Blockchain Technology

Unit - 2

Bitcoin :- Bitcoin is a revolutionary digital currency that has gained immense popularity and disrupted traditional financial system since its inception in 2009. It is often referred to as a cryptocurrency because it relies on cryptographic technique to secure transaction and control the creation of new units.

Origins of Bitcoin and the whitepaper by Satoshi Nakamoto :

→ The origins of bitcoin can be traced back to a whitepaper titled "Bitcoin : A peer-to-peer Electronic cash system" published by an individual or group using the pseudonym Satoshi Nakamoto in oct. 2008.

Challenges Faced by bitcoin :-

1) Regulatory hurdles :- Govt. and regulatory bodies have struggled to categorize and regulate bitcoin, leading to varying degrees of acceptance and legality in different countries.

2) Scalability :- As bitcoin popularity grew, it faced issue with scalability, resulting in slow transaction processing times and high fees during peak usage periods.

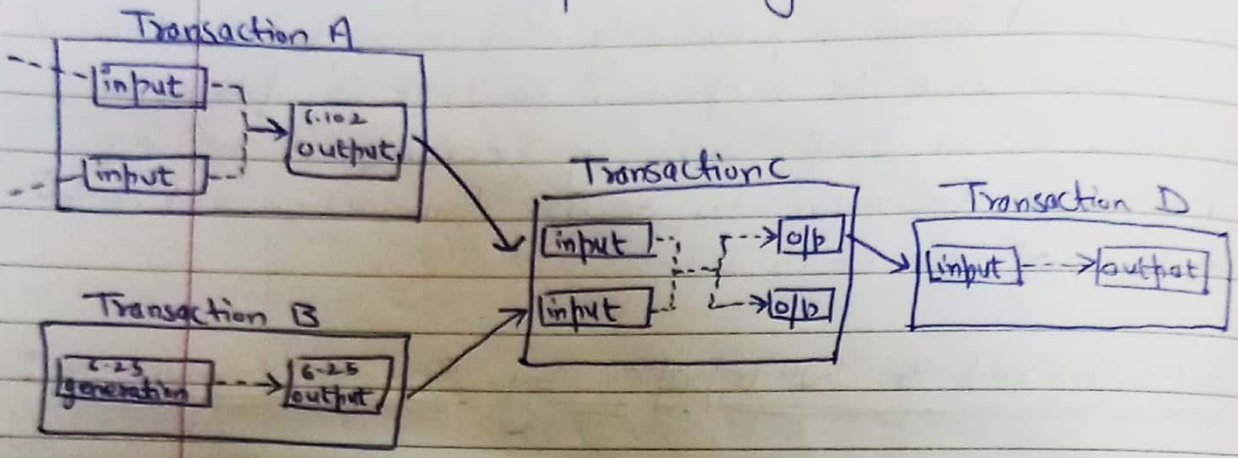
3) Security Concerns :- Despite its cryptographic security, bitcoin exchanges and wallets have been targeted by hackers, leading to the theft of significant amounts of bitcoin.

4) Energy Consumption :- Bitcoin mining, the process by which new coins are created and transactions are confirmed, consumes

Scanned with CamScanner

a substantial amount of energy, leading to concerns about its environmental impact.

Structure and Components of a bitcoin Transaction:



Components :-

1.) Input :- This includes the sender's address, which is also known as the "source" or "from" address. It references the bitcoins being spent and proves ownership through cryptographic signatures.

2.) Output :- The recipent's address, also known as the "destination" or "to address". This specifies where the bitcoins will be sent.

3.) Amount :- The number of bitcoins being transfered in the transaction.

4.) Transaction fee :- Miners are incentivized to include transaction in the blockchain by recieving transaction fees. This is a small amount of bitcoin paid by the sender.

5.) Change output :- Any remaining bitcoin from the input that are sent to the recipent are returned as change to the sender's address.

Scanned with CamScanner

Bitcoin scripting language and its uses :-

→ Bitcoin's scripting language is a unique feature that enables customizable and programmable transactions. It is a simple, stack-based language that defines the conditions under which a transaction can be spent.

Uses :-

1.) Multi-signature wallets :- Script can enforce the requirement of multiple private keys to authorize a transaction, providing enhanced security.

2.) Time-locked transaction :- Bitcoin script allows for the creation of transaction that can only be spent after a specified time or block height.

3.) Escrow service :- Smart contracts can be created using bitcoin script to facilate secure transaction, such as escrow services for online marketplace.

4.) Atomic Swaps :- Scripting enables cross-chain atomic swaps, allowing users to exchange cryptocurrencies without the need for an intermediary.

