MODULE IV

CHAPTER 4

Permissioned Blockchain: Hyperledger Fabric

University Prescribed Syllabus w.e.f Academic Year 2022-2023

Introduction to Framework, Tools and Architecture of Hyperledger Fabric Blockchain.

Components : Certificate Authority, Nodes, Chain codes, Channels, Consensus: Solo, Kafka, RAFT Designing Hyperledger Blockchain Other Challenges : Interoperability and Scalability of blockchain

Self-learning Topics : Fundamentals of Hyperledger Composer

4.1 INTRODUCTION TO PERMISSIONED BLOCKCHAIN FRAMEWORKS

- Many different companies interested in blockchain technology realized that with the help of blockchain technology they could work together rather than working separately and achieve their goals quickly. With this aim, such companies decide to pool their resources to develop industry standard open-source blockchain technology which anyone can use, and in this way, Hyperledger was created keeping the requirements of an enterprise in mind.
- Hyperledger is an open-source enterprise standard distributed ledger technology (DLT) developed under the flagship of Linux Foundation in 2015. In 2016, Hyperledger accepted code from digital assets, code stream, and IBM, which evolved into an enterprise grade solution Hyperledger Fabric.
- Also, Intel incubated its project called Hyperledger Sawtooth. Different use cases of distributed ledgers can have different requirements, the approach of Hyperledger is to cover an entire spectrum of use cases and apply best practices of distributed systems in blockchain for enterprise solutions. To address this diversity, all Hyperledger projects ensures that the projects must include the following properties:

(1) Modular

- Hyperledger is developing modular, extensible frameworks with common building blocks that can be reused. This approach enables developers to experiment with a variety of components as they evolve and to change individual components without affecting the rest of the system.
- This helps developers to create components that can be combined to build distributed ledger solutions well-suited to different requirements.

(2) Secure

- Security is a key consideration for distributed ledgers due to the involvement of high-value transactions or sensitive data in various use cases.
- Security and robustness are the primary concerns in an enterpriseclass blockchain, and they provide critical infrastructure for next-generation business networks.

(3) Interoperable

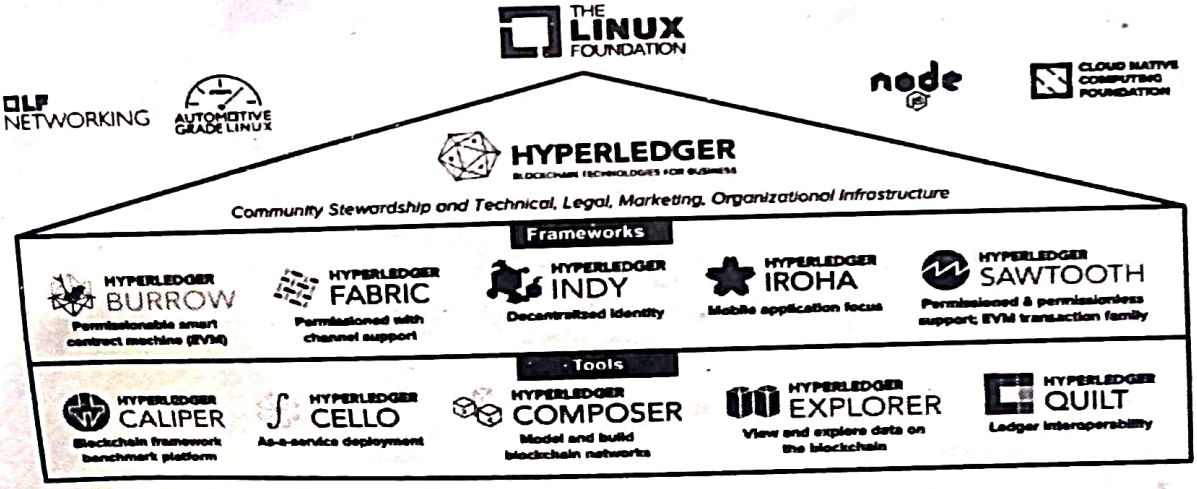
- In future, different blockchain networks will communicate and exchange data to form more complex and powerful networks.
- Hyperledger projects ensure that most smart contracts and applications should be portable across many different blockchain networks, which increases the chances of adoption of Hyperledger technology.

(4) Cryptocurrency agnostic

- Hyperledger exists to create blockchain software for enterprises, not to administer any cryptocurrency. The projects are independent and agnostic of all altcoins, cryptocurrencies, and tokens.
- Hyperledger will not issue its own cryptocurrency, and it may create a token used to manage digital objects.

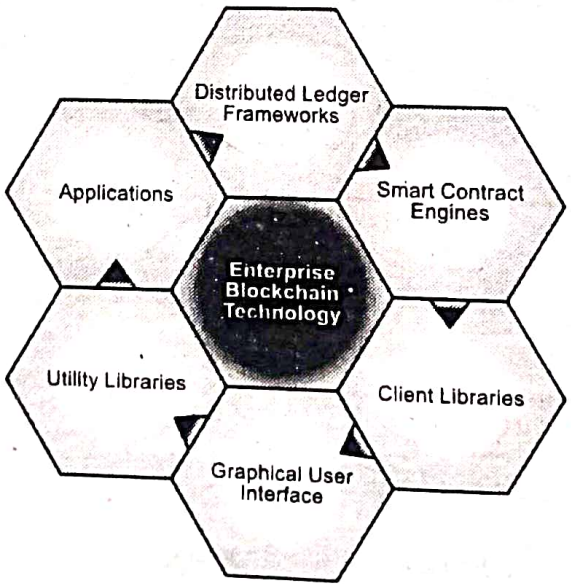
(5) Easy to use

- All Hyperledger projects provide rich and easy-to-use APIs that support interoperability with other systems.
- With these designing principles in mind, Hyperledger provides multiple alternative solutions as reusable modules to interface quickly and easily with Hyperledger's core distributed ledger infrastructure.
- Hyperledger Projects (HLP) are focused to create an enterprise grade, open source distributed ledger framework, and to build an open source technical community to support the ecosystem of HLP solutions.
- Hyperledger also promotes the participation of leading members of the ecosystem including developers, solution providers, and end users. Hyperledger serves as a "greenhouse" that brings together users, developers, and vendors interested in developing and using enterprise blockchains from many different sectors and market spaces. This greenhouse structure incubates new ideas, supports each other with essential resources, and distributes results widely.



(1D2)**Fig. 4.1.1 : Greenhouse Structure of Hyperledger**

- Hyperledger supports many different varieties with optimum consumption of resources. As the greenhouse organization for open-source blockchain development, Hyperledger improves collaboration by creating an environment that streamlines communication, and with better communication, new participants get faster access to necessary information.
  - As newer participants quickly join the collaborative effort, this speeds up development for the benefit of the entire community.
  - To improve productivity, Hyperledger's greenhouse structure encourages specialization where instead of competing with each other and duplicating each other's efforts, specialists are encouraged to work together to accelerate their research and development.
  - These collaborative efforts streamline the development of new projects, and encourage the creation of common components, and it also enhances interoperability and resolves issues between various distributed ledgers due to the better understanding of other projects.
  - The greenhouse structure also helps in the governance of projects and resolutions of dispute and handling of intellectual property.
  - Hyperledger includes a wide range of business blockchain technologies, as shown in Fig. 4.1.2.
  - In Hyperledger, the development of projects encourages the reuse of common modules, enables rapid innovation of components, and promotes interoperability between projects.



