Distributed systems

Understanding distributed systems is essential in order to understand blockchain because basically blockchain at its core is a distributed system. More precisely it is a decentralized distributed system.

Distributed systems *are a computing paradigm whereby two or more nodes work with each other in a coordinated fashion in order to achieve a common outcome and it's modelled in such a way that end users see it as a single logical platform.*

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*** *is 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.

The Byzantine Generals Problem.

In 1982, a thought experiment was proposed by Lamport and others in their research paper, The Byzantine Generals Problem which is available at: https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem/ *whereby a group of army generals who lead different parts of the Byzantine army are planning to attack or retreat from a city. The only way of communicating among them is via a messenger. They need to agree to strike at the same time in order to win. The issue is that one or more generals might be traitors who could send a misleading message. Therefore, there is a need for a viable mechanism that allows for agreement among the generals, even in the presence of the treacherous ones, so that the attack can still take place at the same time. As an analogy to distributed systems, the generals can be considered nodes, the traitors as Byzantine (malicious) nodes, and the messenger can be thought of as a channel of communication among the generals.*

This problem was solved in **1999** by **Castro and Liskov** who presented the **Practical Byzantine Fault Tolerance (PBFT)** algorithm, where consensus is reached after a certain number of messages are received containing the same signed content.

This type of inconsistent behavior of Byzantine nodes can be intentionally malicious, which is detrimental to the operation of the network. Any unexpected behavior by a node on the network, whether malicious or not, can be categorized as Byzantine.

A small-scale example of a distributed system is shown in the following diagram. This distributed system has six nodes out of which one (N4) is a Byzantine node leading to possible data inconsistency. L2 is a link that is broken or slow, and this can lead to partition in the network.

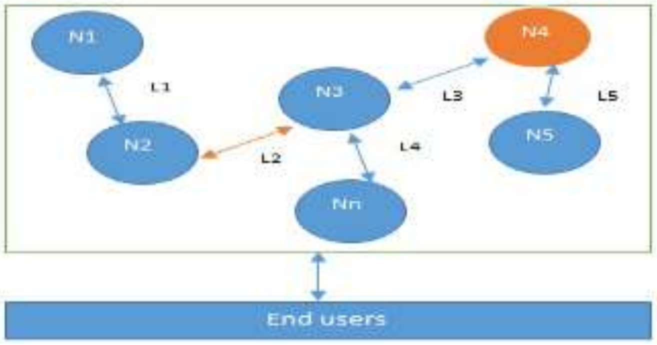


Fig: Design of a distributed system: N4 is a Byzantine node, L2 is broken or a slow network link

ABHISHEK K L

The primary 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 be able to tolerate this and continue to work to achieve the desired result. This problem has been an active area of distributed system design research for many years, and several algorithms and mechanisms have been proposed to overcome these issues.

CAP theorem and Blockchain

Distributed systems are so challenging to design that a hypothesis known as the **CAP theorem** has been proven, which ***states that any distributed system cannot have consistency, availability, and partition tolerance simultaneously:***

- **Consistency** is a property which ensures that *all nodes in a distributed system have a single, current, and identical copy of the data.*
- **Availability** means that the *nodes in the system are up, accessible for use, and are accepting incoming requests and responding with data without any failures as and when required*. In other words, data is available at each node and the nodes are responding to requests.
- **Partition tolerance** ensures that if a group of nodes is unable to communicate with other nodes due to network failures, the distributed system continues to operate correctly. This can occur due to network and node failures.

For Example: Imagine that there is a distributed system with two nodes. Now let us apply the three theorem properties on this smallest of possible distributed systems only with two nodes.

Consistency *is achieved if both nodes have the same shared state; that is, they have the same up-to-date copy of the data.*

Availability *is achieved if both nodes are up and running and responding with the latest copy of data.*

Partition tolerance *is achieved if communication does not break down between two nodes (either due to network issues, Byzantine faults, and so forth), and they are able to communicate with each other.*

*Now think of scenario where a partition occurs and nodes can no longer communicate with each other. If no new updated data comes in, it can only be updated on one node only. In that case, if the node accepts the update, then only that one node in the network is updated and therefore **consistency is lost**. Now, if the update is rejected by the node that would result in **loss of availability**. In that case due to **partition tolerance**, both **availability and consistency are unachievable.***

Blockchain manages to achieve all of these properties: to achieve **fault tolerance, replication is used**. This is a standard and widely-used method to achieve fault tolerance. **Consistency** is achieved using **consensus algorithms** in order to ensure that all nodes have the same copy of the data. This is also called **state machine replication.**

Mining is a process that facilitates the achievement of consensus by using the **PoW (Proof of Work) consensus algorithm.** At a higher level, mining can be defined as a process that is used to add more blocks to the blockchain.

