Modern Systems Analysis and Design

Seventh Edition

Jeffrey A. Hoffer

Joey F. George

Joseph S. Valacich

Chapter 12

Designing Distributed and Internet Systems

Learning Objectives

- ✓ Define the key terms: client/server architecture, local area network LAN, distributed database, and middleware.
- ✓ Distinguish between file server and client/server environments and contrast how each is used in a LAN.
- ✓ Describe alternative designs for distributed systems and their trade-offs.
- ✓ Describe how standards shape the design of Internet-based systems.

Learning Objectives (Cont.)

- ✓ Describe options for ensuring Internet design consistency.
- ✓ Describe how site management issues can influence customer loyalty and trustworthiness as well as system security.
- ✓ Discuss issues related to managing online data, including context development, online transaction processing (OLTP), online analytical processing (OLAP), and data warehousing.

Designing Distributed and Internet Systems

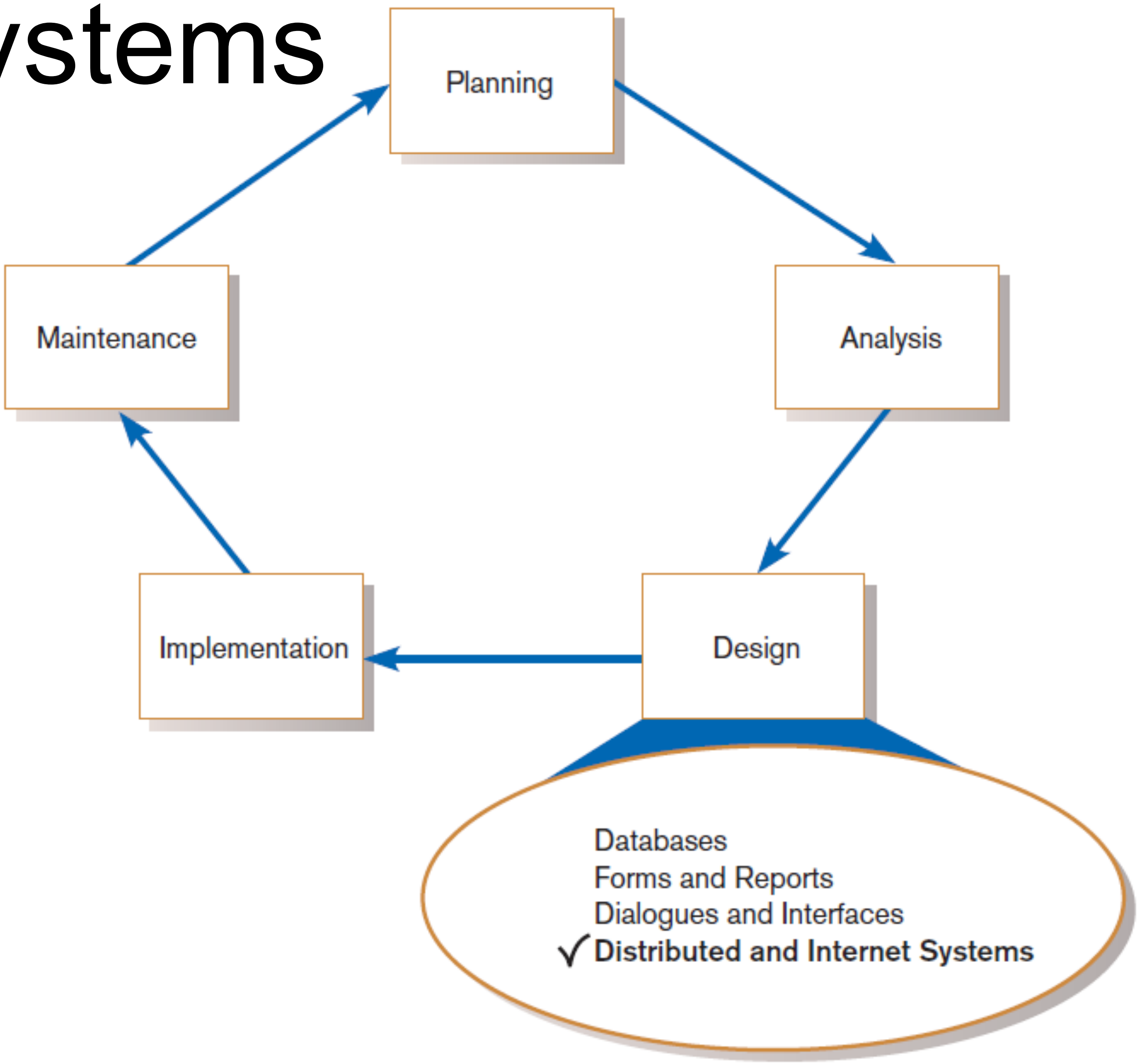


FIGURE 12-1
Systems development life cycle (SDLC)

