**Client-Server**

1. Tell us about the features of the client/server?

#Client/Server is the mode of communication between the no of clients and the server. All the clients can send a request to the server.

CLIENT: Clients are the one who sends the request to the server.

            Client is usually considered as master or they are the first act.

            Once the request is sent to the server, the client expects and receives the response from

            server.

SERVER: server on the other end waits to get a request from client.

                 It will respond to the request from the client.

                 One server will respond to multiple clients.

1. What is a Web server in a client/server environment?

# Web server process is an example of client/server model. A web server is a Software/Hardware that’s uses protocols like HTTP on WWW made by the client and responds to it. Whereas hardware is the computer that has web server software’s to access it.

A web server in a client server environment, which is built or designed for clients and server according to that client/server architecture.

1. What is the role of the presentation layer?

# Presentation layers receives inputs from the client and display output. Basically, what the ever the information client requests will be sent to the next layers in the process and get the output from them.

1. They say this architecture is secure, how is it done in your opinion?

# Client server architecture is secure, as all the required data is stored in a single server. So it’s easy to protect the data. There is no need for a physical server and can be accessed via the web easily. It is secured by firewalls.

1. What is a Database Server in a client server environment?

# A Database server is a system or server that stores and manages the data. It is responsible for receiving the raw input from the clients and sending the output requested.

1. What are Super servers in client server environments?

# Super server is a program that runs in the background to start the other server when needed. They can be ideal when they are unused.

1. Explain 2-Tier and 3-Tier architecture?

# 2-Tier architecture is a software architecture that has two layers in it. Presentation layers which take inputs or requests from the client and the data layer which store the data. Security in 2-tier architecture is low when compared to 3-tier architecture.

3-Tier architecture is a software architecture which has 3 layers. Presentation layer which receives input and display output to the client.

Application layer which is used to calculate and do the logic.

Database layer which stores the data in the server and manages it. It sends the data to the presentation layer as output to the client.

3-Tier architecture has more advantages like, high efficiency, performance, security, scalability, flexibility.

1. What is a File server?

#A file server is the computer responsible for the central storage of the data or files which can be accessed by all the clients.

**SOA & MicroServices**

1. What are the main benefits of SOA?

#Service reusability, easily maintained, availability, scalability, loose coupling, discoverability, comparability, and increased productivity.

1. How can you achieve loose coupling in SOA?

# Loose coupling means that a service can be updated independently no changes are need in other services while updating a service.

1. Are web services and SOA the same?

# Web services are not same as SOA, Web services is a web technology that is build to display the inputs and outputs. SOS is an architectural model for implementing loosely coupled service based application.

1. What is a reusable service?

#A reusable service is the extra functionality a service can be reused in future and how much the current functionality goes beyond the current requirement.

1. What are the disadvantages of SOA?

# High Bandwidth Server, Extra overload and High cost are some of the disadvantages of SOA.

1. What is ESB and where does it fit in?

# ESB, Enterprise Service Bus is an intermediate platform which connects all the services.

1. In SOA do we need to build a system from scratch?

# No, we integrate any system to an existing system by loose coupling.

1. What is the most important skill needed to adopt SOA ?technical or cultural?

# Cultural, as we don’t need to different technology to adopt SOA.

1. List down the advantages of Microservices Architecture?

# Agility, small focused teams, small code base, mix of technologies, fault isolation. Scalability is higher and data isolation.

1. What are the best practices to design Microservices?

#

1. How does Microservice Architecture work?

# Microsoft architectures are easy to simpler and maintain when broken or divided into many smaller pieces that works together. In microservices, software functionality is isolated into multiple independent modules but they are individually responsible for their precise design.all these modules communicate through API’s.

1. What are the pros and cons of Microservice Architecture?

# PROS: improves scalability, better fault isolation, optimizing scaling decisions, localized complexity, small and agile developing teams, increase productivity are some of them.

CONS: Sometimes architecture can gets complicates, careful planning is required, proper size of microservics is required are some of them.

1. What is the difference between Monolithic, SOA and Microservices Architecture?

# Monolothic architecture is basic design of software, it is a single piece which is a one code base.

SOA, is the modular means of breaking the monolithic into smaller components.

Microservices, is still smaller and fined grained approach to achieve same result.

1. What are the challenges you face while working Microservice Architectures?

# Complexity increases with the increase in the microservices.

Monitoring, testing, cyclic dependency, fault tolerance gets difficult.

1. What are the characteristics of Microservices?

# Componentization via services, business capability, products not projects, smart end points, decentralized governance, decentralized data management, design for failure, infrastructure automation,

Evolutionary design.