



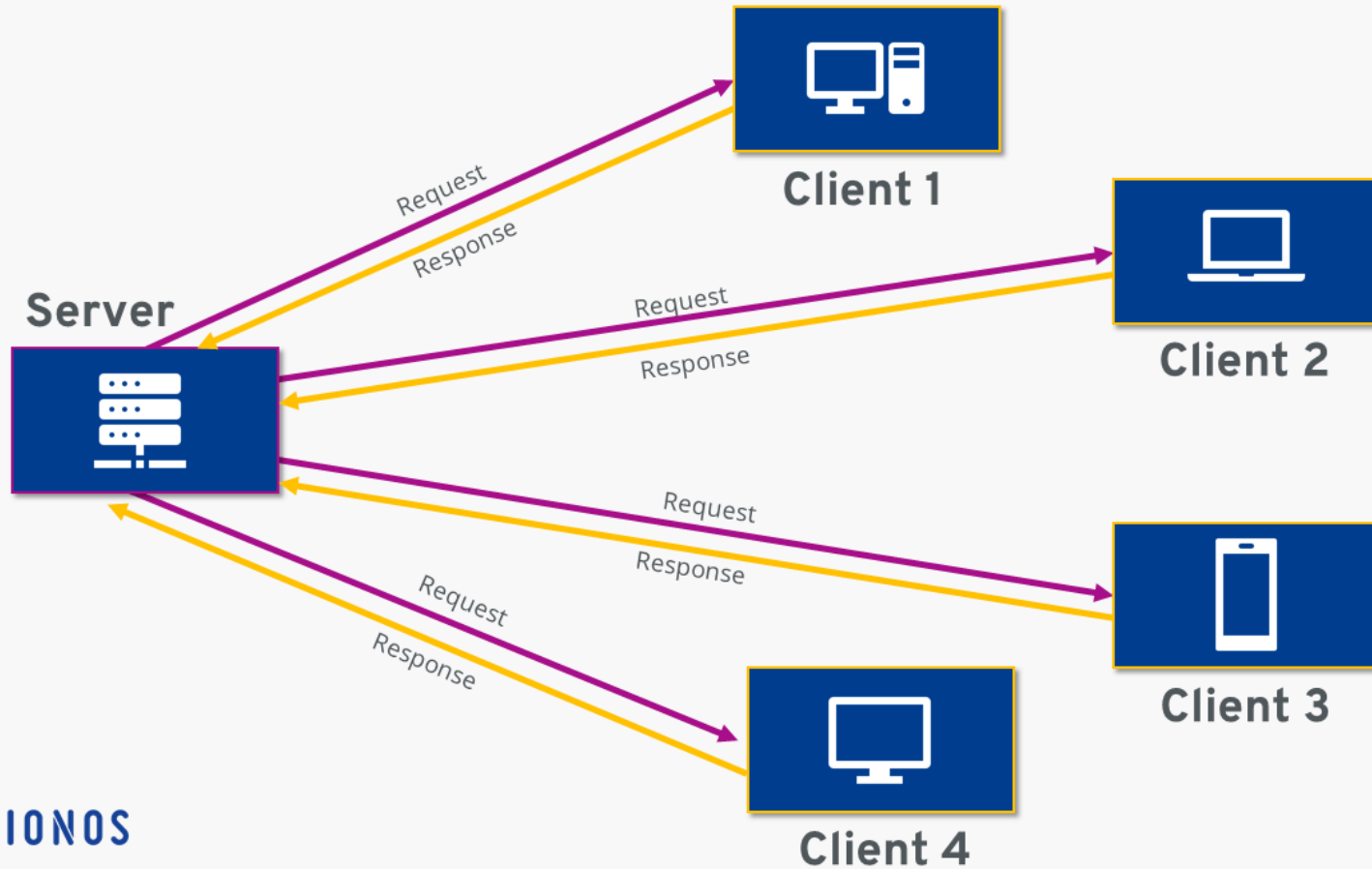
Chapter : 1 Introduction

Prof. Kirtankumar Rathod
Dept. of Computer Science
Indus University

WHAT IS CLIENT/SERVER COMPUTING?

- A Client is any process that requests specific services from the server process.
- A Server is a process that provides requested services for the Client.
- Client and Server processes can reside in same computer or in different computers linked by a network.

Basic Model



A Server for Every Client

- File Server
 - File Server provides clients access to records within files from the server machine.
 - UNIX: Network File Services (NFS) created by Sun Microsystems.
 - Microsoft Windows “Map Drive”
- Print Server
 - This machine manages user access to the shared output devices, such as printers.

A Server for Every Client

- Database Server
 - Database server provides access to data to clients, in response to SQL requests.
- Transaction Servers
 - The data and remote procedures reside on the server.
 - The Server provides access to high-level functions, and implements efficient transaction processing.

A Server for Every Client

- Application Server
 - This machine manages access to centralized application software; for example, a shared database.
- Web Server
 - This machine stores and retrieves Internet (and intranet) data for the enterprise.
- Mail Server
- Fax Server and many more...

Client/Server: Fat or Thin

- A **thin client** is one that conducts a minimum of processing on the client side.
- A **fat client** is one that carries a relatively larger proportion of processing load.
- https://www.webopedia.com/DidYouKnow/Hardware_Software/thin_client.asp

Scenario ...

1. You work for a government / university / corporation with many employees.
2. It is very likely that most of your users have somewhat similar needs in terms of applications and files.
3. All users would most likely want access to Microsoft Word and Excel, your organizations' line of business application(s), your accounting software if you are in the finance department and so on.
4. Further, you have files that need to be accessed by multiple individuals, perhaps simultaneously, perhaps from many different physical locations.
 - How does your organization deal with these requirements?