(1D3)**Fig. 4.1.2 : Types of Hyperledger Enterprise Blockchain Technologies**

4.2 TOOLS AND ARCHITECTURE OF HYPERLEDGER FABRIC BLOCKCHAIN

4.2.1 Hyperledger Project – Framework

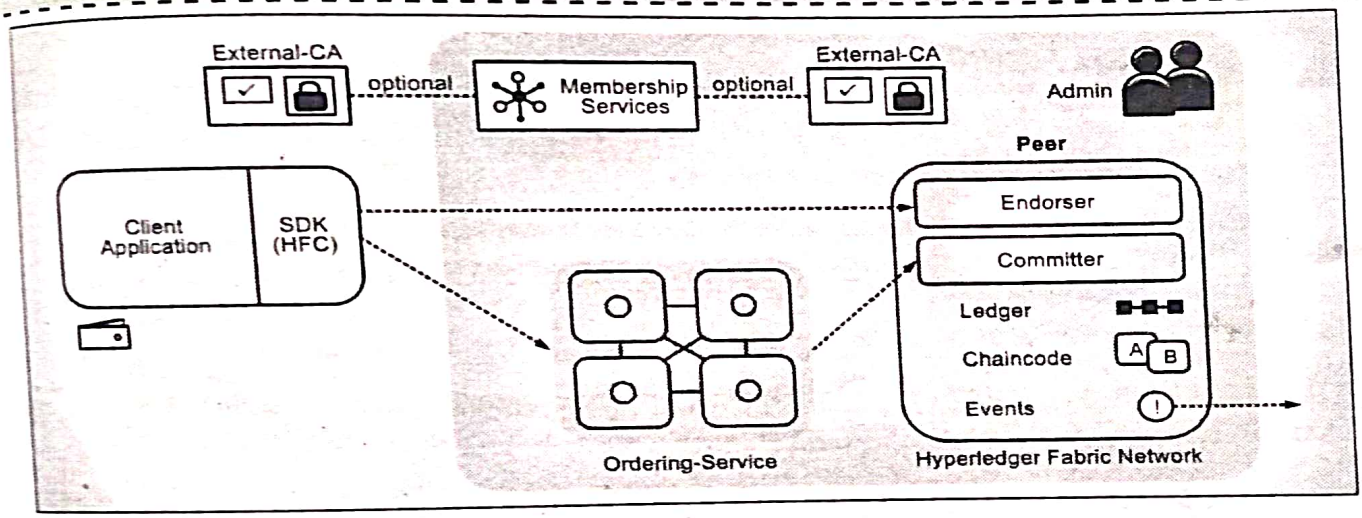
GQ. Describe the structure of Hyperledger.

Currently, there are five different types of distributed ledger frameworks included in the Hyperledger ecosystem that are briefly described below :

(1) Hyperledger Fabric
(2) Hyperledger Sawtooth
(3) Hyperledger Iroha
(4) Hyperledger Burrow
(5) Hyperledger Indy

(1) Hyperledger Fabric

GQ. Explain Hyperledger Fabric V1 architecture.



(1D4)**Fig. 4.2.1 : Hyperledger Fabric V1 Architecture**

- Hyperledger Fabric is a platform for developing distributed ledger solutions, including a modular architecture that delivers high degrees of confidentiality, flexibility, resiliency, and scalability.

- The fabric provides components, such as consensus and membership services, to be plug-and-play. It has container technology to host smart contracts called "chain code" that hold the business rules of the system. Fabric is designed to support pluggable components.
- Architecture of Hyperledger includes various components, shown in Fig. 4.2.1, that work together to execute multiple transactions of users in a peer-to-peer network in a specific order.
- Working of components are briefly explained below.

☞ **Membership Services**

- They provide identity to users using and performing transactions on a blockchain network.
- This identity is recognized in the form of digital certificates. Users will use digital certificates to authorize a user and authenticate transactions through digital signatures.

☞ **Certification Authority (CA)**

- It issues digital certificates to a user, which includes the attributes of standard digital certificates. The certificate can be issued by a traditional external CA or fabric CA.
- Fabric CA is an optional and pluggable module, which may be customized as per the need of an enterprise. These certificates will contain a public key. This key is required for secure communication over the Internet.
- Client application is used to interact with blockchain using Hyperledger Fabric Client SDK. It is used by clients to perform translation on blockchain.

☞ **Admin**

It configures the network, defines the endorsement policy, and manages the blockchain network.

☞ **Ordering Service**

It is run by an organization to arrange a set of transactions submitted by users in order and returns a transaction in order.

☞ **Fabric Network**

- Fabric Network is on the blockchain, an organization running one or multiple peers on the network. Peers record and maintain the ledger, which stores details and states of transections.

- It stores chain codes, which is nothing but smart contracts containing code running on a Hyperledger blockchain network.
- Events can be emitted once the transaction is complete. Peer nodes can perform functions of an endorser, committer, or both.

Endorser

- Endorser executes a chaincode on the network and validates the output. It maintains a copy of the chain code which it executes and verifies the correctness of the output.
- Legitimate transactions are submitted to the ordering service, which returns the order of transactions to be committed, and finally these order transactions are committed by the committer node and recorded on the block. Peers can be either an endorser or a committer or can perform the role of both.
- In fabric, transaction flows from endorsement to ordering and then commit. Client proposes transaction for running chain code and provides input.
- This transaction is submitted to multiple endorsers, which execute the transactions. Endorser executes and validates the received transaction.
- If endorsement policy (e.g., all or more than half or specific endorsers sign and validate the transections) is satisfied, then it is submitted to the ordering services. Ordering service will order the transections and create the blocks.
- Fabric is an extensible blockchain platform for running distributed applications. It supports various consensus protocols; so, it can be utilized in various use cases and trust models.
- In contrast to other blockchain platforms which either require code to be written in a domain-specific language or else rely on a cryptocurrency for running smart contracts, Fabric runs distributed applications written in general-purpose programming languages without any native cryptocurrency.
- Furthermore, Fabric uses a portable notion of membership for the permissioned model, which can be integrated with industry-standard identity management. Fabric maintains such flexibility by considering a novel architectural approach and revamps the way blockchains cope with non-determinism, resource exhaustion, and performance attacks.
- Fabric also includes the notion of channels, which is created to enable a group of participants to form a separate ledger of transactions.
- This is helpful for networks where some participants don't want every transaction, such as in the case of competitors a special price offered to some without informing other participants in the network. If a group of participants forms a channel, only those participants and no others have copies of the ledger for that channel.

