

# Coding Mum

## Lesson 01

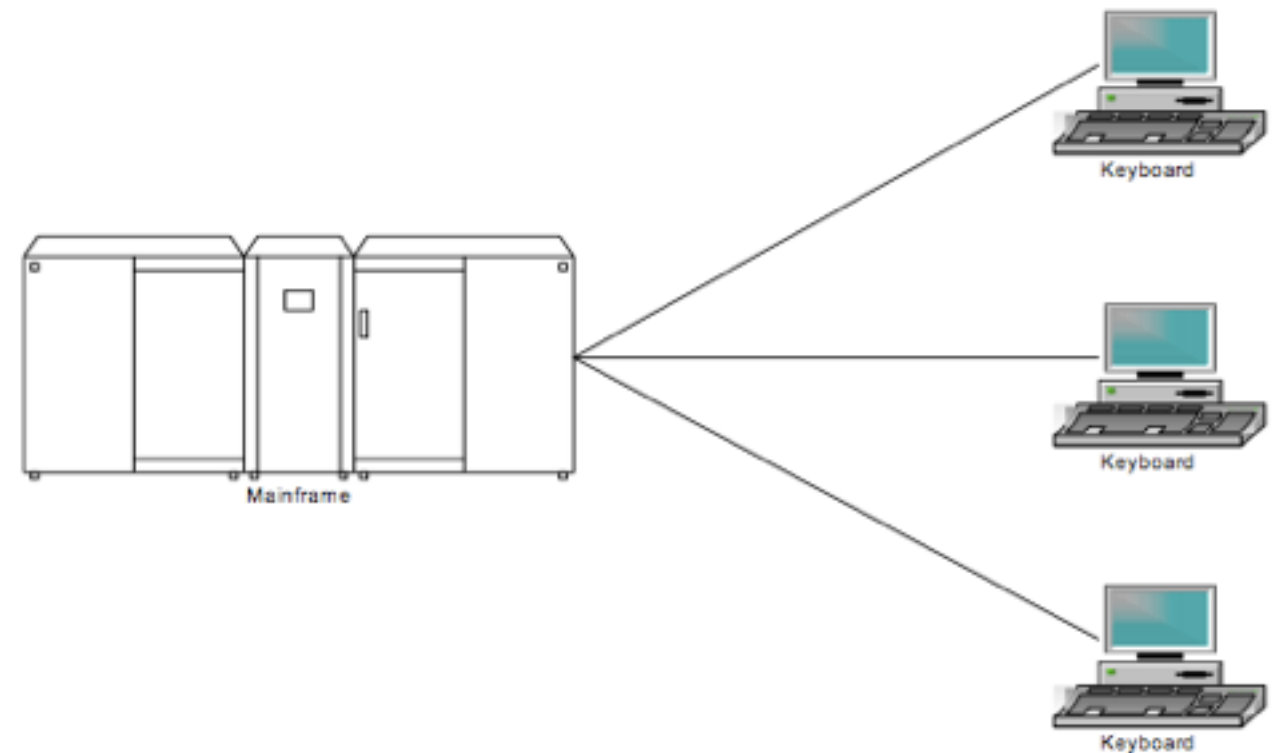
---

### N-Tier Architecture

# Traditional Host System

---

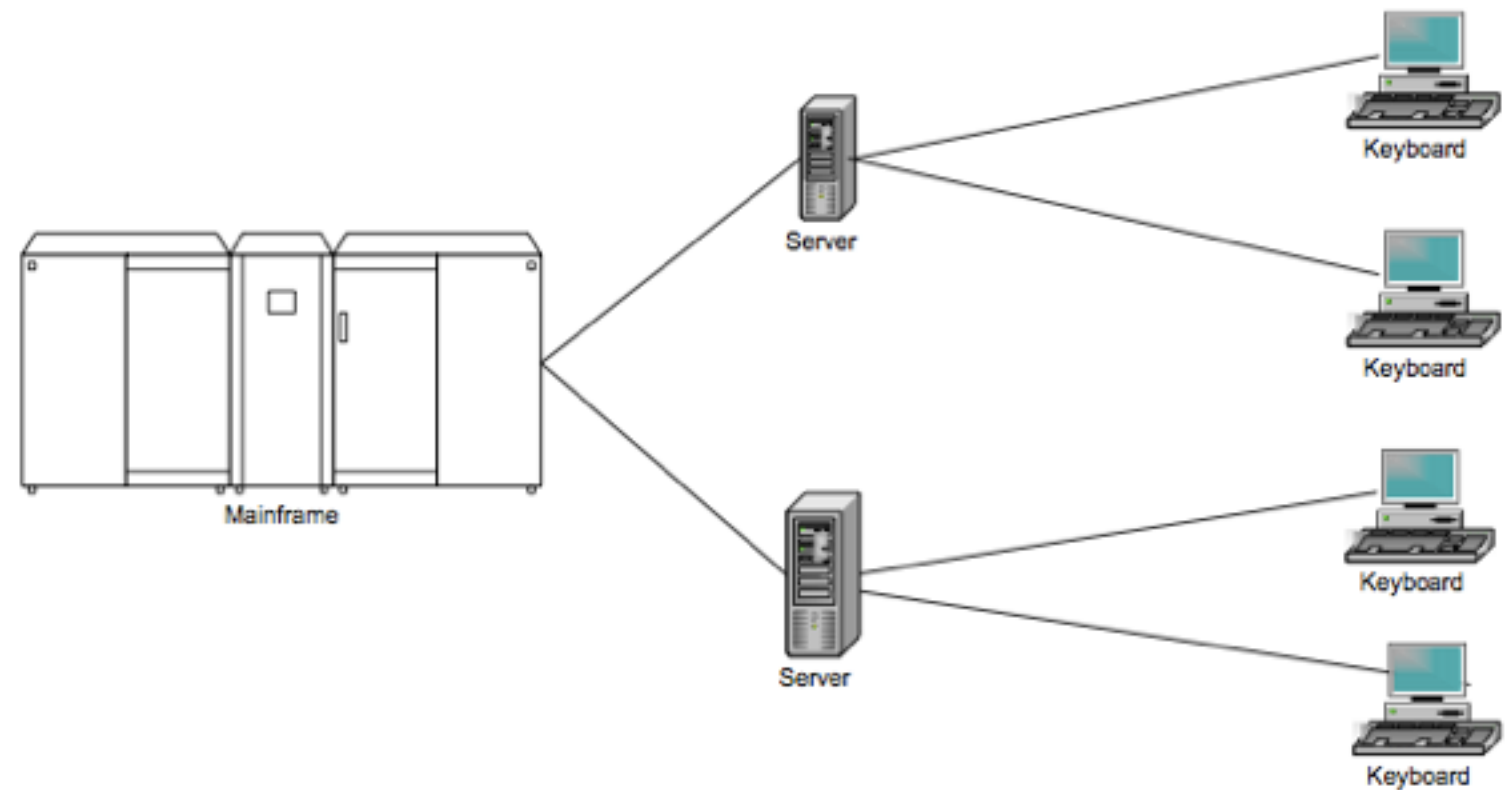
- ▶ A Central Processing System (Mainframe) provides all processing
- ▶ Local Terminals are responsible for display and keyboard for user input and viewing capabilities.
- ▶ Local Terminals do not contain any intelligent processing capabilities.
- ▶ Non client-server system