Ethereum :- Ethereum, introduced by Vitalik Buterin in 2015, is a blockchain platform that extends the capabilities of cryptocurrencies like bitcoin by allowing developers to create decentralized applications (DApps). One of the Ethereum's most notable features is its support for "smart contracts", self-executing agreements with pre-defined rules.

Scanned with CamScanner

Turing Completeness

→ One distinguishing feature of Ethereum's smart contract platform is its Turing completeness. Turing completeness implies that Ethereum's programming language, Solidity allows for the creation of smart contracts that can perform any computation that a turing machine can.

Impact of Turing Completeness on Cryptocurrency :-

1) These companies are responsible for ensuring that a cryptocurrency can be used in everyday transaction because it must be compatible with traditional banking services.
2) A language is Turing complete if it can be used to simulate a turing machine, which means that an appropriately designed program can solve any problem that a universal turing Machine (UTM) can solve.
3) Turing completeness enables the development of highly advanced programs in one language and allows other projects or companies to create highly advanced application using the same tools.

Advantages of Smart Contracts :-

1) Trustlessness :- Smart contracts execute automatically based on predefined rules, reducing the need for trust between parties.
2) Transparency :- Contract terms and execution are recorded on the blockchain, providing an immutable and transparent record of all interaction.

Scanned with CamScanner

3.) Efficiency :- Smart contracts can automate various aspects of contract execution, reducing the need for intermediaries and streamlining process.

4.) Security :- Due to cryptographic verification and the immutability of blockchain records, smart contracts are highly secure against tampering.

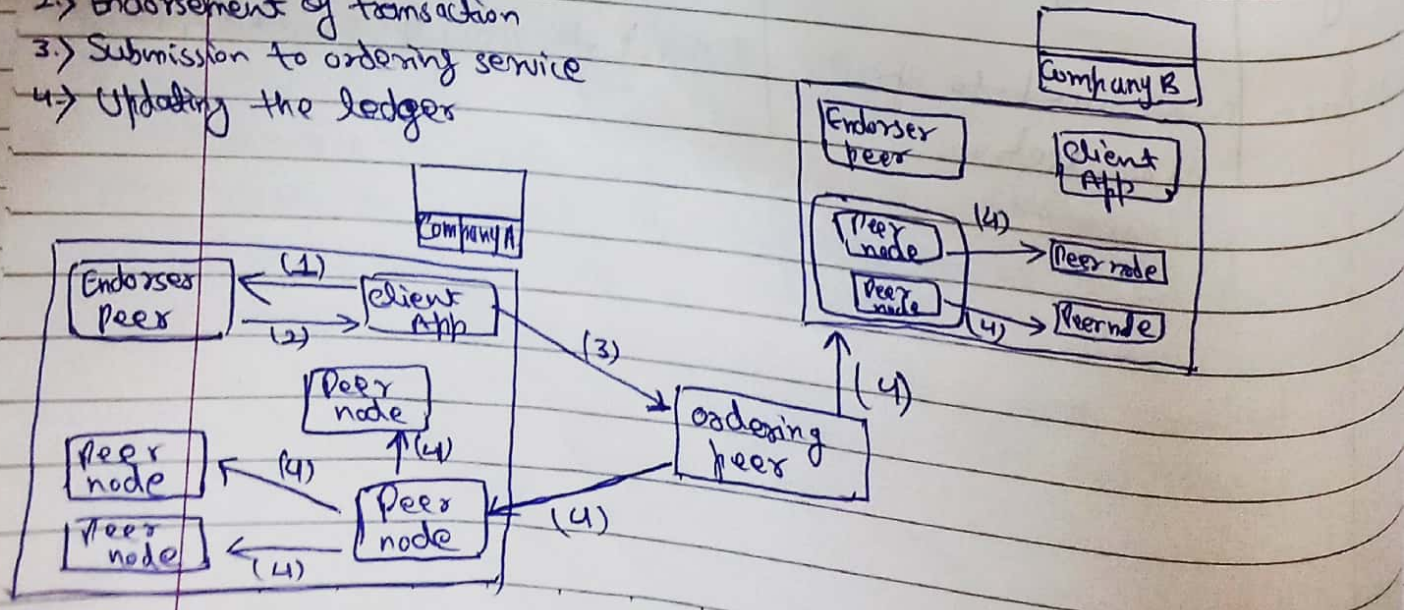
	Smart Contracts	Legal Contracts
Nature	Self-executing code in blockchain.	Written agreements
Enforcement	Automated on blockchain	Requires legal system enforcement.
Trust and Transparency	Blockchain-based, transparent, immutable.	Trust in parties and legal system.
Flexibility	Highly programmable and adaptable.	Typically static, may require renegotiation.
Cost-end Efficiency	Reduces transaction costs, quick execution.	may involve legal fees, time delays.
Jurisdiction	Not tied to specific jurisdiction.	Subject to legal jurisdiction.

