

# Client- Server Architecture

Client-server architecture is a computing model that separates the client (typically a user interface) from the server (which manages data and resources). It is a distributed computing model where tasks and workloads are divided between the client and server. This architecture enables efficient resource sharing, centralized management, and separation of concerns between clients and servers.

IV by IT Vedant



# Key Components of Client-Server Architecture

## Client

The client is the computer or application that initiates requests for services or resources. It often refers to the user interface portion of the software, which interacts with the user and communicates with the server.

## Server

The server is the computer or system that provides services or resources to client requests. It manages and processes the requests, retrieves data from databases or other sources, and sends the appropriate responses back to the client.

# How Client-Server Architecture Works

**1**

## **Step 1: Client Request**

The client initiates a request for a specific service or resource by sending a message to the server.

**2**

## **Step 2: Server Processing**

The server receives the request and processes it based on its capabilities and available resources.

**3**

## **Step 3: Task Execution**

The server performs the requested task or retrieves the requested data.

**4**

## **Step 4: Server Response**

The server sends a response back to the client with the requested information or confirmation that the task was completed.

**5**

## **Step 5: Client Reception**

The client receives the response from the server and typically displays or utilizes the data or result.

# Key Features of Client-Server Architecture

**1**

## **Centralized Management**

Servers can centrally manage and control access to resources, data, and applications.

**2**

## **Scalability**

Additional clients can be added without significantly impacting the server's performance, and servers can be upgraded or added to handle increased workloads.

**3**

## **Resource Sharing**

Multiple clients can access and share resources managed by the server.

**4**

## **Security**

Access control mechanisms can be implemented on the server to ensure data integrity and privacy.

**5**

## **Platform Independence**

Clients and servers can be implemented on different hardware and software platforms, as long as they follow the same communication protocols.



# Examples of Client-Server Architecture



## Web Applications

Web browsers (clients) send requests to web servers, which process the requests and serve web pages or data.



## File Servers

Clients can access and store files on a centralized file server.



## Database Servers

Applications (clients) send queries to database servers, which manage and retrieve data from databases.



## Email Servers

Email clients (e.g., Outlook, Gmail) communicate with email servers to send and receive emails.

Client-server architecture is widely used in various computing environments, from small local networks to large-scale enterprise systems and internet-based applications, enabling efficient resource sharing, centralized management, and separation of concerns between clients and servers.