Client-Server Architecture:

1. What is client-server architecture, and how does it work?

Client server architecture maintains the request and response according to the client request.

Client sends the service as a request to the server the server takes that and process it and give the required services after authentication.

1. Explain the roles of the client and server in this architecture.

Client side is nothing but the request propagator that is the one who need services in normal scenario user and their browser considered as the client side.

Server side is the high performance computer it process the millions of request and make the request fulfilled.

1. What are the benefits of using a client-server model for building applications?

It gives the performance without the server we can’t able to fullfill the client request 24/7 but with this server we can, and another thing is scalability the server is capable of processing many process at a time.

1. Can you give an example of a real-world application that follows a client-server

architecture?

Gmail

E-Commerce

RESTful API (Representational State Transfer):

1. What is a RESTful API, and what are its key principles?

Representational State Transfer API is the popular API that we use for web applications

It acts as a communicator between the client software and the Database to retrive the informations.

KEY PRINCIPLES IS:

Stateless

Resources

JSON Format

Stateless communication that is not storing the clients’s past request informations

1. What are the main components of a RESTful API request?

HTTP methods

URL Endpoint

HEADERS

1. Describe the common HTTP methods used in RESTful APIs (GET, POST, PUT, DELETE) and their purposes.

GET- Used to fetch the datas from the database

POST-Used to Create a new data in the database

PUT-Used to Update the existing data in the database

DELETE-Used to delete the record in the database.

1. What is the significance of status codes (e.g., 200, 404, 500) in RESTful API responses?

200- All ok working fine

404-Page is not available

500-Internal Server Error

1. How is statelessness achieved in RESTful APIs, and why is it important?

Statelessness means that for each and every request from the client side there should be necessary information that is used to fetch the data for the client,

It is important because of the speed and scalability.

1. Explain the concept of resource identification in RESTful APIs and how it's reflected in URL design.

UNIQUE URL

HTTP Methods

Hiearchical Strucutre

Query Parameters

Client-Server Communication:

a. How do clients and servers communicate in a client-server

architecture?

Step1:Client Sends the request

Step2:Server receives the request

Step3:Server process the request

Step4:Server sends response

b. What are some common communication protocols used for client-server

interactions?

**HTTP**

**HTTPS**

**SMTP**

**POP3**

**IMAP**

**FTP**

**SNMP**

**RPC**

c. Describe the steps involved in making a typical request from a client

to a server and receiving a response.

**Client Initiates Request**

**DNS Resolution**

**Establishing Connection**

**Sending the Request**

**Server Receives the Request**

**Processing the Request**

**Generating the Response**

**Sending the Response**

**Client Receives the Response**

**Handling the Response**

**Closing the Connection**

1. What is the role of HTTP headers in client-server communication?

HTTP headers are like “Notes” attached to the request and response in the client server architecture.

Web Services:

1. What are web services, and how do they relate to RESTful APIs?

Web services are the protocols that allows the applications to exchange the data between applications. RESTful API works on this concept it acts as an interface between the applications.

1. What differentiates SOAP (Simple Object Access Protocol) web services from RESTful web services?

SOAP- Simple Object Access Protocol, It is a old technology used for data transfer over web services, It uses Xml file format and follows either HTTP or SMTP protocol to make this happen.

1. What are the advantages of using RESTful web services over other types of web services?

Simplicity, Flexibility, Performance, Platform Independence.

Data Formats:

1. What are the commonly used data formats for exchanging data in RESTful APIs?

JSON, XML, TXT

1. Compare and contrast JSON and XML as data interchange formats in the context of RESTful APIs.

JSON is used for lightweight, simplicity, performance while XML is used for hierarchical data and compatibility requirements.

1. When would you choose JSON over XML or vice versa?

JSON is used for lightweight, simplicity, performance while XML is used for hierarchical data and compatibility requirements.

Security Considerations:

1. What security mechanisms can be employed to secure a RESTful API?

Security mechanisms for securing a RESTful API include authentication (e.g., JWT, OAuth), authorization (e.g., RBAC), HTTPS, input validation, rate limiting, and encryption.

1. How does authentication work in a RESTful API? What are some common authentication methods?

Authentication is like showing Id to enter into restricted area

User name and password

Token -based authentication

Oauth

API kEY

HTTP

JWT

1. Explain the concept of CORS (Cross-Origin Resource Sharing) and how it affects API security.

CORS (Cross-Origin Resource Sharing) is a security feature that allows servers to control which origins can access their resources, thereby protecting APIs from unauthorized cross-origin requests.