**Client Server**

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

Answer:

Client and Server interact with each other with a system of request and response. A client system provides an interface to the user to request services of the server and display the result from server to the end users. Server will wait for the request to from client and check if the request is valid or invalid. The server will respond will an error message if the request is invalid and will send response if request is valid. Clients will be mostly computer or mobile phones whereas servers will be situated elsewhere in the network.

request

Server

Client

response

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

Answer:

Web server is the presentation tier in a Client Server Architecture. It processes http request and respond by serving static content to the users. It displays information related to such services such as browsing merchandise, purchasing and shopping cart contents.

http request/ send url request

Server

Client

http response/ send html page

3.What is the role of the presentation layer

Answer:

Presentation layer is the top most layer in 3 Tier Architecture. It consists of user interface which is accessible through web browser or web based application by which the end user will request. The main purpose of presentation layer is to communicate this request to the application layer. And it will finally display the results to the end user

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

Answer:

Three Tier Architecture is divided into 3 layers as Presentation layer, Application layer and Database layer. Thus client layer and data layer is connected through middle tier. This middle tier is responsible for authenticating users’ identities before sending them to data. Thus data will be more secure. Each layer is separated by firewall which offers additional security.

When it comes to application point of view each application is independent in 3-tier architecture. Thus each application can be modified and deployed separately without affecting others. As this is multilayer system it offers feature upgrade with minimal impact of layers.

Scalability can be achieved horizontally by increasing memory, increasing processor speed and vertically by increasing number of servers when additional request is there.

By considering all these features this architecture can be considered as secure.

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

Answer:

Database Server is the Database Tier which is the third layer in a 3 Tier Architecture.

This is the back end system of the application. Its purpose includes Data analysis, Data storage and Data management. They will process and retrieve data upon request from Application layer.

6.What are Super servers in client server environments?

Answer:

Super server is programmed to start other servers when necessary and uses very few resources in idle state.

request

Server Node

Super Server

Client

results

response

Create on demand

Server A

7.Explain 2-Tier and 3-Tier architecture

Answer:

Two Tier Architecture

Two Tier Architecture is divided into 2 parts.

1.Client Tier

2.Database Tier

There is no intermediate between Client and Server and thus direct communication takes place between Client Machine and Database. Communication is faster as both are tightly coupled.

The main problem with two tier architecture is that the server cannot respond to multiple request at the same time.

Three Tier Architecture

Three Tier Architecture comprise 3 layers.

1.Presentation layer: request from user is taken and displays output to the end user

2.Application layer: All the business logic is written in this layer. This layer is called intermediary layer helps communication faster between client layer and data server.

3.Database layer: This layer stores and manage data.

Performance in 3-Tier Architecture will be high as load can be reduced. It offers high degree of flexibility, improved security and scalability.

The issue with this architecture is it increases complexity.

8.What is a File server?

Answer:

It is a server that provide access to files. In a client server network when a client requests a file system or part of file system, the file server will provide them. This server offers a central storage place for files on internal data media.

**SOA & MicroServices**

1.What are the main benefits of SOA?

Answer:

* Service Reusability: One of the main benefit of SOA is service reusability. Applications are exposed as services which are not limited to a single component but could be used in multiple components.
* Easy Maintenance: Services can be easily updated, upgrade or maintained in a SOA environment without any complication from the other services in the system. This is possible because all services are self-contained in SOA.
* Platform independent: SOA services are publically available to any system independent of the platform in which the system runs.
* Availability: SOA services can be searched and discovered easily as they are in a public platform. Thus services are easily available to any requester.
* Reliability: SOA services are easier to test and debug as they are independent services.
* Scalability: If any service is being requested by several users then it can be available to every user by achieving scalability by adding more servers.

2.How can you achieve loose coupling in SOA?

Answer:

In SOA loose coupling means every business is designed as a self-contained service which are independent of each other, thus services do not have any blockage. This independency can be achieved by the use of service interface.

Loose coupling can be implemented by encapsulating the service implementation logic from the consumers. In SOA service is completely defined in service contract document to hide the logic from end user.

3.Are web services and SOA the same?

Answer:

Service Oriented Architecture is an architecture model implementing loosely coupled services that can communicate with each other.

A Web Service is a software system designed to support machine to-machine interaction over a network by send/receive messages.

Web Services is an implementation of SOA.

4.What is a reusable service?

Answer:

A reusable service is an independent, self-contained, autonomous functionality which can be easily discoverable in the public directory. This services will be platform independent and can be available to any requester. The requester can modify the service according to their requirement.

5.What are the disadvantages of SOA?

Answer:

* Increased overhead when a service interacts with other. Inputs are validated before it is sent to services. This is increase the response time and machine load and thus reduce the overall performance.
* High Implementation cost is required in SOA in terms of human resource, development and technology.
* Require High Bandwidth Server as large number of messages are sent and receive across for each request.

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

Answer:

Enterprise Service Bus(ESB) is a hub which orchestrate all the services in a service oriented architecture. It is a platform that connect all the services. ESB is a middleware that communicate consumers and the providers.