Client/Server: Fat or Thin

- Fat Server:
 - This architecture places more application functionality on the server machine(s).
 - Opposite is Thin Server

Client/Server: Stateless or Stateful

- A **stateless server** is a server that treats each request as an **independent** transaction that is unrelated to any previous request.
 - example of a stateless server is a World Wide Web server.
- The stateful server remembers what client requested previously and at last maintains the information as an incremental reply for each request.
 - example of stateful server is remote file server.

Servers and Mainframes

- A mainframe was 'larger' in terms of:
 - The raw speed expressed in instructions per second, or cycles.
 - The amount of memory that could be addressed directly by a program.



A pair of IBM mainframes. On the left is the [IBM z Systems z13](#). On the right is the [IBM LinuxONE Rockhopper](#).



Inside an IBM System z9 mainframe

A very natural question at this stage is: How do Client/Server Systems differ from Mainframe Systems?

- In mainframe systems all the processing takes place on the mainframe and usually dumb terminals are used to display the data screens.
- The Client/Server environment provides a clear separation of server and client processes, both processes being autonomous.

Various other factors,

- Application development
- Data manipulation
- System management
- Security
- End user platform

Client/Server Functions

- Managing the user interface.
- Accepts and checks the syntax of user inputs.
- Processes application logic.
- Generates database request and transmits to server.
- Passes response back to server
- Accepts and processes database requests from client.
- Checks authorization.
- Ensures that integrity constraints are not violated.
- Performs query/update processing and transmits responses to client.
- Provide concurrent database access.
- Provides recovery control.

Client/Server Topologies



Fig.1.2: Single Client, Single Server

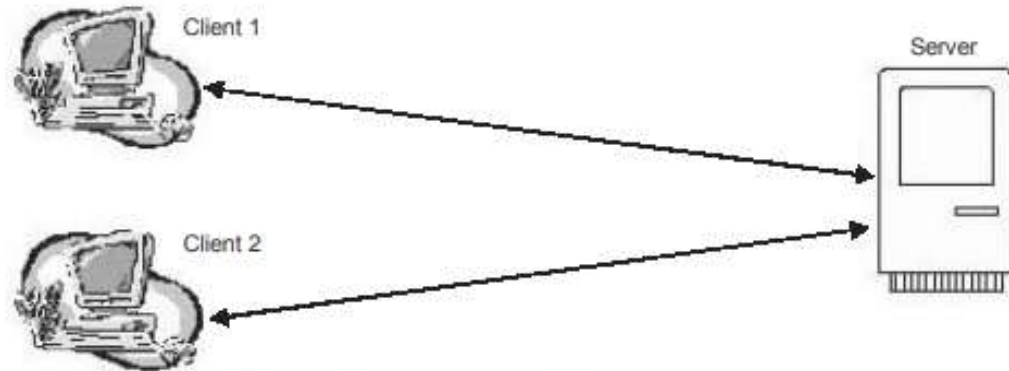


Fig.1.3: Multiple Clients, Single Server

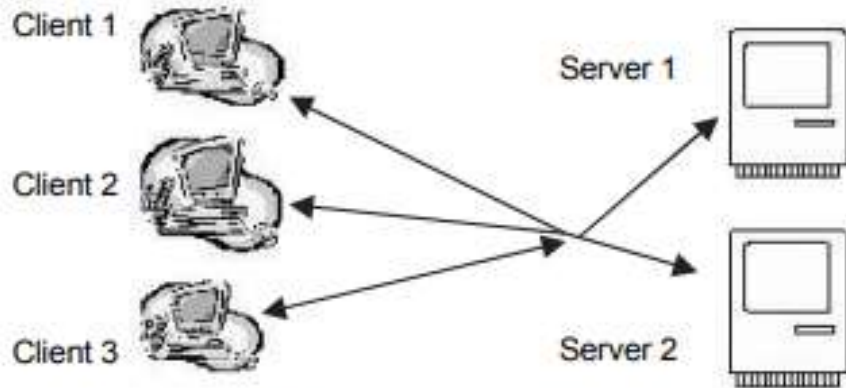
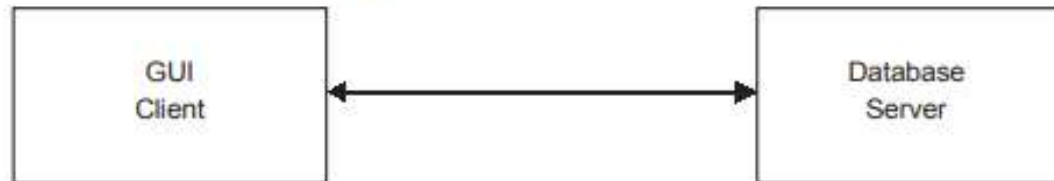


Fig.1.4: Multiple Clients, Multiple Servers

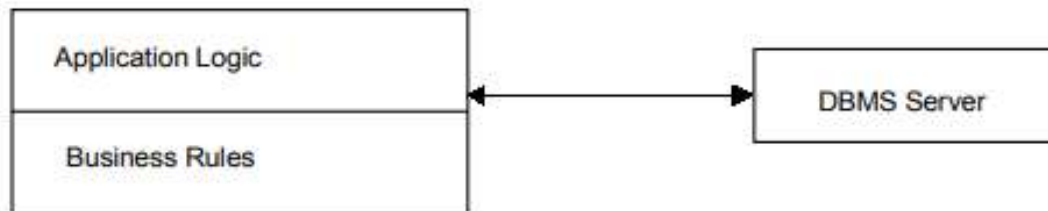
CLASSIFICATION OF CLIENT/SERVER SYSTEMS

- Two-tier Client/Server Model

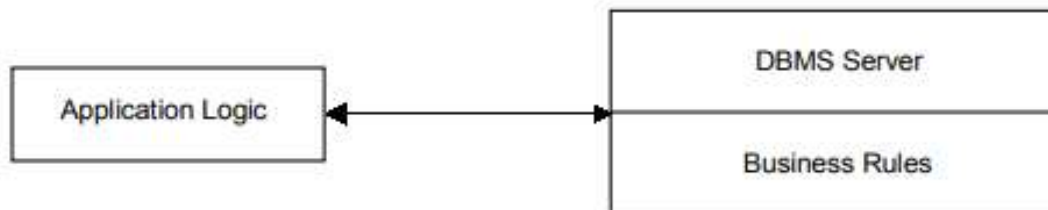
(a) Centralized Two-tier Representation:



