



**BAHRIA UNIVERSITY (KARACHI CAMPUS)**  
**Software Design & Architecture (SEN-221)**  
**ASSIGNMENT # 1 – Spring 2022**  
**Based on: CLO-2**

Class: **BSE-4B**Submission Deadline: **03<sup>rd</sup> April 22**Course Instructor: **ENGR. MAJID KALEEM**Max Marks: **05**

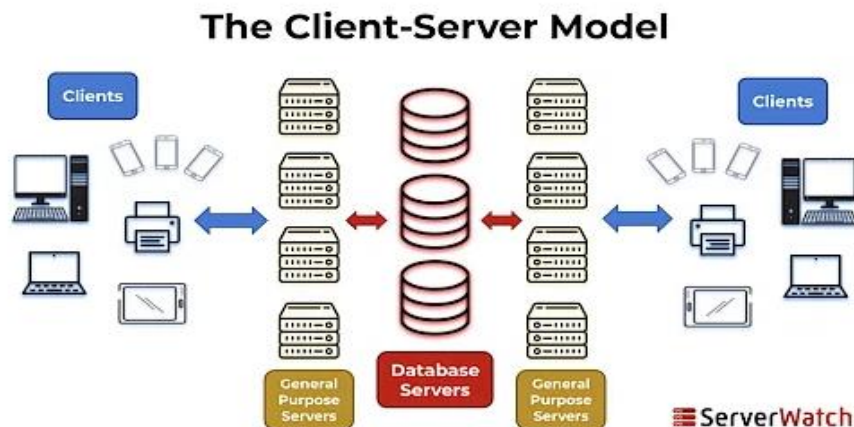
1. Suppose you have to design & develop an online LMS System. You may consider various architectural and design style to address this situation. Keep LMS in mind, describe (separately) each of the architectural styles given below with respect to the following questions:
1. **What** are they?
  2. **Where** are they used?
  3. **Why** are they used?
  4. **How** are they used / implemented?
- a) Client/Server Architecture
  - b) Component-Based Architecture
  - c) MVC Architecture
  - d) Cloud Application Architecture

---

**SOLUTION:**

**a) CLIENT/SERVER ARCHITECTURE:**

The following client-server diagram shows the basics of the architecture:



## **WHAT?**

A client/server architecture refers to a system that hosts, provides, and manages most of the resources and services requested by clients. In this model, all requirements and services are delivered over the network, also known as the network computing model or client-server network.

Client/server architecture, also known as the client/server model, is a network application that divides tasks and workloads between clients and servers that reside on the same system or are connected by a computer network.

## **WHERE?**

Client/server architecture is used in a wide variety of computer systems and applications, including:

- Web application
- Mail system
- Database management system
- File sharing system

Overall, client/server architecture is used in any system that requires centralized data storage, processing, and management, and where multiple clients need to access and interact with a server simultaneously.

## **WHY?**

Client-server architectures are used in software systems because they provide a scalable, efficient, and secure environment for distributing processing tasks across different computers or devices on a network.

Here are some of the main reasons why client/server architecture is used:

- Scalability
- efficiency
- Safety

## **HOW?**

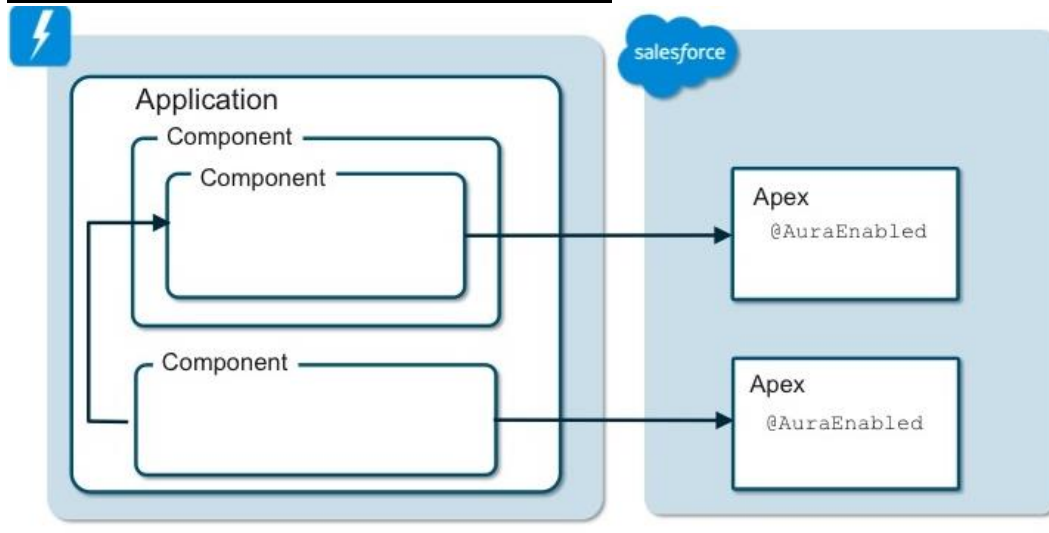
A client-server architecture is used in the following ways:

- First, a client sends a request via a network capable device
- The network server then accepts and processes the user's request.
- Finally, the server delivers the answer to the client.

