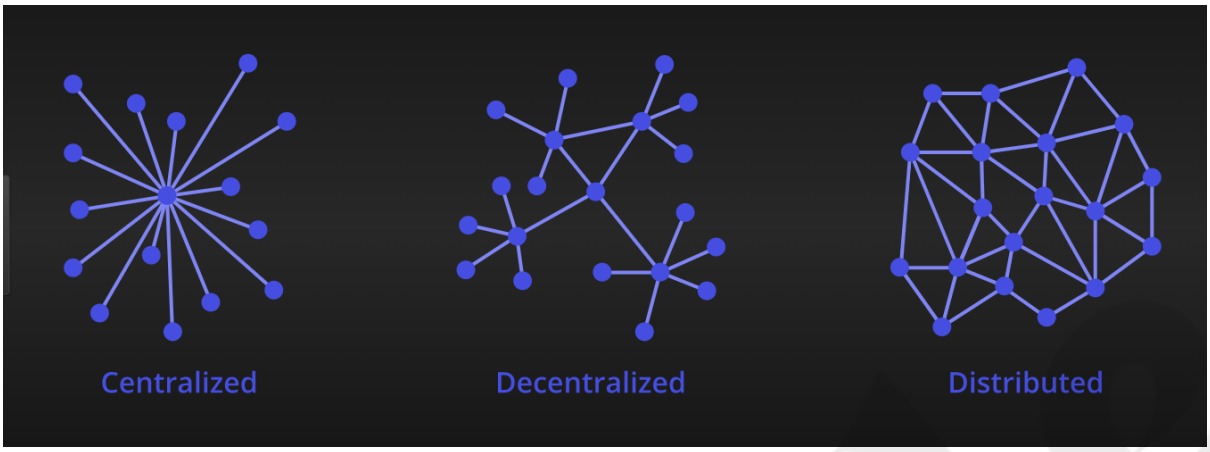
Description: Architectures where applications are built as a collection of loosely coupled services.

Benefits: Improved scalability, easier maintenance, and fault isolation.

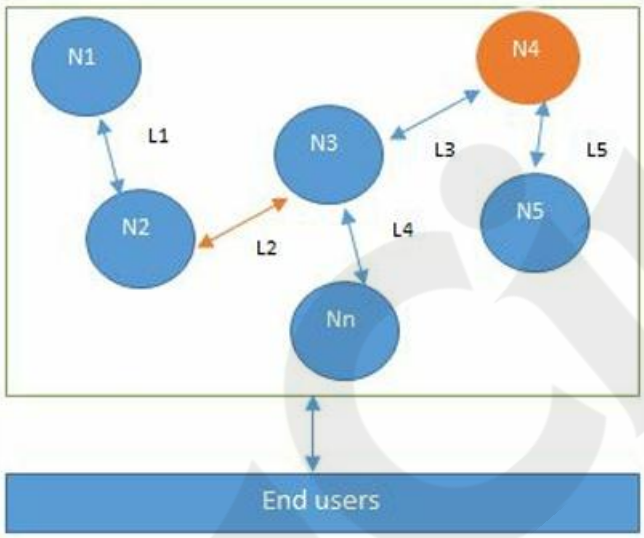
Examples: Netflix, Uber, Amazon.

Comparison between centralized vs decentralized vs distributed systems

Aspect	Centralized Systems	Decentralized Systems	Distributed Systems
Definition	Single central server controls and manages all operations.	Multiple nodes with independent control, no central authority.	Multiple interconnected nodes working together as a single system.
Control	Centralized control with a single point of management.	Distributed control, each node operates independently.	Shared control, nodes collaborate to achieve common goals.
Single Point of Failure	High risk; if the central server fails, the whole system fails.	Reduced risk; failure of one node does not impact the entire system.	Reduced risk; designed for fault tolerance and redundancy.
Scalability	Limited scalability, can become a bottleneck.	More scalable, can add nodes independently.	Highly scalable, can add more nodes to distribute the load.
Resource Utilization	Central server resources are heavily utilized.	Resources are spread across multiple nodes.	Efficient resource sharing across nodes.
Performance	Can be high initially but may degrade with increased load.	Generally good, performance improves with more nodes.	High performance due to parallel processing and resource sharing.
Management	Easier to manage centrally.	More complex, requires managing multiple nodes.	Complex, requires coordination and management of many nodes.
Latency	Lower latency, as operations are managed centrally.	Can vary, depends on the distance between nodes.	Potentially higher latency due to network communication.



Design of a distributed system- N4 is a Byzantine node, L2 is broken or a slow network link



- A node can be defined as an individual player in a distributed system.
- All nodes are capable of sending and receiving messages to and from each other.
- Nodes can be honest, faulty, or malicious and have their own memory and processor.
- A node that can exhibit **arbitrary behavior i**s also known as a **Byzantine node.**
- This arbitrary behavior can be intentionally malicious, which is detrimental to the operation of the network.
- Generally, any unexpected behavior of a node on the network can be categorized as **Byzantine.** This term arbitrarily encompasses any behavior that is unexpected or malicious.

➢ **Challenges:**

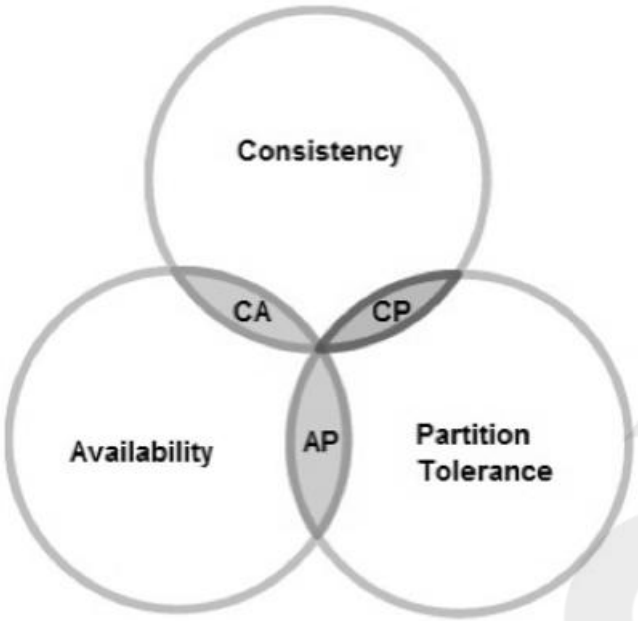
- The main challenge in distributed system design is coordination between nodes and fault tolerance.

- Even if some of the nodes become faulty or network links break, the distributed system should tolerate this and should continue to work flawlessly in order to achieve the desired result.
- This has been an area of active research for many years and several algorithms and mechanisms has been proposed to overcome these issues.

Distributed systems are so challenging to design that a theorem known as the CAP theorem has been proved and states that a distributed system cannot have all much desired properties simultaneously.

CAP THEOREM

- **This is also known as Brewer's theorem, introduced originally by Eric Brewer as a conjecture in 1998; in 2002 it was proved as a theorem by Seth Gilbert and Nancy Lynch.**
- **Definition:** The theorem states that any distributed system cannot have Consistency, Availability, and Partition tolerance simultaneously:
  - **Consistency** is a property that ensures that all nodes in a distributed system have a single latest copy of data.
  - **Availability** means that the system is up, accessible for use, and is accepting incoming requests and responding with data without any failures as and when required.
  - **Partition tolerance** ensures that if a group of nodes fails the distributed system still continues to operate correctly.
  - **It has been proven that a distributed system cannot have all the afore mentioned three properties at the same time.**
- In order to achieve fault tolerance, replication is used.
- This is a common and widely used method to achieve fault tolerance. Consistency is achieved using consensus algorithms to ensure that all nodes have the same copy of data. This is also called **state machine replication**. Blockchain is basically a method to achieve state machine replication
- In **general,** there are **two types of fault** that a node can experience: where a faulty node has simply crashed and where the faulty node can exhibit malicious or inconsistent behavior arbitrarily. This is the type which is difficult to deal with since it can cause confusion due to misleading information.