(b) Business Rules Residing on the Client:



(c) Business Rules Residing on the Server:



CLASSIFICATION OF CLIENT/SERVER SYSTEMS

- Three-tier Client/Server Model

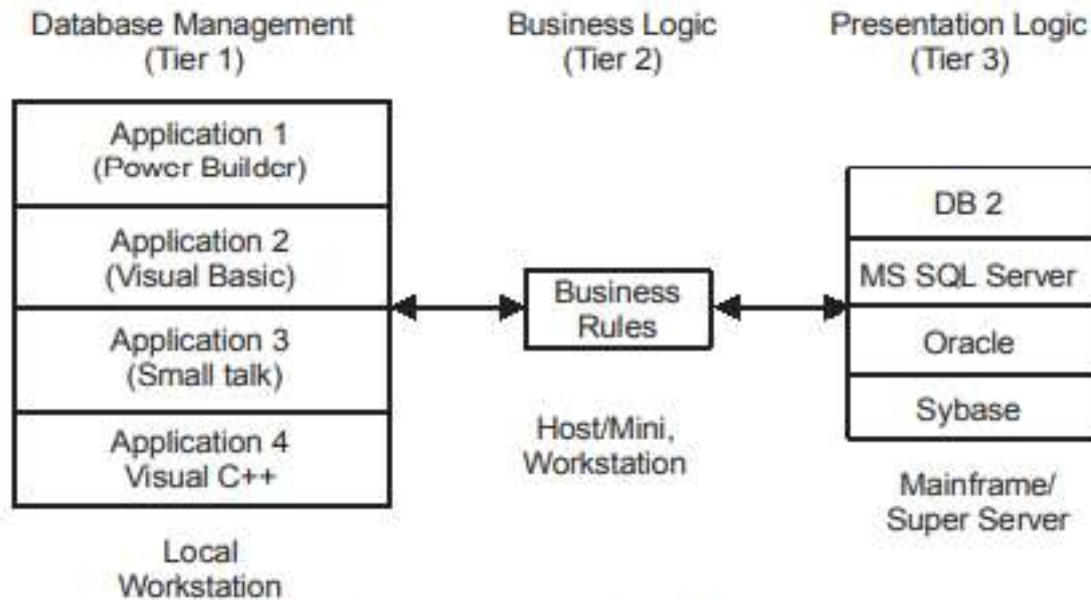


Fig.1.7: Three-tier Architecture

CLASSIFICATION OF CLIENT/SERVER SYSTEMS

- Three-tier Client/Server Model

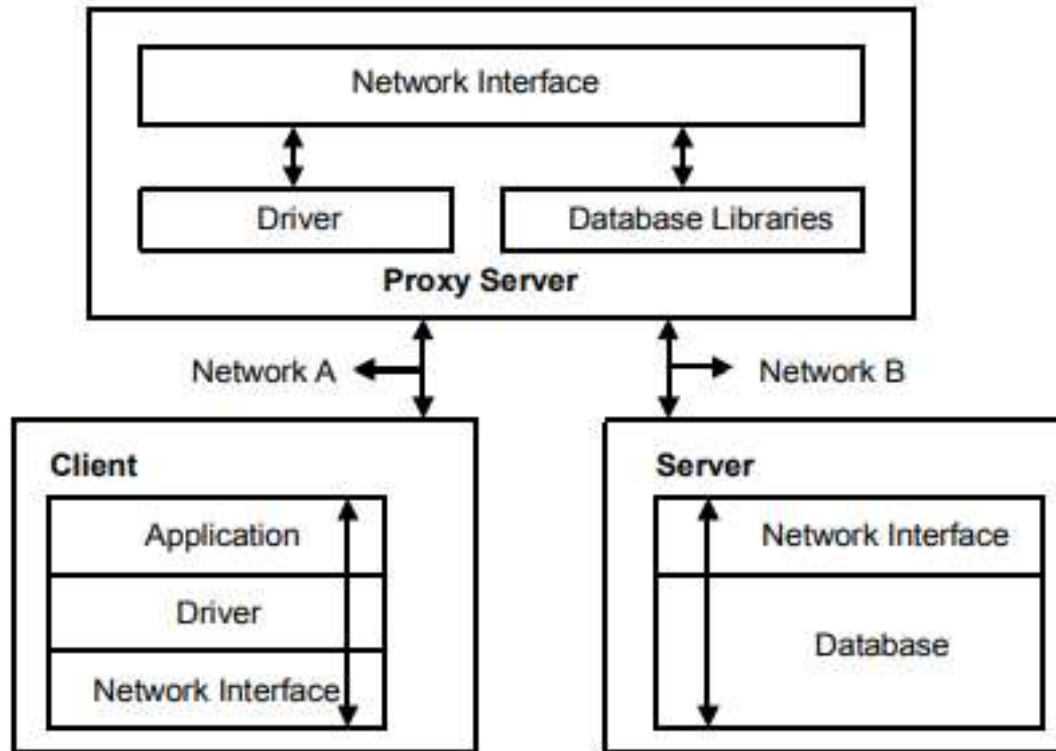
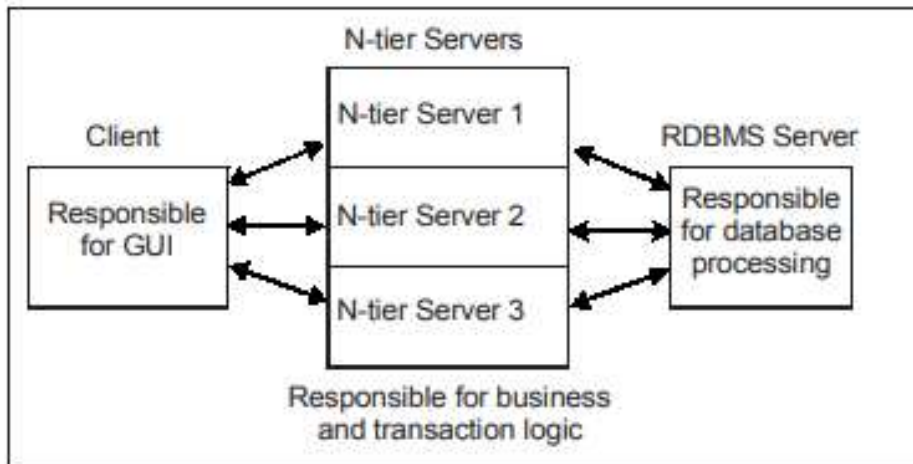


