CHAPTER NO.1

INTRODUCTON TO BLOCKCHAIN

Q.1. What is Blockchain? Explain the types of Blockchain.

Ans.

Blockchain is nothing but Open kind of distributed ledger which is efficient and verifiable and data is being stored permanently.

Key Characteristics:

• **Open:** Anyone can access blockchain.

• **Distributed or Decentralised:** Not under the control of any single authority.

• **Efficient:** Fast and Scalable.

• **Verifiable:** Everyone can check the validity of information because each node maintains a copy of the transactions.

• **Permanent:** Once a transaction is done, it is persistent and can't be altered.

- **Types of Blockchain:**
  1. **Public Blockchain:**
     - A public blockchain is a permissionless.
     - Anyone who has access to the internet can sign in on a blockchain platform to become an authorized node and be a part of the blockchain network.
     - A node or user which is a part of the public blockchain is authorized to access current and past records, verify transactions or do proof-of-work for an incoming block, and do mining.
     - The most basic use of public blockchains is for mining and exchanging cryptocurrencies. Thus, the most common public blockchains are Bitcoin and Litecoin blockchains. Example: Bitcoin, Ethereum, Litecoin

- **Advantages:**
  1. No one person has control over the data
  2. It can be used in public sectors like healthcare and education
  3. It is decentralized, using peer to peer network of computers.
  4. The validators and participants in the blockchain remain anonymous.

- **Disadvantages:**
  1. Consensus Mechanism: Some public blockchain like Bitcoin uses Proof of Work consensus mechanism where the participants need to solve a complex mathematical puzzle to validate a transaction.
  2. It requires the consumption of a lot of resources which is a costly affair.
  3. In public blockchains, one doesn't need to prove his/her identity and just commit your processing power to become a part of the network.
  4. Speed: This is one of the biggest problems with some public blockchain like bitcoin which process only 4.6 transactions per second whereas companies like Visa process 1700 transactions per second.

2. Private Blockchain:

- A private blockchain is a restrictive or permission blockchain operative only in a closed network.
- Private blockchains are usually used within an organization or enterprises where only selected members are participants of a blockchain network.
- The level of security, authorizations, permissions, accessibility is in the hands of the controlling organization. Thus, private blockchains are similar in use as a public blockchain but have a small and restrictive network.
- Private blockchain networks are deployed for voting, supply chain management, digital identity, asset ownership, etc. Examples of private blockchains are; Multichain and Hyperledger projects (Fabric, Sawtooth), Corda, etc.

- **Advantages:**
  1. **Higher Transactions Per Second (TPS)**
     Private blockchains can process much higher TPS when compared to public blockchains, because of the existence of a few authorized participants in the network.
  2. **Scalability:**
     As only a few nodes are authorized and efficient for managing data in private blockchains, transactions can be processed at a much higher speed. Also, the decision-making process is much faster as compared to the public blockchain.

- **Disadvantages:**
  - **Centralized**: Private blockchain is more centralized as it is generally utilized by smaller groups such as businesses and enterprises.
  - **Security**: When it comes to security, private blockchains are more prone to security threats and other vulnerabilities because it has fewer nodes; thus, bad actors can gain access quickly.
  - **Needs Trust**: Private blockchain requires trust as authorized nodes must be trusted to verify and validate authentic transactions, unlike public blockchain.

3. Consortium Blockchain

• A consortium blockchain is a semi-decentralized type where more than one organization manages a blockchain network.

• This is contrary to what we saw in a private blockchain, which is managed by only a single organization. More than one organization can act as a node in this type of blockchain and exchange information or do mining.

- Consortium blockchains are typically used by banks, government organizations, etc.
- Examples of consortium blockchain are; Energy Web Foundation, R3, etc.

4. Hybrid Blockchain:

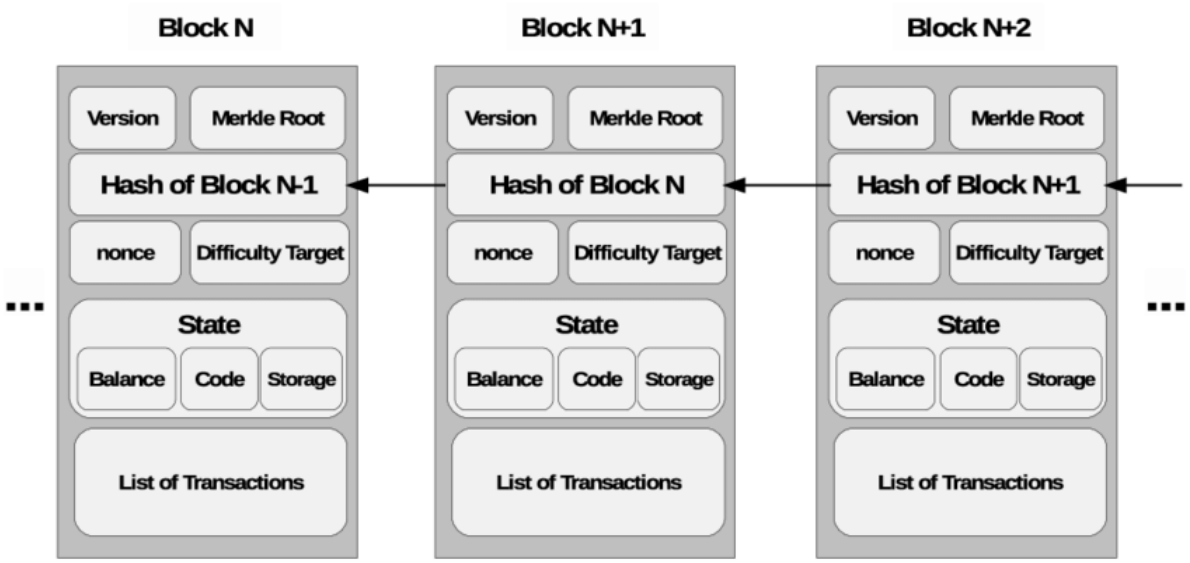
- A hybrid blockchain is a combination of the private and public blockchain.
- Only a selected section of data or records from the blockchain can be allowed to go public keeping the rest as confidential in the private network.
- The hybrid system of blockchain is flexible so that users can easily join a private blockchain with multiple public blockchains.
- The public blockchains increase the hashing and involve more nodes for verification. This enhances the security and transparency of the blockchain network.
- Example of a hybrid blockchain is Dragon chain.

Q.2. Explain Block in Blockchain.

Ans.

- A block in a blockchain network is like a link in a chain. In the field of cryptocurrency, blocks are like records that store valid transactions like a page of a record book and are hashed and encrypted into a hash tree or Merkle tree.
- Blocks are like the building blocks of any blockchain and they are distinguishable from one another as they have different characteristics.
- When a block is completed, it makes room for the following block in the blockchain.
- A block is very secure and is impossible to hack virtually.

- **Structure Of a Block:**



Following are the significant elements of a block:

- **Block Height:** It's the sequence number of the block in the chain of blocks.
- **Block Size:** It's a 4-bytes or 32-bit field that contains the size of the block. It adds size in Bytes.
- **Block Reward:** This field contains the amount rewarded to the miner for adding a block of transactions.
- **Tx Count:** The transaction counter shows the number of transactions contained by the block. The field has a maximum size of 9 bytes.

Q.3. What are the limitations of Blockchain.

Ans.

Limitations Of Blockchain:

1. **Immutable:**

- Immutable means something which cannot be changed or improvised.
- The records of the blockchain cannot be modified once created and broadcasted on the chain network.
- It protects the integrity of data making it tamper-proof, but at the same time, it has its drawbacks.
- No corrections or revision can be done in order to reverse the errors made while entering the data. Once the data is on record, it is imperishable.

2. **Organization and Size:**

- Every transaction needs to be validated or verified by every block over the network in order to reach a mutual consensus.
- A blockchain network may be vast in size, depending upon its growth and dispersion.
- The number of nodes and blocks determines the network size. The bigger the network size, the more time is consumed in the process of reaching consensus in an authentic manner.
- This leads to considerable consumption of time and resources.

3. **Limited Scalibility:**

- Scalability is one of the significant limitations in the blockchain network since each of the nodes needs to verify the transactions processed in the system.

- Because of this, the speed of-processing a transaction gets limited.
- Experts continue to work on distributed ledger technology, which is based on blockchain like Hyperledger fabric to overcome scalability issues.
- There are three aspects of blockchain that have to be addressed — scalability, security, and decentralization.

4. Limited Privacy:

- It is possible to link the user identity with that address and can get information about the user.
- Blockchain transactions are not aligned with one's status as anyone can create a new anonymously and transact through that wallet.
- Identity verification data like security numbers cannot be openly stored in public smart contracts.
- Credential management is another factor that is not managed in an open, ultimately unsecured smart contract.

5. Lack Of Technical Knowledge:

- Even as the blockchain is becoming increasingly popular, most of the investors are finding it difficult to understand the different technical terms, as there is no proper documentation available to help the users.
- Because of this, investors are not able to get their queries resolved. Also, many investors do not know much about Initial Coin Offering (ICC)), which is famous for raising capital in the non-equity market.
- The developers these days are mostly trained in technologies like Java, C++, or Python. The lack of technical expertise in blockchain restricts its development and management.
- Even if it is developed, the users of the blockchain are unaware or incompetent to use its implementation. Thus, awareness about blockchain’s development, implementation, and usage must be spread.

Q.4. What are the challenges of Blockchain?

Ans.

Challenges Of Blockchain:

1. **Transaction Processing Speed (TPS):**

- One of the other challenges faced by blockchain is the scalability in the speed of transaction processing.
- Unlike what people believe, the most popular cryptocurrency Bitcoin can process only an average of 7 transactions per second.
- At the same time, Visa can process 24000 transactions per second, and PayPal can handle about 200 transactions Per second.
- Bitcoin Cash can transact about 50 to 60 transactions per second, but it is still low compared to Visa. Among all the cryptocurrencies, Ethereum is relatively slow and can negotiate only about 20 transactions per second.

2. **Complexity:**

- The technology of blockchain is at its infancy in terms of maturity; this has many new processes, standards, and technical aspects.
- Also, another aspect of blockchain is that it relies on decentralization and cryptography as the main backbone, which further adds to its complexity. Because of this. various challenges arise in its functionalities, such as the speed of transaction.
- The process the transaction, and the limits on data that can be sent or received.
- To have a stable blockchain network, it needs to have many users and nodes connected with a robust network algorithm.

3. **Implementation and Operation Cost:**

- There are additional inputs to consider, but the most influential are:
- Transaction volume
- Transaction Size
- Node hosting methods
- Consensus protocol.
- Each of these uses high computing power with an increase in the size of blockchain with additional nodes and external integrations added. These all add to the operating costs.

4. **Storage Constraints:**

- Under the blockchain, the data is stored by every full node in the network indefinitely since the database in blockchain is append-only and immutable.
- Therefore, storage remains a considerable challenge for practical applications that are built on the blockchain.
- To achieve fast transaction speed in blockchain, the data and the load on the nodes have to be cut down, which is the main challenge. As of now, each node has to store the entire data.
- So if the blockchain can have a mechanism that can allow the nodes to store only the data that is mostly used or locally relevant data for processing transactions, we can process transactions faster. Such solutions can make blockchain more efficient, and there is a need to research on such solutions to overcome the storage constraints.

5. **Energy and Resource Consumption:**

- With additional nodes being added to the blockchain, it grows in size and the number of transactions also increases, which in turn results in the need for more resources and infrastructure to run.
- Miners need to validate every block in a blockchain.

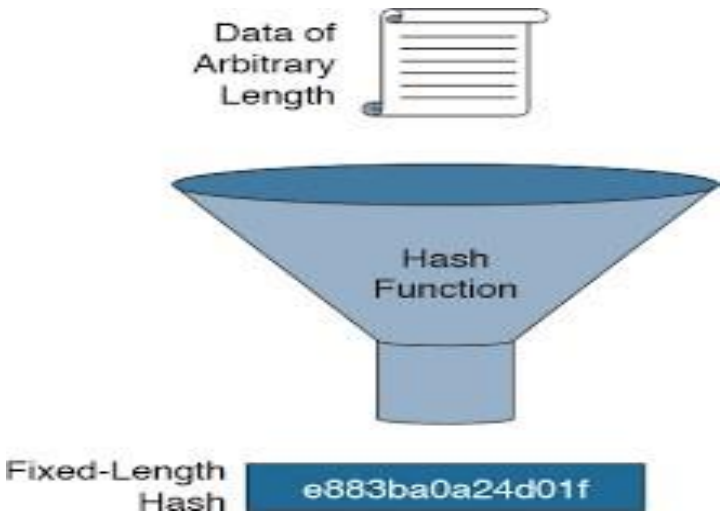
- With an Increase in the size of the blockchain, the time and effort required from miners to verist the network also drastically increases.

- To validate transactions, miners need to come up many mathematical solutions, and they require substantial units of computing power to do that. Since each node has extreme limits offault tolerance to ensure zero downtime.

Q.5. Explain Cryptographic Hash Function.

Ans.

- A cryptographic hash function is a kind of algorithm that can be run on a piece of data, like an individual file or a password, producing a value called a checksum.



- It is commonly used in cryptography since it is a cryptographic function.

Features of Cryptographic Hash Function:

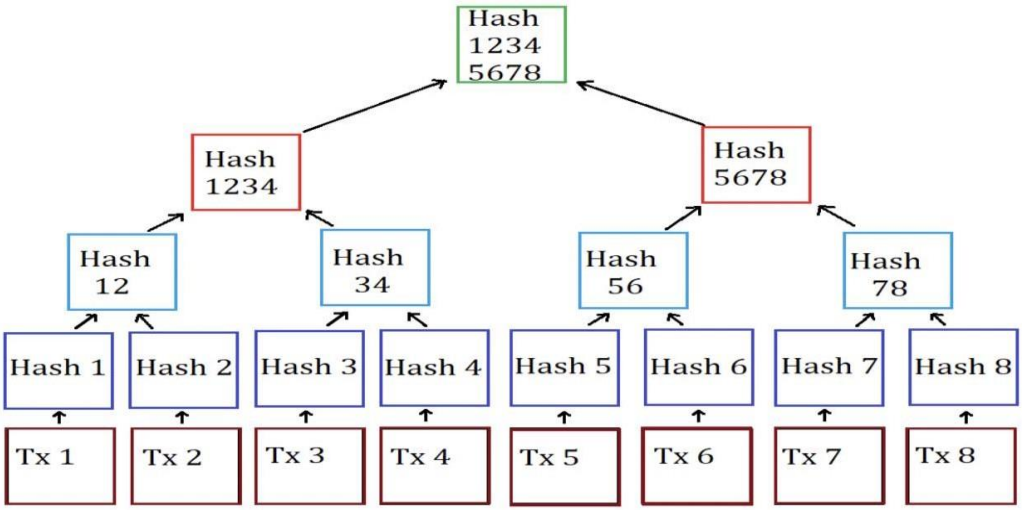
- Fixed Length Output (Hash Value)
- Efficiency in computing
- Collision Resistance, two different messages should not have the same hash value
- Deterministic so the same message always results in the same hash
- Infeasible to generate a message from its hash value except by trying all possible messages.

