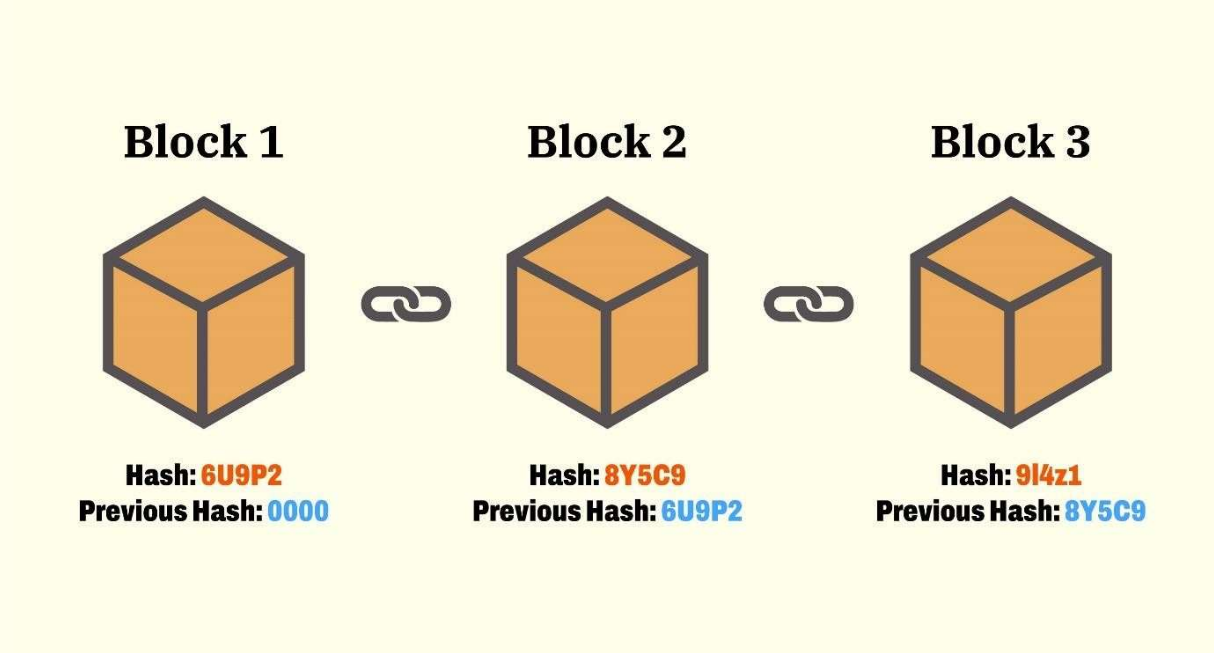
Blockchain and DLT Notes

Bachelor of Engineering (University of Mumbai)

Blockchain and DLT

Chapter 1: Introduction to DLT and Blockchain

Define Blockchain



Blockchain is a decentralized digital ledger technology that enables secure, transparent and tamper-proof recording of transactions and data in a way that is resistant to modification and hacking.

In a blockchain, transactions are recorded in a chronological and permanent way, forming blocks that are linked together in a chain. Each block contains a cryptographic hash of the previous block, ensuring the integrity and immutability of the chain.

This distributed ledger technology allows for the creation of decentralized networks that can be used for a wide range of applications, such as cryptocurrency transactions, supply chain management, voting systems, and more.

Blockchains are typically managed by a network of nodes that work together to validate transactions and maintain the integrity of the system. This decentralized approach to record-keeping makes it difficult for any single party to manipulate the data, increasing trust and security in the system.

Elements of Blockchain

Nodes:

- Nodes are essentially the participants in the blockchain network.
- They can be individual users, organizations, or even computers or devices that are connected to the network.
- Each node maintains a copy of the blockchain ledger and works together with other nodes to validate transactions and add new blocks to the chain.
- Nodes communicate with each other using a peer-to-peer (P2P) network protocol, and they typically use consensus mechanisms to ensure that the data in the blockchain is accurate and secure.

Nonce:

- A nonce is a random number that is added to a block during the mining process.
- It is used in combination with the other data in the block to create a hash that meets certain criteria.
- The hash serves as proof of work, and once a valid hash is found, the block can be added to the blockchain.
- The nonce is a key element in the process of mining cryptocurrency, as it helps to ensure the security and integrity of the blockchain.

Miner:

- Miners are individuals or organizations that use specialized hardware and software to compete in the process of mining cryptocurrency.
- They are essentially trying to solve the cryptographic puzzle and find the correct nonce to add a new block to the blockchain.
- Mining is an important part of the blockchain ecosystem, as it helps to validate transactions and ensure the security of the network.
- In many cases, miners are rewarded with newly minted cryptocurrency as an incentive for their work.

Distributed Ledger:

- A distributed ledger is a type of database that is spread across multiple nodes in a network. In the case of blockchain, the ledger is maintained by a decentralized network of nodes, which ensures the security and immutability of the data.
- Because the ledger is distributed across many nodes, it is extremely difficult for any one node to manipulate or change the data.
- This makes distributed ledgers ideal for use in situations where transparency and security are essential, such as in financial transactions or supply chain management.

Network Consensus:

- Network consensus refers to the process by which the nodes in a blockchain network work together to validate transactions and add new blocks to the chain.
- Consensus mechanisms can vary depending on the blockchain implementation, but typically involve a majority of nodes agreeing on the validity of a transaction before it is added to the blockchain.
- This helps to ensure the security and accuracy of the blockchain, and prevents any one node from making fraudulent or erroneous transactions.

Wallet:

- A wallet is a digital tool that allows users to store and manage their cryptocurrency assets.
- It typically includes a public key for receiving transactions and a private key for authorizing transactions.
- Wallets can be either software-based or hardware-based, and can provide varying levels of security and accessibility.
- Some wallets are designed for use with specific cryptocurrencies, while others can support multiple types of digital assets.
- In general, it's important to choose a secure and reliable wallet that meets your needs and preferences.

Here are some of the **key advantages** of blockchain:

Decentralization: One of the most significant advantages of blockchain is its decentralized nature. Because the data is stored across a distributed network of nodes, there is no single point of failure or vulnerability. This makes blockchain more secure and less susceptible to hacking or fraud.

Transparency: Transactions on a blockchain are visible to all participants, which provides a high degree of transparency and accountability.

Immutability: Once data is added to a blockchain, it cannot be altered or deleted. This means that blockchain provides a high degree of immutability and permanence, which is valuable for record-keeping and audit purposes.

Efficiency: Blockchain transactions can be processed more quickly and efficiently than traditional financial transactions. This is because there are no intermediaries involved, and the transactions are validated by a distributed network of nodes.

Security: Blockchain transactions are secured through advanced cryptographic algorithms, which make them virtually impossible to hack or manipulate. This provides a high degree of security and protects against fraud and theft.

Features of Blockchain

Blockchain technology is characterized by several key features that make it unique and powerful. Here are some of the main features of blockchain:

Decentralization:

- The data in a blockchain is stored across a distributed network of nodes, rather than in a central location.
- This provides a high degree of decentralization and eliminates the need for intermediaries.

Immutability:

- Once data is added to a blockchain, it cannot be altered or deleted.
- This provides a high degree of immutability and permanence, which is valuable for record-keeping and audit purposes.

Transparency:

- Transactions on a blockchain are visible to all participants, which provides a high degree of transparency and accountability.
- This makes it more difficult for bad actors to engage in fraudulent or illegal activity, and can help to build trust between participants.

Security:

- Blockchain transactions are secured through advanced cryptographic algorithms, which make them virtually impossible to hack or manipulate.
- This provides a high degree of security and protects against fraud and theft.

Consensus:

- In order to add new transactions to a blockchain, a majority of nodes in the network must agree that the transaction is valid.
- This consensus mechanism helps to ensure the accuracy and integrity of the blockchain.

Efficiency:

- Blockchain transactions can be processed more quickly and efficiently than traditional financial transactions.
- This is because there are no intermediaries involved, and the transactions are validated by a distributed network of nodes.

Smart contracts:

- Smart contracts are self-executing contracts that are built on top of a blockchain.
- They can be used to automate complex business processes and transactions, and can help to reduce costs and increase efficiency.

Types of Blockchain

There are generally three types of blockchain: public, private, and consortium.

Public Blockchain: A public blockchain is a decentralized blockchain that anyone can join and participate in. It is open to the public and anyone can access the network, view the transactions, and participate in the consensus process. Public blockchains are typically used for cryptocurrencies like Bitcoin and Ethereum, and they are designed to be fully decentralized and trust less.

Some of the key features of public blockchains include:

- No central authority or controlling entity
- Anyone can participate in the network
- Transactions are transparent and publicly visible
- Consensus is achieved through a proof-of-work or proof-of-stake algorithm
- Typically used for cryptocurrencies and other decentralized applications

Private Blockchain: A private blockchain is a blockchain that is restricted to a specific group of participants. It is typically used within an organization or consortium of organizations, and it is not open to the public. Private blockchains are designed to be more centralized than public blockchains, and they offer more control and privacy to their participants.

- Some of the key features of private blockchains include:
- Controlled access to the network
- Transactions are visible only to authorized participants
- Consensus is achieved through a pre-determined group of validators
- Typically used for enterprise applications, such as supply chain management or identity verification

Consortium Blockchain: A consortium blockchain is a hybrid between a public and a private blockchain. It is a blockchain that is controlled by a pre-determined group of organizations, and it is typically used for collaborative projects that require a high degree of trust between participants. Consortium blockchains offer some of the benefits of both public and private blockchains, and they are designed to be more scalable and efficient than public blockchains.

Some of the key features of consortium blockchains include:

- Controlled access to the network by a pre-determined group of organizations
- Transactions are visible only to authorized participants
- Consensus is achieved through a pre-determined group of validators
- Typically used for collaborative projects between organizations, such as supply chain management or interbank transactions.

What is DLT

DLT, or Distributed Ledger Technology, is a type of technology that allows for the creation, storage, and sharing of digital records across a network of computers. DLT is often used in conjunction with blockchain technology, although it can also be used with other types of distributed networks.

At its core, DLT is a way of distributing and synchronizing data across a network of computers. Rather than relying on a centralized database or authority to manage the data, DLT allows multiple parties to maintain copies of the same ledger, and to validate and update that ledger through a consensus mechanism.

One of the key benefits of DLT is that it allows for a high degree of trust and transparency in transactions, without relying on a single point of failure. Because the ledger is maintained by multiple parties, and each party has a copy of the same ledger, it becomes much more difficult for any single party to manipulate or corrupt the data.

DLT is used in a variety of applications, from financial services and supply chain management to healthcare and identity verification. In many cases, DLT is used in conjunction with blockchain technology, which provides an additional layer of security and immutability to the ledger.

Overall, DLT is a powerful tool for creating decentralized, trust less systems that allow for secure and transparent transactions across a network of participants. It has the potential to transform a wide range of industries and applications, and is likely to become an increasingly important technology in the years to come.

Some of the key benefits of DLT include:

Decentralization: DLT allows for the creation of decentralized networks that are not controlled by any single entity.

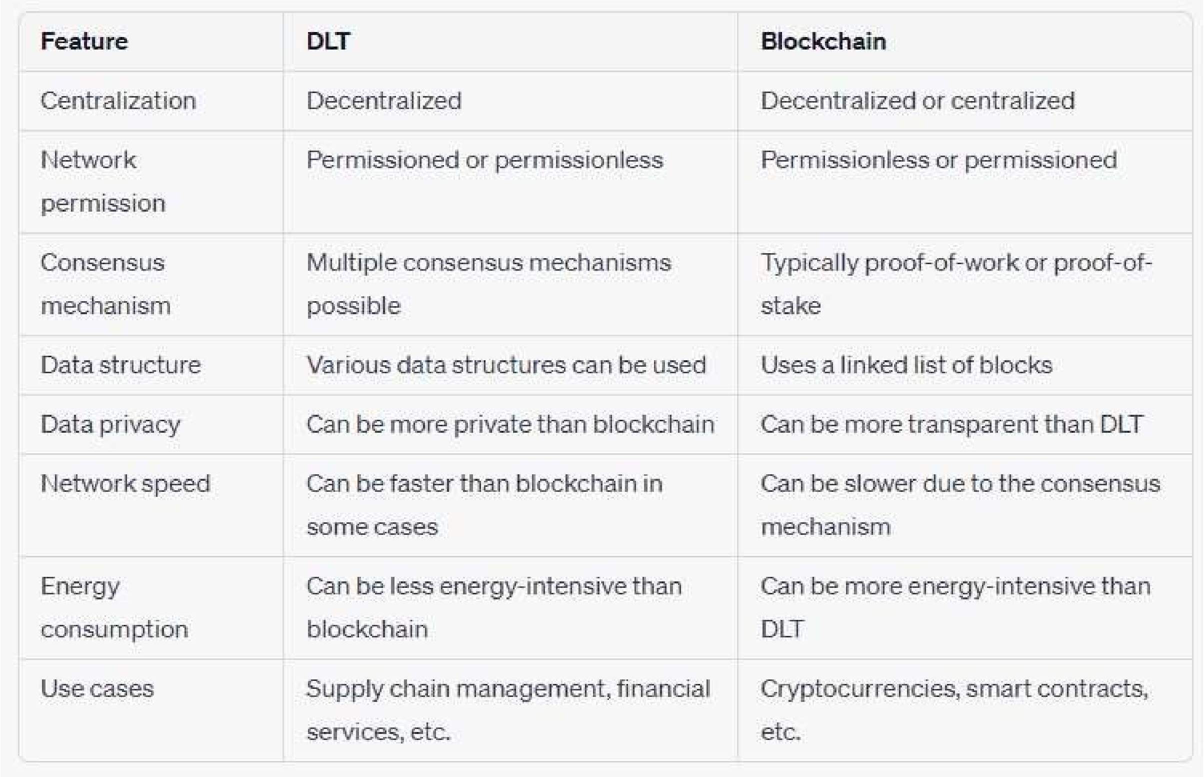
Security: Because DLT uses cryptographic techniques to secure transactions and data, it is highly secure and resistant to hacking and fraud.

Transparency: DLT provides a transparent and auditable record of all transactions, making it easy to track and verify data.

Efficiency: DLT can improve the efficiency of many processes by reducing the need for intermediaries and streamlining data sharing and validation.

Flexibility: DLT can be customized to suit a wide range of applications, making it a versatile tool for many industries and use cases.

Difference Between DLT and Blockchain



Feature	DLT	Blockchain
Centralization	Decentralized	Decentralized or centralized
Network permission	Permissioned or permissionless	Permissionless or permissioned
Consensus mechanism	Multiple consensus mechanisms possible	Typically proof-of-work or proof-of-stake
Data structure	Various data structures can be used	Uses a linked list of blocks
Data privacy	Can be more private than blockchain	Can be more transparent than DLT
Network speed	Can be faster than blockchain in some cases	Can be slower due to the consensus mechanism
Energy consumption	Can be less energy-intensive than blockchain	Can be more energy-intensive than DLT
Use cases	Supply chain management, financial services, etc.	Cryptocurrencies, smart contracts, etc.

CAP Theorem

The CAP theorem, also known as Brewer's theorem, is a fundamental concept in distributed computing that states that it is impossible for a distributed system to simultaneously provide consistency, availability, and partition tolerance. These three properties are referred to as the "CAP" properties, and they represent the core challenges of designing and implementing distributed systems.