BYZANTINE GENERAL PROBLEM

History:

Before discussing consensus in distributed systems, events in history are presented that are precursors to the development of successful and practical consensus mechanisms.

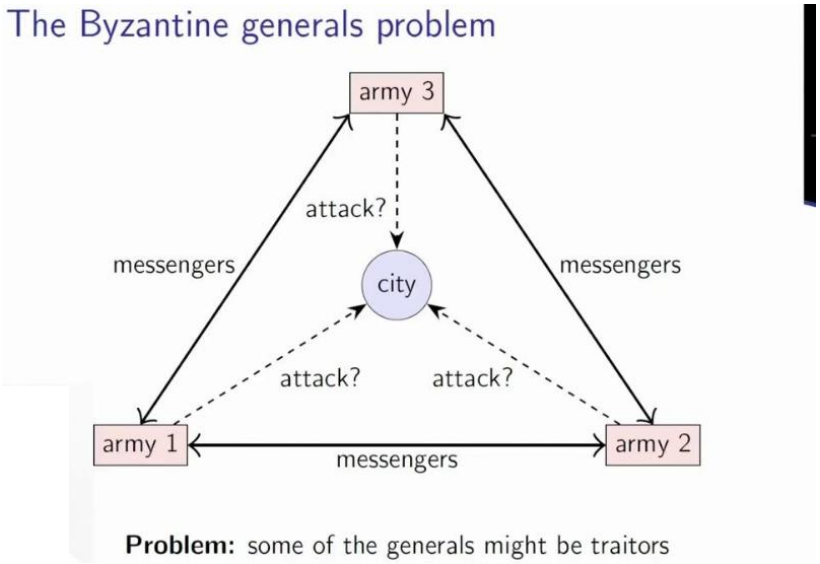
In **September 1962, Paul Baran introduced the idea of cryptographic signatures with his paper On distributed communications networks**. This is the paper where the concept of decentralized networks was also introduced for the very first time.

What is Byzantine General's Problem?

In 1982, The Byzantine General's Problem was invented by Leslie Lamport, Robert Shostak, and Marshall Pease. Byzantine Generals Problem is an impossibility result which means that the solution to this problem has not been found yet as well as helps us to understand the importance of blockchain. It is basically a game theory problem that provides a description of the extent to which decentralized parties experience difficulties in reaching consensus without any trusted central parties.

- The Byzantine army is divided into many battalions in this classic problem called the Byzantine General's problem, with each division led by a general.
- The generals connect via messenger in order to agree to a joint plan of action in which all battalions coordinate and attack from all sides in order to achieve success.

It is probable that traitors will try to sabotage their plan by intercepting or changing the messages. As a result, the purpose of this challenge is for all of the faithful commanders to reach an agreement without the imposters tampering with their plans.



How Bitcoin Solves the Byzantine General's Problem?

In the Byzantine Generals Problem, the untampered agreement that all the loyal generals need to agree to is the blockchain. Blockchain is a public, distributed ledger that contains the records of all transactions. If all users of the Bitcoin network, known as nodes, could agree on which transactions occurred and in what order, they could verify the ownership and create a functioning, trustless money system without the need for a centralized authority. Due to its decentralized nature, blockchain relies heavily on a consensus technique to validate transactions. It is a peer-to-peer network that offers its users transparency as well as trust. Its distributed ledger is what sets it apart from other systems. Blockchain technology can be applied to any system that requires proper verification.

Proof Of Work: The network would have to be provable, counterfeit-resistant, and trust-free in order to solve the Byzantine General's Problem. Bitcoin overcame the Byzantine General's Problem by employing a Proof-of-Work technique to create a clear, objective regulation for the blockchain. Proof of work (PoW) is a method of adding fresh blocks of transactions to the blockchain of a cryptocurrency. In this scenario, the task consists of creating a hash (a long string of characters) that matches the desired hash for the current block.

Counterfeit Resistant: Proof-of-Work requires network participants to present proof of their work in the form of a valid hash in order for their block, i.e. piece of information, to be regarded as valid. Proof-of-Work requires miners to expend significant amounts of energy and money in order to generate blocks, encouraging them to broadcast accurate information and so protecting the network. Proof-of-Work is one of the only ways for a decentralized network to agree on a single source of truth, which is essential for a monetary system. There can be no

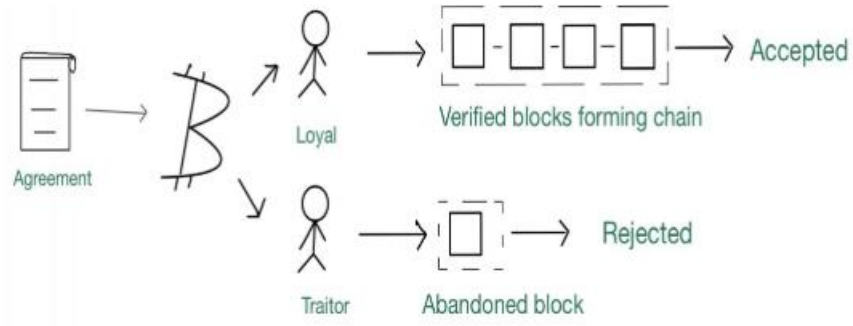
disagreement or tampering with the information on the blockchain network because the rules are objective. The ruleset defining which transactions are valid and which are invalid, as well as the system for choosing who can mint new bitcoin, are both objectives.

Provable: Once a block is uploaded to the blockchain, it is incredibly difficult to erase, rendering Bitcoin's history immutable. As a result, participants of the blockchain network may always agree on the state of the blockchain and all transactions inside it. Each node independently verifies whether blocks satisfy the Proof-of-Work criterion and whether transactions satisfy additional requirements.

Trust-free: If any network member attempts to broadcast misleading information, all network nodes immediately detect it as objectively invalid and ignore it. Because each node on the Bitcoin network can verify every information on the network, there is no need to trust other network members, making Bitcoin a trustless system.

Byzantine Fault Tolerance (BFT) This problem was solved in 1999 by **Castro and Liskov** who presented **the Practical Byzantine Fault Tolerance (PBFT) algorithm**. Later on in 2009, the first practical implementation was made with the invention of bitcoin where the Proof of Work (PoW) algorithm was developed as a mechanism to achieve consensus

- The Byzantine Fault Tolerance was developed as inspiration in order to address the Byzantine General's Problem. The Byzantine General's Problem, a logical thought experiment where multiple generals must attack a city, is where the idea for BFT originated.
- Byzantine Fault Tolerance is one of the core characteristics of developing trustworthy blockchain rules or features is tolerance.
- When two-thirds of the network can agree or reach a consensus and the system still continues to operate properly, it is said to have BFT.
- Blockchain networks' most popular consensus protocols, such as proof-of-work, proof-of-stake, and proof-of-authority, all have some BFT characteristics.
- In order to create a decentralized network, the BFT is essential.
- The consensus method determines the precise network structure. For instance, BFT has a leader as well as peers who can and cannot validate.
- In order to maintain the sequence of the Blockchain SC transactions and the consistency of the global state through local transaction replay, consensus messages must pass between the relevant peers.
- More inventive approaches to designing BFT systems will be found and put into practice as more individuals and companies investigate distributed and decentralized systems. Systems that use BFT are also employed in sectors outside of blockchains, such as nuclear power, space exploration, and aviation.