# Traditional Host System

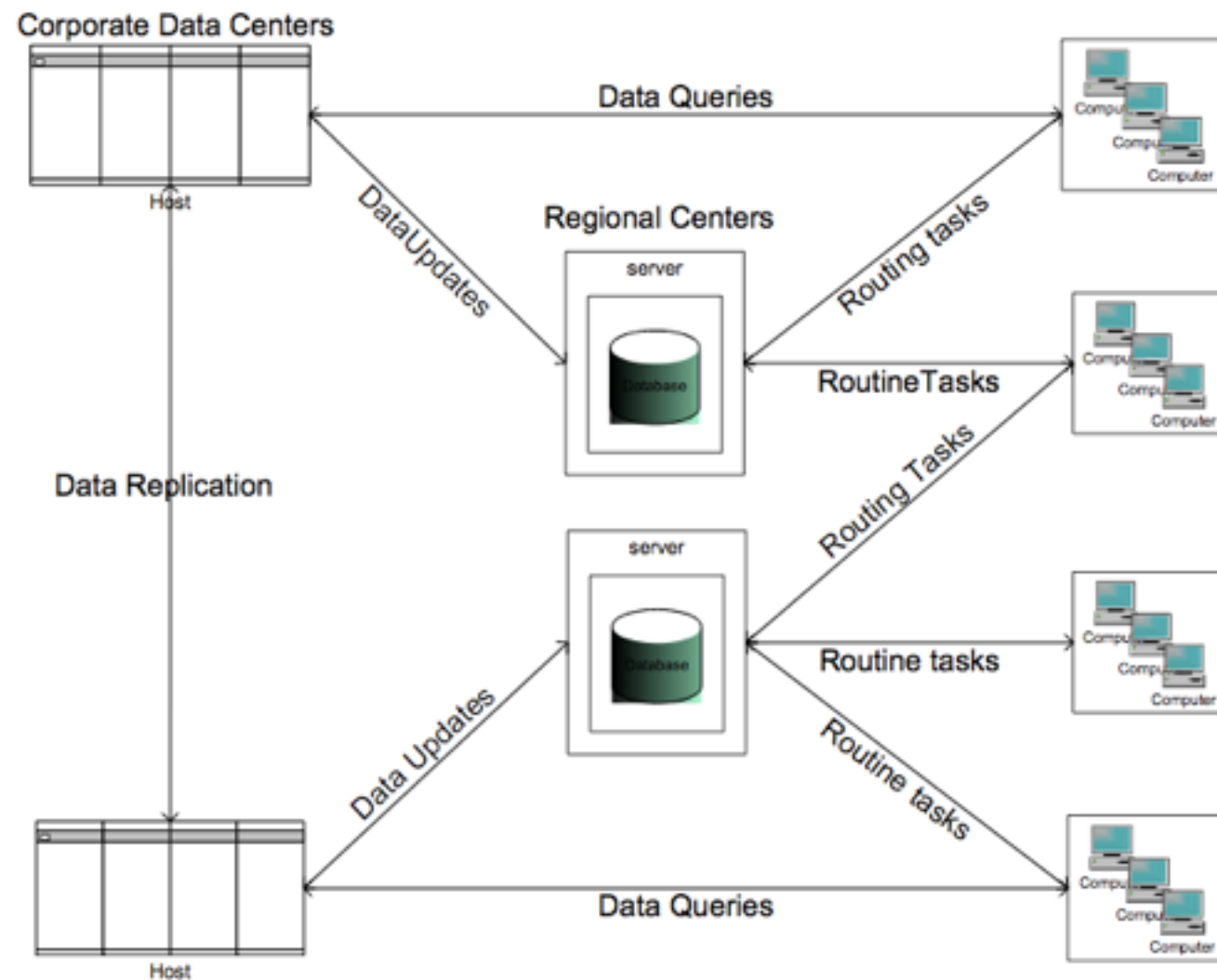
---

- ▶ File Server and retrieval processing provided by File Server
- ▶ Word Processing and spreadsheet processing provided by PC workstation
- ▶ System with LAN



# Distributed Systems

---



- ▶ Both data and transaction processing are divided between one or more computers connected by a network, each computer playing a specific role in the system.

