MCALE331 Sub: Blockchain MCA SEM-III (CBCGS) (2 Year MCA)

Module: Introduction

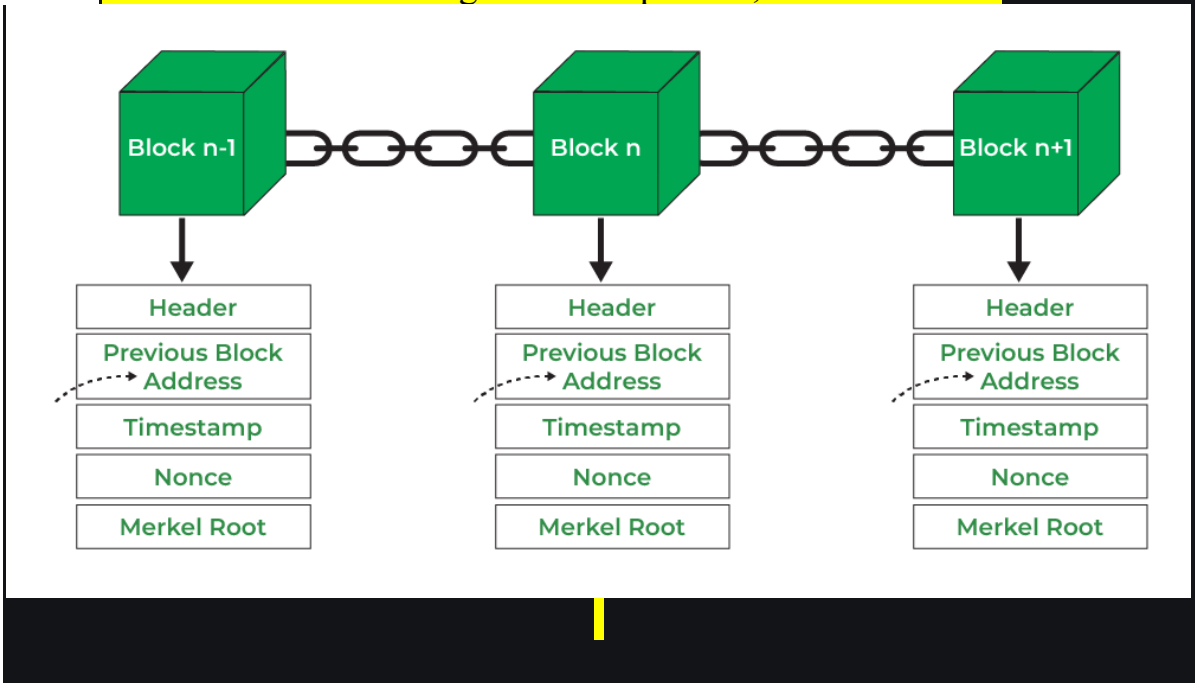
1. **Explain Blockchain and its structure.**

Blockchain is a Distributed Ledger Technology. It is a distributed and decentralized database and it is secured ever as compared to other technologies.

Blockchain is a technology where multiple parties involved in communication can perform different transactions without third-party intervention. Verification and validation of these transactions are carried out by special kinds of nodes.

Benefits of Blockchain:

- It is safer than any other technology.
- To avoid possible legal issues, a trusted third party has to supervise the transactions and validate the transactions.
- There's no one central point of attack.
- Data cannot be changed or manipulated, it's immutable.



1. **Header:** It is used to identify the particular block in the entire blockchain. It handles all blocks in the blockchain. A block header is hashed periodically by miners by changing the nonce value as

part of normal mining activity, also Three sets of block metadata are contained in the block header.

2. **Previous Block Address/ Hash:** It is used to connect the $i+1^{th}$ block to the i^{th} block using the hash. In short, it is a reference to the hash of the previous (parent) block in the chain.
3. **Timestamp:** It is a system verify the data into the block and assigns a time or date of creation for digital documents. The timestamp is a string of characters that uniquely identifies the document or event and indicates when it was created.
4. **Nonce:** A nonce number which uses only once. It is a central part of the proof of work in the block. It is compared to the live target if it is smaller or equal to the current target. People who mine, test, and eliminate many Nonce per second until they find that Valuable Nonce is valid.
5. **Merkel Root:** It is a type of data structure frame of different blocks of data. A Merkle Tree stores all the transactions in a block by producing a digital fingerprint of the entire transaction. It allows the users to verify whether a transaction can be included in a block or not.

Key Characteristics of Blockchain Architecture

- **Decentralization:** In centralized transaction systems, each transaction needs to be validated in the central trusted agency (e.g., the central bank), naturally resulting in cost and the performance jam at the central servers. In contrast to the centralized mode, a third party is not needed in the blockchain. Consensus algorithms in blockchain are used to maintain data stability in a decentralized network.
- **Persistency:** Transactions can be validated quickly and invalid transactions would not be admitted by persons or miners who mining the crypto. It is not possible to delete or roll back transactions once they are included in the blockchain network. Invalid transactions do not carry forward further.
- **Anonymity:** Each user can interact with the blockchain with a generated address, which does not disclose the real identity of the miner. Note that blockchain cannot guarantee perfect privacy preservation due to the permanent thing.
- **Auditability:** Blockchain stores data of users based on the Unspent Transaction Output (UTXO) model.

Every transaction has to refer to some previous unspent transactions. Once the current transaction is recorded into the blockchain, the position of those referred unspent transactions switches from unspent to spent. Due to this process, the transactions can be easily tracked and not harmed between transactions.

- **Transparency:** The transparency of blockchain is like cryptocurrency, in bitcoin for tracking every transaction is done by the address. And for security, it hides the person's identity between and after the transaction. All the transactions are made by the owner of the block associated with the address, this process is transparent and there is no loss for anyone who is involved in this transaction.
- **Cryptography:** The blockchain concept is fully based on security and for that, all the blocks on the blockchain network want to be secure. And for security, it implements cryptography and secures the data using the cipher text and ciphers.

2. **What are the uses of Blockchain?**

1. Asset Management

Blockchain plays a big part in the financial world and it is no different in asset management. In general terms, asset management involves the handling and exchange of different assets that an individual may own such as fixed income, real estate, equity, mutual funds, commodities, and other alternative investments. Normal trading processes in asset management can be very expensive, especially if the trading involves multiple countries and cross border payments. In such situations, Blockchain can be a big help as it removes the needs for any intermediaries such as the broker, custodians, brokers, settlement managers, etc. Instead, the blockchain ledge provides a simple and transparent process that removes the chances of error.

2. Cross-Border Payments

Have you ever tried to make cross-border payments in different currencies from one country to another? This can be a long complicated process and it can take many days for the money to arrive at its destination. Blockchain has helped in simplifying these cross border payments by providing end-to-end remittance services without any intermediaries. There are many remittance companies that offer Blockchain services which can be used to make international remittances within 24 hours.

3. Healthcare

Blockchain can have a big impact on healthcare using smart contracts. These smart contacts mean that a contract is made between 2 parties without needing any intermediary. All the parties involved in the contract know the contract details and the contract is implemented automatically when the contract conditions are met. This can be very useful in healthcare wearing personal health records can be encoded via Blockchain so they are only accessible to primary healthcare providers with a key. They also help in upholding the HIPAA Privacy Rule which ensures that patient information is confidential and not accessible to everyone.

4. Cryptocurrency

Perhaps one of the most popular applications of Blockchain is in Cryptocurrency. Who hasn't heard about bitcoin and it's insane popularity. One of the many advantages of cryptocurrency using blockchain as it has no geographical limitations. So crypto coins can be used for transactions all over the world. The only important thing to keep in mind is exchange rates and that people may lose some money in this process. However, this option is much better than regional payment apps such as Paytm in India that are only relevant in a particular country or geographical region and cannot be used to pay money to people in other countries.

5. Birth and Death Certificates

There are many people in the world who don't have a legitimate birth certificate especially in the poorer countries of the world. According to UNICEF, one-third of all the children under the age of five don't have a birth certificate. And the problem is similar to death certificates as well. However, Blockchain can help in solving this problem by creating a secure repository of birth and death certificates that are verified and can only be accessed by the authorized people.

6. Online Identity Verification

It is not possible to complete any financial transactions online without online verification and identification. And this is true for all the possible service providers any user might have in the financial and banking industry. However, blockchain can centralize the online identity verification process so that users only need to verify their identity once using blockchain and then they can share this identity with whichever service provider they want. Users also have the option to choose their identity verification methods such as user authentication, facial recognition, etc.

7. Internet of Things

Internet of things is a network of interconnected devices that can interact with others and collect data that can be used for gaining useful insights. Any system of "things" becomes IoT once it is connected. The most common example of IoT is perhaps the Smart Home where all the home appliances such as lights, thermostat, air conditioner, smoke alarm, etc. can be connected together on a single platform. But where does Blockchain come into this? Well, Blockchain

is needed for providing security for this massively distributed system. In IoT, the security of the system is only as good as the least secured device which is the weak link. Here Blockchain can ensure that the data obtained by the IoT devices are secure and only visible to trusted parties.

8. Copyright and Royalties

Copyright and royalties are a big issue in creative sectors like music, films, etc. These are artistic mediums and it doesn't sound like they have any link with Blockchain. But this technology is quite important in ensuring security and transparency in the creative industries. There are many instances where music, films, art, etc. is plagiarized and due credit is not given to the original artists. This can be rectified using Blockchain which has a detailed ledger of artist rights. Blockchain is also transparent and can provide a secure record of artist royalties and deals with big production companies. The payment of royalties can also be managed using digital currencies like Bitcoin.

3. **Explain the distributed consensus.**

A consensus mechanism is a fault-tolerant mechanism that is used in computer and blockchain systems to achieve the necessary agreement on a single data value or a single state of the network among distributed processes or multi-agent systems, such as with cryptocurrencies. It is useful in record-keeping, among other things.

There are different kinds of consensus mechanism algorithms, each of which works on different principles.

The proof of work (PoW) is a common consensus algorithm used by the most popular cryptocurrency networks like bitcoin and litecoin. It requires a participant node to prove that the work done and submitted by them qualifies them to receive the right to add new transactions to the blockchain. However, this whole mining mechanism of bitcoin needs high energy consumption and a longer processing time.

The proof of stake (PoS) is another common consensus algorithm that evolved as a low-cost, low-energy consuming alternative to the PoW algorithm. It involves the allocation of responsibility in maintaining the public ledger to a participant node in proportion to the number of virtual currency tokens held by it. However, this comes with the drawback that it incentivizes cryptocoin hoarding instead of spending.

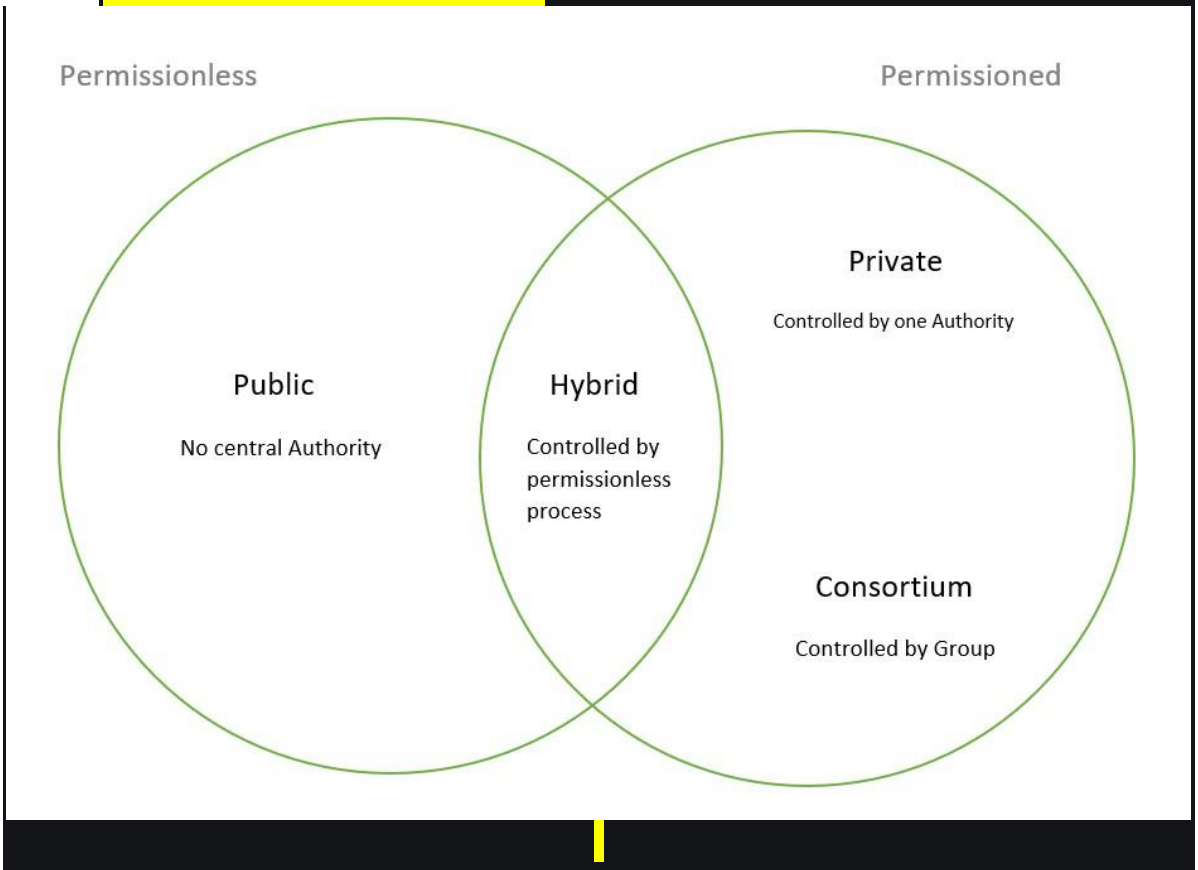
While PoW and PoS are by far the most prevalent in the blockchain space, there are other consensus algorithms like Proof of Capacity (PoC) which allow sharing of memory space of the contributing nodes on the blockchain network. The more memory or hard disk space a node has, the more rights it is granted for maintaining the public ledger. Proof of Activity (PoA), used on the Decred blockchain, is a hybrid that makes use of aspects of both PoW and PoS. Proof of Burn (PoB) is another that requires transactors to send small amounts of cryptocurrency to inaccessible wallet addresses, in effect "burning" them out of existence.