Consumers

Users

Enterprise Service Bus(ESB)

Providers

Service1

Service2

DB

DB

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

Answer:

In SOA all services will be available as a self-contained component. Developers can use the existing services available and make required modification and develop their system. This will improve productivity as there is no need to build system from scratch.

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

Answer:

In SOA every business is exposed as a service. Both technical and cultural skills are needed in SOA. But instead of thinking of technology basis first, it requires organizations to think about the service or functions. SOA focus on creating service oriented organizations. Thus surely cultural skill is most important in SOA.

9.List down the advantages of Microservices Architecture?

Answer:

* Improved fault isolation: Error handling will be easy for granular level services and larger applications will remain unaffected by the failure of a single module.
* Scalability: It is easier for the development teams to scale up and down based the requirement on a specific component.
* Mix of technologies: Services can be written with different technologies. This allows each team to select the technologies that best fit for their functionality.
* Small codebase: Smaller codebase and scope means deployment of the functionality will be faster and continuous deployment can be achieved.
* Agility: In microservice architecture each service implement only one functionality. Then it will be easy to release code, deploy, scale and test each service separately.
* Isolation: Microservices are profitable due to their isolation. If one component fail developers have the option to use another service and application continue to run independently

10.What are the best practices to design Microservices?

Answer:

* Independently develop and deploy services
* Treat servers as Stateless (which does not store the state of the server. Each client request will be unique)
* Keep all code at a similar level of maturity and stability
* Have separate data storage
* Pair the right technology with right microservice
* Use the benefits of REST APIs
* APIs should be simple and effective
* Build dedicated teams
* Monitor everything
* Manage all logs in one place
* Breakdown migration into steps
* Use Automation for independent deployment
* Continuous delivery

11.How does Microservice Architecture work?

Answer:

Microservices are completely isolated from each other and operate in their own environment. Individual applications communicate with each other through specific interface.

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

Answer:

|  |  |
| --- | --- |
| Pros | Cons |
| Error handling will be easy for granular level. | Communication between services is complex |
| It is easier for the development teams to scale up and down based the requirement on a specific component. | More services mean more resources. Then multiple database and transaction management can be difficult |
| It will be easy to test each services separately. | Global testing is difficult. |
| Each services are granular and independent functionality, deployment will be easy. | Increases the complexity of deploying the entire application at once |
| Services can be written with different technologies. This allows each team to select the technologies that best fit for their functionality | Refactoring application across multiple services is harder in microservices |
| Microservices are great for large companies | Microservices can be slower to implement and complicated for small companies |

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

Answer:

|  |  |  |
| --- | --- | --- |
| Monolithic | SOA | Microservices |
| Different components of an application are combined to a single program in a single platform. All functionalities are managed in one place | It is an architecture model implementing loosely coupled services that can communicate with each other. | It is more granular architecture focus on building autonomous components. These granular components are connected together by API’s |
| It is comfortable for small teams to work with | Components can be reused in multiple applications without influencing other services | Focuses mainly on business priorities and capabilities |
| Developers do not need to deploy changes separately as they can do it once. | Independent services can be developed and deployed separately | Small single services can be built, tested, and deployed independently. Since a deployment unit is small, it facilitates and speeds up development and release |
| Components are interconnected and interdependent | Each services are self-contained and independent that are connected to the consumers through Enterprise Service Bus | Each components are independent and are connected through API |
| Code base are massive and difficult to understand and modify. | SOA services are easier to test and debug as they are independent services | Smaller codebase and scope. |
| Limited agility | Agility | Increased agility |
| Mainly used in startup companies | Best suited for complex enterprise systems such as bank. | Architecture is used when you have multiple experienced teams and when the app is complex |

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

Answer:

* For the microservices system to be effective it require expert skills in Development and Operation(DevOps).
* When a problem arises in the application it will be challenging for the developers if they do not have the tracking path of a specific request.
* As the number of microservices increases, managing them gets more challenging.
* Due to different services and their integration in a microservice, global testing in the environment will be complex
* Dependency management across different services is very important because cyclic dependency can cause a serious problem if they are not identified and resolved accurately.
* While designing microservices itself maximum possible failure need to be identified and error handling should be added, Otherwise resolution of issue will be difficult later.

15.What are the characteristics of Microservices?

Answer:

* Microservice architectural approach is a modular approach to develop a single application as a suite of small services.
* Microservices are described through well-defined service contract.
* Each services can have independent code base
* Microservices are composeable. This can be achieved through service orchestration or service choreography.
* Services can use different language and technology
* Microservices are independent and loosely coupled
* Each services can be tested in isolation independent of other services.
* In microservice it attempt to eliminate the dependencies on other microservices
* Each service can implement independent security mechanism
* Services use REST API to communicate with each other
* Each service can choose the best technology for its use, no need to stick into one framework throughout the entire application
* Clear separation of stateful and stateless servers
* Microservices are reusable business services