

Lecture 6:

Client/Server Architecture

Relation to CLOs

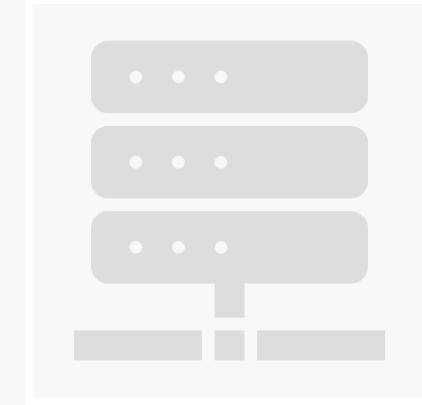
The following Lecture is related to the CLO_2, which is as:

CLO_2: Evaluate the several web technologies and application architectures.

Connection-Oriented Communication



Internet Service is connection oriented.



connection-oriented service requires two applications.

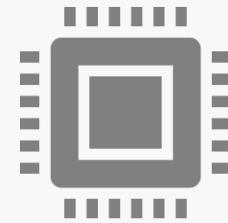
Basic Client/Server Architectures



The client/server architecture was developed to deal with computer environment in a large number of PCs, workstation, file server.

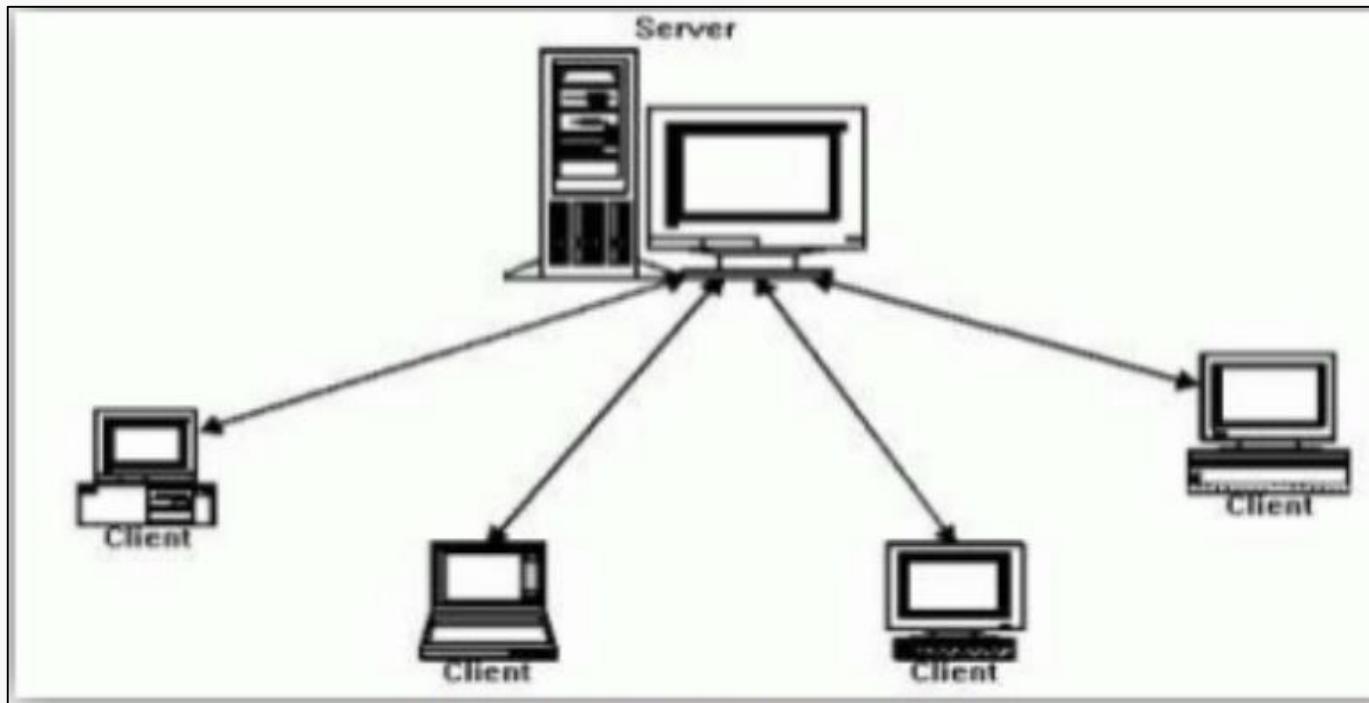


A client is a user machine that provides user interface capabilities and local processing.



A server is a system containing both hardware and software that can provide services to the client machines.