Another, called Proof of History (PoH), developed by the Solana Project and similar to Proof of Elapsed Time (PoET), encodes the passage of time itself cryptographically to achieve consensus without expending many resources.

4. **What are types of Blockchain?**

There are 4 types of blockchain:

- **Public Blockchain.**
- **Private Blockchain.**
- **Hybrid Blockchain.**
- **Consortium Blockchain.**



Let’s discuss each of these topics in detail.

1. Public Blockchain

These blockchains are completely open to following the idea of decentralization. They don’t have any restrictions, anyone having a computer and internet can participate in the network.

- As the name is public this blockchain is open to the public, which means it is not owned by anyone.
- Anyone having internet and a computer with good hardware can participate in this public blockchain.

- All the computer in the network hold the copy of other nodes or block present in the network
- In this public blockchain, we can also perform verification of transactions or records

Advantages:

- **Trustable:** There are algorithms to detect no fraud. Participants need not worry about the other nodes in the network
- **Secure:** This blockchain is large in size as it is open to the public. In a large size, there is greater distribution of records
- **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 in order to participate.
- **Decentralized:** There is no single platform that maintains the network, instead every user has a copy of the ledger.

Disadvantages:

- **Processing:** The rate of the transaction process is very slow, due to its large size. Verification of each node is a very time-consuming process.
- **Energy Consumption:** Proof of work is high energy-consuming. It requires good computer hardware to participate in the network
- **Acceptance:** No central authority is there so governments are facing the issue to implement the technology faster.

Use Cases: Public Blockchain is secured with proof of work or proof of stake they can be used to displace traditional financial systems. The more advanced side of this blockchain is the smart contract that enabled this blockchain to support decentralization. Examples of public blockchain are Bitcoin, Ethereum.

2. Private Blockchain

These blockchains are not as decentralized as the public blockchain only selected nodes can participate in the process, making it more secure than the others.

- These are not as open as a public blockchain.
- They are open to some authorized users only.
- These blockchains are operated in a closed network.
- In this few people are allowed to participate in a network within a company/organization.

Advantages:

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

Disadvantages:

- Security- The number of nodes in this type is limited so chances of manipulation are there. These blockchains are more vulnerable.
- Centralized- Trust building is one of the main disadvantages due to its central nature. Organizations can use this for malpractices.
- Count- Since there are few nodes if nodes go offline the entire system of blockchain can be endangered.

Use Cases: With proper security and maintenance, this blockchain is a great asset to secure information without exposing it to the public eye. Therefore companies use them for internal auditing, voting, and asset management. An example of private blockchains is Hyperledger, Corda.

3. Hybrid Blockchain

It is the mixed content of the private and public blockchain, where some part is controlled by some organization and other makes are made visible as a public blockchain.

- It is a combination of both public and private blockchain.
- Permission-based and permissionless systems are used.
- User access information via smart contracts
- Even a primary entity owns a hybrid blockchain it cannot alter the transaction

Advantages:

- **Ecosystem:** Most advantageous thing about this blockchain is its hybrid nature. It cannot be hacked as 51% of users don’t have access to the network
- **Cost:** Transactions are cheap as only a few nodes verify the transaction. All the nodes don’t carry the verification hence less computational cost.
- **Architecture:** It is highly customizable and still maintains integrity, security, and transparency.
- **Operations:** It can choose the participants in the blockchain and decide which transaction can be made public.

Disadvantages:

- **Efficiency:** Not everyone is in the position to implement a hybrid Blockchain. The organization also faces some difficulty in terms of efficiency in maintenance.
- **Transparency:** There is a possibility that someone can hide information from the user. If someone wants to get access through a hybrid blockchain it depends on the organization whether they will give or not.
- **Ecosystem:** Due to its closed ecosystem this blockchain lacks the incentives for network participation.

Use Case: It provides a greater solution to the health care 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 Ripple network and XRP token.

4. Consortium Blockchain

It is a creative approach that solves the needs of the organization. This blockchain validates the transaction and also initiates or receives transactions.

- Also known as Federated Blockchain.
- This is an innovative method to solve the organization's needs.
- Some part is public and some part is private.
- In this type, more than one organization manages the blockchain.

Advantages:

- **Speed:** A limited number of users make verification fast. The high speed makes this more usable for organizations.
- **Authority:** Multiple organizations can take part and make it decentralized at every level. Decentralized authority, makes it more secure.
- **Privacy:** The information of the checked blocks is unknown to the public view. but any member belonging to the blockchain can access it.
- **Flexible:** There is much divergence in the flexibility of the blockchain. Since it is not a very large decision can be taken faster.

Disadvantages:

- **Approval:** All the members approve the protocol making it less flexible. Since one or more organizations are involved there can be differences in the vision of interest.

- **Transparency:** It can be hacked if the organization becomes corrupt. Organizations may hide information from the users.
- **Vulnerability:** If few nodes are getting compromised there is a greater chance of vulnerability in this blockchain

2 Module: Cryptographic Primitives

1. **Explain properties of Cryptographic Hash Function.**

Properties of Hash Functions

In order to be an effective cryptographic tool, the hash function is desired to possess following properties −

- **Pre-Image Resistance**
  - This property means that it should be computationally hard to reverse a hash function.
  - In other words, if a hash function h produced a hash value z, then it should be a difficult process to find any input value x that hashes to z.
  - This property protects against an attacker who only has a hash value and is trying to find the input.
- **Second Pre-Image Resistance**
  - This property means given an input and its hash, it should be hard to find a different input with the same hash.
  - In other words, if a hash function h for an input x produces hash value h(x), then it should be difficult to find any other input value y such that h(y) = h(x).
  - This property of hash function protects against an attacker who has an input value and its hash, and wants to substitute different value as legitimate value in place of original input value.
- **Collision Resistance**
  - This property means it should be hard to find two different inputs of any length that result in the same hash. This property is also referred to as collision free hash function.
  - In other words, for a hash function h, it is hard to find any two different inputs x and y such that h(x) = h(y).
  - Since, hash function is compressing function with fixed hash length, it is impossible for a hash function not to have collisions. This property of collision free only confirms that these collisions should be hard to find.

- This property makes it very difficult for an attacker to find two input values with the same hash.
- Also, if a hash function is collision-resistant **then it is second pre-image resistant.**

2. **What is Hash Chain?**

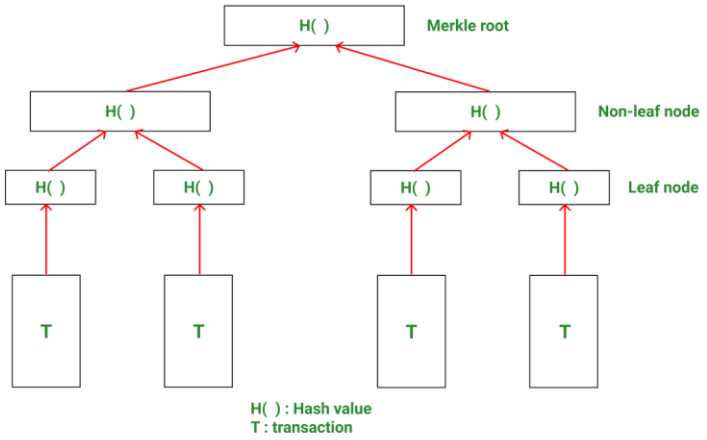
   A hash chain is commonly defined as the repeated application of a cryptographic hash function to a given data asset. This type of hash cryptography can be extremely useful in some specific security setups. By providing a successive chain, hash chains make it harder for a snooping hacker to hijack a data asset through applying a single input.

The idea of a hash chain is that a user supplies an individual input on the first interaction or session, and then adds authenticating data on the next session. Over a set of sessions, those individual hash inputs create a “hash chain” that authenticates a single user input in a more profound way.

As an example, hash chain processes can be similar to a blockchain ledger approach for bitcoin and other cryptocurrency, in that blockchain and other similar systems authenticate an input with previous hash key lists. However, other kinds of hash chains may not have the same specific features and details built into blockchain, which is becoming a gold standard for ledger transparency in the world of global finance.

3. **Explain Merkle Tree and its benefits.**

A hash tree is also known as **Merkle Tree.** It is a tree in which each leaf node is labeled with the hash value of a data block and each non-leaf node is labeled with the hash value of its child nodes labels. A Merkle tree is a binary tree formed by hash pointers, and named after its creator, Ralph Merkle.



The different types of nodes in a Merkle tree are:

- **Root node:** The root of the Merkle tree is known as the Merkle root and this Merkle root is stored in the header of the block.
- **Leaf node:** The leaf nodes contain the hash values of transaction data. Each transaction in the block has its data hashed and then this hash value (also known as transaction ID) is stored in leaf nodes.
- **Non-leaf node:** The non-leaf nodes contain the hash value of their respective children. These are also called intermediate nodes because they contain the intermediate hash values and the hash process continues till the root of the tree.

Further, a Merkle tree is binary in nature**. This means that the** number of leaf nodes needs to be even **Advantages of Merkle Tree**

1. **Efficient verification:** Merkle trees offer efficient verification of integrity and validity of data and 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. Enable trustless transfer of cryptocurrency in the peer-to-peer, distributed system by the quick verification of transactions.

2. **No delay:** There is no delay in the transfer of data across the network. Merkle trees are extensively 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.
4. **Unaltered transfer of data:** Merkle root helps in making sure that the blocks sent across the network are whole and unaltered.
5. **Tampering Detection:** Merkle tree gives an amazing advantage to miners to check whether any transactions have been tampered with.
    - Since the transactions are stored in a Merkle tree which stores the hash of each node in the upper parent node, any changes in the details of the transaction such as the amount to be debited or the address to whom the payment must be made, then the change will propagate to the hashes in upper levels and finally to the Merkle root.
    - The miner can compare the Merkle root in the header with the Merkle root stored in the data part of a block and can easily detect this tampering.
6. **Time Complexity:** Merkle tree is the best solution if a comparison is done between the time complexity of searching a transaction in a block as a Merkle tree and another block that has transactions arranged in a linked list, then-
    - **Merkle Tree search:** O(logn), where n is the number of transactions in a block.
    - **Linked List search:** O(n), where n is the number of transactions in a block.

4. **Explain Public Key Cryptography in detail.**

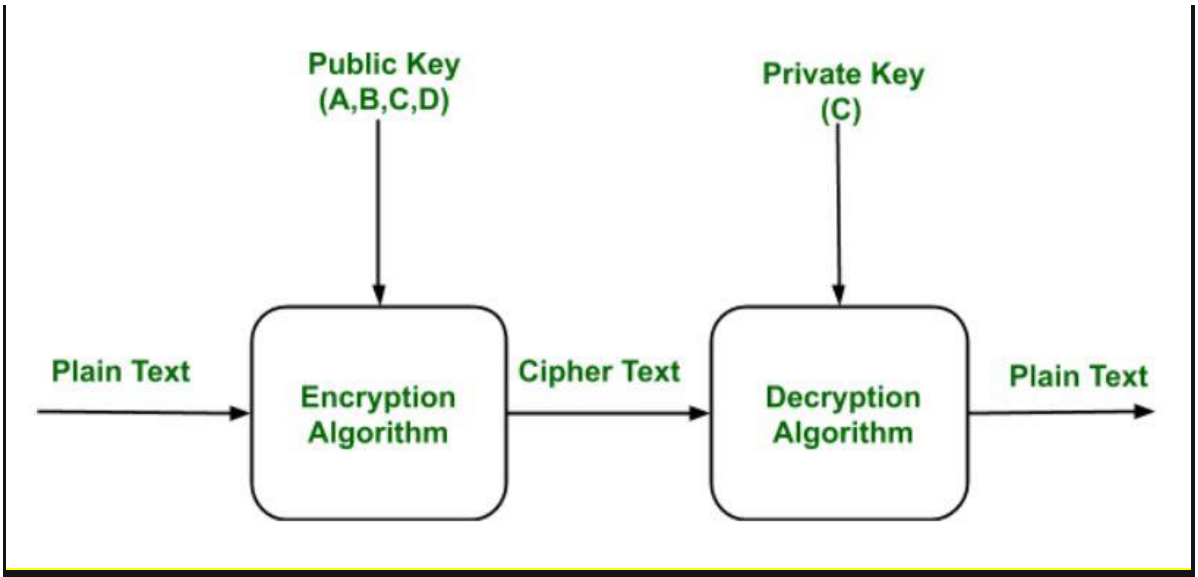
One algorithm is used for encryption and a related algorithm decryption with pair of keys, one for encryption and other for decryption.
Receiver and Sender must each have one of the matched pair of keys (not identical) .
One of the two keys must be kept secret.
If one of the key is kept secret, it is very impossible to decipher message.
Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be impractical to determine the other key.