Q.6. Explain Merkle Tree with diagram.

Ans.

1. A Merkle tree sums up all transactions in a block by generating a digital fingerprint of the whole set of operations, allowing the user to check whether a transaction is included in a block.
2. Merkle trees are created by repetitively hashing pairs of nodes until only one hash is left, this hash is better called the Merkle Root orthe Root Hash.
3. They are constructed from the bottom, from the hashes of individual transactions called Transaction IDs.
4. Thus, every leaf node is a hash of transactional data, and each non-leaf node is a hash of itsprevious hash.
5. Merkle trees are binary and consequently require an equal number of leaf nodes.
6. If the figure of transactions is odd, the last hash will be matched once it creates an even number of leaf nodes.

- **Example of Merkle Tree:**



- We're going to apply to Bitcoin primarily because the usage of Merkle Trees is essential to cryptocurrencies but also simple to understand. For example, if Bitcoin didn't have Merkle Trees, every node on the network would need to maintain a full copy of every transaction that has ever happened on Bitcoin.
- Any authentication request on Bitcoin would take an incredibly large packet of data to be sent over the network, so you need to have it on your own to verify the data.
- A computer used for validation would have to use a lot of processing power to compare ledgers to ensure that there were no changes.
- Merkle Trees fix this problem. They hash records in the accounting, which efficientlysegregates the data proof from the data itself.
- Proving that a transaction is valid only includes giving small amounts of information across the network.

- It allows you to demonstrate that both variants of the ledger are the same for titular amounts of computing power and network bandwidth.

- **Importance Of Merkle Tree:**

1. Merkle Trees are vital because they make Merkle proof possible. These enable us to quickly verify that the input was included in the specific data set and in what order.
2. Merkle Trees are effective, too. They allow us to compress large data sets by removing all unnecessary branches while keeping the only ones we need to prove.
3. In the world of Blockchain, this means that Merkle Trees provides the following critical features:
   - Ability to verify that a transaction is included in a block
   - Light-clients.
   - Full efficiency and scalability
   - Simplified Payment Authentication

➢ **Advantages:**

1. **Efficient verification-** Merkle trees offer efficient verification of integrity and validity of dataand significantly reduce the amount of memory required for verification. The proof of verification does not require a huge amount of data to be transmitted across the blockchain network.
2. **No delay-** There is no delay in the transfer of data across the network. Merkle trees areextensively used in computations that maintain the functioning of cryptocurrencies.
3. **Less disk space-** Merkle trees occupy less disk space when compared to other data structures.

Q.7. Distinguish between Public, Private and Hybrid Blockchain.

Ans.

	Public	Private	Hybrid
Definition	The public blockchain is open to everyone where anyone can participate.	Private blockchain is controlled by owners and access is limited to certain users.	The hybrid blockchain is a combination of the public and private blockchain. This means that some process is kept private and others public.
Transparency	The public blockchain is completely transparent.	he private blockchain is only transparent to the users who are granted access.	Hybrid blockchain transparency depends on how the owners set the rules.
Incentive	Public blockchain incentivizes participants for growing the network.	The private blockchain is limited and hence have no similar incentive as that of a public blockchain.	Hybrid blockchain can opt to incentivize users if they want to.
Use-case	Can be used in almost every industry. Good for public projects. It is also good for creating cryptocurrency for commercial use.	Private blockchain is great for organization blockchain implementation as they require complete control over their workflow.	Hybrid is best suited for projects that can neither go private or public and have a lack of trust. The supply chain is a great example. It is also effective in banking, finance, IoT, and others.
KYC needed	No	Yes	Yes
Transactional Cost	Costly	Not so costly	Not so costly
Carries basic property of blockchain	Yes	Yes	Yes

Q.8. Difference between Public, Private and Consortium Blockchain.

Ans.

Characteristics	Public Blockchain	Private Blockchain	Consortium Blockchain
Permission Read	Public Class	Could be public or restricted	May be public or restricted
Determination of Consensus	All miners	Only one organization	Designated set of nodes
Efficiency	Low	High	High
Immutability	Impossible to tamper	Could be tampered	Could be tampered
Centralized	No	Yes	Partial
Consensus	Permissionless	Permissioned	Permissioned

CHAPTER NO.2

CRYPTOCURRENCY

Q.1. What is Cryptocurrency? Explain the Types of Cryptocurrency.

Ans.

1. Cryptocurrencies are digital currency designed to be faster, cheaper, and more reliable than money.
2. Cryptocurrency is an open-source encrypted ledger of transactions built on blockchain technology.
3. On the blockchain network, the cryptocurrency is created via mining, users can transact directly with each other over the internet/online platforms.
4. Cryptocurrency users can record and verify each other's transactions, and the acceptance of the network is mandatory before it can be committed on to the blockchain ledger.

- **Types of cryptocurrencies:**

- **Bitcoin**
- **Altcoin**
- **Tokens**

- **Bitcoin:**

1. Bitcoin was introduced to the public in 2009 by an anonymous developer or group of developers using the name Satoshi Nakamoto.
2. Bitcoin is a cryptocurrency, a virtual currency designed to act as money and a form of payment outside the control of any one person, group, or entity, and thus removing the need for third-party involvement in financial

transactions. It is rewarded to blockchain miners for the work done to verify transactions and can be purchased on several exchanges.

3. It has since become the most well-known cryptocurrency in the world. Its popularity has inspired the development of many other cryptocurrencies. These competitors either attempt to replace it as a payment system or are used as utility or security tokens in other blockchains and emerging financial technologies.

- **Altcoin:**

1. Altcoins or coins are, in some instances, also referred to as 'currency tokens' that should not be confused with the broader term of tokens.
2. The main features that are common to all altcoins are:

• They are peer-to-peer digital currencies that involve a mining process.

• They possess their independent blockchain.

• They possess the characteristics of money, i.e., they are fungible, divisible and have limited supply, and typically meant to operate only as a means of payment.

- **Tokens:**

1. Bitcoin was the coin that introduced us to the world of cryptocurrencies.
2. At the time, it was the only peer-to-peer transaction system for currencies.
3. Eventually, others types of coins came into existence which were called altcoins, as they were created as an alternative to Bitcoin.
4. With time, many new and different types of cryptocurrencies were invented to support different applications. The creation and growth of Ethereum gave birth to the term “token” which soon became a universal term for all currencies developed on the Ethereum blockchain.
5. Depending on the application and functions of different tokens, they were further divided into different categories, as explained below.

- **Utility Token**
- **Security Token**

- **Utility Token:**

1. Utility tokens offer the holder the right or license to use a product or service, e.g., ETH (Ethereum). Issued during an ICQ these tokens can, in the future, be redeemed to access the product or services of the company or project.
2. They are considered free of regulatory restrictions as they are not built to function as an investment instrument, but to facilitate the funding of the ICOs.
3. The token holder may have some rights within the ecosystem.
4. However, they do not have any decisioning power in the direction that the company or project can move.

- **Security Token:**

1. Security tokens, also referred to as Equity tokens, entitle the holder to a share or in the company/startup. In other words, they can be considered as a digitized, tokenized form of the traditional securities.
2. Security tokens are issued during STO or Security Token Offering, which is a variation of ICO with more regulatory and due diligence controls fostering investor confidence.
3. They are seen as an investment opportunity. As the tokens represent percentage ownership of the company, including voting rights and decision-making power, they are subject to regulatory restrictions.
4. Tokens can be broadly classified as utility and security, there is the emergence of hybrid tokens that takes the best of both worlds. It can combine the advantages of both utility tokens and security tokens, as incentives are distributed across all participants

Q.2. What are the characteristics of Cryptocurrency.

Ans.

1. **Decentralized:**

- Fiat currencies are issued by the government and regulated by the Central Bank. Thus, government-issued policies and control of supply can affect the value of the currency cryptocurrency, on the other hand, is not backed or controlled by any bank or central government.
- All the nodes/computers in the network work together in mining or processing a transaction. This decentralized feature means that no central body can control or influence the value of the cryptocurrency.

2. **Form Of Existence:**

- While fiat can in both physical (coins and paper notes) and digital forms that allow for electronic of money, cryptocurrency can only exist in digital form.
- They are not tangible and are essentially developed by software code and cryptographic algorithms.

3. **Limited Supply:**

- Another key feature of cryptocurrency is that the supply is limited.
- The maximum supply of cryptos that can ever be generated or mined is defined when the genesis block is created.

4. **Global Access:**

- The decentralized nature of blockchain allows anyone to access and transact in cryptocurrency irrespective of their geographical location so long as they have access to the Internet.

- There are no cross-border restrictions and their use is not limited to only those having access to banks. This allows for faster processing times and very low transaction fees as third-party intervention is removed.

5. **Anonymity and Transparency:**

- The personal details are stored in traditional databases, and these transactions can be monitored, tracked, and retrieved by the central authority.
- Privacy and anonymity is a major concern. In cryptocurrency, a transaction is linked to the person's cryptocurrency addresses and not the person's name, address, or any other personal details. Thus, anonymity is maintained.

6. **Impossible to Duplicate:**

- The Underlying blockchain technology of the cryptocurrency is that of decentralization.
- Cryptographic encryption and consensus protocols make counterfeiting cryptocurrency impossible.

Q.3. What is the difference between Altcoin, Utility Token and Security Token.

Ans.

	Altcoin	Utility Tokens	Security Tokens
Origin	They are built from scratch, either using the open source code of an existing cryptocurrency or using original code. They exist on their separate blockchain	They are built from templates of an existing blockchain via smart contract	They are built from templates of an existing blockchain via smartcontract
Purpose	Means of payment	Created for funding blockchain projects/ companies Distributed during an ICO	Created for funding blockchain projects/ companies Distributedduring an ICO
Usage	Acts as currency or cash	Represents the right to use a product or service in the future	Acts as an investment product, it represents the shares of the company
Value	Value is dictated by market supply and demand	Used as a utility and not linked to the performance of the company. The users get voting and other usage rights within the specific blockchain it operates.	Gives part ownership to the token holders. Its value is directly linked to the valuation of the company

Security	Protection as strong as the strength of the blockchain network protocols and user diligence	Suspectable to ICO scams and frauds	The strict regulations aim to protect investors from scams and frauds.
Regulation	Typically, not regulated or controlled by any bank or central authority	Typically, unregulated	Strictly regulated by financial marketauthority

Q.4. Explain Cryptocurrency wallet and its types.

Ans.

- A digital wallet or cryptocurrency wallet is a software program that stores the users private and public keys enabling the user to transact crypto assets.
- It is a management system that interacts with various blockchains to enable users to send and receive digital currency and monitor their balance.
- Cryptocurrencies are stored immutably on the blockchain using the user's public key. This public key is used by other wallets to send funds to the user's wallet address.
- However, the private key is required if users want to spend cryptocurrency from their address.

➢ **Types Of Wallet:**

- **Hot Wallet:**

1. A hot wallet is designed for online day-to-day transactions. It is connected to the internet at all times and hence a strong candidate for hackers. For this very reason, it is not advisable to use a hot wallet for long-term storage.

2. Based on their installation characteristics, they are differentiated into three types as follows.

1. **Desktop Wallets**
2. **Mobile Wallets**
3. **Online Wallets**

1. Desktop Wallet:

- Desktop wallets, as the name suggests, can be downloaded and installed on your desktop or laptop. As they are locally stored, the user has complete control of the wallet.
- They come with a variety of features, including build-in exchange platforms, multi-currency support, etc.
- The downside, however, is that the wallet is dependent on the security features installed on your computer.
- Hence, if the computer gets hacked or is virus-infected, there is the risk of losing all your funds.

2. Mobile Wallet:

- Mobile wallets are designed to operate on smartphone devices. Once installed, the smartphone application operates just like its desktop counterpart, but it may not have all the savvy features.
- However, it circumvents the constraints of the desktop wallet with its ease of accessibility, as you can easily carry it with you wherever you go and make purchases from merchants who accept cryptocurrency payments using QR codes.
- However, they are more vulnerable to malicious apps and viruses than desktop wallets.
- Hence, it is recommended that security features like wallet encryption are installed on the mobile devices and adequate backup is provided for private keys in case the smartphone is lost or broken.

3. Online Wallets:

- Online wallets are also called web wallets. They run on the cloud, and hence the user does not need to download or install any application.
- They can be accessed from any computing device via a web browser.
- They are the most convenient of the hot wallets and preferred by fresh cryptocurrency users.
- Online wallets are hosted and controlled by a third party and hence are the most vulnerable. Though some service providers offer to manage your private keys on your behalf, many web wallets have come with the option where the user can take full control of the keys or share control via multiple signatures. MyEtherWallet, GreenAddress, and MetaMask are examples of online wallets.

4. Cold Wallet:

- A cold wallet is a digital wallet that is not connected to the internet. Unlike hot wallets, they are not free.
- Being offline, they are more secure and used for long-term storage of Cryptocurrencies. Hardware wallets and paper wallets are the two types of cold wallets.

(i) Hardware Wallet:

- Hardware wallets are physical, electronic devices that use Random Number Generator (RNG) to generate the public/private key that is stored in the device.
- It connects to the internet whenever the user needs to send or receive payments and disconnects once the transaction is executed.
- Transactions are confirmed through the private keys that are saved offline.

- Hardware wallets have the facility to generate a PIN to protect the device as well as a recovery phrase in case the wallet is lost.

- Though more secure than hot wallets, they are less user-friendly and difficult to access. Trezor and Ledger wallets are popular hardware wallets.

(ii) Paper Wallet:

- Paper wallets are offline and, as the name suggests, it is a piece of paper with the crypto address and its private key is physically printed out in the form of codes. These codes can then be scanned to execute cryptocurrency transactions.
- Paper wallets are completely secure as the question of hacking does not arise with paper wallets.
- However, a big disadvantage is that the paper wallet contains only a single public/private pair, and hence they can be used only once for the whole amount in one transaction. If you spend a partial amount, you will lose the remaining balance.
- Also, they need to be kept safe from water, fire, wear and tear, or anyone photographing the QR code.

Q.5. Difference between Hot Wallet and Cold Wallet.

Ans.

Sr. No.	Hot Wallet	Sr. No.	Cold Wallet
1.	Online Storage: Connected to Internet.	1.	Offline Storage: Not connected to Internet.
2.	Recommended for short term storage, convenient for regular day to day transaction.	2.	Recommended for long term storage.