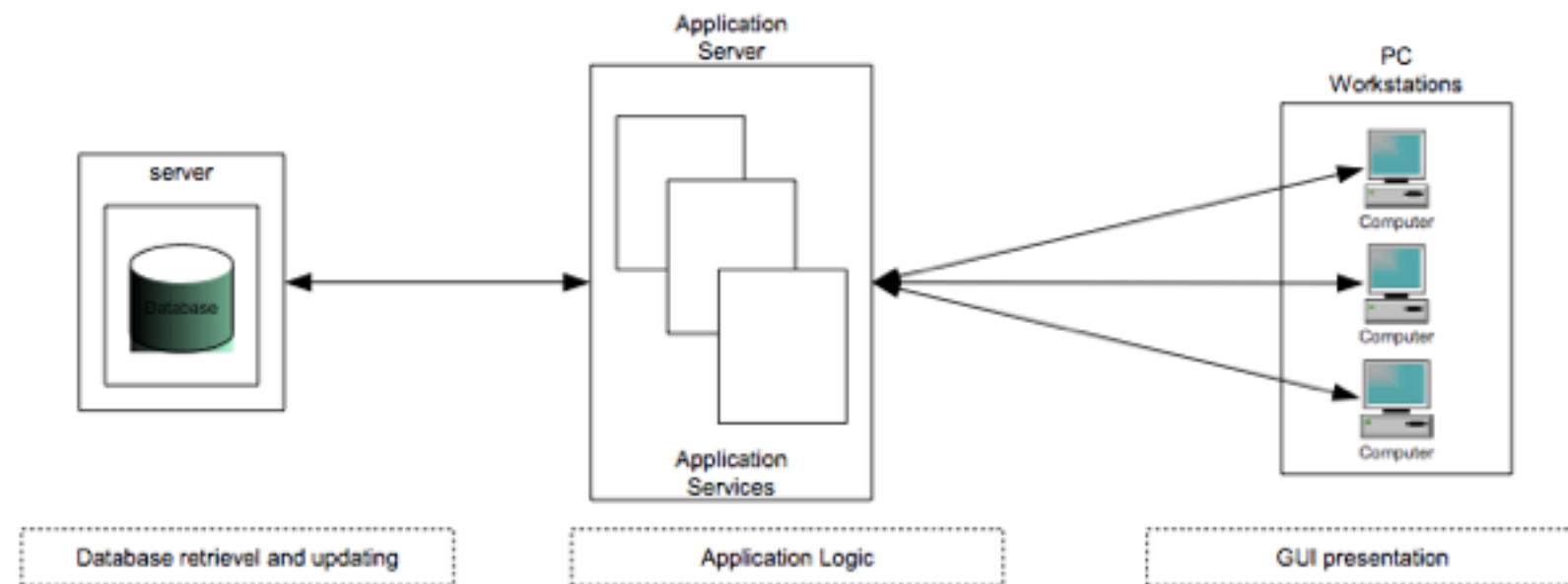
# Client-Server Models

- Complements distributed systems  
Responds to limitations found in the two host data processing models:

1. The traditional mainframe host model which a single mainframe provides shared data access to many dumb terminals, and;
2. The local area network (LAN) model which many isolated systems access file server that provides no central processing power.

Provides integration of data and service  
Application Processing provided by multiple tiers