Characteristics of Public Encryption key:

- Public key Encryption is important because it is infeasible to determine the decryption key given only the knowledge of the cryptographic algorithm and encryption key.
- Either of the two keys (Public and Private key) can be used for encryption with other key used for decryption.
- Due to Public key cryptosystem, public keys can be freely shared, allowing users an easy and convenient method for encrypting content and verifying digital signatures, and private keys can be kept secret, ensuring only the owners of the private keys can decrypt content and create digital signatures.
- The most widely used public-key cryptosystem is RSA (Rivest–Shamir–Adleman). The difficulty of finding the prime factors of a composite number is the backbone of RSA.

Example:

Public keys of every user are present in the Public key Register. If B wants to send a confidential message to C, then B encrypt the message using C Public key. When C receives the message from B then C can decrypt it using its own Private key. No other recipient other than C can decrypt the message because only C know C's private key.



Components of Public Key Encryption:

- **Plain Text:**
  This is the message which is readable or understandable. This message is given to the Encryption algorithm as an input.
- **Cipher Text:**
  The cipher text is produced as an output of Encryption algorithm. We cannot simply understand this message.
- **Encryption Algorithm:**
  The encryption algorithm is used to convert plain text into cipher text.
- **Decryption Algorithm:**
  It accepts the cipher text as input and the matching key (Private Key or Public key) and produces the original plain text
- **Public and Private Key:**
  One key either Private key (Secret key) or Public Key (known to everyone) is used for encryption and other is used for decryption

Weakness of the Public Key Encryption:

- Public key Encryption is vulnerable to Brute-force attack.
- This algorithm also fails when the user lost his private key, then the Public key Encryption becomes the most vulnerable algorithm.
- Public Key Encryption also is weak towards man in the middle attack. In this attack a third party can disrupt the public key communication and then modify the public keys.
- If user private key used for certificate creation higher in the PKI(Public Key Infrastructure) server hierarchy is compromised, or accidentally disclosed, then a “man-in-the-middle attack” is also

possible, making any subordinate certificate wholly insecure. This is also the weakness of public key Encryption.

Applications of the Public Key Encryption:

- **Encryption/Decryption:**
  Confidentiality can be achieved using Public Key Encryption. In this the Plain text is encrypted using receiver public key. This will ensure that no one other than receiver private key can decrypt the cipher text.
- **Digital signature:**
  Digital signature is for senders authentication purpose. In this sender encrypt the plain text using his own private key. This step will make sure the authentication of the sender because receiver can decrypt the cipher text using senders public key only.
- **Key exchange:**
  This algorithm can use in both Key-management and securely transmission of data.

5. **What are uses of hash function in Blockchain?**

The blockchain has a number of different uses for hash functions. Some of the most common uses of the hash function in blockchain are:

- **Merkle Tree:** This uses hash functions to make sure that it is infeasible to find two Merkle trees with the same root hash. This helps to protect the integrity of the block header by storing the root hash within the block header and thus protecting the integrity of the transactions.
- **Proof of Work Consensus:** This algorithm defines a valid block as the one whose block header has a hash value less than the threshold value.
- **Digital signatures:** Hash functions are the vital part of digital signatures that ensures data integrity and are used for authentication for blockchain transactions.
- **The chain of blocks:** Each block header in a block in the blockchain contains the hash of the previous block header. This ensures that it is not possible to change even a single block in a blockchain without being detected. As modifying one block requires generating new versions of every following block, thus increasing the difficulty.

6. **Explain uses of digital signature in Blockchain.**

Digital signing in blockchain technology aims to authenticate transactions. That means checking if the sender is authorized to make the transactions. The user has to prove to the network that he's authorized to spend his balance. The balance is also verified by checking the transactions made to the account.

Each node in the network verifies the submitted transaction and then makes an informed decision asking the whole network to add it. And that's how the use of digital signing network participants can authorize the transaction and account. The concept of digital signing helps to create digital certificates by certificate authorities (CA). Also, it's used for electronic signature (or E-signature) over documents. Moreover, adding public-key infrastructure (PKI) to digital signature generates a kind of multi-factor authentication system for the user. Various other encryption algorithms are implemented to ensure security.

The above article explains one of the biggest applications of hashing and public-key cryptography, Digital Signing. It deeply elaborates the process of digital signing using an example and how it can be secured using public-key cryptography. Moreover, it tells about how digital signing is used in blockchain networks for verifying transactions.

Hope this helped you understand some basic primitives of cryptography in the blockchain. Stay tuned to learn more concepts of blockchain.

3 Module: Bitcoin

1. **Explain Structure of Bitcoin block.**

   A Bitcoin block records the data related to a Bitcoin transaction. The blocks are mined one after the other with all the transactions in the network being recorded permanently. Being a very secure network, the Bitcoin blockchain makes it very difficult to modify or delete the data that has been registered on a block. More or less, a block acts like a book of the ledger with each page referring to one block and connected to both the previous and the next blocks by the addressed called “hashes”. A blockchain is created by linking these individual block ledgers one after the other. A block structure has several elements.

   A block structure has several elements.

. Block Identifiers

2. Block Header

3. Transactions

1. Block Identifiers: The block identifiers are the elements that identify a block’s address, its height, and its size. There are the main block identifiers:
a. Hash b. Block Height

2. Block Header: A block in a blockchain is identified by its unique block header. In the Proof of Work mechanism, the block headers are hashed many times to generate a unique hash for each block header. This block header hash becomes the identifier of the block. The Bitcoin block header length is 80 bytes and it consists of the following metadata:

a. 4-byte Version

b. 4-byte Timestamp

c. 4-byte Difficulty Target

d. 4-byte Nonce

e. 32-byte Previous Block Hash

f. 32-byte Merkle Root

3. Transactions: When a block is added to a blockchain, it is called a confirmed transaction or a confirmation. After a transaction, when the other transactions get confirmed, that initial transaction gets further confirmed. A Bitcoin block is mined every 10 minutes.

2. **What is double spending program and how to solve it?**

Double spending means spending the same money twice. As we know, any transaction can be processed only in two ways. One is offline, and another is online.

Offline: A transaction which involves physical currency or cash is known as an offline transaction.

Online: A transaction which involves digital cash is known as an online transaction.

In a physical currency, the double-spending problem can never arise. But in digital cash-like bitcoin, the double-spending problem can arise. Hence, bitcoin transactions have a possibility of being copied and rebroadcasted. It opens up the possibility that the same BTC could be spent twice by its owner. Bitcoin handles the double-spending problem by implementing a confirmation mechanism and maintaining a universal ledger called blockchain.

Let us suppose you have 1 BTC and try to spend it twice. You made the 1 BTC transaction to Alice. Again, you sign and send the same 1 BTC transaction to Bob. Both transactions go into the pool of unconfirmed transactions where many unconfirmed transactions are stored already. The unconfirmed transactions are transactions which do not pick by anyone. Now, whichever transaction first got confirmations and was verified by miners, will be valid. Another transaction which could not get enough confirmations will be pulled out from the network.

3. **Explain Script (Bitcoin Scripting Language) in detail.**
   El Bitcoin Script is the language that Bitcoin uses to do everything it can do, from sending funds from a wallet to allowing the creation of multi-user accounts. All these functions contained in a simple and powerful extensible tool.

This is based on a series of linear structures, known as **stack**, which contain existing data in order **LIFO (Last In - Firt Out)**. Each instruction in this language is executed one after the other consecutively.

This language is not Full Turing because its functionality is limited and cannot loops. So it is not capable of solving any type of problem such as **turing machines**. However, this limitation is intentional as this prevents infinite or endless looping and error execution. Where malicious parts of the program can be free to create complicated operations to consume the rate of **hash** and slow down the Bitcoin system through infinite loops.

A programming language is necessary because it allows us to write programs and that computers execute our wishes. In **Bitcoin**In order to communicate our wishes, the **opcodes (OP CODES)**, which serve various functions. Like memory manipulation, math, loops, function calls, among many others.

Therefore, **Bitcoin Script is essentially a set of programmed instructions that are recorded with every transaction made**. **The purpose of creating a script language in Bitcoin is to provide a series of easy and flexible parameters to enable a transaction.** So when **Satoshi Nakamoto** developed Bitcoin, disabled various functions, including multiplication. So the script is kept simple in terms of programming. So, **it is this programming language that determines whether or not an operation can be performed**. **The Bitcoin script also prevents the creation of errors in the system and the unnecessary use of very complex transactions.**

4. **Explain importance of consensus mechanism in Blockchain.**

Consensus mechanisms form the backbone of all cryptocurrency blockchains, and are what make them secure. Before we delve into the different consensus mechanisms, we need to first define what it means for blockchains to achieve consensus.

A blockchain is a decentralised, distributed, and oftentimes public digital ledger that is used to record transactions. Each of these transactions is recorded as a 'block' of data, which **needs to be independently verified by peer-to-peer computer networks before they can be added to the chain**. This system helps to secure the blockchain against fraudulent activity and addresses the problem of 'double-spending'.

In order to guarantee that all participants ('nodes') in a blockchain network agree on a single version of history, blockchain networks like Bitcoin and Ethereum implement what's known as consensus mechanisms (also known as consensus protocols or consensus algorithms). These mechanisms aim to make the system fault-tolerant.

5. **Explain distributed consensus algorithm used in Permissionless Blockchain. (Proof of Work, Proof of Stake, Proof of Elapsed Time, Proof of Stake, Proof of Burn) .**
6. **Explain Proof of Work.**
7. **Explain Proof of Stake.**

Consensus is the process by which a group of peers – or nodes – on a network determine which blockchain transactions are valid and which are not. Consensus mechanisms are the methodologies used to achieve this agreement. It's these sets of rules that help to protect networks from malicious behaviour and hacking attacks. There are many different types of consensus mechanisms, depending on the blockchain and its application. While they differ in their energy usage, security, and scalability, they all share one purpose: to ensure that records are true and honest. Here's an overview of some of the best-known types of consensus mechanisms used by distributed systems to reach consensus.

Proof of Work (PoW)

Used by Bitcoin, Ethereum, and many other public blockchains, proof of work (PoW) was the very first consensus mechanism created. It is generally regarded to be the most reliable and secure of all the consensus mechanisms, though concerns over scalability are rife. While the term 'proof of work' was first coined in the early 1990s, it was Bitcoin founder Satoshi Nakamoto that first applied the technology in the context of digital currencies.

In PoW, miners essentially compete against one another to solve extremely complex computational puzzles using high-powered computers. The first to come up with the 64-digit hexadecimal number ('hash') earns the right to form the new block and confirm the transactions. The successful miner is also rewarded with a predetermined amount of crypto, known as a 'block reward'.

As it requires large amounts of computational resources and energy in order to generate new blocks, the operating costs behind PoW are notoriously high. This acts as a barrier of entry for new miners, leading to concerns about centralisation and scalability limitations.

And it's not just the costs that are high. The most common criticism of PoW is the impact the electrical consumption has on the environment. This has led many to seek more sustainable, energy-efficient consensus protocols, such as proof of stake (PoS).

Proof of Stake (PoS)

As the name suggests, this popular method of consensus revolves around a process known as staking. In a proof of stake (PoS) system, miners are required to pledge a 'stake' of digital currency for a chance to be randomly chosen as a validator. The process is not unlike a lottery whereby the more coins you stake, the better your odds.

Unlike in PoW where miners are incentivised by block rewards (newly generated coins), those who contribute to the PoS system simply earn a transaction fee.

PoS is seen as a more sustainable and environmentally-friendly alternative to PoW, and one that's more secure against 51% attack. However, as the system favours entities with a higher number of tokens, PoS has drawn criticism for

its potential to lead to centralisation. Prominent PoS platforms include Cardano (ADA), Solana (SOL), and Tezos (XTC).

Delegated Proof of Stake (DPoS)

A modification of the PoS consensus mechanism, delegated proof of stake (DPoS) relies upon a reputation-based voting system to achieve consensus. Users of the network 'vote' to select 'witnesses' (also known as 'block producers') to secure the network on their behalf. Only the top tier of witnesses (those with the most votes) earn the right to validate blockchain transactions.

To vote, users add their tokens to a staking pool. Votes are then weighted according to the size of each voter's stake – so the more skin in the game, the more voting power. Elected witnesses who successfully verify transactions in a block receive a reward, which is usually shared with those who voted for them.

Witnesses in the top tier are always at risk of being replaced by those deemed more trustworthy and who therefore get more votes. They can even be voted out if they fail to fulfil their responsibilities or try to validate fraudulent transactions. This helps to incentivise witnesses to remain honest at all times, ensuring the integrity of the blockchain.

Though less prevalent than PoS, DPoS is regarded by many as being more efficient, democratic, and financially inclusive than its predecessor. It is used by Lisk (LSK), EOS.IO (EOS), Steem (STEEM), BitShares (BTS), and Ark (ARK).

Proof of Burn (PoB)

Another more sustainable alternative to Bitcoin's PoW algorithm is proof of burn (PoB). In PoB, miners gain the power to mine a block by 'burning' (destroying) a predetermined amount of tokens in a verifiable manner – namely, sending them to an 'eater address' where they cannot be recovered or spent. The more coins burned, the greater the chances of being randomly selected.

Unlike in PoS where miners are able to retrieve or sell their locked coins should they ever leave the network, burned coins are irretrievably lost. This method of requiring miners to sacrifice short-term wealth in order to gain the lifetime privilege to create new blocks helps to encourage long-term commitment from miners. The act of burning coins also leads to coin scarcity, limiting inflation and driving up demand.

