**Client-Server**

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

The client and server should follow a common communication protocol so they can easily interact with each other. All the communication protocols are available at the application layer. A server can only accommodate a limited number of client requests at a time.

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

A Web server is software or hardware that uses HTTP (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web (WWW). Web server software controls how a user accesses hosted files. It is accessed through the domain names of websites and ensures the delivery of the site's content to the requesting user.

1. What is the role of the presentation layer?

The presentation layer acts as a translator between the application and the network, mainly addressing the syntax representation of user information, ie, providing formatted representations and translation data services. Data compression, decompression, encryption, decryption are completed in this layer.

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

The Security is achieved by Firewall. It makes sure all the data is secure and nothing is accessible.

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

Database server: A computer that is responsible for database storage, access, and processing in a client/server environment. Sometimes used to describe a two-tier client/server environment.

1. What are Super servers in client server environments?

A super-server starts other servers when needed, normally with access to them checked by a TCP wrapper. It uses very few resources when in idle state. This can be ideal for workstations used for local web development, client/server development or low-traffic daemons with occasional usage (such as ident and SSH).

1. Explain 2-Tier and 3-Tier architecture

A two-tier architecture is a database architecture where presentation layer runs on a client and .data is stored on a Server. 3-tier architecture is consisting of the Presentation layer (PC, Tablet, Mobile, etc.), the Application layer (server) and Database Server.

1. What is a File server?

A file server is a central server in a computer network that provides file systems or at least parts of a file system to connected clients. File servers therefore offer users a **central storage place** for files on internal data media, which is **accessible to all authorized clients.**

**SOA & MicroServices**

1. What are the main benefits of SOA?

The advantages of SOA:

* Service Reusability
* Easy Maintenance
* Platform Independent
* Availability
* Reliability
* Scalability

1. How can you achieve loose coupling in SOA?

One strategy for achieving loose coupling is to use the service interface (the WSDL for a SOAP Web Service) to limit this dependency, hiding the service implementation from the consumer. Loose coupling can be addressed by encapsulating the service functionalities in a manner that limits the impact of changes to the implementation on the service interface. However, at some point you will need to change the interface and manage versioning without impacting service consumers, in addition to managing multiple security constraints, multiple transports, and other considerations.

1. Are web services and SOA the same?

There are some key differences between Web services and SOA. Web services define a web technology that can be used to build applications that can send /receive messages using SOPA over HTTP. However, SOA is an architectural model for implementing loosely coupled service-based applications.

1. What is a reusable service?

The service reusability principle is a design principle, applied within the service-orientation design paradigm, to create services that can be reused across a business. These reusable services are designed so that their solution logic is independent of any business process or technology. A reusable service should be governed at the enterprise level throughout its entire lifecycle, from design-time through run-time. Its reuse should be promoted through a prescriptive process, and that reuse should be measured.

1. What are the disadvantages of SOA?
   1. Stand alone, non-distributed applications that do not necessitate application or component integration; that would include, for instance, a word processing application that does not require request and response-based calls.
   2. Short lived applications or applications that are in any way limited in scope.
2. What is ESB and where does it fit in?

The core technology capability of an SOA is the Enterprise Service Bus (ESB) or a similar network and messaging backbone. As an SOA infrastructure component, the ESB is the carrier for all consumer and service interactions. It is also a potential bottleneck.

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

No, if we need to integrate any existing system you just can loosely couple wrappers which help in wrapping all customer services and expose all functionalities in a generic manner.

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

Surely cultural. SOA does require people to think of business and technology differently. Instead of thinking of technology first (e.g., If we implement this system, what kinds of things can we do with it?), practitioners must first think in terms of business functions, or services (e.g., My company does these business functions, so how can I set up my IT system to do those things for me most efficiently?).

It is expected that adoption of SOA will change business IT departments, creating service-oriented (instead of technology-oriented) IT organizations.

1. List down the advantages of Microservices Architecture.

* Agile
* Small focused teams
* Small code based
* Mix of technologies
* Fault isolation
* Scalability
* Data isolation

1. 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

1. How does Microservice Architecture work?

The main idea behind a microservice architecture is that applications are simpler to build and maintain when broken down into smaller pieces that work seamlessly together. These modules communicate with each other through simple, universally accessible application programming interfaces (APIs).

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

Pros:

* Greater agility
* Faster time to market
* Faster development
* Better scalability
* Easier to create a CI/CD pipeline for single responsibility services
* Isolated services have better fault tolerance
* Cloud readiness

Cons:

* Needs more collaboration
* Harder to test and [monitor](https://raygun.com/blog/monitoring-microservices/) because of the complexity of the architecture
* Poorer performance, as microservices need to communicate
* Harder to maintain the network
* Doesn’t work without the proper corporate culture (DevOps culture, automation practices, etc.)

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

A monolithic architecture is the traditional unified model for the design of a software program. Monolithic, in this context, means composed all in one piece. In a tightly coupled architecture, each component and its associated components must be present in order for code to be executed or compiled.

SOA uses Enterprise Service Bus for communication whereas microservices use much simpler messaging systems. Each microservice stores data independently while in SOA components

share the same storage.

1. What are the challenges you face while working Microservice Architectures?
   1. Needs more collaboration
   2. Harder to test and [monitor](https://raygun.com/blog/monitoring-microservices/) because of the complexity of the architecture
   3. Poorer performance, as microservices need to communicate
   4. Harder to maintain the network
   5. Doesn’t work without the proper corporate culture (DevOps culture, automation practices, etc.)
2. What are the characteristics of Microservices?
   1. Componentization via Services.
   2. Organized around Business Capabilities.
   3. Products not Projects.
   4. Smart endpoints and dumb pipes.
   5. Decentralized Governance.
   6. Decentralized Data Management.
   7. Infrastructure Automation.
   8. Design for failure.