

Chapter 9:

The Client/Server Database Environment

Objectives

- Definition of terms
- List advantages of client/server architecture
- Explain three application components: presentation, processing, and storage
- Suggest partitioning possibilities
- Distinguish between file server, database server, three-tier, and n-tier approaches
- Describe and discuss middleware
- Explain database linking via ODBC and JDBC

Client/Server Systems

- Networked computing model
- Processes distributed between clients and servers
- Client–Workstation (usually a PC) that requests and uses a service
- Server–Computer (PC/mini/mainframe) that provides a service
- For DBMS, server is a database server

Application Logic in C/S Systems

Presentation Logic

- Input–keyboard/mouse
- Output–monitor/printer

GUI Interface

Processing Logic

- I/O processing
- Business rules
- Data management

**Procedures, functions,
programs**

Storage Logic

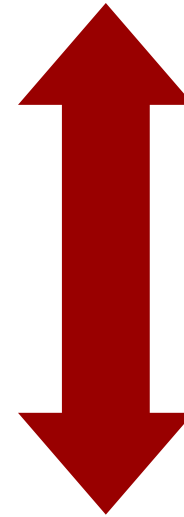
- Data storage/retrieval

DBMS activities

Client/Server Architectures

- File Server Architecture
- Database Server Architecture
- Three-tier Architecture

