**Client-Server**

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

Ans: In client server architecture the client send request to the server for resources and the server send back the response to the client with the desired information .The client and server uses same communication protocol for exchanging information At a time the server can handle only a number of request from the client .They will communicate over the internet to send requests and information .

* Since the information is placed in the single place , security such as authentication and authorization can be done for the data .
* Since the information is the server side , we have the advantage of accessing the information form anywhere with out any issues , ie is easy to access the information .
* There will not be any issues about the platform used by the client and the server, because it is platform independent.
* Since the client and the server is independent of each other, upgrading ,replacing is easy because they are not tightly coupled .

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

Ans : Server is computer that runs the web service. The requests by the client is processed by the server and the corresponding information requested by the client is send back to the client in form of response .It is the web server who authenticate and authorize the request by the client .It is the web server who handle the situation when there are many number of request for the web service is received by the server . so basically the thing that handles the request from the client and necessarily process the respond back with corresponding information requested by the client .

1. What is the role of the presentation layer?

Ans: The presentation layer the interface the user communicates with the servers. The client send request for information using the presentation layer , along with that the information received from the server side is viewed by the client with the help of presentation layer .The raw data received by the application layer from the system layer is processed and converted according to the presentation layer and is displayed by the presentation layer. In the presentation layer it can be mobile application , or a website , or and web site in mobile phone , tablet ,computer web browser etc.

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

Ans :

There is a fire wall setup between client and server server communication .

Use SSL certificate on the server to ensure the traffic is secure

Since the it follows the hierarchy pattern even if the front end is being hacked the main vulnerable data is in the database layer so it is much more safer because it is guarded by the other two layers.

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

Ans : Database server is the core component of the client server architecture , it is the backend part of the client server environment. It from where the data is stored or data is saved . when the client request for the data , the data is fetched from the database server , and the fetched data is processed in the application layer , and then given as the response to the client . so database system commonly applied is RDMS , commonly deployed database server is MS SQL server , Oracle SQL database server.

Their main function is to maintain files .

1. What are Super servers in client server environments?
2. Explain 2-Tier and 3-Tier architecture?

Ans :

2 Tier architecture

Two-tier architecture consists of two layers ,Client and server .I this style the client communicates directly to the server for the services and the client gets served by the server . The processing logic may be embedded inside the client itself or in the server side . Security in this layer is much lesser than compared to that of 3 tier .

3 Tier architecture

In this style the main implementation logic is implemented in the layer called application layer which reside in between the presentation layer and the database layer . The client request Is passed through the application layer and various layers of firewalls and authentication and data is fetched from the database layer and id processed in the application layer and the processed data is send to the presention layer , and visualised by the end user . Since this style there is an hierarcial model of separation by different layers and also the main data resides in the database layer which is hidden, is much more secure and more scalable , efficient , less time consuming than 2 tier .

1. What is a File server?

Ans: 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.** The server administrator defines strict rules regarding which users have which **access rights**: For instance, the configuration or file authorizations of the respective file system enable the admin to set which files can be seen and opened by a certain user or user group, and whether data can only be viewed or also added, edited, or deleted. 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**. This enables files to be accessed and saved on the file server even when users are on the go.

**SOA & MicroServices**

1. What are the main benefits of SOA?

Service-oriented architecture (SOA) is a style of [software design](https://en.wikipedia.org/wiki/Software_design) where services are provided to the other components by [application components](https://en.wikipedia.org/wiki/Application_components), through a [communication protocol](https://en.wikipedia.org/wiki/Communications_protocol) over a network. A SOA service is a discrete unit of functionality that can be accessed remotely and acted upon and updated independently, such as retrieving a credit card statement online. SOA is also intended to be independent of vendors, products and technologies .

* Improved information flow Ability to expose internal functionality Organizational flexibility.
* Configuration flexibility.
* Improved reliability.
* Ability to scale operations to meet different demand level.
* Ability to develop new functions rapidly.
* Application of security policy.
* Application of security policy.

1. How can you achieve loose coupling in SOA?

Ans :In SOA , each functionalities implemented as an separate services , such that when there is need of that functionality ,the corresponding service is connected and they are served . so each services are build independent from each other such that they are loosely coupled and they work independently. These loosely coupled servies make it easy for trouble shooting , scalable, flexible , fault tolerant etc.

1. Are web services and SOA the same?

Ans : SOA is an architectural style for building software applications that use services available in a network such as the web. It promotes loose coupling between software components so that they can be reused. Applications in SOA are built based on services. A service is an implementation of a well-defined functionality, and such services can then be consumed by clients in different applications processes.SOA allows for the reuse of existing assets where new services can be created from an existing services . In other words, it enables us to make use of existing investments by allowing them to reuse existing applications, and promises interoperability between heterogeneous applications and technologies. SOA also provides a level of flexibility.

**Web service** is a standardized medium to propagate communication between the client and server applications on the World Wide Web. A web service is a software module that is designed to perform a certain set of tasks. Client request a information from the webservice, and after the information is processed , it is send to the user and the user view the information in the front end.