Consistency refers to the requirement that all nodes in the system see the same data at the same time. In other words, if a value is written to one node, it must be immediately available to all other nodes in the system.

Availability refers to the requirement that the system must always be available for clients to read and write data, even in the face of network failures or other problems.

Partition tolerance refers to the ability of the system to function even when the network is partitioned or split into separate segments.

According to the CAP theorem, any distributed system can only guarantee two out of these three properties at any given time. In other words, if a system provides consistency and partition tolerance, it may not be able to provide full availability in the face of network failures. Similarly, if a system provides availability and partition tolerance, it may sacrifice some degree of consistency.

The CAP theorem has important implications for the design and implementation of distributed systems. Different systems may prioritize different CAP properties based on their specific needs and requirements, and it is important to carefully consider the trade-offs between consistency, availability, and partition tolerance when designing and deploying distributed systems.

Byzantine Generals problem Page 1-17

The **Byzantine Generals problem** is a classic problem in computer science and distributed systems that explores the challenges of achieving consensus in a network of nodes that may be unreliable or even actively malicious.

In the problem, a group of Byzantine generals are planning to attack a city. The generals are separated by distance and can only communicate through messengers. Some of the generals may be traitors who are trying to sabotage the attack by sending false information to other generals. The goal is to ensure that all loyal generals agree on a plan of attack, despite the possibility of traitors in the network.

The problem is often formulated as a game-theoretic problem in which each general must decide whether to attack or retreat based on the messages received from other generals. If all loyal generals agree to attack, then the attack will succeed. However, if even one general is a traitor and sends false information, it can lead to confusion and disagreement among the loyal generals, potentially resulting in a failed attack.

The Byzantine Generals problem has important implications for the design of distributed systems, as it highlights the challenges of achieving consensus in a network with unreliable or malicious nodes. Various solutions have been proposed to address this problem, such as the use of redundancy, cryptographic techniques, and consensus algorithms like the Byzantine Fault Tolerance algorithm.

Consensus Mechanism

A consensus mechanism is a process used in distributed systems to achieve agreement on a single data value or a state of the network among multiple nodes or participants. In blockchain networks, the consensus mechanism is a critical component that ensures the validity and security of transactions and blocks added to the chain.

Consensus mechanisms enable participants to reach agreement on the current state of the network by verifying and validating transactions, and then adding them to the blockchain in a secure and immutable manner. There are several types of consensus mechanisms, including Proof-of-Work (PoW), Proof-of-Stake (PoS), Delegated Proof-of-Stake (DPoS), and many others. Each consensus mechanism has its own unique set of rules, incentives, and penalties that govern how participants can validate transactions and create new blocks in the blockchain.

Overall, consensus mechanisms are essential for ensuring the integrity and security of distributed networks, and they are a key component of many blockchain-based applications.

Proof of Work (PoW)

Proof-of-Work (PoW) is a consensus mechanism used in many blockchain networks to validate transactions and create new blocks on the chain. In a PoW system, nodes or participants in the network must solve complex mathematical problems using computational power to verify transactions and add them to the blockchain.

The process of solving these problems, known as "mining," involves using powerful computers to compete to solve the problem first. The first node to solve the problem is rewarded with cryptocurrency or other incentives. This process of solving the problem is referred to as "proof of work."

The difficulty of the mathematical problem is adjusted regularly to ensure that the rate of block creation remains constant, which helps to maintain the security and integrity of the blockchain. The PoW mechanism is known for its high energy consumption, as the computational power required to solve the problems is significant.

Bitcoin, the first and most well-known blockchain network, uses a PoW consensus mechanism. However, other networks have since emerged, such as Ethereum, that use alternative consensus mechanisms such as Proof-of-Stake (PoS).

Advantages

1. High Level Security
2. It allow miners to earn crypto rewards

Disadvantages

1. It is expensive
2. It is not easily scalable
3. It uses ton of energy

Proof-of-Stake (PoS)

Proof-of-Stake (PoS) is a consensus mechanism used in some blockchain networks to validate transactions and create new blocks on the chain. In a PoS system, the nodes or participants that validate transactions and create new blocks are selected based on the amount of cryptocurrency they hold or "stake" in the network.

In a PoS system, participants can become validators by holding a certain amount of cryptocurrency in a designated wallet. These validators are then randomly selected to validate transactions and create new blocks on the blockchain. The rewards for validating transactions and creating new blocks are proportional to the amount of cryptocurrency held by the validator.

Compared to Proof-of-Work (PoW), PoS is less energy-intensive and requires less computational power, making it more environmentally friendly. Additionally, PoS systems are less susceptible to attacks, such as 51% attacks, as the cost of acquiring a majority stake in the network is much higher.

Ethereum is in the process of transitioning from a PoW to a PoS consensus mechanism, with the launch of Ethereum 2.0. Other blockchain networks, such as Cardano, already use PoS as their primary consensus mechanism.

Advantages

1. It offers fast and inexpensive transactions
2. It is an energy efficient mechanism

Disadvantages

1. More prone to attack as attacker needs to spend only some crypto currency
2. Difficulty in achieving initial distribution

Cryptographic primitives are essential components of blockchain technology that enable secure transactions, maintain privacy, and prevent fraud. Here are some of the cryptographic primitives commonly used in blockchain:

Hash functions:

- A hash function is a mathematical function that takes input data and returns a fixed-size output called a hash.
- In blockchain, hash functions are used to create unique digital fingerprints of data, such as transactions or blocks, which can be used to verify the integrity and authenticity of the data.

Public-key cryptography:

- Public-key cryptography, also known as asymmetric cryptography, is a cryptographic system that uses a pair of keys, one public and one private, to encrypt and decrypt data.
- In blockchain, public-key cryptography is used to generate digital signatures, which are used to verify the identity of the sender and ensure the integrity of the data.

Digital signatures:

- Digital signatures are a cryptographic technique used to authenticate the sender of a message and ensure the integrity of the message.
- In blockchain, digital signatures are used to verify the authenticity of transactions and ensure that only authorized parties can initiate transactions.

Merkle trees:

- A Merkle tree is a data structure that is used to verify the integrity of large datasets.
- In blockchain, Merkle trees are used to create a hash of all the transactions in a block, which can be used to verify that the block has not been tampered with.

Consensus algorithms:

- Consensus algorithms are used in blockchain to ensure that all nodes in the network agree on the state of the blockchain.
- There are several types of consensus algorithms, such as Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS), each with its own cryptographic primitives and security properties.

Overall, cryptographic primitives are an essential component of blockchain technology that enables its security, privacy, and decentralization.

Block in Blockchain

Blocks are an essential component of the blockchain technology. A block is a collection of verified transactions that are grouped together and added to the blockchain. Each block has a unique cryptographic hash that identifies it and links it to the previous block, forming a chain of blocks, hence the name "blockchain".

In a blockchain network, nodes work together to validate and verify transactions by solving complex cryptographic puzzles. Once a block of transactions is verified, it is broadcast to the network and added to the blockchain. The transactions in the block are then considered confirmed and cannot be altered or deleted.

A block typically contains several pieces of information, including:

Block header:

- The block header contains the block's unique cryptographic hash, as well as information about the block's timestamp, nonce, and difficulty level.
- The block header also includes a reference to the previous block in the chain, creating a linked sequence of blocks that forms the blockchain.

Transactions:

- A block contains a set of verified transactions that have been processed and validated by the network.
- Each block contains a set of transactions that have been verified by the network and are ready to be added to the blockchain.
- Depending on the specific blockchain, these transactions may include transfers of cryptocurrency, updates to smart contracts, or other types of data.

Previous block hash:

- Each block is linked to the previous block in the blockchain by including the previous block's hash in the current block's header.
- This reference takes the form of the previous block's unique cryptographic hash, which is included in the block header of the current block.
- By linking each block in this way, the blockchain creates an unbroken chain of data that is easy to verify and difficult to alter.

Merkle root:

- The transactions in each block are typically arranged in a tree-like structure called a Merkle tree, with each transaction serving as a leaf node in the tree.
- The Merkle root is a single hash value that represents the entire tree, and it is included in the block header of the block.

- This helps ensure the integrity of the data in the block, as any tampering with even a single transaction would cause the Merkle root to change.

Blocks play a crucial role in the security and integrity of the blockchain. They provide a tamper-proof record of transactions that can be verified and validated by all nodes on the network. Additionally, the use of blocks allows the blockchain to be distributed across multiple nodes in the network, ensuring that no single entity has control over the network or the data stored within it.

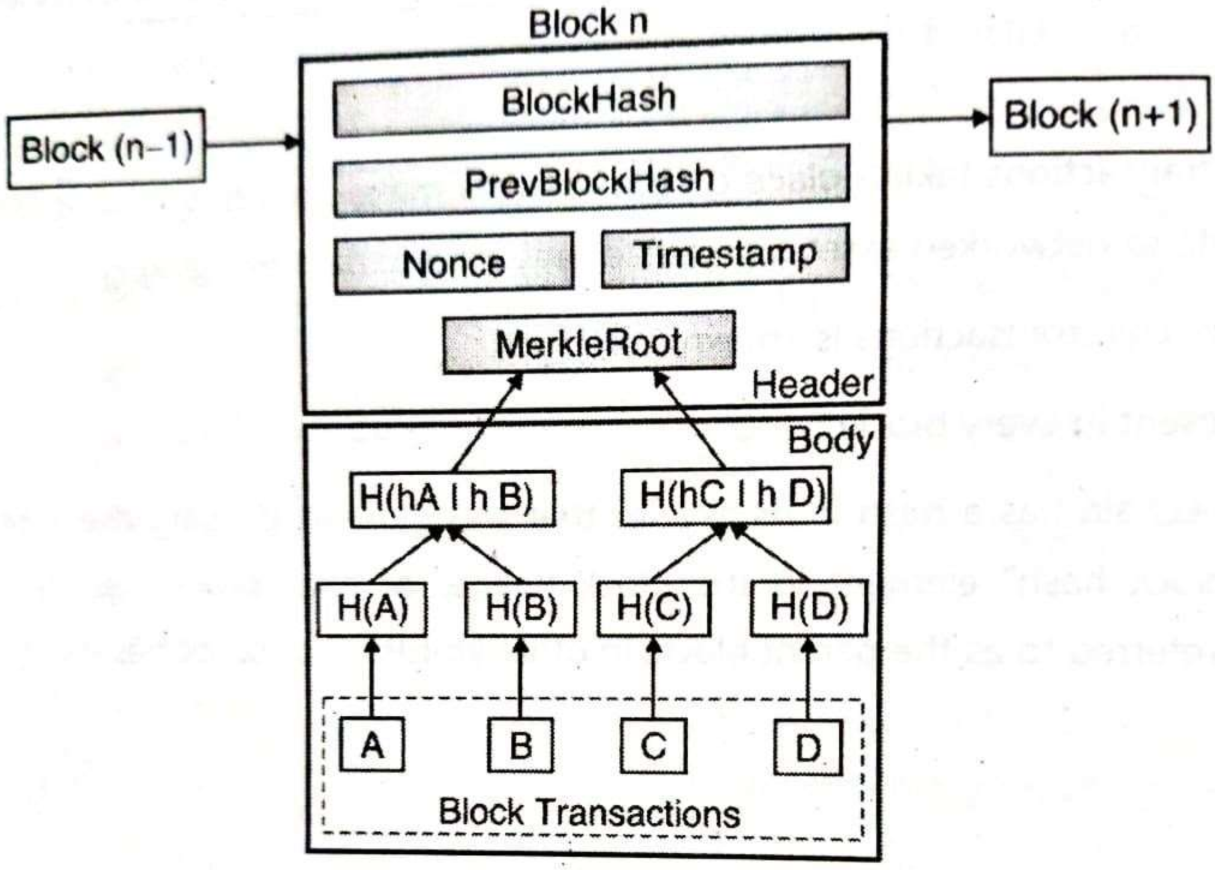


Fig. 1.11.2 : Structure of a block

Genesis block

The genesis block is the very first block in a blockchain network. It is the initial block in the chain that is created by the blockchain's creator, and from which all subsequent blocks are added. The genesis block is unique in that it has no previous block to reference in its header, as there are no previous blocks in the blockchain when it is created.

The genesis block contains a set of transactions, similar to any other block in the blockchain. However, these transactions typically have a special purpose, such as creating the initial supply of cryptocurrency in the network, or setting up the initial state of smart contracts. In some cases, the genesis block may also contain a special message or quote from the creator of the blockchain.

The creation of the genesis block is an important step in the launch of a new blockchain network, as it sets the initial parameters for the network and establishes the foundation for all subsequent blocks. The creation of the genesis block is typically followed by a period of mining or validation, during which additional blocks are added to the chain and the network begins to operate in earnest.

Here are some key characteristics of a genesis block:

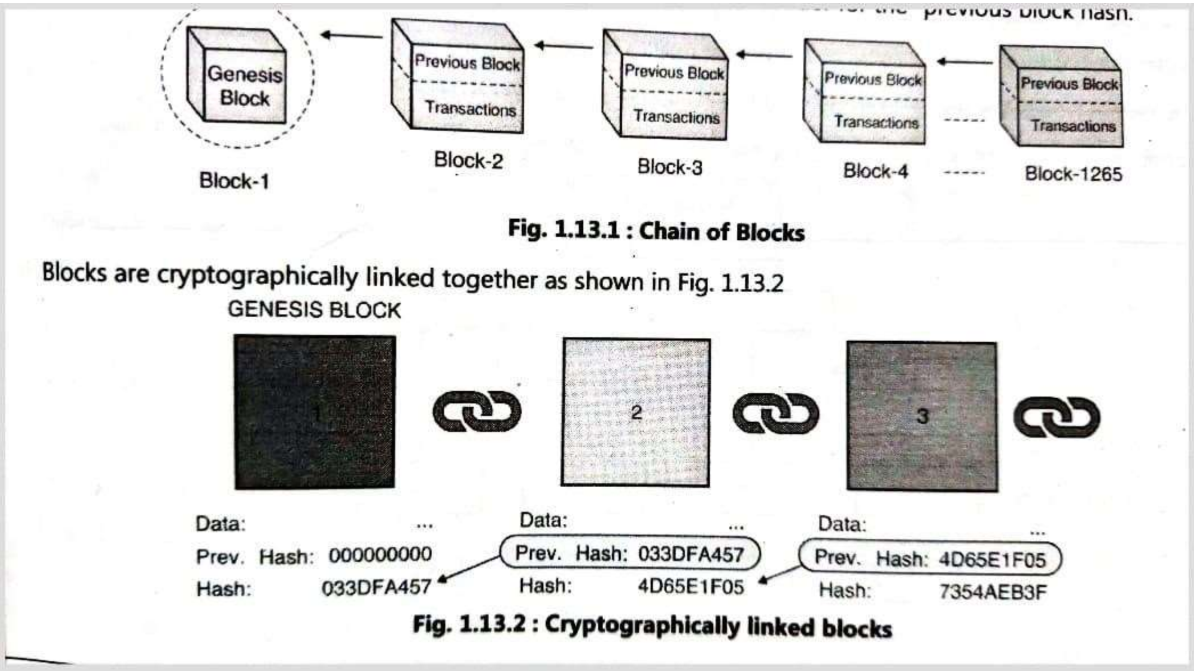
Unique hash: The genesis block has a unique hash that distinguishes it from all other blocks in the network. This hash is typically hardcoded into the network's software, and all subsequent blocks in the blockchain are linked to this hash.

