

A hand holding a glass with a blue and orange liquid gradient.

MIS 131: Information Systems Administration

Part VII: Database Systems Architecture

Conceptual Layers of an Information System

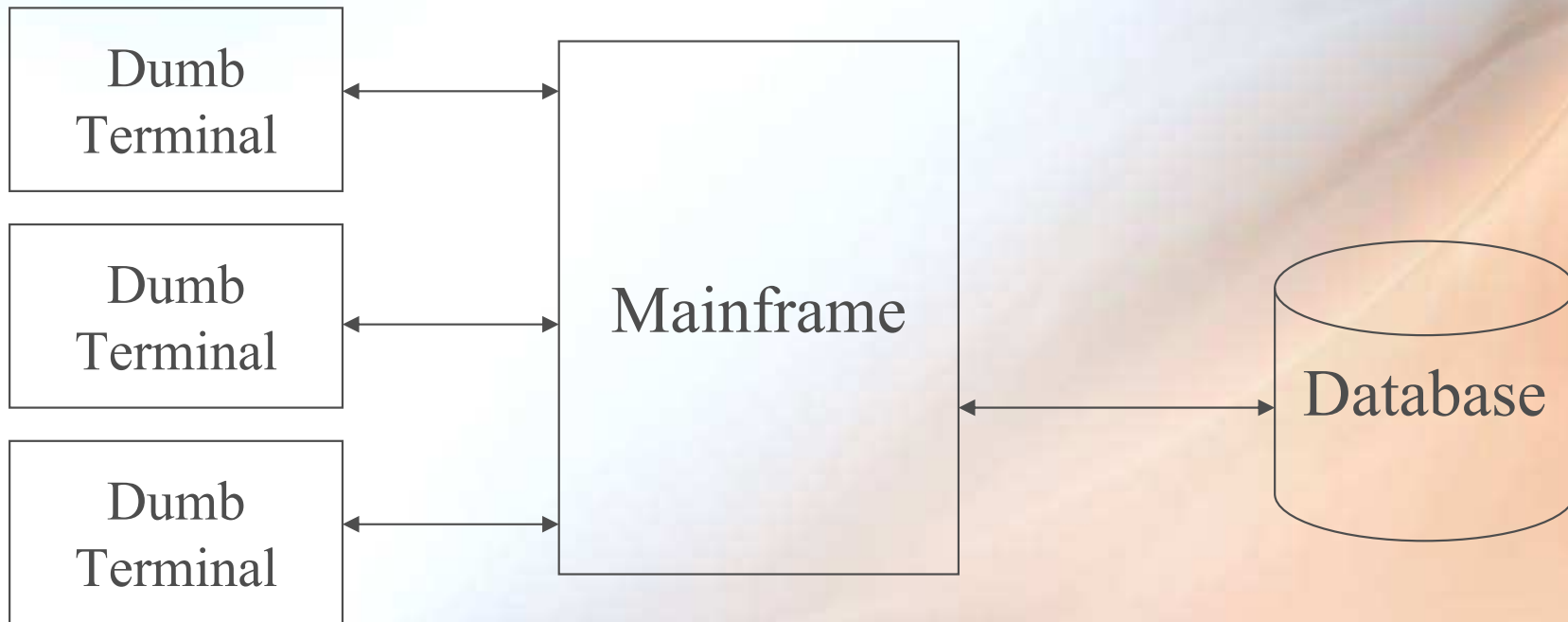
- **Presentation**
 - The actual user interface
- **Presentation logic**
 - Any processing done in UI (e.g. editing input data)
- **Application logic**
 - Logic and processing required to support actual business application and rules
- **Data manipulation**
 - Commands and logic to store and retrieve data
- **Data layer**
 - Actual stored data

System Architecture

- **Centralized**
 - Central computer corporate databases (e.g. mainframe)
 - Personal computer databases
- **Distributed**
 - File server computing
 - Client/server computing
 - Network computing
- **Segregation of layers depends on architecture**