**SOA is an architectural paradigm that uses the existing web services to implement the business logic, so that existing is being fully leveraged .**

1. What is a reusable service?

Ans: SOA is an example of reusable services. In SOA every functionalities are implemented as services and these services work independently , they are loosely coupled . since they are loosely coupled they can work indepented of each other .so any changes to the SOA services will not effect the other services . This makes the service reusable ,ie where where we want to implement the same functionality ,the same service can be used again and again ,since the service is loosely coupled to other servies it can service anything who calls the services which authorized to access the service ,So where ever we want to implement a logic, if that logic is already provided by the a web service , since it is loosely couples , it can be used any where.

1. What are the disadvantages of SOA?

* High bandwidth requirement : because inSOA there will be many services that is being communicating in an instant of time, so high bandwidth need to be allocated inorder maintain a safer unintrupted tranfer of information.
* **High Cost**– It is expensive in terms of human resource, development, and technology.
* Extra overload : 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

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

Ans : An [enterprise service bus](https://www.innovativearchitects.com/KnowledgeCenter/business-connectivity/ESB-EAI-SOA.aspx) (ESB) refers to software architecture that allows for the integration of enterprise applications and services, such as middleware infrastructure platforms. The best way to think of an ESB is to compare it to a router. It provides the connections between applications that need to communicate with one another. Businesses use ESBs in enterprise application integration. The interactions and communication between components take place across the bus. In that respect, an ESB is similar to a physical computer bus; it handles data transfer or message exchange without writing any code. As an infrastructure software service-oriented model, ESB works as a managed message system that provides routing, data transformation, translation upon a client’s request, and event interpretation. ESB comes in handy when you need to transform messages into a format the application can interpret, to change data content, or execute services via a rule engine.

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

Ans : No , we not need to build the entire system form scratch in SOA because , SOA is based of loosely coupled , reusable , scalable , modular servies make sure that a product is being fully leveraged. So when it comes to SOA , we do not need to build it fully from scratch , we check whether there is a existing asset which fit our need need , so it will save us time .

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

Ans: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.

Ans :

* **Microservices are independently deployable**
* **Microservices are independently scalable**
* **Microservices reduce downtime through fault isolation**

1. What are the best practices to design Microservices?
   * The Single Responsibility Principle

#### **Have a separate data store(s) for your microservice**

#### **Use asynchronous communication to achieve loose coupling**

#### **Fail fast by using a circuit breaker to achieve fault tolerance**

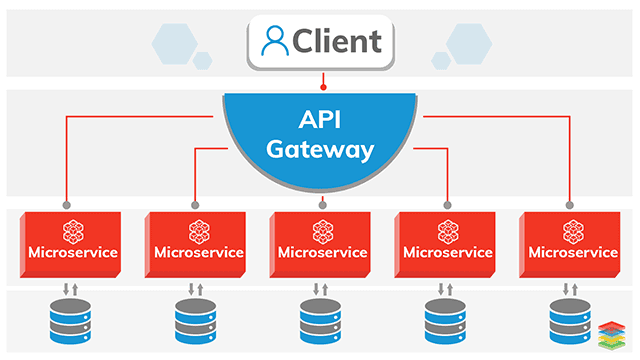
#### **Proxy your microservice requests through an API Gateway**

#### **Ensure your API changes are backwards compatibleVersion your microservices for breaking changes**

#### **Have dedicated infrastructure hosting your microservice**

#### How does Microservice Architecture work?

#### Ans : Microservices are an architectural approach to creating cloud applications. Each application is built as a set of services, and each service runs in its own processes and communicates through APIs, in approach , when the client want to access a microservice , first the request is passed through the API gateway any the api gateway will provide authentication to the client and then api gateway will route the client request to corresponding microservice, and the mircroservice gets the work done and send back the response to the client .