Cryptocurrencies that use the proof of burn protocol include Slimcoin (SLM), Counterparty (XCP), and Factom (FCT).

Proof of Elapsed Time (PoET)

Usually used on permissioned blockchain networks (those that require participants to identify themselves), proof of elapsed time (PoET) leverages trusted computing to enforce random waiting times for block construction. It was developed by Intel in early 2016 and is based on a special set of CPU instructions called Intel software guard extensions (SGX).

A time-lottery-based consensus algorithm, PoET works by randomly assigning different wait times to every node in the network. During the waiting period, each of these nodes goes to 'sleep' for that specified duration. The first to wake up (that is, the one with the shortest waiting time) is awarded the mining rights. This randomisation guarantees that every participant is equally as likely to be the winner, ensuring fairness within the network.

The PoET consensus mechanism is highly efficient, less resource-intensive, and scalable. It has been implemented in Hyperledger's Sawtooth.

8. **What consensus algorithm used in Bitcoin Blockchain?**

On the Bitcoin blockchain, for instance, the consensus mechanism is known as Proof-of-Work (PoW), which requires the exertion of computational power in order to solve a difficult but arbitrary puzzle in order to keep all nodes in the network honest.

Proof of Work (PoW)

Used by Bitcoin, Ethereum, and many other public blockchains, proof of work (PoW) was the very first consensus mechanism created. It is generally regarded to be the most reliable and secure of all the consensus mechanisms, though concerns over scalability are rife. While the term 'proof of work' was first coined in the early 1990s, it was Bitcoin founder Satoshi Nakamoto that first applied the technology in the context of digital currencies.

In PoW, miners essentially compete against one another to solve extremely complex computational puzzles using high-powered computers. The first to come up with the 64-digit hexadecimal number ('hash') earns the right to form the new block and confirm the transactions. The successful miner is also rewarded with a predetermined amount of crypto, known as a 'block reward'.

As it requires large amounts of computational resources and energy in order to generate new blocks, the operating costs behind PoW are notoriously high. This acts as a barrier of entry for new miners, leading to concerns about centralisation and scalability limitations.

And it's not just the costs that are high. The most common criticism of PoW is the impact the electrical consumption has on the environment. This has led many to seek more sustainable, energy-efficient consensus protocols, such as proof of stake (PoS).

9. **Explain the mining process in detail.**

Mining is the process that Bitcoin and several other cryptocurrencies use to generate new coins and verify new transactions. It involves vast, decentralized networks of computers around the world that verify and secure blockchains – the virtual ledgers that document cryptocurrency transactions. In return for contributing their processing power, computers on the network are rewarded with new coins. It's a virtuous circle: the miners maintain and secure the blockchain, the blockchain awards the coins, the coins provide an incentive for the miners to maintain the blockchain.

How does mining work?

There are three primary ways of obtaining bitcoin and other cryptocurrencies. You can buy them on an exchange like Coinbase, receive them as payment for goods or services, or virtually "mine" them. It's the third category that we're explaining here, using Bitcoin as our example.

You might have considered trying bitcoin mining yourself. A decade ago, anyone with a decent home computer could participate. But as the blockchain has grown, the computational power required to maintain it has increased. (By a lot: In October 2019, it required 12 trillion times more computing power to mine one bitcoin than it did when the first first blocks were mined in January 2009.) As a result, amateur bitcoin mining is unlikely to be profitable for hobbyists these days. Virtually all mining is now done by specialized companies or groups of people who band their resources together. But it's still good to know how it works.

- Specialized computers perform the calculations required to verify and record every new bitcoin transaction and ensure that the blockchain is secure. Verifying the blockchain requires a vast amount of computing power, which is voluntarily contributed by miners.
- Bitcoin mining is a lot like running a big data center. Companies purchase the mining hardware and pay for the electricity required to keep it running (and cool). For this to be profitable, the value of the earned coins has to be higher than the cost to mine those coins.
- What motivates miners? The network holds a lottery. Every computer on the network races to be the first to guess a 64-digit hexadecimal number known as a "hash." The faster a computer can spit out guesses, the more likely the miner is to earn the reward.
- The winner updates the blockchain ledger with all the newly verified transactions – thereby adding a newly verified "block" containing all of those transactions to the chain – and is granted a predetermined amount of newly minted bitcoin. (On average, this happens every ten minutes.) As of late 2020, the reward was 6.25 bitcoin – but it will be reduced by half in 2024, and every four years after. In fact, as the difficulty of mining increases, the reward will keep decreasing until there are no more bitcoin left to be mined.
- There will only ever be 21 million bitcoin. The final block should theoretically be mined in 2140. From that point forward, miners will no

longer rely on newly issued bitcoin as reward, but instead will rely on the fees they charge for making transactions.

Why is mining important?

Beyond releasing new coins into circulation, mining is central to Bitcoin's (and many other cryptocurrencies') security. It verifies and secures the blockchain, which allows cryptocurrencies to function as a peer-to-peer decentralized network without any need for oversight from a third party. And it creates the incentive for miners to contribute their computing power to the network.

4. Module: Permission Blockchain

1. What is Smart Contract? Explain in detail.

Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met.

Smart contracts work by following simple "if/when…then…" statements that are written into code on a blockchain. A network of computers executes the actions when predetermined conditions have been met and verified. These actions could include releasing funds to the appropriate parties, registering a vehicle, sending notifications, or issuing a ticket. The blockchain is then updated when the transaction is completed. That means the transaction cannot be changed, and only parties who have been granted permission can see the results.

Within a smart contract, there can be as many stipulations as needed to satisfy the participants that the task will be completed satisfactorily. To establish the terms, participants must determine how transactions and their data are represented on the blockchain, agree on the "if/when...then…" rules that govern those transactions, explore all possible exceptions, and define a framework for resolving disputes.

Then the smart contract can be programmed by a developer – although increasingly, organizations that use blockchain for business provide templates, web interfaces, and other online tools to simplify structuring smart contracts.

Benefits of smart contracts

Speed, efficiency and accuracy

Once a condition is met, the contract is executed immediately. Because smart contracts are digital and automated, there's no paperwork to process and no time spent reconciling errors that often result from manually filling in documents.

Trust and transparency

Because there's no third party involved, and because encrypted records of transactions are shared across participants, there's no need to question whether information has been altered for personal benefit.

Security

Blockchain transaction records are encrypted, which makes them very hard to hack. Moreover, because each record is connected to the previous and subsequent records on a distributed ledger, hackers would have to alter the entire chain to change a single record.

Savings

Smart contracts remove the need for intermediaries to handle transactions and, by extension, their associated time delays and fees.

2. **Explain distributed consensus algorithm used in Permissioned Blockchain. (Paxos, RAFT, Byzantine Fault Tolerance, Practical BFT)**

A distributed consensus ensures a consensus of data among nodes in a distributed system or reaches an agreement on a proposal. This topic may be very familiar to any technicians that work with distributed systems such as HDFS, MQ, ZooKeeper, Kafka, Redis, and Elasticsearch. With the rapid development and the increasing complexity of distributed networks, developers have always been exploring possible solutions to solve this persistent problem in both theory and practice.

Next, with the rise of blockchain technology, especially public blockchains in open networks and private blockchains in permissioned networks, this consensus problem has once again received much attention and needs to be considered from a new perspective.

In this article, we will look into the problems and challenges of distributed consensus and corresponding consensus algorithms. We will also briefly analyze the applicability and limitations of these consensus algorithms and discuss the combination of these traditional consensus algorithms and new blockchain technologies. Later, this article concentrates on the consensus algorithm and mechanism in the public blockchain field from the perspective of the reliability of human beings. This article also considers the association between the distributed consensus algorithms in traditional computer science and the consensus mechanism in blockchain and shows how new consensus ideas can be seen in the public blockchain field.

Problems and Challenges of Distributed Consensus

The Paxos Algorithm

One of the most famous distributed consensus algorithms is Paxos suggested by Lamport, though its complexity is also "notorious". Lamport proposed this creative mechanism that is practical and can be implementable through engineering and can ensure the consistency of distributed systems to the maximum extent. Paxos is widely used in many distributed systems, including

Chubby and ZooKeeper. Basic Paxos (single decree, that is, to only agree on a value each time) has two roles: A Proposer can process client request and actively propose a proposal value. An Acceptor passively responds to the information sent by a Proposer, votes on proposals made, and persists values and states during the decision-making process. To simplify the model, the Learner role can be ignored. This does not affect the decision-making in the model.

The Raft Algorithm

Due to the complexity of Paxos, Ongaro presented a simpler algorithm in 2014—Raft. Raft is much easier to understand and implement in engineering. This is also the initial objective of Raft. Many easy-to-understand design details have been made on the condition that the functionality is not affected.

The Raft algorithm is a leader-based asymmetric model. A node in a system can only be in one of the three states at any point in time: leader, follower, and candidate. In the initial stage, all the nodes are followers. To become the leader, a node (follower) must become a candidate and launch a round of electoral votes. If the node does not receive enough votes, the node becomes a follower again. However, if it receives a majority of the votes, the node becomes the leader. If the leader encounters failures and finds that a new leader is elected after it recovers from failures, the original leader automatically goes back to the follower state.

Raft also introduces the Term concept to identify expired information in a timely manner. A term is similar to an epoch file in ZooKeeper. A term number increases monotonically over time and at most one leader can be elected in a given term. If the logs have a last entry with different terms, then the log with the later term is more up-to-date.

Raft also introduces the heartbeat packet and timeout. To maintain its authority, an elected leader must continuously send a heartbeat packet to the other nodes in the cluster. If a follower does not receive the heartbeat packet during a given election timeout, the leader is considered to have crashed and the follower changes its status to candidate and starts a leader election.

The leader election in Raft is implemented through the heartbeat and random timeout. Log replication is implemented through strong leadership: The leader receives the client command, appends it to its log and replicates the log to other followers. Raft ensures safety by only allowing a leader to decide whether to commit a log or not.

Election and replication will not be described here in detail. For more information about election and replication in Raft. Note that the leader election and the normal operations orchestrated by the leader are relatively simple. The leader change process is actually a bit more complex in Raft.

However, although the principle/mechanism of Raft is not exactly the same as Paxos, the problems that they solve and the trade-off policies that they adopt can be considered similar. That is to say, Raft can only solve crash faults, emphasizing fault tolerance, safety, and consistency and weakens the level of liveness and availability.

Practical Byzantine Fault Tolerance (PBFT)

Although many discussions on BFT solutions have been conducted since the Byzantine Generals' Problem raised by Lamport in 1982, many solutions for these problems are inefficient, slow and complex. The situation improved in 1999 when Castro and Liskov presented the Practical Byzantine Fault Tolerance (PBFT) algorithm. PBFT is the first algorithm of its kind with the complexity reduced from the exponential level to the polynomial level. PBFT enables several thousand TPS and feasible solutions to nodes acting maliciously in practice. It is proven that the PBFT algorithm will work normally if the number of malicious nodes in a system is no more than 1/3 of the total nodes.

All the nodes in a PBFT system are ordered sequentially with one node being the leader node and others considered as backup nodes. All the nodes in a system communicate with each other and reach consensus based on the majority principle. Each PBFT consensus round is called a view. The leading node is changed during every view and can be replaced with a protocol called a view change if a certain amount of time has passed without the leading node broadcasting the request. This replica timeout mechanism ensures that the

crashed or malicious leader can be detected and that a new view starts by re-electing a new leader.

Byzantine Fault Tolerance

Then, there's Byzantine fault tolerance (BFT), which related to the entire distributed network being evaluated in a larger environment. In addition to physical hardware, it is also necessary to take some "man-made" factors. After all, it is specific persons instead of machines that perform misconduct. Assume that a distributed network is relatively open, for example, a private network of tens of companies in a specific industry. Or assume a completely open network, for example, a network that anyone has access to. Node machines and software on these machines are deployed by individual companies or individuals themselves. If the benefit is tempting enough, a person may launch DDoS attacks on one of these nodes, making authorized, often malicious changes to software code and to the code execution logic, or even the data that is persisted on disks in the network. In such case, we face bigger challenges. In addition to the unreliable communication networks and machine hardware, we need to consider and deal with the "troublemakers" in the system.

3. What is RAFT?

Due to the complexity of Paxos, Ongaro presented a simpler algorithm in 2014—Raft. Raft is much easier to understand and implement in engineering. This is also the initial objective of Raft. Many easy-to-understand design details have been made on the condition that the functionality is not affected.

The Raft algorithm is a leader-based asymmetric model. A node in a system can only be in one of the three states at any point in time: leader, follower, and candidate. In the initial stage, all the nodes are followers. To become the leader, a node (follower) must become a candidate and launch a round of electoral votes. If the node does not receive enough votes, the node becomes a follower again. However, if it receives a majority of the votes, the node becomes the leader. If the leader encounters failures and finds that a new leader is elected after it recovers from failures, the original leader automatically goes back to the follower state.

Raft also introduces the Term concept to identify expired information in a timely manner. A term is similar to an epoch file in ZooKeeper. A term number increases monotonically over time and at most one leader can be elected in a given term. If the logs have a last entry with different terms, then the log with the later term is more up-to-date.

Raft also introduces the heartbeat packet and timeout. To maintain its authority, an elected leader must continuously send a heartbeat packet to the other nodes in the cluster. If a follower does not receive the heartbeat packet during a given election timeout, the leader is considered to have crashed and the follower changes its status to candidate and starts a leader election.

The leader election in Raft is implemented through the heartbeat and random timeout. Log replication is implemented through strong leadership: The leader receives the client command, appends it to its log and replicates the log to other followers. Raft ensures safety by only allowing a leader to decide whether to commit a log or not.