No parent block: Since the genesis block is the first block in the chain, it has no parent block to reference. This means that it doesn't have any previous block data to reference, which is why it is typically hardcoded into the network's software.

Initial parameters: The genesis block contains the initial parameters of the blockchain network. This includes things like the initial difficulty level of mining, the block reward for mining a block, and the maximum supply of tokens that can be created.

Unique data: The genesis block contains unique data that is specific to the network. This can include things like the date and time the network was launched, the names of the founders, and other relevant information.

Linking blocks in a blockchain



previous block hash.

Fig. 1.13.1 : Chain of Blocks

Blocks are cryptographically linked together as shown in Fig. 1.13.2

Fig. 1.13.2 : Cryptographically linked blocks

Linking blocks in a blockchain involves creating a chain of blocks where each block is cryptographically linked to the previous block. This is achieved through the use of a cryptographic hash function that takes the data of the previous block and creates a unique hash that is added to the current block. This process is called hashing.

When a block is added to the blockchain, it is verified by nodes on the network to ensure that the data is valid and that the hash of the block is correct. Once the block is verified, it is added to the blockchain, and the next block can be created. The next block will contain the hash of the previous block, which creates a chain of blocks that are linked together.

The linking of blocks in a blockchain provides several benefits. First, it ensures that the data in the blockchain is secure and tamper-proof since any changes to a block would cause the hash of the block to change, which would invalidate the entire chain. Second, it provides a transparent and auditable record of all transactions on the network since every block contains a record of all transactions that occurred on the network.

Overall, linking blocks in a blockchain is a critical component of its security and functionality. By creating a chain of blocks that are cryptographically linked together, a blockchain provides a tamper-proof and transparent record of all transactions on the network.

Merkle Tree

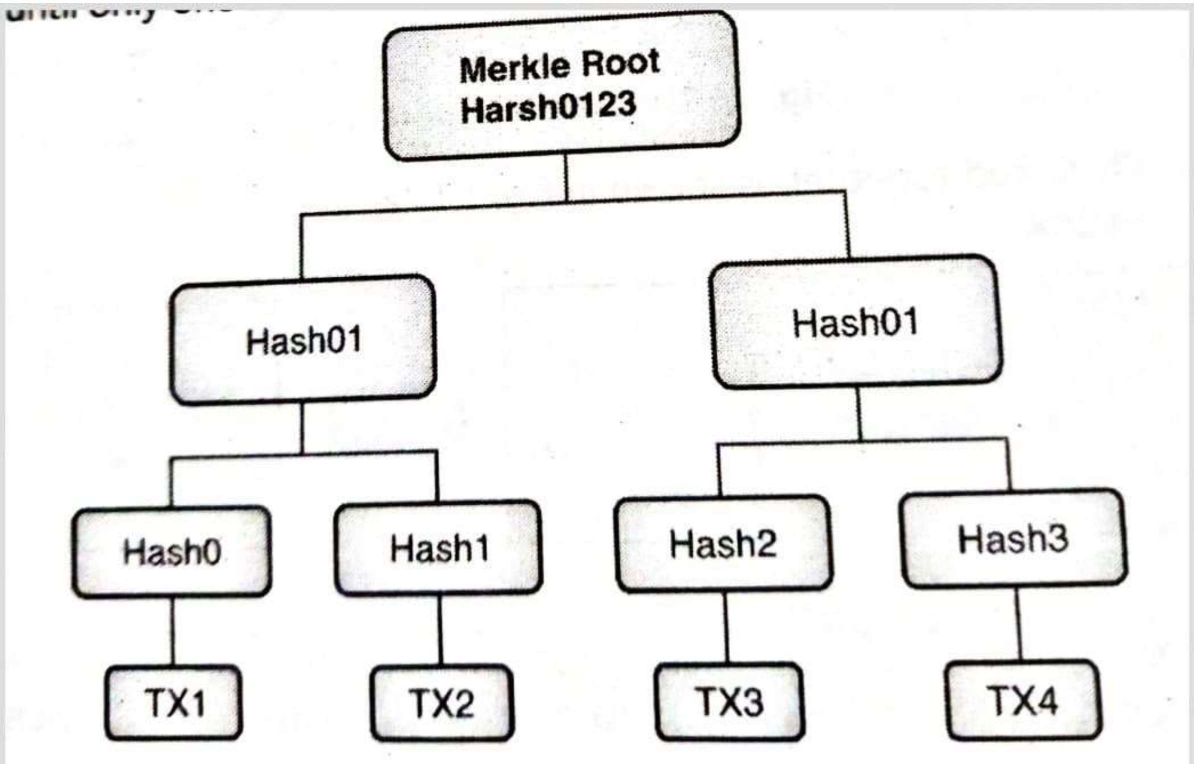


Fig. 1.14.1

A Merkle tree, also known as a hash tree, is a data structure used in blockchain technology to efficiently and securely verify the integrity of large amounts of data. It was named after its inventor, Ralph Merkle.

A Merkle tree is a binary tree where each leaf node represents a data block, and each non-leaf node represents a hash of its child nodes. The root node of the tree is called the Merkle root, which represents the entire set of data blocks in the tree.

To verify the integrity of the data, a node can check the Merkle root against a trusted source, such as a trusted blockchain node or a hash published in a public record. By verifying the Merkle root, the node can ensure that the entire set of data blocks has not been tampered with or modified in any way.

Merkle trees are used extensively in blockchain technology to verify the integrity of transaction data. In a blockchain, each block contains a set of transaction data, and the block header contains a hash of the Merkle root of the transaction data. This ensures that the entire set of transaction data is verified and validated when a new block is added to the chain.

The Merkle tree is a powerful data structure that allows for efficient and secure verification of large amounts of data. Its use in blockchain technology has helped to create a secure and tamper-proof system for recording and verifying transaction data.

Chapter 2: Bitcoin

What is Bitcoin

Bitcoin is a decentralized digital currency that was invented by an unknown person or group of people using the name Satoshi Nakamoto in 2008. It is a form of electronic cash that allows for peer-to-peer transactions without the need for a centralized intermediary, such as a bank or government.

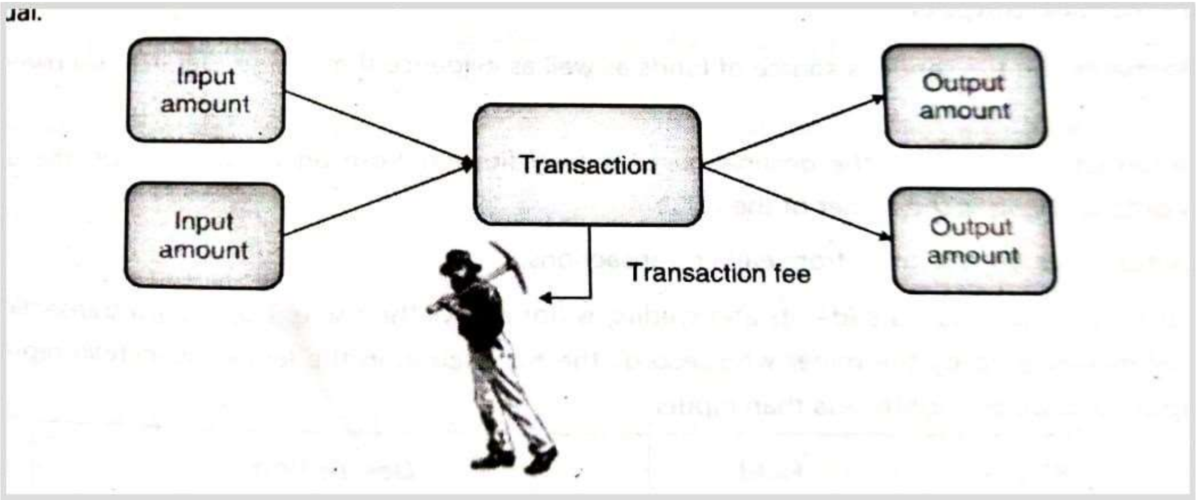
Bitcoin operates on a decentralized network called the blockchain, which is a public ledger that records all Bitcoin transactions. The blockchain is maintained by a network of users, called nodes, who validate transactions and maintain the integrity of the network.

Bitcoin transactions are made using a digital wallet, which stores a user's private keys that allow them to send and receive Bitcoin. Transactions are verified by miners who use powerful computers to solve complex mathematical problems and add new blocks to the blockchain. In return, miners receive newly minted Bitcoin as a reward.

One of the key features of Bitcoin is its limited supply. The total supply of Bitcoin is capped at 21 million, and it is designed to become increasingly difficult to mine as the supply approaches this limit. This is intended to prevent inflation and ensure the value of Bitcoin remains stable.

Bitcoin has been praised for its ability to provide financial freedom and anonymity, as transactions are not tied to a user's identity. However, it has also been criticized for its association with illegal activities and its potential to be used for money laundering and other illicit purposes.

Bitcoin Transaction



A Bitcoin transaction is the transfer of Bitcoin value between two or more Bitcoin wallets. The transaction is recorded on the Bitcoin blockchain, which is a public ledger that maintains a record of all transactions.

Bitcoin transactions are designed to be fast, secure, and transparent. The decentralized nature of the blockchain ensures that transactions are validated and recorded by a network of users, rather than a centralized intermediary. This makes Bitcoin transactions more resistant to fraud and manipulation than traditional financial transactions.

To initiate a Bitcoin transaction, the sender must first have a Bitcoin wallet, which contains their private key. The private key is a secret code that is used to sign the transaction and verify that the sender is the owner of the Bitcoin they are sending.

Once the sender has initiated the transaction, it is broadcast to the Bitcoin network, where it is verified and validated by nodes on the network. Each node on the network maintains a copy of the blockchain, which contains a record of all previous transactions.

The nodes on the network use complex mathematical algorithms to verify the transaction and ensure that the sender has sufficient funds to complete the transaction. Once the transaction is validated, it is added to a block, which is then added to the blockchain.

Each transaction on the Bitcoin blockchain contains several key components, including:

Inputs: These are the Bitcoin addresses that the sender is using to send Bitcoin. The inputs are derived from previous transactions on the blockchain and must be verified to ensure that the sender has sufficient funds to complete the transaction.

Outputs: These are the Bitcoin addresses that the sender is sending Bitcoin to. Each output includes the amount of Bitcoin being sent and the receiving address.

Fees: Bitcoin transactions require a fee to be paid to the miners who validate the transaction and add it to the blockchain. The fee is typically based on the size of the transaction and the level of demand for block space on the network.

Signatures: Each transaction requires a digital signature from the sender's private key to verify that they are the rightful owner of the Bitcoin being sent.

Concept of bitcoin

Bitcoin is a decentralized digital currency that operates without a central bank or administrator. It allows peer-to-peer transactions without intermediaries and provides a transparent, secure, and anonymous platform for conducting financial transactions.

Bitcoin Ownership is established by:

Keys:

- Keys refer to the private and public keys that are used to facilitate transactions on the network.
- **Private keys** are secret codes that are used to sign transactions and prove ownership of Bitcoins. They are created by Bitcoin wallet software and should be kept secret and secure.
- **Public keys** are derived from private keys and are used to create Bitcoin addresses. Bitcoin addresses are unique identifiers that represent the destination of a Bitcoin transaction.
- Together, private and public keys form a key pair that is used to verify the authenticity of transactions and protect against fraud. When you sign a transaction using your private key, the network verifies the signature using your public key. If the signature is valid, the transaction is added to the blockchain and the ownership of the Bitcoins is transferred to the recipient.

Bitcoin Address

- A Bitcoin address is a string of alphanumeric characters that represents the destination of a Bitcoin transaction. It is similar to a bank account number, but instead of identifying a bank account, it identifies a Bitcoin wallet.
- A Bitcoin address is generated by a user's Bitcoin wallet software, which creates a unique public key that is then hashed to produce the address. This address is made up of a series of characters, ranging from 26 to 35 characters in length, and is represented as a sequence of letters and numbers.
- When you want to receive Bitcoins from someone, you provide them with your Bitcoin address. This address is used to identify the recipient of the transaction and to ensure that the Bitcoins are sent to the correct wallet.
- Example. 1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2

Bitcoin Wallet

- A Bitcoin wallet is a software program that stores digital keys and enables users to send, receive, and manage their Bitcoins.
- Bitcoin wallets come in different forms, including desktop, mobile, web, and hardware wallets.
- Desktop wallets are installed on a computer and offer a high level of security, but they are not as convenient to use as mobile or web wallets.
- Mobile wallets are installed on smartphones and offer a convenient way to manage Bitcoins on-the-go.

- Web wallets are accessed through a web browser and are easy to use but may not offer the same level of security as other types of wallets.
- Hardware wallets are physical devices that store digital keys offline, providing the highest level of security.

Bitcoin Transaction UTXO 2-12

UTXO stands for Unspent Transaction Output, which is a fundamental concept in the Bitcoin network. In a UTXO-based system, every Bitcoin transaction creates one or more UTXOs, which are essentially unspent outputs of that transaction.

In a UTXO-based transaction, when you want to send Bitcoins to someone, you must first identify one or more UTXOs that belong to your Bitcoin address and that have not been spent in previous transactions. You then specify the amount of Bitcoins you wish to send, along with the recipient's Bitcoin address. The transaction is then signed using your private key and broadcast to the network.

When the network receives your transaction, it verifies that the UTXOs you are spending are legitimate and that you have sufficient funds to cover the transaction. If the transaction is valid, the UTXOs you are spending are marked as spent, and new UTXOs are created for the recipient of the transaction.

It's important to note that UTXOs are indivisible, meaning that if you have a UTXO with 2 Bitcoins and you want to send 1 Bitcoin to someone, you must create a new UTXO for yourself with the remaining 1 Bitcoin. This means that Bitcoin transactions often involve multiple UTXOs, and the Bitcoin network must keep track of all UTXOs in the system to ensure the integrity and security of the network.

For example, if you are **sending 1 Bitcoin** to someone and you have a UTXO with 2 Bitcoins, your transaction would create two new UTXOs: one with 1 Bitcoin that is sent to the recipient's Bitcoin address, and one with 1 Bitcoin that is sent back to your Bitcoin address as change.

This process of creating new UTXOs and marking old ones as spent is what enables the Bitcoin network to maintain the integrity of the blockchain and prevent double-spending. Each UTXO is uniquely identified and can only be spent once, which ensures that transactions are processed in a secure and transparent manner.

Validation of transaction in Bitcoin

In the Bitcoin network, transactions are validated through a process called "mining". This process involves adding new transactions to the Bitcoin blockchain, which is essentially a public ledger that records all Bitcoin transactions ever made.

Miners use powerful computers to solve complex mathematical problems in order to validate transactions and add them to the blockchain. This process involves verifying that the transaction is legitimate and that the sender has sufficient funds to complete the transaction.

Once a miner successfully validates a transaction, it is added to the blockchain and becomes a permanent part of the public ledger. This means that the transaction cannot be reversed or altered without the consensus of the network.

In addition to mining, Bitcoin also uses a system of digital signatures to ensure that transactions are valid. Each user has a unique digital signature that is used to verify their identity and authorize transactions. This system helps prevent fraud and ensures that transactions are secure.

The validation process involves several steps, including:

Verification of the transaction inputs: The inputs to the transaction must be valid, which means that they must have been created by a previous transaction and must be unspent.