ABHISHEK K L

The history of blockchain and Bitcoin

Electronic cash

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. Just as understanding the concept of distributed systems is necessary to comprehend blockchain technology, the idea of electronic cash is also essential in order to appreciate the first and astonishingly successful application of blockchain, Bitcoin, or more broadly cryptocurrencies in general.

Two fundamental e-cash system issues need to be addressed: accountability and anonymity. Accountability is required to ensure that cash is spendable only once (double-spend problem) and that it can only be spent by its rightful owner. Double spend problem arises when same money can be spent twice. As it is quite easy to make copies of digital data, this becomes a big issue in digital currencies as you can make many copies of same digital cash. Anonymity is required to protect users' privacy. As with physical cash, it is almost impossible to trace back spending to the individual who actually paid the money.

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.

In 2009, the first practical implementation of an electronic cash (e-cash) system named Bitcoin appeared. The term cryptocurrency emerged later. For the very first time, it solved the problem of distributed consensus in a trustless network. It used **public key cryptography** with a **Proof of Work (PoW)** mechanism to provide a secure, controlled, and decentralized method of minting digital currency. The key innovation was the idea of an ordered list of blocks composed of transactions and cryptographically secured by the PoW mechanism.

Looking at all the technologies mentioned earlier and their relevant history, it is easy to see how concepts from electronic cash schemes and distributed systems were combined to create Bitcoin and what now is known as blockchain. This concept can also be visualized with the help of the following diagram:

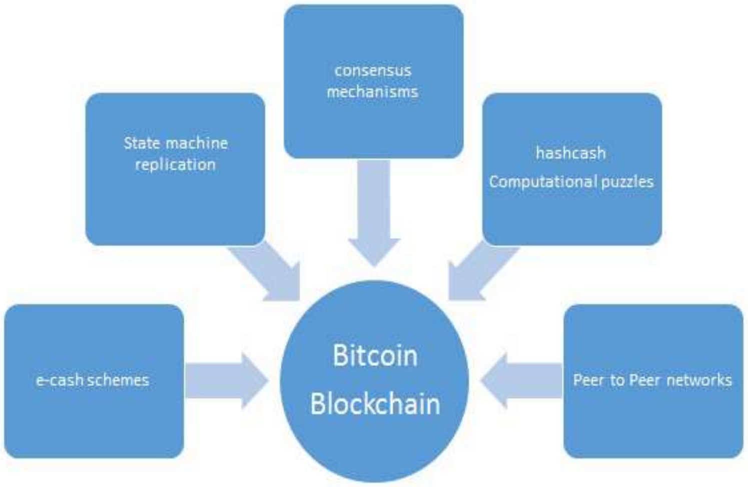


Fig: The various ideas that supported the invention of Bitcoin and blockchain

ABHISHEK K L

Introduction to blockchain

In 2008, a groundbreaking paper entitled Bitcoin: A Peer-to-Peer Electronic Cash System was written on the topic of **peer-to-peer electronic cash** under the pseudonym **Satoshi Nakamoto**. It introduced the term chain of blocks. No one knows the actual identity of Satoshi Nakamoto. After introducing Bitcoin in 2009, he remained active in the Bitcoin developer community until 2011. He then handed over Bitcoin development to its core developers and simply disappeared. Since then, there has been no communication from him whatsoever, and his existence and identity are shrouded in mystery. The term chain of blocks evolved over the years into the word blockchain.

Layman's definition: ***Blockchain is an ever-growing, secure, shared record keeping system in which each user of the data holds a copy of the records, which can only be updated if all parties involved in a transaction agree to update.***

Technical definition: ***Blockchain 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.***

We will look at all keywords in the definitions one by one.

Peer-to-peer
The first keyword in the technical definition is peer-to-peer. This means that there is no central controller in the network, and all participants talk to each other directly. This property allows for cash transactions to be exchanged directly among the peers without a third-party involvement, such as by a bank.

Distributed ledger
Which simply means that a ledger is spread across the network among all peers in the network, and each peer holds a copy of the complete ledger.

Cryptographically-secure
Next, we see that this ledger is cryptographically-secure, which means that cryptography has been used to provide security services which make this ledger secure against tampering and misuse. These services include non-repudiation, data integrity, and data origin authentication. You will see how this is achieved later in Chapter 3, Symmetric Cryptography which introduces the fascinating world of cryptography.

ABHISHEK K L

Blockchain Architecture

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 in the following diagram. It is analogous to SMTP, HTTP, or FTP running on top of TCP/IP.