3. DatabaseServer
4. ApplicationServer
5. PCWorkstation

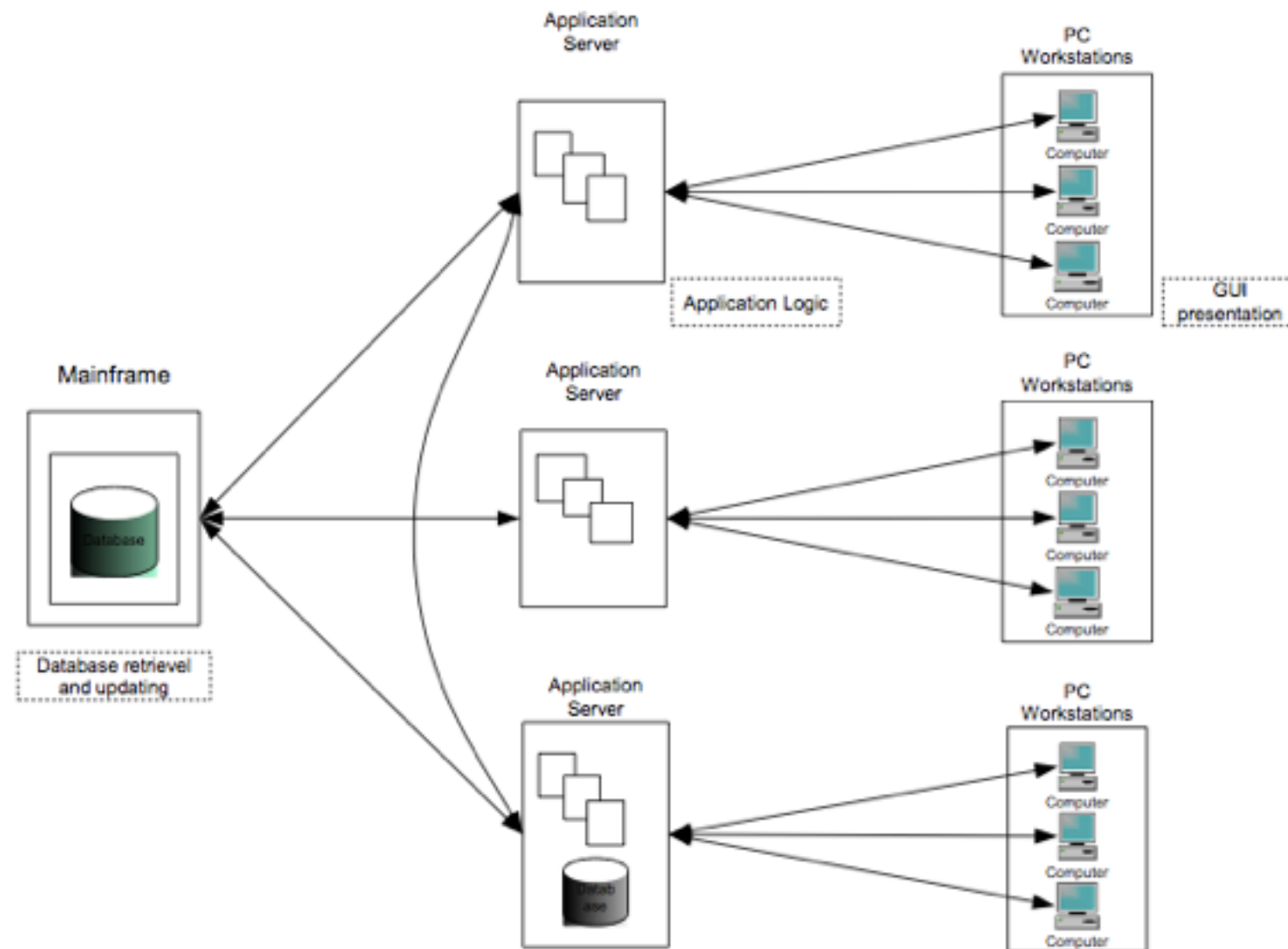


# Distributed Client-Server Model

- ▶ Application processing provided by all tiers of the network

1. Mainframe
2. Application Servers
3. Workstations

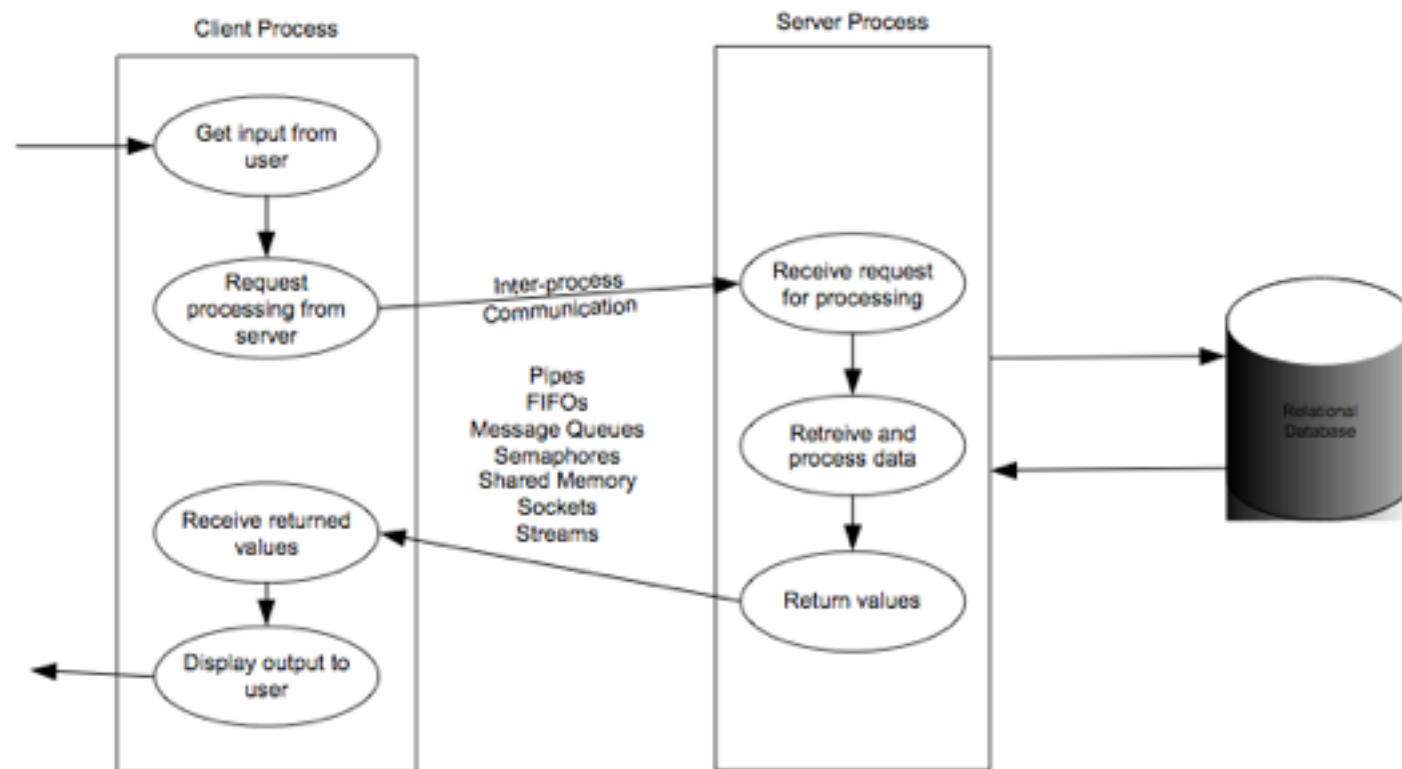
Multiple databases to support distributed data requirements  
Supports high volume, load balancing and scalability (extendibility)  
Requires extensive network administration and application management.



# Inter-process Communication

---

- ▶ Basis for client/server computing
- ▶ Client process communicates with server process
- ▶ Each process performs separate functions
- ▶ Data is passed between processes using IPC functions



# Benefits of the Client/Server Model

---

► Divides Application Processing across multiple machines:

- Non-critical data and functions are processed on the client
- Critical functions are processed on the server

Optimizes Client Workstations for data input and presentation (e.g., graphics and mouse support)

Optimizes the Server for data processing and storage (e.g., large amount of memory and disk space)

Scales Horizontally – Multiple servers, each server having capabilities and processing power, can be added to distribute processing load.

Scales Vertically - Can be moved to more powerful machines, such as minicomputer or a mainframe to take advantage of the larger system's performance

Reduces Data Replication - Data stored on the servers instead of each client, reducing the amount of data replication for the application.