CONSENSUS

Definition: Consensus is a process of agreement between distrusting nodes on a final state of data.

- In order to achieve consensus different algorithms can be used.

Distributed Consensus: It is easy to reach an agreement between two nodes (for example in client-server systems) but when multiple nodes are participating in a distributed system and they need to agree on a single value it becomes very difficult to achieve consensus. This concept of achieving consensus between multiple nodes is known as distributed consensus.

CONSENSUS MECHANISMS

- A consensus mechanism is a set of steps that are taken by all, or most, nodes in order to agree on a proposed state or value. For more than three decades this concept has been researched by computer scientists in the industry and Academia.
- Consensus mechanisms have recently come into the limelight and gained much popularity with the advent of bitcoin and blockchain. There are various requirements which must be met in order to provide the desired results in a consensus mechanism.
- The following are their requirements with brief descriptions:

  Agreement: All honest nodes decide on the same value.

  Termination: All honest nodes terminate execution of the consensus process and eventually reach a decision.

  Validity: The value agreed upon by all honest nodes must be the same as the initial value proposed by at least one honest node.

  Fault tolerant: The consensus algorithm should be able to run in the presence of faulty or malicious nodes (Byzantine nodes).

  Integrity: This is a requirement where by no node makes the decision more than once. The nodes make decisions only once in a single consensus cycle.

TYPES OF CONSENSUS MECHANISM

- There are various types of consensus mechanism; some common types are described as follows:
  i. **Byzantine fault tolerance-based:** With no compute intensive operations such as partial hash inversion, this method relies on a simple scheme of nodes that are publishing signed messages. Eventually, when a certain number of messages are received, then an agreement is reached.
  ii**. Leader-based consensus mechanisms:** This type of mechanism requires nodes to compete for the leader-election lottery and the node that wins it proposes a final value.
  Example:
- Many practical implementations have been proposed such as **Paxos,** the most famous protocol introduced by Leslie Lamport in 1989.
- In Paxos nodes are assigned various roles such as Proposer, Acceptor, and Learner. Nodes or processes are named replicas and consensus is achieved in the presence of faulty nodes by agreement among a majority of nodes.
- Another alternative to Paxos is **RAFT**, which works by assigning any of three states, that is, Follower, Candidate, or Leader, to the nodes. A Leader is elected after a candidate node receives enough votes and all changes now have to go through the Leader, who commits the proposed changes once replication on the majority of follower nodes is completed.

THE HISTORY OF BLOCKCHAIN

[GENERAL OVERVIEW OF BLOCKCHAIN**]**

Blockchain technology has evolved over decades through the contributions of cryptographers, computer scientists, and innovators. Below is a chronological overview of the key milestones in the history of blockchain:

Pre-Bitcoin Era (Before 2008)

1991-1992: Birth of Cryptographic Timestamping Stuart Haber and W. Scott Stornetta proposed a system for cryptographically timestamping digital documents to prevent backdating and tampering. Introduced the concept of linked blocks secured using cryptographic hashes.

1992: Merkle trees were added to improve efficiency, allowing multiple document certificates in a single block.

1998: First Concept of Digital Currency

Computer scientist Nick Szabo introduced the idea of Bit Gold, a decentralized digital currency.

Used cryptographic puzzles and proof-of-work mechanisms, similar to Bitcoin but lacked a working implementation.

2004: Reusable Proof of Work (RPoW)

Hal Finney developed Reusable Proof of Work (RPoW), which allowed users to exchange non-replicable proof-of-work tokens.Laid the foundation for decentralized digital money.

The Bitcoin Era (2008-2013)

2008: The Birth of Bitcoin

Satoshi Nakamoto, a pseudonymous entity, published the Bitcoin whitepaper:

"Bitcoin: A Peer-to-Peer Electronic Cash System" - Introduced **Proof of Work (PoW)** as a consensus mechanism. Proposed a decentralized ledger to prevent double-spending without intermediaries.

2009: Bitcoin Blockchain Goes Live

January 3, 2009: Satoshi Nakamoto mined the genesis block (Block 0) with the message:

"The Times 03/Jan/2009 Chancellor on brink of second bailout for banks."

January 12, 2009: First-ever Bitcoin transaction occurred between Satoshi Nakamoto and Hal Finney.

2010: First Commercial Bitcoin Transaction

Laszlo Hanyecz purchased two pizzas for 10,000 BTC, marking the first real-world Bitcoin transaction.

Bitcoin Marketplaces Emerge: Exchanges like Mt. Gox were founded.

2011-2013: Growth of Bitcoin & Early Altcoins

2011: The first major Bitcoin alternative, Litecoin (LTC), was introduced.

2012: Introduction of the Bitcoin Halving mechanism, reducing mining rewards every four years.

2013: Bitcoin reached $1,000 for the first time.

Blockchain 2.0 – Smart Contracts & Ethereum (2013-2017)

2013: Ethereum is Proposed

Vitalik Buterin, a Bitcoin developer, proposed Ethereum, a blockchain supporting smart contracts (self-executing contracts stored on the blockchain).

2015: Ethereum Goes Live

July 30, 2015: Ethereum was launched with Ethereum Virtual Machine (EVM), allowing developers to create Decentralized Applications (dApps).

2016: The DAO Hack & Ethereum Fork

- The DAO (Decentralized Autonomous Organization) raised $150 million in ETH but was hacked due to a smart contract vulnerability.
- Led to a hard fork, splitting Ethereum into:
- Ethereum (ETH) – The new chain.
- Ethereum Classic (ETC) – The original chain.

2017: Rise of ICOs & Blockchain Adoption

Initial Coin Offerings (ICOs) surged, raising billions in funding for blockchain projects.

Governments and enterprises started experimenting with blockchain.

Blockchain 3.0 – Scalability, DeFi, NFTs (2018-Present)

2018-2019: Institutional Adoption & Challenges

Governments and banks explored Central Bank Digital Currencies (CBDCs).

Enterprise blockchains like Hyperledger and R3 Corda gained traction.

2020-2021: DeFi Boom & NFT Explosion

Decentralized Finance (DeFi): Platforms like Uniswap, Aave, and Compound revolutionized lending and trading.

Non-Fungible Tokens (NFTs): Digital assets gained mainstream adoption, with NFT sales reaching billions.

2022-Present: Ethereum 2.0 & Web3

Ethereum 2.0 (The Merge) transitioned from PoW to PoS, reducing energy consumption.

Web3, Metaverse, and Layer 2 solutions (like Polygon, Optimism) improved blockchain scalability.

Governments continue regulating cryptocurrencies while blockchain adoption in industries like supply chain, finance, and healthcare grows.

Conclusion

Blockchain technology has evolved from a cryptographic concept in 1991 to a global decentralized ecosystem. With smart contracts, DeFi, and Web3, blockchain is set to revolutionize industries beyond just cryptocurrency.

The concept of electronic cash or digital currency is not new. Since the 1980s, e-cash protocols have existed that are based on a model proposed by David Chaum.

