

Client-Server

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

In a client/server architecture, client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them.

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

A Web server is software that uses HTTP (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web (WWW).

3. *What is the role of the presentation layer?*

Presentation layer is the front-end layer of the application and the interface with which end-users will interact through a web-based application. It is also used to present data to the application layer in an accurate way

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

I believe that in a 3-tier architecture, the presence of an extra layer in between makes this architecture more secure from hacking activities. Also, more the layers, more will be the number of firewalls present which makes it again more secure.

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

The term database server may refer to both hardware and software used to run a database, according to the context. As software, a database server is the back-end portion of a database application, following the traditional client-server model. This back-end portion is sometimes called the instance. It may also refer to the physical computer used to host the database. When mentioned in this context, the database server is typically a dedicated higher-end computer that hosts the database.

6. *What are Super servers in client server environments?*

To my understanding super server controls the use of other servers. It monitors the arrival of a client request and starts the appropriate server service.

7. *Explain 2-Tier and 3-Tier architecture*

In 2 tier architecture, direct communication takes place between client and server. It is divided into client tier and database tier. The main problem of two tier architecture is the server cannot respond multiple request same time, as a result it cause a data integrity issue.

In 3 tier architecture, it is divided into 3: presentation layer, application layer and database layer. The application layer acts as an interface between Client layer and Data Access Layer.

8. *What is a File server?*

A file server is a server that provides access to files. It acts as a central file storage location that can be accessed by multiple systems. File servers can be configured in multiple ways. For example, in a home setting, a file server may be set to automatically allow access to all computers on the local network (LAN). In a business setting where security is important, a file server may require all client systems to log in before accessing the server. Others may only grant access to a specific list of machines, which can be defined by MAC address or IP address.

SOA & MicroServices

1. *What are the main benefits of SOA?*

- Loose Coupling
- Flexibility
- Easier Testing and Debugging
- Scalability
- Reusability

2. *How can you achieve loose coupling in SOA*

Being on a SOA stack means that your infrastructure and architecture are split up into various services. As a consequence, you write software that tends to be loosely coupled

3. *Are web services and SOA the same?*

They are related, but not the same. Web services are a form of SOA implementation. SOA doesn't require Web services. Web services are a set of implementation standards designed for service-oriented architectures.

4. *What is a reusable service?*

The service reusability is to create services that can be reused across a business. These reusable services are designed so that their solution logic is independent of any particular business process or technology.

5. *What are the disadvantages of SOA?*

- In SOA, all inputs are validated before it is sent to the service. If you are using multiple services then it will overload your system with extra computation.
- SOA is costly in terms of human resource, development, and technology
- As some web service sends and receives messages and information frequently so it easily reaches a million requests per day. So it involves a high-speed server with a lot of data bandwidth to run a web service.

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

The Enterprise Service Bus connects all the services together over a bus like infrastructure. It acts as communication center in the SOA by allowing linking multiple systems, applications and data

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

No, we can also implement SOA by exposing existing services

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

Cultural, SOA requires people to think in terms of business functions or services for eg: "my company does these functions, how can I set up my IT?" should be the way of thinking rather "if I implement this technology, how will it benefit my company?".

9. *List down the advantages of Microservices Architecture.*

Scalability and reusability, as well as efficiency. Easy to scale and integrate with third-party services. Components can be spread across multiple servers

10. *What are the best practices to design Microservices?*

- Domain-Driven Design
- Database per Microservice
- Micro Front ends
- Continuous Delivery
- Observability
- Unified Tech Stack
- Asynchronous Communication
- Infrastructure over Libraries
- Organizational Considerations

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

PROS

- Greater agility
- Faster time to market
- Faster development cycles
- Platform and language independent services

CONS

- harder to test and monitor due to the complexity
- Security issues
- Needs more collaboration
- Poorer performance due to network latency and message processing

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

Monolithic apps consist of interdependent, indivisible units and feature very low development speed. SOA is broken into smaller, moderately coupled services, and features slow development. Microservices are very small, loosely coupled independent services and feature rapid continuous development.

In a service-oriented architecture, they communicate through a piece of specialized software called an enterprise storage bus. Microservices communicate via more lightweight protocols. It is worth noting that microservices arose out of the SOA, and are sometimes considered a kind of SOA, or successor to the concept.

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

- Managing Microservices
- Monitoring
- Embracing DevOps Culture
- Fault Tolerance
- Testing
- Cyclic Dependencies

14. What are the characteristics of Microservices?

- Services are built around business capabilities , independently deployable and packaged, each running in its own process.
- Each Service should have separate database layer
- Each Service can have independent codebase, CI/CD tooling sets
- Each Service can be tested in isolation without dependent on other services.
- Each Service should have monitoring and troubleshooting capabilities for operation team
- Each Service can implement independent security mechanism
- Services can use HTTP(Rest) or messaging for communication or any other lightweight communication protocol.
- Each Service can be run without waiting for other service to go online
- Service can use different language,framework and technologies
- Maintain Independent Revisions and Build Environments to maintains compatibility with other services.

HTTP fundamentals

1. *What are the basic Features of HTTP?*

- **Connectionless:** The client and server know about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.
- **Media independent:** Any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content.
- **Stateless:** The server and client are aware of each other only during a current request. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

2. *What are request methods in HTTP?*

- **GET:** to retrieve information
- **POST:** used to send data to the server
- **PUT:** To replaces all current representations with the uploaded content.
- **DELETE:** Removes all current representations of the target which is given by a URI.
- **HEAD:** transfers the status line and header section only
- **TRACE:** Performs a message loop-back test along the path to the target

3. *What are the differences between GET and POST methods?*

GET is used to retrieve information and POST is used to send data to the server

4. *What is status code in HTTP?*

Status codes are issued by a server in response to a client's request made to the server.

- *1xx informational response* – the request was received, continuing process
- *2xx successful* – the request was successfully received, understood, and accepted
- *3xx redirection* – further action needs to be taken in order to complete the request
- *4xx client error* – the request contains bad syntax or cannot be fulfilled
- *5xx server error* – the server failed to fulfill an apparently valid request

5. *What are the header fields in HTTP?*

There are four types of HTTP message headers:

- **General-header:** for both request and response messages.
- **Client Request-header:** for request messages only.
- **Server Response-header:** for response messages.
- **Entity-header:** define meta information about the entity-body

6. *What is URI?*

Uniform Resource Identifier is a string of characters that unambiguously identifies a particular resource.

7. *What are Idempotent methods and why do we call them?*

An idempotent method is a method that can be called many times without different outcomes. It would not matter if the method is called only once, or ten times over.

8. *What is Session State in HTTP?*

Session state is a method keep track of the a user session during a series of HTTP requests. It allows a developer to store data about a user as he/she navigates through ASP.NET web pages in a .NET web application.

9. *What is HTTPS?*

Hypertext transfer protocol secure (HTTPS) is the secure version of HTTP, which is the primary protocol used to send data between a web browser and a website. HTTPS is encrypted in order to increase security of data transfer. This is particularly important when users transmit sensitive data, such as by logging into a bank account, email service, or health insurance provider.