Election and replication will not be described here in detail. For more information about election and replication in Raft. Note that the leader election

and the normal operations orchestrated by the leader are relatively simple. The leader change process is actually a bit more complex in Raft.

However, although the principle/mechanism of Raft is not exactly the same as Paxos, the problems that they solve and the trade-off policies that they adopt can be considered similar. That is to say, Raft can only solve crash faults, emphasizing fault tolerance, safety, and consistency and weakens the level of liveness and availability.

4. Explain Byzantine Fault Tolerance.

Byzantine Fault Tolerance

Then, there's Byzantine fault tolerance (BFT), which related to the entire distributed network being evaluated in a larger environment. In addition to physical hardware, it is also necessary to take some "man-made" factors. After all, it is specific persons instead of machines that perform misconduct. Assume that a distributed network is relatively open, for example, a private network of tens of companies in a specific industry. Or assume a completely open network, for example, a network that anyone has access to. Node machines and software on these machines are deployed by individual companies or individuals themselves. If the benefit is tempting enough, a person may launch DDoS attacks on one of these nodes, making authorized, often malicious changes to software code and to the code execution logic, or even the data that is persisted on disks in the network. In such case, we face bigger challenges. In addition to the unreliable communication networks and machine hardware, we need to consider and deal with the "troublemakers" in the system.

5. Explain Practical Byzantine Fault Tolerance.

Practical Byzantine Fault Tolerance (PBFT)

Although many discussions on BFT solutions have been conducted since the Byzantine Generals' Problem raised by Lamport in 1982, many solutions for these problems are inefficient, slow and complex. The situation improved in 1999 when Castro and Liskov presented the Practical Byzantine Fault Tolerance (PBFT) algorithm. PBFT is the first algorithm of its kind with the complexity reduced from the exponential level to the polynomial level. PBFT enables several thousand TPS and feasible solutions to nodes acting maliciously in practice. It is proven that the PBFT algorithm will work normally if the number of malicious nodes in a system is no more than 1/3 of the total nodes.

All the nodes in a PBFT system are ordered sequentially with one node being the leader node and others considered as backup nodes. All the nodes in a system communicate with each other and reach consensus based on the majority principle. Each PBFT consensus round is called a view. The leading node is changed during every view and can be replaced with a protocol called a view change if a certain amount of time has passed without the leading node broadcasting the request. This replica timeout mechanism ensures that the crashed or malicious leader can be detected and that a new view starts by re-electing a new leader.

6. Differentiate between Permission Blockchain and Permissionless Blockchain.

Permissionless Blockchains

Permissionless blockchain ledgers are notable for their radical decentralization. This includes mechanisms for full transparency of transactions, open-source development models, and a lack of central authority.

Permissionless blockchains have a few key benefits:

- **Open Architecture**: Permissionless blockchains do not rely on a central authority to manage the network. This means that any user can add a node, and this node can participate in the network.
- **Transparency**: Every transaction is open and visible to any node on the network—a critical part of maintaining auditable transactions. Conversely, however, this also means that any sensitive data put into a transaction is also completely visible.
- **Pseudo-Anonymity:** Users on permissionless systems can remain semi-anonymous. That is, the user can participate through the use of an alphanumeric ID, and so long as no real connection is drawn between that ID and the user, it's difficult to trace one from the other.
- **Radical Decentralization:** The decentralization of user activity and transaction verification means that the network can grow quickly without the central authority to manage it. Everything operates via peer-to-peer mechanisms.
- **Network Resilience:** While transaction data isn't very secure, the network is resilient to attack. Hackers can only overwhelm the network by controlling 51% or more of the nodes, which is prohibitively difficult in a massive network.

Permissioned Blockchain

Permissioned blockchains are closed versions of their permissionless counterparts. While this changes the landscape of what they can do, it also empowers them to serve in many different roles that can bolster decentralized enterprise applications.

At its heart, a permissioned blockchain is one where a central authority controls aspects of the network, from user access to data encryption and access, typically through making the blockchain private.

Some of the benefits of permissionless blockchains include the following:

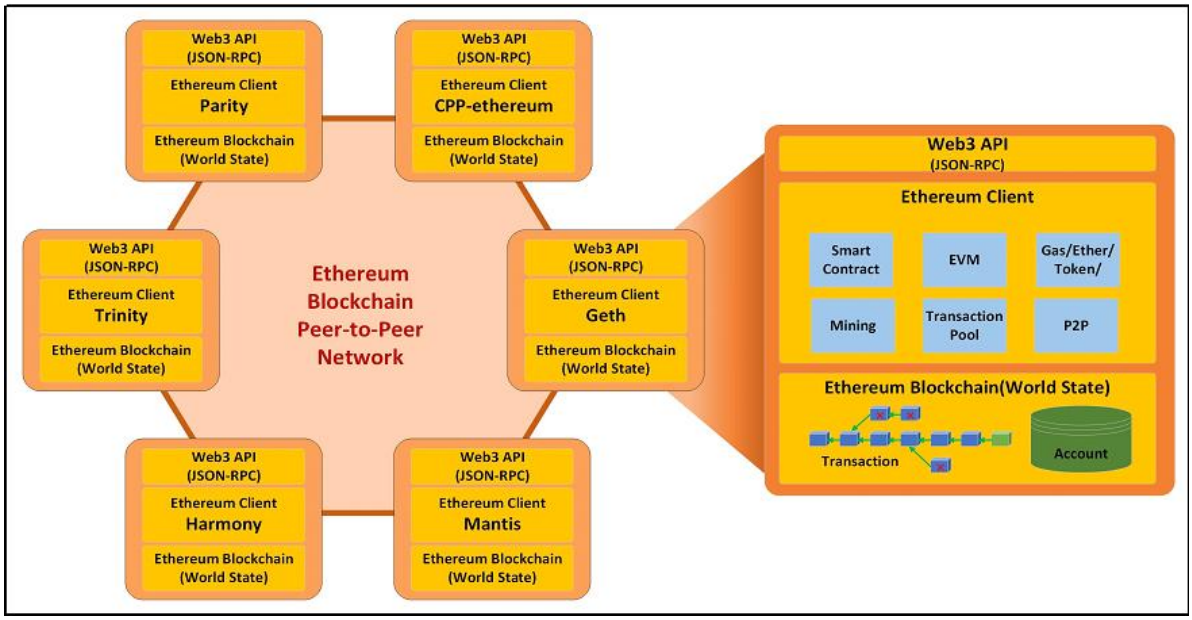
- **Security for Sensitive Data:** While permissionless systems provide robust network resilience, the trade off is that all data is transparent. Permissioned blockchains trade radical decentralization for more security around private information. This makes a permission system viable for storing personal data, login and identity credentials on the blockchain, etc.
- **Customization:** Permissionless systems are simple and overarching in their scope, meaning that they usually provide a baseline of functionality that can work over a huge scope of work. Permissioned systems, on the other hand, provide central operators more control over how that ledger works, what it stores, and who can use it.
- **Faster**: Permissionless systems, especially those that rely on proof-of-work verification systems, are painfully slow with any appreciable scaling. Permissioned systems, on the other hand, can scale readily and with the need of their organization.
- **Decentralization:** While the decentralizing capabilities of a permissioned blockchain aren't as vast as those for their permissionless counterparts, they are still decentralized. This means that they don't provide the same honeypot that, say, a database system might.

In general, enterprise permissioned blockchains, while less public and decentralized than their counterparts, are more secure, scalable, and configurable.

5. Module: Ethereum

1. **Explain architecture of Ethereum.**

The Ethereum Foundation envisioned Ethereum as a decentralized computing platform that enables anyone to create, store, and run smart contract-based Decentralized Applications, or DApps.



Ethereum clients run the EVM and can technically be written in any popular programming language. There are many different implementations of Ethereum clients. Ethereum makes it possible for such a variety of different client implementations since every implementation has to conform to the protocol specification defined in the Ethereum Yellow Paper (https://github.com/ethereum/yellowpaper). There are many advantages with such a variety of Ethereum client implementations, including the following:

- It makes the network more resilient against bugs.
- It prevents the centralization of developer resources.
- In general, competitions between teams help to find the best solutions to common and challenging issues.
- Each client may have a different focus, strength, and weakness in mining, prototyping, DApp development, and more. DApp developers or private Ethereum blockchain operators may choose the ones fitting their own special needs.

Ethereum clients provide a set of web3 APIs over JSON-RPC for DApps interacting with an Ethereum blockchain. From your web or wallet application, you can use the web3 object provided by the web3.js library to communicate with the Ethereum network. It works with any Ethereum client. Behind the scenes, it connects to a local or remote Ethereum node and makes RPC calls. In some sense, this is like the old client-server model, where DApps are the client, and the entire Ethereum network as a whole, acts as a server. To DApps, the Ethereum network is just like a giant world computer, assembled together with thousands of computing devices throughout the internet. Once you connect to the network, you could connect to any node in the decentralized network.

Beyond smart contracts and the EVM, an Ethereum client provides all blockchain components to maintain world state and state transitions in the blockchain network, including the following:

- Managing transaction and state transition with the Ethereum blockchain
- Maintaining world state and account state
- Managing P2P communication Block finalization with mining
- Managing transaction pool
- Managing cryptoassets, gas, ether, and tokens

2. **Explain Ethereum Virtual Machine.**

Ethereum Virtual Machine (EVM) is designed as the runtime environment for smart contracts in Ethereum. It is sandboxed and isolated from the other parts of the system. This means that any operation on EVM should not affect your data or programs in any way, no matter how many times you call a particular function on it.

- An EVM is the runtime environment that executes Ethereum smart contracts.
- Ethereum contains its own Turing-complete scripting language, called Solidity, and with this comes a need to execute this code.
- A program called the Ethereum Virtual Machine (EVM) can do this task.
- It runs on top of the Ethereum network, meaning that all nodes reach a consensus about what code should be executed at every given time.

How Does EVM Works?

Ethereum Virtual Machine (EVM) is a program which executes scripts used to implement certain operations usually in Ethereum blockchain. The Ethereum Virtual Machine makes the process of creating new tokens on Ethereum Blockchain easy. Here, script means a set of instructions or an algorithm which tells the computer what it needs to do in order for something to work properly. The EVM requires that one has access over any network node so as to be able to execute the desired commands and create new tokens on the blockchain without any difficulties.

- In Ethereum, there is something called a smart contract. These contracts have some computer code which facilitates the exchange of money and information.
- These contracts are predefined by the creator of the smart contract, in order to ensure that a certain outcome will happen based on either what happens or doesn't happen.
- Ethereum Virtual Machine provides Turing complete environment for execution of scripts and smart contracts. This means that anything that can be implemented with a computer can be run on EVM.

Ethereum Virtual Machine (EVM) has two parts:

- **EVM (the part that runs solidity source code):** The EVM is written in C++ and uses LLVM as its compiler. It is a full-featured virtual machine with all the features that you would want in a general purpose Smart Contract Virtual Machine, such as support for

multiple programming languages, security features, runtime environments and more. It also allows you to write custom EVM bytecode .

- **Uncles:** These are small pieces of smart contracts or data stored on the blockchain. This is a useful feature because it allows for you to store metadata about your program. EVM Assembly: This is the bytecode of EVM, which you can use as your programming language.

Purpose of EVM

The Ethereum Virtual Machine (EVM) is a Turing complete programmable machine, which can execute scripts to produce arbitrary outcomes. It has been built with the purpose of being a "world computer" and has immense power.

- It is the computer that stores data on blockchain, like bitcoin, but it also executes code in smart contracts on the Ethereum network.
- The machine is made to be able to run any kind of Crypto-contract that can be built on Ethereum's blockchain. It does this by using a programming language called Solidity, which is compiled into the EVM for execution.
- The intention behind writing code on the Ethereum network is to create smart contracts and programs that automatically execute things when certain conditions are met. If a terms or condition is not met, the system can execute it in an "exit" function as well.
- For example, if an account has been hacked, the hacker cannot steal money from the system, because they don't have the budget or authority to do so.

3. **Explain Ethereum Mining Process.**

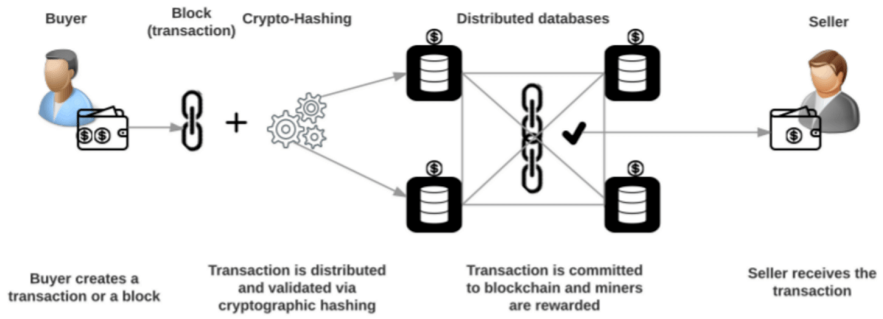
- Ethereum Mining is a process of creating and adding a block of transactions to the blockchain network of Ethereum.
- Currently, it uses the Proof-of-Work consensus mechanism.
- All transactions taking place in the Ethereum network need to get approved by the miners.
- Miners use a Hashing Scrypt (Ethash) to solve computationally hard puzzles for successfully mining the blocks of transactions, in the Ethereum Blockchain Network.
- This process helps secure the network from attacks like hacking or manipulation of identity.
- **Types of Ethereum Mining**

There are three ways of performing Ethereum Mining :

- Pool Mining
- Solo Mining
- Cloud Mining

How Does the Mining Process Work?