# Client/Server 2-Tier Architecture

Two-tier client/server architectures have 2 essential components

1. A ClientPC
2. A Database Server

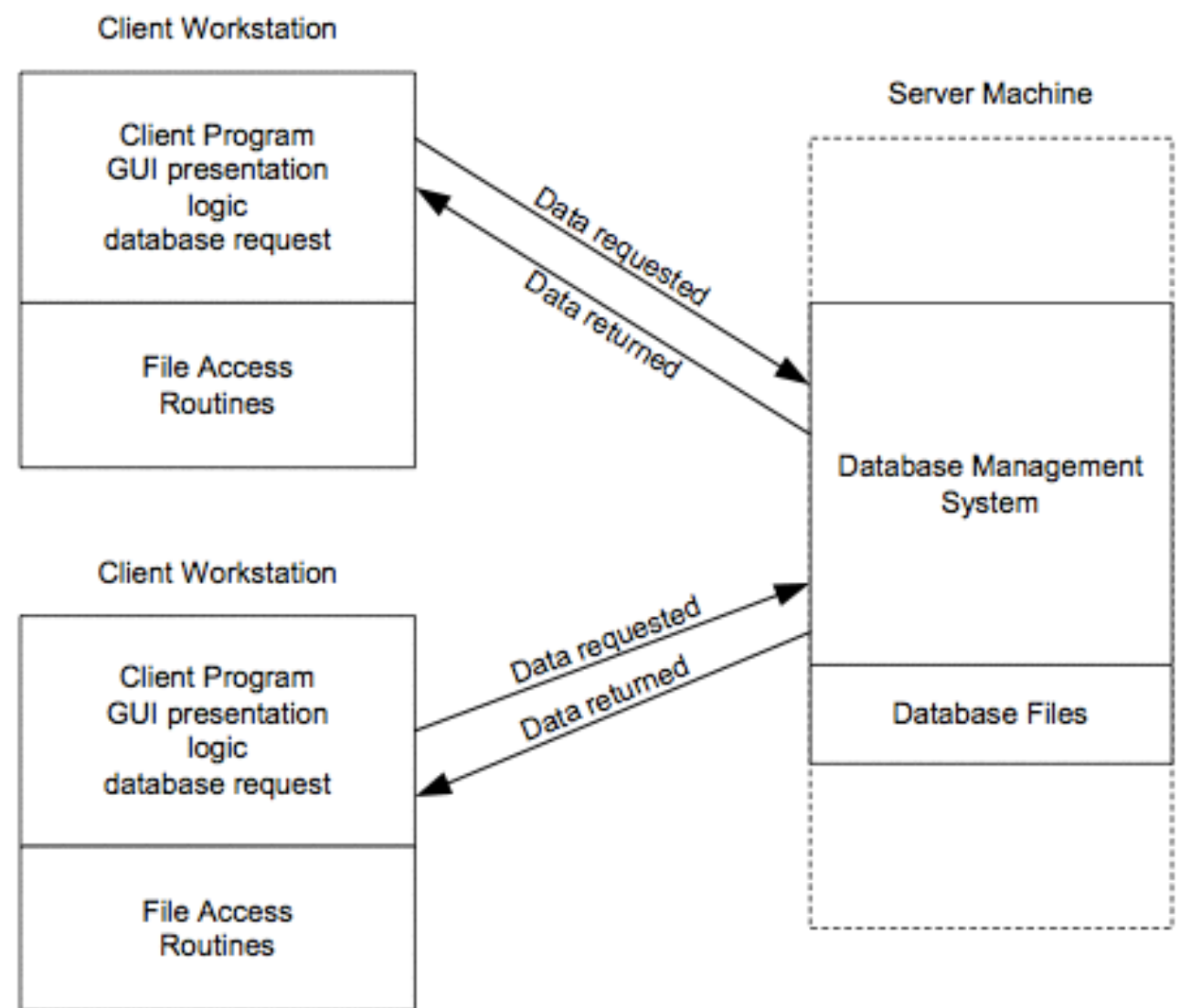
2-Tier Considerations:

Client program accesses database directly

1. Requires a code change to port to a different database
2. Potential bottleneck for data requests
3. High volume of traffic due to data shipping

Client program executes application logic

4. Limited by processing capability of client workstation (memory, CPU)
5. Requires application code to be distributed to each client workstation



# Two-tier Pros and Cons

---

Advantages	Disadvantages
<p><i>Development Issues:</i></p> <ul style="list-style-type: none"><li>• Simple structure</li><li>• Easy to setup and maintain</li></ul>	<p><i>Development Issues:</i></p> <ul style="list-style-type: none"><li>• Complex application rules difficult to implement in database server – requires more code for the client</li><li>• Complex application rules difficult to implement in client and have poor performance</li><li>• Changes to business logic not automatically enforced by a server – changes require new client side software to be distributed and installed</li><li>• Not portable to other database server platforms</li></ul>
<p><i>Performance:</i></p> <ul style="list-style-type: none"><li>• Adequate performance for low to medium volume environments</li><li>• Business logic and database are physically close, which provides higher performance.</li></ul>	<p><i>Performance:</i></p> <ul style="list-style-type: none"><li>• Inadequate performance for medium to high volume environments, since database server is required to perform business logic. This slows down database operations on database server.</li></ul>