Verification of the transaction outputs: The outputs of the transaction must be valid Bitcoin addresses and the amount being sent must be less than or equal to the total amount of the inputs.

Verification of the digital signatures: Each input to the transaction must have a corresponding digital signature that verifies the identity of the sender and authorizes the transaction.

Verification of the transaction fees: Transactions with higher fees are typically given priority by miners, so the transaction must include a sufficient fee to incentivize miners to include it in the next block.

Once the transaction is validated, it is added to a pool of unconfirmed transactions. Miners then select transactions from this pool and include them in the next block of the blockchain.

To add a block to the blockchain, miners must solve a complex mathematical problem called a proof-of-work. This involves finding a hash that meets a specific set of criteria. The first miner to solve the problem and add the block to the blockchain is rewarded with newly created Bitcoin and transaction fees.

Once a block is added to the blockchain, the transactions in that block are considered confirmed and cannot be reversed or altered without the consensus of the network.

Elliptic Curve Cryptography (ECC) Page: 2-14

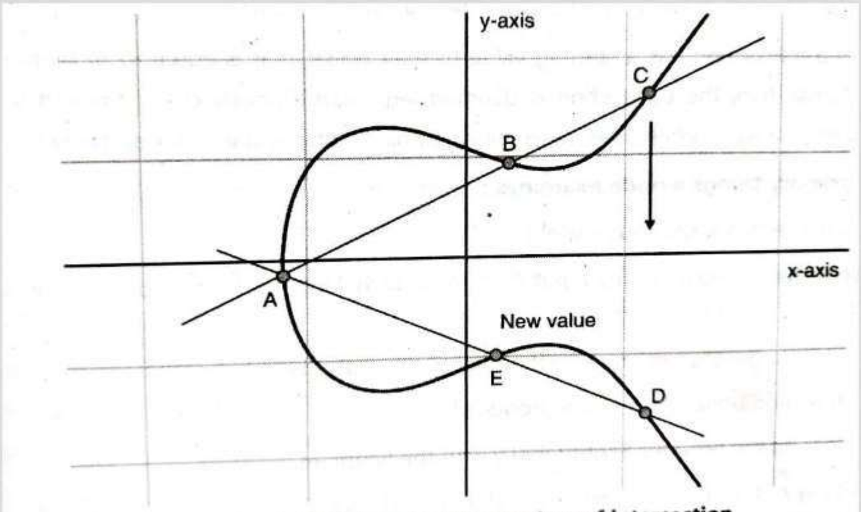


Fig. 2.7.1 : ECC showing three points pf intersection

Elliptic Curve Cryptography (ECC) is a type of public-key cryptography that is used to secure data in a wide range of applications, including digital signatures, key agreement protocols, and encryption. It is based on the properties of elliptic curves over finite fields and provides a higher level of security than other traditional public-key algorithms such as RSA.

The basic idea behind ECC is to use points on an elliptic curve as the basis for generating cryptographic keys. An elliptic curve is a mathematical curve defined by an equation of the form **y^2 = x^3 + ax + b**, where a and b are constants. The curve is symmetric about the x-axis and has a unique point at infinity.

To generate a public-private key pair in ECC, a user selects a point on the elliptic curve and a random integer. The point is then multiplied by the integer using a defined mathematical operation known as scalar multiplication. The resulting point is the public key, while the integer is the private key.

One of the main advantages of ECC is its ability to provide the same level of security as other public-key algorithms such as RSA, but with much smaller key sizes. This makes it particularly useful in situations where bandwidth and storage are limited, such as mobile devices and embedded systems.

In addition to its smaller key sizes, ECC also offers other advantages such as faster computation, better resistance to attacks based on quantum computing, and more efficient use of computational resources.

However, ECC is not without its challenges. One of the main challenges is the selection of appropriate parameters, such as the elliptic curve and the size of the finite field. Poorly chosen parameters can lead to security vulnerabilities, and there is ongoing research to develop better methods for parameter selection.

BASE 58 Page 215

BIP-38 Page 2-16

Multi-signature address

A multi-signature address, also known as a multi-sig address, is a type of Bitcoin address that requires multiple signatures to authorize a transaction. It is created using a special type of Pay to Script Hash (P2SH) transaction.

A multi-sig address is associated with a script that specifies the conditions under which the funds can be spent. The script typically requires a certain number of signatures from a specified set of private keys before a transaction can be authorized.

For example, a 2-of-3 multi-sig address would require two out of three specified private keys to sign a transaction before it can be authorized. This can provide an extra layer of security for Bitcoin transactions, as it requires multiple parties to agree before funds can be spent.

Multi-sig addresses can be useful in a variety of applications, such as:

Custody services: In a custody service, multiple parties may be responsible for approving transactions, and a multi-sig address can be used to ensure that no single party can spend the funds without the approval of others.

Escrow services: In an escrow service, a multi-sig address can be used to hold funds until the terms of an agreement are met. The funds can only be released when all parties have agreed to the release.

Business partnerships: In a business partnership, a multi-sig address can be used to ensure that all parties have equal control over the funds and that no single party can make unilateral decisions.

Multi-sig addresses are a powerful tool for enhancing the security of Bitcoin transactions. They allow for more complex spending conditions and can be used in a variety of applications where multiple parties are involved.

Advantages of using multi signature addresses in Bitcoin:

Increased security: Multisig addresses require multiple signatures to authorize a transaction, making them more secure than single signature addresses. This means that even if one of the private keys is compromised, the funds in the address remain secure.

Shared control: Multisig addresses are useful for businesses and organizations that require multiple people to manage their funds. By requiring multiple signatures, the address ensures that no one person can make a transaction without the approval of the others.

Disadvantages of using multisignature addresses in Bitcoin:

Higher fees: Multisig transactions require more computational resources to process, which can result in higher transaction fees.

Complexity: Multisig addresses can be more complex to set up and manage than single signature addresses, requiring more technical knowledge.

Pay to Script Hash Page 2-20

Transaction script

In Bitcoin, a transaction script is a simple programming language that is used to specify the conditions under which funds in a transaction can be spent. A transaction script is embedded in the output of a transaction, which determines who can spend the funds and under what conditions.

Transaction scripts are written in a stack-based language called Script, which is similar to Forth programming language. The script contains a series of opcodes, which are used to perform specific operations such as mathematical calculations and comparisons.

The most common type of transaction script in Bitcoin is the Pay-to-Public-Key-Hash (P2PKH) script. This script requires the spender to provide a digital signature that matches the public key associated with a specific Bitcoin address. The script checks the validity of the signature and the public key, and if they match, the funds are transferred to the new address specified in the transaction output.

Another common type of transaction script is the Pay-to-Script-Hash (P2SH) script, which allows more complex spending conditions to be specified. P2SH scripts enable the use of multi signature addresses and other advanced spending conditions.

Transaction scripts provide a flexible and powerful mechanism for specifying spending conditions in Bitcoin transactions. They enable the creation of more complex transactions beyond simple transfers of funds, allowing for a wide range of use cases and applications to be built on top of the Bitcoin protocol.

Script Address in bitcoin

In Bitcoin, a script address (also known as a pay-to-script-hash or P2SH address) is a type of address used to send and receive Bitcoin transactions.

The script address is a 20-byte hash that is derived from a script (a sequence of instructions) that specifies the conditions for spending the Bitcoin sent to the address. These conditions can be complex, and may involve multiple signatures or require a specific sequence of events to occur before the funds can be spent.

To create a script address, the script is first hashed using the SHA256 algorithm, and then the resulting hash is hashed again using the RIPEMD-160 algorithm. The resulting 20-byte hash is then encoded in base58 format to create the final script address.

Script addresses start with the number 3, and are typically longer than regular Bitcoin addresses. They are commonly used for more advanced transaction types, such as multi-signature transactions or transactions that involve time-locks.

When sending Bitcoin to a script address, the sender includes a script that satisfies the conditions specified by the script address. When the receiver wants to spend the Bitcoin, they must provide a script that also satisfies the conditions. The script is then verified by the Bitcoin network before the transaction is confirmed.

Incentive bases engineering

Bitcoin is a decentralized digital currency that operates on a blockchain network. It was designed with a specific set of incentives to ensure that the network is secure and reliable. These incentives are known as the Bitcoin incentive system or Bitcoin mining incentive.

Bitcoin miners are individuals or organizations who use their computing power to validate transactions on the Bitcoin network. In return for their efforts, they receive a block reward, which is currently set at 6.25 bitcoins per block. This reward serves as an incentive for miners to continue verifying transactions and keeping the network secure.

The mining process is designed to be difficult and resource-intensive, which helps to prevent fraud and ensures that the network is secure. Miners must compete to solve a complex mathematical puzzle, and the first miner to solve the puzzle receives the block reward.

In addition to the block reward, miners also receive transaction fees for including transactions in their blocks. These fees are paid by users who want their transactions to be prioritized and included in the next block. The higher the fee, the more likely it is that the transaction will be processed quickly.

The incentive system is designed to encourage miners to act in the best interests of the network. If a miner attempts to manipulate the system or engage in fraudulent behavior, they risk losing their block reward and transaction fees. This provides a strong disincentive for bad actors and helps to maintain the integrity of the network.

The Bitcoin incentive system is a key component of the network's security and reliability. By providing strong incentives for miners to act honestly and maintain the network, Bitcoin has become one of the most successful cryptocurrencies in the world.

The Extended Bitcoin Network

The Extended Bitcoin Network refers to the various nodes and users of the Bitcoin network beyond its core protocol. These include individuals, organizations, and projects that build applications and services on top of the Bitcoin blockchain or utilize its underlying technology.

The Bitcoin network is open-source, meaning anyone can build on it and extend its functionality. One of the most popular extended Bitcoin networks is the Lightning Network. This is a layer 2 payment protocol that enables fast and cheap micropayments. By using payment channels that are settled off-chain, the Lightning Network can process a large number of transactions per second without congesting the main Bitcoin blockchain.

Other extended networks include Bitcoin wallets, payment processors, and exchanges that allow users to buy, sell, and store Bitcoin. Additionally, Bitcoin mining pools and hardware manufacturers also form part of the extended Bitcoin network as they contribute to the security and processing power of the network.

the extended Bitcoin network is a complex and interconnected ecosystem of developers, users, and businesses working together to make Bitcoin more useful, accessible, and valuable. By building on top of the blockchain and creating new applications and services, these entities are helping to shape the future of the digital economy.

Bitcoin Relay Network

The BRN is a decentralized network that functions as a layer on top of the existing Bitcoin network. Its primary purpose is to improve the speed and efficiency of the Bitcoin network by reducing latency and increasing the propagation speed of new transactions and blocks.

When a new transaction is broadcast on the Bitcoin network, it typically takes a certain amount of time for that transaction to be propagated to all of the nodes on the network. This can lead to delays and inefficiencies in the system, particularly if there are large numbers of transactions being processed simultaneously.

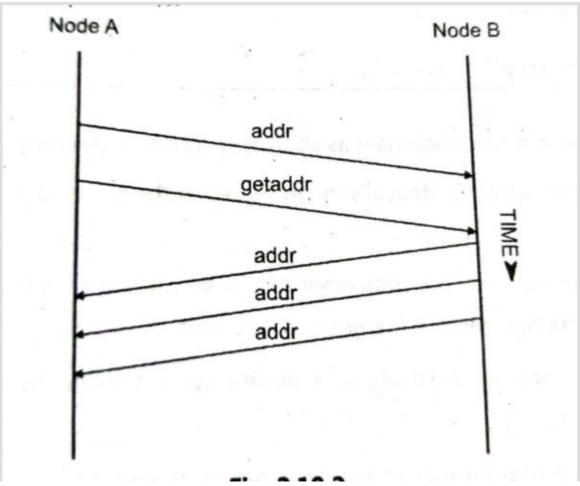
The BRN works by creating a mesh network of nodes that are interconnected and constantly exchanging information about new transactions and blocks. When a new transaction is broadcast on the network, it is immediately relayed to all of the other nodes on the network, ensuring that it is propagated as quickly as possible.

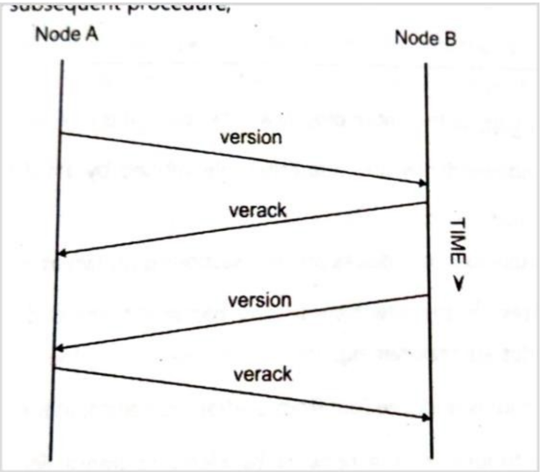
This helps to reduce the time it takes for transactions to be confirmed on the Bitcoin network, which can be particularly important for applications that require fast, reliable transaction processing. For example, if you are using Bitcoin to make a purchase online, you want the transaction to be confirmed as quickly as possible so that you can receive your goods or services without delay.

Another advantage of the BRN is that it helps to improve the reliability and security of the Bitcoin network. By creating a mesh network of interconnected nodes, the BRN ensures that the network is more resilient to attacks and failures. Even if some nodes on the network go offline or are attacked, the other nodes can continue to function normally and maintain the integrity of the blockchain.

The Bitcoin Relay Network is an important innovation that helps to improve the speed, efficiency, and reliability of the Bitcoin network. By reducing latency and increasing the propagation speed of new transactions and blocks, the BRN makes it easier to use Bitcoin for a wide range of applications and use cases, and helps to ensure the long-term viability of the Bitcoin network.

network discovery





In the Bitcoin network, network discovery refers to the process by which nodes discover and connect with each other. When a node first joins the network, it needs to find other nodes to connect with in order to participate in the network and propagate transactions and blocks.

Bitcoin uses a peer-to-peer (P2P) network architecture, where each node in the network is connected to multiple other nodes. When a node first starts up, it typically connects to a few known nodes, often called "seed nodes", which are hardcoded into the client software. These seed nodes help the new node to discover other nodes in the network.

Once a node has connected to one or more other nodes, it can request a list of their peers, and start connecting to those as well. This process continues recursively, allowing the node to rapidly discover and connect with a large number of nodes in the network.

In order to prevent malicious nodes from flooding the network with fake nodes, Bitcoin uses a technique called "address flooding prevention". This involves limiting the rate at which a node can send addresses to other nodes, and verifying that the addresses it receives are valid before connecting to them.

Network discovery is an essential component of the Bitcoin network, allowing nodes to connect and communicate with each other, and enabling the decentralized operation of the network.

Information typically included in Bitcoin network discovery:

1. IP address
2. Port number
3. Node ID
4. Blockchain data
5. Services offered
6. Protocol version.

A Simplified Payment Verification (SPV) Node is a type of cryptocurrency node that allows users to participate in a blockchain network without having to store the entire blockchain ledger. SPV nodes are commonly used by mobile wallets and other lightweight applications that need to quickly verify the validity of transactions on the network.

Instead of downloading and storing the entire blockchain, an SPV node only downloads a small portion of the blockchain data, typically the block headers. This data contains information such as the block's timestamp, hash, and transaction count, which allows the node to verify the authenticity of transactions without having to download the entire blockchain.