Scanned with CamScanner

	Bitcoin Scripting	Ethereum Smart Contracts
Platform	Bitcoin blockchain	Ethereum blockchain
Purpose	Primarily for transfering and storing value.	Versatile for various application.
Programming language	Limited and stack-based script.	Turing-complete programming language.
Functionality	Limited to basic transaction logic.	Can execute complex computation and business logic
Use Cases	Mainly for peer-to-peer value transfer.	Supports a wide range of application, including DeFi
Flexibility	Less flexible due to limited scripting capabilities	Highly flexible and customizable.
State management	Stateless, with no internal storage.	Can maintain internal state and data storage.

Hyperledger Fabric Diagram

1) Creation of the proposal
2) Endorsement of transaction
3) Submission to ordering service
4) Updating the ledger



Scanned with CamScanner

(•) Hyperledger Fabric

→ Hyperledger Fabric is an open source blockchain framework developed under the linux Foundation's hyperledger project. It is designed to offer a highly modular and customizable platform for developing enterprise-grade blockchain applications. Fabric stands out due to its "plug-and-play" architecture which enables organizations to tailor blockchain n/w to their specific needs.

(•) Plug-and-Play platform :-

→ Hyperledger Fabric's "plug-and-play" architecture is one of its key strenghts. It allows organization to select and combine various components to create a blockchain n/w tailored to their requirements. These components include consensus algorithm, membership service, smart contract languages (chaincode) and permissioned n/w settings.

(•) Solidity :-

→ Solidity is a high level, statically-typed programming language specifically designed for writing smart contracts on the ethereum blockchain. These contracts automatically execute and enforce themselves when pre-defined conditions are met. Solidity plays a vital role in enabling the development of decentralized Apps and automating various processes on the ethereum n/w.

Scanned with CamScanner

Key features of Solidity :-

1.) Smart Contract development
→ Solidity allows developers to create complex, self-executing contracts by defining the contract's logic, data structures and functions in the code.

2.) Turing Complete
→ Solidity is turing complete, meaning it can express any computation that can be performed by a turing machine. This flexibility enables developers to build a wide variety of applications.

3.) Security Emphasis
→ Security is paramount when writing smart contracts as any vulnerabilities can lead to significant financial losses.

4.) Static Typing
→ Solidity uses static typing, which means variable types are explicitly defined during contract development.

5.) Smart Contract Deployment
→ After writing solidity code, developers compile it into Ethereum virtual machine (EVM) bytecode, which can then be deployed to the ethereum blockchain.

Scanned with CamScanner

Importance of Solidity in the Etherum Ecosystem :—

1.) Decentralized Applications
→ Solidity enables the creation of decentralized application (dApps) that run on the etherum blockchain. These dApps offer a wide range of services, from decentralized finance (DeFi) to non-fungible tokens (NFT's).

2.) Tokenization
→ The majority of tokens created on Etherum, including the popular ERC-20 and ERC-721 tokens, are built using solidity. Tokenization has transformed fundraising, gaming and digital asset representation.

3.) DeFi and Financial Services
→ Solidity is the foundation for many DeFi protocols, such as decentralized exchange (DEX's), lending platform and yield farming projects.

4.) Smart Contract Auditing
→ Due to the potential financial and security implications of smart contracts, auditing solidity code has become a specialized field.

Scanned with CamScanner

Some important Questions :-

Ques.) List out limitation of hyperledger fabric?

Ans.)

1.) Complexity
→ Hyperledger fabric is known for its complexity, especially in comparison to some other blockchain platforms. It offers a high degree of customization and flexibility but this also means that it can be challenging to set up, configure and maintain the technology.

2.) Scalability
→ While hyperledger fabric provides some level of scalability, it may not be as scaleable as some other blockchain platforms, espicially in public blockchain networks.

3.) Lack of native Cryptocurrency
→ Hyperledger fabric does not have a native cryptocurrency like bitcoin or Ethereum. While this can be an advantage for enterprise use cases where native tokens are not required.

4.) Limited Smart Contract Language Support
→ Hyperledger fabric primarily supports smart contracts written in Chaincode, which is typically implemented in languages like Go and Node.JS.

Scanned with CamScanner

Ques) What are the advantages of Ethereum Smart Contracts?

Ans)

1) Trust and Transparency :- Smart Contracts are stored on the ethereum blockchain, which is a decentralized and tamper proof ledger. This ensures transparency and trust in the execution of the contract.

2) Immutable :- Once deployed on the Ethereum blockchain, smart contracts are immutable, meaning they cannot be altered or tampered with. This immutability ensures that the terms and conditions of the contract remain unchanged throughout its execution.