3.	Easily accessible and preferred by traders and new users.	3.	Not easily accessible.
4.	Free usage	4.	Need to be purchased.
5.	Susceptible to cyber-attacks and malware.	5.	No risk of hacks or malware from internet connections, however it should be physically protected from loss or breakage.
6.	Types of Wallets are: Software wallets, online wallets.	6.	Types of Wallets are: Hardware Wallets and Paper Wallets.

Q.6. What are the uses of Cryptocurrency?

Ans.

1. **Ecosystem Players:**

- A blockchain is only as strong as its community. If the players in the community are honest and engaged, it makes for a sturdy and sustainable cryptocurrency ecosystem.
- The different players or actors that directly or indirectly constitute the blockchain ecosystem or community.

2. **Cryptomining:**

- A miner needs to have some level of technical knowledge and expertise in setting up computing software and equipment. Blockchains vary in the mining systems that they use.
- However, they all have some form of a consensus algorithm and an incentive system.

3. **Airdrop:**

- Airdrops are a marketing strategy that crypto projects employ to incentivize the use of their platform.

- New projects may airdrop crypto into your wallet as part of an initial offering, or as a reward for promoting the brand. Airdrops are a way to acquire digital currency without buying it.

4. Token Or Coin Burning:

- Token or Coin burning is a process of permanently removing coins out of circulation to reduce the total supply.
- The transaction is transparent on the blockchain for anyone to confirm that the coin(s) were sent to the address, but at the same time, the address does not belong to anyone and has no practical value.

Q.7. Explain Consensus Mechanism in Blockchain.

Ans.

1. Proof-Of-Work [POW]:

- Proof-of-Work, is the original consensus algorithm in a Blockchain network. In Blockchain, this algorithm is used to confirm transactions and produce new blocks to the chain.
- With PoW, miners compete against each other to complete transactions on the network and get rewarded. In a network users send each other digital tokens.
- A decentralized ledger gathers all the transactions into blocks. However, care should be taken to confirm the transactions and arrange blocks. This responsibility bears on special nodes called miners, and a process is called mining.
- The main working principles are a complicated mathematical puzzle and a possibility to easily prove the solution.

2. Proof-Of-Stake [POS]:

- Proof of Stake, or PoS, is a blockchain protocol that validates transactions and protects the blockchain from 51% attacks.
- Its purpose is identical to the Proof of Work protocol.
- Instead of workers (nodes) using computational power to validate transactions, consequently creating a negative impact on the environment with a lot of electricity used, Proof of Stake requires nodes to stake the balance. These nodes then validate the transactions.

3. Proof-Of-Burn [POB]:

- There are various versions of Proof of Burn in blockchain, with the most acknowledged version being Iain Stewart's algorithm.
- He's also the inventor of the Proof of Burn consensus mechanism. Here, the concept of "burning the coins" means investing the native coins in virtual mining rigs (mining powers).
- It allows miners with the most virtual mining rigs or a miner who invested the most coins – to add his new block of transactions to the network.
- Hence, the number of burnt coins shows miners' commitment to the network.

4. Proof-Of-Elapsed Time [POET]:

- PoET is a consensus algorithm used in a permissioned blockchain network to decide mining rights and next block miner.
- A permissioned blockchain network requires participants to prove their identity, whether they are allowed to join. Hence, it needs permission (or invitation) to join the decentralized network as a new participant.
- Intel associated with Linux Foundation in the development of Hyperledger Sawtooth.
- They aimed to build a highly scalable private blockchain network.

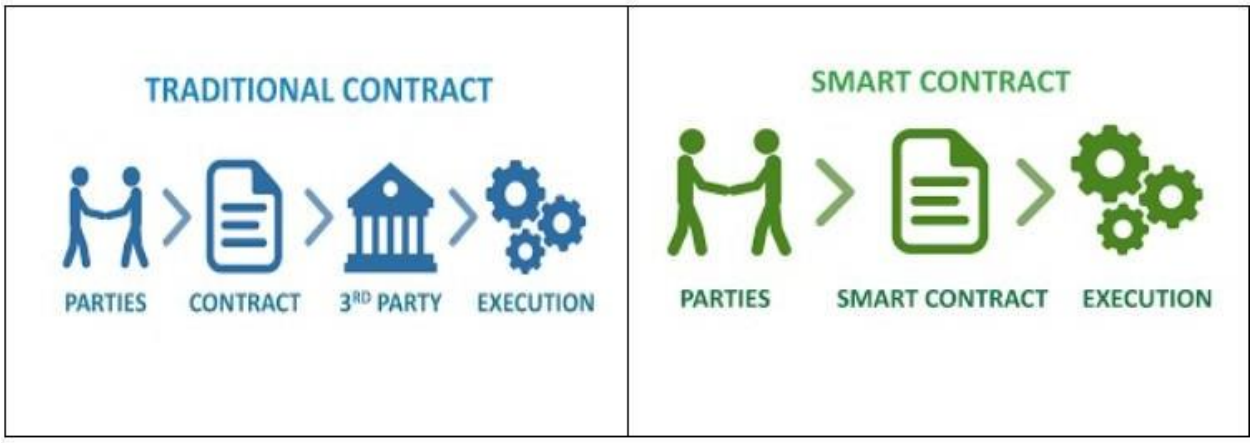
CHAPTER NO.3

SMART CONTACT

Q.1. What is Smart Contract? What are the benefits of Smart Contract?

Ans.

1. A smart contract is a computer program or a transaction protocol that is intended to automatically execute, control, or document legally relevant events and actions according to the terms of a contract or an agreement.
2. The main contribution of smart contract is making the blockchain applications programmable, and it brought the blockchain applications beyond just transfer currency.
3. It makes the verification of the terms of any agreement automatic.
4. Thus all operations, which depend on a condition, can be figured out without any middleman or third party's intervention.

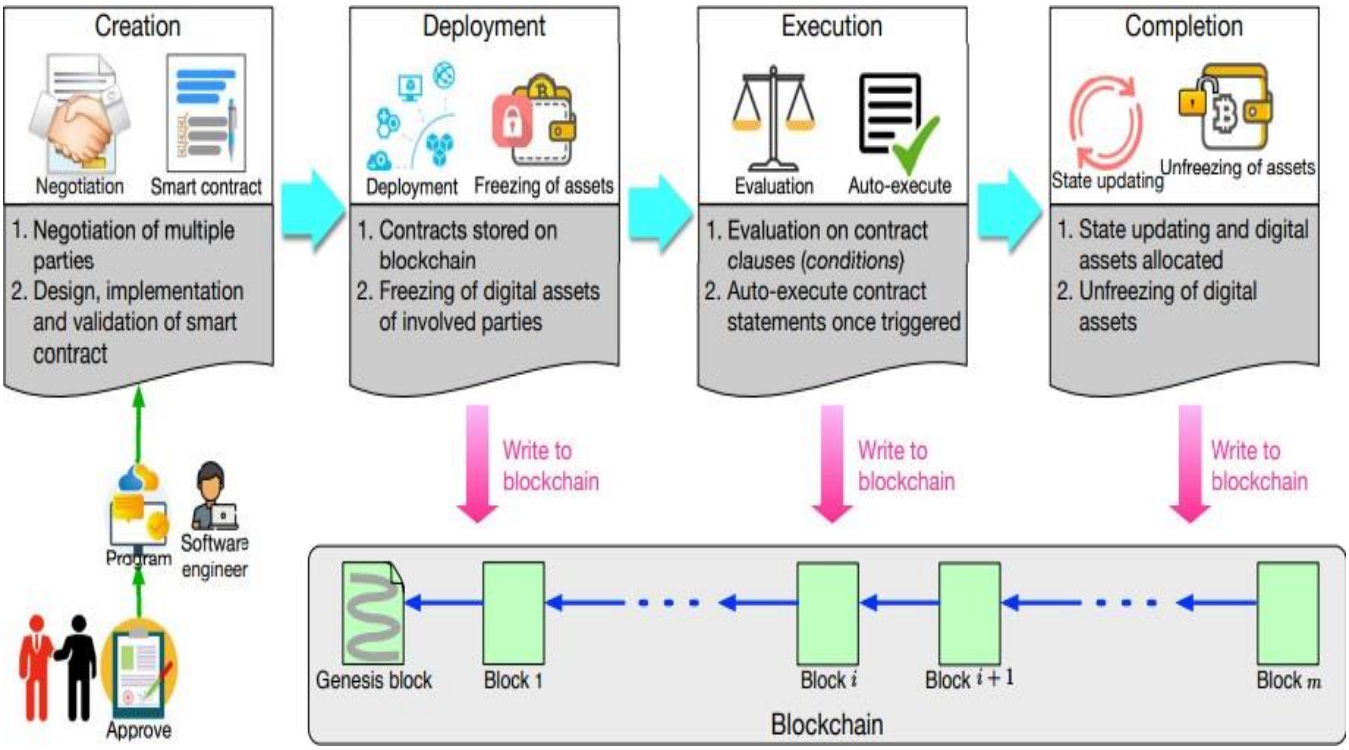


- ## Benefits Of Smart Contract:

- **Transparency:** Since smart contracts are deployed to the Ethereum network like a transaction after compile,these contracts are fully accessible and visible to all the relevant parties.
- **Security:** The conditions and environments of where to keep smart contracts are a secure place, much ascryptocurrencies have.
- **Trust:** By keeping smart contracts in a place that is as secure as cryptocurrencies, the secure, autonomous, andtransparent nature of blockchain can be provided for smart contracts too. Knowing that the possibility of makingmanipulations on information placed in a block is very low helps us to trust.
- **Speed:** Smart contracts live on the internet and run on software code. Thus it verifies transactions very fast. This speed can save many wasted hours caused by humans.
- **Saving:** Smart contracts eliminate the need for having a middleman or third-party authority to determine whatto do according to the terms of the agreement. Thus, they save wasted money besides wasted time, too. That means there is no need for lawyers, banks, witnesses, and any other intermediaries.

Q.2. Explain the Life Cycle of Smart Contract?

Ans.



Creation of smart contracts.

- Several involved parties first negotiate on the obligations, rights and prohibitions on contracts.
- After multiple rounds of discussions and negotiations, an agreement can reach. Lawyers or counselors will help parties to draft an initial contractual agreement.
- Software engineers then convert this agreement written in natural languages into a smart contractwritten in computer languages including declarative languages and logic-based rule languages .
- Similar to the development of computer software, the procedure of the smart contract conversion iscomposed of design, implementation and validation (i.e., testing).
- It is worth mentioning that the creation of smart contracts is an iterative process involving with multiple rounds of negotiations and iterations. Meanwhile, it is also involved with multiple parties, such as stakeholders, lawyers and software engineers

Deployment of smart contracts.

- The validated smart contracts can then be deployed to platforms on top of blockchains.
- Contracts stored on the blockchains cannot be modified due to the immutability of block-chains.
- Any emendation requires the creation of a new contract. Once smart contracts are deployed onblockchains, all the parties can access the contracts through the blockchains.

- Moreover, digital assets of both involved parties in the smart contract are locked via freezing thecorresponding digital wallets.
- For example, the coin transfers (either incoming or outgoing) on the wallets relevant to the contractare blocked. Meanwhile, the parties can be identified by their digital wallets.

Execution of smart contracts.

- After the deployment of smart contracts, the contractual clauses have been monitored and evaluated.
- Once the contractual conditions reach (e.g., product reception), the contractual procedures (orfunctions) will be automatically executed.
- It is worth noting that a smart contract consists of a number of declarative statements with logicalconnections.
- When a condition is triggered, the corresponding statement will be automatically executed,consequently a transaction being executed and validated by miners in the blockchains .
- The committed transactions and the updated states have been stored on the blockchains thereafter.

Completion of smart contracts.

- After a smart contract has been executed, new states of all involved parties are updated. Accordingly,the transactions during the execution of the smart contracts as well as the updated states are stored in blockchains.
- Meanwhile, the digital assets have been transferred from one party to another party (e.g., money transfer from the buyer to the supplier). Consequently, digital assets of involved parties have been unlocked. The smart contract then has completed the whole life cycle.

Q.3. Explain the Types of Smart Contract.
Ans.

1. **Smart Legal Contract:**

- The most common type of smart contract, a smart legal contract, involves similar legal requirements (i.e.- mutual assent, expressed by a valid offer and acceptance; adequate consideration; capacity; and legality) to itstraditional counterpart, and it is put in place to hold concerned parties accountable for fulfilling their end of an agreement.
- When set up properly, a smart contract is legally enforceable and requires the parties to fulfill their obligations;failure to fulfill obligations in the

contract may result in legal action that can be automatically triggered by thesmart contract against the party.

2. Decentralized Autonomous Organizations:

- DAO stands for decentralized autonomous organization, which is a fancy term for a group of people who agree to abide by certain rules for a common purpose. Those rules are written into the code of the organizationvia smart contracts—algorithms that run when certain criteria are met.
- It can be defined as communities that exist on the blockchain. These communities can be defined by a set of agreed upon rules which are coded via smart contracts. Every participant and their actions are subject to the community's rules with the task of enforcing these rules. These rules are made up of many smart contracts andwork together to watch over activities in the community.

3. Application Logic Contracts:

- Application Logic Contracts—or ALCs—contain an application-based code that remains in-step with other blockchain contracts.
- They enable communication across different devices, such as the merging of the Internet of Things (IoT) withblockchain technology. ALCs are a pivotal piece of multi-function smart contract and mostly work under a managing program.

4. DApps (Distributed or Decentralized Apps):

- These software applications run on a P2P environment and are not hosted on a central server. They use blockchain to store data, and as such the program is designed in a way that it's not controlled by any single entity.
- Smart Contracts need a network to function on, and DApps helps integrate their usage efficiently.

Q.4. What are the uses and limitations of Smart Contract?

Ans.

- Uses of Smart Contracts:

- Smart contracts can be used in a variety of fields, from healthcare to supply chain to financial services. Someexamples are as follows:

1. Government voting system:

- Smart contracts provide a secure environment making the voting system less susceptible to manipulation. Votes using smart contracts would be ledger-protected, which is extremely difficult to decode.
- Moreover, smart contracts could increase the turnover of voters, which is historically low due to the inefficientsystem that requires voters to line up, show identity, and complete forms. Voting, when transferred online using smart contracts, can increase the number of participants in a voting system.

2. Healthcare:

- Blockchain can store the encoded health records of patients with a private key. Only specific individuals wouldbe granted access to the records for privacy concerns. Similarly, research can be conducted confidentially andsecurely using smart contracts.

- All hospital receipts of patients can be stored on the blockchain and automatically shared with insurance companies as proof of service. Moreover, the ledger can be used for different activities, such as managing supplies, supervising drugs, and regulation compliance.