(2) Hyperledger Sawtooth

GQ. Explain Hyperledger Sawtooth.

- Hyperledger Sawtooth provides a platform for building, deploying, and running distributed ledgers. Distributed ledgers provide a digital record (e.g., asset ownership).
- The records are maintained without any central authority or implementation. Sawtooth aims to facilitate safe and enterprise use of smart contracts keeping distributed ledgers distributed instead of storing on the central server.
- Sawtooth is highly modular, which enables enterprises and consortiums to select their blockchain applications by themselves.
- Sawtooth contains several technical innovations including :
  - (i) **Dynamic consensus :** Consortiums can change consensus algorithms on a running blockchain simply by issuing a transaction.
  - (ii) **Proof of elapsed time (PoET) :** A consensus algorithm with the scalability of PoW with low power consumption.
  - (iii) **Compatibility :** Transaction families enable users to write smart contract logic in the language of their choice. It can also integrate other smart contract interpreters including Hyperledger Burrow's Ethereum Virtual Machine.
  - (iv) **Parallelism :** Advanced parallel scheduler of Sawtooth splits blocks into parallel flows. Parallelism produces faster block processing to improve the performance of blockchains compared to traditional databases.
  - (v) **Privacy :** Sawtooth nodes can be deployed into clusters with separate permissions. It provides privacy and confidentiality among participants of that distinct chain. There is no exposure or leak of transaction patterns or other confidential information from central services. However, an intermediary Hyperledger Quilt is required to connect separate chains.
- Sawtooth provides scalability, security, privacy, and modular design. Its consensus model PoET boosts scalability, and the transaction families increase the scope of smart contracts and reduce the potential attack surface. Sawtooth also allows trusted execution environments and the roles they play in private transactions.

(3) Hyperledger Iroha

GQ. Describe Hyperledger Iroha.

- Hyperledger Iroha is a blockchain framework designed to be simple and easy to incorporate into projects that require DLT. Iroha is the third distributed ledger platform under Hyperledger and was included in October 2016.

- It was developed by Soramitsu in Japan and was proposed to Hyperledger by Soramitsu, Hitachi, NTT Data, and Colu.
- Features of Iroha are :
  - (i) Simple structure (ii) Domain-driven C++ design
  - (iii) Emphasis on mobile application development (iv) Chain-based BFT consensus algorithm (Sumeragi)
- Iroha follows a different approach from Fabric and Sawtooth. It provides features that are helpful for creating applications for end users.

(4) Hyperledger Burrow

GQ. What is Hyperledger Burrow ?

- Burrow is the fourth distributed ledger platform included within Hyperledger in April 2017. It was developed and proposed to Hyperledger by Monax. Hyperledger Burrow is a permissioned smart contract machine.
- It provides a modular blockchain client with a smart contract interpreter developed partly to the specifications of the EVM. Burrow provides a strongly deterministic, smart contract-focused blockchain design.
- Burrow includes the following components :
  - (i) **Consensus engine :** It maintains the networking stack between nodes, and orders transactions for use by the application engine.
  - (ii) **Application blockchain interface (ABCI) :** It provides the interface specification to connect consensus engine with application engine.
  - (iii) **Smart contract application engine :** It provides a strongly deterministic smart contract engine to developers for operating complex industrial processes.
  - (iv) **Gateway :** It provides programmatic interfaces for system integrations and user interfaces.

(5) Hyperledger Indy

GQ. Explain in brief Hyperledger Indy.

- Hyperledger Indy is a distributed ledger used for decentralized identities. Indy provides tools, libraries, and components for creating and using independent digital identities on blockchains or distributed ledgers. Indy includes reusable components and provides interoperable services across applications, administrative domains, and organizations operating independently and avoids sharing information.
- Indy provides information and verification of data about the other interacting party that enables trusted interactions between enterprises. Using Indy, friends, competitors, and even antagonists can rely upon the shared source of truth and work together.

- Key features of Hyperledger Indy are as follows :
  (i) **Self-sovereignty :** Indy stores identity artifacts on a ledger with distributed ownership such that only authorized users can change or remove identity. These artifacts include public keys, proofs of existence, cryptographic accumulators that enable revocation, etc.
  (ii) **Privacy :** Indy maintains privacy so that every identity owner can operate without creating any correlation risk.
  (iii) **Claim verification :** Identity claims are verifiable from credentials such as birth certificates, driver's licenses, passports, etc. However, these are combined and transformed in powerful ways using zero-knowledge proofs to avail the disclosure of selective data based on the requirement of a specific context.
- This blend of self-sovereignty, privacy, and claim verification is effective and brings lots of potential benefits. Individuals and organizations can benefit from richer and more secure interactions.
- Despite the advanced cryptography under the hood, Indy's API is simple and straightforward. This API includes about 50 C callable functions with idiomatic wrappers for many mainstream programming languages.

4.2.2 Hyperledger Project – Tools

GQ. What are the different types of tools and utility libraries used by Hyperledger ?

Hyperledger projects also include tools and utility libraries. Brief description of various Hyperledger block chain tools are given below :

(1) Hyperledger Caliper

- Hyperledger Caliper is a blockchain benchmark tool used to measure the performance of any blockchain implementation.
- It is a general tool that provides performance evaluations for different blockchain solutions based on a set of neutral and commonly accepted rules.
- Caliper generates reports based on various performance indicators, includes resource utilization, transaction latency, and transactions per second (TPS).
- Caliper community is continuously setting new performance indicators and benchmark use cases.
- Caliper is used as a reference to help choose the blockchain implementation suitable for a company's specific needs. It is an in-house tool, and its result is not published.
- Hyperledger Caliper gives a functioning benchmark tool that can run on multiple Hyperledger frameworks.

(2) Hyperledger Cello

- Hyperledger Cello facilitates an on-demand deployment model to the blockchain ecosystem for quick and easy adoption of blockchain technologies by enterprises.
- It provides automatic creation, termination, and management of blockchains The multitenant chain service of Cello is efficient and can run on top of various infrastructures, including bare metal, virtual machines, cloud platforms like Amazon Web Services (AWS), and container platforms like Docker Swarm and Kubernetes, that boost the efficiency of "Blockchain-as-a-Service" (BaaS).
- Cello provides a real-time dashboard for users to :
  (i) View the status of the blockchain system.
  (ii) See statistics of blockchain events, chain code performance, and system utilization.
  (iii) Manage blockchains by creating, configuring, and deleting.
  (iv) Manage chain code by deploying and uploading a private chain code.
- Hyperledger Cello supports Hyperledger Fabric as the main blockchain implementation. The architecture of most of the components follows the microservice style with pluggable implementations.
- The main programming languages used are Python and JavaScript.