3) Decentralization :- Ethereum operates on a decentralized n/w of nodes, ensuring that smart contracts are not controlled by any single entity. This decentralization reduces the risk of a single point of failure and makes the system resilient.

4) Automation :- Smart contracts automatically execute predefined actions when specific conditions are met. This automation eliminates the need for intermediaries or trusted third parties, reducing the potential for errors and fraud.

Ques) What is the role of scripting language in bitcoin?

Ans) In bitcoin, scripting language plays a crucial role in defining the condition under which transaction can be spent or executed. The scripting language used in bitcoin is a simplified and intentionally limited scripting language designed to provide flexibility while ensuring security.

1) Defining Transaction conditions

→ Bitcoin transaction involves input and output. The scripting language is used to specify the conditions that must be met to spend a particular output.

Scanned with CamScanner

2.) Security and control :-
→ Scripting allows user to define who has control over a specific set of funds.

3.) Flexibility :-
→ While bitcoin's scripting language is intentionally limited for security reasons, it still offers flexibility. Users can create various types of conditions, including multi-signature wallets, time-locked transaction and more.

Ques.) What is the concept of channel in hyperledger fabric in blockchain technology ? [5 Marks]

Ans.) In hyperledger fabric, a channel is a key concept that plays a crucial role in the permissioned blockchain network's privacy and data segregation. Channels are a way to enable multiple parties to transact privately and securely on the same blockchain n/w.

1.) Privacy and data Segregation

→ In many business scenerios, it is essential to keep certain transaction confidential and restrict access to only relevant participants. Channels address the requirement by allowing different subset of participants within the same n/w to create isolated communication channel.

2.) Multiple ledgers

→ Each channel has its own distributed ledger, separate from the main ledger of the n/w.

Scanned with CamScanner

This means that data and transaction on one channel are not visible or accessible on other channels or the main ledger.

3.) Consensus within channel

→ Each channel has its consensus mechanism and policies which may be different from those of other channels on the same n/w. This allows participants to tailor the consensus process to meet their specific requirement.

4.) Partitioning of Resources

→ Channels also allow for the partitioning of resources, such as compute and storage, to support scalability and performance optimization. Participants in one channel can focus their resources on their specific use case without being affected by the activities of other channels.

5.) Channel Configuration

→ Channel configuration includes defining the set of organization that are members of the channel and specifying the policies and parameters unique to that channel, such as endorsement policies and access control uses.

Ques.) What is the use of smart contract for peer-to-peer lending?

[5 Marks]

Ans.) Smart Contracts in blockchain technology can play a significant role in peer-to-peer (P2P) lending by automating and enhancing various aspects of the lending process.

Scanned with CamScanner

1.) Automated Loan Origination :-

a) Borrowers initiate a loan request by specifying loan details, including the loan amount, interest rate and long term.

b) If the borrower meets the criteria, the smart contract automatically generates a loan agreement with the specified terms.

2.) Interest Calculation and Repayments :-

a) Smart contracts automatically calculate the interest and installment amount based on the agreed-upon terms.

b) Late payments or missed payments trigger pre-defined penalty clauses or automated collection processes.

3.) Trust and Security :-

a) Smart contracts provide trust in the lending process because the terms and condition are predefined and self-executing, reducing the risk of fraud or manipulation.

4.) Global Accessibility :-

→ P2P lending platforms powered by smart contract can facilate cross-border lending, allowing borrowers and lenders from different parts of the world to participate in lending activities.

5.) Lower Operational Costs :-

→ By automating various aspects of the lending process, P2P lending platforms can reduce operational costs associated with intermediaries and manual processes.

Scanned with CamScanner

Ques.) What is the role of peer-to-peer in hyperledger fabric in blockchain technology?

Ans.) In hyperledger Fabric, the term "peer-to-peer" primarily refers to the communication and interaction between peer nodes within the blockchain n/w. In a hyperledger fabric n/w there are different types of peer nodes, including endorsing peers, committing peers and anchor peers each with specified roles in the n/w.

1.) Endorsing peers :-
(•) Endorsing peers play a critical role in the transaction endorsement process.
(•) When a client initiates a transaction proposal, it sends the proposal to a set of endorsing peers.
(•) P2P communication among endorsing peers ensures that all participating organization have the same view of the proposed transaction result.

2.) Committing peers :-
(•) Committing peers are responsible for validating and ordering transaction before adding them to the blockchain ledger.
(•) They receive endorsed transaction from endorsing peers and verify the endorsements.
(•) Once a block is formed, it is distributed to all committing peers for commitment.

3.) Anchor peers :-
(•) Anchor peers serve as the communication end-points for organization participating in a channel.
(•) They play a crucial role in maintaining the cross-organization communication within a channel.

Scanned with CamScanner