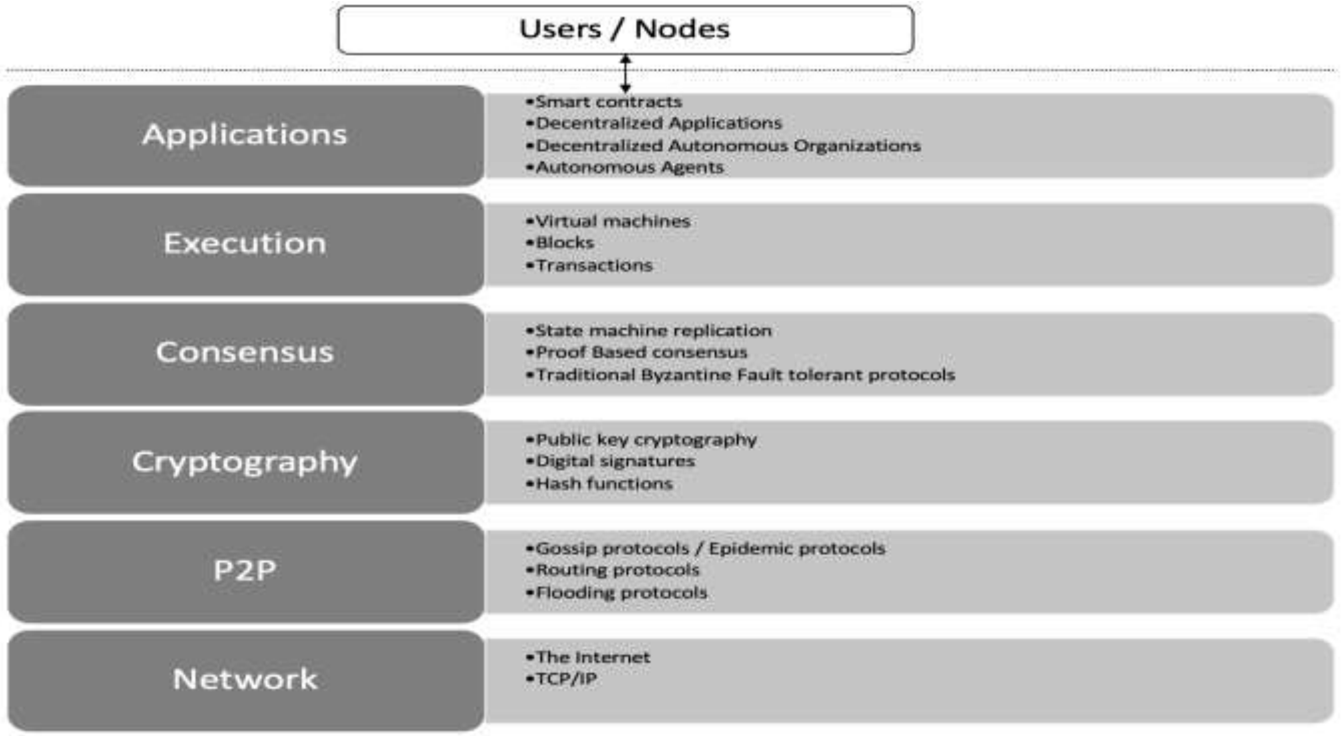


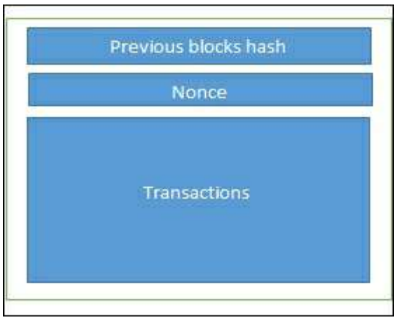
Fig: The network view of a blockchain (Layered Architecture)

At the bottom layer in the preceding diagram, there is the internet, which provides a basic communication layer for any network. In this case, a peer-to-peer network runs on top of the internet, which hosts another layer of blockchain. That layer contains transactions, blocks, consensus mechanisms, state machines, and blockchain smart contracts. All of these components are shown as a single logical entity in a box, representing blockchain above the peer-to-peer network. Finally, at the top, there are users or nodes that connect to the blockchain and perform various operations such as consensus, transaction verification, and processing. These concepts will be discussed in detail later in this book.