Ethereum Mining follows a specific set of steps to function.



1. A user requests a transaction with the help of the private key of his digital wallet account.
2. Then the request is shared worldwide with the Ethereum network.
3. Next, the requested transaction is added to a list of pending transactions that need to be added to the Ethereum blockchain network.
4. The miner then verifies and validates the requested transaction and performs a complex mathematical puzzle on the transaction data.
5. Once the requested transaction is verified and it stores a copy of it in EVM, the process of "Proof-of-Work" begins for the respective block.
6. Then, the nodes of the Ethereum Network verify that the checksum of the state of the miner's block matches the checksum of their updated state of EVM after execution of all transactions.
7. Only after that, the block is added to the Ethereum Blockchain Network.
8. On successfully mining the block, it rewards the miner with some amount of Ether in their wallet.
9. Then the requested transaction is approved and credited to the respective wallet/wallets.

Each transaction is mined only once, but every participant of the Ethereum Network verifies it.

4. **What is Solidity?**

Solidity is a brand-new programming language created by the **Ethereum** which is the second-largest market of cryptocurrency by capitalization, released in the year 2015 led by Christian Reitwiessner. Some key features of solidity are listed below:

- Solidity is a high-level programming language designed for implementing smart contracts.
- It is statically-typed object-oriented(contract-oriented) language.
- Solidity is highly influenced by Python, c++, and JavaScript which runs on the Ethereum Virtual Machine(EVM).
- Solidity supports complex user-defined programming, libraries and inheritance.
- Solidity is primary language for blockchains running platforms.
- Solidity can be used to creating contracts like voting, blind auctions, crowdfunding, multi-signature wallets, etc.

5. **Explain types of accounts in Ethereum.**

An Ethereum account is similar to a bank account, but for ethers or ETH, where Ethereum can be held, transferred to other accounts, and can also be used to execute smart contracts. An Ethereum account is an entity that is composed of an Ethereum address along with a private key. The first 20 bytes of the SHA3 hashed public key is the Ethereum address.

Types of Ethereum Accounts

Below are the two types of Ethereum Accounts:

1. Externally Owned Account: This is the most basic type of Ethereum account, it functions similarly to a bitcoin account. A private key controls the Ethereum address for EOAs. A person can open as many EOAs as they require. It is created whenever a wallet is created, and it is made with a private key that is required to access EOAs, check balances, send and receive transactions, and establish smart contracts.

Advantages:

1. Transactions from an external account to a contract account can trigger code that can execute many different actions, such as transferring tokens or even creating a new contract.
2. Externally Owned Accounts cannot list incoming transactions.

2. Contract-Based Account: Contract-based accounts can perform all of the functions of an externally owned account, but unlike EOAs, they are formed when a contract code is deployed, are governed by contract codes, and are accessed using a unique address. When one party accepts a contract, a unique account is formed which contains all of the charges associated with that contract. Each contract is granted a distinct serial number, which is referred to as a contract account.

Advantages:

1. A contract account can list incoming transactions.
2. Contract accounts can be set up as Multisig Accounts.
3. A Multisig Account can be structured such that it has a daily limit that you specify, and only if the daily limit is exceeded will multiple signatures be required.

Disadvantages:

1. Creating contract accounts costs gas because they use the valuable computational and storage resource of the network.
2. Contract accounts can't initiate new transactions on their own. Instead, contract accounts can only fire transactions in response to

other transactions they have received either from an externally owned account or from another contract account.

Types of Contract Accounts

Below are the three types of contract accounts:

1. **Simple Account:** The account is created and owned by a single account holder.
2. **Multisig (multisignature) Account:** A Multisig Wallet contains several owner Accounts, one of which is also the creator Account.
3. **Simplest Account:** A Multisig Wallet contains several owner Accounts, one of which is also the creator Account.

6. **Explain the term Gas, Block, Transactions, Ether with respect to Ethereum.**

Gas is the fee required to successfully conduct a transaction or execute a contract on the Ethereum blockchain platform. Fees are priced in tiny fractions of the cryptocurrency ether (ETH)—denominations called gwei (10^{-9} ETH). Gas is used to pay validators for the resources needed to conduct transactions.1

The exact price of the gas is determined by supply, demand, and network capacity at the time of the transaction.

Blocks are batches of transactions with a hash of the previous block in the chain. This links blocks together (in a chain) because hashes are cryptographically derived from the block data. This prevents fraud, because one change in any block in history would invalidate all the following blocks as all subsequent hashes would change and everyone running the blockchain would notice.

An Ethereum transaction refers to an action initiated by an externally-owned account, in other words an account managed by a human, not a contract. For example, if Bob sends Alice 1 ETH, Bob's account must be debited and Alice's must be credited. This state-changing action takes place within a transaction.

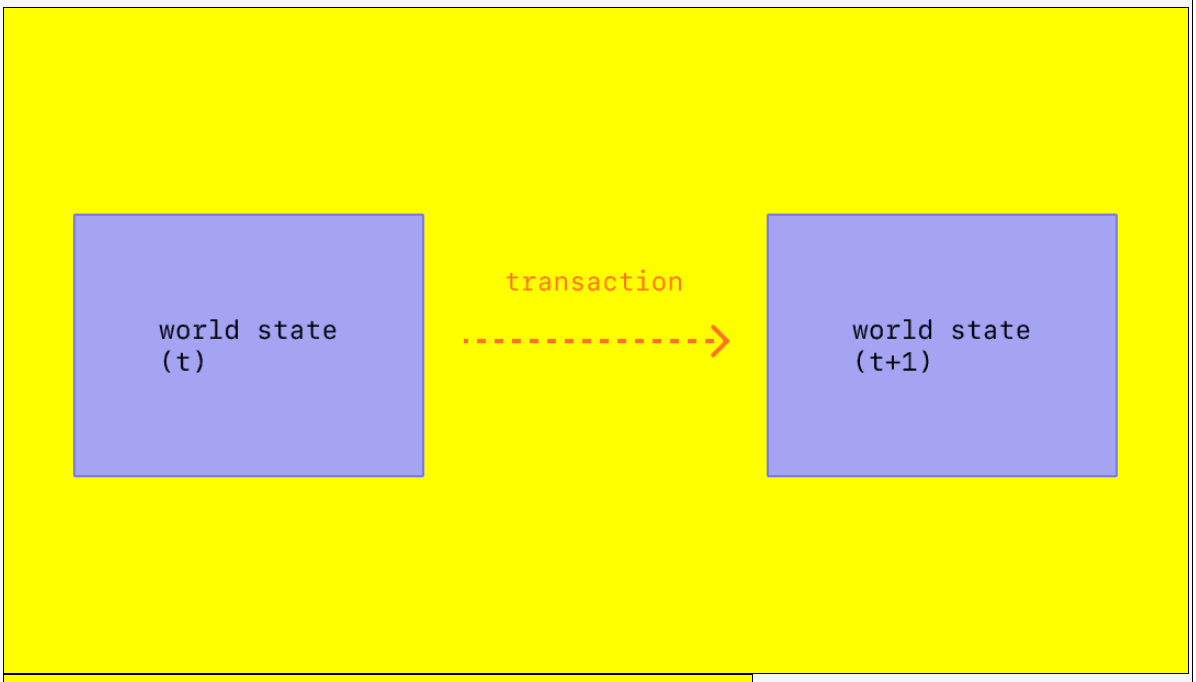


Diagram adapted from Ethereum EVM illustrated

Transactions, which change the state of the EVM, need to be broadcast to the whole network. Any node can broadcast a request for a transaction to be executed on the EVM; after this happens, a validator will execute the transaction and propagate the resulting state change to the rest of the network.

Transactions require a fee and must be included in a validated block. To make this overview simpler we'll cover gas fees and validation elsewhere.

A submitted transaction includes the following information:

- recipient – the receiving address (if an externally-owned account, the transaction will transfer value. If a contract account, the transaction will execute the contract code)
- signature – the identifier of the sender. This is generated when the sender's private key signs the transaction and confirms the sender has authorized this transaction
- nonce - a sequencially incrementing counter which indicates the transaction number from the account
- value – amount of ETH to transfer from sender to recipient (in WEI, a denomination of ETH)
- data – optional field to include arbitrary data
- gasLimit – the maximum amount of gas units that can be consumed by the transaction. Units of gas represent computational steps
- maxPriorityFeePerGas - the maximum amount of gas to be included as a tip to the validator
- maxFeePerGas - the maximum amount of gas willing to be paid for the transaction (inclusive of baseFeePerGas and maxPriorityFeePerGas)

Ether (ETH) is the main token of the Ethereum blockchain and the world's second-largest cryptocurrency by market capitalization. Just like the largest cryptocurrency, bitcoin, ether can be used to send payments directly to another person without the need for an intermediary such as a bank.

The long-term vision for Ethereum is to power more than just financial transactions. Software developers are able to build applications on Ethereum, ranging from decentralized platforms for lending money to social media networks.

For any Ethereum-based app, ether acts as the primary "fuel." Any activity on the blockchain requires an amount of ether to power it, also known as "gas."

In Ethereum, ether can be used for the following things:

- **Payments:** Like bitcoin, ether can be used for payments. Users can send ether to another user and, just like cash, the payment doesn't require a third party to process or approve it.

- **Powering decentralized applications**: Ether is required in order to use decentralized apps (dapps) built on Ethereum, from staking ERC-20 tokens for yield farming to completing functions such as governance voting.

- **Transactions fees:** Every Ethereum action – from payments to using dapps – requires a fee.

7. **Explain Hyperledger Fabric in detail.**

Hyperledger Fabric is an open source, permissioned blockchain framework, started in 2015 by The Linux Foundation. It is a modular, general-purpose framework that offers unique identity management and access control features, which make it suitable for a variety of industry applications such as track-and-trace of supply chains, trade finance, loyalty and rewards, as well as clearing and settlement of financial assets. A Hyperledger Fabric network is comprised of unique organizations (or members) that interact with each other on the network. For example, an organization could be a bank in a network comprised of financial institutions or a shipping partner in a supply chain network. From a Fabric component perspective, each organization has a Fabric certificate authority and one or more peer nodes. A Fabric network also has an ordering service shared by all organizations in the network, and this component helps process transactions for the network. We will share more details about each of these concepts and components below:

An organization in a network is defined by a root certificate specific to that organization. Users and other components (like peer nodes – see below) in that organization are also identified by certificates, and these certificates are derived from this root certificate, ensuring other organizations in the network can relate a user to their organization. These certificates also specify the permissions for each entity on the network, like read-only versus full access on a channel.

A root certificate for an organization is stored in the Fabric certificate authority (CA). The Fabric CA also issues certificates for users in an organization and handles other related operations. An enterprise-grade Fabric CA utilizes a variety of components and can deployed in a variety of ways using a Hardware Security Module (HSM) for root certificate protection.

An organization also creates one or more peer nodes as components to carry out operations on behalf of that organization. Specifically, a peer node endorses transactions proposed on the network, stores and executes smart contract code (known as chaincode in Fabric), and stores a local copy of the ledger for access. Fabric clients typically interact with peer nodes to read the ledger, add new chaincode to the network, or propose a new transaction. A peer node typically runs on its own computer, like an Amazon EC2 instance.

Finally, a Fabric network also includes of an ordering service shared by all members of the network. The ordering service makes sure new transactions on the network are properly ordered in new blocks and have the proper

endorsements. The ordering service then broadcasts a new block of transactions to peer nodes in each organization. Peer nodes update their local copy of the ledger with this new block.

8. **Explain architecture of Hyperledger Fabric.**

The validating peers run a BFT consensus protocol for executing a replicated state machine that accepts three types of transactions as operations: Deploy transaction: Takes a chaincode (representing a smart contract) written in Go as a parameter; the chaincode is installed on the peers and ready to be invoked. Invoke transaction: Invokes a transaction of a particular chaincode that has been installed earlier through a deploy transaction; the arguments are specific to the type of transaction; the chaincode executes the transaction, may read and write entries in its state accordingly, and indicates whether it succeeded or failed. Query transaction: Returns an entry of the state directly from reading the peer's persistent state; this may not ensure linearizability. Each chaincode may define its own persistent entries in the state. The blockchain's hash chain is computed over the executed transactions and the resulting persistent state. Validation of transactions occurs through the replicated execution of the chaincode and given the fault assumption underlying BFT consensus, i.e., that among the n validating peers at most $f < n/3$ may "lie" and behave arbitrarily, but all others execute the chaincode correctly. When executed on top of PBFT consensus, it is important that chaincode transactions are deterministic, otherwise the state of the peers might diverge. A modular solution to filter out non-deterministic transactions that are demonstrably diverging is available and has been implemented in the SIEVE protocol [3]. Membership among the validating nodes running BFT consensus is currently static and the setup requires manual intervention. Support for dynamically changing the set of nodes running consensus is planned for a future version. As the fabric implements a permissioned ledger, it contains a security infrastructure for authentication and authorization. It supports enrollment and transaction authorization through public-key certificates, and confidentiality for chaincode realized through in-band encryption. More precisely, for connecting to the network every peer needs to obtain an enrollment certificate from an enrollment CA that is part of the membership services. It authorizes a peer to connect to the network and to acquire transaction certificates, which are needed to submit transactions. Transaction certificates are issued by a transaction CA and support pseudonymous authorization for the peers submitting transactions, in the sense that multiple transaction certificates issued to the

same peer (that is, to the same enrollment certificate) cannot be linked with each other. Confidentiality for chaincodes and state is provided through symmetric-key encryption of transactions and states with a blockchain-specific key that is available to all peers with an enrollment certificate for the blockchain. Extending the encryption mechanisms towards more fine-grained confidentiality for transactions and state entries is planned for a future version.

