**Client-Server**

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

**Servers** are Robot-type programs that constantly run and exchange information with remote users. **Clients** are programs that access and exchange information with server. A client sends requests to the server and it expects and receive response from the server. A server is listening to the client and if it receives a request, server treats it and sends a response.

2.What is a Web server in a client server environment?

A web server is a software and hardware that uses HTTP and other protocols to respond to the requests send by the clients through internet. A web server stores, process and deliver the website content to the clients

3.What is the role of the presentation layer

It is one of the tier in the 3 tier client-server architecture. This creates the views that user see. In this layer we can see inputs given by the users and also output from the server.

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

Three Tier Client-Server Architecture is more secure. Here we have three tiers. The user interactions are done at the Presentation tier, the processing and calculations are done at the Application tier and the storing and managing data is done at the Database tier. As a result we can introduce firewalls in between these tiers and which make it more secure. If a hacker tries to break in to the database he need to crack many firewalls which lies in between these tiers. So Three Tier Client – Server Architecture is more secure

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

Database server is a computer which is responsible for database storage. It only deals with storing and retrieving data in the database. Database server is useful for organizations having a lot of data to deal with on a regular basis. When the client want to process data frequently, it is better to have a Database server.

6.What are Super servers in client server environments?

Super server is a server process that monitor the arrival of client requests and starts the appropriate server service.

7.Explain 2-Tier and 3-Tier architecture

2-Tier Architecture

Here, there are basically two layers(tiers), Client and Server. These two layers communicate each other through internet. This structure is quiet easy to maintain and modify. The client sends request to server and the server process the request and sends response. If the client sends many requests the server will not be able to handle it properly and it can decrease the overall performance of the system. This is the main disadvantage of 2-Tier Architecture.

3-Tier Architecture

Here there is three layers(tiers) in this architecture. The user interactions are done at the Presentation tier, the processing and calculations are done at the Application tier and the storing and managing data is done at the Database tier. Here each tier have their own functions to perform. This increase efficiency. 3 Tier Architecture is more secure since we can add many firewalls in between the tiers.

8.What is a File server?

A File server is a central server instance in a computer that helps the connected clients to access the server’s storage spaces. With file servers connected to the internet and configured accordingly, users cannot only access the files via the local network but also benefit from **remote access**

**SOA & MicroServices**

1.What are the main benefits of SOA?

Service Oriented Architecture means In a software system, we can connect every services/systems together using ESB. ESB act like a communicator service. Or simply, SOA is a software design where services/systems connect with each other using a common communicator called ESB. All these systems may have different methods for communication. So while using ESB, every system will be connected to ESB and through ESB, communication happens.

The main benefit of Service Oriented Architecture is that if we change any system, we don’t need to change its communication with every other services, we only need to connect it with ESB. Every services connects ESB so it will be easy to make changes.

2.How can you achieve loose coupling in SOA

3.Are web services and SOA the same?

SOA is an architectural style used for building software having different services communicating each other. Thus, a webservice is a building block in SOA. These services may use web service concept for communication. When we combine all these services and connectivity, we have an application that falls under SOA.

4.What is a reusable service?

These are services that can be reused through the entire project.

5.What are the disadvantages of SOA?

> SOA is very costly in terms of human resources, technology and development.

> In SOA, if we are using multiple services, it will overload our system.

> In SOA, some services send and receive messages frequently so it reaches million requests per day. So it need high speed server with a lot of data bandwidth to run a web service

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

ESB implements a communication system between software systems in a Service Oriented Architecture. Every systems will be connected to it for establishing communication. The best way to think of an ESB is to compare it to a Router.

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

No. we don’t need to build a system from scratch in SOA. We can integrate existing system in SOA

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

Cultural skill. SOA require people to think of business and technology differently.

9.List down the advantages of Microservices Architecture.

* Each microservices can be deployed independently.
* We can assign specific microservices to specific groups so one group can focus only on one service
* Microservices are independently scalable
* If a specific microservice fails, we can isolate that failure to that particular service which prevent other services from failing.

10. How does Microservice Architecture work?

In Microservice architecture, we can isolate software functionality into multiple independent modules that are responsible to perform defined tasks. These modules communicate each other through APIs.

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

Pros

* Greater agility
* Better scalability
* Faster development cycles
* Isolated services have better fault tolerance

Cons

* Harder to test and monitor due to its complexity of architecture
* Harder to maintain the network
* Doesn’t work without proper corporate cultures like DevOps

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

**Monolithic Architecture**

Monolithic architecture is a concept when complete software is developed as a single unit. All functionality, features and modules are developed and integrate and deployed in the form of a single entity. A single platform containing user interface, business logic and database layer on the same page. Monolithic application has a single shared database for each functionality and every feature combined in a single program as a unified model.

**Service Oriented Architecture**

SOA is an architectural style where we can connect every services/systems together using ESB. ESB act like a communicator service. Or simply, SOA is a software design where services/systems connect with each other using a common communicator called ESB. All these systems may have different methods for communication. So while using ESB, every system will be connected to ESB and through ESB, communication happens.

**Microservice Architecture**

In Microservice architecture, we can isolate software functionality into multiple independent modules that are responsible to perform defined tasks. These modules communicate each other through APIs. So it is very easy to update or add data to any modules in this architecture.

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

* Managing microservices as the number of microservices increases, managing them gets more challenging
* Monitoring these microservices
* Fault tolerance because It is important that individual services do not bring down the overall system
* Testing is much more complex in a microservices environment due to the different services and their integration

14.What are the characteristics of Microservices?

* Since the software functionality is broken down to independent modules, each modules can be deployed and modified without affecting the other functional aspects of the system
* All modules may not be connected with same database. Each one can have its own database. So it can prevent overloading

15. What are the best practices to design Microservices?

* Create separate databases for each microservices.
* Keep all code in a microservice at a similar level of stability - if we need to add or rewrite some of the code in a deployed microservice that’s working well, create a new microservice for the new or changed code, leaving the existing microservice in place
* Do a separate build for each microservice.
* Deploying microservices in containers.
* Treat servers as interchangeable members of a group. They all perform the same functions, so we don’t need to be concerned about them individually. our only concern is that there are enough servers to produce the amount of work we need, and we can use autoscaling to adjust the numbers up and down. If one stops working, it’s automatically replaced by another one