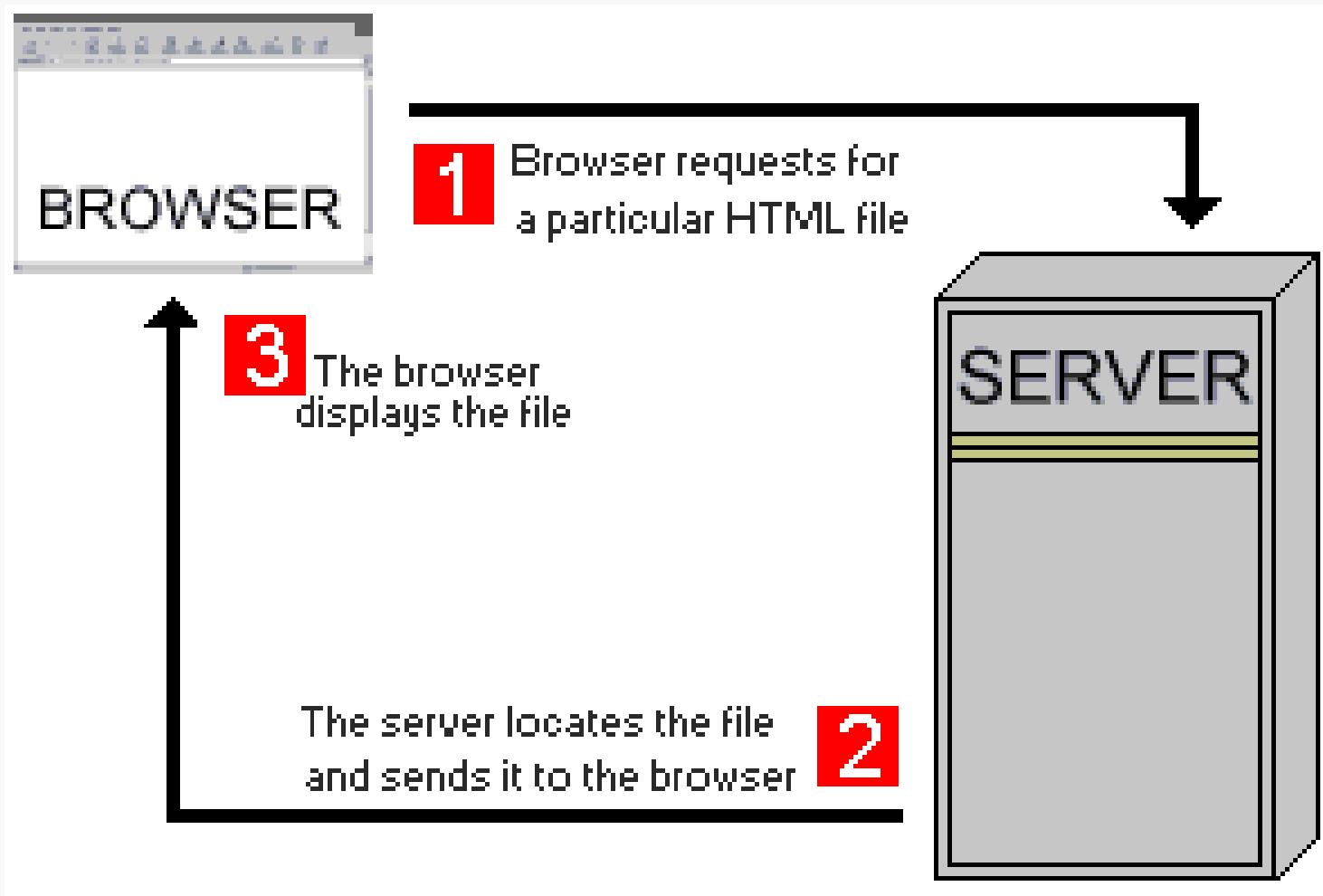
Basic Client/Server Architectures



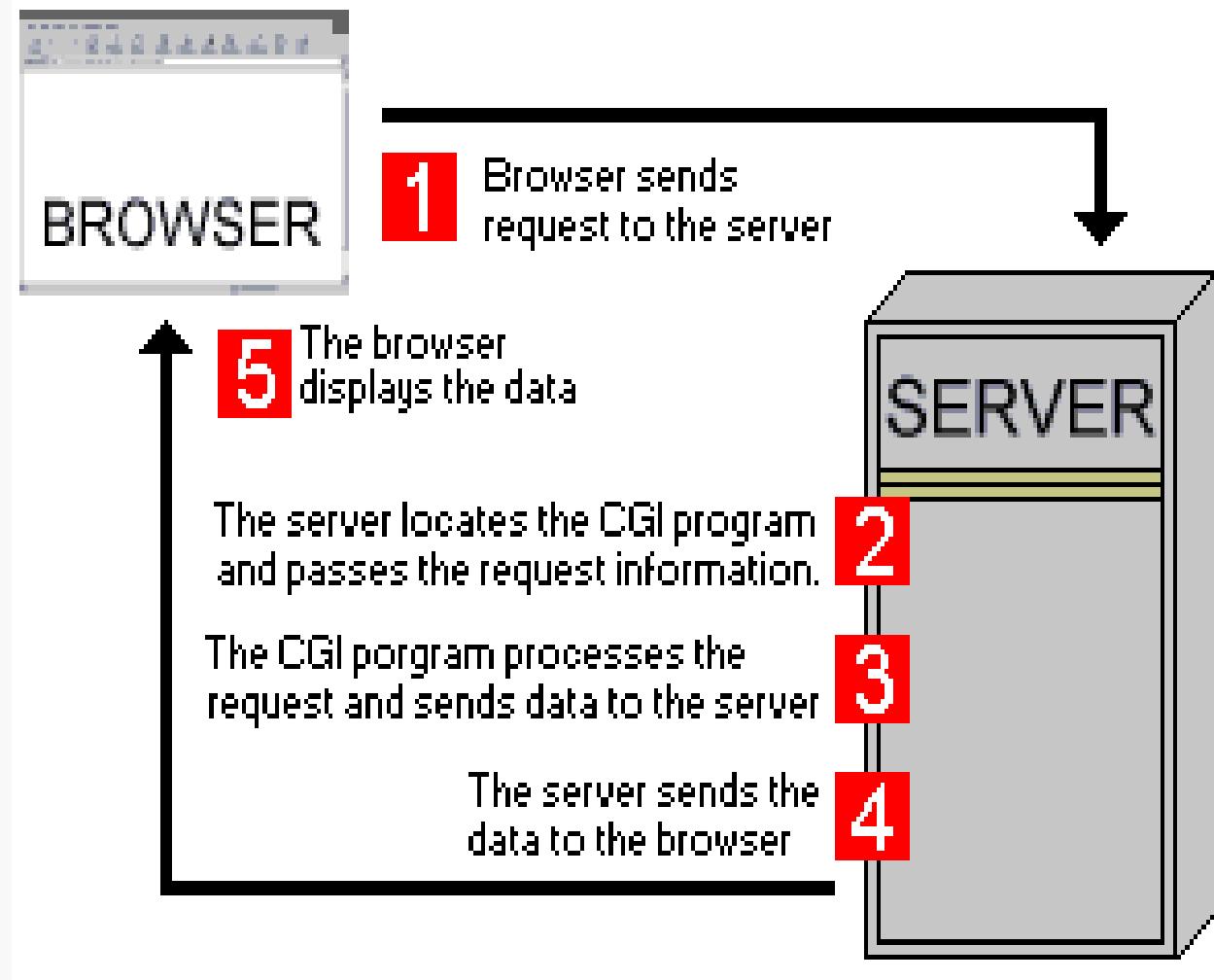
Client/Server architecture in the web

- World Wide Web (WWW) or Web revolves around the client/server architecture
- Client computer system uses Browsers (like Internet Explorer, Netscape Navigator, Mozilla etc.) **to interact** with Internet servers using Protocols
- **Protocols** help in the accurate transfer of data - requests from browser & responses from server
- Commonly used protocols :
 - ❖ *HTTP (Hyper Text Transfer Protocol)*
 - ❖ *FTP (File Transfer Protocol)*
 - ❖ *SMTP (Simple Mail Transfer Protocol)*

Client / Server Model Static HTML Pages



Client / Server Model Dynamic HTML Pages



A **CGI program** is any **program** designed to accept and return data that conforms to the **CGI** specification. The **program** could be written in any **programming** language, including C, Perl, Java, or Visual Basic. **CGI programs** are the most common way for Web servers to interact dynamically with users.