From a business standpoint, a blockchain can be ***defined as a platform where peers can exchange value / electronic cash using transactions without the need for a centrally-trusted arbitrator.*** For example, for cash transfers, banks act as a trusted third party. In financial trading, a central clearing house acts as an arbitrator between two trading parties. This concept is compelling, and once you absorb it, you will realize the enormous potential of blockchain technology. This disintermediation allows blockchain to be a decentralized consensus mechanism where no single authority is in charge of the database. Immediately, you'll see a significant benefit of decentralization here, because if no banks or central clearing houses are required, then it immediately leads to cost savings, faster transaction speeds, and trust.

ABHISHEK

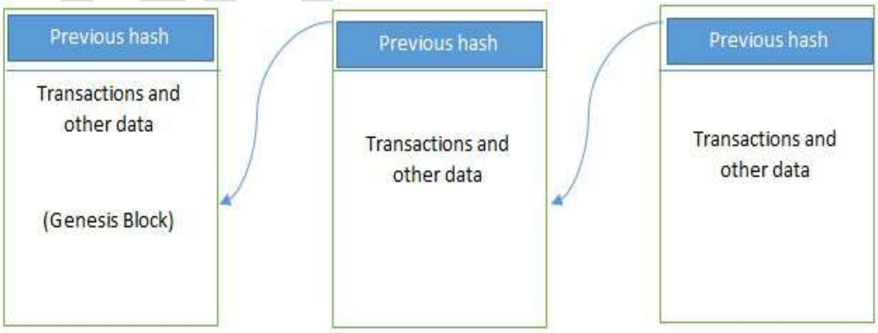
The structure of a block



- **Block is** *merely a selection of transactions bundled together and organized logically.*
- **A transaction** *is a record of an event, for example, the event of transferring cash from a sender's account to a beneficiary's account. A block is made up of transactions, and its size varies depending on the type and design of the blockchain in use.*
- **A genesis block** is the first block in the blockchain that is hardcoded at the time the blockchain was first started.
- **A reference to a previous block** *is also included in the block unless it is a genesis block.*
- **A nonce** *is a number that is generated and used only once. A nonce is used extensively in many cryptographic operations to provide replay protection, authentication, and encryption. In blockchain, it's used in PoW consensus algorithms and for transaction replay protection.*

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.

Generic elements of a blockchain


Previous hash
Transactions and other data
(Genesis Block)
Previous hash
Transactions and other data
Previous hash
Transactions and other data

Generic structure of a blockchain

- **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

ABHISHEK K L

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.

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

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

- **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.

- **Scripting or programming language**
  Scripts or programs perform various operations on a transaction in order to facilitate various functions. For example, in Bitcoin, transaction scripts are predefined in a language called Script, which consist of sets of commands that allow nodes to transfer tokens from one address to another. Bitcoin script language cannot be called Turing complete. In simple words, Turing complete language means that it can perform any computation. It is named after Alan Turing who developed the idea of Turing machine that can run any algorithm however complex. Turing complete languages need loops and branching capability to perform complex computations. Therefore, Bitcoin's scripting language is not Turing complete, whereas Ethereum's Solidity language is.

- **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). EVM is used in Ethereum blockchain, while CVM is a virtual machine developed for and used in an enterprise-grade blockchain called Chain Core.

- **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.

- **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.

ABHISHEK K L

- **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. Smart contracts have many use cases, including but not limited to identity management, capital markets trade finance, record management, insurance, and e-governance. Smart contracts will be discussed in more detail in Chapter 9, Smart Contracts.

Features of a blockchain

- **Distributed consensus:** Distributed consensus is the primary underpinning of a blockchain. This mechanism allows a blockchain to present a single version of the truth, which is agreed upon by all parties without the requirement of a central authority.
- **Transaction verification:** Any transactions posted from the nodes on the blockchain are verified based on a predetermined set of rules. Only valid transactions are selected for inclusion in a block.
- **Platform for smart contracts:** A blockchain is a platform on which programs can run to execute business logic on behalf of the users. Not all blockchains have a mechanism to execute smart contracts; however, this is a very desirable feature, and it is available on newer blockchain platforms such as Ethereum and MultiChain.
- **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.

ABHISHEK K L

A more recent example is Zcash, which will be discussed in more detail in later chapters. 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.

How blockchain works

ABHISHEK K L

How blockchain accumulates 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:

- BlockChain Version 1.0 was introduced in 2005 by Hall Finley, who implements **DLT (Distributed Ledger Technology)** represents its first application based on Crypto currency.
- This type of Version is permission less as any participant will perform valid transaction of Bitcoin.
- This tier was introduced with the invention of Bitcoin, and it is primarily used for cryptocurrencies. Also, as Bitcoin was the first implementation of cryptocurrencies, it makes sense to categorize this first generation of blockchain technology to include only cryptographic currencies.
- All alternative cryptocurrencies as well as Bitcoin fall into this category. It includes core applications such **as payments and applications**. This generation started in 2009 when Bitcoin was released and ended in early 2010.
- In BlockChain 2.0, **BitCoin** is replaced with **Ethereum**. Thus, BlockChain 2.0 was successfully processing high number of Transactions on Public network rapidly.