The Process of Designing Distributed and Internet Systems

- This process is similar to designing single-location systems.
- Due to multi-location deployment, numerous design issues must be considered.
- There is more opportunity for failure due to number of components.
- Main issues involve ensuring reliability, availability, survivability, performance.

Deliverables and Outcome

- Document that consolidates system design information:
  - □ Description of each site
  - □ Description of data usage for each site
  - □ Description of business process for each site
  - □ Contrasts of alternative IS architectures for site, data and processing needs of each site

Designing Distributed Systems

- Distributed systems use:
  - LAN-based file server architecture.
  - Client/server architecture.

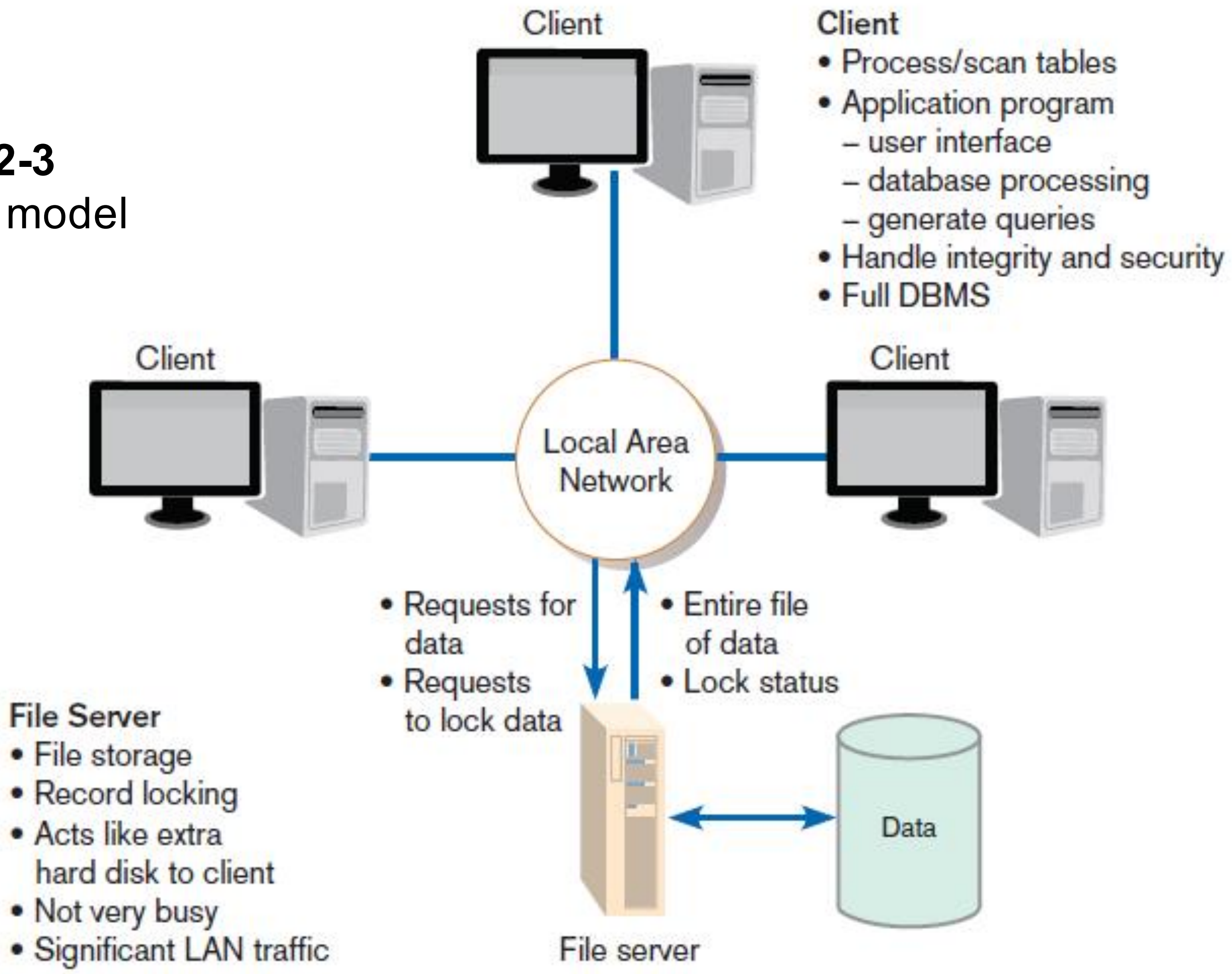
Designing Systems for Local Area Networks (LANs)