**Client does
extensive processing**

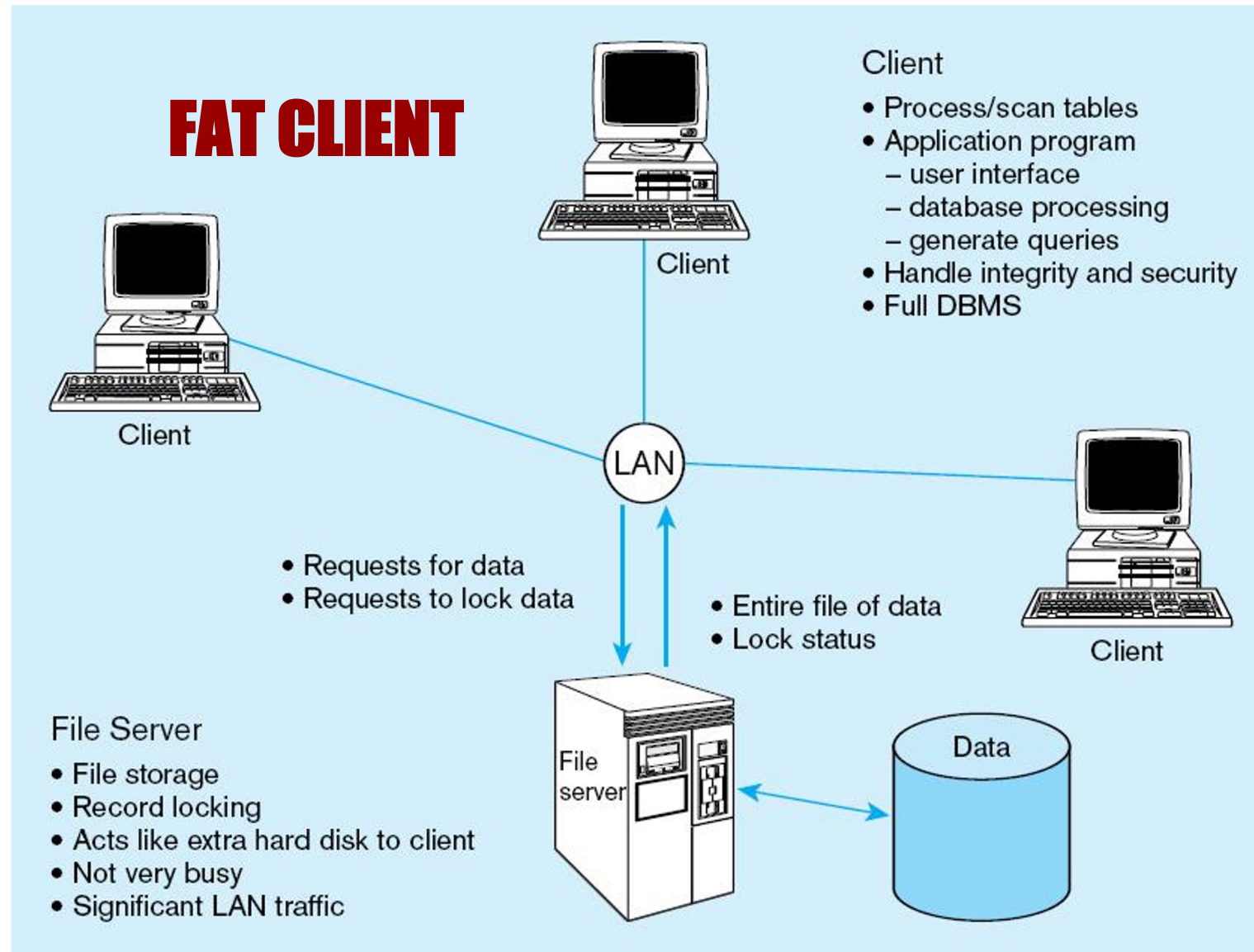


**Client does little
processing**

File Server Architecture

- All processing is done at the PC that requested the data
FAT CLIENT
- Entire files are transferred from the server to the client for processing
- Problems:
 - Huge amount of data transfer on the network
 - Each client must contain full DBMS
 - Heavy resource demand on clients
 - Client DBMSs must recognize shared locks, integrity checks, etc.

