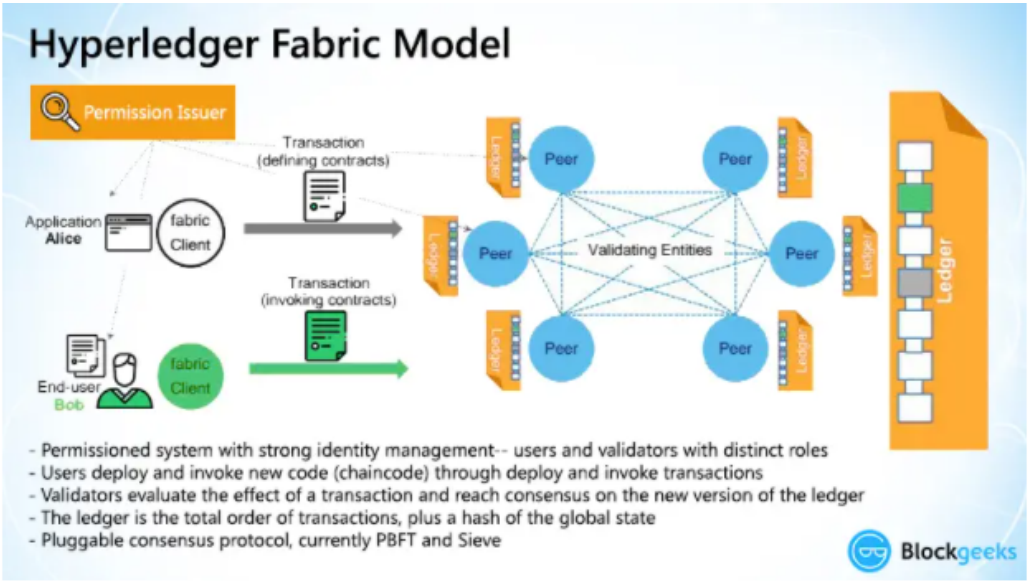
upGrad

# Interview Questions:
# Hyperledger Fundamentals

| | |
|---|---|
| **Q1** | What is a Hyperledger? |
| **Reference** | What Is Hyperledger? [The Most Comprehensive Step-by-Step Guide!] |
| **Ans** | A Hyperledger is a private blockchain network that is created in collaboration with Linux. This network does not support any cryptocurrency, but it has the basic architecture of a blockchain network. It can be used as an enterprise blockchain network to build applications. The architecture of the network can be seen in the image given below: (Image is taken from the reference link)   |

| | |
|---|---|
| **Q2** | What is a Hyperledger Composer? |
| **Reference** | https://www.hyperledger.org/wp-content/uploads/2017/05/Hyperledger-Composer-Overview.pdf |
| **Ans** | A Hyperledger Composer is a set of collaboration tools for building blockchain business networks that enable owners of business and developers to create smart contracts and blockchain applications to solve |

| | |
|---|---|
| | their business problems. Built with JavaScript, by using tools including node.js, npm, CLI and popular editors, Composer offers business centric abstractions as well as sample applications with easy to test devops processes, to make robust blockchain solutions that can drive alignment across business requirements with technical development. With a Hyperledger Composer network, a business person can work with a developer to:<br>1) Define the assets that can be exchanged in a blockchain-based use case<br>2) Define the business rules around whatever transactions are possible<br>3) Define the participants, identity, and access controls for what roles exist and which ones can execute what types of transactions |

| | |
|---|---|
| **Q3** | What is a Hyperledger Sawtooth? |
| **Reference** | Blockchain Technology Project |
| **Ans** | A Hyperledger Sawtooth is a blockchain platform under the Hyperledger umbrella which supports consensus algorithms such as proof of elapsed time (PoET) and practical byzantine fault tolerance (PBFT).<br>This platform separates the application from the core system, so that the smart contracts can create the business rules for applications without the need to know the basic design of the core system. |

| | |
|---|---|
| **Q4** | What is a Hyperledger Indy? |
| **Reference** | Blockchain Technology Projects |
| **Ans** | A Hyperledger Indy gives libraries, tools and reusable components to distribute the digital identities that are created on blockchains or various other distributed ledgers. This is done so that they are interoperable across the administrative domains, applications, and any other silo. Indy can also be used as standalone powering the decentralisation of identity or is interoperable with the other blockchains. |