Client-Server Model of Interaction

The following table summarizes the interaction.

Server Application	Client Application
Starts first	Starts second
Does not need to know which client will contact it	Must know which server to contact
Waits passively and arbitrarily long for contact from a client	Initiates a contact whenever communication is needed
Communicates with a client by both sending and receiving data	Communicates with a server by sending and receiving data
Stays running after servicing one client, and waits for another	May terminate after interacting with a server

Characteristics of Clients and Servers

Client Software:

- Is an arbitrary application program that becomes a client temporarily when remote access is needed, but also performs other computation locally.
- Is invoked directly by a user and executes only for one session.
- Runs locally on a user's computer.
- Actively initiates contact with a server.
- Can access multiple servers as needed, but actively contacts one remote server at a time.
- Does not require special hardware or a sophisticated operating system.

Characteristics of Clients and Servers

Server Software:

- Is a special-purpose, privileged program dedicated to providing one service, but can handle multiple remote clients at the same time.
- Is invoked automatically when a system boots, and continues to execute through many sessions.
- Runs on a shared computer.
- Waits passively for contact from arbitrary remote clients.
- Accepts contact from arbitrary clients but offers a single service.
- Requires powerful hardware and a sophisticated operating system.

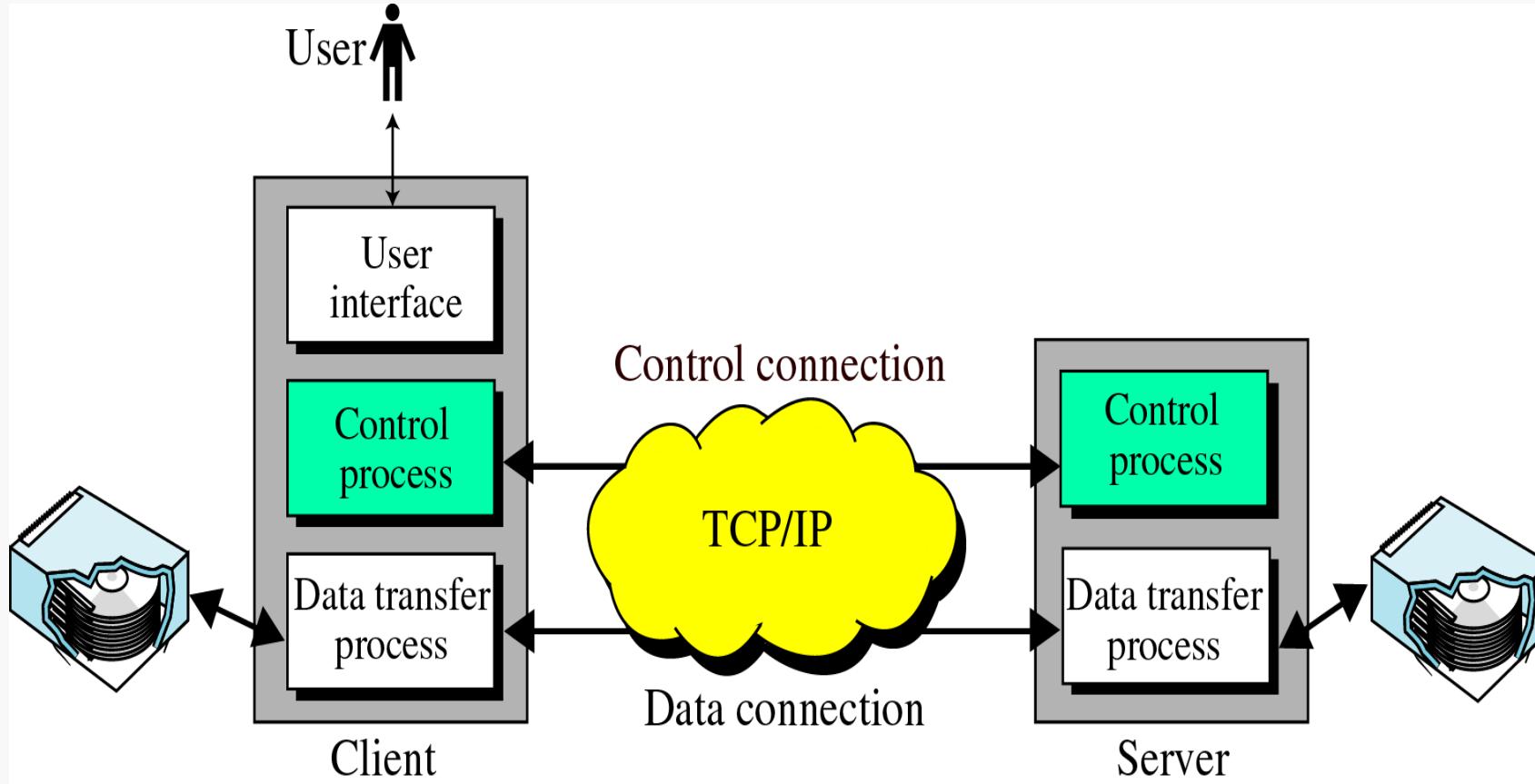
Requests, Response, and Direction of Data Flow

- Information can flow in either or both directions between a client and server. Although many services arrange for the client to send one or more requests and the server to return responses, other interactions are possible.

Types of Servers

- File Server
- Database Server
- Application Server
- Web Server
- Object Server
- Others – Chat Server, Fax Server, Mail server, News Server ...