# 3-Tier Client-Server Architecture

---

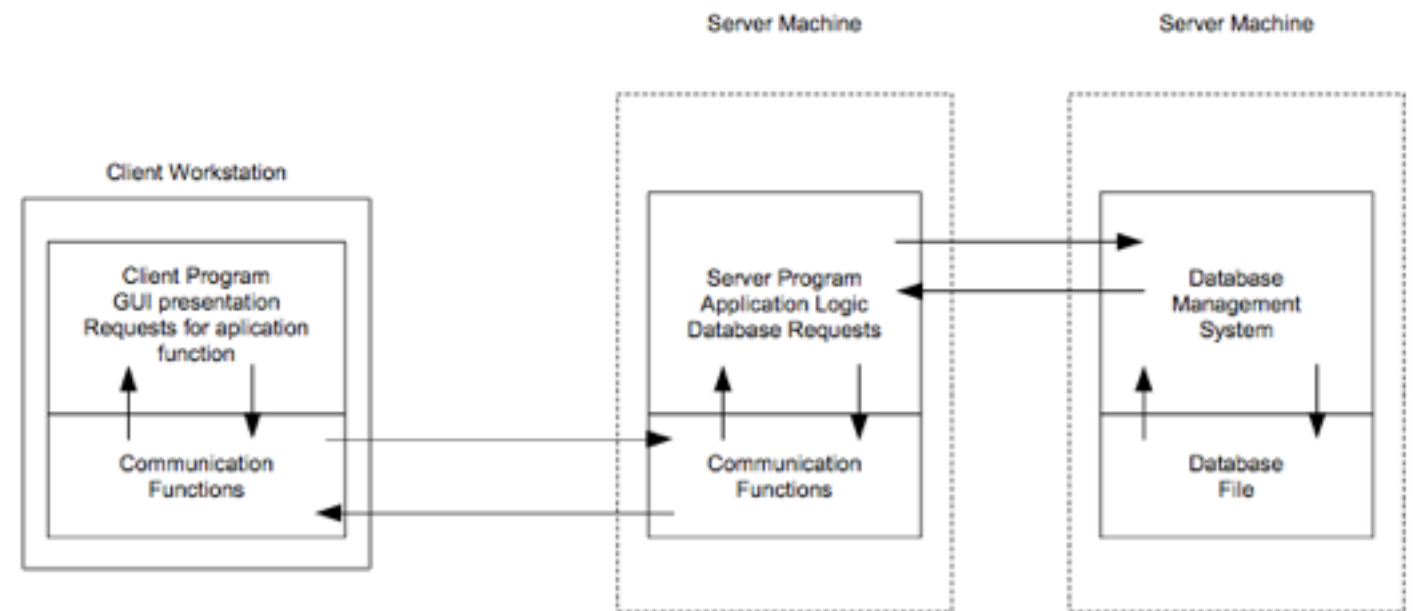
3-Tier client-server architectures have 3 essential components:

- A Client PC
- An Application Server
- A Database Server

Architecture Considerations:

- Client program contains presentation logic only
- Less resources needed for client workstation
- No client modification if database location changes
- Less code to distribute to client workstations

- One server handles many client requests
- More resources available for server program
- Reduces data traffic on the network



# 3-Tier Pros and Cons

---

Advantages	Disadvantages
<p><i>Development Issues:</i></p> <ul style="list-style-type: none"><li>• Complex application rules easy to implement in application server</li><li>• Business logic off-loaded from database server and client, which improves performance</li><li>• Changes to business logic automatically enforced by server – changes require only new application server software to be installed</li><li>• Application server logic is portable to other database server platforms by virtue of the application software</li></ul>	<p><i>Development Issues:</i></p> <ul style="list-style-type: none"><li>• More complex structure</li><li>• More difficult to setup and maintain.</li></ul>
<p><i>Performance:</i></p> <ul style="list-style-type: none"><li>• Superior performance for medium to high volume environments</li></ul>	<p><i>Performance:</i></p> <ul style="list-style-type: none"><li>• The physical separation of application servers containing business logic functions and database servers containing databases may moderately affect performance.</li></ul>

# Middleware

---

Simplifies 3-tier application development and administration by providing an extra application server layer to manage communication between components.

## Middleware Characteristics:

- ▶ Simplifies partitioning of application processing among clients and servers
- ▶ Manages distributed transactions among multiple databases
- ▶ Communicates with heterogeneous database products within a single application.
- ▶ Supports application scalability
- ▶ Supports service requests prioritization, load-balancing, data dependant routing and queuing.

