1) What is EVM ? Explain characteristics of Ethereum Blockchain ?

A **Ethereum Virtual Machine (EVM):**

→ The Ethereum Virtual Machine (EVM) is a crucial component of the Ethereum blockchain that enables of the Ethereum Blockchain that enables the execution of smart contracts and decentralized applications (DAapps)

What is an Ethereum Virtual Machine (EVM):

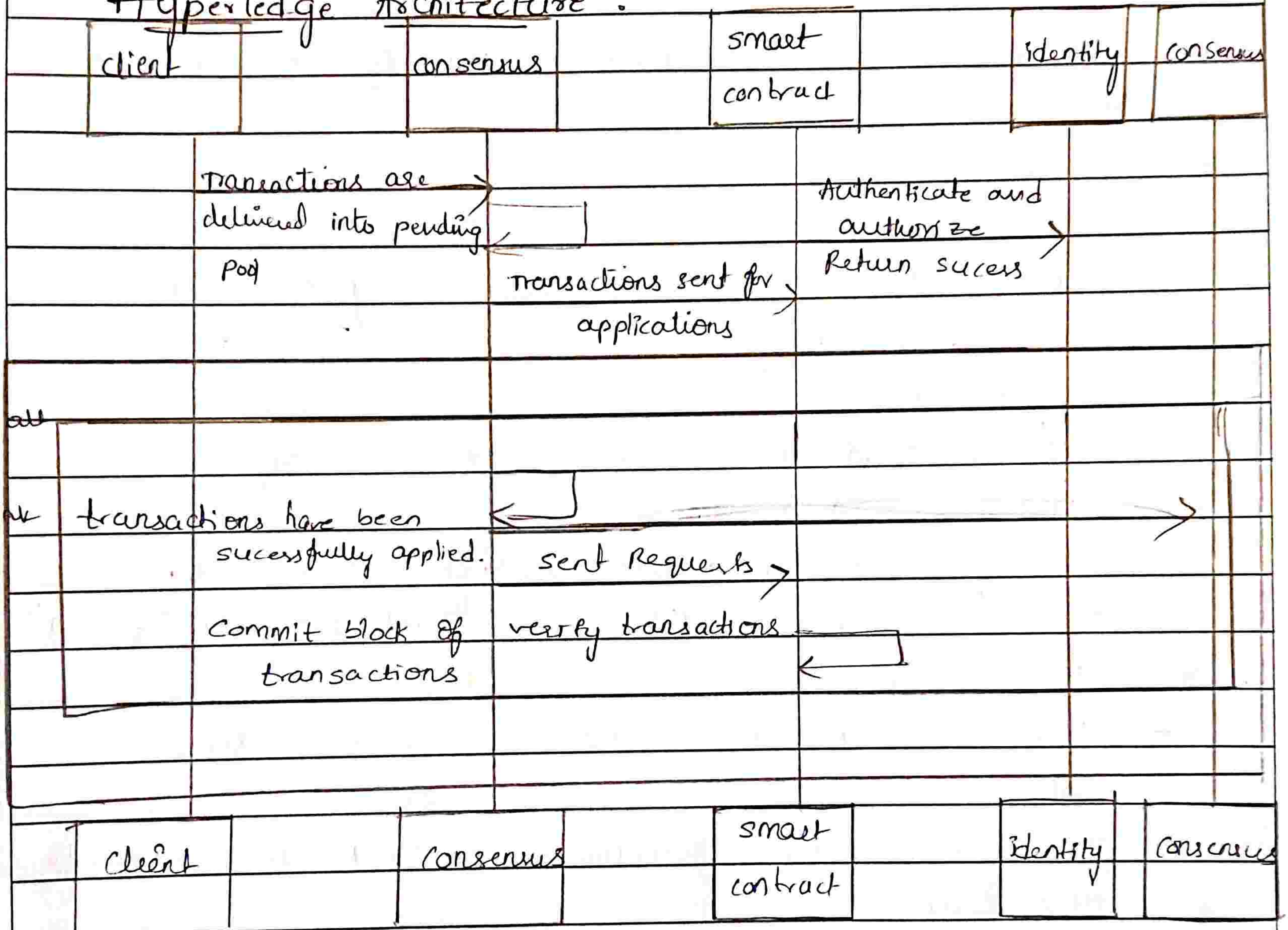
1. Decentralized : The EVM operates on a distributed network of nodes, ensuring that no single entity controls the execution of contracts.
2. Turing completeness : The EVM is a Turing complete, meaning it can execute any computation.
3. Smart contract Execution : When a smart contract is deployed, the EVM Handles the contract's Execution Based on the inputs it's Received.
4. Gas Mechanism : To prevent abuse of resources, every operation in the EVM Requires a certain amount of "gas" to measure computational effort.
5. Isolation : Each smart contract operates in isolation, which means that "execution of one contract doesn't effect the execution of other contract".
6. State Management : The EVM maintains a global state, tracking the current state of all accounts and contracts on the Ethereum network