9. **Explain Transaction flow in Hyperledger Fabric.**

This flow assumes that a channel is set up and running. The application user has registered and enrolled with the organization's Certificate Authority (CA) and received back necessary cryptographic material, which is used to authenticate to the network.

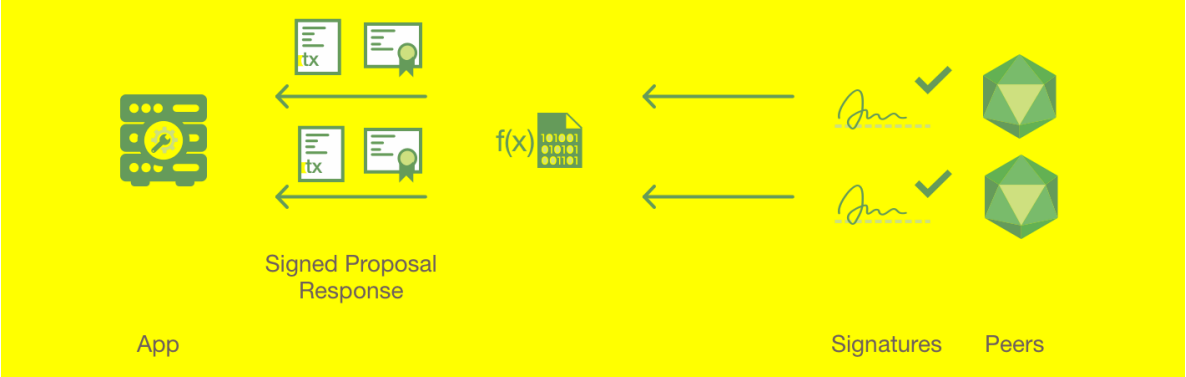
The chaincode (containing a set of key value pairs representing the initial state of the radish market) is installed on the peers and deployed to the channel. The chaincode contains logic defining a set of transaction instructions and the agreed upon price for a radish. An endorsement policy has also been set for this chaincode, stating that both `peerA` and `peerB` must endorse any transaction.

1. **Client A initiates a transaction**

What's happening? Client A is sending a request to purchase radishes. This request targets `peerA` and `peerB`, who are respectively representative of Client A and Client B. The endorsement policy states that both peers must endorse any transaction, therefore the request goes to `peerA` and `peerB`.

Next, the transaction proposal is constructed. An application leveraging a supported SDK (Node, Java, Go) utilizes one of the available API's to generate a transaction proposal. The proposal is a request to invoke a chaincode function with certain input parameters, with the intent of reading and/or updating the ledger.

The SDK serves as a shim to package the transaction proposal into the properly architected format (protocol buffer over gRPC) and takes the user's cryptographic credentials to produce a unique signature for this transaction proposal. The SDK submits the transaction proposal to a target peer, which will manage the transaction submission on behalf of the client. The target peer first forwards the transaction proposal to other peers for execution, as required by the endorsement policy.

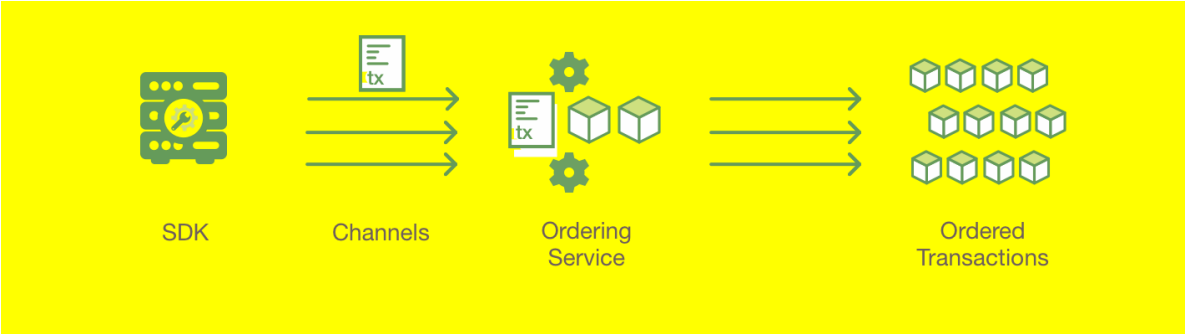


2. Endorsing peers verify signature & execute the transaction

The endorsing peers verify (1) that the transaction proposal is well formed, (2) it has not been submitted already in the past (replay-attack protection), (3) the signature is valid (using the MSP), and (4) that the submitter (Client A, in the example) is properly authorized to perform the proposed operation on that channel (namely, each endorsing peer ensures that the submitter satisfies the channel's *Writers* policy). The endorsing peers take the transaction proposal inputs as arguments to the invoked chaincode's function. The chaincode is then executed against the current state database to produce transaction results including a response value, read set, and write set (i.e. key/value pairs representing an asset to create or update). No updates are made to the ledger at this point. The set of these values, along with the endorsing peer's signature is passed back as a "proposal response" to the target peer.

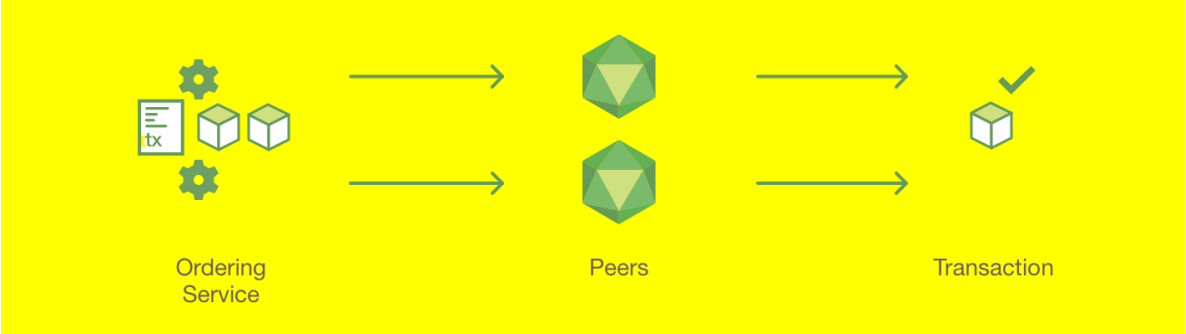
3. Proposal responses are inspected

The target peer verifies the proposal responses are the same prior to proceeding with the transaction submission. The architecture is such that even if a transaction is submitted without this check, the endorsement policy will still be checked and enforced when each peer validates transactions prior to committing them.



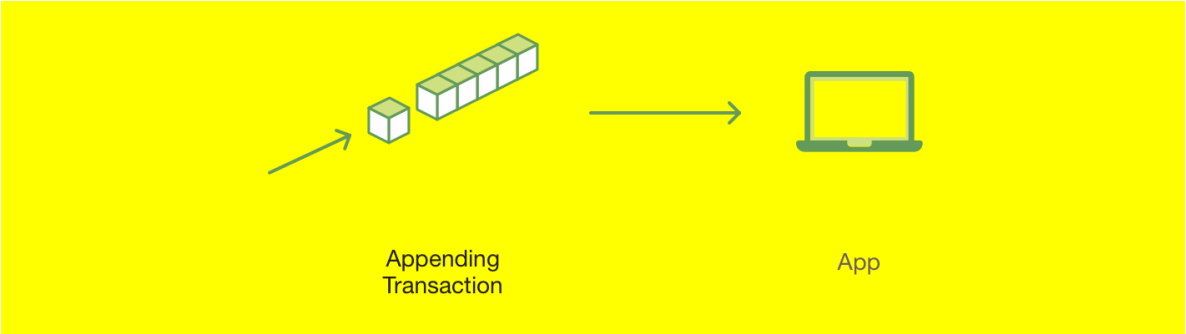
4. Target peer assembles endorsements into a transaction

The target peer “broadcasts” the transaction proposal and response within a “transaction message” to the ordering service. The transaction contains the Channel ID, the read/write sets, and a signature from each endorsing peer. The ordering service does not need to inspect the entire content of a transaction in order to perform its operation, it simply receives transactions, orders them, and creates blocks of transactions per channel.



5. Transaction is validated and committed

The blocks of transactions are “delivered” to all peers on the channel. The transactions within the block are validated to ensure endorsement policy is fulfilled and to ensure that there have been no changes to ledger state for read set variables since the read set was generated by the transaction execution. Transactions in the block are tagged as being valid or invalid.



6. Ledger updated

Each peer appends the block to the channel’s chain, and for each valid transaction the write sets are committed to current state database. An event is emitted by each peer to notify the client application that the transaction (invocation) has been immutably appended to the chain, as well as notification of whether the transaction was validated or invalidated.

10. **Explain Membership and Identity Management in Hyperledger Fabric.**

Hyperledger Fabric is an implementation of distributed ledger technology (DLT) that delivers enterprise-ready network security, scalability, confidentiality and performance, in a modular blockchain architecture. Hyperledger Fabric delivers the following blockchain network functionalities:

Identity management

To enable permissioned networks, Hyperledger Fabric provides a membership identity service that manages user IDs and authenticates all participants on the network. Access control lists can be used to provide additional layers of permission through authorization of specific network operations. For example, a specific user ID could be permitted to invoke a chaincode application, but blocked from deploying new chaincode.

Membership Service Provider (MSP) comes into play — **it identifies which Root CAs and Intermediate CAs are trusted to define the members of a trust domain, e.g., an organization**, either by listing the identities of their members, or by identifying which CAs are authorized to issue valid identities for their members, or — as will usually be the case — through a combination of both.

The power of an MSP goes beyond simply listing who is a network participant or member of a channel. An MSP can identify specific **roles** an actor might play either within the scope of the organization the MSP represents (e.g., admins, or as members of a sub-organization group), and sets the basis for defining **access privileges** in the context of a network and channel (e.g., channel admins, readers, writers).

The configuration of an MSP is advertised to all the channels where members of the corresponding organization participate (in the form of a **channel MSP**). In addition to the channel MSP, peers, orderers, and clients also maintain a **local MSP** to authenticate member messages outside the context of a channel and to define the permissions over a particular

component (who has the ability to install chaincode on a peer, for example).

In addition, an MSP can allow for the identification of a list of identities that have been revoked — as discussed in the Identity documentation — but we will talk about how that process also extends to an MSP.

11. **Explain Gossip Protocol. How it is used in Hyperledger Fabric.**

Gossip protocol is a communication protocol that allows state sharing in distributed systems. Most modern systems use this peer-to-peer protocol to disseminate information to all the members in a network or cluster.

This protocol is used in a decentralized system that does not have any central node to keep track of all nodes and know if a node is down or not.

Hyperledger Fabric optimizes blockchain network performance, security and scalability by dividing workload across transaction execution (endorsing and committing) peers and transaction ordering nodes. This decoupling of network operations requires a secure, reliable and scalable data dissemination protocol to ensure data integrity and consistency. To meet these requirements, Hyperledger Fabric implements a **gossip data dissemination protocol**. The gossip-based data dissemination protocol performs three primary functions on a Hyperledger Fabric network:

1. Manages peer discovery and channel membership, by continually identifying available member peers, and eventually detecting peers that have gone offline.
2. Disseminates ledger data across all peers on a channel. Any peer with data that is out of sync with the rest of the channel identifies the missing blocks and syncs itself by copying the correct data.
3. Bring newly connected peers up to speed by allowing peer-to-peer state transfer update of ledger data.

Gossip-based broadcasting operates by peers receiving messages from other peers on the channel, and then forwarding these messages to a number of randomly-selected peers on the channel, where this number is a configurable constant. Peers can also exercise a pull mechanism, rather than waiting for delivery of a message. This cycle repeats, with the result of channel membership, ledger and state information continually being kept current and in sync. For dissemination of new blocks, the **leader** peer on the channel pulls the data from the ordering service and initiates gossip dissemination to peers. Leader election

The leader election mechanism is used to **elect** one peer per each organization which will maintain connection with ordering service and initiate distribution of newly arrived blocks across peers of its own organization. Leveraging leader election provides system with ability to efficiently utilize bandwidth of the ordering service. There are two possible operation modes for leader election module:

1. **Static** - system administrator manually configures one peer in the organization to be the leader, e.g. one to maintain open connection with the ordering service.
2. **Dynamic** - peers execute a leader election procedure to select one peer in an organization to become leader, pull blocks from the ordering service, and disseminate blocks to the other peers in the organization..

Gossip messaging

Online peers indicate their availability by continually broadcasting “alive” messages, with each containing the **public key infrastructure (PKI)** ID and the signature of the sender over the message. Peers maintain channel membership by collecting these alive messages; if no peer receives an alive message from a specific peer, this “dead” peer is eventually purged from channel membership. Because “alive” messages are cryptographically signed, malicious peers can never impersonate other peers, as they lack a signing key authorized by a root certificate authority (CA).

6. Module: Case Study

1. **Explain how Blockchain is used in Government.**
   Governments and public sector organizations leverage blockchain technology to move away from siloed and inefficient centralized systems. Current systems are inherently insecure and costly, while blockchain networks offer more secure, agile, and cost-effective structures.

A blockchain-based government has the potential to solve legacy pain points and enable the following advantages:

- Secure storage of government, citizen, and business data
- Reduction of labor-intensive processes
- Reduction of excessive costs associated with managing accountability
- Reduced potential for corruption and abuse
- Increased trust in government and online civil systems

The distributed ledger format can be leveraged to support an array of government and public sector applications, including digital currency/payments, land registration, identity management, supply chain traceability, health care, corporate registration, taxation, voting (elections and proxy), and legal entities management.