Blockchain 2.0:

- In this version, the BlockChain is not just limited to Cryptocurrencies but it will extend up to **Smart Contracts.**
- Smart contract is a small computer programs that "live" in the blockchain. They are free computer programs which executed automatically and checked conditions which are defined earlier like facilitation, verification or enforcement. The big advantage of this technology that blockchain offers, making it impossible to tamper or hack Smart Contracts. A most prominent example is the Ethereum Blockchain, which provides a platform where the developer community can build distributed applications for the Blockchain network.

ABHISHEK K L

- This tier includes various financial assets, such as **derivatives, options, swaps, and bonds**. Applications that go beyond currency, finance, and markets are incorporated at this tier.
- **Ethereum, Hyperledger**, and other newer blockchain platforms are considered part of Blockchain 2.0.
- This generation started when ideas related to using blockchain for other purposes started to emerge in 2010.

Blockchain 3.0:

- This third blockchain generation is used to implement applications **beyond the financial services** industry and is used in **government, health, media, the arts, and justice**.
- This generation of blockchain emerged around 2012 when multiple applications of blockchain technology in different industries were researched.
- After Version 2.0, new version was introduced which includes DApps which is known as Decentralized Apps. A DApp is like a conventional app, it can have **frontend written in any language** that makes **calls to its backend,** and its **backend code is running on decentralized Peer-To-Peer Network.** It makes use of decentralized storage and communication which can be Ethereum Swarm etc.
- There are many decentralized Applications like **BitMessage, BitTorrent, Tor, Popcorn**, and Blockchain for supply chain management etc.

Blockchain X.0:

- This generation represents a vision of blockchain singularity where one day there will be a public blockchain service available that anyone can use just like the Google search engine.
- It will provide services for all realms of society. It will be a public and open distributed ledger with general-purpose rational agents (Machina economicus) running on a blockchain, making decisions, and interacting with other intelligent autonomous agents on behalf of people, and regulated by code instead of law or paper contracts. This does not mean that law and contracts will disappear, instead law and contracts will be implementable in code.

Types of blockchain

1. Public blockchain

- Public blockchain is **non-restrictive and permission less**, and anyone with internet access can sign on to a blockchain platform to become an authorized node.
- These blockchains are completely open to following the idea of decentralization.
- Its decentralized nature requires some method for verifying the authenticity of data. That method is a **consensus algorithm** whereby participants in the blockchain reach agreement on the current state of the ledger. **Proof of work (PoW) and proof of stake (PoS) are two common consensus methods**.

Advantages:

1. **Trustable:** There are algorithms to detect fraud. Participants need not worry about the other nodes in the network.
2. **Secure:** This blockchain is large as it is open to the public. In a large size, there is a greater distribution of records.

3. **Anonymous Nature:** It is a secure platform to make your transaction properly at the same time, you are not required to reveal your name and identity to participate.
4. **Decentralized:** There is no single platform that maintains the network, instead every user has a copy of the ledger.

Disadvantages:

1. **Processing [Low Transaction per Second-TPS]:** The rate of the transaction process is very slow, due to its large size. Verification of each node is a very time-consuming process.
2. **Energy Consumption:** Proof of work is highly energy-consuming. It requires good computer hardware to participate in the network.
3. **Acceptance:** No central authority is there so governments are facing the issue of implementing the technology faster.
4. **Scalability:** Since it's a large distributed network which is negative impact on scalability.

Use Cases:

1. Examples of public blockchains are **Bitcoin and Ethereum.**
2. **Raising Funds**
3. **Voting**

2. Private Blockchain

How it works.

- A private blockchain works in a **restrictive environment** like a closed network or is under the control of a single entity.
- They're also known as **permissioned blockchains or enterprise blockchains.**
- While it operates like a public blockchain network in the sense that it uses peer-to-peer connections and decentralization, this type of blockchain is on a much smaller scale.
- Instead of just anyone being able to join and provide computing power, private blockchains typically are operated on a small network inside a company or organization.

Advantages:

The controlling organization sets permission levels, security, authorizations and accessibility. For example, an organization setting up a private blockchain network can determine which nodes can view, add or change data. It can also prevent third parties from accessing certain information.