File server



File server

❖ Objectives of File Server

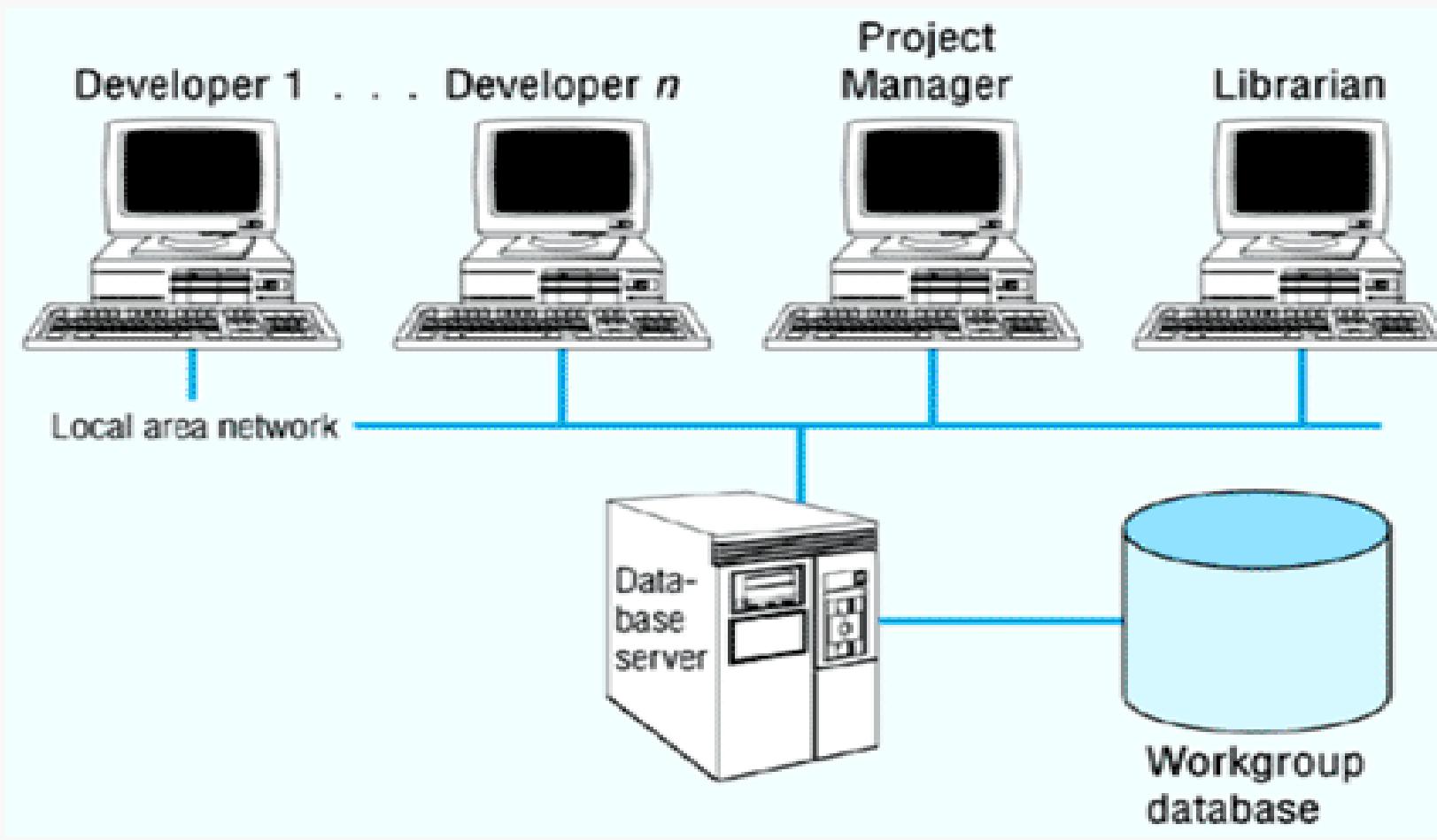
- *To promote sharing of files (computer programs and/or data)*
- *To encourage indirect or implicit (via programs) use of remote computers*
- *To shield a user from variations in file storage systems among hosts*
- *To transfer data reliably and efficiently*

Database Server

- A computer in a LAN dedicated to database storage and retrieval
- The database server holds the Database Management System (DBMS) and the databases
- Upon requests from the client machines, it searches the database for selected records and passes them back over the network