(3) Hyperledger Composer

- Hyperledger Composer provides a toolset for easy and fast creation of smart contracts and blockchain applications to solve business problems.
- The main goal is to facilitate easy integration of blockchain applications with present systems and thus accelerate time-to-value.
- Hyperledger Composer can speed up the development of use cases and deploy a blockchain solution. Composer also enables users to quickly model an existing business network and integrate existing systems and data with blockchain applications.
- A network can contain assets such as tangible or intangible goods, services, or property and transactions related to them.
- As part of the model, users can define the interaction of transactions with assets. Business networks include involved participants with a unique identity across several different business networks.
- Hyperledger Composer supports the existing Hyperledger Fabric blockchain infrastructure and runtime.

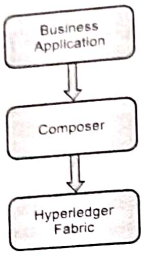


Fig. 4.2.2 : Application development with Hyperledger Composer

(4) Hyperledger Explorer

- Hyperledger Explorer provides a dashboard for viewing blocks, node logs, statistics, smart contracts, transactions, and other information stored in the blockchain.
- Users can query and view complete details of specific blocks or transactions. Explorer can be integrated with any authentication or authorization platforms, either commercial or opensource, to provide the functions appropriate to a user's privileges.
- The characteristics of the explorer are as follows :
  (i) Generic and cross-platform
  (ii) Easy to install, implement, maintain, and extend
  (iii) Supports standard package manager and use of latest technologies
- Explorer currently supports the Hyperledger Fabric framework.

(5) Hyperledger Quilt

- Hyperledger Quilt is an interoperability solution for Hyperledger projects. It provides interoperability between ledger systems with the help of Interledger Protocol (ILP).
- ILP is a simple, open-source enterprise-grade protocol that maintains a global namespace for accounts to make transactions across ledgers. By implementing ILP, Quilt provides :
  (i) A set of rules for enabling ledger interoperability with basic escrow semantics.
  (ii) A standard for a ledger-independent address format and data packet.

(iii) A framework for designing high-level protocols for specific use cases.

- Quilt provides libraries and reference implementations of the core interledger components. This will enable distributed ledger solutions from Hyperledger members, the private ledgers from financial institutions, the wallets from IoT companies, and supply chain systems to connect with one another to perform distributed atomic transactions.

Different use cases

- We are living in the information era where the world is highly interconnected. This connection will be stronger, and we will be closer in the future.
- The data is produced and shared on a high scale, and more collaboration will be needed to solve the common challenges. Hyperledger aims to work on resolving the issues by developing an ecosystem where the organization will collaborate together in a trusted environment, which will ensure security and privacy.
- Hyperledger is exploring blockchain technologies to build innovative products which can provide solutions to different domains. The technologies can be applied to a wide range of domains, and a few use cases are listed below :
  (i) **Banking :** Applying for a loan
  (ii) **Financial services :** Post-trade processing
  (iii) **Healthcare :** Credentialing physicians
  (iv) **IT :** Managing portable identities
  (v) **Supply chain management :** Product traceability
- Hyperledger has useful tools available which can be applied to the use cases listed above and resolve current issues; in some cases, a proof-of-concept has already been developed.

4.2.3 Hyperledger Fabric Architecture

- Hyperledger fabric is an opensource project managed by the Linux Foundation and contributed by IBM. It is intended as a foundation for developing enterprise-grade applications and industry solutions.
- The application developed with Hyperledger does not require cryptocurrency and allows programmable smart contracts, which include the application logic of the system, Hyperledger Fabric leverages containers to host smart contracts also called chaincodes.
- These enterprise applications and solutions are designed with the following features:
  - Modular and Extensible Architecture
  - Pluggable Component

- Key features of Hyperledger Indy are as follows :
  - **(i) Self-sovereignty :** Indy stores identity artifacts on a ledger with distributed ownership such that only authorized users can change or remove identity. These artifacts include public keys, proofs of existence, cryptographic accumulators that enable revocation, etc.
  - **(ii) Privacy :** Indy maintains privacy so that every identity owner can operate without creating any correlation risk.
  - **(iii) Claim verification :** Identity claims are verifiable from credentials such as birth certificates, driver's licenses, passports, etc. However, these are combined and transformed in powerful ways using zero-knowledge proofs to avail the disclosure of selective data based on the requirement of a specific context.
- This blend of self-sovereignty, privacy, and claim verification is effective and brings lots of potential benefits. Individuals and organizations can benefit from richer and more secure interactions.
- Despite the advanced cryptography under the hood, Indy's API is simple and straightforward. This API includes about 50 C callable functions with idiomatic wrappers for many mainstream programming languages.

4.2.2 Hyperledger Project – Tools

GQ. What are the different types of tools and utility libraries used by Hyperledger ?

Hyperledger projects also include tools and utility libraries. Brief description of various Hyperledger block chain tools are given below :

(1) Hyperledger Caliper

- Hyperledger Caliper is a blockchain benchmark tool used to measure the performance of any blockchain implementation.
- It is a general tool that provides performance evaluations for different blockchain solutions based on a set of neutral and commonly accepted rules.
- Caliper generates reports based on various performance indicators, includes resource utilization, transaction latency, and transactions per second (TPS).
- Caliper community is continuously setting new performance indicators and benchmark use cases.
- Caliper is used as a reference to help choose the blockchain implementation suitable for a company's specific needs. It is an in-house tool, and its result is not published.
- Hyperledger Caliper gives a functioning benchmark tool that can run on multiple Hyperledger frameworks.

(2) Hyperledger Cello

- Hyperledger Cello facilitates an on-demand deployment model to the blockchain ecosystem for quick and easy adoption of blockchain technologies by enterprises.
- It provides automatic creation, termination, and management of blockchains. The multitenant chain service of Cello is efficient and can run on top of various infrastructures, including bare metal, virtual machines, cloud platforms like Amazon Web Services (AWS), and container platforms like Docker Swarm and Kubernetes, that boost the efficiency of "Blockchain-as-a-Service" (BaaS).
- Cello provides a real-time dashboard for users to :
  - (i) View the status of the blockchain system.
  - (ii) See statistics of blockchain events, chain code performance, and system utilization.
  - (iii) Manage blockchains by creating, configuring, and deleting.
  - (iv) Manage chain code by deploying and uploading a private chain code.
- Hyperledger Cello supports Hyperledger Fabric as the main blockchain implementation. The architecture of most of the components follows the microservice style with pluggable implementations.
- The main programming languages used are Python and JavaScript.

(3) Hyperledger Composer

- Hyperledger Composer provides a toolset for easy and fast creation of smart contracts and blockchain applications to solve business problems.
- The main goal is to facilitate easy integration of blockchain applications with present systems and thus accelerate time-to-value.
- Hyperledger Composer can speed up the development of use cases and deploy a blockchain solution. Composer also enables users to quickly model an existing business network and integrate existing systems and data with blockchain applications.
- A network can contain assets such as tangible or intangible goods, services, or property and transactions related to them.
- As part of the model, users can define the interaction of transactions with assets. Business networks include involved participants with a unique identity across several different business networks.
- Hyperledger Composer supports the existing Hyperledger Fabric blockchain infrastructure and runtime.