- **LAN**: the cabling, hardware, and software used to connect workstations, computers, and file servers located in a confined geographical area
  - Typically within one building or campus

File Servers

- **File server**: a device that manages file operations and is shared by each client PC attached to a LAN

FIGURE 12-3
File server model



Client
- Process/scan tables
- Application program
  - user interface
  - database processing
  - generate queries
- Handle integrity and security
- Full DBMS

- Requests for data
- Requests to lock data

- Entire file of data
- Lock status

File Server
- File storage
- Record locking
- Acts like extra hard disk to client
- Not very busy
- Significant LAN traffic

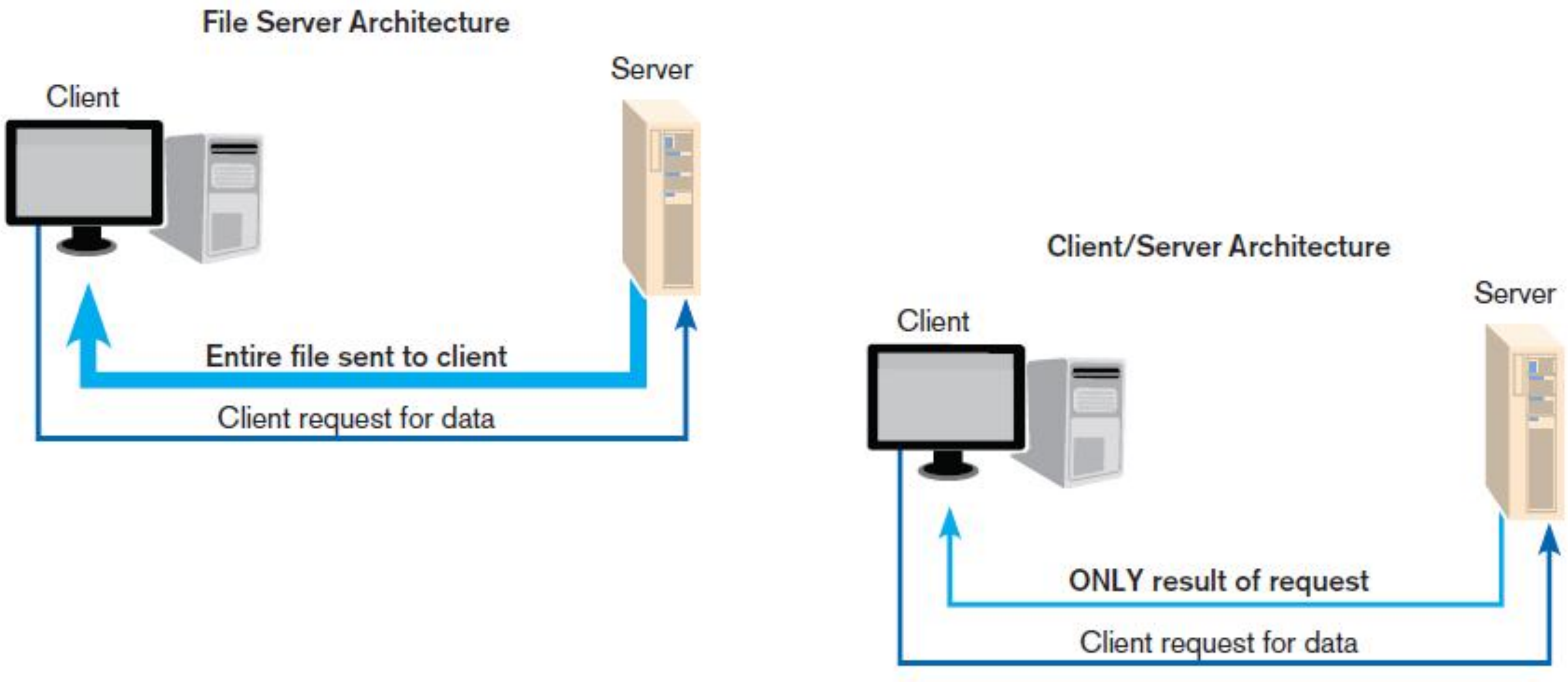
Limitations of File Servers

- Excessive data movement