Database Server - Example

Software Project Team – sharing a database

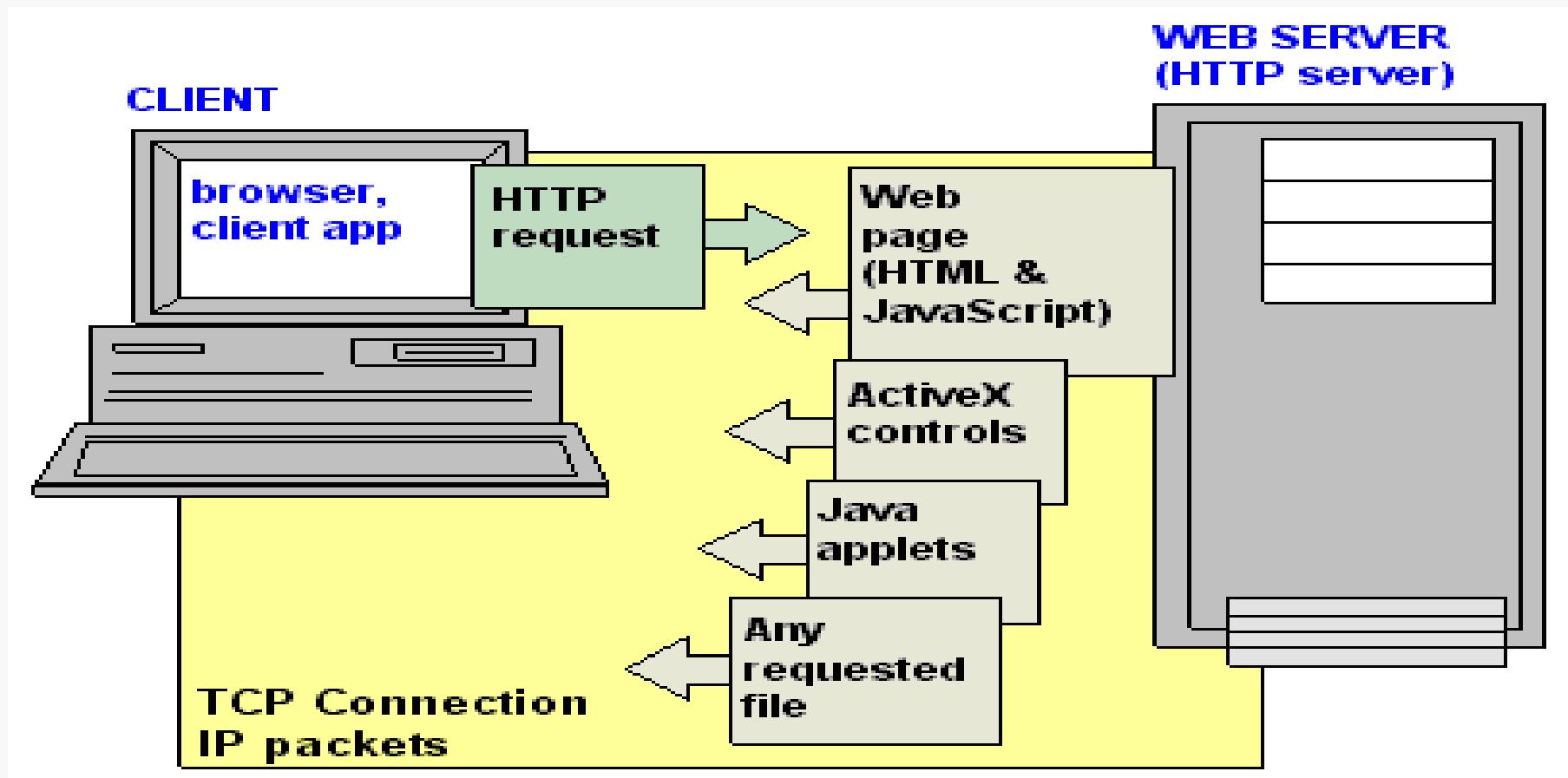


Application Server

- Application server runs the programs and processes the data
- Application servers are typically used for complex transaction-based applications
- Application server performs the **business logic functions** and some data access
- An application server provides **middle tier processing** between the user's machine and the database management system (DBMS)

Web Server

Web Server delivers web pages to browsers



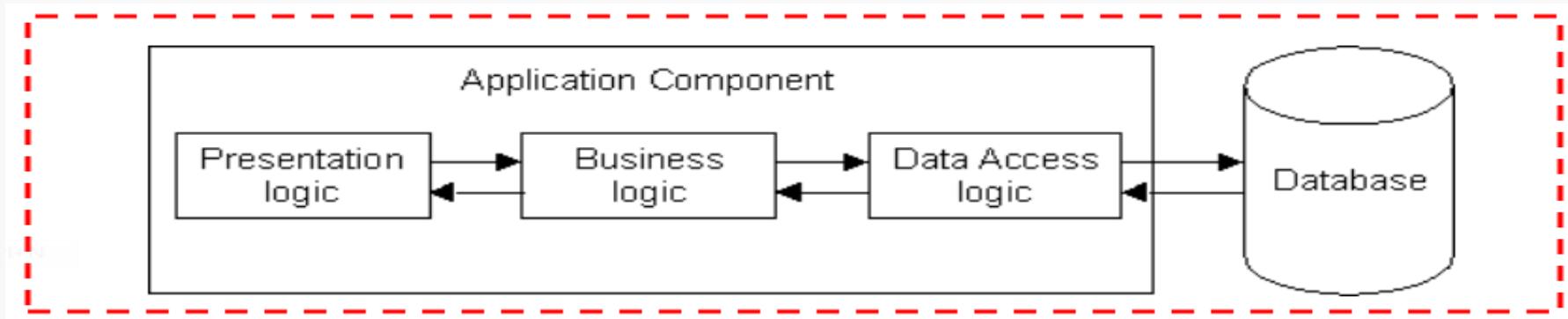
Client/Server Application

- ❖ Has three functional units:
 - *Presentation logic or user interface (for example, ATM machines)*
 - *Business logic (for example software that enables a customer to request an account balance)*
 - *Data (for example, records of customer accounts)*
- ❖ Functional units can reside on the client or on multiple servers
- ❖ Middleware is the software that connects / communicates between client and server

The Tiers (layers)Frontend

- ❖ *Frontend (presentation) tier:*
 - *How the data should appear to the user and how the user should interact with the interface*
- ❖ Application logic tier:
 - *Data Processing e.g. Business Logic & calculations*
 - *Can be referred to as: Business logic/rules.*
- ❖ Data(resource) management tier:
 - *Deals with the different data sources of the Information Systems; essentially it can be considered as a Database Management System*

One - Tier Architectures



Monolithic Systems:

All 3 layers are on the same machine: Presentation, application logic, and resource management were merged into a single tier

All code and processing kept on a single machine

One - Tier Architectures

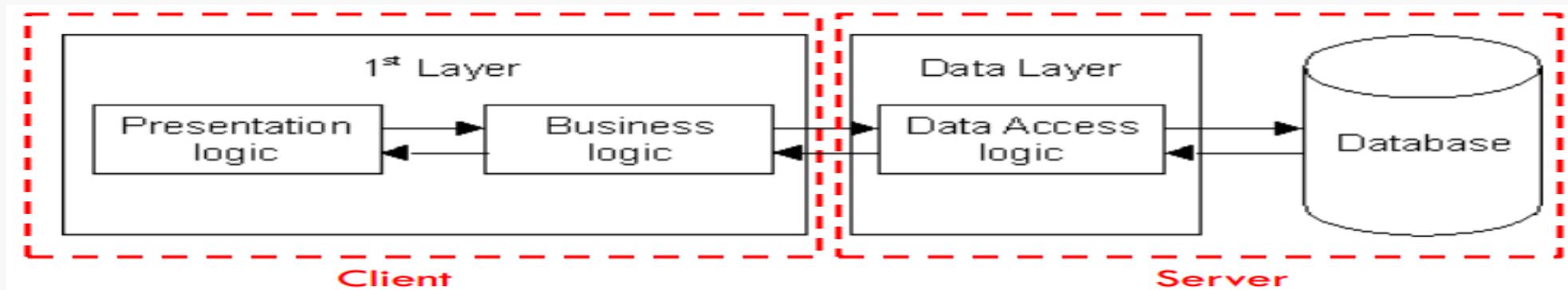
■ Advantages:

- *easy to optimise performance*
- *no compatibility issues between layers*
- *no client development, maintenance....*

■ Disadvantages:

- *monolithic pieces of code and hard to modify*
- *Old system so lack of qualified programmers for these systems*

Two-tier Architecture



This is called **two-tier** architectures because the software components are distributed over two systems: client and server

- **Database runs on Server**
- **Presentation and logic layers still tightly connected (coupled)**