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

**pros**

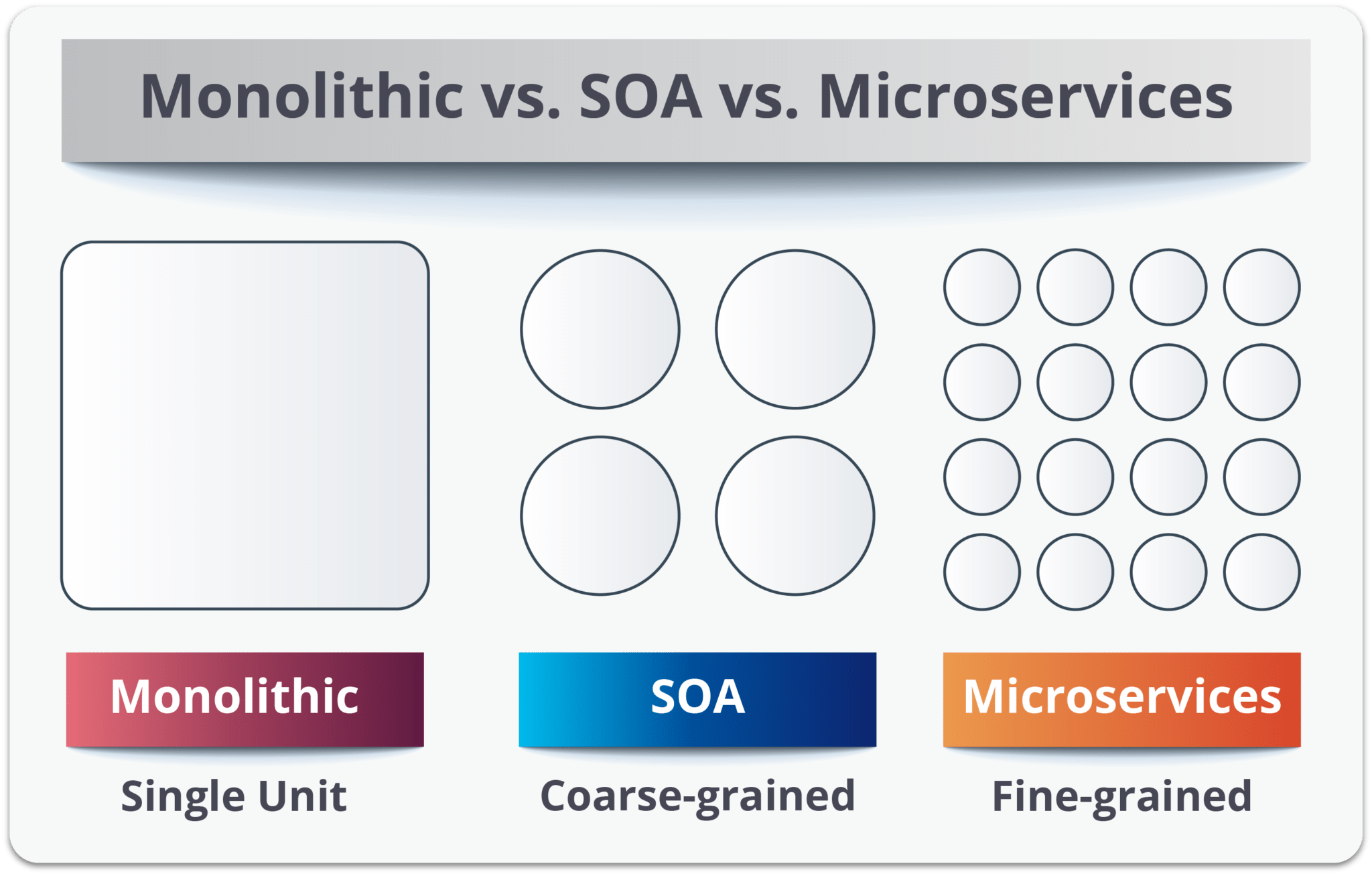
* Improved fault isolation: Larger applications can remain mostly unaffected by the failure of a single module
* Ease of understanding: With added simplicity, developers can better understand the functionality of a service.
* Smaller and faster deployments: Smaller codebases and scope = quicker deployments, which also allow you to start to explore the benefits of Continuous Deployment.
* Scalability: Since your services are separate, you can more easily scale the most needed ones at the appropriate times, as opposed to the whole application. When done correctly, this can impact cost savings.

**Cons**

* **Communication between services is complex**: Since everything is now an independent service, you have to carefully handle requests traveling between your modules. In one such scenario, developers may be forced to write extra code to avoid disruption. Over time, complications will arise when remote calls experience latency.
* **More services equals more resources**: Multiple databases and transaction management can be painful
* **Large vs small product companies**: Microservices are great for large companies, but can be slower to implement and too complicated for small companies who need to create and iterate quickly, and don’t want to get bogged down in complex orchestration.
* **Deployment challengers**: The product may need coordination among multiple services.

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

Ans : here are several points to consider when deciding whether microservices or SOA is better for a particular business. SOA is a modular means of breaking up monolithic applications into smaller components, while microservices provides a smaller, more fine-grained approach to accomplishing the same objective. Both of these architectures are routinely run in the cloud, which increases the flexibility for building and deploying applications. Ultimately, the best approach depends on each business’s own unique needs and use case.



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

## **Managing Microservices**

As the number of microservices increases, managing them gets more challenging. It is important that management is planned before or while microservices are being built

## **Monitoring**

since you have multiple services making up the same functionality previously supported by a single application. When a problem arises in the application, finding the root cause can be challenging if you do not have a means of monitoring and tracking the path a specific request took,

## **Fault Tolerance**

t is important that individual services do not bring down the overall system. Fault tolerance at the service level, and more importantly, at the overall solution level, is critical. Given the complexity of a microservices environment and the complex dependency chains, failure is inevitable. Microservices need to be able to withstand both internal and external failures.

## **Testing**

Testing is much more complex in a microservices environment due to the different services, their integration, and interdependencies.

1. What are the characteristics of Microservices?

Ans:

* + Microservice architectural style is an approach to developing a single application as a suite of small services
  + Services are built around business capabilities , independently deployable and packaged, each running in its own process
  + Each Service can be tested in isolation without dependent on other services
  + Each service can pick the best technology stack for its use cases
  + Each Service should have monitoring and troubleshooting capabilities for operation team
  + Services can use HTTP(Rest) or messaging for communication or any other lightweight communication protocol.