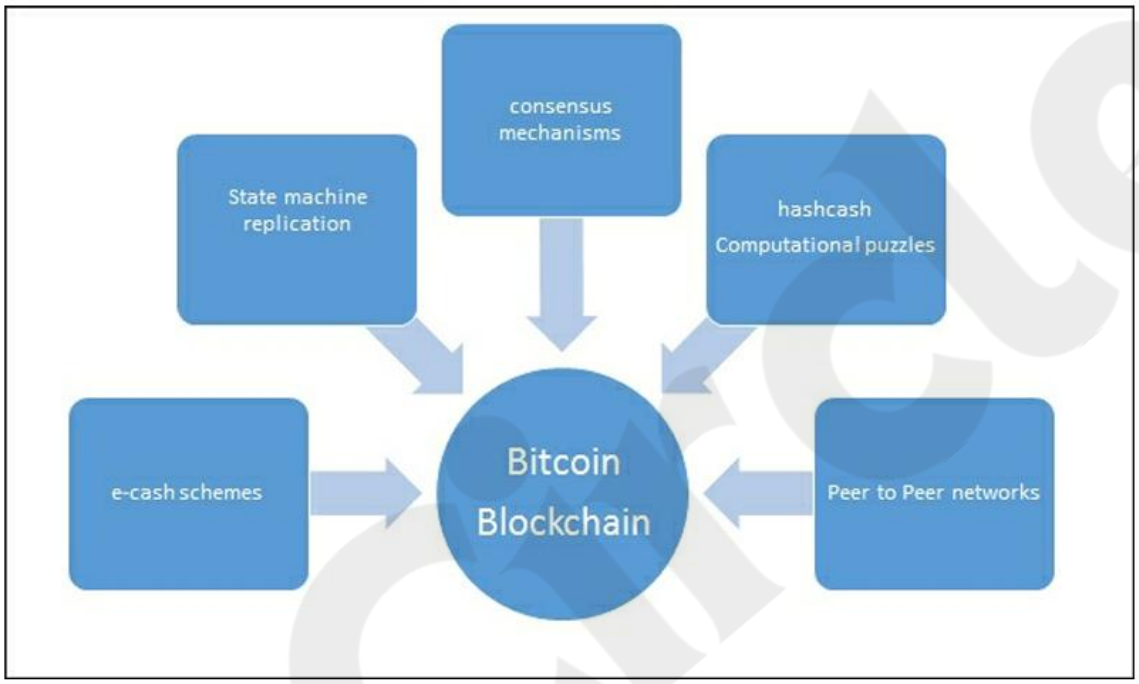
ELECTRONIC CASH

- Just as understanding the concepts of distributed systems is necessary in order to understand blockchain technology, the idea of electronic cash is also essential to appreciate the first and astonishingly successful application of blockchain: **the bitcoin, or broadly cryptocurrencies.**
- Theoretical concepts in distributed systems such as consensus algorithms provided the basis of the practical implementation of Proof of Work algorithms in bitcoin; moreover, ideas from different electronic cash schemes also paved the way for the invention of cryptocurrencies, specifically bitcoin.
- The concept of electronic cash Fundamental issues that need to be addressed in e-cash systems are accountability and anonymity.

- David Chaum addressed both of these issues in his seminal paper in 1984 by introducing two cryptographic operations, namely blind signatures and secret sharing. **Cryptography and Technical Foundations**. At the moment, it is sufficient to say that blind signatures allow signing a document without actually seeing it and secret sharing is a concept that allows the detection of using the same e-cash token twice (double spending).
- After this other protocols emerged such as Chaum, Fiat, and Naor (CFN), e-cash schemes that introduced anonymity and double spending detection. Brand's e-cash is another system that improved on CFN, made it more efficient, and introduced the concept of security reduction to prove statements about the e-cash scheme.
- Security reduction is a technique used in cryptography to prove that a certain algorithm is secure by using another problem as a comparison. Put another way, a cryptographic security algorithm is as hard to break as some other hard problem; thus by comparison it can be deduced that the cryptographic security algorithm is secure too.
- A different but relevant concept called **hashcash was introduced by Adam Back in 1997** as a PoW system to control e-mail spam. The idea is quite simple: if legitimate users want to send e-mails then they are required to compute a hash as a proof that they have spent a reasonable amount of computing resources before sending the e-mail.
- Generating hashcash is a compute intensive process but does not inhibit a legitimate user from sending the e-mail because the usual number of e-mails required to be sent by a legitimate user is presumably quite low. On the other hand, if a spammer wants to send e mails, usually thousands in number, then it becomes infeasible to compute hashcash for all e-mails, thus making the spamming effort expensive; as a result this mechanism can be used to thwart e-mail spamming.
- Hashcash takes a considerable amount of computing resources to compute but is easy and quick to verify. Verification is performed by the user who receives the e-mail. Hashcash is popularized by its use in the bitcoin mining process. This idea of using computational puzzles or pricing functions to prevent e-mail spam was introduced originally in 1992 by Cynthia Dwork and Moni Naor.
- Pricing function was the name given to the hard functions that are required to be computed before access to a resource can be granted. Later, Adam Back invented hashcash independently in 1997, which introduced the usage of computing hash functions as PoW.
- **In 1998 b-money was introduced by Wei Dai and proposed the idea of creating money via solving computational puzzles such as hashcash**. It's based on a peer-to-peer network where each node maintains its own list of transactions.
- Another similar idea **by Nick Szabo called BitGold was introduced in 2005** and also proposed solving computational puzzles to mint digital currency.
- **In 2005 Hal Finney introduced the concept of cryptographic currency by combining ideas from b-money and hashcash puzzles but it still relied on a centralized trusted authority.** There were multiple issues with the schemes described in infeasible preceding paragraphs. These problems range from no clear solution of disagreements between nodes to reliance on a central trusted third party and trusted timestamping.
- **In 2009 the first practical implementation of a cryptocurrency named bitcoin was introduced; for the very first time it solved the problem of distributed consensus in**

a trustless network. It uses public key cryptography with hashcash as PoW to provide a secure, controlled, and decentralized method of minting digital currency. The key innovation is the idea of an ordered list of blocks composed of transactions and cryptographically secured by the PoW mechanism.

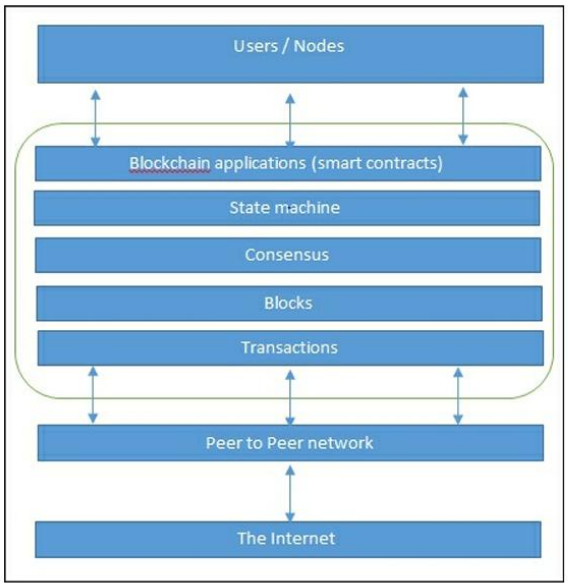
- Looking at all the aforementioned technologies and their history, it is easy to see how ideas and concepts from electronic cash schemes and distributed systems were combined together to invent bitcoin and what now is known as blockchain. This can also be visualized with the help of the following diagram.