3. Supply chain:

- Traditionally, supply chains suffer due to paper-based systems where forms pass through multiple channels toget approvals. The laborious process increases the risk of fraud and loss.

- Blockchain can nullify such risks by delivering an accessible and secure

digital version to parties involved inthe chain. Smart contracts can be used for inventory management and the automation of payments and tasks.

4. Financial services:

- Smart contracts help in transforming traditional financial services in multiple ways. In the case of insurance claims, they perform error checking, routing, and transfer payments to the user if everything is found appropriate.
- Smart contracts incorporate critical tools for bookkeeping and eliminate the possibility of infiltration of accounting records. They also enable shareholders to take part in decision making in a transparent way. Also, they help in trade clearing, where the funds are transferred once the amounts of trade settlements are calculated.

➢ **Limitations Of Smart Contract:**

1. **Difficult to change**

   Changing smart contract processes is almost impossible, any error in the code can be time-consuming and expensive to correct.

2. **Possibility of loopholes**

   According to the concept of good faith, parties will deal fairly and not get benefits unethically from a contract.However, using smart contracts makes it difficult to ensure that the terms are met according to what was agreed upon.

3. **Third party**

   Although smart contracts seek to eliminate third-party involvement, it is not possible to eliminate them. Thirdparties assume different roles from the ones they take in traditional contracts. For example, lawyers will not be needed to prepare individual contracts; however, they will be needed by developers to understand the termsto create codes for smart contracts.

4. **Vague terms:**

   Since contracts include terms that are not always understood, smart contracts are not always able to handle terms and conditions that are vague.

BLOCKCHAIN
CHAPTER NO.4
PUBLIC BLOCKCHAIN

Q.1. What is Public Blockchain? What are its characteristics?
Ans.

- **Public Blockchain:**

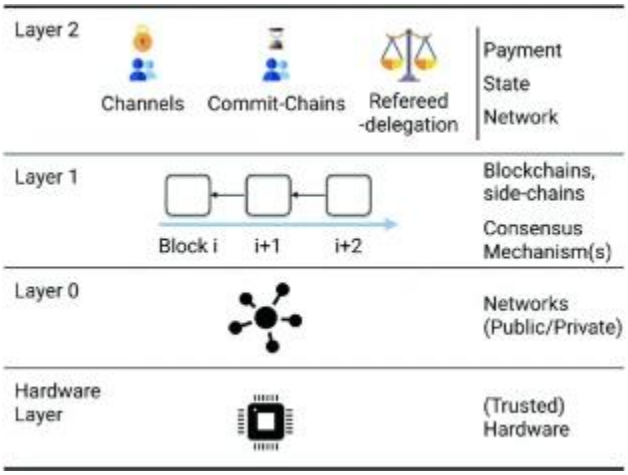
1. Public Blockchain are open networks that allow anyone to participate in the network i.e. public blockchain is permissionless.
2. In this type of blockchain anyone can join the network and read, write, or participate within the blockchain.
3. In a public blockchain, one doesn't need any permission to initiate or access a transaction or participate in a consensus process in order to create a block.
4. There is anonymity maintained in a public blockchain with the help of high cryptographic protocols.

- **Characteristics Of Public Blockchain:**

1. It is an open network where nodes can join and leave without the permission of anyone.
2. All nodes in the network can verify a new piece of data added to the network.
3. It is secure to the 51% rule.
4. There is no need to use your real name or identity, everything can be hidden.

Q.2. Explain the Layers of Blockchain in detail.

Ans.



1. **Hardware Layer:**
- Blockchain are Peer-to-Peer (P2P) networks that allow clients to connect with peer clients to make data sharing faster and easier.
- It is nothing but a vast network of devices communicating with each other and requesting data from one another.

2. **Layer 0:**
- Layer 0 is the initial stage of blockchain that allows various networks to function, such as Bitcoin, Ethereum.
- It also provides blockchain with a facility of cross-chain interoperability communication from top to different layers.

3. **Layer 1:**
- Layer1 blockchain is an advancement in layer 0. Under this layer, the blockchain network is maintained functionally.
- Any changes and issues arising in the new protocol in layer 0 will also affect layer 1. Example: Bitcoin, Ethereum, Ripple etc.

4. Layer 2:

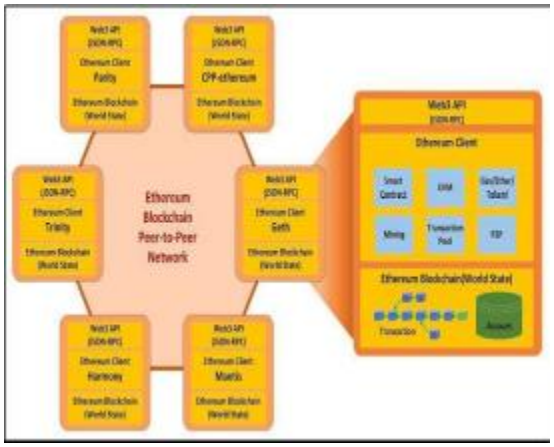
- Layer 0 has many interactions that have been removed by layer 2.
- It works with third-party integration and removes the limitations of layer 1.

5. Layer 3:

- Layer 3 blockchain is also referred to as the "application layer".
- Here, the blockchain protocol is split into two significant sub-layers, that being, application and execution.

Q.3. Describe the Architecture Of Ethereum in detail.

Ans. Architecture Of Ethereum:



1. Every Ethereum node runs an EVM and all the smart contracts run within this virtual machine.
2. Client applications can connect to an Ethereum client using a web3j SDK(s/w development kit).
3. It has to address smart contracts, Ether value that will be transferred to the recipient, nonce, gas, gas limit, and signature.

4. **Some of the Critical Ethereum components are:**

(i) **Mining Nodes:** Mining nodes are responsible for writing all transactions that have occurred in the Ethereum network in the block.

(ii) **EVM (Ethereum Virtual Machine):** EVM is the runtime environment for smart contracts in Ethereum. It focuses on providing security and executing untrusted code by computers all over the world.

(iii) **Ether:** Ether is the transactional token that facilitates operations on the Ethereum network. Ether is a form of payment for network participants to execute their requested operations on the network.

(iv) **Gas:** Gas in Ethereum is a unit of measurement used to measure the work done by Ethereum to carry out transactions or any interaction within the network.

(v) **Transaction:**

(vi) **Accounts:** Ethereum has two types of accounts: Externally controlled account and contract account.

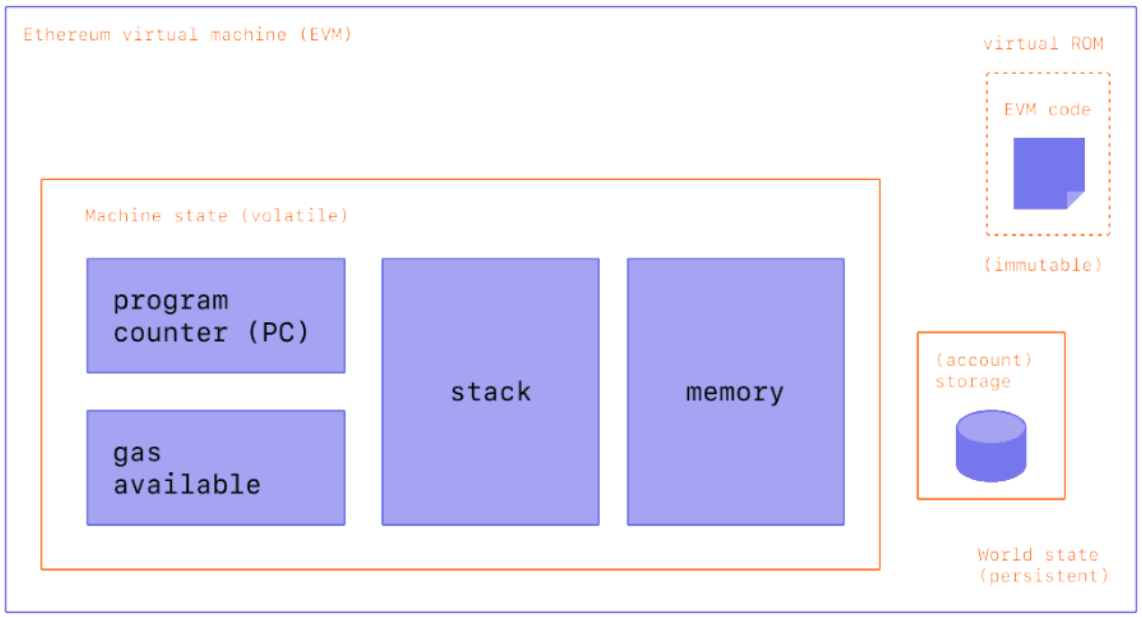
Ethers are stored in the externally controlled account. It is possible to transfer ether from one externally controlled account to another account.

Q.4. What is EVM (Ethereum Virtual Machine). Explain in detail.

Ans.

- **EVM (Ethereum Virtual Machine):**
- EVM is the runtime environment for smart contracts in Ethereum.
- It focuses on providing security and executing untrusted code by computers all over the world.

- EVM specialised in preventing Denial-of-Service attack and ensures that programs do not have access to each other's state, ensuring communication can be established without any potential interference.
- EVM has been designed to serve as a runtime environment for smart contracts based on Ethereum.



- Its main purpose is to compute the network's state and to run and compile various types of smart contract code into a readable format called 'Bytecode.'
- During execution, the EVM maintains a transient memory (as a word-addressed byte array), which does not persist between transactions.
- Ethereum has the same approach as JVM but in a different smart contract language called Solidity.
- These contracts are then converted into bytecodes and uploaded on the blockchain for execution on the EVM (Ethereum Virtual Machine) running on various hardware platforms.

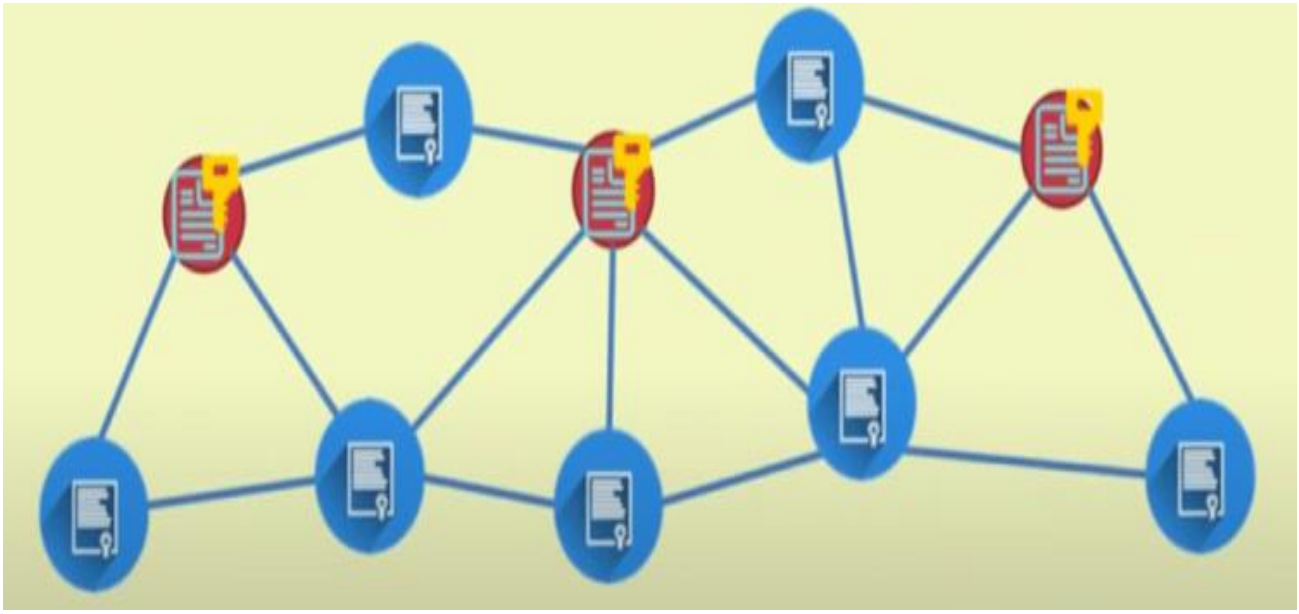
Q.5. Difference between Bitcoin and Ethereum.

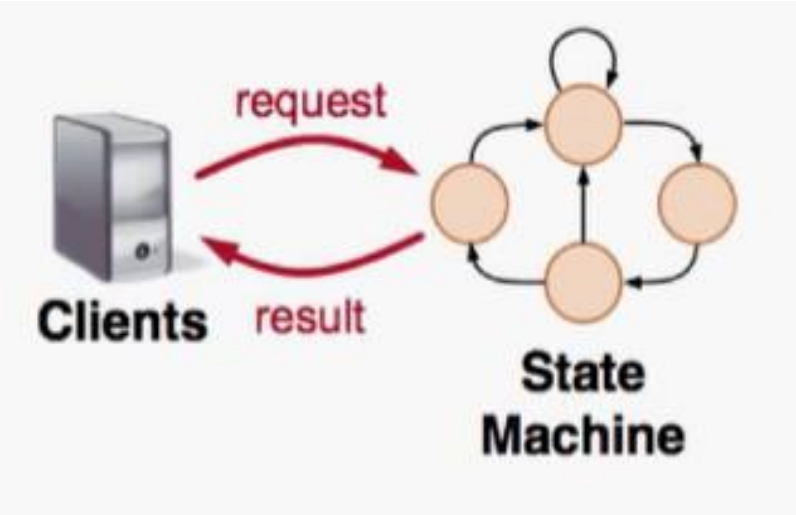
Ans.

Sr.No.	Bitcoin	Sr.No.	Ethereum
1.	Bitcoin was invented by a person or group of people with the name Satoshi Nakamoto in 2008.	1.	Ethereum was proposed by Vitalik Buterin in 2013.
2.	The purpose of bitcoin was to replace national currencies during the financial crisis of 2008.	2.	The purpose of Ethereum was to utilize blockchain technology for maintaining a decentralized payment network and storing computer code.
3.	It does not have smart contracts.	3.	Ethereum allows us to create smart contracts.
4.	Generally, bitcoin transactions are only for keeping notes.	4.	Ethereum transactions may contain some executable code.
5.	Bitcoin runs on the SHA-256 hash algorithm.	5.	Ethereum runs on the Keccak-256 hash algorithm.
6.	The Proof-of-Work (PoW) is the consensus mechanism used by the Bitcoin network.	6.	The Proof-of-Stake is the consensus mechanism used by Ethereum.
7.	The block time of bitcoin is 10 minutes.	7.	The block time of Ethereum is 14 to 15 seconds.
8.	The bitcoin blockchain has a block limit of 1 MB.	8.	The Ethereum blockchain does not have a block limit.