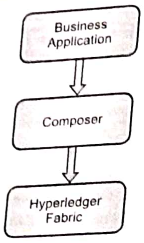


Fig. 4.2.2 : Application development with Hyperledger Composer

(4) Hyperledger Explorer

- Hyperledger Explorer provides a dashboard for viewing blocks, node logs, statistics, smart contracts, transactions, and other information stored in the blockchain.
- Users can query and view complete details of specific blocks or transactions. Explorer can be integrated with any authentication or authorization platforms, either commercial or opensource, to provide the functions appropriate to a user's privileges.
- The characteristics of the explorer are as follows :
  - (i) Generic and cross-platform
  - (ii) Easy to install, implement, maintain, and extend
  - (iii) Supports standard package manager and use of latest technologies
- Explorer currently supports the Hyperledger Fabric framework.

(5) Hyperledger Quilt

- Hyperledger Quilt is an interoperability solution for Hyperledger projects. It provides interoperability between ledger systems with the help of Interledger Protocol (ILP).
- ILP is a simple, open-source enterprise-grade protocol that maintains a global namespace for accounts to make transactions across ledgers. By implementing ILP, Quilt provides :
  - (i) A set of rules for enabling ledger interoperability with basic escrow semantics.
  - (ii) A standard for a ledger-independent address format and data packet.

  - (iii) A framework for designing high-level protocols for specific use cases.
- Quilt provides libraries and reference implementations of the core interledger components. This will enable distributed ledger solutions from Hyperledger members, the private ledgers from financial institutions, the wallets from IoT companies, and supply chain systems to connect with one another to perform distributed atomic transactions.

Different use cases

- We are living in the information era where the world is highly interconnected. This connection will be stronger, and we will be closer in the future.
- The data is produced and shared on a high scale, and more collaboration will be needed to solve the common challenges. Hyperledger aims to work on resolving the issues by developing an ecosystem where the organization will collaborate together in a trusted environment, which will ensure security and privacy.
- Hyperledger is exploring blockchain technologies to build innovative products which can provide solutions to different domains. The technologies can be applied to a wide range of domains, and a few use cases are listed below :
  - (i) **Banking :** Applying for a loan
  - (ii) **Financial services :** Post-trade processing
  - (iii) **Healthcare :** Credentialing physicians
  - (iv) **IT :** Managing portable identities
  - (v) **Supply chain management :** Product traceability
- Hyperledger has useful tools available which can be applied to the use cases listed above and resolve current issues; in some cases, a proof-of-concept has already been developed.

4.2.3 Hyperledger Fabric Architecture

- Hyperledger fabric is an opensource project managed by the Linux Foundation and contributed by IBM. It is intended as a foundation for developing enterprise-grade applications and industry solutions.
- The application developed with Hyperledger does not require cryptocurrency and allows programmable smart contracts, which include the application logic of the system, Hyperledger Fabric leverages containers to host smart contracts also called chaincodes.
- These enterprise applications and solutions are designed with the following features:
  - Modular and Extensible Architecture
  - Pluggable Component

- Permissioned Network
- Scalability
- Security
- Speed
- Interoperability

- The modular architecture uses pluggable components (consensus and membership services, etc.).This flexibility opens the possibility of a wide range of customization of applications to support various enterprise use cases.
- These applications can be scaled to meet a large number of users in a secure way.
- Confidentiality is achieved by creating networks of networks for sharing information only to relevant organizations using channels.
- Completely isolated transactions can be performed to make data private. So, unlike open and permissionless systems, Hyperledger Fabric provides a permissioned, scalable, and secure platform to build solutions that support private transactions and confidential contracts.
- Hyperledger fabric architecture of N participating organization is described in the figure.
- An organization is a business entity participating in a Hyperledger Fabric-based permissioned blockchain network and deployed in the fabric infrastructure. Components of the fabric infrastructure are explained below.

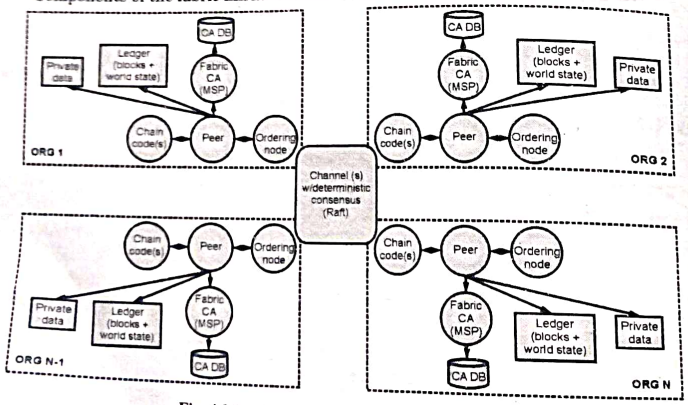


Fig. 4.2.3 : Hyperledger Fabric Architecture

System Architecture

- The distributed network of Hyperledger Fabric includes many peers interacting with each other. Peers hold state and ledger and are also capable of executing transactions using chaincodes. Transactions are then "endorsed" and only endorsed transactions may be committed and have an effect on the state.
- The validating peers run a Byzantine fault tolerance consensus protocol for executing a replicated state machine that accepts three types of transactions as operations:
  - **Deploy transaction :** Takes a chaincode (representing a smart contract) written in Go as a parameter; the chaincode is installed on the peers and ready to be invoked.
  - **Invoke transaction:** Invokes a transaction of a particular chaincode that has been installed earlier through a deploy transaction. Note that the arguments are specific to the type of transaction. The chaincode executes the transaction and indicates whether it succeeded or failed.
  - **Query transaction:** Returns an entry of the state directly from reading the peer's persistent state, this may not ensure linearizability.
- Each chaincode may define its own persistent entries in the state. The blockchain's hash chain is computed over the executed transactions and the resulting persistent state.
- Validation of transactions occurs through the replicated execution of the chaincode and given the faulty assumption underlying BFT consensus, i.e., that among the n validating peers at most $f<n/3$ may "lie" and behave arbitrarily, but all others execute the chaincode correctly.
- When executed on top of PBFT consensus, it is important that chaincode transactions are deterministic; otherwise, the state of the peers might diverge. A modular solution to filter out non-deterministic transactions is implemented in the SIEVE protocol.
- Membership among the validating nodes running BFT consensus is currently static and the setup requires manual intervention.
- The fabric implements a permissioned ledger. It contains a security infrastructure for authentication and authorization. It supports enrolment and transaction authorization through public-key certificates and confidentiality for chaincode realized through in-band encryption.
- To connect with the network, every peer needs to obtain an enrolment certificate from an enrolment CA that is part of the membership services. It authorizes a peer to connect to the network and to acquire transaction certificates, which are needed to submit transactions.
- Transaction certificates are issued by a transaction CA and support pseudonymous authorization for the peers submitting transactions, in the sense that multiple transaction certificates issued to the same peer (i.e., to the same enrolment certificate) cannot be linked with each other.