To verify a transaction, an SPV node sends a request to full nodes on the network to provide the relevant transaction data. The full nodes then provide the data, which the SPV node can use to verify the transaction.

SPV nodes provide a lightweight alternative to full nodes, making it possible for users with limited storage space or processing power to participate in a blockchain network. However, because they rely on full nodes for transaction data, they are not as secure as full nodes and are vulnerable to certain types of attacks.

Advantages of Simplified Payment Verification (SPV) Nodes:

- **Lightweight:** SPV nodes only need to download a small portion of the blockchain data, making them suitable for mobile devices and other lightweight applications**.**
- **Faster:** Because they only need to download a small portion of the blockchain data, SPV nodes can verify transactions faster than full nodes**.**

Disadvantages of Simplified Payment Verification (SPV) Nodes:

- **Reduced security:** SPV nodes rely on full nodes for transaction data, which means they are vulnerable to certain types of attacks such as a 51% attack.
- **Limited functionality**: SPV nodes do not have access to the entire blockchain data and therefore have limited functionality compared to full nodes.

Transaction pool

In the Bitcoin network, a transaction pool (also known as mempool) is a list of unconfirmed transactions that have been broadcasted to the network by users, but have not yet been added to the blockchain by miners.

When a user initiates a Bitcoin transaction, it is first broadcasted to the nodes on the network. Each node then verifies the transaction and adds it to its own local transaction pool. The transaction will be forwarded to other nodes on the network, and eventually propagated to all nodes.

Miners are responsible for confirming transactions and adding them to the blockchain. They do this by creating a new block that includes a list of verified transactions, which they then broadcast to the network. To incentivize miners to include their transactions in the next block, users can attach a transaction fee to their transactions.

Miners typically select transactions with the highest transaction fees from the transaction pool to include in the next block they mine. Transactions with lower fees may remain unconfirmed in the pool for a longer time, as miners have no incentive to include them in their blocks.

The transaction pool is constantly changing, as new transactions are added and confirmed transactions are removed. Nodes in the Bitcoin network can estimate the time it will take for a transaction to be confirmed by analysing the transaction pool and the current state of the network. Users can also check the size and status of the transaction pool to determine the best transaction fee to include with their transactions.

The transaction pool in Bitcoin is a temporary holding area for unconfirmed transactions that have been broadcasted to the network. It plays an important role in the Bitcoin network by allowing users to submit transactions and miners to select which transactions to include in their blocks. The transaction pool also helps users estimate the time it will take for their transactions to be confirmed and choose an appropriate transaction fee.

What is Bitcoin Test net

The Bitcoin Testnet is a network created for testing and experimenting with Bitcoin-related applications and software, without using real bitcoins or affecting the main Bitcoin network. It is essentially a separate blockchain that operates in the same way as the main Bitcoin blockchain but uses a different set of tokens called "testnet coins."

The Testnet was created to provide a safe and sandboxed environment for developers and users to test new features and updates to Bitcoin software before deploying them on the main network. It allows developers to experiment with new features, test their code for bugs and errors, and ensure the compatibility of their applications with the Bitcoin network without risking the loss of real bitcoins.

Testnet coins can be obtained for free from various sources such as testnet faucets or mining them yourself. These coins have no real value and can be freely exchanged, transferred, and used without any financial risk. Testnet coins can be easily distinguished from real bitcoins as their addresses start with "m" or "n" instead of "1" or "3".

There are three main types of testnets used in the Bitcoin ecosystem, each with a different purpose and scope:

Main Network (Mainnet):

- The main network, also known as the production network, is the live and operational Bitcoin blockchain that supports real-world transactions and the exchange of real bitcoins.
- This network is the main focus of miners, developers, traders, and users who are actively participating in the Bitcoin economy.
- Changes made to the main network are permanent and irreversible, making it critical to ensure that any software or protocol changes have been thoroughly tested before being deployed.

Global Testing Network (Testnet):

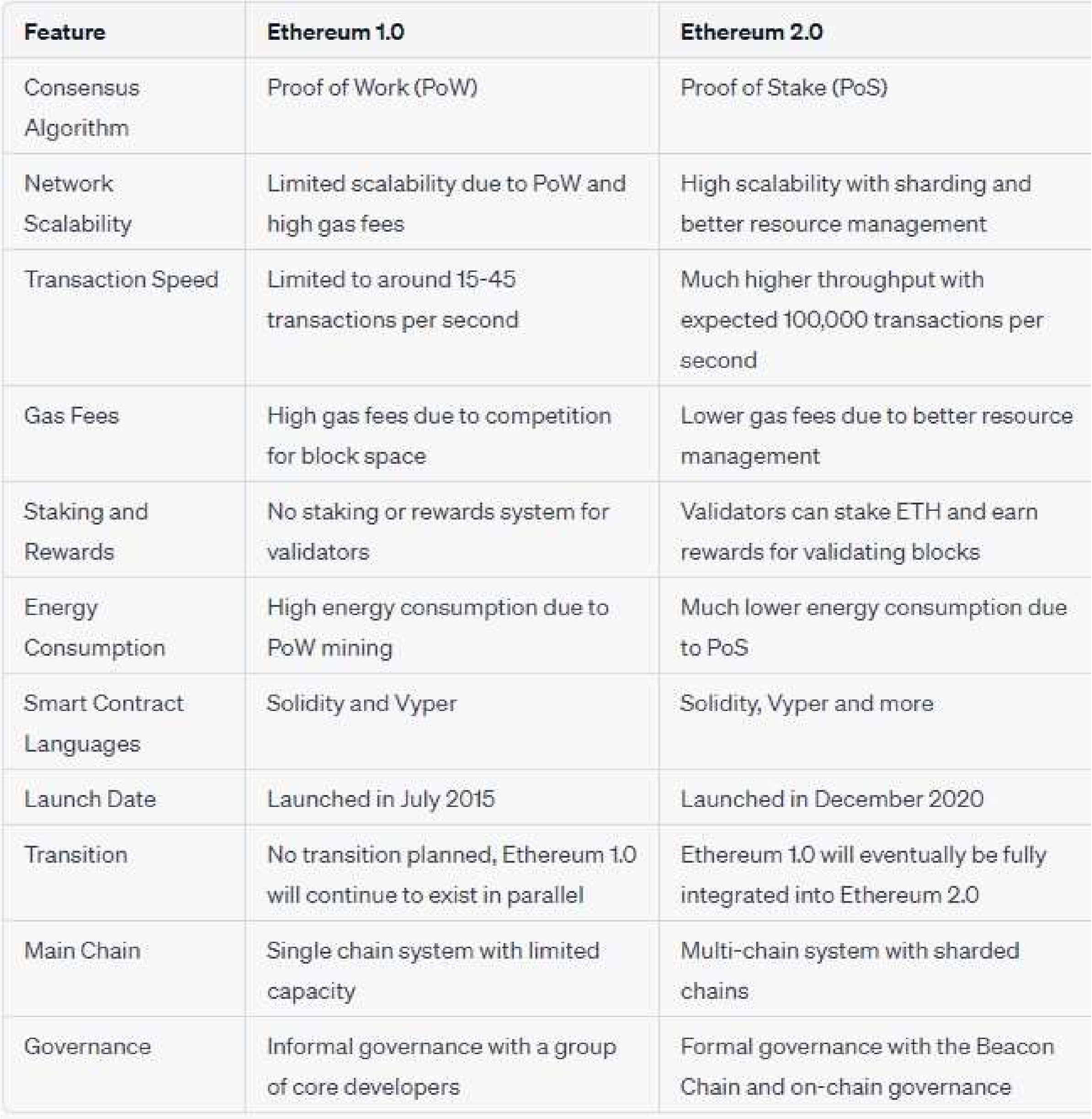
- The global testing network, also known as the public testnet, is a separate blockchain that runs in parallel to the main network, but with its own set of testnet coins.
- The testnet is designed to provide a safe and sandboxed environment for developers to test new features and updates to Bitcoin software without risking the loss of real bitcoins or affecting the main network.
- The testnet is open to the public and can be accessed by anyone interested in testing and experimenting with Bitcoin-related applications.

Local Regression Testing Network:

- The local regression testing network, also known as the private testnet, is a network that runs locally on a developer's computer or a small network of computers.
- This type of testnet is typically used for regression testing, which involves running a suite of automated tests to ensure that new changes to the Bitcoin codebase do not break existing functionality.
- Private testnets allow developers to test and iterate on new features quickly without relying on external networks or third-party services.

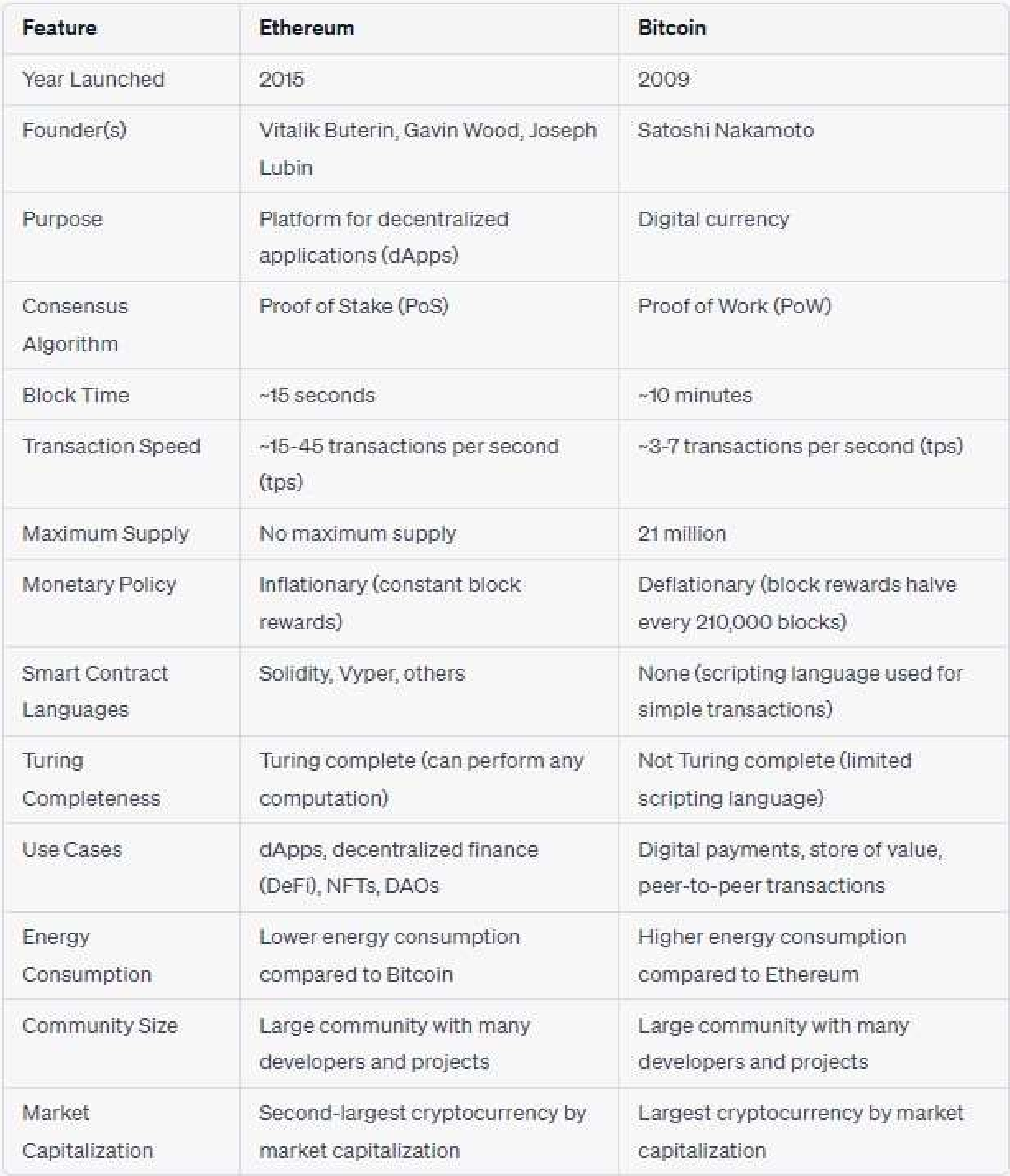
CHAPTER 3: Permissionless blockchain Ethereum

Difference Between Ethereum 1.0 and Ethereum 2.0



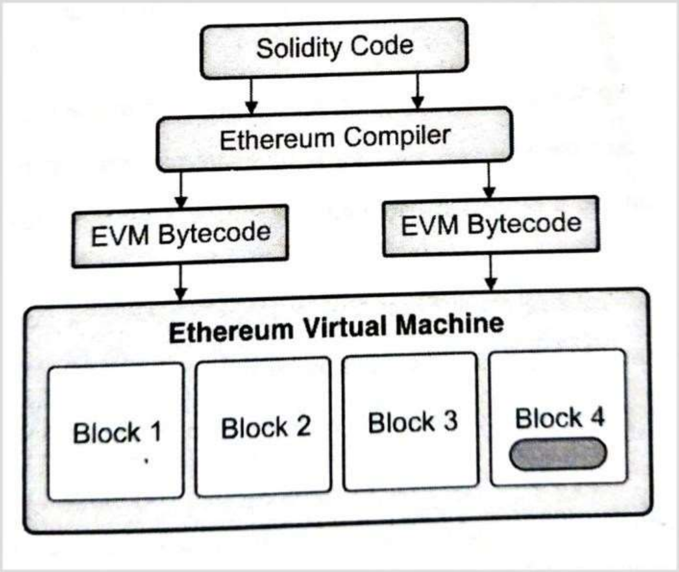
Feature	Ethereum 1.0	Ethereum 2.0
Consensus Algorithm	Proof of Work (PoW)	Proof of Stake (PoS)
Network Scalability	Limited scalability due to PoW and high gas fees	High scalability with sharding and better resource management
Transaction Speed	Limited to around 15-45 transactions per second	Much higher throughput with expected 100,000 transactions per second
Gas Fees	High gas fees due to competition for block space	Lower gas fees due to better resource management
Staking and Rewards	No staking or rewards system for validators	Validators can stake ETH and earn rewards for validating blocks
Energy Consumption	High energy consumption due to PoW mining	Much lower energy consumption due to PoS
Smart Contract Languages	Solidity and Vyper	Solidity, Vyper and more
Launch Date	Launched in July 2015	Launched in December 2020
Transition	No transition planned, Ethereum 1.0 will continue to exist in parallel	Ethereum 1.0 will eventually be fully integrated into Ethereum 2.0
Main Chain	Single chain system with limited capacity	Multi-chain system with sharded chains
Governance	Informal governance with a group of core developers	Formal governance with the Beacon Chain and on-chain governance

Tabular Difference Between Ethereum and Bitcoin



Feature	Ethereum	Bitcoin
Year Launched	2015	2009
Founder(s)	Vitalik Buterin, Gavin Wood, Joseph Lubin	Satoshi Nakamoto
Purpose	Platform for decentralized applications (dApps)	Digital currency
Consensus Algorithm	Proof of Stake (PoS)	Proof of Work (PoW)
Block Time	~15 seconds	~10 minutes
Transaction Speed	~15-45 transactions per second (tps)	~3-7 transactions per second (tps)
Maximum Supply	No maximum supply	21 million
Monetary Policy	Inflationary (constant block rewards)	Deflationary (block rewards halve every 210,000 blocks)
Smart Contract Languages	Solidity, Vyper, others	None (scripting language used for simple transactions)
Turing Completeness	Turing complete (can perform any computation)	Not Turing complete (limited scripting language)
Use Cases	dApps, decentralized finance (DeFi), NFTs, DAOs	Digital payments, store of value, peer-to-peer transactions
Energy Consumption	Lower energy consumption compared to Bitcoin	Higher energy consumption compared to Ethereum
Community Size	Large community with many developers and projects	Large community with many developers and projects
Market Capitalization	Second-largest cryptocurrency by market capitalization	Largest cryptocurrency by market capitalization

Ethereum Virtual Machine



The Ethereum Virtual Machine (EVM) is a software component that runs on the Ethereum blockchain, serving as the runtime environment for smart contracts. It is a virtual machine that executes bytecode instructions, which are generated from high-level programming languages like Solidity, Vyper, and others.