Permissioned Blockchain – State Machine Replication

The state machine replication helps us to achieve a consensus in a permission model. We do not need to execute a smart contract to all the nodes. Rather, the selected subset of contract executor executes it and propagates it with other nodes to ensure the contract's status is propagated to all the nodes uniformly in the network, and they are on the same page. The distributed state machine replication technology ensures consensus in a permission blockchain environment.





Understanding of State Machine Concept

State Machine is characterized by a set of parameters such as set of Inputs, set of Outputs, and the Transition States.

A set of state (S) based on the system design

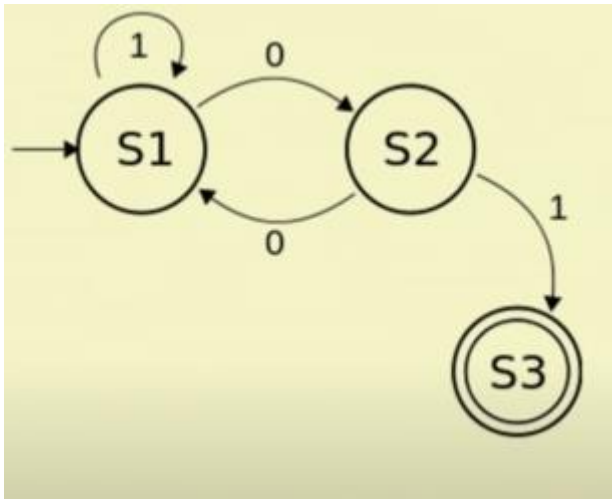
A set of inputs (I)

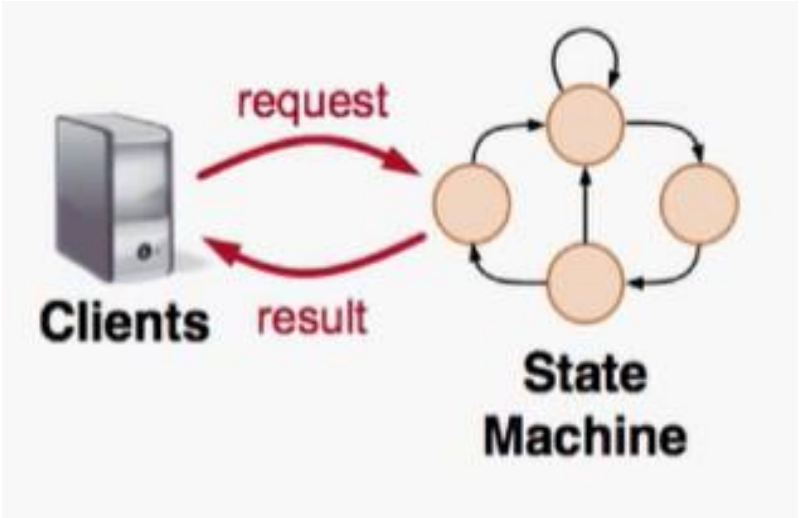
A set of outputs (O)

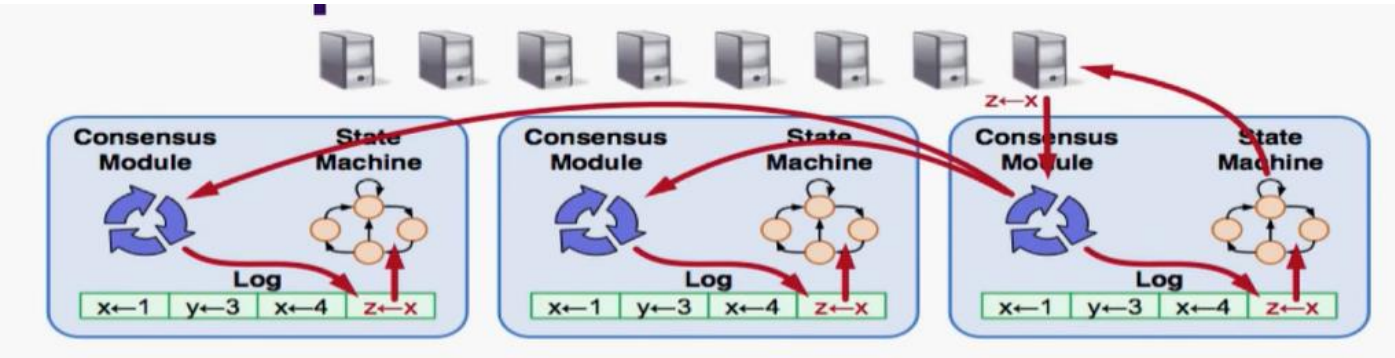
A transition function S x I –> S; takes the current state and input value and produces a set as the output.

A output function S x I –> O

A start state

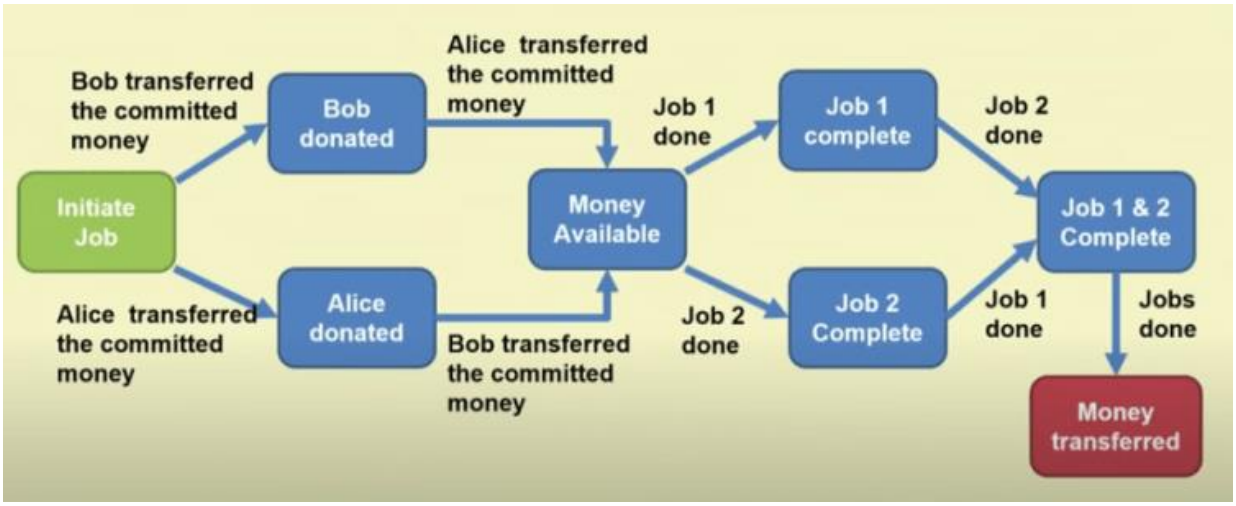






Example of State Machine: Smart Contract – Crowd-Funding

In general, any algorithm can be represented by the finite state machine and we will understand it with an example of Crowdfunding platform in smart contract. The smart contract state machine representation are as follows:



In the crowdfunding platform, there are mainly two parties which include the project proposes and project funders. The project proposers propose the project to the funders, and if they are interested, they will invest money in their project. The funders will release funds after the completion of a certain job. In the above diagram, Alice and Bob are the two funders, and they are transferring money after completing the proposed jobs by the proposers. Once the entire job is completed, money will be transferred to them.

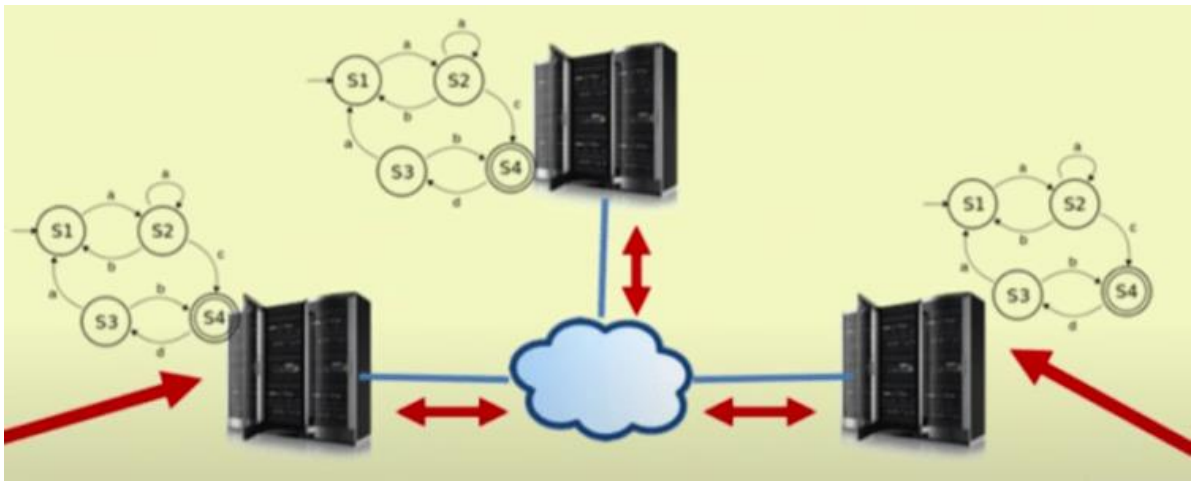
Distributed State Machine Replication Mechanism

In a typical distributed architecture, the distributed state machine replication mechanism work in the following way.

It has multiple distributed servers which work independently in a distributed fashion. Initially, copies of the state machine are placed on each server that resides in the closed network, assume that each server knows the information about other servers.

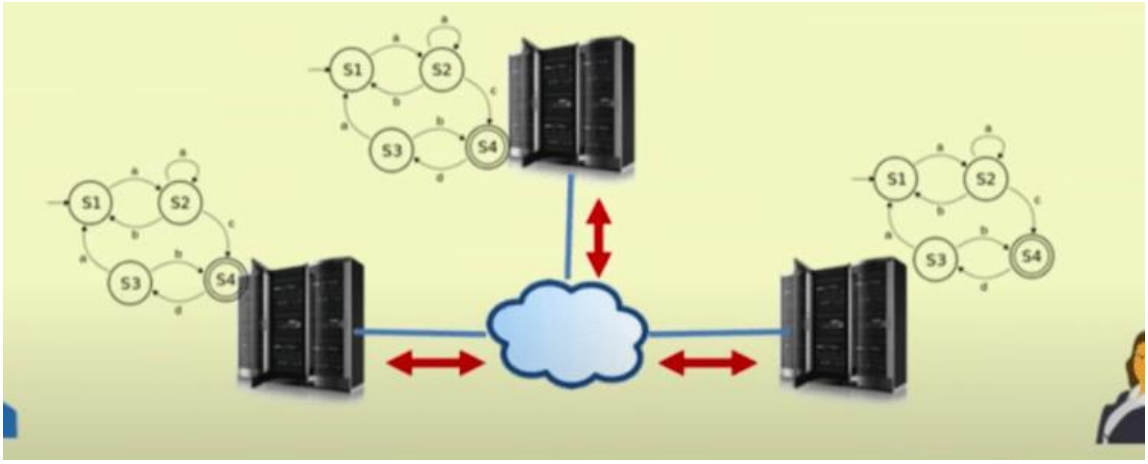
In the distributed setup, it may be possible that different users are communicating to different servers. The servers get the client's request independently. In our example, Alice and Bob are two clients who are transferred their share of money via the distributed network. However, at this moment, this information is locally stored in the connected servers. The end goal is to ensure that all three servers reach the same state after a certain time. i.e., Each server must have information about Alice and Bob have transferred their share of the money.

To achieve this, the state machine replication mechanism works in this principle that, individually, all the servers propagate their state to others. However the problem is to understand ordering or sequences in which requests came from the users.



This can be achieved using the timestamp concept. So, whenever Alice and Bob are making the transactions, the transactions are associated with the timestamp. This way, information will be executed in each server in that particular ordering algorithm. Consequently, all will execute in the same order in which it was support to be executed. This way, all servers are reaching the same state.

Once execution is finished, then the states should be propagated in the entire network. It might be possible that one of the servers was offline due to some failure. This way, all the servers will have a state copy. So that update is done through synchronization of the state machine.



At the end, there will be certain outputs and those outputs will be updated to clients.

In the entire procedure, there are two glitches. The first one is that we need to maintain ordering in service, and the second is that in the presence of failure, the system needs to ensure that all the individual servers are on the same page. From our example, in the end, all the three servers must know that Alice and Bob had shared their money, and money has been transferred to the recipient.

Why do we apply the state machine replication-based consensus algorithm in a permissioned model compared to the challenge-response-based consensus algorithm in the permissionless model?

There is a natural reason to use state machine replication-based consensus over permissioned blockchains are there as follows:

The network is closed, the nodes know each other, so state replication is possible among the known nodes.

Avoid mining overhead, do not need to spend anything like power, time, bitcoin other than message passing.

However, the consensus is still required as machines can be faulty or behave maliciously.

Application of State Machine Replication

One typical application of the state machine replication is in the flight control system. This technique is applied when there are multiple flights that want to coordinate their positions among themselves.

A state machine replication-based algorithm for consensus is also applied for fund transferring systems in a distributed environment. In an open environment, we have challenge -response based consensus algorithm to achieve a consensus.

For other distributed applications like a distributed ledger election, where all the nodes collectively need to elect one leader in the system, consensus is ensured when the entire node selects the same leader, or the same leader is elected at the end of the route.

Distributed consensus algorithm is used for this kind of agreement protocols where the nodes collectively need to come to a specific agreement.

Introduction

Whenever it comes to consensus discussion, the **Byzantine general's problem** is one of the most complex and controversial. In 2008, with the inception of Bitcoin, Satoshi Nakamoto claimed to solve the byzantine problem with the **proof of work (PoW)** consensus mechanism. However, it was just the first step in achieving consensus in a **decentralized** network. This article will explain the Byzantine problem and **Byzantine Fault Tolerance (BFT) Consensus Mechanism** in Blockchain. It further goes through a **practical byzantine fault tolerance (pBFT) consensus approach** to tackle the problem.

Let's begin with the Byzantine general's problem.

What is the Byzantine Generals' Problem?

Byzantine Generals' problem was acknowledged in 1982 as a **logical decision puzzle.** Its basis on **how generals of the same side with different troops might have a communication problem in making decisions about the next move against the enemy**.

Let's see the below diagram to understand the problem thoroughly.

The problem states like a group of generals with their army are about to attack their enemy. They surrounded the enemy's castle from 4 different directions. Now how would they communicate the decision of attacking or retreating at the same time?

Here, **a synchronized and concurrent attack on the enemy will be a success**.

Following are problems that may arise while sharing the decision from one general to another:

- The messenger might get captured while delivering the decision.
- What if an imposter altered the message received.
- How can a general make sure if he received the message from the expected general?
- What if other generals become traitors and they send the message to attack, but they actually retreat.