Figure 9-2 File server model



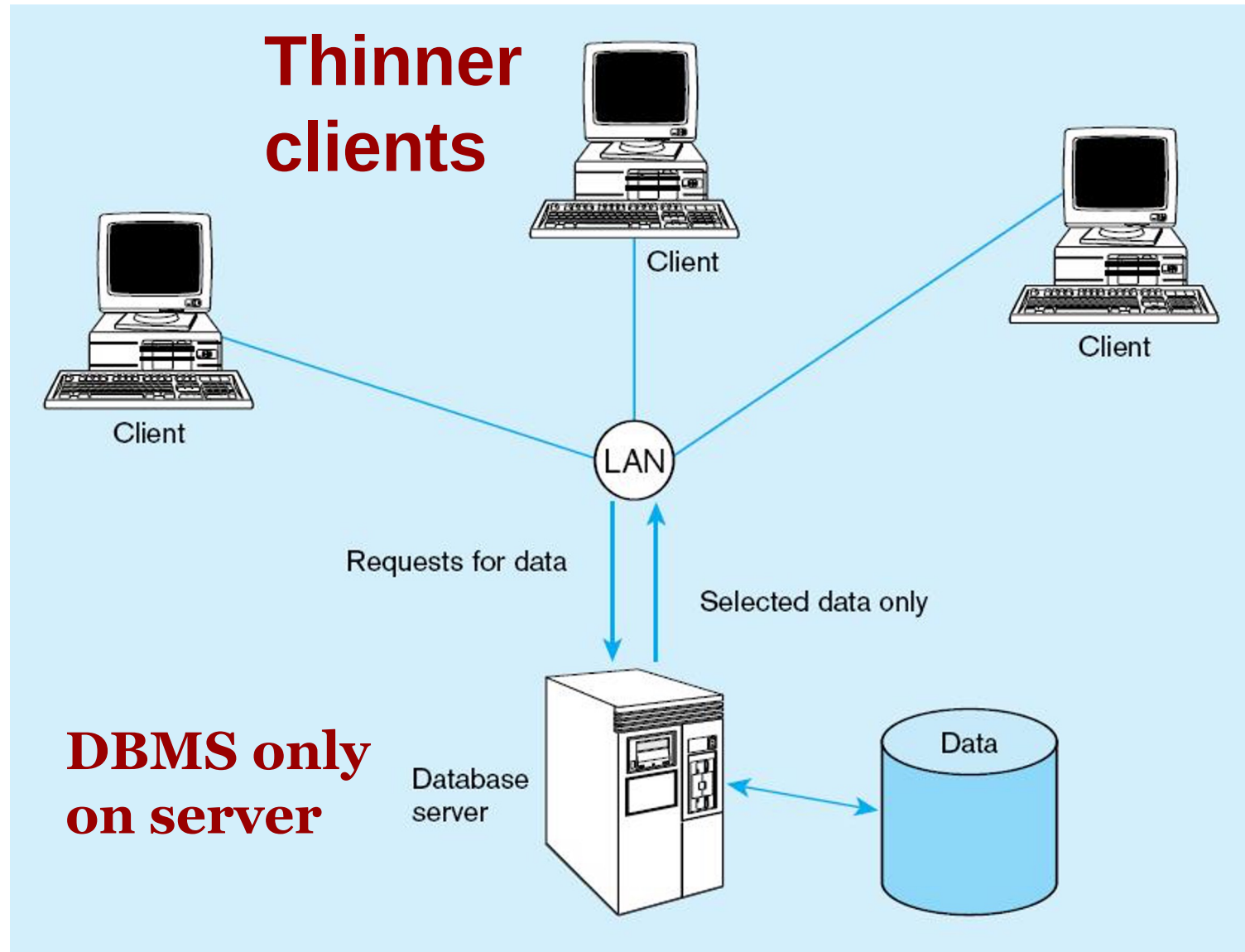
Two-Tier Database Server Architectures

- Client is responsible for
 - I/O processing logic
 - Some business rules logic
 - Server performs all data storage and access processing
- **DBMS is only on server**

Advantages of Two-Tier Approach

- Clients do not have to be as powerful
- Greatly reduces data traffic on the network
- Improved data integrity since it is all processed centrally
- **Stored procedures** □ DBMS code that performs some business rules done on server

Figure 9-3 Database server architecture (two-tier)