What are the Blockchain Use Cases in Government and the Public Sector?

- Smart Cities
- Central Banking
- Validation of Education and Professional Qualifications
- Tracking Vaccinations
- Tracking Loans and Student Grants
- Payroll Tax Collection

How will blockchain impact smart cities?

A smart city uses information technology and data to integrate and manage physical, social, and business infrastructures to streamline services to its

inhabitants while ensuring efficient and optimal utilization of available resources. In combination with technologies, IoT, cloud computing, and blockchain technology, governments can deliver innovative services and solutions to the citizens and local municipalities.

Blockchain can provide the mechanism to create a secure infrastructure to manage these functions. Specifically, it can provide a secure interoperable infrastructure that allows all smart city services and functions to operate beyond currently envisioned levels. ConsenSys has worked toward realizing smart city initiatives in Dubai and Zug.

How will blockchain impact central banking?

Real-time gross settlement is the continuous process of settling interbank payments in central bank records as opposed to settlement at the end of each day. Blockchain enables a significant increase in transaction volume and network resilience which enables central banks to process RTGS at a faster pace, with heightened security.

How will blockchain streamline the validation of educational and professional qualifications?

Keeping academic and professional attainment data on an encrypted identity wallet empowers individuals to control access to their data. It also enables schools, universities, and employers to validate “attestations” for courses and work achieved.

How will blockchain technology impact vaccination tracking?

Recording vaccination data on the blockchain enables schools, insurance, and medical providers to validate vaccinations quickly. This process automatically triggers corresponding micropayments and delegates access to benefits based on medical status.

How will blockchain technology manage tracking student loans and grants?

Smart contracts can be programmed to manage loan and grant applications, dispense loans, and track compliance with the terms and conditions. This automated performance tracking enables real-time data and increased transparency, compliance, and security.

How will blockchain technology impact the collection of payroll tax?

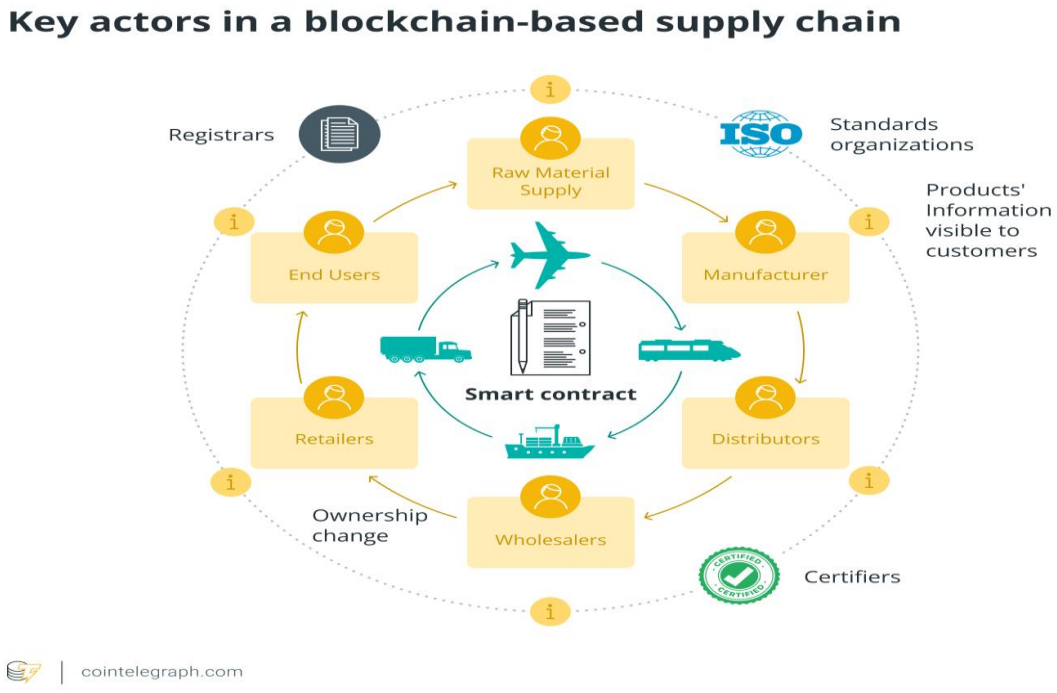
Smart contracts can streamline the tax collection process by matching tax data with income transactions and calculating tax and social security deductions. A blockchain-based system automatically transfers net salary and tax payments to their respective recipients. Coordinated automation brings efficiency, speed, and security to tax collection.

2. **Explain how Blockchain is used in Supply Chain Management.**

Blockchain-based supply chain networks might need a closed, private and permissioned blockchain with limited actors, in contrast to Bitcoin and other financial blockchain applications, which may be public. However, the possibility of a more open set of partnerships may still exist.

In blockchain-based supply networks, four key actors play roles, including registrars, standard organizations, certifiers, and actors:

- Registrars: They provide network actors with distinct identities.
- Standard organizations: These organizations develop blockchain rules and technical specifications or standards schemes, such as Fairtrade, for environmentally friendly supply chains.
- Certifiers: They certify individuals for involvement in supply chain networks.
- Actors: A registered auditor or certifier must certify participants or actors, such as producers, sellers and buyers, to retain the system's credibility.



How a product is "owned" or transferred by a specific actor is an intriguing feature of structure and flow management and among the benefits of blockchain in supply chain management. But does blockchain make supply chain management more transparent?

As the concerned parties are required to fulfill a smart contract condition before a product is transferred (or sold) to another actor to validate the exchange of goods or services, and the blockchain ledger is updated with transaction information after all participants have complied with their duties and processes, overall transparency across the value chain is improved.

Additionally, the nature, quantity, quality, location and ownership product dimensions are transparently specified by blockchain technology. As a result, customers can view the continuous chain of custody and transactions from the raw materials to the final sale, eliminating the requirement for a reliable central organization to administer and maintain digital supply chains.

3. **Explain how Blockchain is used in Financial Services.**

The Ethereum blockchain enables more open, inclusive, and secure business networks, shared operating models, more efficient processes, reduced costs, and new products and services in banking and finance. It enables digital securities to be issued within shorter periods of time, at lower unit costs, with greater levels of customization. Digital financial instruments may thus be tailored to investor demands, expanding the market for investors, decreasing costs for issuers, and reducing counterparty risk.

Over the last five years, the technology has matured for enterprise-grade use demonstrating the following benefits:

- **Security**: Its distributed consensus based architecture eliminates single points of failure and reduces the need for data intermediaries such as transfer agents, messaging system operators and inefficient monopolistic utilities. Ethereum also enables implementation of secure application code designed to be tamper-proof against fraud and malicious third parties— making it virtually impossible to hack or manipulate.
- **Transparency**: It employs mutualized standards, protocols, and shared processes, acting as a single shared source of truth for network participants
- **Trust**: Its transparent and immutable ledger makes it easy for different parties in a business network to collaborate, manage data, and reach agreements
- **Programmability**: It supports the creation and execution of smart contracts— tamper proof, deterministic software that automates business logic – creating increased trust and efficiency
- **Privacy**: It provides market-leading tools for granular data privacy across every layer of the software stack, allowing selective sharing of data in business networks. This dramatically improves transparency, trust and efficiency while maintaining privacy and confidentiality.
- **High-Performance**: It's private and hybrid networks are engineered to sustain hundreds of transactions per second and periodic surges in network activity
- **Scalability**: It supports interoperability between private and public chains, offering each enterprise solution the global reach, tremendous resilience, and high integrity of the mainnet

According to a report by Jupiter Research, blockchain deployments will enable banks to realize savings on cross-border settlement transactions of up to $27 billion by the end of 2030, reducing costs by more than 11%. Ethereum specifically has already demonstrated disruptive economics, creating over 10x cost advantages against incumbent technologies. Financial institutions acknowledge that distributed ledger technology will save billions of dollars for banks and major financial institutions over the next decade.

How does blockchain impact capital markets?

Capital markets refers to the pairing of issuers with demand for capital, to investors with corresponding risk and return profiles. Whether issuers be entrepreneurs, startups or large organizations, the process of raising capital can be challenging. Firms face increasingly stringent regulations, longer times to get to market, volatility from interest rates and liquidity risk. Particularly in emerging markets, they must navigate the lack of rigorous monitoring, thorough regulation and sufficient market infrastructure for issuing, settlement, clearing, and trading. Blockchain offers multiple benefits for several capital market use cases:

- Elimination of a single point of failure through decentralized utilities
- Facilitation of capital market activities streamlining processes, reducing costs and decreasing settlement times
- Digitization of processes and workflows, reducing operational risks of fraud, human error, and overall counterparty risk
- Digitization or tokenization of assets and financial instruments, making them programmable and much easier to manage and trade. In token form, they gain wider market access through increased connectivity and the possibility of fractionalized ownership. This results in increased liquidity and decreased cost of capital.

How does blockchain impact asset management?

Venture capital firms, private equity firms, real estate funds, and specialty markets are facing demands to improve liability risk management, adapt more dynamic decision-making structures, and address the increasing complexity of ever-changing regulations. Blockchain can effectively streamline asset and stakeholder management. It allows:

- Automated fund launch

- Seamless stakeholder engagement with digitized assets and services
- Digitization of portfolio and existing holdings for wider market access, liquidity and fractionalization
- Customizable built-in privacy settings for transaction confidentiality
- Voting and other shareholder rights and obligations programmed into digital assets, resulting in seamless user experience and reduced risks of human error
- Creation and enforcement of incentive mechanisms to promote participation and punish nefarious activity
- Improved governance and transparency for investors and stakeholders
- Efficient cap table management
- Automated fund administration
- Automated transfer agency in asset management

How does blockchain impact global payments and remittances?

Global payments and remittances today are executed by a number of intermediaries that exact tolls for their service. It takes 2 to 7 days and costs a global average of 6.94% to send $200 between countries. This means that remittances are directly reduced by $48B through fees, intermediaries, and financial institutions. Blockchain can streamline payment and remittance processes, reducing settlement times and significantly reducing costs. It allows:

- Rapid and secure domestic retail payments
- Rapid and secure domestic wholesale and securities settlement
- Rapid and secure cross border payments
- Real-time gross settlement between central banks, commercial banks, and independent banks
- Digitized KYC/AML data and transaction history, reducing risks of fraud and enabling real-time authentication
- Automated regulatory oversight and auditing
- Multiple forms of payment enabled on blockchain: Tokenized fiat, stablecoin, and cryptocurrency

How does blockchain impact banking and lending?

Core banking comprises of transaction, loan, mortgage, and payment services. Many of these services depend on legacy processes of execution. For example, between information verification, credit scoring, loan

processing and distribution of funds— it takes 30 to 60 days for individuals to secure a mortgage, and 60 to 90 days for small or medium enterprises to secure a business loan. Blockchain can streamline banking and lending services, reducing counterparty risk, and decreasing issuance and settlement times. It allows:

- Authenticated documentation and KYC/AML data, reducing operational risks and enabling real-time verification of financial documents
- Streamlined credit prediction and credit scoring markets, instantaneously informed by the collation of user activity and sanctioned data across a network
- Automated syndicate formation, underwriting, and disbursement of funds i.e. principal and interest payments, reducing cost, delay and friction of syndication
- Facilitated collateralization of assets because digitization enables real-time asset management, tracking, and enforcement of regulatory controls

How does blockchain impact trade finance?

Trade finance refers to the infrastructure, processes and funding that support international trade supply chains. The industry continues to rely on paper-based processes that are susceptible to security vulnerabilities. Individual transactions can take as long as 90-120 days in order to process letters of credit, verify documents, and establish trust among stakeholders. Blockchain can digitize the entire trade finance lifecycle with increased security and efficiency. It can enable more transparent governance, decreased processing times, lower capital requirements and reduced risks of fraud, human error, and overall counterparty risk. It allows:

- Digitized and authenticated documentation (i.e. letters of credit and bill of lading) and KYC/AML data with real-time verification of financial documents
- Asset digitization to enable faster settlement times
- Creation of more efficient financing structures through shared secure networks and digitized processes
- Creation of a consistent financing vehicle for the entire trade lifecycle, eliminating the legacy practice of negotiating independent finance vehicles for each stage of the trade

How does blockchain impact insurance?

Property and casualty insurance claims are prone to fraud and claim assessments can extend long periods of time. Blockchain can securely streamline data verification, claims processing, and disbursement, reducing processing time significantly. It allows:

- Authenticated documentation and KYC/AML data, reducing the risk of fraud and facilitating claim assessments
- Automated claims processing with the use of smart contracts
- Automated parameterized contracts to pay out upon occurrence of certain risk
- Automated disbursement of insurance payments
- Tokenized reinsurance markets to facilitate policy reinsurance in open marketplaces, stepping away from traditional broker and relationship-based systems

How does blockchain facilitate compliance?

Regulatory compliance has become increasingly important in the commerce and finance space. It is necessary in order to ensure that financial institutions respect laws, rules, and regulations applicable to their activities. It is a huge challenge for firms to keep up with the pace and complexity of regulatory change— particularly when firms operate across borders and are thus exposed to multiple regulatory regimes. Blockchain offers these benefits:

- Unique governance and compliance attributes programmed into digital assets
- Streamlined processes that automate data verification and reporting, facilitate regulatory oversight, reduce operational friction, and eliminate errors associated with manual auditing and other activities— all in real-time
- Creation and enforcement of incentive structures to improve network governance