Fig.1.8: Three-tier Client/Server Model

CLASSIFICATION OF CLIENT/SERVER SYSTEMS

- N-tier Client/Server Model



CLIENTS/SERVER—ADVANTAGES

- Performance and reduced workload
- Workstation independence
- System interoperability
- Scalability
- Data integrity
- Data accessibility
- System administration
- Sharing resources among diverse platforms
- Location independence of data processing

CLIENTS/SERVER— DISADVANTAGES

- Maintenance cost
- Training cost
- Hardware cost
- Software cost
- Complexity

Chapter 2: Driving Forces Behind Client/ Server Computing

Prof. Kirtankumar Rathod
Dept. of Computer Science
Indus University

DRIVING FORCES

- Business Perspective
 - The changing business environment
 - Globalization
 - The growing need for enterprise data access
 - The demand for end user productivity gains based on the efficient use of data resources

DRIVING FORCES

- Technology Perspective
 - Standardization
 - HCI
 - Information dissemination
 - Enterprise Computing and the Network Management

DEVELOPMENT OF CLIENT/SERVER SYSTEMS

- Development Tools
 - GUI-based development
 - Object-oriented development with a central repository for data and applications
 - Support for multiple database
 - Complete SDLC (System Development Life Cycle) support from planning to implementation and maintenance.
 - Team development support
 - Support for third party development tools (CASE, libraries, and so on)
 - Support for multiple platforms (OS, Hardware, and GUIs)
 - Multiple network protocol support (TCP/IP, IXP/SPX, NetBIOS, and so on)

DEVELOPMENT OF CLIENT/SERVER SYSTEMS

- Development Phases
 - 1) Information System Infrastructure Self-study
 - 2) Client/Server Infrastructure Definition
 - 3) Selecting a Window of Opportunity
 - 4) Management Commitment
 - 5) Implementation
 - 6) Review and Evaluation

CLIENT/SERVER STANDARDS

- A standard is a publicly defined method to accomplish specific tasks or purposes within a given discipline and technology.
 - For example, the Institute of Electrical and Electronics Engineers (IEEE), American National Standards Institute (ANSI), International Organization for Standardization (ISO)

CLIENT/SERVER SECURITY

1. Trust-based security.
2. Security through obscurity.
3. Password scheme.
4. Biometric system.

Emerging Client/Server Security Threats

- Threats to local computing environment from mobile code.
- Threats to servers that include impersonation, **eavesdropping**, **denial of service**, **packet reply**, and **packet modification**.

ORGANIZATIONAL EXPECTATIONS

- Flexibility and adaptability.
- Improved employee productivity.
- Improved company work flow and a way to re-engineering business operations.
- New opportunities to provide competitive advantages.
- Increased customer service satisfaction.

IMPROVING PERFORMANCE OF CLIENT/SERVER APPLICATIONS

1. Offload work to server
2. Reduce total execution time
3. Use a multitasking client

SINGLE SYSTEM IMAGE (SSI)

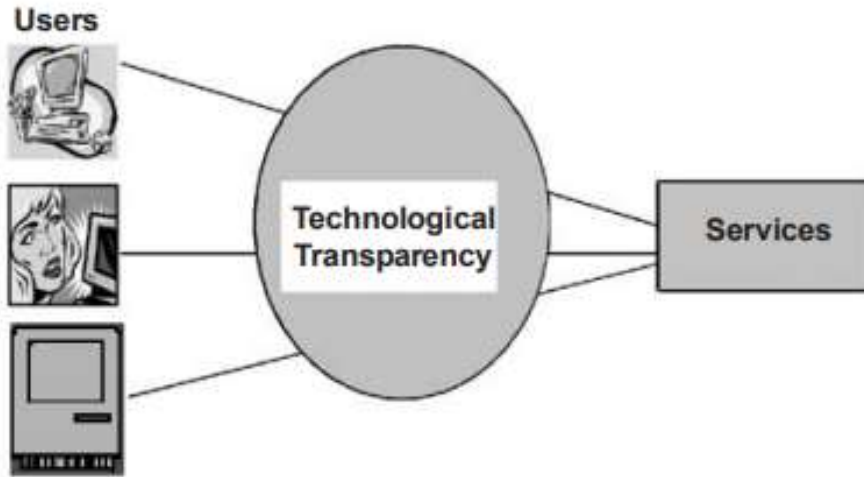


Fig.2.2: Single Image System

- “A SSI is the illusion, created by software or hardware, that presents a collection of resources as one, more powerful resource.”

- The user need not to bother about how the processors (both the client and the server) are working, where the data storage take place and which networking scheme has been selected to build the system.
- Security, scalability and administration costs are three of the key issues.