The EVM is a deterministic machine, meaning that given the same input and code, it will produce the same output every time. This is essential for the trust and security of the Ethereum network, as it ensures that all nodes on the network will reach the same conclusion when executing a smart contract.

The EVM is also a stack-based machine, meaning that it uses a stack data structure to store and manipulate data. Instructions are executed by pushing data onto the stack, manipulating it, and then popping it off the stack. The EVM has a variety of instructions for arithmetic, logical, and control flow operations, among others.

One of the most significant features of the EVM is its gas system. Gas is a unit of measurement used to determine the cost of executing a smart contract. Each instruction in the EVM has a gas cost associated with it, which is used to calculate the total cost of executing the contract. This cost is paid by the sender of the transaction that triggers the contract execution, and it helps prevent abuse of the network by limiting the amount of computational resources that can be used.

Ethereum Virtual Machine (EVM) is Turing complete, which means that it is capable of performing any computation that can be performed by a Turing machine, a theoretical computing device that can simulate any algorithmic computation.

The EVM achieves Turing completeness by having a set of instructions that allow for loops, conditional statements, and the ability to jump to different parts of the program. These instructions allow for the implementation of any algorithm, making the EVM capable of executing any smart contract that can be written in a Turing-complete programming language like Solidity or Vyper.

Deterministic Wallet

A deterministic wallet is a type of cryptocurrency wallet that generates a sequence of private keys from a single "seed" value using a predetermined algorithm. This seed value can be used to regenerate all the private keys associated with the wallet. The use of a deterministic algorithm provides some important benefits over non-deterministic wallets, such as greater security and ease of backup and recovery.

Deterministic wallets were first introduced in 2011 with the release of the Hierarchical Deterministic (HD) Wallet standard, also known as BIP32. This standard introduced a hierarchical structure to deterministic wallets, allowing for the creation of an unlimited number of private keys and addresses from a single seed value.

Advantages:

- **Backup and Recovery**: Deterministic wallets require only the seed value to be backed up, making backup and recovery easier and less error-prone.
- **Security:** As the seed value can be used to regenerate all private keys associated with the wallet, deterministic wallets provide a higher level of security compared to non-deterministic wallets.

Disadvantages:

- **Complexity**: Deterministic wallets can be more complex to set up and use compared to non-deterministic wallets.
- **Single Point of Failure**: The seed value is a single point of failure, and if it is lost or compromised, all the private keys associated with the wallet will also be lost or compromised.

Non-deterministic

A non-deterministic wallet is a type of cryptocurrency wallet that generates a unique private key for each address created. The private keys are randomly generated by the wallet software and are not mathematically related to each other. This means that if a private key is lost or stolen, the other private keys and addresses associated with the wallet remain secure.

Non-deterministic wallets are the traditional type of cryptocurrency wallet and were the first type of wallet used for storing cryptocurrencies. Each time a new address is created, a new private key is generated, and the wallet software keeps track of all the private keys associated with the wallet.

Advantages:

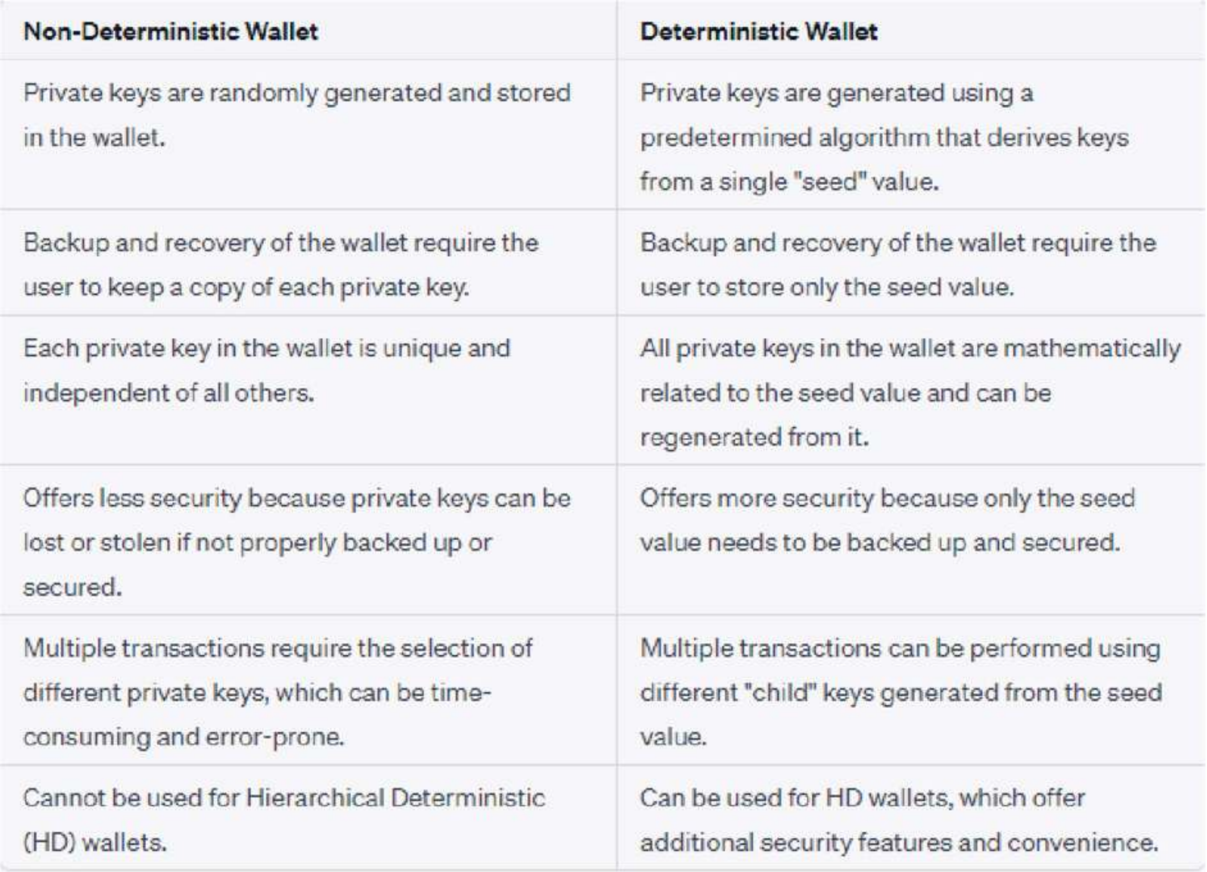
Unique Private Keys: Non-deterministic wallets generate a unique private key for each address created, making them more secure in case one private key is lost or stolen.

Easy to Use: Non-deterministic wallets are generally simpler to set up and use compared to deterministic wallets.

Disadvantages:

Backup and Recovery: With non-deterministic wallets, each private key must be backed up individually, making backup and recovery more time-consuming and difficult compared to deterministic wallets.

Security: As each private key is independent of the others, non-deterministic wallets are more vulnerable to attacks or breaches, as a single compromised private key can put the entire wallet at risk.



Non-Deterministic Wallet	Deterministic Wallet
Private keys are randomly generated and stored in the wallet.	Private keys are generated using a predetermined algorithm that derives keys from a single "seed" value.
Backup and recovery of the wallet require the user to keep a copy of each private key.	Backup and recovery of the wallet require the user to store only the seed value.
Each private key in the wallet is unique and independent of all others.	All private keys in the wallet are mathematically related to the seed value and can be regenerated from it.
Offers less security because private keys can be lost or stolen if not properly backed up or secured.	Offers more security because only the seed value needs to be backed up and secured.
Multiple transactions require the selection of different private keys, which can be time-consuming and error-prone.	Multiple transactions can be performed using different "child" keys generated from the seed value.
Cannot be used for Hierarchical Deterministic (HD) wallets.	Can be used for HD wallets, which offer additional security features and convenience.

Ethereum network Structure

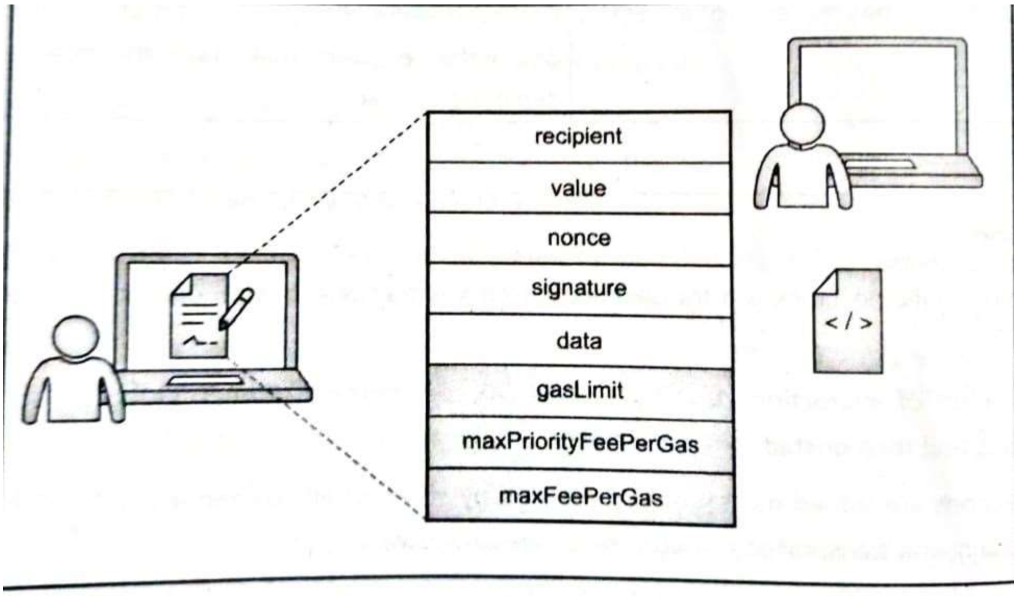


Fig. 3.7.2 : Transaction Structure and Components

The Ethereum network is a decentralized, peer-to-peer network that allows for the creation and execution of smart contracts and decentralized applications (dApps).

The Ethereum network operates on a blockchain, which is a distributed ledger that records all transactions and changes to the network. Each node on the network has a copy of the blockchain, which is updated with every new transaction that is added to the network.

When a transaction is initiated on the Ethereum network, it contains several important pieces of information:

Transaction Nonce:

- The transaction nonce is a unique number that is included in each transaction on the Ethereum network.
- It is used to ensure that transactions are processed in the correct order, so that multiple transactions from the same account cannot be processed out of sequence.
- Each transaction from an account must have a different nonce number.

Transaction Gas:

- Gas is a unit of measurement used to represent the computational work required to process a transaction on the Ethereum network.
- Each transaction requires a certain amount of gas, and the sender must include a gas limit and gas price with the transaction.
- The gas limit is the maximum amount of gas that the sender is willing to use for the transaction, while the gas price is the amount of Ether that the sender is willing to pay for each unit of gas used.
- The total transaction fee is the product of the gas limit and gas price.

Recipient:

- The recipient of an Ethereum transaction can be either an externally owned account (EOA) or a smart contract.
- If the recipient is an EOA, the address is simply the public key associated with the account.
- If the recipient is a smart contract, the address is the contract's unique address on the Ethereum network.

Values and Data:

- An Ethereum transaction can include a value, which represents the amount of Ether being transferred with the transaction.
- In addition, the transaction can include data, which is used to provide additional information or instructions for the transaction.
- For example, if the recipient is a smart contract, the data may include function calls and arguments that will be executed by the contract.

Transmitting Value to EOA:

- To transmit value to an EOA, the sender includes the recipient's Ethereum address in the transaction and specifies the amount of Ether being sent as the transaction value.
- The recipient can then access the Ether in their account using their private key.

Transmitting Value to Contracts:

- To transmit value to a smart contract, the sender includes the contract's Ethereum address in the transaction and specifies the amount of Ether being sent as the transaction value.
- In addition, the transaction may include data that specifies function calls and arguments to be executed by the contract.
- The contract can then access the Ether sent to it using its own code and the Ethereum Virtual Machine (EVM).

Smart contracts

A smart contract is a self-executing digital contract that runs on a blockchain. It is essentially a program that is stored on the blockchain and is capable of automatically enforcing the terms of an agreement between two or more parties. Smart contracts are built using programming languages like Solidity and are executed by the Ethereum Virtual Machine (EVM).

Smart contracts are unique in that they are designed to be tamper-proof and completely transparent. Once a smart contract is deployed on the blockchain, it becomes a permanent part of the blockchain's history and cannot be modified or deleted. This ensures that the terms of the contract are enforced fairly and transparently, without the need for intermediaries or third-party arbitrators.

Smart contracts can be used to automate a wide range of processes and applications, from financial transactions and supply chain management to voting systems and digital identity verification. For example, a smart contract could be used to automate the process of buying and selling a house. The contract could specify the terms of the sale, such as the purchase price, the conditions for the transfer of ownership, and the deadline for payment. Once the terms of the contract are met, the contract would automatically execute, transferring ownership of the house to the buyer and payment to the seller.

Smart contracts are a powerful and innovative technology that has the potential to revolutionize many industries and processes. By enabling secure, transparent, and automatic execution of agreements and transactions, smart contracts can reduce costs, increase efficiency, and improve trust in a wide range of applications.

What is Solidity

Solidity is a high-level programming language that is used to write smart contracts on the Ethereum blockchain. It was developed by the Ethereum Foundation and is designed to be easy to learn and use, while also providing a range of advanced features for developers.

Solidity is a statically-typed language, which means that variables and functions must be defined with a specific data type. It also supports inheritance, which allows developers to reuse code from other contracts, as well as modifiers, which can be used to add additional functionality to functions.

One of the key features of Solidity is its support for contract-oriented programming. This allows developers to define and interact with smart contracts in a way that is similar to traditional object-oriented programming. Solidity also supports events, which can be used to notify other contracts or external applications about changes on the blockchain.

Solidity is a Turing-complete language, which means that it can be used to create complex algorithms and data structures. This makes it possible to write sophisticated smart contracts that can perform a wide range of tasks and interact with other contracts and applications on the blockchain.

Solidity is a powerful and flexible programming language that is essential for developing smart contracts on the Ethereum blockchain. It is widely used by developers around the world and has a growing community of contributors who are working to improve the language and expand its capabilities.

Web3

Web3, also known as Web3.js, is a JavaScript library that provides a simple and easy-to-use interface for interacting with the Ethereum blockchain. It is an essential tool for developers who are building decentralized applications (dApps) that run on the Ethereum network.