  - Entire data table must be transferred, instead of individual records.
- Need for powerful client workstations
  - Each client workstation must devote memory to a full DBMS.
- Decentralized data control
  - Record concurrency control, recovery, and security are complicated.

Designing Systems for a Client/Server Architecture

- **Client/server architecture**: a LAN-based computing environment in which central database server or engine performs all database commands sent to it from client workstations, and application programs on each client concentrate on user interface functions

File Server vs. Client/Server



File Server vs. Client/Server (cont.)

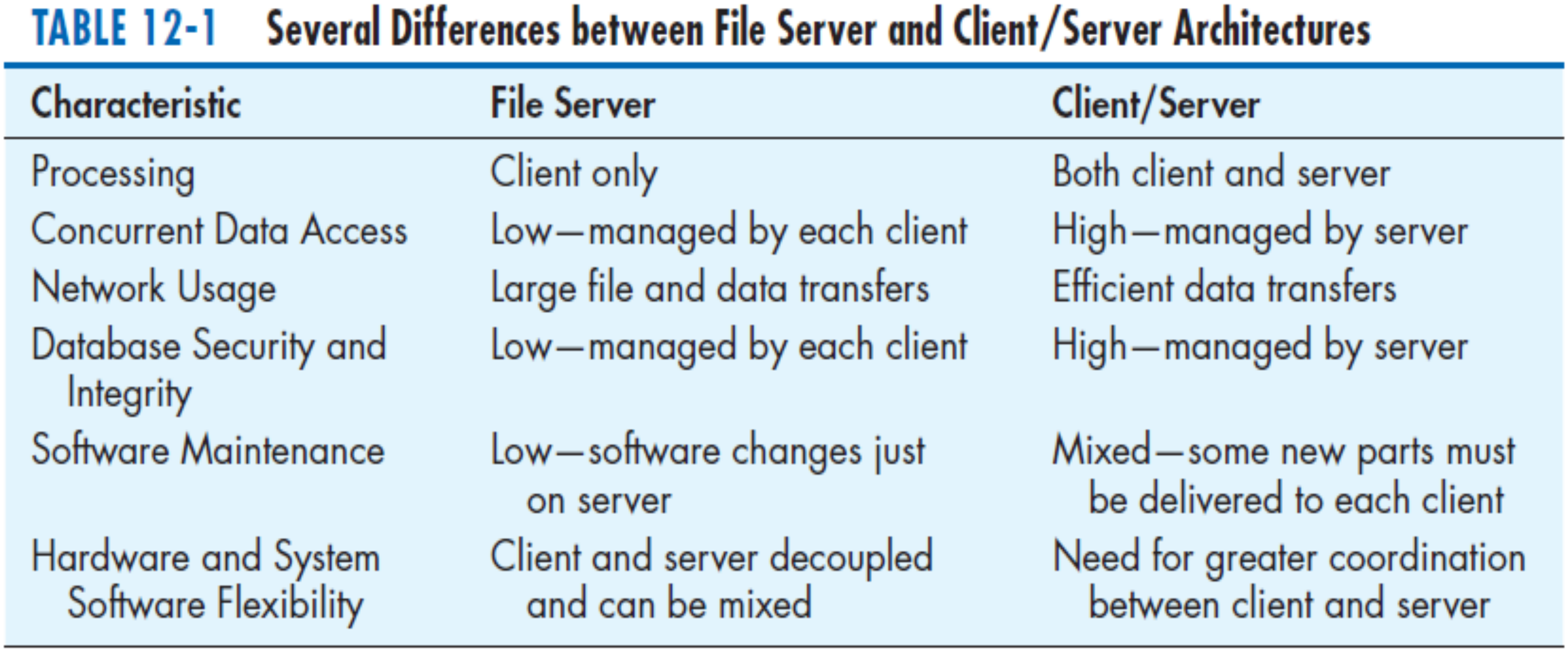


TABLE 12-1 Several Differences between File Server and Client/Server Architectures