DOWNSIZING AND RIGHTSIZING

- The downward migrations of business applications are often from mainframes to PCs due to low costing of workstation.
- Moves the Client/Server applications to the most appropriate server platform, in that case the servers from different vendors can co-exist and the network is known as the 'system'.

Chapter 3: Architectures of Client/Server Systems

Prof. Kirtankumar Rathod
Dept. of Computer Science
Indus University

INTRODUCTION

- The Client/Server technology is a model, for the interaction between simultaneously executing software processes.
- The limitations of file sharing architectures led to the emergence of the Client/Server architecture.
- This approach introduced a database server to replace the file server.

COMPONENTS

- Client/Server architecture is based on hardware and software components that interact to form a system.
- There are 3 main components
 1. Hardware
 2. Software
 3. Communication middleware

Components

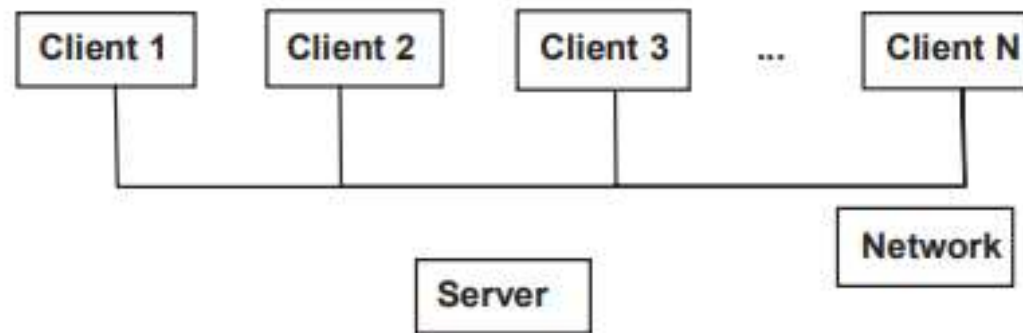


Fig.3.1: Structure of a Client/Server System

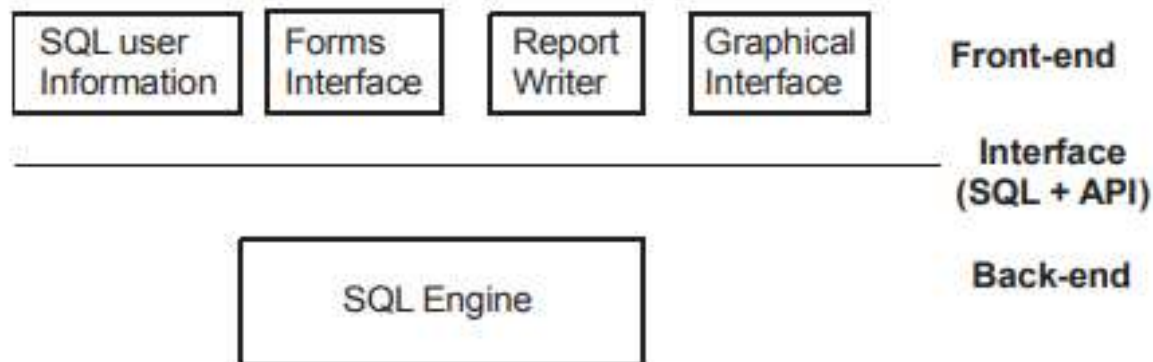


Fig.3.2: Front-end and Back-end Functionality

Interaction between the Components

- Communication middleware ensures that the messages between clients and servers are properly routed and delivered.
- Requests are handled by the database server, which checks the validity of the request, executes them, and send the result back to the clients.

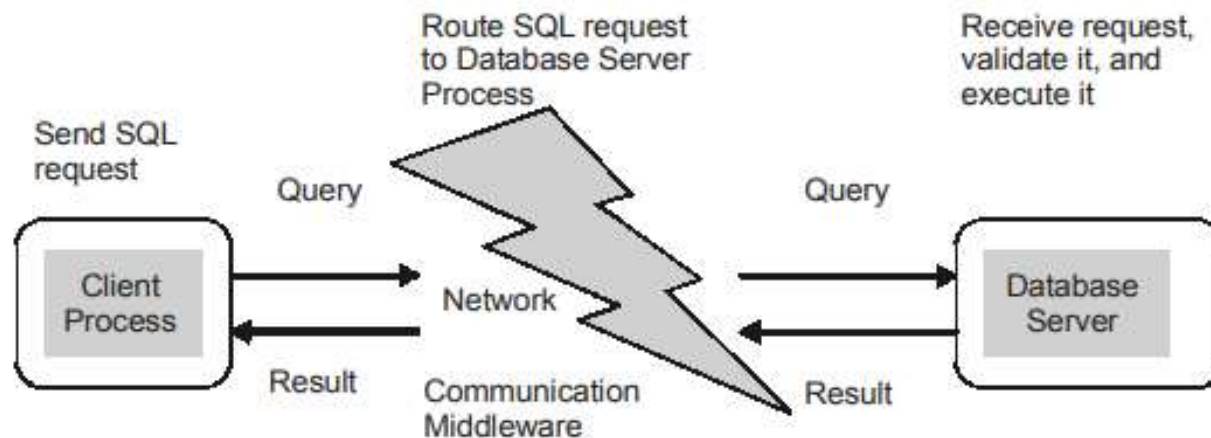


Fig.3.3: Components Interaction

PRINCIPLES BEHIND CLIENT/SERVER SYSTEMS

- These principles generating the Client/Server architecture constitute the foundation on which most current generation Client/Server system are built.
 1. Hardware independence
 2. Software independence
 3. Open access to services
 4. Process distribution
 5. Standards

PRINCIPLES BEHIND CLIENT/SERVER SYSTEMS

1. The Client, Server, and communication middleware, processes run on multiple hardware platforms without any functional differences.
2. Also, support multiple operating systems, multiple network protocols and multiple applications.
3. All client in the system must have open access to all the services provided within the network, and these services must not be dependent on the location of the client or the server.