- Confidentiality for chain codes and states is provided through symmetric key encryption of transactions and states with a blockchain-specific key that is available to all peers with an enrolment certificate for the blockchain. Extending the encryption mechanisms towards more fine-grained confidentiality for transactions and state entries is planned for a future version.
- The Hyperledger Fabric architecture includes various components to facilitate the key blockchain features and provide isolation from the contract development construct.
- Hyperledger fabric architecture is used by a blockchain developer in the following manner:

1. A developer creates the application and smart contract on the network.
2. The developer uses DEPLOY to
   (a) Deploy an app on a server
   (b) Deploy a smart contract on a peer
3. Smart contracts enable every registered user to order
   (a) INVOKE to interact with the app
   (b) QUERY to retrieve information
4. A smart contract can emit an event that the app subscribes.

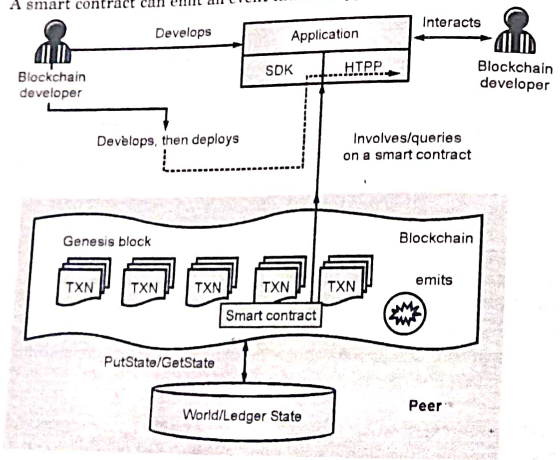


Fig. 4.2.4 : Use of Hyperledger Fabric Architecture

- In the above process, the participating entities or organizations can secretly communicate with each other using channels. Channels create a tunnel between parties engaged in the activities and keep communication isolated from other organizations; in this way, the organization can create and join channels for secretly exchanging information. An organization can be a part of multiple channels and access information or transactions associated with only those channels.

4.3 COMPONENTS OF HYPERLEDGER FABRIC

- Hyperledger fabric components facilitate blockchain features and provide ease of development of the application.
- These components also represent the Hyperledger Fabric infrastructure components. In this section, the components of Hyperledger Fabric and their role and services are explained.
- Fig. 4.3.1 represents the components of Hyperledger Fabric infrastructure. It includes three main key components MSP, ordering service, and peers.

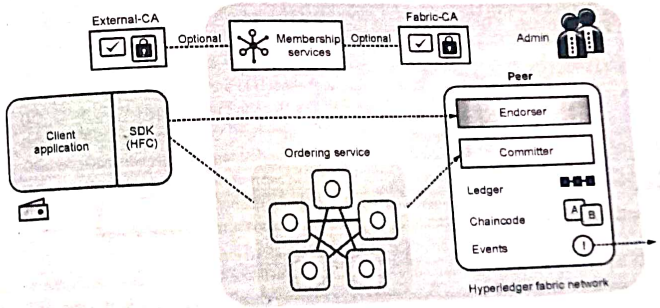


Fig.4.3.1 : Hyperledger Fabric infrastructure components

4.3.1 Membership Service Provider (MSP)

- Hyperledger fabric applications run in permissioned blockchain networks; hence, MSP becomes an essential component of Hyperledger Fabric.
- It provides identity management services to the network participants and allows the registration of unknown entities to join and participate in the network.

4.3.1(A) Certificate Authority

- Hyperledger Fabric CA is the default Certificate Authority component, which issues PKI-based certificates to network member organizations and their users. The CA issues one root certificate (rootCert) to each member and one enrolment certificate (ECert) to each authorized user.
- Certification authority-based membership service can be implemented using the Fabric CA module. Fabric CA can use any X.509-based PKI infrastructure to issue and manage digital certificates.
- The certificate issued by CA provides information about permissions, roles, and attributes for the use of a channel.
- The ability of a user to query or invoke a transaction on any channel depends on the permissions, roles, or attributes maintained in the issued certificate.
- To maintain a high degree of anonymity and unlink ability of transaction, identity mixer-based implementation is used. Identity mixed allows users to transact without revealing their identity and also send multiple transactions without revealing that the transactions are coming from the same user.

4.3.1(B) Nodes

A "Node" is only a logical function in the sense that multiple nodes of different types can run on the same physical server. What counts is how nodes are grouped in "trust domains" and associated with logical entities that control them.

There are three types of nodes:

1. Client	2. Ordering Service	3. Peer

1. Client

A client that submits an actual transaction-invocation to the endorsers, and broadcasts transaction-proposals to ordering service. In short, clients communicate with both peers and the ordering service

2. Ordering Service

- Ordering services create a new block of ordered transactions and then distribute it to peers belonging to a specific channel.
- The ordering service includes a dedicated order node that picks, order, and create a batch of ordered transactions (blocks) and signs to create a hash chain.
- The implementer of the order node provides Atomic Broadcast API and distributes the newly created block to the peers of a specific channel in the network.

This service is pluggable and has various implementations for different needs :

- Solo is used for development testing.
- RAFT is used for production.
- Kafka-based implementation is used for production with high fault tolerance.

3. Peers

- Peers are responsible for executing the smart contract and also stores the ledger.
- A peer who joined the channel can access all the transactions of that channel. Peers can join multiple channels to maintain confidentiality among different organizations.
- To eliminate non-determinism, two types of peers are included in the network:
  - **Endorser peers** only execute a transaction but does not commit it. These uncommitted transactions are picked by an orderer who batches them and forwards them to committer nodes.
  - **Committer peers** get batches of endorsed transactions from an orderer node and performs validation and then commits transactions.

The remaining components help to manage the node and perform network activities such as

- **Client** interacts with hyper ledger fabric blockchain network as per the role, permission, and attributes. These attributes are obtained from the certificate issued and maintained by the specific implementation of CA.
- **Admin** installs the chaincode on the target peers of the network. With the help of the orderer, the network admin instantiates the chaincode on a specific channel. Admin also creates an endorsement policy, which defines which peer can authenticate transaction result before adding it onto the ledger of all the peers on the channel.
- **Ledger** is channels chain and current state data, which is maintained by each peer on the channel. State data is a database that stores the current state of the ledger state, which can be created, updated, and deleted. In the beginning, when a ledger is created, the world state is empty, and as any valid transaction is committed first, the state data is updated first, and then it is updated in the ledger.

4.3.1(c) Chain Codes

Chaincode is installed by the network admin on the peers and channels. It interacts with the ledger shared on the blockchain network.

There are two different types of chaincodes: Nodes

- **System chaincode :** It typically handles system-related transactions such as lifecycle management and policy configuration.
- **Application chaincode :** It manages application states on the ledger, including digital assets or arbitrary data records.

- Smart contracts are defined within a chaincode and a chaincode can include multiple smart contracts. Chaincode acts as a container for a group of smart contracts, and these smart contracts include application domain-specific logic along with endorsement policy provided via chaincode. If the transaction executes and is validated based on the endorsement policy, then the world state changes and results are appended into the ledger.
- **Events** create notifications for important blockchain operations, e.g., addition of a new block and notification related to a smart contract.