Characteristic	File Server	Client/Server
Processing	Client only	Both client and server
Concurrent Data Access	Low—managed by each client	High—managed by server
Network Usage	Large file and data transfers	Efficient data transfers
Database Security and Integrity	Low—managed by each client	High—managed by server
Software Maintenance	Low—software changes just on server	Mixed—some new parts must be delivered to each client
Hardware and System Software Flexibility	Client and server decoupled and can be mixed	Need for greater coordination between client and server

Designing Systems for a Client/Server Architecture

- Application processing is divided between client and server.
- Client manages the user interface.
- Database server is responsible for data storage and query processing.

Designing Systems for a Client/Server Architecture (Cont.)

- **Database engine**: the (back-end) portion of the client/server database system running on the server that provides database processing and shared access functions

Designing Systems for a Client/Server Architecture (Cont.)

- **Client**: the (front-end) portion of the client/server database system that provides the user interface and data manipulation functions

Designing Systems for a Client/Server Architecture (Cont.)

- **Application program interface** (**API**): software building blocks that are used to ensure that common system capabilities, such as user interfaces and printing, as well as modules are standardized to facilitate data exchange between clients and servers
  - Common API interface can be used by any kind of DBMS (MySQL, Sybase, or Oracle)

Client/Server Advantages and Cautions

- Advantages
  - Leverages benefits of microcomputer technology
  - Processing performed close to data source
    - Improves response time
    - Reduces network traffic
  - Facilitates use of GUIs
  - Encourages acceptance of open systems

Client/Server Advantages and Cautions

- Cautions
  - Difficult migration from file server to client/server
  - Compatibility issues
  - Limited system design and performance monitoring tools

Advanced Forms of Client/Server Architectures

- **Application server**: a computing server where data analysis functions primarily reside
- **Three-tiered client/server**: advanced client/server architectures in which there are three logical and distinct applications—data management, presentation, and analysis—that are combined to create a single information system

Advanced Forms of Client/Server Architectures (Cont.)

- **Middleware**: a combination of hardware, software, and communication technologies that bring data management, presentation, and analysis together into a three-tiered client/server environment