How can the system be sure that each general will attack at the same time from their designated location? Is there no way but to trust each other completely?

Blockchain seems to resolve this problem with the Byzantine fault tolerance (BFT) consensus mechanism.

What is Byzantine Fault Tolerance (BFT)?

To ensure the success of the generals' team, they need an algorithm that could adhere to the following conditions:

- All the troop generals need to agree on the next action of the plan.
- The generals should be trustworthy and loyal to the system.
- Generals must not get influenced to become network traitors.
- They need to follow the algorithm of the system.
- The group of generals needs to reach a consensus or decision, irrespective of the traitors' actions.
- The system or network should not lead to a 51% attack at any point of action.

Byzantine Fault Tolerance (BFT) is a consensus approach that resists a system to get into the Byzantine Generals' problem. It also means the system should stay intact even if one of the nodes (or general) fails. *In addition, BFT aims to reduce the effect of malicious byzantine nodes (or general) on the network.*

What is Practical Byzantine Fault Tolerance (PBFT)?

In an attempt to overcome the Byzantine problems, Barbara Liskov and Miguel Castro introduced a Practical Byzantine Fault Tolerance (pBFT) consensus algorithm in 1999. **They aim to ensure a practical byzantine state machine replication for tolerating malicious or byzantine nodes.**

The pBFT follows an asynchronous approach. The following are essential aspects of the pBFT consensus algorithm:

- All nodes are assembled in a sequence.
- One network node serves as a **leader node**, and the rest of them are **backup nodes**.
- The primary or **leader node** serves the client's request. It works as a moderator between client and backup nodes.
- All nodes are capable of communicating with other nodes to check the honest nodes.
- Honest nodes should be able to reach a consensus for the next global change in the network based on majority rule.
- It identifies the source of the message to make sure it's sent by the correct sender.
- Ensures the message has not been modified or corrupted in between.

As we went through the principles of pBFT. Let's see how it works?

How does the PBFT Algorithm work?

The PBT highly depends on the condition that the maximum number of malicious or byzantine nodes must exceed one-third of all the nodes in the network. Hence, the security of the network is directly dependent upon the number of total honest nodes.

In short, a pBFT system can handle 'f' faulty or byzantine nodes where there are 3f + 1 total number of nodes on the network.

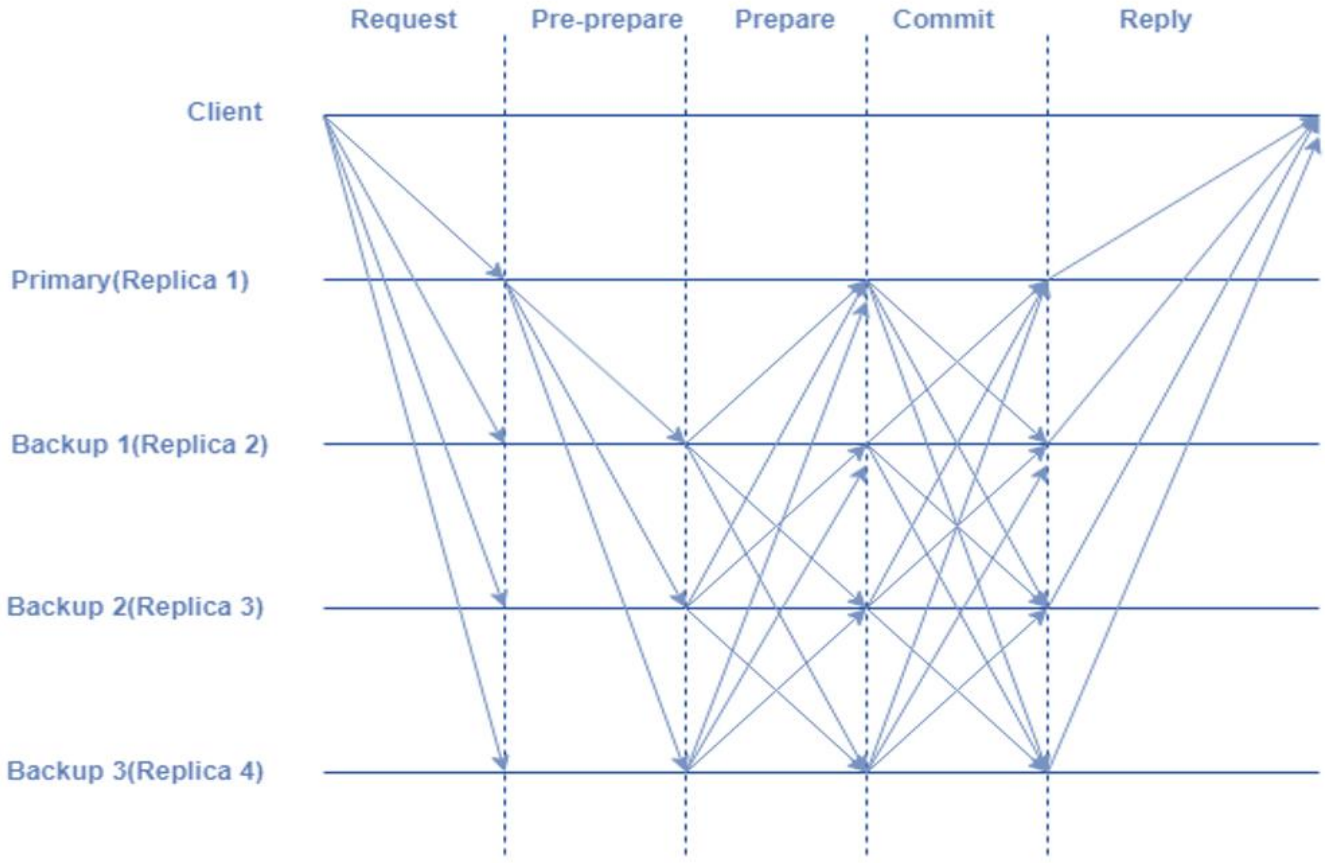
Following is the process of pBFT consensus algorithm:

- A **client** sends a request to the leader node.
- Then the **leader node** floods the request to all **backup nodes**.
- All the nodes work on the request and send a reply to the client.
- The client waits for **(f + 1)** replies from all the nodes with the same result. Here, **f = number of possible faulty nodes.**

The pBFT mechanism consists of 3 phases:

- **Pre-prepare phase:** The leader node sends out a pre-prepared message to each backup node.
- **Prepare:** After receiving the pre-prepared message from the leader, the backup nodes send the prepared message as a reply to all other nodes including the leader. A node is considered prepared only if it has received pre-prepared by the leader and seen (2f + 1) number of prepared messages from other nodes.
- **Commit:** If the nodes are prepared, they send a commit message. If a node receives (f + 1) commit messages, they carry out the client's request.

The below diagram shows the working of the pBFT algorithm.



The whole process of verification in a distributed system uses the concept of **digital signatures**. The validity message and sender are ensured using sequence numbers and metadata.

Blockchain like Zilliqa, Hyperledger fabric, and Tendermint uses the Practical Byzantine Fault Tolerance (pBFT) algorithm.

Benefits of PBFT

Following are the advantages of the pBFT consensus algorithm:

- A pBFT doesn't require carrying out high mathematical computations like PoW.
- It is an energy-efficient consensus model.
- A block of transactions here does not need to follow multiple confirmations by each node. Hence, it requires less time.

- As pBFT requires every node to participate and serves the client request, each node gets the reward. Hence low reward variance between each node.

Limitations of PBFT

Following are the disadvantages of the pBFT consensus algorithm:

- pBFT has a high communication overhead that will increase with the number of nodes in the network.
- It has scalability issues with more extensive networks.
- pBFT is susceptible to **Sybil attacks** in which one node controls or acts as multiple network nodes.

Permissioned Blockchain – Raft Consensus

The idea behind the Raft consensus algorithm is that the nodes (i.e., server computers) collectively select a leader, and the remaining nodes become the followers. The leader is responsible for state transition log replication across the followers under the closed distributed environment, assuming that all the nodes are trustworthy and have no malicious intent.

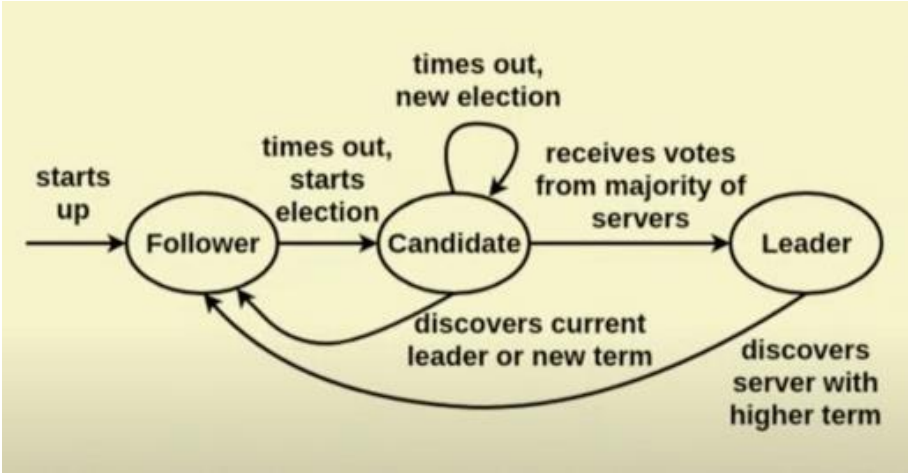
The basic idea of Raft came from the fact that in a distributed environment, we can come to a consensus based on the Paxos algorithm and elect a leader. Interestingly, if we have a leader in the system, we can avoid multiple proposers proposing something altogether.

In the case of Paxos, we don't have any straightforward mechanism to elect a leader. However, to elect a leader, multiple proposers propose the thing simultaneously. Consequently, the protocol becomes complex, and the acceptors have to accept one of the proposals from the proposer. In that case, we use the highest proposal number for the tie-breaking mechanism and embed a certain algorithm in Paxos to ensure that every proposal coming from a different proposer is unique. Thus, all these internal details make the Paxos more complicated.

In a distributed environment and under a synchronous assumption (closed environment), it is possible to design a consensus algorithm. First, we will elect a leader and then the tasks of the leader to propose something. There will be a single proposer, and all the acceptors are followers of the leader. They may either accept or reject the leader's opinion.

Raft Overview

The system starts up and has a set of follower nodes. The follower nodes look for a leader. If a timeout happens, there is no leader, and there is a need to elect a leader in the system. A few candidates stand for a leader in the election process, and the remaining nodes vote for the candidate. The candidates who receive the majority votes become the leader. The leader proposes a proposal, and the followers can either vote for or against that proposal.



Raft consensus algo

An example from the database replication: We have distributed multiple replicated servers, and we want to build a consensus among these multiple replicated servers. Whenever some transactions are coming up from the clients, we want these replicated servers to decide whether to commit those transactions collectively.

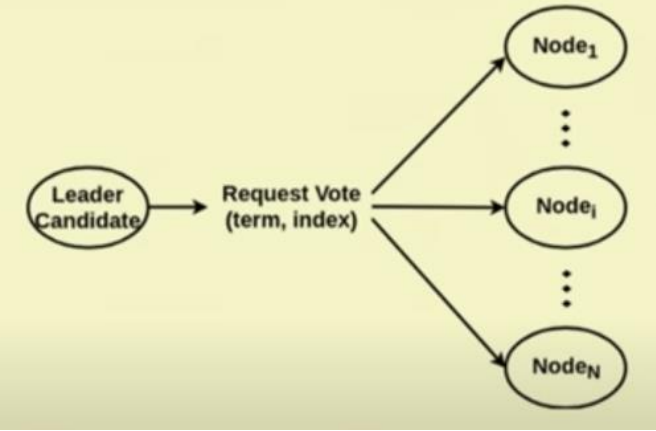
Raft Consensus Algorithm

Electing the Leader: Voting Request

The first part of the Raft is to elect a leader, and for that, there should be some leader candidates. The nodes sense the network, and if there is no leader candidate, then one of the nodes will announce that I want to be a leader. The leader candidate requests the votes. This voting request contains 2 parameters:

Team: The last calculated number known to candidate + 1.

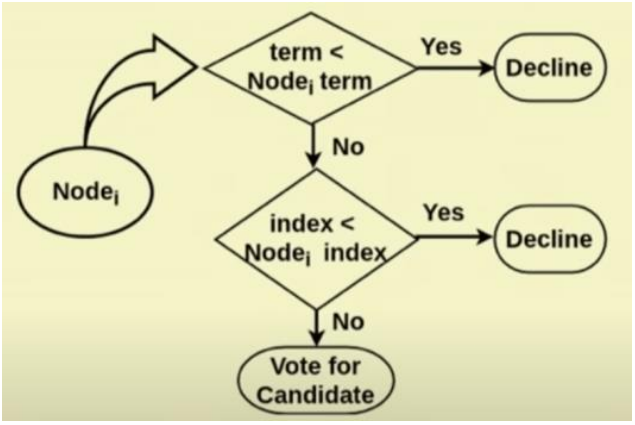
Index: Committed transactions available to the candidate.



These algorithms work in multiple rounds, and the term indicates a new voting round. If the last voting finishes, then the next term will be old term number + 1; The index indicates committed transactions available to the candidate. It is just like an increasing number to distinguish between already committed and new transactions.

Electing the Leader: Follower Node's Decision Making

Once the nodes receive a voting request, their task is to vote pro or against the candidate. So, this is the mechanism to elect a leader in the Raft consensus algorithm. Each node compares the received term and index with the corresponding current known values.

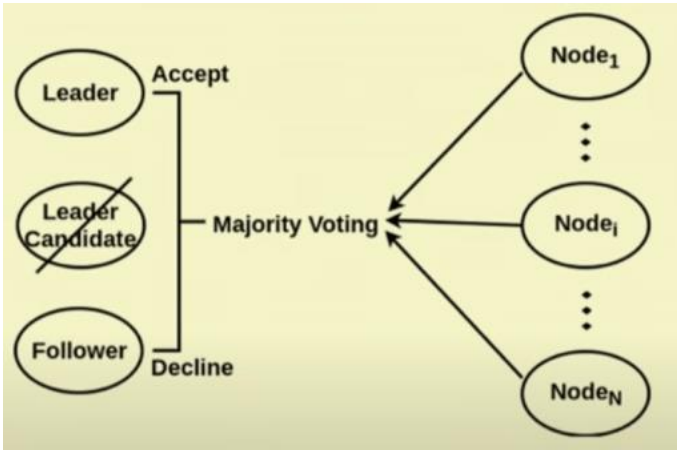


The node(i) receives the voting request. It compares the already seen team with the newly received team. If a newly received team is less than the already seen team, then it discards because the node considers this request as an old request.