Two-tier Architecture

Advantages

- Portability

المقصود بها تشغيلها على مختلف المنصات

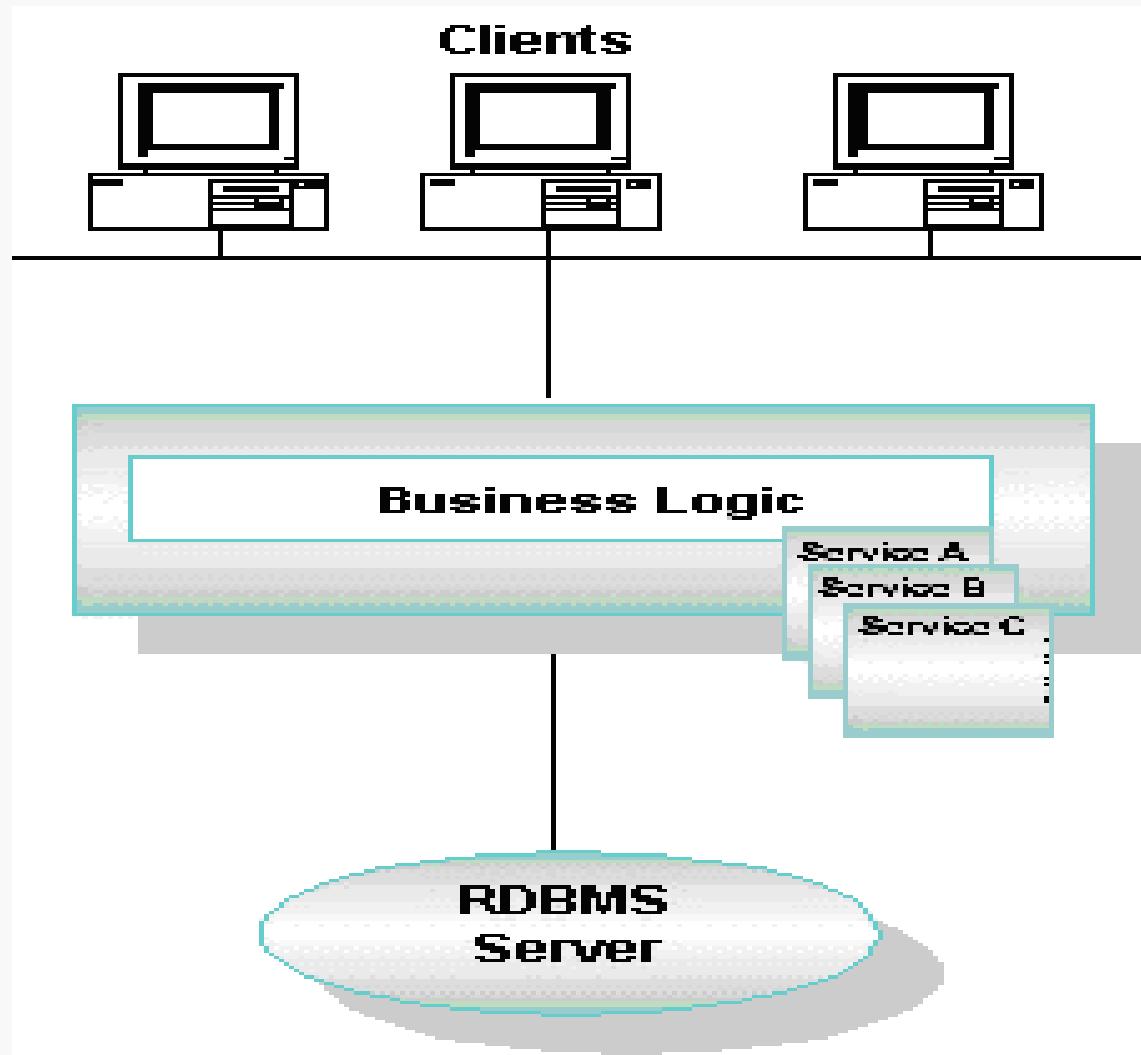
Disadvantages

- Scalability is not great
- legacy system problems

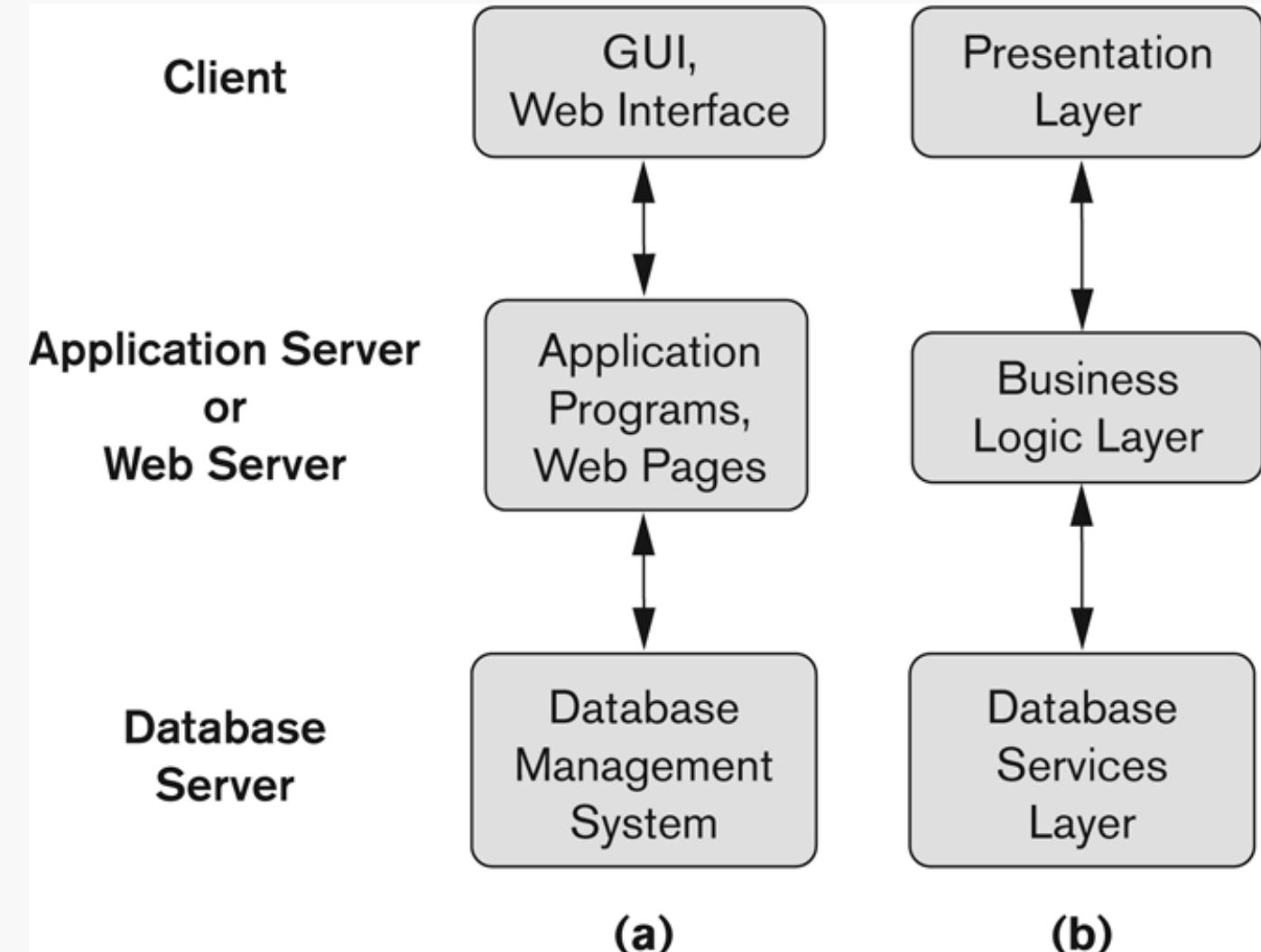
Three-tier architecture

- The intermediate layer or **middle layer** is sometimes called the **application server** or **Web server**
- Three-tier Architecture Can Enhance Security:
 1. Database server only accessible via middle tier
 2. Clients cannot directly access database server

Three -Tier Architecture



Three-tier client-server architecture



Three-tier architecture