1. **Speed:** The rate of the transaction is high, due to its small size. Verification of each node is less time-consuming.
2. **Scalability:** We can modify the scalability. The size of the network can be decided manually.
3. **Privacy:** It has increased the level of privacy for confidentiality reasons as the businesses required.
4. **Balanced:** It is more balanced as only some users have access to the transaction which improves the performance of the network.

Disadvantages:

1. **Security:** The number of nodes in this type is limited so chances of manipulation are there. These blockchains are more vulnerable.

2. **Centralized:** Trust building is one of the main disadvantages due to its central nature. Organizations can use this for malpractices.
3. **Count:** Since there are few nodes if nodes go offline the entire system of blockchain can be endangered.

"You can think of private blockchains as being the intranet, while the public blockchains are more like the internet,"

Use Cases:

- "For example, companies may choose to take advantage of blockchain technology while not giving up their competitive advantage to third parties. **They can use private blockchains for trade secret management, for auditing,"**
- Other use cases for private blockchain include **supply chain management, asset ownership** and **internal voting.**
- Examples of **Private Blockchain** are **Hyper ledger** and **corda .**

3. Hybrid Blockchain

How it works.

- Hybrid blockchain combines elements of both private and public blockchain. It lets organizations set up a private, permission-based system alongside a public permissionless system, allowing them to control who can **access** specific **data stored in the blockchain,** and what data will be opened up publicly.

- Typically, transactions and records in a hybrid blockchain are not made public but can be verified when needed, such as by allowing access through a **smart contract.** Confidential information is kept inside the network but is still verifiable. Even though a private entity may own the hybrid blockchain, it cannot alter transactions.
- When a user joins a hybrid blockchain, they have full access to the network. The user's identity is protected from other users, unless they engage in a transaction. Then, their identity is revealed to the other party
- User access information via **smart contracts.**

Advantages. One of the big advantages of hybrid blockchain is that, because it works within a closed ecosystem, outside hackers can't mount a 51% attack on the network. It also protects privacy but allows for communication with third parties. Transactions are cheap and fast, and it offers better scalability than a public blockchain network.

Disadvantages. This type of blockchain isn't completely transparent because information can be shielded. Upgrading can also be a challenge, and there is no incentive for users to participate or contribute to the network.

Use cases.

- Hybrid blockchain has several strong use cases, including real estate. Companies can use a hybrid blockchain to run systems privately but show certain information, such as listings, to the public.
- Retail can also streamline its processes with hybrid blockchain, and highly regulated markets like financial services can also see benefits from using it.

- **Medical records can be stored in a hybrid blockchain. The record can't be viewed by random third parties, but users can access their information through a smart contract. Governments could also use it to store citizen data privately but share the information securely between institutions.**

Note: It provides a greater solution to the healthcare industry, government, real estate, and financial companies. It provides a remedy where data is to be accessed publicly but needs to be shielded privately.

Examples of Hybrid Blockchain are the Ripple network and XRP token.

4. Consortium Blockchain

1. **How it works.** The fourth type of blockchain, consortium blockchain, also known as a federated blockchain, is similar to a hybrid blockchain in that it has private and public blockchain features. But it's different in that multiple organizational members collaborate on a decentralized network. Essentially, a consortium blockchain is a private blockchain with limited access to a particular group, eliminating the risks that come with just one entity controlling the network on a private blockchain.

2. **Advantages.** A consortium blockchain tends to be more secure, scalable and efficient than a public blockchain network. Like private and hybrid blockchain, it also offers access controls.

3. **Disadvantages.** Consortium blockchain is less transparent than public blockchain. It can still be compromised if a member node is breached, and the blockchain's own regulations can impair the network's functionality.

Use cases.

- Banking and payments are two uses for this type of blockchain. Different banks can band together and form a consortium, deciding which nodes will validate the transactions. Research organizations can create a similar model. Consortium blockchain is ideal for supply chains, particularly food and medicine applications.
- Food tracking of the organizations frequently collaborates with their sectors making it a federated solution ideal for their use.
- **Examples of consortium Blockchain are Tendermint and Multichain.**

Consensus

Consensus is the backbone of a blockchain and, as a result, it provides decentralization of control through an optional process known as mining. The choice of the consensus algorithm is also governed by the type of blockchain in use; that is, not all consensus mechanisms are suitable for all types of blockchains. For example, in public permissionless blockchains, it would make sense to use PoW instead of a simple agreement mechanism that is perhaps based on proof of authority. Therefore, it is essential to choose an appropriate consensus algorithm for a particular blockchain project.