The newly received team is greater than the already seen team. It checks for the newly received index number with the already seen index number. If the newly received index number is greater than already seen, it votes for the candidate; else, it declines.

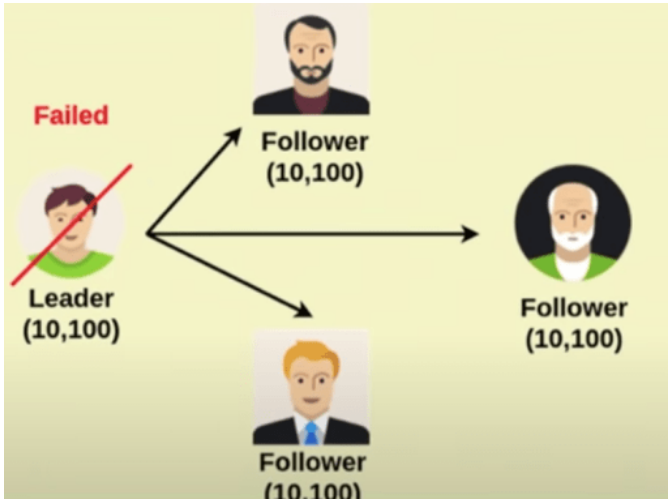
Electing the Leader: Majority Voting

Every node sends their vote and candidates who get majority vote becomes a leader, and commit the corresponding log entry. in other words, If a certain leader candidate, receives majority of the vote from the nodes, then that particular candidate becomes a leader and other becomes the follower of that node.

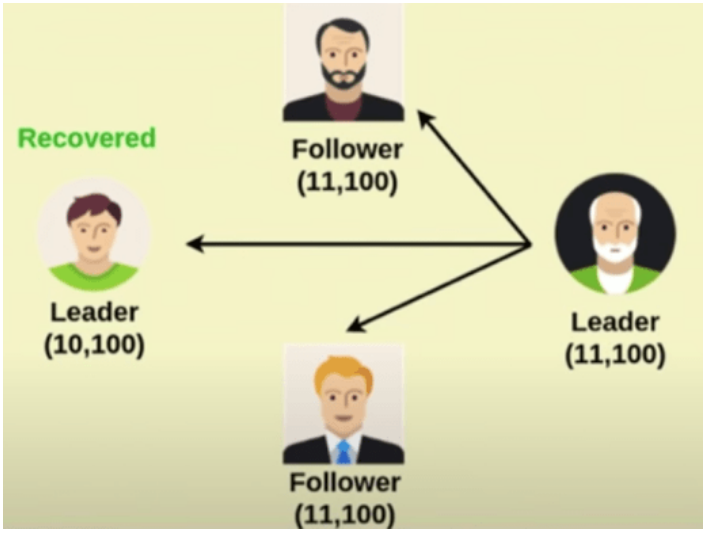


Multiple Leader Candidates: Current Leader Failure

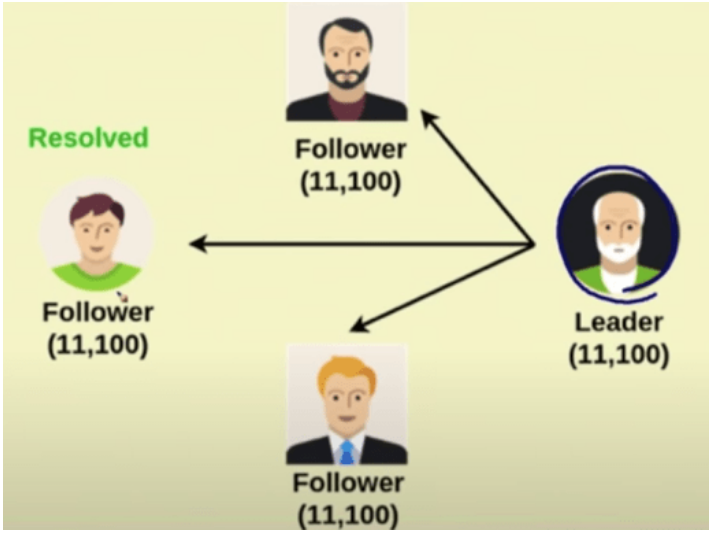
Let us understand a scenario where there is a leader, and three followers and the current team is 10, and the commit index value is 100. Suppose the leader node has failed or followers didn't receive a heartbeat message within the heartbeat timeout period.



After the timeout, one of the nodes will become a leader candidate, initiates a leader election, and becomes a new leader with team 11 and commit index value 100. The new leader periodically sends the heartbeat message to everyone to indicate his presence.

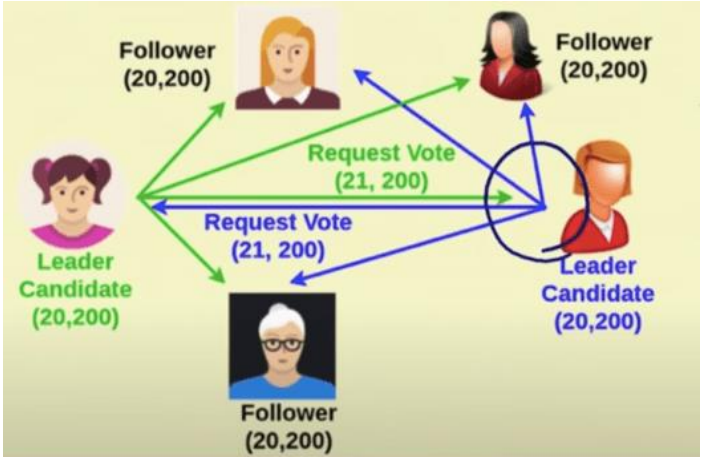


In the meantime, the old leader gets recovered, and he also receives a heartbeat message from the new leader. The old leader understands that a new term has started. Then the old leader will change his status from leader to follower. So this is the one way to handling a new leader by utilizing the team parameter.

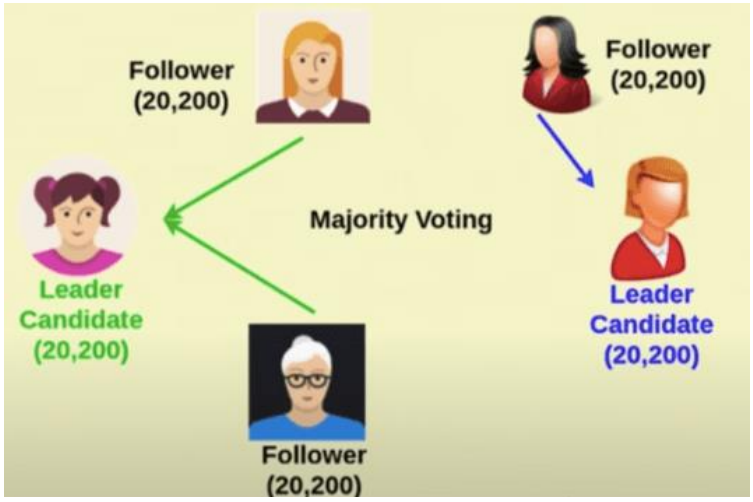


Multiple Leader Candidates: Simultaneous Request Vote

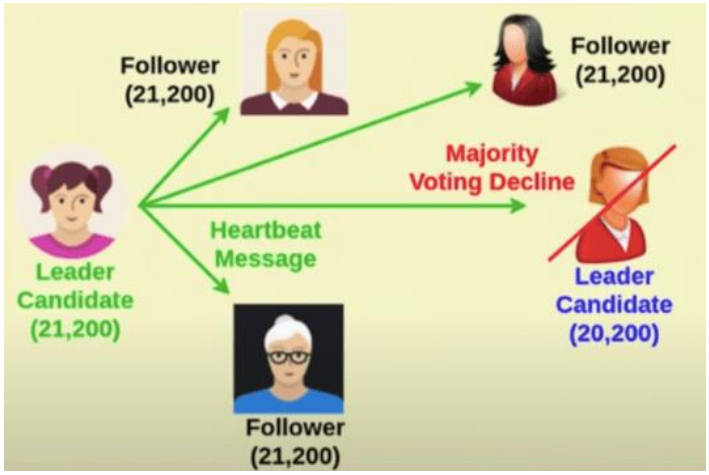
Let us understand a scenario where there is a leader, and three followers and the current team is 20, and the commit index value is 200. Suppose the leader node has failed or followers didn't receive a heartbeat message within the heartbeat timeout period. It may be possible that multiple followers sense the timeout period simultaneously and become a leader candidate, and initiates the leader election procedure independently. Two nodes send the request messages with team 21 at the same time in the network.



There are two leader candidates, and both are sending voting request messages, at the same time, for round (term) 21. Then, they look for the majority voting. In this example, the first candidate receives two votes, and the second candidate receives one vote, so based on the majority voting, the first candidate is a winner.

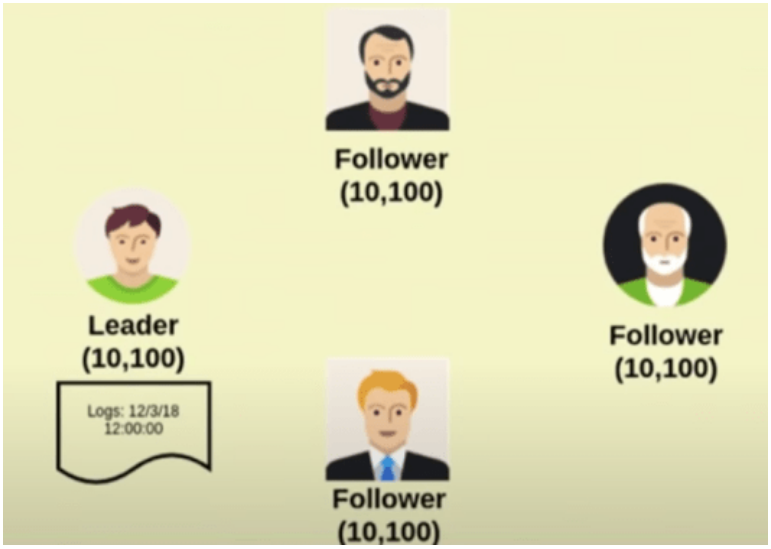


The node which gets the majority votes send a heartbeat message to everyone. Another leader candidate also received the heartbeat message from the winner, and this leader candidate falls back to a follower from the leader candidate.

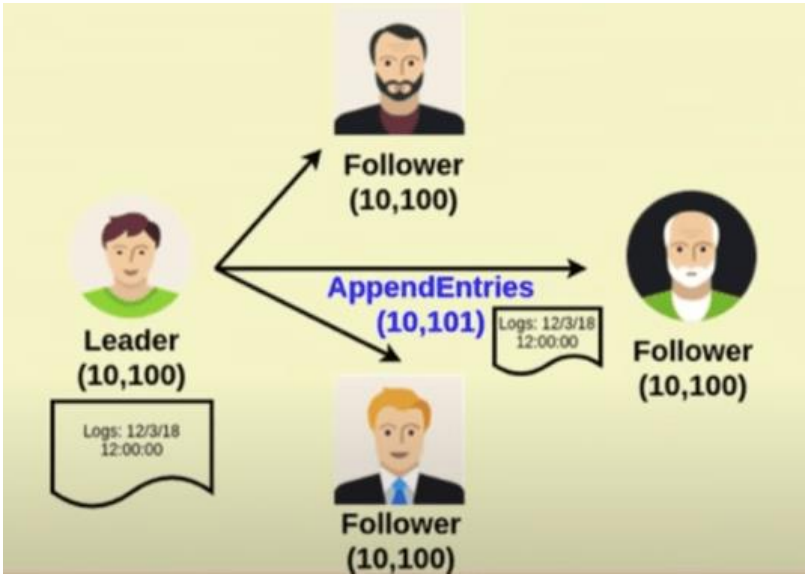


Committing Entry Log

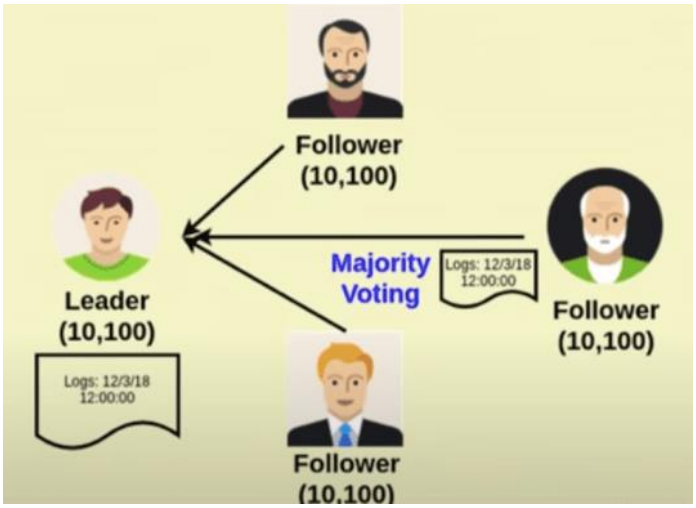
In the above sections, we have seen the procedure to elect a leader and other special cases. Now we will understand how the transactions are managed in a closed distributed environment. Let us consider that the current term value is 10, and the index value is 100, which means most of the nodes have seen and committed transaction index value number 100.



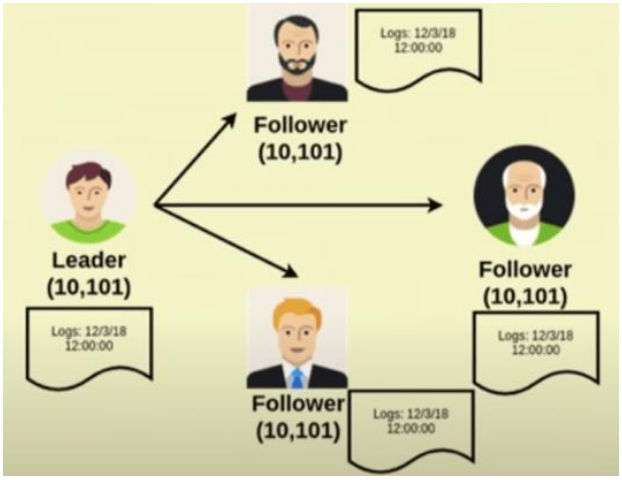
The leader proposes a new transaction, adds an entry log with term 10 and the new transaction index value as 101. Further, the leader sends a message called append entries to all the followers, and they collectively vote either for or against this transaction.



The leader receives the vote for this transaction index value 101. The followers' node votes for or against this transaction. If the majority says that they are fine with committing this particular log. Then, the leader considers that the transaction log is approved by the followers.