### **Example:**

file server: Use a file server if you want to store your files in cloud-based services such as Google Documents or Ms. Office. A file server is a central storage location for files accessed by many clients.

### **b) COMPONENT-BASED ARCHITECTURE:**



### **WHAT?**

Component-based architecture is a software design and development approach that focuses on combining off-the-shelf modular components to create software systems. Each component is designed to be self-contained, reusable, and interchangeable with others. This architecture promotes separation of concerns and modularity, making software applications easier to develop, maintain, and scale.

### **WHERE?**

Component-based architecture is used in software application development that focuses on building and integrating independent, reusable software components. This approach is commonly used in large enterprise applications, web applications, and complex systems where components can be developed and tested separately and then seamlessly integrated into the application.

### **WHY?**

Component-based architecture is used for several reasons:

- Flexibility
- Scalability
- Maintainability
- Reusability

Overall, component-based architecture promotes code reuse, scalability, maintainability, and flexibility, making it an ideal choice for modern software development projects.

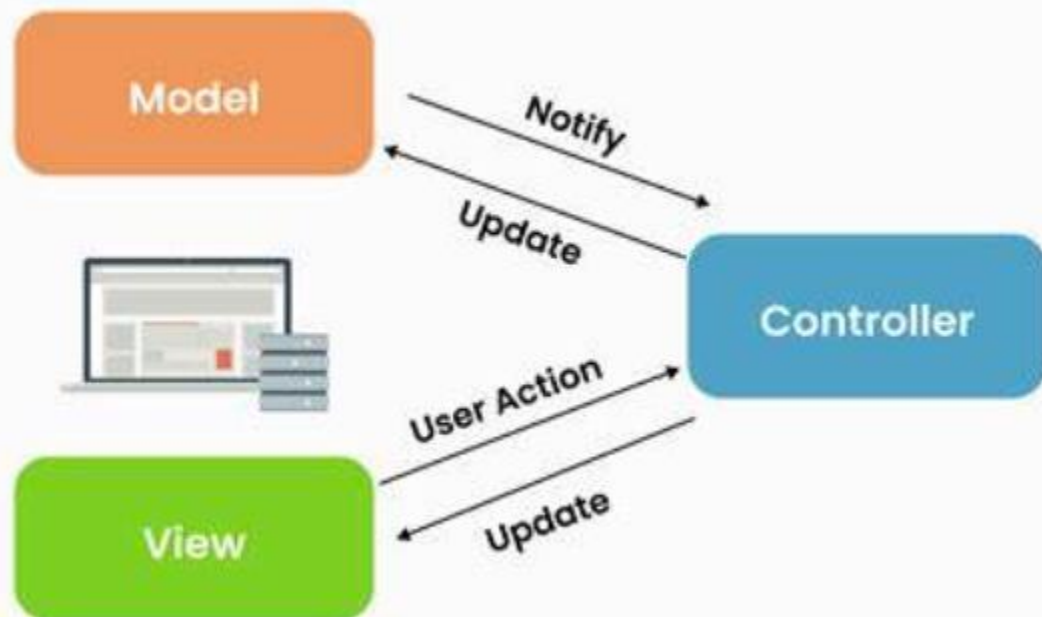
### **HOW?**

Component-based architecture is used to develop applications by dividing them into individual modular components that can be developed, tested, and maintained independently. These components are self-contained and reusable across multiple applications, simplifying the development and maintenance of complex applications. This architecture promotes loose coupling between components and allows flexibility and scalability in application design.

### **EXAMPLE:**

One of the prime examples of component-based architecture is the Angular framework used to build web applications. Angular is built on a component-based architecture, where each component represents standalone functionality that can be reused across multiple sites or applications. Each component has its own template, style, and logic, making it easier to develop, test, and maintain. This architecture promotes modularity and separation of concerns, making it easier for developers to build complex applications.

### **c) MVC ARCHITECTURE:**



### **WHAT?**

MVC architecture turns complex application development into a more manageable process. This allows multiple developers to work on the application simultaneously. MVC has three main components:

- model
- View

- controller

### **WHERE?**

The MVC design and architecture pattern is a great tactic for web application development. Today, many companies decide to develop such applications based on his MVC architecture.

### **WHY?**

The reasons MVC is used are:

- Organize large web applications
- Easily modifiable
- Accelerate the development process
- Easy planning and maintenance
- Return data without formatting
- Multiple views
- Good platform for search engine optimization

### **HOW?**

MVC stands for Model, View, and Controller. MVC splits an application into three components Model, View, and Controller which are us follow.