Characteristics of Ethereum Blockchain:

- smart contracts : Ethereum Allows the creation and deployment of smart contracts. smart contracts are created mainly using a programming language called "solidity". Solidity is an Object oriented programming language that is comparatively easy to learn.
- Ethereum Virtual Machine (EVM): it is designed to operate as a runtime environment for compiling and deploying Ethereum-based smart contracts
- Ether: Ether is cryptocurrency of the Ethereum network.
- Decentralized applications (Daaps): Dapp has its backend code running on a decentralized peer-to-peer network.
- Decentralized autonomous Organization (DAOs): it is a decentralized organization that works in a democratic and decentralized fashion

2) Explain the architecture of Hyper ledger fabric with neat diagram ?

A Hyperledge Architecture :


client
consensus
smart contract
identity
consensus
Transactions are deliverd into pending pod
Transactions sent for applications
Authenticate and authorize
Return sucess
transactions have been sucessfully applied.
sent Requests
Commit block of transactions
verify transactions
Client
Consensus
smart contract
Identity
consensus

1. Consensus layer :

→ Ensures transaction order, accuracy, and agreement across peers

→ communicates with clients/peers via the communication layer

2. smart contract Layer :

→ validates transactions using business logic

→ In Hyperledger fabric, smart contracts are called chaincode (Go, Node.js, java)

3. communication Layer :

→ Manages peer-to-peer communication securely using TLS

→ Ensures transaction delivery and system sychronization.

4. Data Store Abstraction :

→ supports levelDB and couchDB.

→ private data is stored separately with hashes recorded on ledgers

5. Crypto Abstraction :

→ supports flexible cryptographic techniques.

6. identity service :

→ Manages user identities, authentication, and permissions via Certification Authorities (Fabric CA)

7. policy Service :

→ Manages endorsement, consensus, and acess policies

→ policies Ensures governance among network members.

8. API Layer :

→ Admin API : Manages network configurations

→ common API : Manages assets, participants, transactions.

→ Runtime API : supports transaction functions and queries.

9. Interoperation :

→ Enables communication between different blockchain networks to enhance integration.

* Benefits :

- productivity
- IP Management
- Data privacy
- Rich Queries
- Anonymity

* limitations :

- Developer shortage
- Fewer Use cases
- complex Architecture
- Limited APIs
- lower fault Tolerance

3) Discuss the below Libraries and tools of Hyper Ledger fabric a) Indy b) Aries c) Caliper

a) Indy :

purpose : Designed specifically for decentralized identity Management

key Features :

→ Supports self-sovereign identity

→ includes tools, libraries and components to create independent and verifiable digital identities

Use case Example: A university issues a digital degree certificate to a student, which the student can later share with an employer without involving the university again

b) Aries:

purpose: A set of tools and libraries Built on top of indy to facilitate peer-to-peer identity interactions

key features:

→ provides communication protocols for secure and verifiable credential exchange.

→ Acts as a middleware for agents to connect, issue credentials, and establish trust.