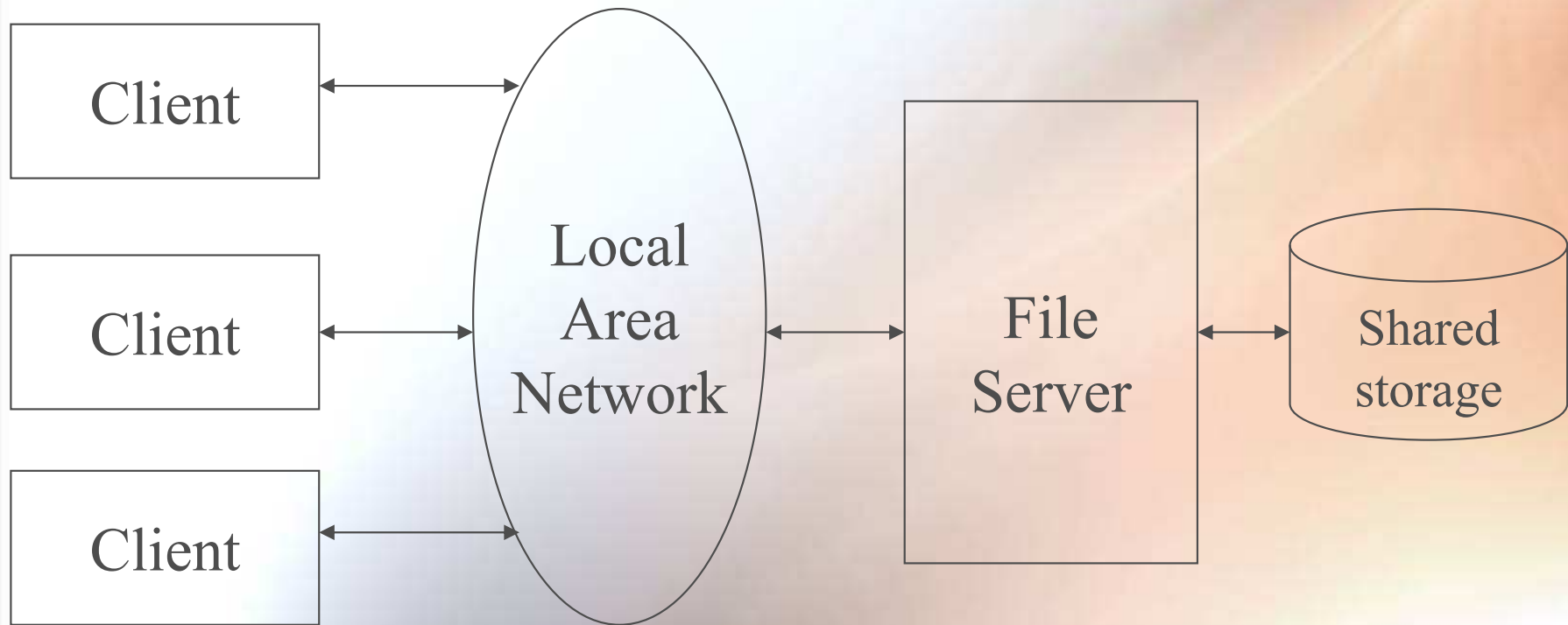
Centralized Architecture

- **Host-based** - processing is performed in the host computer only



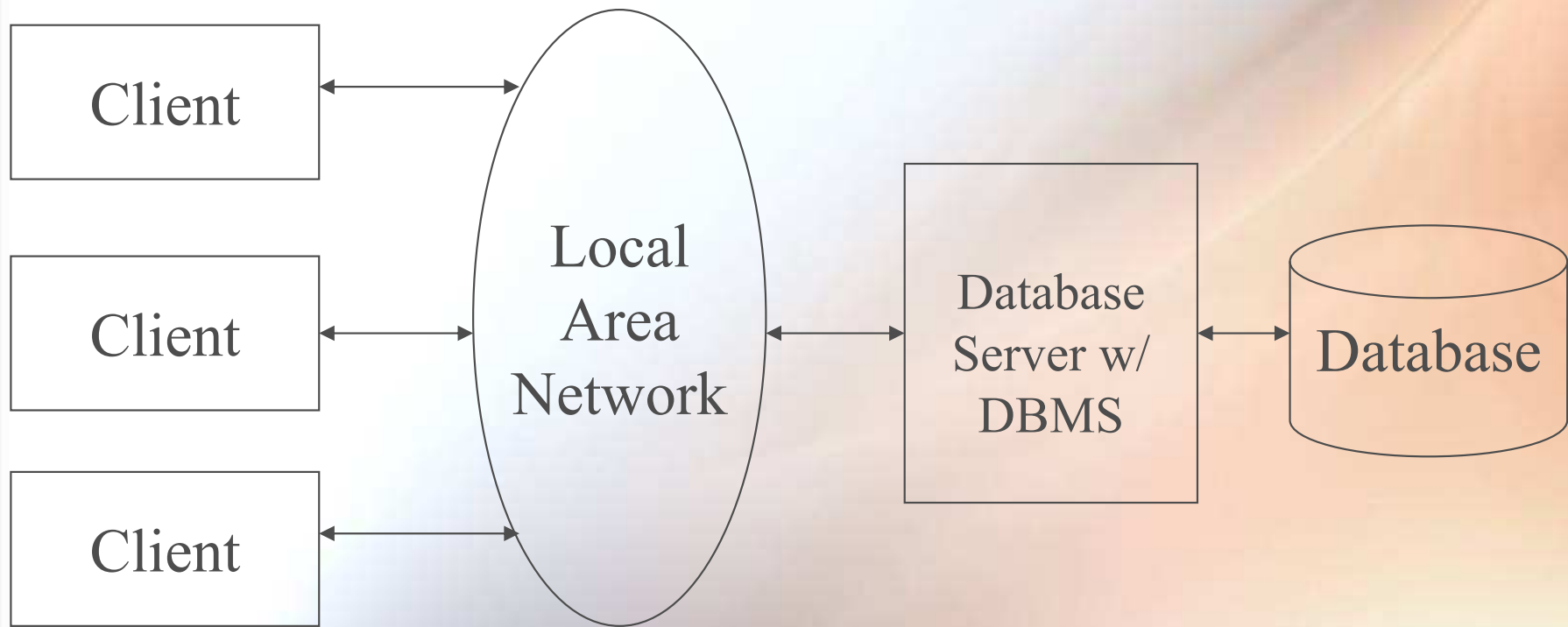
Distributed Systems

- **File server computing**
 - Single file serves as database; must be downloaded to client
 - Processing is performed in the client computer only
 - Only use of file server is to store and transport the DB



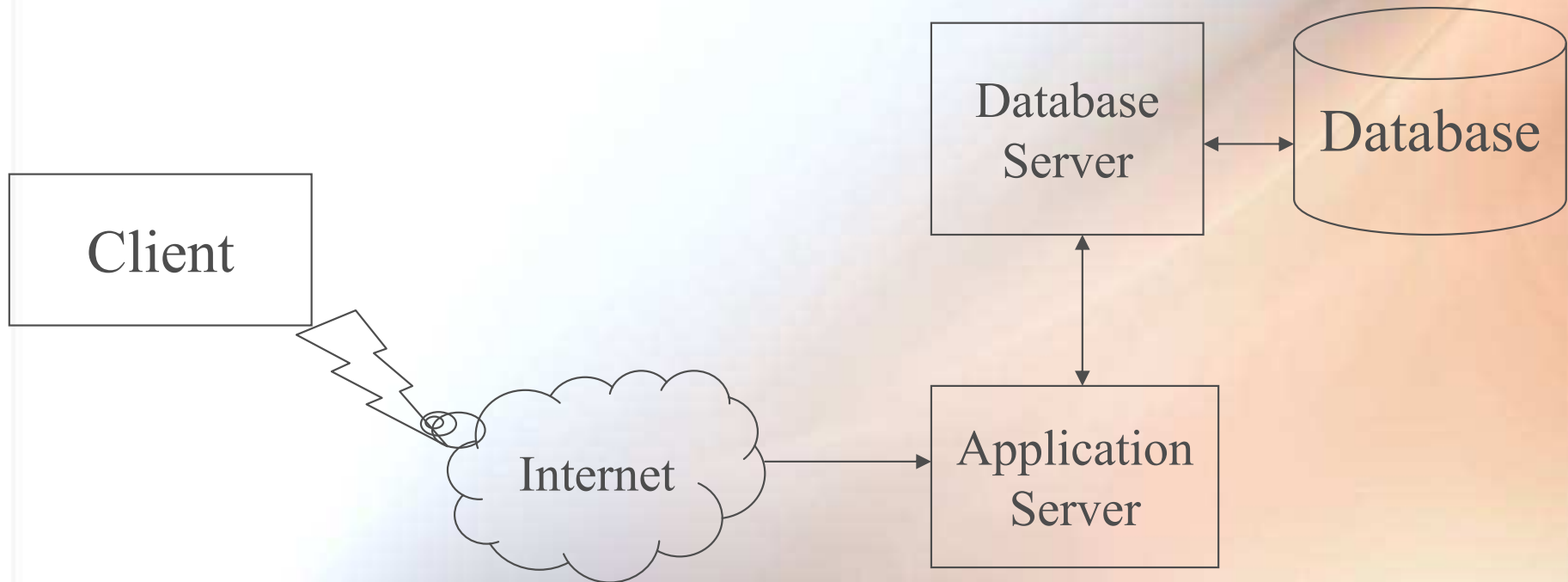
Distributed Systems

- **Client/server computing**
 - Processing is performed in both client and database server
 - Selected data is sent to client



Distributed Systems

- **Network computing (Internet-based)**
 - Application server, database server, and client are separate
 - Usually uses thin clients



Client/Server Computing

- **A solution in which the various IS application layers are distributed between client and server**
- **Client**
 - Present the user interface
 - Forms queries or commands in specific language (e.g. SQL)
 - May be thin or thick client
- **Server**
 - Responds to queries with requested data
 - Hides server system from client and from end user