- The **presentation layer** displays information to the user
- The **business logic layer** handles intermediate rules and constraints before data is passed up to the user or down to the DBMS
- If the bottom layer is split into two layers (a web server and a database server), then it is a **4-tier** architecture (possible to the **n-tier**)

Three-tier architecture

❖ Advantages

- *Increased scalability by distributing the application layer across many nodes*
- *Flexibility*

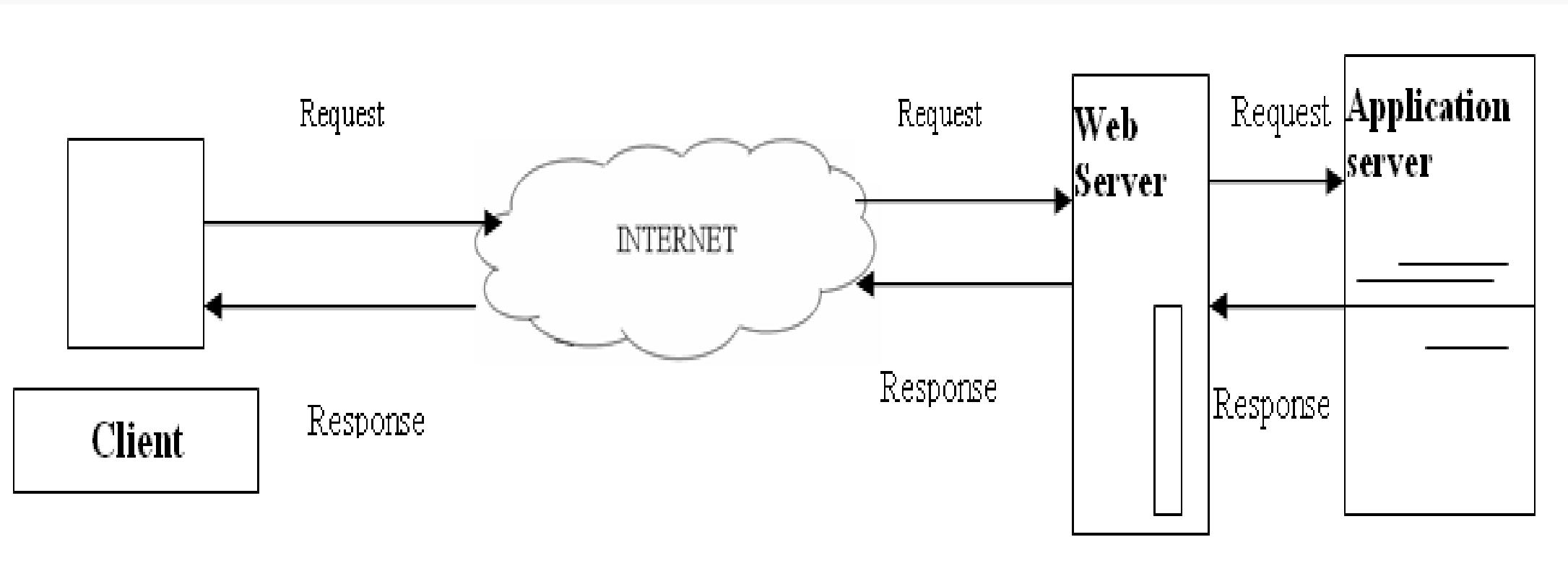
❖ Disadvantages

- *performance loss if distributed over the internet due to restricted bandwidth*
- *problem when integrating of the 3 tiers*