4.3.1(D) Channels

- A fabric network can have multiple channels. Channels allow organizations to utilize the same network while maintaining separation between multiple blockchains. Only members(peers) of the channels are allowed to see the transaction created by any member in a channel. In other words, channels partition the network in order to allow transaction visibility for stakeholders only.
- Only the members of the channel are involved in consensus, while other members of the network do not see the transactions on the channel. The peer can maintain multiple ledgers. And peer can be connected to multiple channels.
- Fig. Blue ledger is maintained by P1 and P3, whereas Orange Ledger is maintained by P2 and P4, but black ledger is maintained by P1, P2, P3 and P4.

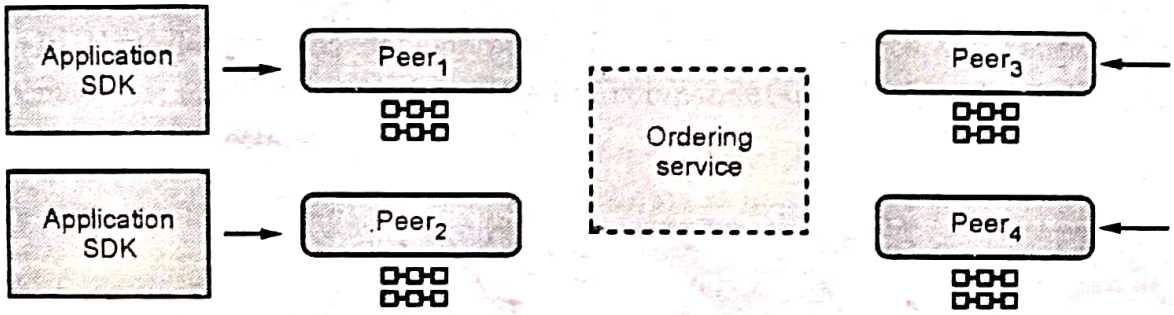


Fig. 4.3.2

- The configuration of the channel is maintained by configtx.yaml file. Using this file we generate channel.tx file and then create a channel using it. Chaincode is installed on all participating peers in a channel, whereas chaincode is instantiated on a channel. A channel contains all the configurations of communication between peers. It holds the list of peers along with who are endorsing, anchor, leader peers. When a client communicates with the network using SDK, the SDK first gets a list of all endorsing peers to which the transaction request needs to send. Using this list the SDK sends transaction requests to peers. Peers that do participate in multiple channels simulate and commit transactions to different ledgers. Orderers are also a part of channels.

4.3.1(E) Consensus : Solo, Kafka, RAFT

In distributed ledger technology, consensus has recently become synonymous with a specific algorithm, within a single function. However, consensus encompasses more than simply agreeing upon the order of transactions, and this differentiation is highlighted in Hyperledger Fabric through its fundamental role in the entire transaction flow, from proposal and endorsement, to ordering, validation and commitment. In a nutshell, consensus is defined as the full-circle verification of the correctness of a set of transactions comprising a block.

- Consensus is achieved ultimately when the order and results of a block's transactions have met the explicit policy criteria checks. These checks and balances take place during the lifecycle of a transaction, and include the usage of endorsement policies to dictate which specific members must endorse a certain transaction class, as well as system chaincodes to ensure that these policies are enforced and upheld.
- Prior to commitment, the peers will employ these system chaincodes to make sure that enough endorsements are present, and that they were derived from the appropriate entities. Moreover, a versioning check will take place during which the current state of the ledger is agreed or consented upon, before any blocks containing transactions are appended to the ledger. This final check provides protection against double spend operations and other threats that might compromise data integrity, and allows for functions to be executed against non-static variables.
- In addition to the multitude of endorsement, validity and versioning checks that take place, there are also ongoing identity verifications happening in all directions of the transaction flow. Access control lists are implemented on hierarchical layers of the network (ordering service down to channels), and payloads are repeatedly signed, verified and authenticated as a transaction proposal passes through the different architectural components.
- To conclude, consensus is not merely limited to the agreed upon order of a batch of transactions; rather, it is an overarching characterization that is achieved as a byproduct of the ongoing verifications that take place during a transaction's journey from proposal to commitment.
    - **SOLO :** It is the Hyperledger Fabric ordering mechanism most typically used by developers experimenting with Hyperledger Fabric networks. SOLO involves a single ordering node. In this, the transactions are ordered in chronological order to form a block.
    - **Kafka :** It is the Hyperledger Fabric ordering mechanism that is recommended for production use. This ordering mechanism utilizes Apache Kafka, an open source stream processing platform that provides a unified, high-throughput, low-latency platform for handling real-time data feeds. In this case, the data consists of endorsed transactions and RW sets. The Kafka mechanism provides a crash fault-tolerant solution to ordering service.

- **RAFT :** Raft is a modern, reliable and relatively less complicated distributed consensus algorithm that is frequently used in modern software solutions such as Consul, etcd, RabbitMQ and so on. Raft is the first step toward Fabric's development of a byzantine fault tolerant (BFT) ordering service.

4.4 OTHER CHALLENGES : INTEROPERABILITY AND SCALABILITY OF BLOCKCHAIN

4.4.1 Interoperability

- Three key factors lead to the problem of blockchain interoperability:
  - **Data Privacy :** Different blockchains feature different levels of data privacy, making it hard to determine which data should be shared and which should not.
  - **Data Security :** Blockchains use different encryption methods, so data cannot be transferred securely from one blockchain to another.
  - **Lack of Standardization :** There is no standard for blockchain interoperability, so it's difficult for companies and developers to understand what standards they need to conform to in order to make their blockchains interoperable.
- That said, the rapid development of blockchain technology ever since Ethereum entered the market with its smart contract capabilities has empowered several projects to unveil unique solutions to address the problem of interoperability. One such solution that is becoming increasingly popular among crypto enthusiasts is cross-chain technology.
- With cross-chain protocols, blockchains sharing similar networks can seamlessly exchange value and information with each other without the need for any intermediaries. But cross-chain applications are now facing yet another challenge: a series of security breaches across multiple cross-chain protocols and decentralized applications (dApps) has rattled the blockchain ecosystem.
- A range of promising solutions has been developed in recent years, most of which have successfully solved the interoperability challenges to an extent. However, the first generation of interoperability projects is rather limited in speed, functionality, and scalability.
- At present, there is no such interoperability solution that facilitates fast, functional, and inexpensive exchange of any digital asset across any chain. At the same time, most decentralized exchanges (DEXs) suffer from their own set of problems, including slow transaction speeds, a limited selection of tokens, and an inferior user experience.
- This, in turn, has translated to momentum for centralized exchanges (CEXs), which are now wielding outsized control over what was supposed to be a "decentralized ecosystem." Additionally, most of the existing CEXs and DEXs are overly focused on listing (and promoting) ERC-20 tokens, which effectively hides other network-based tokens seeking to gain mainstream attention.