| | |
|---|---|
| **Q5** | What is a Hyperledger Burrow? |
| **Reference** | Blockchain Technology Projects |
| **Ans** | A Hyperledger Burrow is a single-binary blockchain network distribution that is focussed on speed, developer ergonomics, and simplicity. It supports both ethereum virtual machine (EVM) and webAssembly (WASM) based smart contracts and uses byzantine fault tolerance (BFT) consensus through the tendermint algorithm. It has a well structured event system and can handle a relational database mapping of data on chain. Governance and permissioning is built-in and can be amended by the on-chain proposal transactions. It is optimised for the public permissioned proof-of-stake use cases, but can also be used for private or consortium networks. |

| | |
|---|---|
| **Q6** | What are the various consensus algorithms that are used in the Hyperledger variants? |
| **Reference** | Blockchain Technology Projects |
| **Ans** | The consensus algorithms used by various platforms under Hyperledger are:<br>Proof of stake (PoS), byzantine fault tolerance (BFT), proof of elapsed time (PoET), practical byzantine fault tolerance (PBFT) and many more. |

| | |
|---|---|
| **Q7** | Explain why a Hyperledger is a consortium blockchain. |
| **Ans** | A consortium blockchain is a system that is 'semi-private' and has a controlled user group, but works across different organisations. There are many advantages to consortium systems, and a lot of blockchain platforms are setting themselves up as the backbone for these cross-company solutions. The goal of a Hyperledger is to provide blockchain support for enterprises so it has to be a consortium blockchain. |

upGrad

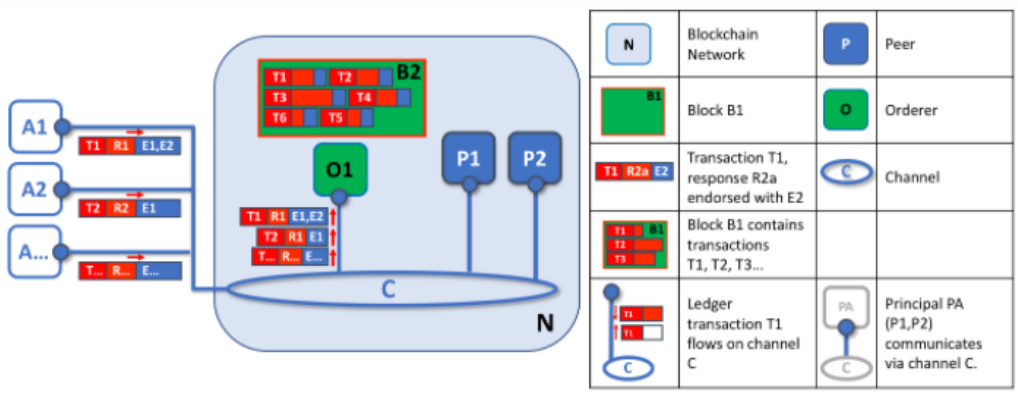
| | |
|---|---|
| **Q8** | What is the consensus algorithm that is used in a Hyperledger Fabric? |
| **Reference** | https://www.skcript.com/svr/consensus-hyperledger-fabric/ |
| **Ans** | The Hyperledger Fabric consensus is divided into three phases: Endorsement, ordering, and validation.<br>• The participants endorse a transaction, and this is the policy that defines the endorsement phase.<br>• The endorsed transaction will then move to the ordering phase. In this phase, the order of the transaction is to be committed to the ledger.<br>• The final phase of validation takes the block of transactions and the correctness of the block is finally validated. |

| | |
|---|---|
| **Q9** | What are the pros and cons of using a public blockchain for businesses? |
| **Reference** | Pros and Cons of Hyperledger Fabric for Blockchain Networks<br>Issues with Hyperledger Fabric |
| **Ans** | The pros of a public blockchain are as follows:<br>- No need for trust as there are a lot of participants.<br>- It is more secure due to the huge number of nodes participating in the network.<br>- A public network is quite transparent.<br>The cons of a public blockchain are as follows:<br>- The speed of the network can be less and can affect the business transactions.<br>- There are a lot of scalability issues with the public blockchain network.<br>- Since the network is public, anyone can view the network which may not be favourable for a business application. |

| | |
|---|---|
| **Q10** | What is an ordering service? What are the different types of ordering services available in the Hyperledger Fabric? |

| | |
|---|---|
| **Reference** | https://hyperledger-fabric.readthedocs.io/en/release-1.4/orderer/ordering_service.html |
| **Ans** | In Hyperledger Fabric, a set of nodes collaborated together to perform the ordering of the transactions. These nodes are called the ordering service. The types of ordering services in Fabric are: Raft, solo and kafka. |