Consensus is a process of agreement between distrusting nodes on the final state of data. **To achieve consensus, different algorithms are used**. It is easy to reach an agreement between two nodes (in client-server systems, for example), but when multiple nodes are participating in a distributed system and they need to agree on a single value, it becomes quite a challenge to achieve consensus. This process of attaining agreement common state or value among multiple nodes despite the failure of some nodes is known as distributed consensus.

Note: **Consistency** is achieved using **consensus algorithms** in order to ensure that all nodes have the same copy of the data. This is also called **state machine replication.**

Consensus mechanism

A consensus mechanism is a set of steps that are taken by most or all nodes in a blockchain to agree on a proposed state or value. For more than three decades, this concept has been researched by computer scientists in industry and academia. Consensus mechanisms have most recently come into the limelight and gained considerable popularity with the advent of blockchain and Bitcoin.

There are various requirements that must be met to provide the desired results in a consensus mechanism. The following describes these requirements:

- **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 that no node can make the decision more than once in a single consensus cycle.

Consensus in blockchain

Consensus is a distributed computing concept that has been used in blockchain in order to provide a means of agreeing to a single version of the truth by all peers on the blockchain network. This concept was previously discussed in the distributed systems section of this chapter. In this section, we will address consensus in the context of blockchain technology. Some concepts presented here are still relevant to distributed systems theory, but they are explained from a blockchain perspective.

Roughly, the following describes the two main categories of consensus mechanisms:

1. **Proof-based, leader-election lottery based, or the Nakamoto consensus** whereby a leader is elected at random (using an algorithm) and proposes a final value. This category is also referred to as the fully

ABHISHEK K L

decentralized or permissionless type of consensus mechanism. This type is well used in the Bitcoin and Ethereum blockchain in the form of a PoW mechanism.

2. **BFT (Byzantine fault tolerance)-**based is a more traditional approach based on rounds of votes. This class of consensus is also known as the consortium or permissioned type of consensus mechanism.

BFT-based consensus mechanisms perform well when there are a limited number of nodes, but they do not scale well. On the other hand, leader-election lottery based (PoW) type consensus mechanisms scale very well but perform very slowly.

The consensus algorithms available today, or that are being researched in the context of blockchain, are presented here. The following is not an exhaustive list, but it includes all notable algorithms.

1. **Proof of Work (PoW):**
   - The PoW consensus algorithm involves verifying a transaction through the mining process.
   - The Proof of Work consensus algorithm involves solving a computationally challenging puzzle in order to create new blocks in the Bitcoin blockchain. The process is known as 'mining', and the nodes in the network that engages in mining are known as 'miners'.
   - The incentive for mining transactions lies in economic payoffs, where competing miners are rewarded with 6.25 bitcoins and a small transaction fee.
   - **This scheme** is used in **Bitcoin, Litecoin, and other cryptocurrency** blockchains. **Currently, it is the only algorithm that has proven to be astonishingly successful against any collusion attacks on a blockchain network**, such as the Sybil attack.

2. **Proof of Stake (PoS):**
   - Proof of stake" (PoS) is a blockchain consensus mechanism where users validate transactions and create new blocks on a network by **"staking"** a certain amount of the **cryptocurrency** they hold, essentially proving their ownership and financial interest in the system, allowing them to be selected to participate in validating new blocks on the blockchain; the more coins they stake, the higher their chance of being chosen to validate transactions and earn rewards for doing so.
   - Users who stake coins are called "validators" and are responsible for verifying transaction data.
   - Validators are selected randomly to confirm transactions and validate block information. This system randomizes who gets to collect fees rather than using a competitive rewards-based mechanism like proof-of-work.
   - Compared to "proof of work" (PoW), PoS generally consumes significantly less energy as it doesn't require intensive computational power to validate blocks.
   - Another important concept in PoS is coin age, which is a criterion derived from the amount of time and 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.

3. **Proof of Capacity(PoC)**
   - The PoC mechanism heavily relies on free space available in the hard drive.
   - This is because there are many solutions to a coin's hash problem that a trader needs to store. It is highly efficient as compared to PoW and PoC mechanisms.
   - Coins such as Burst, Storj, SpaceMint and Chia use these mechanisms.

4. **Delegated Proof of Stake (DPoS):**
   - This is an innovation over standard PoS, whereby each node that has a stake in the system can delegate the validation of a transaction to other nodes by voting.
   - It is used in the BitShares blockchain.

ABHISHEK K L

5. **Proof of Elapsed Time (PoET):**
   - Introduced by Intel in 2016, PoET uses a Trusted Execution Environment (TEE) to provide randomness and safety in the leader election process via a guaranteed wait time.
   - It requires the Intel Software Guard Extensions (SGX) processor to provide the security guarantee for it to be secure.
   - It uses a lottery system to decide the next block creator. Thus, it gives a fair chance to all traders to create the next block. It is an efficient process involving utilising lesser resources and low energy consumption.