- Likewise, over-the-counter (OTC) trading, too, is accompanied by several hurdles. Since all trades are processed through layers of intermediaries (brokers, escrow service providers, smart contract providers, etc.), it leads to unnecessary costs and slow transaction times.

4.4.2 Scalability

- Because each node must validate the transactions executed in the system, scalability is a significant limitation in the blockchain network. As a result, the rate at which a transaction can be processed is limited.
- Researchers are continuing to work on distributed ledger technology based on blockchain-like Hyperledger Fabric to address scalability issues.
- Scalability, security, and decentralisation are the three components of blockchain that must be addressed. You can only determine two of the three attributes at any given time. For example :
  (1) Security and decentralisation can be implemented, but scalability must be compromised. The Bitcoin blockchain is having issues in this area.
  (2) Scalability and security can coexist, but security decentralisation must be sacrificed. BigTable and Cassandra are two examples of such decentralisation compromising functionalities.
  (3) Although decentralisation and scalability can be prioritised, blockchain security will bear the brunt, which is undesirable for many industries. Such characteristics include private chains. Because blockchains use a consensus algorithm that requires all transactions to be verified, there is a limit to the number of transactions that can be completed in a given amount of time. There are solutions available today, such as distributed ledger technology, that allow for more transactions per second. On the other hand, it has a negative impact on the blockchain's transaction rate or speed.
- Scalability is one of the most significant drawbacks of blockchain. Most public blockchain consensus systems that operate in a decentralised manner must strike a balance between minimal network output and a high degree of centralization.
- The following is a definition of the scalability problem:
  (1) Every node on the blockchain network processes every transaction and maintains a copy of the entire state.
  (2) Because inter-node latency grows logarithmically with each additional node, the blockchain network becomes less secure as more nodes are added to its system.
  (3) As the size of the blockchain increases, more storage capacity and computational power are required to ensure that the network works at peak efficiency.
  (4) As the blockchain grows in size, only a few nodes will be able to process a block at some point, posing the possibility of reverting to centralization.

- The challenge of scaling a blockchain can be comes down to two main factors :
  (1) The blockchain itself expands in size as more people utilise it. Each node must keep a more detailed storage history. The bitcoin blockchain alone has a storage of 163 Gigabytes as of March 2018. Every node must store this amount of data, which means that anyone starting a bitcoin node must have at least this much storage space. This figure is only going to rise.
  (2) There is no incentive to be a node that manages the ledger. Miners, who receive a bitcoin block reward for each block they create and add to the chain, are the only people who have an incentive to keep the network running.
- Maintaining a node becomes more difficult and expensive for a single person as a result of these two issues. Miners may run the network, but nodes are in charge of keeping the ledger up to date.
- This creates a challenge for any blockchain; in order to scale, the network must be centralised. Developers, blockchain corporations, and research organisations all have different solutions to offer. Sharding is one solution that is currently being used.
- Each node in the sharding process is not required to handle all of the data. The entire end-to-end blockchain state can be split into separate shards, removing the need for each node to keep all of the data. Specific nodes would only have to save a subset of the state with shards.
- As a result, when transactions are initiated on the blockchain, they are only directed to the shards that they impact.. Rather than processing the entire state, each shard will only process a subset of it. However, in order to automatically establish reliable connectivity across the shards, a commanding mechanism is required.
- Some initiatives are also exploring the possibility of nodes running independently on regular off-the-shelf hardware, delivering complete decentralisation while also addressing the scalability issue.
- Experts are building an all-peer-to-peer network using Shardus (a project). The blockchain will benefit from fast throughput, low latency, and immediate finality, as well as security assurance, as a result of sharding and auto-scaling. This is supported by a proof-of-quorum algorithm and the Shardus Distributed Ledger.
- As the network grows in size, this one addresses the issues of linear scalability and state sharding.
- Some other potential solutions for scalability problem are :
  (1) **Segwit (Bitcoin architecture-specific) :** Segregated witness (Segwit) is a technique for isolating transaction signatures (i.e., "witnesses") from the remaining transaction data, thereby removing some unnecessary weight from the blocks.
  (2) **Increasing the block size (Bitcoin architecture-specific) :** More transactions can be stored in each block if the block size is increased. This lets the network to handle more transactions per second.

(3) **Off-chain state channels :** These state channels include mechanisms that allow interactions to occur within the blockchain rather than outside of it.

(4) **Plasma :** This solution employs a group of smart contracts that run on top of a root blockchain (i.e., the main Ethereum blockchain).

(5) **Off-chain computations :** TrueBit is an example of an off-chain calculating solution that enables scalability transactions across Ethereum smart contracts. TrueBit, like state channels, relies on an external layer of the blockchain to do the heavy lifting. Off-chain computation is a notion in which all difficult mathematical operations are handled by a layer apart from the blockchain. This will not only relieve the Ethereum Blockchain of its strain, but will also lower transaction verification and processing expenses. This computing model is not used by all blockchain nodes.

Instead, only a select few will perform difficult computations in conjunction with a deposit. If the participant's solution is correct, the participant receives both the award and the deposit. Otherwise, the deposit is forfeited. Verifiers are also used to perform verification in off-chain computation. By adding a separate layer to the blockchain, this approach ensures that procedures with slow transaction speeds are executed independently. Transactions can be executed in less than a second by moving them off-chain, while information from public ledgers has a higher level of privacy.

(6) **Proof-of-stake (PoS)** : Instead of computing power, stakeholders vote with money (or, in the case of Ethereum, ether). Bitcoins and Ethereum work on the Pow case, which is a system in which difficult mathematical equations must be solved in order for the verification mechanism and transaction processing to function. As a result, the speed of transactions on Ethereum's blockchain slows. Because PoW has drawbacks, a new proof-of-stake mechanism was developed to achieve faster transactional speeds and higher levels of security.

The other major distinction between the proof-of-work and proof-of-stake models is that in POS, validators do not earn ethers but instead receive a transaction fee for their efforts. PoS will have much faster transaction speeds.

(7) **Transient blockchain :** Saito has come up with a novel solution to both issues. To avoid the growing pains of keeping large ledgers permanently, Saito devised a transitory blockchain that deletes itself at predetermined intervals known as 'Genesis Periods.' A user can keep any information on-chain by paying a small fee, which is effectively the cost of the storage that it uses. The second option presented by Saito is to provide an incentive to nodes that provide storage capacity to increase the amount of data they can store. Nodes with more storage

capacity effectively become miners themselves by having the ability to claim new Saito tokens as they are issued. The more storage capacity they provide, the better their chances of receiving a large sum of money. As a result, this is similar to the probabilistic mechanism used by Bitcoin miners to spend money in order to make money.

- The difference is that Saito, by doing work for the network, assists it in scaling. Although these technologies have brought limited improvements to date, the fundamental constraints are due to present systems' low storage and computing capabilities.
- **Hardware scalability :** The cost of hardware is rising as the popularity of blockchain grows. This raises the cost of entrance into mining, causing it to become more centralised. Aside from that, there are energy and economic concerns.

Chapter Ends...

❑❑❑