PRINCIPLES BEHIND CLIENT/SERVER SYSTEMS

4. Client and server processes must be autonomous entities with clearly defined boundaries and functions. The client and server process must fully utilize the processing power of the host computers.
5. All the principles that are formulated must be based on standards applied within the Client/Server architecture. Ex. ODBC, TCP/IP, IPX, etc...

CLIENT COMPONENTS

1. Powerful hardware
2. An operating system capable of multitasking
3. Communication capabilities
4. A graphical user interface

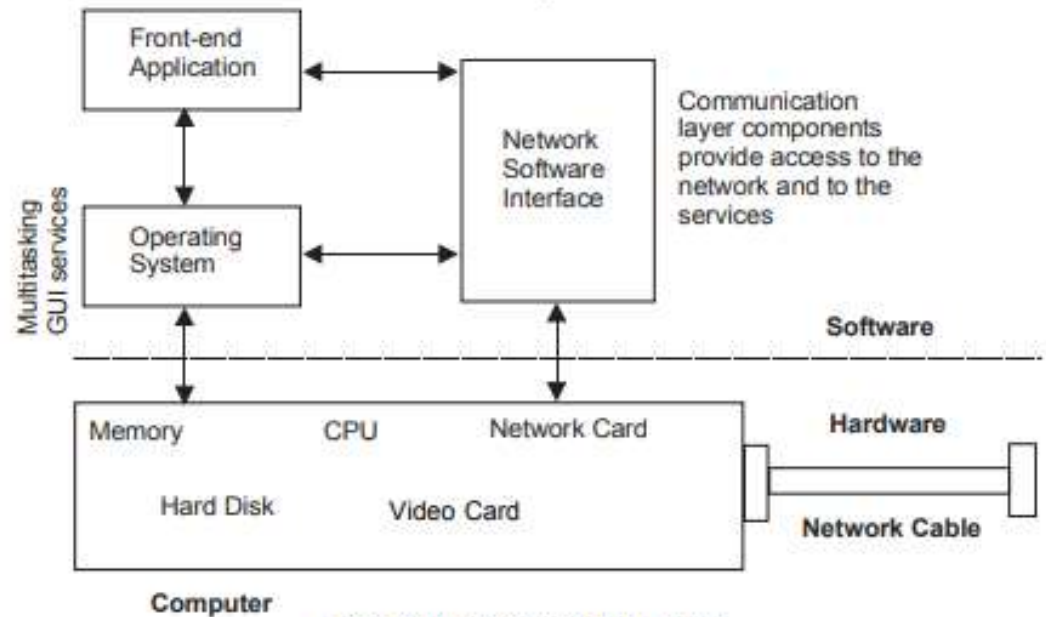


Fig.3.5: Client Components

SERVER COMPONENTS

Services provided by server are:

1. File services
2. Print services
3. Fax services
4. Communication services
5. Database services
6. Transaction services
7. Groupware services
8. Object application services
9. Web application services and many more...

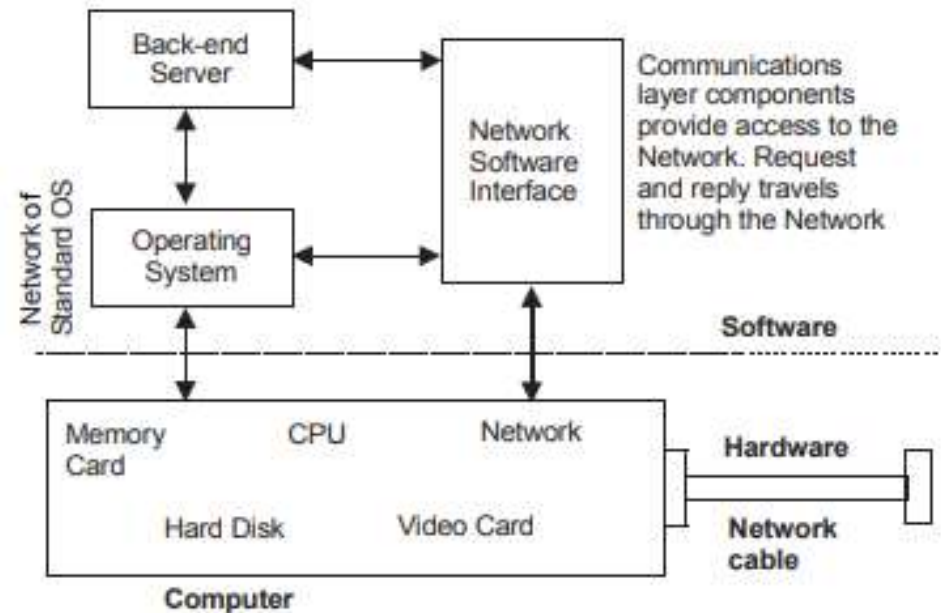


Fig.3.6: Server Components

The Complexity of Servers

- Servers are usually much more difficult to build than clients because they need to accommodate multiple concurrent requests.
 - A single master program that is responsible for accepting new requests.
 - A set of slaves that are responsible for handling individual requests.

Master server performs the following five steps:

1. Open port
2. Wait for client
3. Choose port
4. Start slave
5. Continue

COMMUNICATIONS MIDDLEWARE COMPONENT

- The communication middleware software provides the means through which clients and servers communicate to perform specific actions.

ARCHITECTURE FOR BUSINESS INFORMATION SYSTEM

- Distribution brings a new design dimension into the architecture of information systems.
 - *How do I partition the business information system into a number of client and server components, so that my users' functional and non-functional requirements are met?*

Three-Layer Architecture

- A Business Information System, in which many (spatially distributed) users work in parallel on a large amount of data.