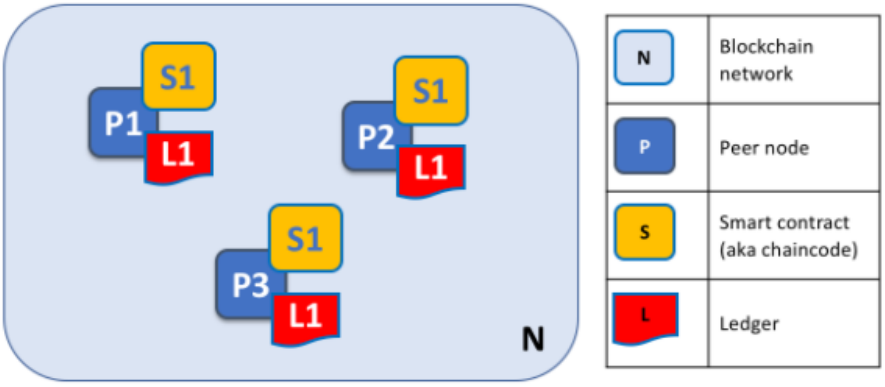
| | |
|---|---|
| **Q11** | What is a transaction flow? |
| **Reference** | https://hyperledger-fabric.readthedocs.io/en/release-1.4/orderer/ordering_service.html#orderers-and-the-transaction-flow |
| **Ans** | The transaction flow can be seen in the image below: (Taken from the reference link) |




*The first role of an ordering node is to package proposed ledger updates. In this example, application A1 sends a transaction T1 endorsed by E1 and E2 to the orderer O1. In parallel, Application A2 sends transaction T2 endorsed by E1 to the orderer O1. O1 packages transaction T1 from application A1 and transaction T2 from application A2 together with other transactions from other applications in the network into block B2. We can see that in B2, the transaction order is T1,T2,T3,T4,T6,T5 – which may not be the order in which these transactions arrived at the orderer! (This example shows a very simplified ordering service configuration with only one ordering node.)*

| | |
|---|---|
| **Q12** | What is a Raft? |

| | |
|---|---|
| **Reference** | https://hyperledger-fabric.readthedocs.io/en/release-1.4/orderer/ordering_service.html#raft |
| **Ans** | Raft is a type of ordering service that is available in the Hyperledger Fabric. This follows a leader follower mechanism, where the leader is elected and the decisions taken by the leader are replicated by the follower nodes. |

| | |
|---|---|
| **Q13** | What is a peer? |
| **Reference** | https://hyperledger-fabric.readthedocs.io/en/release-1.4/peers/peers.html |
| **Ans** | A peer in the Hyperledger Fabric network is its basic entity. Peers host ledgers and smart contracts.<br>They can be visualised through the following image:<br>  <br>*A blockchain network is comprised of peer nodes, each of which can hold copies of ledgers and copies of smart contracts. In this example, the network N consists of peers P1, P2 and P3, each of which maintain their own instance of the distributed ledger L1. P1, P2 and P3 use the same chaincode, S1, to access their copy of that distributed ledger.* |

| | |
|---|---|
| **Q14** | Is a Hyperledger private permissioned or public permissioned? |
| **Ans** | Hyperledger is a private permissioned network. |

| | |
|---|---|
| **Q15** | Where can you use a Hyperledger? |
| **Ans** | The Hyperledger tools or blockchain implementations are meant for enterprise applications, where privacy and permissions are very important. |

| | |
|---|---|
| **Q16** | What are the main advantages of a Hyperledger over traditional databases? |
| **Reference** | https://stackoverflow.com/questions/40397614/how-is-a-managed-permissioned-blockchain-different-from-a-relational-database |
| **Ans** | The value proposition of permissioned blockchain systems over traditional databases is simple: **integrity through cryptographically signed history**. When it comes to maintaining a shared database between entities with imperfect trust, the permissioned blockchains have some great additional features. They are as follows:<br>• The database can contain application logic in the form of constraints on valid transactions. This kind of constraint goes beyond the regular database stored procedures, because it cannot be circumvented under any circumstances.<br>• The database has per-row permissions that use public key cryptography. Further, every transaction presents a publicly auditable proof that its creator(s) had the right to delete or modify its prior rows.<br><br>Of course, not by any coincidence, these are very relevant features for the inter-company financial ledger databases. **Signed commitments with immutable history is all that is required for proof of integrity.** Moreover, by assuming that commitments are immutable (transactions can only be reversed by adding a new commitment that reverses the actions of the previous commitment), you only need to keep track of the most recent commitment. |