6. **Proof of Activity (PoA):**
   - This scheme is a combination of PoS and PoW, which ensures that a stakeholder is selected in a pseudorandom but uniform fashion.
   - This is a comparatively more energy-efficient mechanism as compared to PoW. It utilizes a new concept called Follow the Satoshi.
   - In this scheme, PoW and PoS are combined together to achieve consensus and good level of security. This scheme is more energy efficient as PoW is used only in the first stage of the mechanism, after the first stage it switches to PoS which consumes negligible energy.

7. **Proof of Capacity (PoC):**
   - This scheme uses hard disk space as a resource to mine the blocks.
   - This is different from PoW, where CPU resources are used. In in PoC, hard disk space is utilized for mining and as such is also known as hard drive mining.
   - Coins such as Burst, Storj, SpaceMint and Chia use these mechanisms

8. **Proof of Storage (PoS):**
   - This scheme allows for the outsourcing of storage capacity. This scheme is based on the concept that a particular piece of data is probably stored by a node which serves as a means to participate in the consensus mechanism.
   - Several variations of this scheme have been proposed, such as Proof of Replication, Proof of Data Possession, Proof of Space, and Proof of Space-Time.

9. **Proof of Deposit (PoD):** In this case, nodes that wish to participate in the network have to make a security deposit before they can mine and propose blocks. This mechanism is used in the Tendermint blockchain.

10. **Proof of Importance (PoI):** This idea is significant and different from PoS. PoI not only relies on how large a stake a user has in the system, but it also monitors the usage and movement of tokens by the user in order to establish a level of trust and importance. It is used in the NEM coin blockchain. More information about this coin is available at NEM's website https://nem.io.

Reputation-based mechanisms: As the name suggests, a leader is elected by the reputation it has built over time on the network. It is based on the votes of other members.

PBFT (Practical Byzantine Fault Tolerance): This mechanism achieves state machine replication, which provides tolerance against Byzantine nodes. Various other protocols including 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.

ABHISHEK K L

Benefits and limitations of blockchain

Benefits of blockchain technology are as follows:

- **Decentralization:** This is a core concept and benefit of the 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:** Because blockchains are shared and everyone can see what is on the blockchain, this allows the system to be transparent. As a result, trust is established. This is more relevant in scenarios such as the disbursement of funds or benefits where personal discretion in relation to selecting beneficiaries needs to be restricted.

- **Immutability:** Once the data has been written to the blockchain, it is extremely difficult to change it back. It is not genuinely immutable, but because changing data is so challenging and nearly 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 every node, the system becomes highly available. Even if some nodes leave the network or become inaccessible, the network as a whole continues to work, thus making it highly available. This redundancy results in high availability.

- **Highly secure:** All transactions on a blockchain are cryptographically secured and thus provide network integrity. Simplification of current paradigms: The current blockchain model in many industries, such as finance or health, is somewhat disorganized. In this model, multiple entities maintain their own databases and data sharing can become very difficult due to the disparate nature of the systems. However, as a blockchain can serve as a single shared ledger among many interested parties, this can result in simplifying the 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 enabling the quick settlement of trades. Blockchain 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 trusted third party or clearing house is required in the blockchain model, this can massively eliminate overhead costs in the form of the fees which are paid to such parties.

As with any technology, some challenges need to be addressed in order to make a system more robust, useful, and accessible. Blockchain technology is no exception. In fact, much effort is being made in both academia and industry to overcome the **challenges posed by blockchain technology**. The most sensitive blockchain problems are as follows:

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

All of these issues and possible solutions will be discussed in detail in Chapter 18, Scalability and Other Challenges.

ABHISHEK K L

Practice Questions

1. Explain the layered architecture of the generic Blockchain with the help of neat diagram.
2. Discuss the Byzantine general's problem in detail with the help of an Example.
3. Discuss the generic structure of Block and Blockchain.
4. Explain the generic structure of a Blockchain and explain its elements.
5. Explain the benefits and limitations of Blockchain.
6. Define Consensus and explain the different consensus algorithms in detail.
7. Explain the CAP Theorem in the context of Blockchain.
8. Recommend the type of blockchain for the following cases: where (i) users are anonymous (ii) Applications that expects high data throughput.
9. Analyse the Byzantine Generals Problem and explain Practical Byzantine Fault Tolerance solution to the problem.

ABHISHEK K L