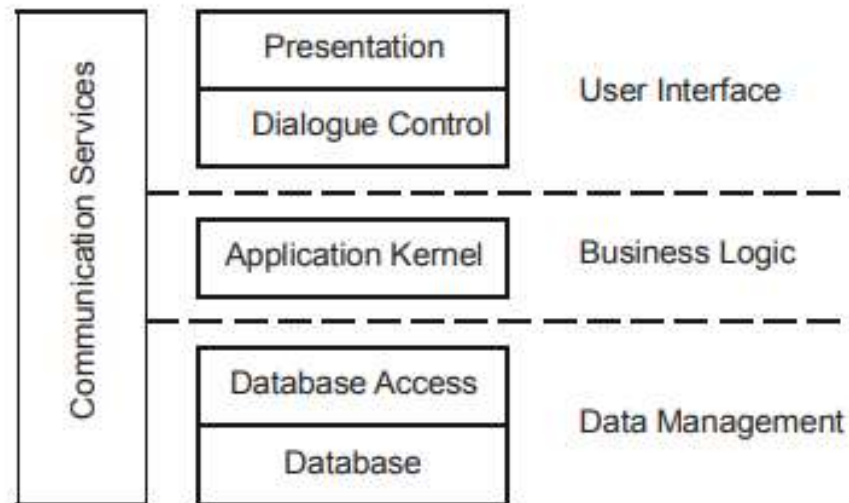
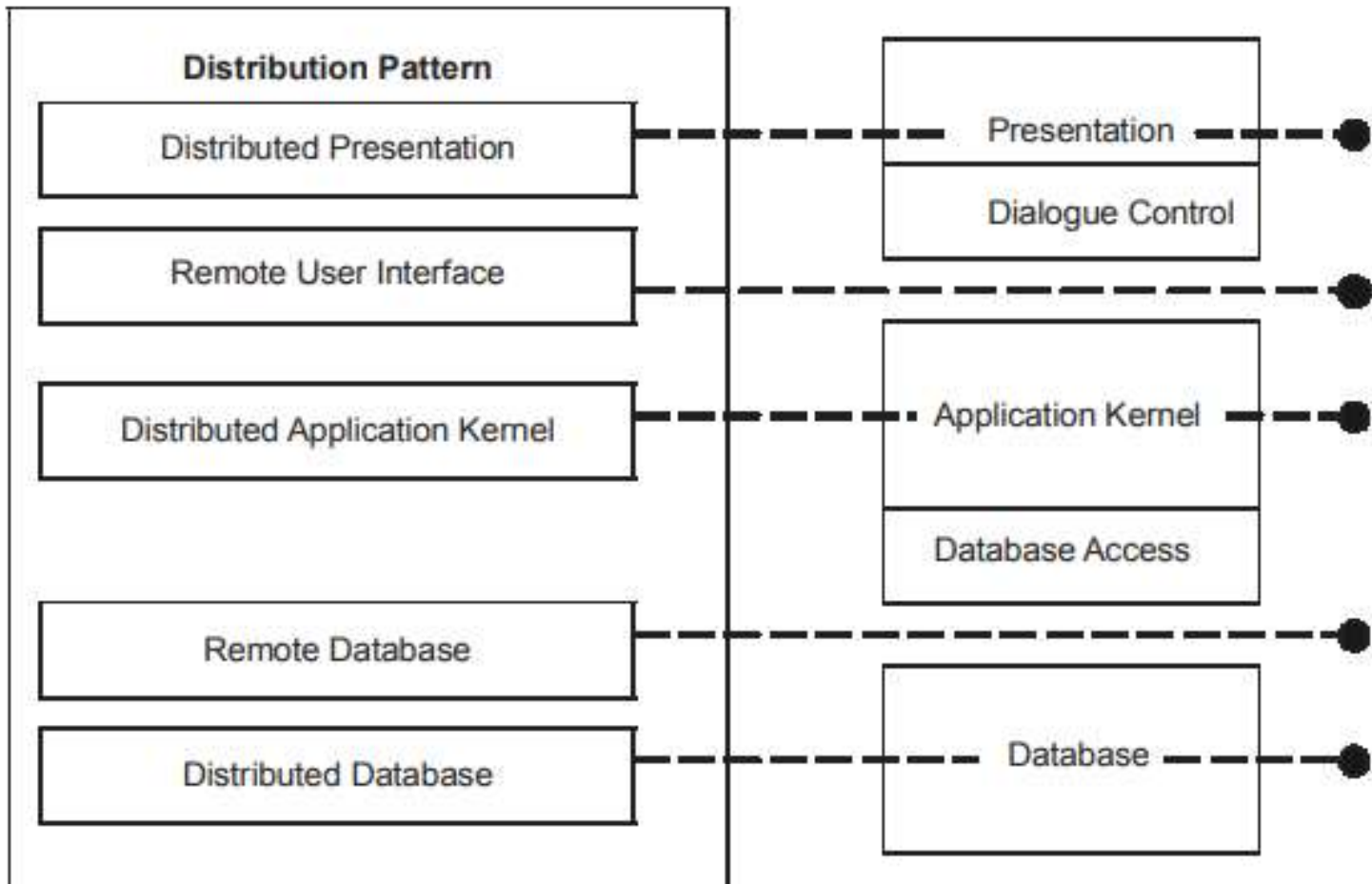


Fig.3.8: Three-Layer Architecture for Business Information System

Distribution Pattern



Distribution Pattern

- Distributed presentation:

One part of the presentation component is packaged as a distribution unit and is processed separately from the other part of the presentation, which can be packaged together with the other application layers.

- Remote user interface:

The whole user interface becomes a unit of distribution and acts as a client of the application kernel on the server side.

Distribution Pattern

- Distributed application kernel:
The pattern splits the application kernel into two parts which are processed separately.
- Remote database:
The database is a major component of a business information system with special requirements on the execution environment.
- Distributed database:
The database is decomposed into separate database components, which interact by means of interprocess communication facilities.