| | |
|---|---|
| | If the commitment signer is a known entity, a single honest 'auditor' is all that is required to keep the commitment signer honest. Anyone closely watching the signer will be able to easily prove that the signer modified the history.<br><br>Another use case is where the permissioned participants are a limited group of cooperating parties, where there is no particular enduring trust. The NASDAQ example is this use case. A known set of participants who currently **remove the trust requirements by manual records (usually spread sheets) and expensive lawyers**. A blockchain style shared database, whilst slower than an SQL DB, solves the proof of integrity in this case, in a both faster and less expensive manner than the current manual or legal processes. |

| | |
|---|---|
| **Q17** | What is a membership service provider (MSP), and what are its roles? |
| **Reference** | https://hyperledger-fabric.readthedocs.io/en/release-1.4/membership/membership.html |
| **Ans** | As the Hyperledger Fabric is a permissioned network, participants need to prove their identity. It is the MSP that turns an identity into a **role** by identifying specific privileges that an actor has on a node or channel.<br><br>  <br>*Identities are similar to your credit cards that are used to prove you can pay. The MSP is similar to the list of accepted credit cards.* |

| | |
|---|---|
| **Q18** | What is TPS in a Hyperledger? |
| **Reference** | https://www.ibm.com/blogs/research/2018/02/architecture-hyperledger-fabric/ |
| **Ans** | The performance numbers obtained show that the Hyperledger Fabric that is deployed in a single cloud data center achieves an end-to-end throughput of more than 3,500 transactions per second with the latency of less than one second. While the research team refrains from making any direct comparisons to the other systems, the benchmarking effort has given the team some valuable insight in the behaviour of Fabric.<br>It must also be noted that a Hyperledger Fabric is a complex distributed system and its performance depends on multiple parameters, ranging from choice of the distributed application, to transaction size, and theto consensus implementation to the hardware. You can take this work as a starting point to further evaluate and optimise the Hyperledger Fabric, but this benchmark already demonstrates that the platform is extremely robust. |

| | |
|---|---|
| **Q19** | What is the orderer system channel? |
| **Reference** | https://hyperledger-fabric.readthedocs.io/en/release-1.1/ordering-service-faq.html |
| **Ans** | The orderer system channel (sometimes also called the ordering system channel) is the channel that initially the orderer is bootstrapped with. It is used to orchestrate channel building. The orderer system channel can define consortia and the initial configuration for new channels. At channel creation time, the organisation's definition in the /Channel groups's values, consortium consortium, and policies, along with the /Channel/Orderer group's values and policies, together form the new initial channel definition. |

| | |
|---|---|
| **Q20** | If the application is updated, should the orderer system channel be updated as well? |

| | |
|---|---|
| **Reference** | https://hyperledger-fabric.readthedocs.io/en/release-1.1/ordering-service-faq.html |
| **Ans** | Once the application channel is made, it is managed independently of other channels. Depending on the changes, the modification may or may not be desirable to port to other channels. In general, the MSP changes should be synchronised across all channels, while the policy modifications are more likely to be specific to a specific channel. |

| | |
|---|---|
| **Q21** | Can an organisation perform in both an ordering and application role? |
| **Reference** | https://hyperledger-fabric.readthedocs.io/en/release-1.1/ordering-service-faq.html |
| **Ans** | Although it is possible, it is a highly discouraged method. By default, the /Channel/Orderer/BlockValidation rule allows any valid certificate of the ordering organisations to sign blocks. If an organisation is acting both in an ordering and application role, then this policy should be updated to restrict the block signers to the subset of certificates authorised for ordering. |

| | |
|---|---|
| **Q22** | If someone wants to write a consensus implementation for Fabric. Where can they begin? |
| **Reference** | https://hyperledger-fabric.readthedocs.io/en/release-1.1/ordering-service-faq.html |
| **Ans** | A consensus plugin has to implement the Consenter and Chain interfaces that are defined in the consensus package. There are two plugins that are made against these interfaces: kafka and solo. |