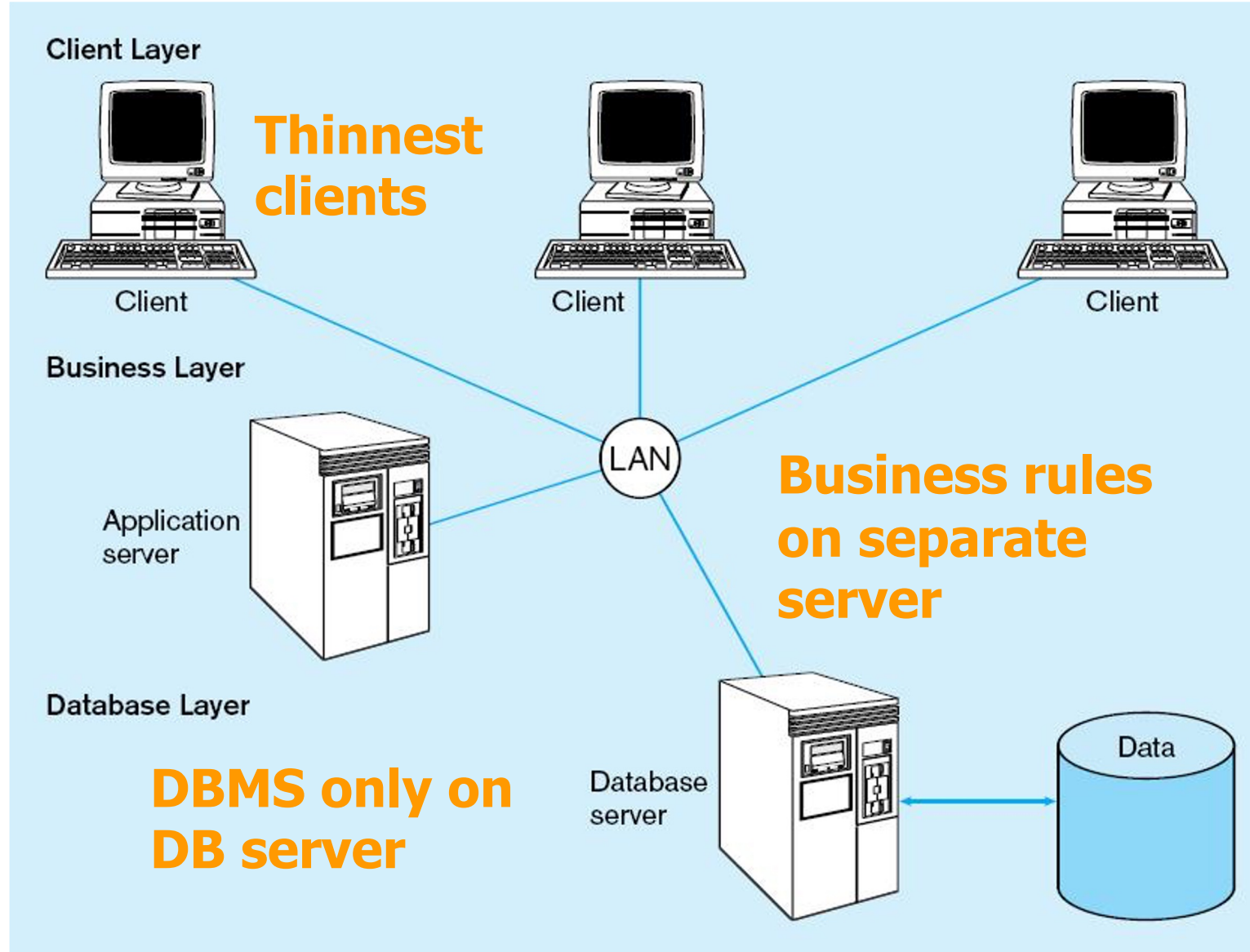
Three-Tier Architectures

Client	GUI interface (I/O processing)	<i>Browser</i>
Application server	Business rules	<i>Web Server</i>
Database server	Data storage	<i>DBMS</i>

Thin Client

- PC just for user interface and a little application processing. Limited or no data storage (sometimes no hard drive)

Figure 9-4a Generic three-tier architecture



Advantages of Three-Tier Architectures

- Scalability
- Technological flexibility
- Long-term cost reduction
- Better match of systems to business needs
- Improved customer service
- Competitive advantage
- Reduced risk

Application Partitioning

- Placing portions of the application code in different locations (client vs. server) AFTER it is written
- Advantages
 - Improved performance
 - Improved interoperability
 - Balanced workloads