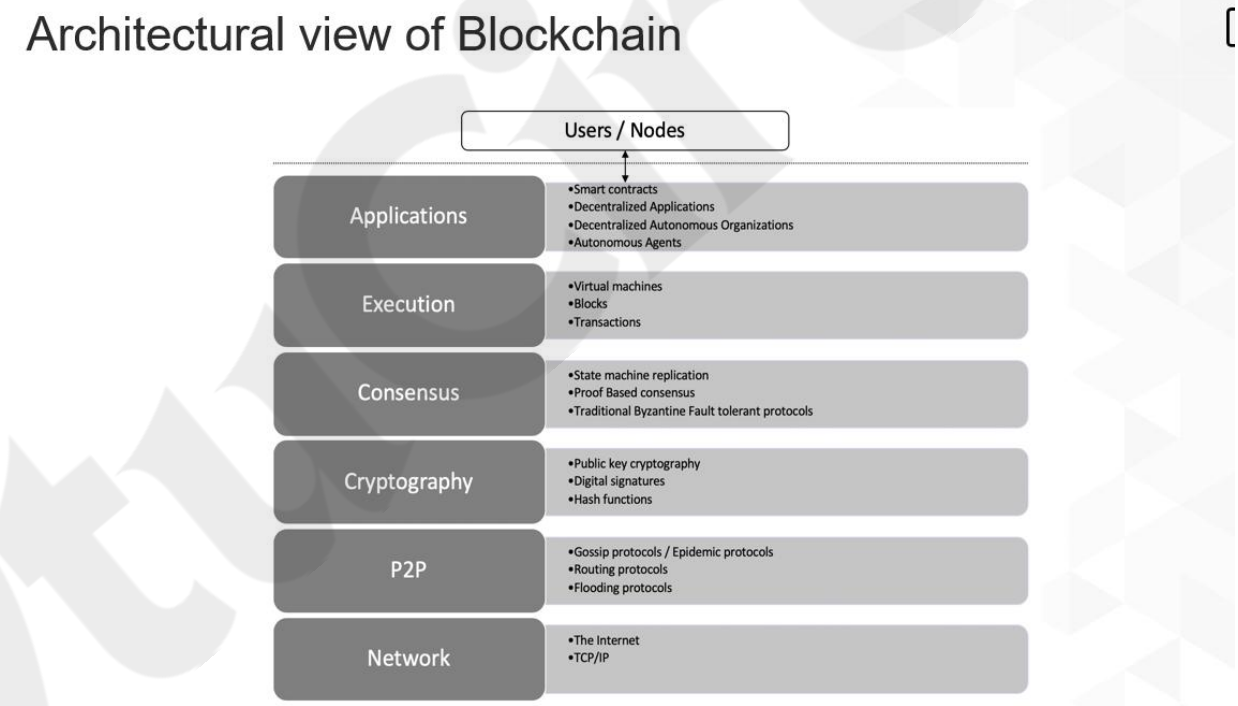
INTRODUCTION TO BLOCKCHAIN

Definition: Blockchain at its core is a peer-to-peer distributed ledger that is cryptographically secure, append only, immutable (extremely hard to change), and updateable only via consensus or agreement among peers.

Blockchain can be thought of as a layer of a distributed peer-to-peer network running on top of the Internet, as can be seen below in the diagram. It is analogous to SMTP, HTTP, or FTP running on top of TCP/IP. This is shown in the following diagram:

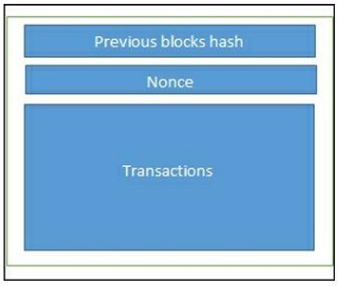


The network view of a blockchain



Definition2: From a business point of view a blockchain can be defined as a platform whereby peers can exchange values using transactions without the need for a central trusted arbitrator.

The Structure of a block:



The structure of a block

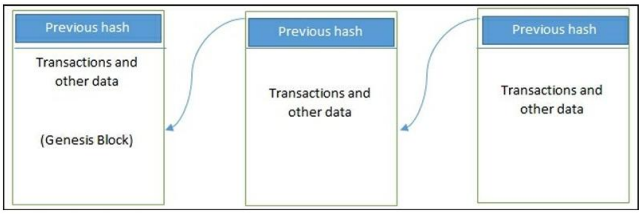
- A block is simply a selection of transactions bundled together in order to organize them logically. It is made up of transactions and its size is variable depending on the type and design of the blockchain in use.
- A reference to a previous block is also included in the block unless it's a genesis block.
- A genesis block is the first block in the blockchain that was hardcoded at the time the blockchain was started.
- The structure of a block is also dependent on the type and design of a blockchain, but generally there are a few attributes that are essential to the functionality of a block, such as the block header, pointers to previous blocks, the time stamp, nonce, transaction counter, transactions, and other attributes

VARIOUS TECHNICAL DEFINITIONS OF BLOCKCHAINS

- **Blockchain** is a decentralized consensus mechanism. In a blockchain, all peers eventually come to an agreement regarding the state of a transaction.
- Blockchain is a distributed shared ledger. Blockchain can be considered a shared ledger of transactions. The transaction are ordered and grouped into blocks. Currently, the real-world model is based on private databases that each organization maintains whereas the distributed ledger can serve as a single source of truth for all member organizations that are using the blockchain.
- Blockchain is a data structure; it is basically a linked list that uses hash pointers instead of normal pointers. Hash pointers are used to point to the previous block.

GENERIC ELEMENTS OF A BLOCKCHAIN

The structure of a generic blockchain can be visualized with the help of the following diagram:



Generic structure of a blockchain

1. **Addresses:** Addresses are unique identifiers that are used in a transaction on the blockchain to denote senders and recipients.

- An address is usually a public key or derived from a public key. While addresses can be reused by the same user, addresses themselves are unique.
- In practice, however, a single user may not use the same address again and generate a new one for each transaction.
- This newly generated address will be unique. Bitcoin is in fact a pseudonymous system. End users are usually not directly identifiable but some research in de-anonymizing bitcoin users have shown that users can be identified successfully.
- As a good practice it is suggested that users generate a new address for each transaction in order to avoid linking transactions to the common owner, thus avoiding identification.

2. Transaction:

- A transaction is the fundamental unit of a blockchain.
- A transaction represents a transfer of value from one address to another

3. Block:

A block is composed of multiple transactions and some other elements such as the previous block hash (hash pointer), timestamp, and nonce.

4. Peer-to-peer network: As the name implies, this is a network topology whereby all peers can communicate with each other and send and receive messages.

5.Scripting or programming language

- This element performs various operations on a transaction. Transaction scripts are predefined sets of commands for nodes to transfer tokens from one address to

another and perform various other functions. Turing complete programming language is a desirable feature of blockchains; however, the security of such languages is a key question and an area of important and ongoing research.

6. Virtual machine This is an extension of a transaction script. A virtual machine allows Turing complete code to be run on a blockchain (as smart contracts) whereas a transaction script can be limited in its operation.

- Virtual machines are not available on all blockchains; however, various blockchains use virtual machines to run programs, for example Ethereum Virtual Machine (EVM) and Chain Virtual Machine (CVM).

7.State machine A blockchain can be viewed as a state transition mechanism whereby a state is modified from its initial form to the next and eventually to a final form as a result of a transaction execution and validation process by nodes.

8. Nodes A node in a blockchain network performs various functions depending on the role it takes. A node can propose and validate transactions and perform mining to facilitate consensus and secure the blockchain. This is done by following a consensus protocol. (Most commonly this is PoW.) Nodes can also perform other functions such as simple payment verification (lightweight nodes), validators, and many others functions depending on the type of the blockchain used and the role assigned to the node.

9. Smart contracts These programs run on top of the blockchain and encapsulate the business logic to be executed when certain conditions are met. The smart contract feature is not available in all blockchains but is now becoming a very desirable feature due to the flexibility and power it provides to the blockchain applications.