Use case Example: Enables two parties to establish trust and exchange verified identity information digitally and securely

c) Caliper:

purpose: A benchmarking tool for evaluating the performance of blockchain implementation

key features:

→ supports multiple blockchain platforms like fabric etc

→ useful for identifying bottlenecks and comparing frameworks under real use case conditions.

use case Example: A company compares fabric and sawtooth performance using caliper before deciding on the best platform for their supply chain solution

- Examples of NFTs :
  - Cryptokitties (2017) : Digital cats on Ethereum blockchain, each unique and able to "reproduce" new kitties ; fans spend over $20 million shortly after launch.
  - OpenSea Categories :
    i) photography (e.g., ocean photos by user erubes1)
    ii) Sports (celebrity art collection)
    iii) Trading cards (for games or collectibles)
    iv) utility NFTs (membership perks)
    v) virtual worlds (digital land, avatar items)
    vi) Art (pixel to abstract pieces)
    vii) collectibles (e.g., Bored Ape Yacht club) .
    viii) Domain names (website ownership)
- Benefits of NFTs :
  - market Efficiency : Direct sales without agents, especially for art and digital assets .
  - investing :
    → Tokenize assets like wine, real estate .
    → Blockchain automates ownership transfers with smart contracts
  - Security :
    → personal information secured via blockchain
    → Enables fractional ownership of physical assets like real estate and art, allowing more people to invest .

4) Discuss Non-fungible Tokens ?

A
- Non-fungible TOkens (NFTs):

→ NFTs are unique digital assets recorded on a blockchain with special IDs and metadata, making them non-interchangeable.

→ They can represent digital (or) real-world items like art, real estate or collectibles

- key Takeaways:

→ NFT's are unique and cannot be replicated.

→ They improve efficienuy in buying/selling assets and reduce fraud.

→ NFT's can represent identities, property rights or ownership stakes

- How NFT's Work:

→ NF's are created (minted) by recording their info on a blockchain. Each mi created NFT's gets a

→ unique ID linked to an owner's address. Smart contracts often manage ownership and transfers.

- Blockchain and fungibility:

→ Cryptocurrencies are fungible (one bitcoin = another bitcoin), but NFTs are non-fungible, that means each is unique, like a digital passport

5) Describe Decentralized autonomous Organization (DAO:)?

A **Decentralized Autonomous Organization (DAO):**

→ A DAO is a decentralized, automated Organization where rules and decisions are encoded in smart contracts. First proposed by Dan Larimer (2015) and refined by Vitalik Buterin (2016), DAOs operate without a central authority and are distributed across millions of computers.

components of DAO :

1) NO central legal entity
2) self-enforcing code
3) Token acts as an incentive for validators.

Need of DAO :

→ DAOs remove the need for trust between online collaborators - trust lies in transparent, verifiable code

DAOs vs Traditional Organization

DAO	Traditional Organization
1. Casting a ballot is needed by individual for any progressions to be implemented	1. Depending on the structure, changes can be requested from the sole party, or casting a ballot might advertised
2. completely democratized	2. usually progressive
3. Administrations offered are taken care of consequently in a decentralized way.	3. Requires Human taking care of, or Halfway controlled mechanization, inclined to control.

Steps for launching a DAO :

1. Create and test smart contracts
2. fund the DAO
3. Deploy it on a Blockchain

smart contract creation   funding   Deployment

DAO Examples :

1. DASH cryptocurrency
2. charities
3. Freelancer networks
4. venture funds

How DAOs Work :

→ Tasks normally Handled by Humans (like HR and finance) are automated by code. Humans (shakeholders) vote on decisions.

DAO Membership Models :

1) Token-based membership : Open and completely permissionless
   Example : MakerDAO
2) Share-based membership : Requires a proposal to join
   Example : MolochDAO

Advantages : Decentralized

- Decentralized Decision making
- Community-driven collaboration
- solves principal-agent problems

Disadvantages of DAO :

- Security
- slow Decision Making
- The Bikeshedding Effect
- No legitimate structure of circulating DAOs