Common Logic Distributions

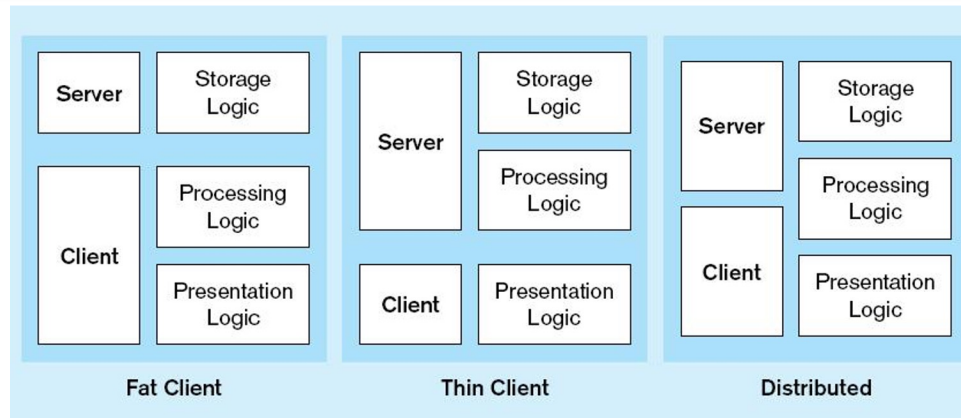
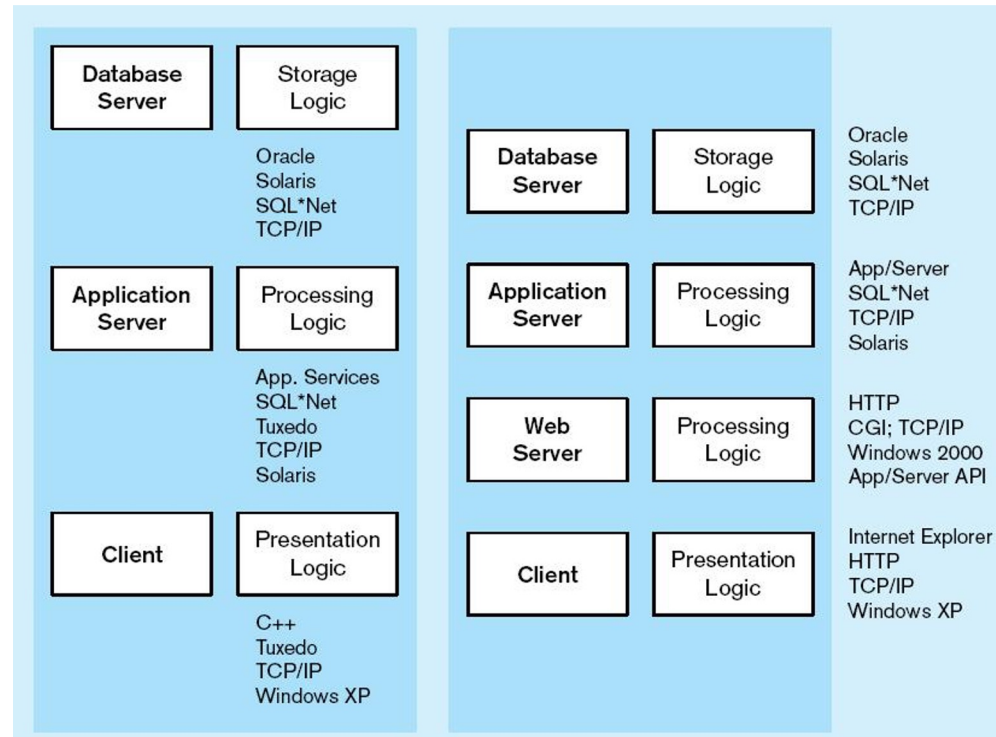


Figure 9-5a Two-tier client-server environments

Processing logic could be at client, server, or both

Figure 9-5b n -tier client-server environment

Processing logic will be at application server or Web server



Middleware

- Software that allows an application to *interoperate* with other software
- No need for programmer/user to understand internal processing
- Accomplished via *Application Program Interface* (API)

The “*glue*” that holds client/server applications together

Database Middleware

- ODBC—Open Database Connectivity
 - Most DB vendors support this
- OLE-DB
 - Microsoft enhancement of ODBC
- JDBC—Java Database Connectivity
 - Special Java classes that allow Java applications/applets to connect to databases

Client/Server Security

- Network environment □ complex security issues
- Security levels:
 - System-level password security
 - for allowing access to the system
 - Database-level password security
 - for determining access privileges to tables; read/update/insert/delete privileges
 - Secure client/server communication
 - via encryption

Keys to Successful Client-Server Implementation

- Accurate business problem analysis
- Detailed architecture analysis
- Architecture analysis before choosing tools
- Appropriate scalability
- Appropriate placement of services
- Network analysis
- Awareness of hidden costs
- Establish client/server security

Benefits of Moving to Client/Server Architecture

- Staged delivery of functionality speeds deployment
- GUI interfaces ease application use
- Flexibility and scalability facilitates business process reengineering
- Reduced network traffic due to increased processing at data source
- Facilitation of Web-enabled applications