FEATURES OF BLOCKCHAIN:

A blockchain performs various functions. These are described below in detail.

- **Distributed consensus**:
  Distributed consensus is the major underpinning of a blockchain. This enables a blockchain to present a single version of truth that is agreed upon by all parties without the requirement of a central authority.
- **Transaction verification:**
  Any transactions posted from nodes on the blockchain are verified based on a predetermined set of rules and only valid transactions are selected for inclusion in a block.
- **Platforms for smart contracts**:
  A blockchain is a platform where programs can run that execute business logic on behalf of the users. As explained earlier, not all blockchains have a mechanism to execute smart contracts; however, this is now a very desirable feature.
- **Transferring value between peers:**
  Blockchain enables the transfer of value between its users via tokens. Tokens can be thought of as a carrier of value.
- **Generating cryptocurrency**:
  This is an optional feature depending on the type of blockchain used. A blockchain can generate cryptocurrency as an incentive to its miners who validate the transactions and spend resources in order to secure the blockchain.

- **Smart property**:
  For the first time it is possible to link a digital or physical asset to the blockchain in an irrevocable manner, such that it cannot be claimed by anyone else; you are in full control of your asset and it cannot be double spent or double owned.

  Compare it with a digital music file, for example, which can be copied many times without any control; on a blockchain, however, if you own it no one else can claim it unless you decide to transfer it to someone. This feature has far-reaching implications especially in Digital Rights Management (DRM) and electronic cash systems where double spend detection is a key requirement. The double spend problem was first solved in bitcoin.
- **Provider of security**:

  Blockchain is based on proven cryptographic technology that ensures the integrity and availability of data.

  - Generally, confidentiality is not provided due to the requirements of transparency. This has become a main barrier for its adaptability by financial institutions and other industries that need privacy and confidentiality of transactions.
  - As such it is being researched very actively and there is already some good progress made. It could be argued that in many situations confidentiality is not really needed and transparency is preferred instead.
  - For example, in bitcoin confidentiality is not really required; however, it is desirable in some scenarios. Research in this area is very ripe and already major progress has been made towards providing confidentiality and privacy on blockchain. A more recent example is Zcash
  - Other security services such as nonrepudiation and authentication are also provided by blockchain as all actions are secured by using private keys and digital signatures.
- **Immutability**:

  This is another key feature of blockchain: records once added onto the blockchain are immutable. There is the possibility of rolling back the changes but this is considered almost impossible to do as it will require an unaffordable amount of computing resources.

  - For example, in much desirable case of bitcoin if a malicious user wants to alter the previous blocks then it would require computing the PoW again for all those blocks that have already been added to the blockchain. This difficulty makes the records on a blockchain practically immutable.
- **Uniqueness**:
  This feature of blockchain ensures that every transaction is unique and has not been spent already. This is especially relevant in cryptocurrencies where much desirable detection and avoidance of double spending are a key requirement.
- **Smart contracts**
  Blockchain provides a platform to run smart contracts. These are automated autonomous programs that reside on the blockchain and encapsulate business logic and code in order to execute a required function when certain conditions

are met. This is indeed a revolutionary feature of blockchain as it allows flexibility, programmability, and much desirable control of actions that users of blockchain need to perform according to their specific business requirements

APPLICATIONS OF BLOCKCHAIN TECHNOLOGY

Blockchain technology has a wide range of applications across various industries due to its decentralized, transparent, and secure nature. Here are some key applications:

1. **Cryptocurrency & Digital Payments**
   - Bitcoin, Ethereum, and other cryptocurrencies.
   - Cross-border payments (Ripple, Stellar).
   - Stablecoins (USDT, USDC).

2**. Supply Chain Management**
   - Tracking products from origin to delivery.
   - Ensuring product authenticity (anti-counterfeit).
   - Transparency in logistics (IBM Food Trust).

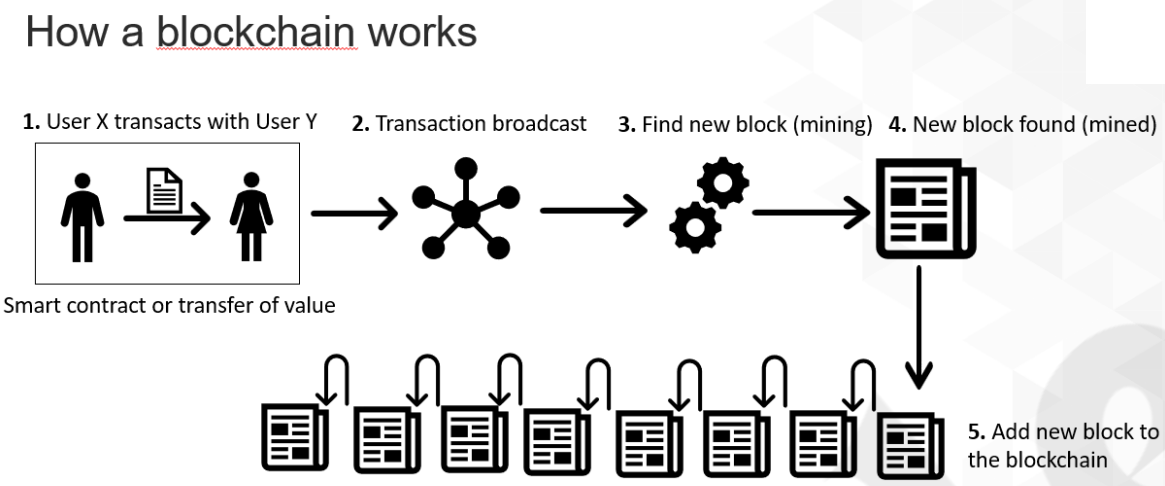
3**. Healthcare**
   - Secure storage of medical records.
   - Patient data sharing between hospitals.
   - Drug supply chain verification.

4. **Finance & Banking**
   - Smart contracts (automated transactions without intermediaries).
   - Decentralized finance (DeFi platforms like Uniswap, Aave).
   - Fraud prevention.
   - Cross-border payments.

5. **Real Estate**
   - Property ownership verification.
   - Transparent land registries.
   - Tokenized assets (fractional property ownership).

6. **Voting Systems**
   - Transparent and tamper-proof digital voting.
   - Remote voting solutions.

7. **Identity Management**
   - Digital IDs.
   - Self-sovereign identity.
   - Preventing identity theft.
8. **Intellectual Property & Royalties**
   - Protecting copyrights.
   - Automated royalty payments.
   - Digital content ownership verification (NFTs).
9. **Internet of Things (IoT)**
   - Secure device communication.
   - Data integrity for smart homes and cities.
10. **Gaming & Virtual Worlds**
    - In-game asset ownership.
    - Play-to-earn gaming models.
    - NFTs for virtual items.
11. **Charity & Donations**
    - Transparent donation tracking.
    - Proof of how funds are used.
12. **Energy Sector**
    - Peer-to-peer energy trading.
    - Carbon credit tracking.
13. **Insurance**
    - Automated claims processing.
    - Fraud detection.
14. **Education**
    - Digital certificates.
    - Academic credential verification.
15. **Government Services**
    - Birth certificates, marriage certificates.
    - Taxation and public spending tracking.