Web3.js allows developers to connect to an Ethereum node and interact with the blockchain in a variety of ways, such as sending and receiving transactions, querying data from the blockchain, and interacting with smart contracts. It provides a set of functions and methods that abstract away the complexities of interacting with the Ethereum network, making it much easier for developers to build dApps.

One of the key features of Web3.js is its support for the Ethereum JSON-RPC API. This API allows developers to communicate with an Ethereum node using HTTP requests, which makes it possible to build web applications that interact with the blockchain without requiring users to run a full node.

Web3.js is an open-source project that is maintained by the Ethereum Foundation and a community of contributors. It is constantly evolving and improving, with new features and capabilities being added on a regular basis.

Web3.js is an essential tool for developers who are building dApps on the Ethereum network. It provides a powerful and flexible interface for interacting with the blockchain, and is an important part of the Ethereum ecosystem.

A smart contract is a self-executing program that runs on a blockchain. It is a computer code that is designed to automatically execute the terms of an agreement between parties when certain predetermined conditions are met. Smart contracts have become increasingly popular in recent years, and they are used in a variety of applications, including financial transactions, supply chain management, and voting systems.

The lifecycle of a smart contract can be divided into four stages: development, deployment, execution, and termination. Let's take a closer look at each stage:

Development: The first stage in the lifecycle of a smart contract is development. In this stage, the contract is created and programmed by a developer or a team of developers. The contract is written in a programming language such as Solidity or Vyper, which is specific to the blockchain platform on which it will be deployed. During development, the contract is tested and debugged to ensure that it functions as intended.

Deployment: Once the contract is developed, it is deployed onto the blockchain. Deployment involves uploading the contract code onto the blockchain network and creating a new instance of the contract that can be interacted with by users. The contract is then verified and audited to ensure that it is secure and free from vulnerabilities.

Execution: After the contract is deployed, it becomes available for execution. In this stage, users can interact with the contract by sending transactions to it. When certain predetermined conditions are met, the contract executes automatically and the agreed-upon terms of the contract are enforced. For example, in a financial transaction, the contract might automatically transfer funds from one party to another when a certain condition is met.

Termination: The final stage in the lifecycle of a smart contract is termination. This occurs when the contract reaches the end of its lifecycle or when it is terminated by one of the parties involved. When the contract is terminated, it is removed from the blockchain network and is no longer available for execution.

Remix IDE and Truffle

Remix IDE is an open-source integrated development environment (IDE) for writing, testing, and deploying smart contracts on the Ethereum blockchain. It is a web-based tool that allows developers to write Solidity code, the programming language used to write smart contracts, and test them on a local blockchain before deploying them to the main Ethereum network

Remix IDE provides a user-friendly interface for developers to write and debug Solidity code. It includes features like syntax highlighting, autocompletion, and error highlighting, which make it easier for developers to write code without errors. Additionally, it has a built-in compiler and debugger that help developers to test their contracts on a local blockchain network and catch errors before deploying the contracts to the main Ethereum network.

Remix IDE also provides a variety of tools and plugins that developers can use to enhance their workflow. For instance, it has plugins for code formatting, security analysis, and contract deployment, among others.

Truffle is an open-source development framework used to build, test, and deploy decentralized applications (dApps) and smart contracts on the Ethereum blockchain. It provides developers with a suite of tools and services that simplify the process of building blockchain-based applications.

Truffle

Truffle includes several components, including the Truffle Suite, which is a suite of development tools for Ethereum developers. The Truffle Suite includes Truffle Framework, Ganache, Drizzle, and other tools that enable developers to write smart contracts in Solidity, test them on a local blockchain, and deploy them to the main Ethereum network.

Truffle Framework is the main component of the Truffle Suite. It is a development framework that enables developers to create, compile, and deploy smart contracts on the Ethereum blockchain. Truffle Framework includes a built-in compiler, testing framework, and contract deployment tools that make it easier for developers to create and test their smart contracts.

Ganache is a local blockchain emulator that enables developers to test their smart contracts in a sandboxed environment. It simulates the behavior of the Ethereum network, allowing developers to test their applications without incurring any gas fees.

Drizzle is a front-end library for dApps that integrates with Truffle Framework. It simplifies the process of building and maintaining user interfaces for dApps and provides a range of features, including caching, state management, and event handling.

CHAPTER 4: Permissioned Blockchain: Hyperledger Fabric

Hyperledger Fabric is an open-source enterprise-grade distributed ledger technology (DLT) platform that provides a modular architecture designed to support various use cases for private, permissioned blockchain networks. It is one of the several projects under the Hyperledger umbrella, which is an initiative of the Linux Foundation to support the development of open-source blockchain and distributed ledger technologies for business applications.

Hyperledger Fabric allows multiple organizations to participate in a network, each with their own identities, roles, and permissions, and it provides the ability to define channels, which are private sub-networks within the larger network where specific transactions can be kept confidential between a subset of network participants. It uses a consensus mechanism based on smart contracts, called "chaincode," to execute transactions and enforce business logic.

Hyperledger Fabric provides a high degree of flexibility in terms of architecture, consensus mechanisms, and privacy models, making it suitable for a wide range of use cases, such as supply chain management, trade finance, identity verification, and more. Its features also include support for smart contract development in various programming languages, pluggable consensus mechanisms, and robust access control mechanisms.

Five Blockchain Framework

Hyperledger Iroha: A blockchain framework designed for simple integration into infrastructure projects that require distributed ledger technology. It has a focus on mobile and web application development and provides a simple, modern API.

Hyperledger Fabric: An enterprise-grade blockchain framework that provides a modular architecture designed to support various use cases for private, permissioned blockchain networks. It allows multiple organizations to participate in a network, and it provides the ability to define channels to keep specific transactions confidential between a subset of network participants.

Hyperledger Burrow: A smart contract blockchain framework that provides a permissioned Ethereum virtual machine. It allows the execution of smart contracts on a private blockchain network, and it supports Ethereum's EVM bytecode and Solidity programming language.

Hyperledger Sawtooth: A blockchain framework designed for scalability and modularity. It provides a pluggable consensus mechanism and supports multiple programming languages for smart contract development. It also has a focus on supporting advanced features such as dynamic consensus and parallel transaction processing.

Hyperledger Indy: A blockchain framework designed for decentralized identity management. It provides a decentralized public key infrastructure, credential registry, and verifiable claims system to enable privacy-preserving, self-sovereign identity management.

- **Indy-Node:** Indy-Node It is a node implementation of the Indy ledger. It is responsible for maintaining the ledger state, processing transactions, validating requests, and communicating with other nodes on the network. Indy nodes work together in a decentralized manner to maintain the integrity of the ledger and ensure that all participants have a consistent view of the ledger.
- **Indy-Agent:** Indy-Agent is a software agent that acts on behalf of an individual or organization to interact with other entities on the Indy network. An Indy agent is used to manage an individual's decentralized identity and to facilitate secure and private interactions with other agents on the network. Indy agents can also be used for other purposes, such as managing credentials, verifying claims, and creating and signing transactions.
- **Indy-Ledger:** Indy-Ledger is the distributed ledger at the heart of the Indy ecosystem. It is a tamper-proof, append-only data store that maintains a complete history of all transactions and interactions on the network. The Indy ledger is used to store decentralized identities, credentials, claims, and other data relevant to the management of digital identities. The Indy ledger is designed to ensure privacy, security, and transparency, and to provide a reliable and trustworthy infrastructure for decentralized identity management.

Hyperledger Tools

Hyperledger Composer: It's a tool that simplifies the creation of smart contracts and blockchain applications. It provides a simple way to model business networks and create application logic using popular programming languages like JavaScript.

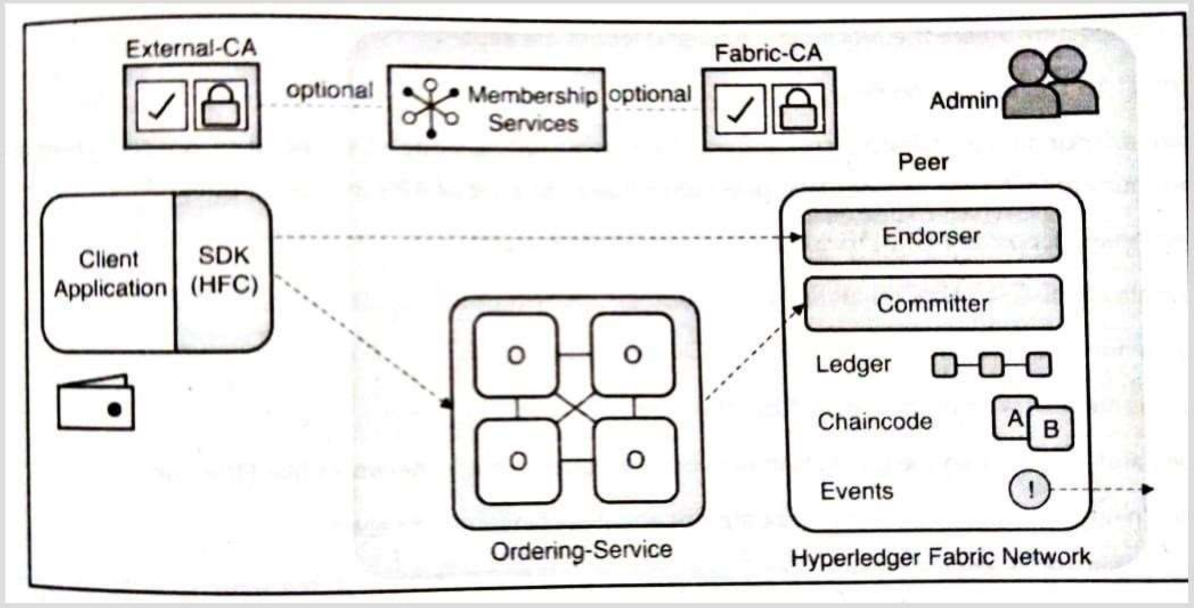
Hyperledger Caliper: It's a benchmarking tool for measuring the performance of blockchain platforms. It provides a set of common benchmarks and allows users to define their custom benchmarks to evaluate the performance of different blockchain platforms.

Hyperledger Cello: It's a tool that simplifies the deployment and management of blockchain networks. It allows users to create and manage blockchain networks using a web-based interface and provides support for various blockchain platforms.

Hyperledger Explorer: It's a tool that provides a visual representation of blockchain networks. It allows users to view the blocks, transactions, and other data stored on a blockchain network in real-time. It can also be used to monitor the health and performance of a blockchain network.

Hyperledger Quilt: It's a tool that provides interoperability between different blockchain networks. It allows different blockchain networks to communicate with each other using the Inter-ledger Protocol (ILP), a protocol for sending payments across different ledgers.

Hyperledger Architecture



Nodes: A node is a participant in the network that stores a copy of the ledger and runs smart contracts called chaincode. Hyperledger Fabric has three types of nodes:

- **Peer nodes**: These nodes store a copy of the ledger and run chaincode. They can endorse and validate transactions, and communicate with other peers and clients.
- **Ordering nodes**: These nodes manage the ordering of transactions into blocks and deliver them to peers for validation and commitment to the ledger.
- **Client nodes**: These nodes submit transaction proposals to the network and interact with the ledger.

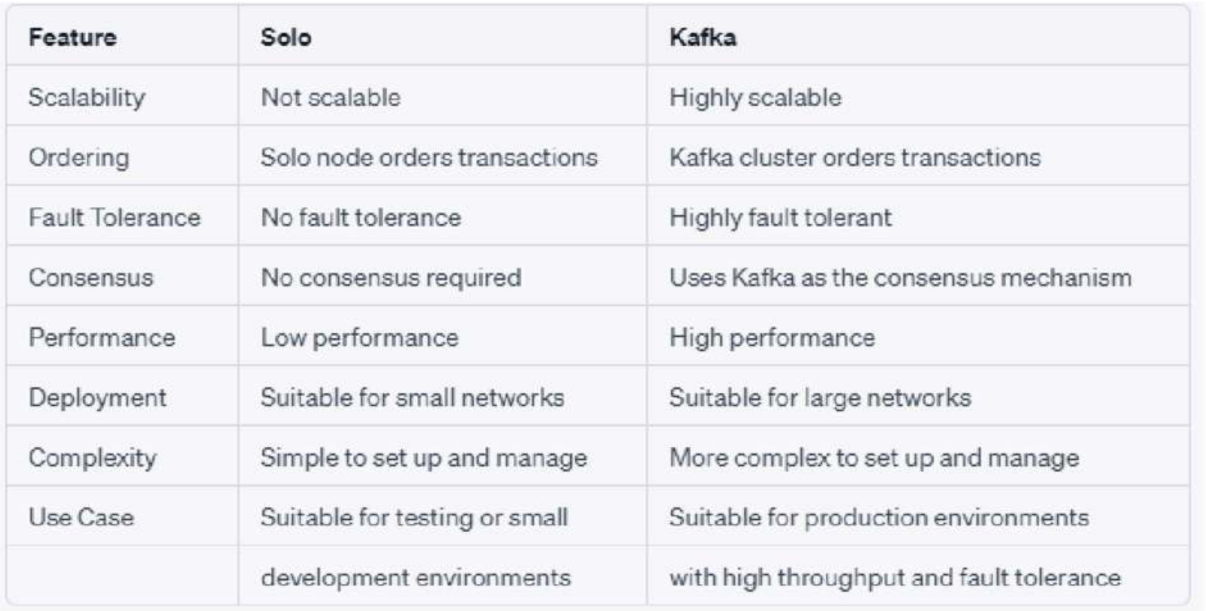
Fabric CA: Fabric Certificate Authority (CA) is a component of Hyperledger Fabric that manages user identities, issues certificates, and enforces access control policies. It is responsible for creating and managing the digital certificates used to authenticate network participants.

Channel: A channel is a private communication pathway between a subset of network participants. It allows different groups of participants to transact privately and securely without interfering with other channels. Each channel has its own ledger and smart contracts.

Membership Service Provider (MSP): MSP is a component of Fabric that manages the identities of participants in the network. It defines the policies for authenticating network participants and grants access to resources based on those policies.

Chaincode: Chaincode is the smart contract that defines the business logic of the network. It is written in a programming language like Go, Java, or Node.js, and is executed on peer nodes. Chaincode can interact with the ledger, submit transactions, and perform other network functions.

Difference Between Solo and Kafka



Feature	Solo	Kafka
Scalability	Not scalable	Highly scalable
Ordering	Solo node orders transactions	Kafka cluster orders transactions
Fault Tolerance	No fault tolerance	Highly fault tolerant
Consensus	No consensus required	Uses Kafka as the consensus mechanism
Performance	Low performance	High performance
Deployment	Suitable for small networks	Suitable for large networks
Complexity	Simple to set up and manage	More complex to set up and manage
Use Case	Suitable for testing or small	Suitable for production environments
	development environments	with high throughput and fault tolerance

RAFT is a consensus algorithm designed for distributed systems that enables a group of nodes to agree on a shared state. It is commonly used in distributed databases, key-value stores, and other distributed systems that require fault tolerance and high availability.

The name RAFT is an acronym for "Replicated Agreed upon Fault Tolerant" and was introduced in a research paper by Diego Ongaro and John Ousterhout in 2014. The RAFT algorithm is designed to be easy to understand, easy to implement, and robust in the face of network failures and node crashes.