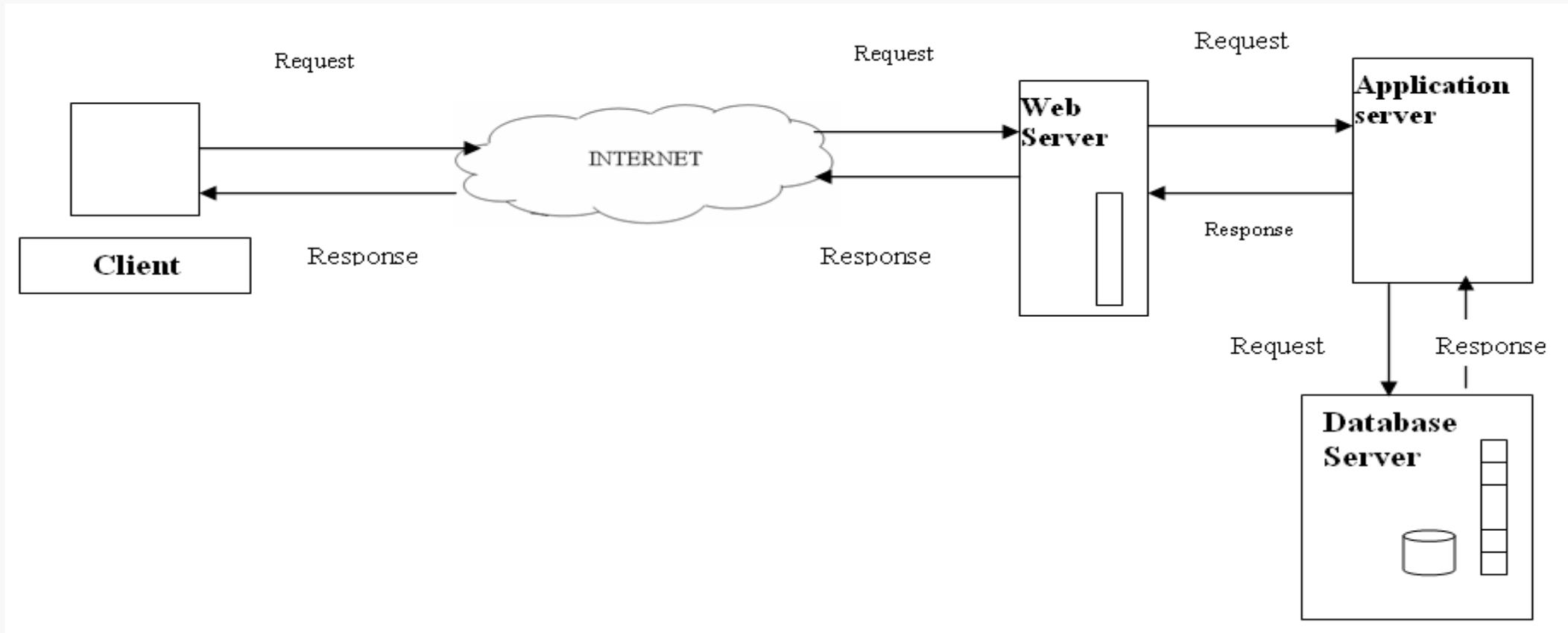
Three -Tier Architecture

- ❖ The Middle Tier can be
 - *A Transaction Processing Monitor*
 - *A Message Server*
 - *An Application Server*
 - *An Object Server*

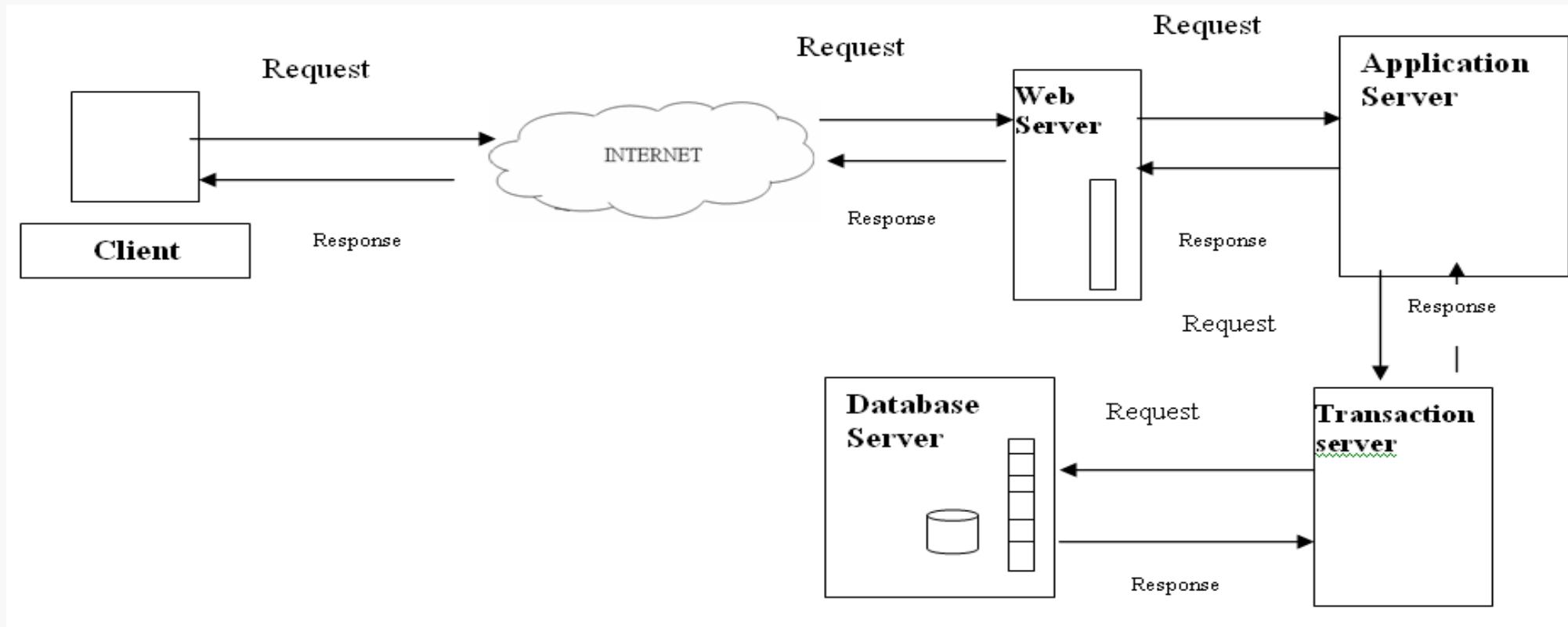
Three -Tier Architecture - Internet



Multi -Tier Architecture



Multi -Tier Architecture



Advantages of Client/Server

- Distribution or sharing of processing load among several independent computers
- Centralized data. Hence easy to administer
- Centralized data storage on server, hence easy to implement security of data access
- Clients with the appropriate permissions may be permitted to access and change data
- Is capable of functioning with multiple different clients each with different capabilities

Disadvantages of Client/Server

- Traffic congestion on the network (number of simultaneous client requests to a server) can cause a server to become overloaded and unable to service client's requests
- Under client-server, should a critical server fail, clients' requests cannot be fulfilled. Hence, lack of robustness is a cause for concern

General benefits of adding tiers: middleware components

Increase in the number of tiers means:

- Increase flexibility, functionality and ways of distribution

However this is offset by (disadvantages):

- Increases communication costs
- Complexity rises leading to higher cost and risk
- Increases in duplication can lead to redundancy.
- The above issues can be lessened or overcome by using **clustering with higher spec machines**; fragment the work to take advantage of the system architecture (parallel (cluster) system). Remember that a cluster is a system with many nodes (computers) that are integrated together in such a way as to behave as a single system.