HOW BLOCKCHAINS ACCUMULATE BLOCKS

1. A node starts a transaction by signing it with its private key.

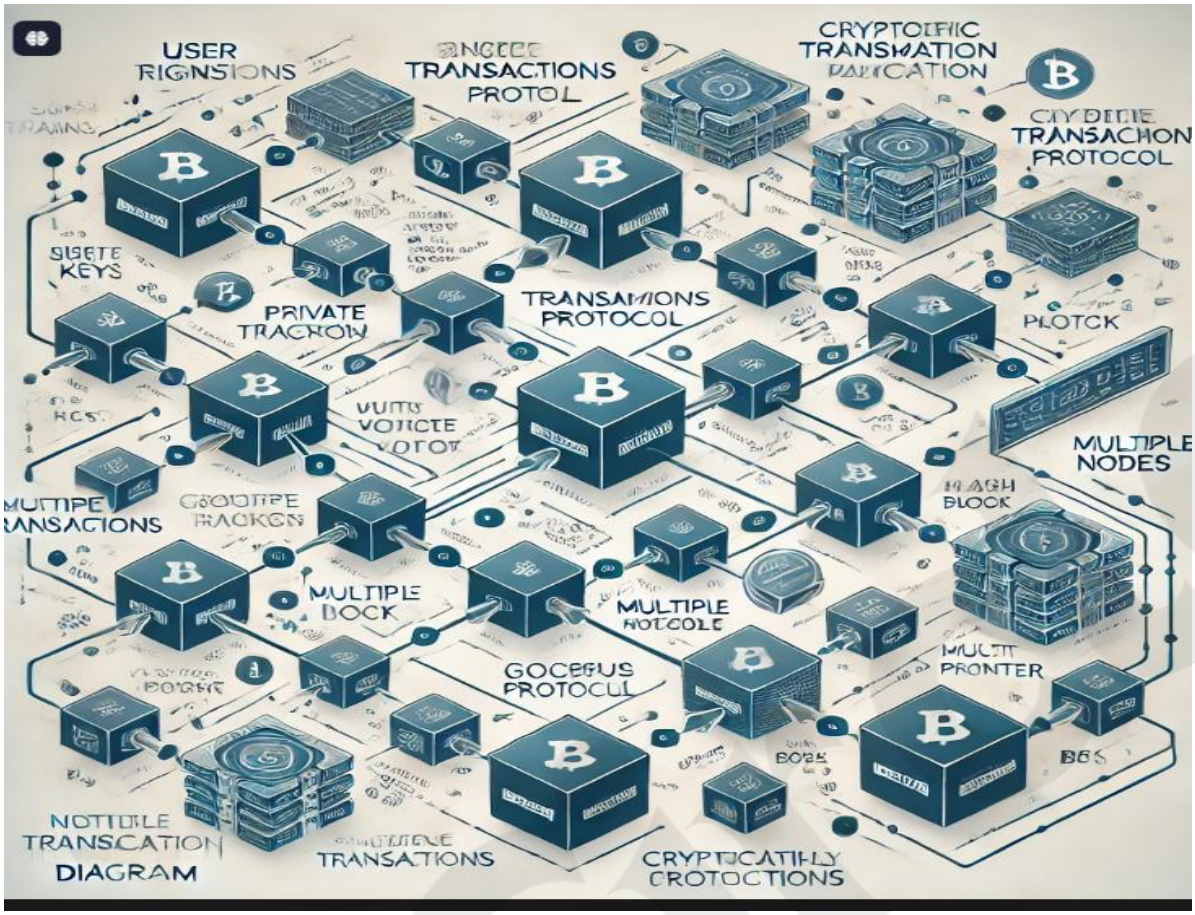
2. The transaction is propagated (flooded) by using much desirable Gossip protocol to peers, which validates the transaction based on pre-set criteria. Usually, more than one node is required to validate the transactions.

3. Once the transaction is validated, it is included in a block, which is then propagated on to the network. At this point, the transaction is considered confirmed.

4. The newly created block now becomes part of the ledger and the next block links itself cryptographically back to this block. This link is a hash pointer. At this stage, the transaction gets its second confirmation and the block gets its first.

5. Transactions are then reconfirmed every time a new block is created. Usually, six confirmations in the bitcoin network are required to consider the transaction final.

Steps 4 and 5 can be considered non-compulsory as the transaction itself is finalized in step 3; however, block confirmation and further transaction reconfirmations, if required, are then carried out in steps 4 and 5



TIERS OF BLOCKCHAIN TECHNOLOGY

Blockchain 1.0:

- This was introduced with the invention of bitcoin and is basically used for cryptocurrencies.
- Also, as bitcoin was the first implementation of cryptocurrencies it makes sense to categorize Generation 1 of blockchain technology to only include cryptographic currencies.
- All alternative coins and bitcoin fall into this category. This includes core applications such as payments and applications.

Blockchain 2.0

- Generation 2.0 blockchains are used by financial services and contracts are introduced in this generation. This includes various financial assets, for example derivatives, options, swaps, and bonds. Applications that are beyond currency, finance, and markets are included at this tier.

Blockchain 3.0

- Generation 3 blockchains are used to implement applications beyond the financial services industry and are used in more general-purpose industries such as government, health, media, the arts, and justice.

Generation X (Blockchain X)

- This is a vision of blockchain singularity where one day we will have a public blockchain service available that anyone can use just like the Google search engine. It will provide services in all realms of society.
- This is a public open distributed ledger with general-purpose rational agents (Machina Economicus) running on blockchain, making decisions and interacting with other intelligent autonomous agents on behalf of humans and regulated by code instead of law or paper contracts.

TYPES OF BLOCKCHAIN

I. Public blockchains

- As the name suggests, these blockchains are open to the public and anyone can participate as a node in the decision-making process.
- Users may or may not be rewarded for their participation. These ledgers are not owned by anyone and are publicly open for anyone to participate in.
- All users of the permission-less ledger maintain a copy of the ledger on their local nodes and use a distributed consensus mechanism in order to reach a decision about the eventual state of the ledger.
- These blockchains are also known as permission-less ledgers.

II. Private blockchains

- Private blockchains as the name implies are private and are open only to a consortium or group of individuals or organizations that has decided to share the ledger among themselves.

III. Semi-private blockchains

- Here part of the blockchain is private and part of it is public. The private part is controlled by a group of individuals whereas the public part is open for participation by anyone.

IV. Sidechains

- More precisely known as pegged sidechains, this is a concept whereby coins can be moved from one blockchain to another and moved back. Common uses include the creation of new altcoins (alternative cryptocurrencies) whereby coins are burnt as a proof of adequate stake.
- There are two types of sidechain. The example provided above for burning coins is applicable to a one-way pegged sidechain. The second type is called a two-way pegged sidechain, which allows the movement of coins from the main chain to the sidechain and back to the main chain when required.

V. Permissioned ledger:

- A permissioned ledger is a blockchain whereby the participants of the network are known and already trusted.
- Permissioned ledgers do not need to use a distributed consensus mechanism, instead an agreement protocol can be used to maintain a shared version of truth about the state of the records on the blockchain.
- There is also no requirement for a permissioned blockchain to be private as it can be a public blockchain but with regulated access control.

VI. Distributed ledger

- As the name suggests, this ledger is distributed among its participants and spread across multiple sites or organizations. This type can either be private or public. The key idea is that, unlike many other blockchains, the records are stored contiguously instead of sorted into blocks. This concept is used in Ripple.

VII. Shared ledger

- This is generic term that is used to describe any application or database that is shared by the public or a consortium.
- Fully private and proprietary blockchains These blockchains perhaps have no mainstream application as they deviate from the core idea of decentralization in blockchain technology.
- Nonetheless in specific private settings within an organization there might be a need to share data and provide some level of guarantee of the authenticity of the data. These blockchains could be useful in that scenario. For example, for collaboration and sharing data between various government departments.

VIII. Tokenized blockchains

- These blockchains are standard blockchains that generate cryptocurrency as a result of a consensus process via mining or via initial distribution.