Advantages of Client/Server

- **Leverage the benefits of microcomputer technology**
- **Allows most of the processing to be close to the source as possible, improving response time and reducing network traffic**
- **Facilitates the use of GUI available on the client**
- **Allows for and encourages acceptance of open systems**

Issues in Client/Server

- **Server must be able to grow as more clients are added**
- **Gateways from server to existing mainframe or mini will be necessary for data sharing amongst different databases**
- **Capabilities must be provided for server for back-up/recovery, security, and uninterrupted power**

Issues in Client/Server

- **More complex because it requires close communication between server and client for proper database processing**
- **No standards for distributing processing between client and server**
- **Sophisticated front end often required high end micros**

Tiered Architecture in C/S Systems

- **Distributed presentation**
 - A solution in which the presentation layer (and probably the presentation logic) is shifted from server of a legacy system to GUI client
 - Sometimes called the poor person's client/server system
 - Advantage: Very few changes in system
 - Disadvantage: Functionality not improved

Tiered Architecture in C/S Systems

- **Distributed data**
 - Also known as two-tiered client/server computing
 - Simplest form of true client/server computing
 - Places the information system's stored data on a server and business logic and user interface on the clients

Tiered Architecture in C/S Systems

- **Distributed data and application**
 - Also called three-tiered or n-tiered client/server computing
 - Data and data manipulation layer placed on one server; application logic layer has its own server
 - Clients execute a minimum of the overall system's components

Distributed Databases

- **Distributed Database**
 - Single logical database
 - Physically located in multiple computers
 - Interconnected by networks - data is shared
 - Homogeneous and heterogeneous databases
- **Decentralized Database**
 - Collection of independent databases
 - Physically located in multiple computers
 - Not connected by networks - no sharing of data

Advantages of Distributed Databases

- **Increased reliability and availability**
- **Local control**
- **Modular growth**
- **Lower communication costs**
- **Faster response**

Disadvantages of Distributed Databases

- **Software costs and complexity**
- **Data integrity**
- **Slow response if not properly distributed or queries not formulated properly**

Options for Distributed Databases

- **Data replication**
- **Horizontal partitioning**
- **Vertical partitioning**
- **Combination of above**

Data Replication

- **Store a copy of the database in two or more sites**
- **Advantages**
 - Reliability
 - Fast response
- **Disadvantages**
 - Large storage requirements
 - Complexity and cost of updating

Horizontal Partitioning

- Some rows of a relation are put in one site and another set of rows in another site, etc.
- Advantages
 - Local optimization
 - Security
- Disadvantages
 - Inconsistent access speed
 - Back-up vulnerability

Vertical Partitioning

- **Some columns of a relation are projected in one site and another set of columns in another site, etc.**
- **Advantages and disadvantages are same as horizontal partitioning**

Horizontal vs. Vertical Partitioning

- **Horizontal partitioning supports an organizational design in which functions are replicated (e.g. branches)**
- **Vertical partitioning applies across organizational functions with reasonably separate data requirements (e.g. departments)**

Distributed DBMS

- **The DBMS that coordinates access to data at various nodes.**
- **Determine the location from which to retrieve requested data**
- **If necessary, translate request from one node to a proper request to another node**
- **Provide data management functions such as security, concurrency, deadlock control, query optimization, and failure recovery**

Objectives of Distributed DBMS

- **Location transparency**
 - Even if data is located in different sites, it can be treated as if it were all located in one site
- **Replication transparency**
 - Even if the data is replicated, it maybe treated as if it were a single item on a single site

Objectives of Distributed DBMS

- **Failure transparency**
 - Either all the actions of a transaction are committed or none at all
- **Concurrency transparency**
 - For concurrent transactions, each transaction appears as if it were the only activity in the system

Middleware

- **Utility software that interfaces systems built with incompatible technologies**
- **Sometimes called plumbing because it connects two sides of an application and passes data between them**
- **Said to be the “slash” in client / server**

Classes of Middleware

- **Presentation middleware**
- **Application middleware**
- **Database middleware**

Presentation Middleware

- **Allows a programmer to build UI components that can talk to web browsers or desktop GUI**
- **Corresponds to presentation logic framework**
- **Example**
 - **HTTP allows the programmer to communicate with a browser through a standard interface**

Application Middleware

- **Enables two programmer-written processes (programs) on two different processors to communicate with one another in whatever way suited to the overall application**
- **Corresponds to the application logic framework**
- **Essential in multi-tier application development**
- **Examples**
 - Remote procedure calls (RPCs)
 - Message queues
 - Object request brokers (ORBs)

Database Middleware

- Allows a programmer to pass SQL commands to a database engine for processing
- Corresponds to the data manipulation logic framework
- Examples
 - ODBC
 - JDBC