EXISTING CLIENT/SERVER ARCHITECTURE

- Mainframe-based Environment
- LAN-based Environment
- Internet-based Environment

Mainframe-based Environment

- All the processing takes place on the mainframe and usually dumb terminals that are known as end user platform are used to display the data on screens.
- Mainframe systems have very limited data manipulation capabilities.

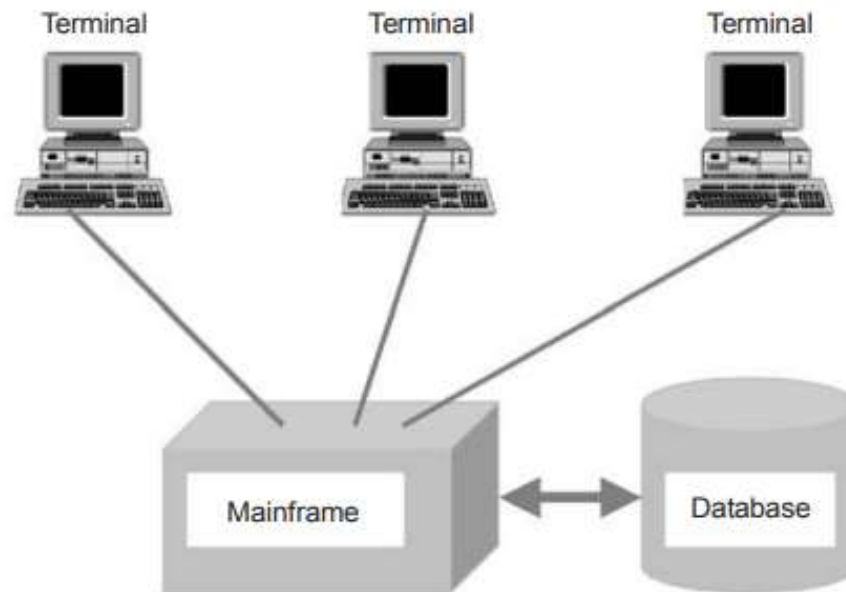


Fig.3:10: Mainframe-based Environment

LAN-based Environment

- Each server in the Client/Server based LAN environment provides a set of shared user services to the clients.
- These servers enable many clients to share access to the same resources and enable the use of high performance computer systems to manage the resources.

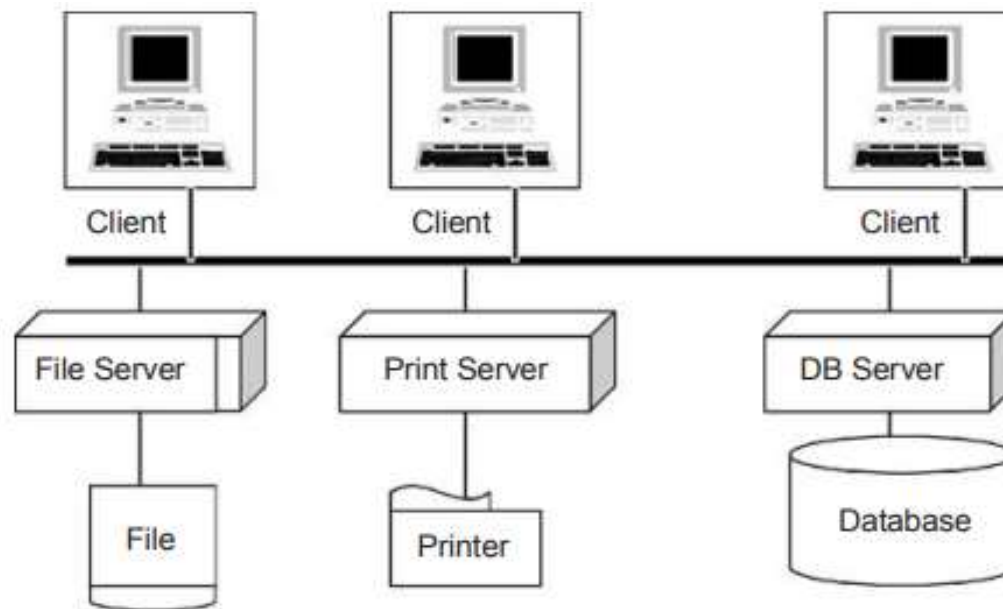


Fig.3.11: LAN Environment

Internet-based Environment

- The web browsers are universal clients.
- The client is simply here to interpret the server's command and render the contents of an HTML page to the user.
- Browsers are there to take care of all the platform-specific details.

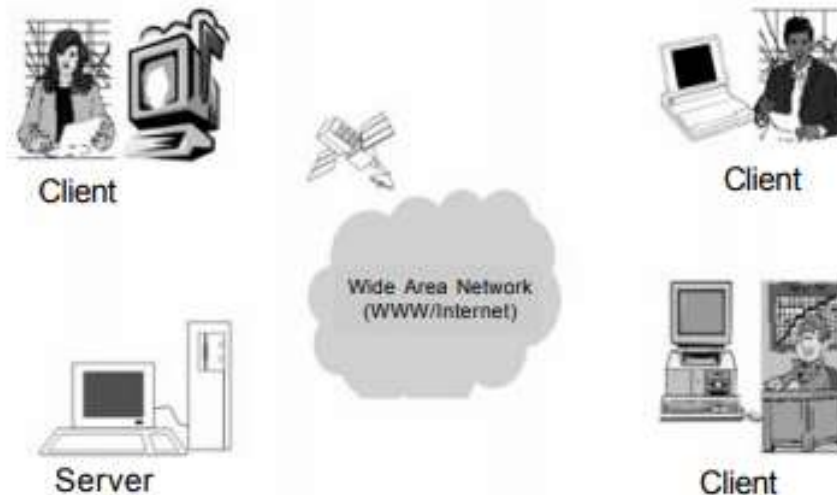


Fig.3.12: Internet-based Environment