### **MODEL:**

Models represent data. Models deal with data states. A C# class is used to describe the model. Model elements store data retrieved from the database.

### **VIEW:**

Display model data to customers and allow them to customize it. View in ASP.NET MVC is HTML, CSS, and a proprietary syntax that simplifies the communication between models and regulators.

### **CONTROLLER:**

A request handle has a controller. Controllers manage customer requirements. Clients typically use views to initiate HTTP requests that are handled by controllers. The controller processes the request and returns the appropriate view in response.

### **EXAMPLE:**

Each car consists of 3 main parts.

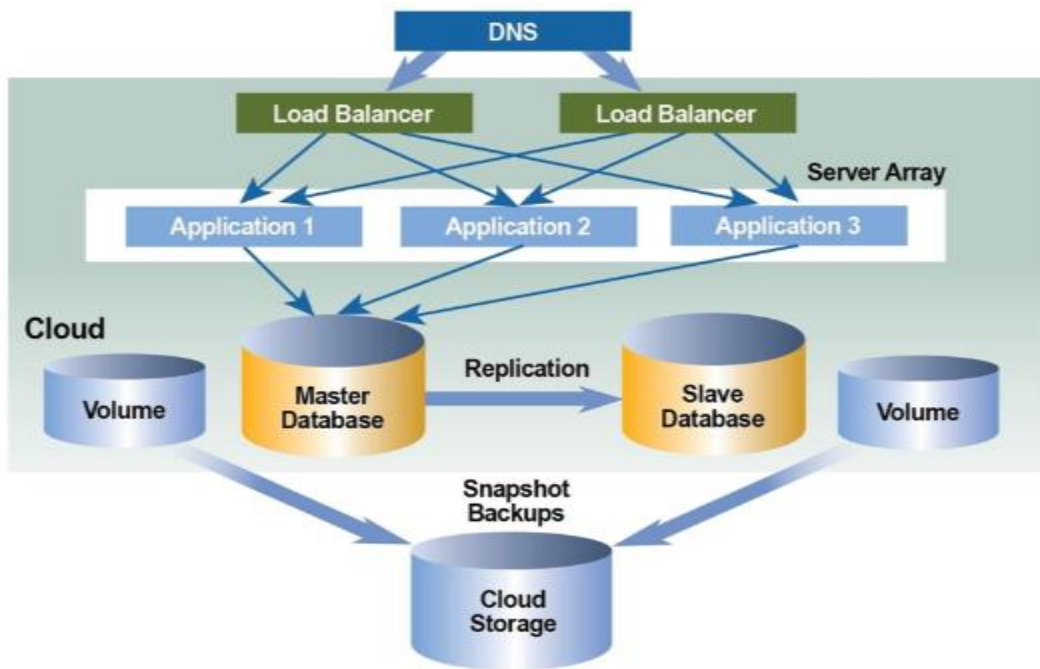
View = User Interface:

(Transmissions, fairings, brakes, etc.)

controller = system

model = storage (petrol, gas or diesel tank)

d) **CLOUD APPLICATION ARCHITECTURE:**



**WHAT?**

A cloud architecture is a way of combining technology components to create a cloud that pools resources with virtualization technology and distributes them across a network.

Cloud Application Architecture is a framework for designing and deploying applications that run on cloud infrastructure. This architecture is specifically designed to take advantage of cloud computing benefits such as scalability, flexibility, and high availability.

**WHERE?**

Cloud-based architectures are well suited for building modern web applications, deploying web applications, connecting to new databases, and analyzing user behavior. A traditional cloud-based application architecture includes load balancers, web servers, application servers, and databases.

**WHY?**

Cloud Application Architecture is used for many reasons, including:

**Scalability:**

Cloud application architecture makes it easy to scale resources up or down as needed. This means your application can handle sudden spikes in traffic without incurring downtime or performance degradation.

**Reliability:**

Cloud providers typically offer high availability and redundancy features that help maintain access to your applications, even in the event of hardware failures or other issues.

**HOW?**

Cloud Application Architecture is used to build scalable, reliable, and cost-effective applications that can be accessed from anywhere in the world, enabling companies to innovate quickly and meet changing market demands.

**EXAMPLE:**

An example of a cloud application architecture is the serverless architecture used by the Amazon Web Services (AWS) Lambda service. In this architecture, applications are built with stateless functions triggered by specific events. B. Modification of Data or User Interactions. Features run on a cloud platform and are managed by a cloud provider, allowing developers to focus on building applications without worrying about managing infrastructure.

## **REFERENCES:**

[Client Server Architecture Article](#)

[Component Based Architecture](#)

[MVC Architecture](#)

[Cloud Application Architecture](#)