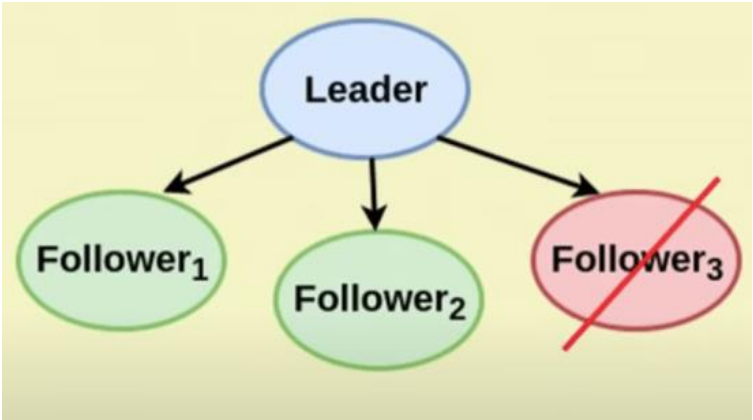


After successful acceptance of the entry log, the leader sends an accept message based on the majority voting to all the individual followers to update the committed index to 101.



Handling Failure

Multiple kinds of failures exist in the environment. However, Paxos and Raft consensus algorithms only support Crash or Network fault. The followers may have crashed, but the system can tolerate up to N/2 -1, where N is the total number of nodes in the environment, as it does not affect the system due to the majority voting. This indicates that the majority of the followers are non-faulty, and they can send a vote. The leader can take the majority decision whether to accept or reject a particular transaction.



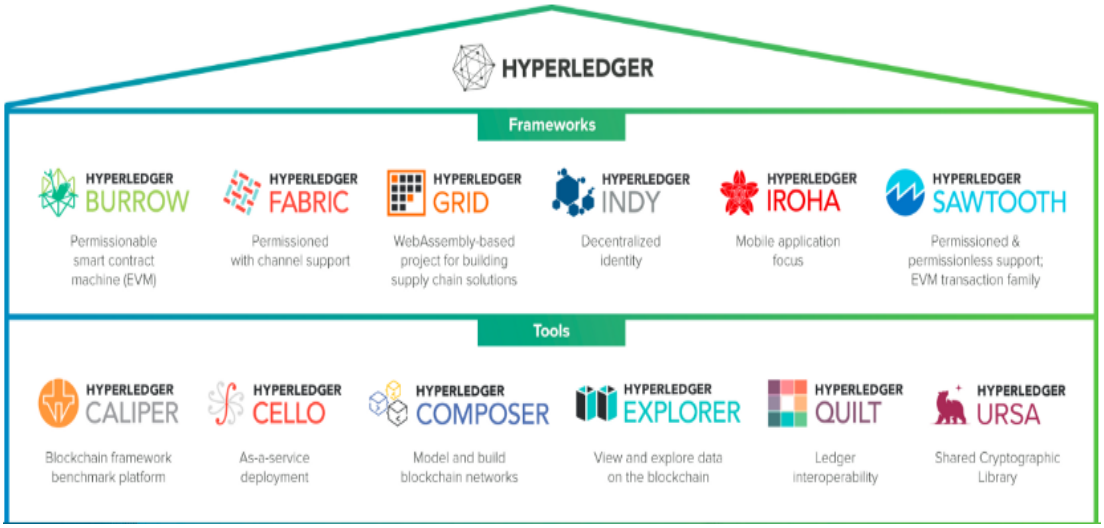
Hyperledger

- Blockchains are IT projects with a twist of business built in and they do not need to be difficult. Blockchains are revolutionary in some regards due to how disruptive they can be in some industries. Because they have been disruptive does not mean they need to be disruptive in other ways such as when migrating

- The Linux Foundation hosts Hyperledger and provides a governance structure and oversight to the Hyperledger community. It is a global open-source project and the result of collaboration from technology leaders.

- According to the Hyperledger official website, *"Hyperledger is an open-source collaborative effort created to advance cross-industry blockchain technologies.*" Although it is hosted by the Linux Foundation, it is a global collaboration among industry leaders in finance, banking, IoT, technology, manufacturing, and supply chains.

- The Linux Foundation also embraces a modular umbrella approach to enterprise blockchains. Hyperledger is an open-source software licensing model, which allows the user to model code and distribute it in an appropriate manner.



- Hyperledger has six frameworks and six tools and utilities

- The umbrella strategy, also referred to as the *greenhouse strategy*, is a proven model that the Linux Foundation has used repeatedly in the other projects it maintains. Historically, the Linux Foundation provides excellent management and insight into how to manage an open source project for consortium members.

Hyperledger Framework

Framework	Application
Indy	Decentralized Identity
Iroha	Mobile Application Focused
Sawtooth	Permissioned & permissionless support; EVM transaction family
Burrow	Permissionable smart contract machine (EVM)
Fabric	Permissioned with channel support
Grid	WebAssembly-based project for building supply chain solutions

Hyperledger Burrow:

It is a framework for executing smart contracts in permissioned blockchains. The goal of Hyperledger burrow is to facilitate cross-industry applications for smart contracts. It is built around the BFT consensus algorithm.

Hyperledger Fabric:

This is a kind of framework implementation which works for the development of blockchain applications. It also stretches the support to modular architecture and produces various other solutions. The most distinguished feature of this particular framework is that it allows some other components like consensus and membership services. Another distinct feature is its induction of container technology in its core mechanism which helps in hosting smart contracts. It is mainly called chaincode which is accountable for the applicability of the entire system.

Some of its salient features are:

>Confidential information can be shared through private channels.
>It provides membership services and ordering service (pluggable consensus) as well. >Storage of world state is enabled through CouchDB

Hyperledger Sawtooth:

It is an open-source project and used as an enterprise-level blockchain system used for creating and operating distributed ledger applications. Hyperledger sawtooth supports a variety of consensus algorithms like PBFT, and PoET.

Hyperledger Indy:

It is a project that is made for decentralized identity. It offers lots of libraries, tools, and reusable components for creating decentralized identities.

Hyperledger Iroha:

It is a blockchain platform designed for infrastructure projects that need distributed ledger technology. It is used to build identity management platforms like national IDs. It can integrate with Linux, macOS, and Windows platforms.

Properties	Iroha	Sawtooth	Fabric	Indy	Burrow
Modularity	Less	High	High	Average	Less
Flexibility	High	Average	Average	Average	Average
Scalability	Less	High	Less	Average	Average
Membership service	No	No	Yes	No	No
Decentralized identity	No	No	No	Yes	No

Hyperledger Tools

1. Hyperledger Caliper

Caliper is a Blockchain tool hosted by the Linux Foundation. It lets you compute the performance of specific Blockchain implementations by leveraging a set of predefined use cases. Caliper can also generate reports on different performance factors, including resource utilization, transaction latency, and transactions per second (TPS).

2. Hyperledger Cello

Cello is a Blockchain module toolkit. It is essentially an on-demand "as-a-service" deployment model developed for the Blockchain ecosystem. Cello provides a multi-tenant chain service that can work on top of multiple infrastructures, including container platforms and virtual machines. It reduces the efforts required to build, maintain, and terminate blockchains.

3. Hyperledger Explorer

Hyperledger Explorer is a Blockchain module explicitly designed for developing user-driven web applications. It can be used for viewing, deploying, invoking/querying blocks, network information, transaction data, chaincodes, and other relevant data that is stored in a Blockchain ledger.

4. Hyperledger Composer

Composer is both a development framework and toolkit designed to make the development of Blockchain applications and smart contracts more seamless and convenient. You can use Composer to develop and deploy Blockchain applications rapidly. It leverages tools like Node.js, CLI, NPM, etc., to provide business-focused abstractions, sample apps, and easy-to-test DevOps processes.

5. Hyperledger Quilt

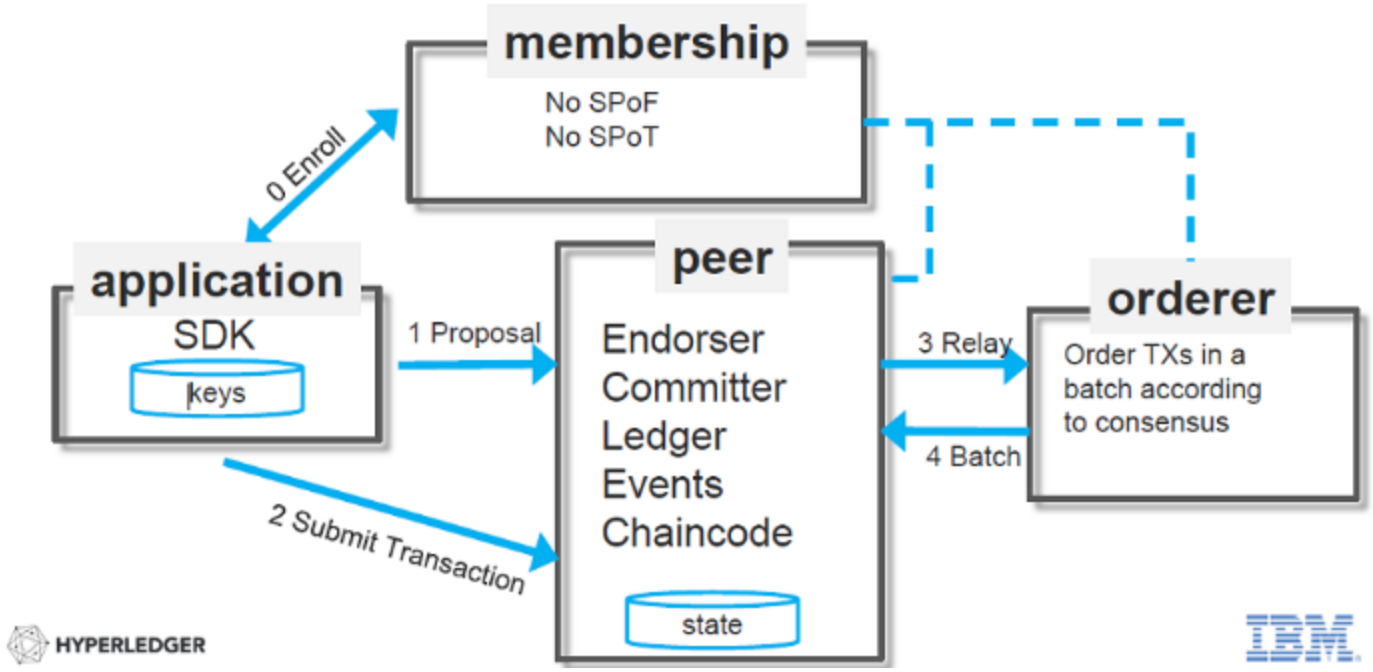
Quilt is a one of the business Blockchain tools that aims to facilitate interoperability between ledger systems by implementing the Interledger protocol (ILP), which is a payments protocol used for moving value across both distributed and non-distributed ledgers. Thanks to ILP, Quilt can also enable atomic swaps between a single account namespace for accounts and ledgers.

Hyperledger Fabric:

Hyperledger Fabric

Hyperledger Fabric is by far the most widely used of the frameworks.

The diagram below represents **application communication viewpoint** at key **building blocks** of **Hyperledger Fabric 1.0 architecture:**



Following are some of the key building blocks of Hyperledger Fabric 1.0 Architecture: **Membership Services Provider**: Enrolls the clients

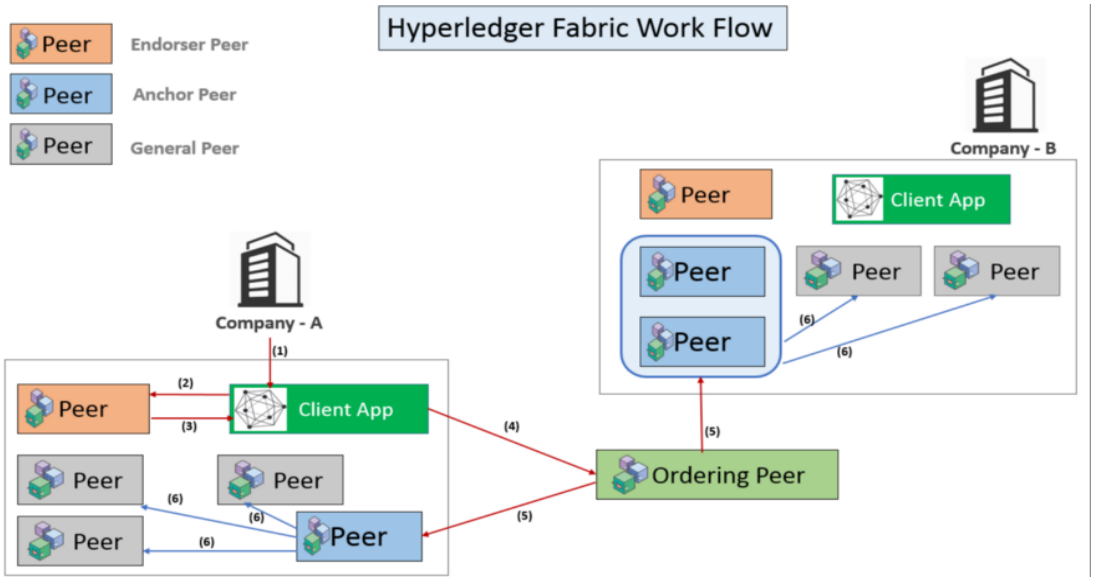
Peers: Peer nodes can be endorser (endorse proposal for transaction) and committer nodes (write block of transactions to ledger)

Chaincode: Smart contract written in Java/Go which is invoked by a transaction. Peer nodes having chaincode becomes the endorser for that chaincode. ESCC (Endorser system chain code) executes the chaincode using proposal and read-write set information.

Ledger: Ledger which holds the copy of transactions in form of blocks

Ordering service: Consenter service which validates the transaction using VSCC (Validation system chaincode), orders the transaction in a block and sends it to peer nodes (endorsers & committers)

Architecture and Working of Hyperledger Fabric:



Step 1:

First a participant(client) in the member organization initiates a transaction request.

Step 2:

The transaction invocation request is broadcast to the Endorser peer by the client application.

Step 3:

The transaction is validated by the endorser peer reviewing the certificate details and others. Then it runs the chaincode (i.e. Smart Contract) and gives the Client the Endorsement responses. As part of the endorsement response, the endorse peer sends a message approving or rejecting the transaction.

Step 4:

The transaction in now by the client to the Ordered peer to be properly ordered and added to a block.

Step 5:

The transaction is added to a block by the Ordered node, which then sends the block on to the anchor nodes of various organization that are members of the Hyperledger Fabric network.

Step 6:

The block was then broadcast by the anchor nodes to the peers within their own organization. The most recent block is then added to the local ledger by each of these peers. Thus, the ledger is updated across the entire block is then added to the local ledger by each of these peers. Thus, the ledger is updated across the entire network.