IX. Token less blockchains

- These are probably not real blockchains because they lack the basic unit of transfer of value but are still valuable in situations where there is no need to transfer value between nodes and only sharing some data among various already trusted parties is required.
- Consensus is the backbone of a blockchain and provides decentralization of control as a result through an optional process known as mining. The choice of consensus algorithm is also governed by the type of blockchain in use. Not all consensus mechanisms are suitable for all types of blockchains.
- For example, in public permission-less blockchains it would make sense to use PoW instead of some basic agreement mechanism that perhaps is based on proof of authority. Therefore it is essential to choose a consensus algorithm appropriately for a blockchain project.

CONSENSUS IN BLOCKCHAIN

- Consensus is basically a distributed computing concept that has been used in blockchain in order to provide a means of agreeing to a single version of truth by all peers on the blockchain network.
- Roughly, the following two categories of consensus mechanism exist

1. Proof-based, leader-based, or the Nakamoto consensus whereby a leader is elected and proposes a final value

2. Byzantine fault tolerance-based, which is a more traditional approach based on rounds of votes.

Proof of Work

- This type of consensus mechanism relies on proof that enough computational resources have been spent before proposing a value for acceptance by the network. This is used in bitcoin and other cryptocurrencies.
- Currently, this is the only algorithm that has proven astonishingly successful against Sybil attacks.

Proof of Stake

- This algorithm works on the idea that a node or user has enough stake in the system;
- for example the user has invested enough in the system so that any malicious attempt would outweigh the benefits of performing an attack on the system.
- This idea was first introduced by Peercoin and is going to be used in the Ethereum blockchain.
- Another important concept in Proof of Stake (PoS) is coin age, which is a derived from the amount of time and the number of coins that have not been spent.
- In this model, the chances of proposing and signing the next block increase with the coin age.

Delegated Proof of Stake Delegated Proof of Stake (DPOS) is an innovation over standard PoS whereby each node that has stake in the system can delegate the validation of a transaction to other nodes by voting.

- This is used in the bitshares blockchain. Proof of Elapsed Time Introduced by Intel, it uses Trusted Execution Environment (TEE) to provide randomness and safety in the leader election process via a guaranteed wait time. It requires the Intel SGX (Software Guard Extensions) processor in order to provide the security guarantee and for it to be secure.,

Deposit-based consensus

- Nodes that wish to participate on the network have to put in a security deposit before they can propose a block.

- Proof of importance This idea is important and different from Proof of Stake. Proof of importance not only relies on how much stake a user has in the system but it also monitors the usage and movement of tokens by the user to establish a level of trust and importance. This is used in Nemcoin.

Federated consensus or federated Byzantine consensus :

Used in the stellar consensus protocol, nodes in this protocol keep a group of publicly trusted peers and propagates only those transactions that have been validated by the majority of trusted nodes.

Reputation-based mechanisms As the name suggests, a leader is elected on the basis of the reputation it has built over time on the network. This can be based on the voting from other members.

Practical Byzantine Fault Tolerance Practical Byzantine Fault Tolerance (PBFT) achieves state machine replication, which provides tolerance against Byzantine nodes. Various other protocols, including but are not limited to PBFT, PAXOS, RAFT, and Federated Byzantine Agreement (FBA), are also being used or have been proposed for use in many different implementations of distributed systems and blockchains.

CAP THEOREM AND BLOCKCHAIN

- Strangely, it seems that the CAP theorem is violated in blockchain, and especially in the most successful implementation: bitcoin, but this is not the case.
- In blockchains consistency is sacrificed in favor of availability and partition tolerance.
- In this scenario, Consistency (C) on the blockchain is not achieved simultaneously with Partition tolerance (P) and Availability (A), but it is achieved over time. This is called **eventual consistency,** where consistency is achieved as a result of validation from multiple nodes over time.
- For this purpose, the concept of mining was introduced in bitcoin; this is a process that facilitates the achievement of consensus by using a consensus algorithm called **PoW.**
- At a higher level, mining can be defined as a process that is used to add more blocks to the blockchain.

BENEFITS OF BLOCKCHAIN

- **Decentralization**: This is a core concept and benefit of blockchain. There is no need for a trusted third party or intermediary to validate transactions; instead a consensus mechanism is used to agree on the validity of transactions.

- **Transparency and trust**: As blockchains are shared and everyone can see what is on the blockchain, this allows the system to be transparent and as a result trust is established. This is more relevant in scenarios such as the disbursement of funds or benefits where personal discretion should be restricted.
- **Immutability:** Once the data has been written to the blockchain, it is extremely difficult to change it back. It is not truly immutable but, due to the fact that changing data is extremely difficult and almost impossible, this is seen as a benefit to maintaining an immutable ledger of transactions.
- **High availability:** As the system is based on thousands of nodes in a peer-to-peer network, and the data is replicated and updated on each and every node, the system becomes highly available. Even if nodes leave the network or become inaccessible, the network as a whole continues to work, thus making it highly available.
- **Highly secure**: All transactions on a blockchain are cryptographically secured and provide integrity.
- **Simplification of current paradigms**: The current model in many industries such as finance or health is rather disorganized, wherein multiple entities maintain their own databases and data sharing can become very difficult due to the disparate nature of the systems. But as a blockchain can serve as a single shared ledger among interested parties, this can result in simplifying this model by reducing the complexity of managing the separate systems maintained by each entity.
- **Faster dealings**: In the financial industry, especially in post-trade settlement functions, blockchain can play a vital role by allowing the quicker settlement of trades as it does not require a lengthy process of verification, reconciliation, and clearance because a single version of agreed upon data is already available on a shared ledger between financial organizations.
- **Cost saving:** As no third party or clearing houses are required in the blockchain model, this can massively eliminate overhead costs in the form of fees that are paid to clearing houses or trusted third parties.

CHALLENGES AND LIMITATIONS OF BLOCKCHAIN TECHNOLOGY

As with any technology there are challenges that need to be addressed in order to make a system more robust, useful, and accessible. Blockchain technology is no exception; in fact a lot of effort is being made in Academia and Industry to overcome the challenges posed by blockchain technology.

A selection of the most sensitive challenges are presented as follows:

- Scalability
- Adaptability
- Regulation Relatively immature technology Privacy
- Privacy

Even though blockchain technology has revolutionized many industries, it still faces several challenges and limitations that hinder its widespread adoption.

Scalability: It is one of the biggest drawbacks of blockchain technology as it cannot be scaled due to the fixed size of the block for storing information. The block size is 1 MB due to which it can hold only a couple of transactions on a single block.

Immaturity: Blockchain is only a couple-year-old technology so people do not have much confidence in it, they are not ready to invest in it several applications of blockchain are doing great in different industries but still it needs to win the confidence of even more people to be recognized for its complete utilization.

Energy Consuming: For verifying any transaction, a lot of energy is used so it becomes a problem according to the survey it is considered that 0.3 percent of the world's electricity had been used by 2018 in the verification of transactions done using blockchain technology.

Time-Consuming: To add the next block in the chain miners, need to compute nonce values many times so this is a time-consuming process and needs to be speed up to be used for industrial purposes.

Legal Formalities: In some countries, the use of blockchain technology applications is banned like cryptocurrency due to some environmental issues they are not promoting to use of blockchain technology in the commercial sector.

Storage: Blockchain databases are stored on all the nodes of the network creates an issue with the storage, increasing the number of transactions will require more storage.

Regulations: Blockchain faces challenges with some financial institutions. Other aspects of technology will be required in order to adopt blockchain in a wider aspect.