In the RAFT algorithm, each node in the network operates in one of three roles: leader, follower, or candidate. The leader is responsible for managing the replication of the shared state among all the nodes. The followers listen to the leader and replicate the shared state. The candidate is a temporary role that nodes assume during the leader election process.

Leader Election: The first step in the Raft consensus algorithm is the election of a leader node. Nodes in the network exchange heartbeat messages to ensure that they are all still online. If a node does not receive a heartbeat message from the leader node within a certain timeout period, it assumes that the leader has failed and initiates a new leader election.

Leader Proposes Transactions: Once a leader has been elected, it begins proposing new transactions to the network. The leader sends the proposed transaction to all other nodes in the network. If a majority of nodes in the network accept the proposed transaction, it is considered committed and added to the blockchain.

Replication: Once a transaction has been committed, the leader node sends the new block to all other nodes in the network for replication. The nodes then validate the block and add it to their copy of the blockchain.

Failure Handling: In the event of a node failure, the other nodes in the network detect the failure and initiate a new leader election. Once a new leader has been elected, the consensus process resumes as before.

Security: To prevent malicious nodes from attacking the network, Raft uses a mechanism called log replication. Each node keeps a log of all transactions and replicates the log to other nodes in the network. If a malicious node attempts to modify the log, other nodes in the network will detect the inconsistency and reject the transaction.

Challenges in blockchain

Interoperability and scalability are two of the biggest challenges facing blockchain technology today.

Interoperability: Interoperability refers to the ability of different blockchain networks to communicate and share data with each other. With the increasing number of blockchain platforms and applications, achieving interoperability has become crucial for the wider adoption and integration of blockchain technology. However, different blockchain networks may use different consensus algorithms, smart contract languages, or data formats, which makes it challenging to establish a seamless communication between them. Interoperability solutions, such as cross-chain bridges and interoperability protocols, are being developed to address this challenge.

Scalability: Scalability refers to the ability of a blockchain network to handle an increasing number of transactions and users without compromising its performance or security. The scalability challenge arises because of the limited processing power and storage capacity of individual nodes in the network. The more transactions that are added to the blockchain, the larger the blockchain becomes, and the more difficult it becomes to process and store new transactions. Various scaling solutions are being developed, such as sharding, sidechains, and off-chain protocols, to improve the scalability of blockchain networks.

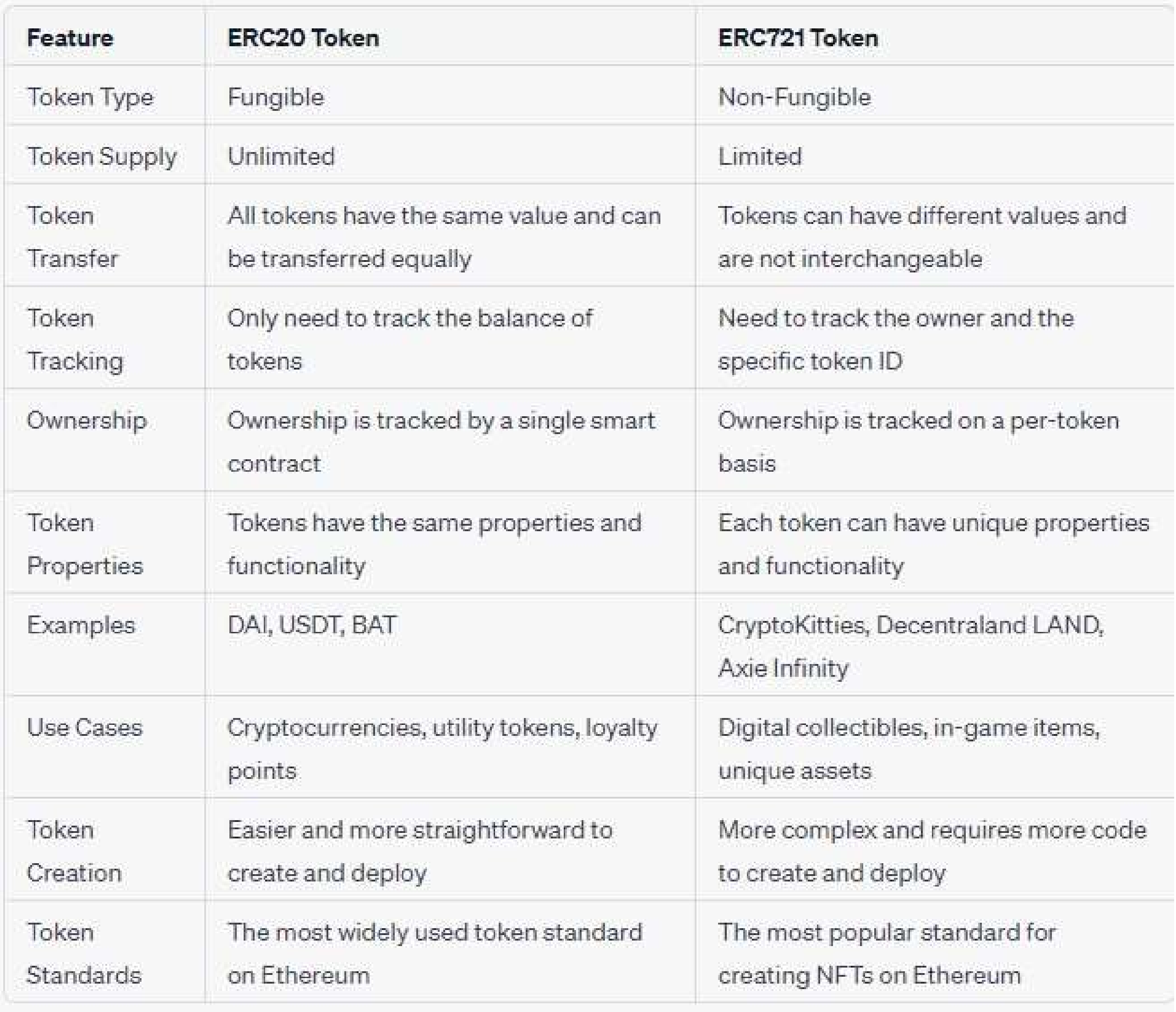
Regulation: The lack of clear and consistent regulations around blockchain technology makes it challenging for businesses and individuals to adopt and use blockchain technology. Regulatory frameworks are needed to ensure that blockchain applications comply with legal and ethical standards.

Adoption: While blockchain technology has shown great potential in various industries, its adoption is still limited. Lack of awareness, limited understanding of blockchain technology, and reluctance to invest in new technologies are some of the factors hindering its adoption.

Security: Blockchain technology is based on cryptography and is considered secure. However, security threats such as hacking, phishing, and other cyber-attacks are still a concern. The development of robust security protocols and measures is necessary to protect blockchain networks and applications from such threats.

CHAPTER 5: Crypto assets and Cryptocurrencies

Difference between ERC20 and ERC721



Feature	ERC20 Token	ERC721 Token
Token Type	Fungible	Non-Fungible
Token Supply	Unlimited	Limited
Token Transfer	All tokens have the same value and can be transferred equally	Tokens can have different values and are not interchangeable
Token Tracking	Only need to track the balance of tokens	Need to track the owner and the specific token ID
Ownership	Ownership is tracked by a single smart contract	Ownership is tracked on a per-token basis
Token Properties	Tokens have the same properties and functionality	Each token can have unique properties and functionality
Examples	DAI, USDT, BAT	CryptoKitties, Decentraland LAND, Axie Infinity
Use Cases	Cryptocurrencies, utility tokens, loyalty points	Digital collectibles, in-game items, unique assets
Token Creation	Easier and more straightforward to create and deploy	More complex and requires more code to create and deploy
Token Standards	The most widely used token standard on Ethereum	The most popular standard for creating NFTs on Ethereum

NFT

NFT stands for Non-Fungible Token, which is a type of digital asset that represents ownership of a unique or rare item. Unlike fungible tokens such as cryptocurrencies that have interchangeable values, NFTs are one-of-a-kind digital assets that cannot be replicated or exchanged for something else.

NFTs are built using blockchain technology, typically on the Ethereum blockchain, and are stored on a decentralized network of computers that ensures their authenticity and ownership. Each NFT has a unique identifier that is recorded on the blockchain, making it impossible to duplicate or modify without the owner's consent.

NFTs can represent a wide range of digital assets, including art, music, videos, virtual real estate, and in-game items. They are typically sold and bought using cryptocurrency, and the ownership and transfer of the NFT are tracked on the blockchain.

NFTs have gained significant attention in recent years due to their potential as a new way for artists, creators, and collectors to monetize and trade digital assets. They offer a new level of ownership and authenticity to digital assets that were previously difficult to monetize or authenticate.

Attributes of NFT

Uniqueness: NFTs are unique digital assets that cannot be replicated or duplicated. Each NFT has a distinct identifier that is recorded on the blockchain, making it a one-of-a-kind asset.

Indivisibility: Unlike fungible tokens, NFTs cannot be divided into smaller parts. They are indivisible and represent a whole digital asset.

Immutability: The ownership and authenticity of an NFT are recorded on a blockchain, making it virtually impossible to modify or tamper with the data.

Scarcity: NFTs can be created with a limited supply, making them rare and valuable.

Interoperability: NFTs can be easily exchanged between different platforms and ecosystems, making them highly versatile and accessible.

Programmability: NFTs can be programmed with smart contracts, enabling creators to set rules around ownership, transfer, and usage of the asset.

Ownership: NFTs provide proof of ownership, giving creators and collectors a new way to monetize and profit from digital assets.

Digital ownership: NFTs allow for the ownership of digital assets, even those that are not physical or tangible, such as digital art, music, or virtual real estate.

Royalties: NFTs can be programmed to include a royalty fee that creators receive each time the asset is resold.

Initial Coin Offering (ICO)

Initial Coin Offering (ICO) is a type of crowdfunding campaign that uses cryptocurrency as a means of raising funds for a new project or startup. In an ICO, a company or project issues a new cryptocurrency or token to the public in exchange for an existing cryptocurrency, typically Bitcoin or Ethereum.

ICOs typically start with a whitepaper that outlines the project's objectives, the technology used, the timeline for development, and the intended use of the funds raised. Investors who are interested in the project can purchase the new cryptocurrency or token using their existing cryptocurrency holdings.

ICOs have gained popularity as a way for startups and new projects to raise funds without the need for traditional venture capital or other forms of financing. They offer a way for investors to get in on the ground floor of a new project and potentially profit from the success of the venture.

However, ICOs are often unregulated, and there have been instances of fraud and scams in the space. As a result, many governments and regulatory bodies have introduced guidelines and regulations for ICOs to protect investors.

Overall, ICOs represent a new and innovative way of raising funds for startups and new projects, but investors should exercise caution and do their due diligence before investing in any ICO.

Phases of ICO

Idea: The first phase of an ICO is the idea generation stage. This is where the founders of the project come up with the concept of the project, identify the problem they aim to solve, and outline their vision for the project.

Team: Once the idea is established, the project founders will begin to assemble their team. They will identify the key roles needed to bring the project to life, such as developers, marketers, and advisors, and recruit individuals with the necessary skills and expertise.

White Paper: The white paper is a comprehensive document that outlines the details of the project, including the problem it solves, the proposed solution, the token economics, the team, the roadmap, and the go-to-market strategy. The white paper is typically shared with potential investors and the wider crypto community to generate interest and gain support.

Token Delivery: Once the project has gained sufficient interest and support, the founders will begin the process of issuing tokens. This involves creating the tokens on the blockchain and distributing them to investors who have contributed to the project.

Token Trading: Once the tokens are distributed, they will be available for trading on cryptocurrency exchanges. The value of the tokens will be determined by market forces, such as supply and demand.

Deliver Project: The final phase of an ICO is the delivery of the project. This involves using the funds raised during the ICO to develop and launch the project, as outlined in the white paper. The success of the project will ultimately depend on its ability to deliver on its promises and meet the expectations of its investors and users.

What is STO

A Security Token Offering (STO) is a fundraising mechanism that allows companies to raise capital by selling digital tokens that represent ownership in the company or its assets. STOs are similar to Initial Coin Offerings (ICOs) but differ in that they are subject to securities regulations and are backed by real assets or equity.

An STO is similar to a traditional IPO in that it allows a company to raise capital from investors, but with several key differences. Firstly, STOs are conducted on a blockchain platform, which allows for greater transparency, security, and efficiency in the issuance and trading of securities. Secondly, STOs are subject to securities regulations, which means that they are required to comply with relevant securities laws in the jurisdiction where they are offered.

The tokens issued in an STO can represent ownership in the underlying asset, such as real estate or commodities, or they can represent equity in the company itself. The tokens can be traded on a secondary market, providing investors with liquidity and the ability to buy and sell their holdings.

However, STOs are subject to complex securities laws and regulations, which can vary by jurisdiction. This requires companies conducting an STO to have significant legal and regulatory expertise to ensure compliance with the relevant laws and regulations. Companies must also ensure that the tokens issued in the STO are properly classified as securities and are registered with the relevant regulatory authorities.

Advantages

- Offering of security tokens are governed internationally
- Regulations cut down on frauds
- Trade in STO takes place on trusted exchange

Disadvantages

- STOs are subject to complex securities laws and regulations, which can vary by jurisdiction. Companies conducting an STO must have significant legal and regulatory expertise to ensure compliance with the relevant laws and regulations.
- STOs are subject to securities regulations, which means that only accredited investors may be able to participate. This can limit the pool of potential investors.

Difference Between Page 5-15 Tech knowledge

There are thousands of different cryptocurrencies in existence, but here are some of the most popular and widely traded ones:

Bitcoin (BTC):

- Bitcoin is the original cryptocurrency, created by an anonymous person or group using the pseudonym Satoshi Nakamoto in 2009.
- It uses a decentralized ledger called the blockchain to record and verify transactions, and has a limited supply of 21 million coins.
- Bitcoin is widely used as a store of value and a means of payment, and has a large and active community of users and developers.

Ethereum (ETH):

- Ethereum is a blockchain platform that was launched in 2015 by Vitalik Buterin and a team of developers.
- It enables developers to create decentralized applications and smart contracts, which are self-executing contracts that can automate complex financial transactions.
- Ethereum has its own cryptocurrency, called Ether (ETH), which is used to pay for transactions on the platform and incentivize developers to build and maintain the network.

Ripple (XRP):

- Ripple is a digital currency that is designed for fast, low-cost international money transfers.
- It is used by banks and other financial institutions to settle transactions, and has partnerships with companies like MoneyGram and Santander.
- Unlike most other cryptocurrencies, Ripple does not use a decentralized blockchain; instead, it uses a network of servers to validate transactions.

Cardano (ADA):

- Cardano is a blockchain platform that was launched in 2017 by IOHK, a research and development company led by Charles Hoskinson.
- It aims to provide a more secure and sustainable way of executing and verifying smart contracts, and uses a proof-of-stake consensus algorithm that is designed to be more energy-efficient than Bitcoin's proof-of-work algorithm.
- Cardano has its own cryptocurrency, called ADA, which is used for transactions and to incentivize network participants.

Dogecoin (DOGE):

- Dogecoin is a cryptocurrency that started as a joke in 2013, based on the popular "Doge" meme.
- Despite its origins, it has gained a large following and has been used for charitable causes and online tipping.
- Dogecoin has a friendly and lighthearted community, and is known for its distinctive branding and mascot (a Shiba Inu dog).