Middleware Environments

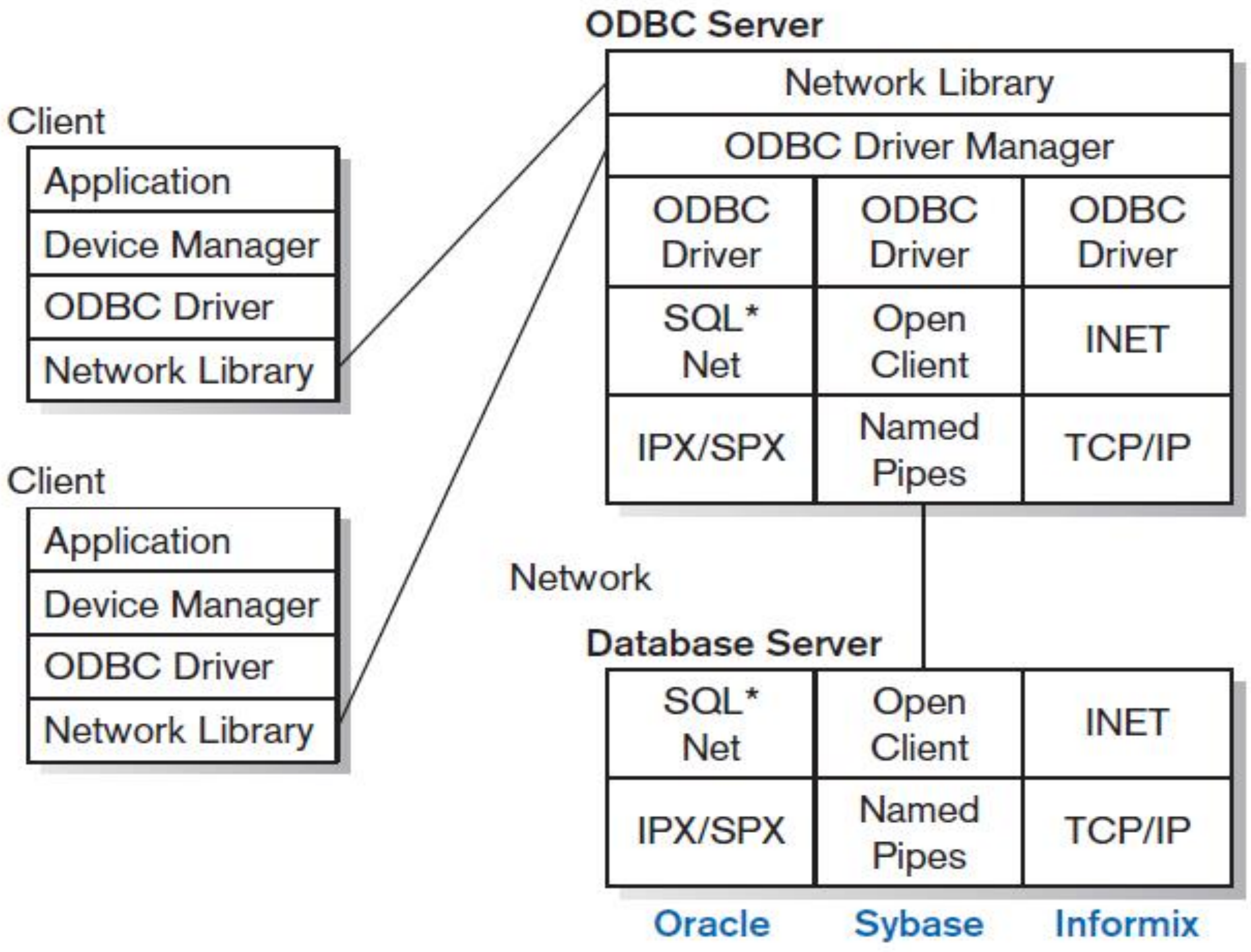


FIGURE 12-6
ODBC middleware environment

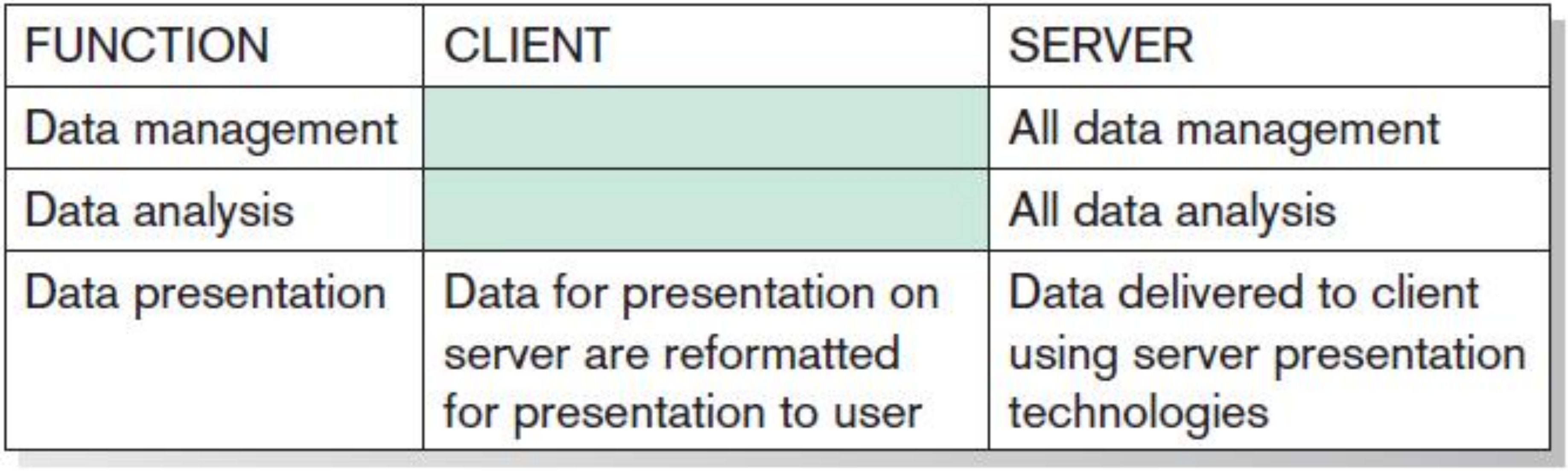
Advanced Forms of Client/Server Architectures (Cont.)

- Applications can be partitioned in a way that best fits the organizational computing need
- Easier customization: application code resides on application server, so change done only in one place
- Easier maintenance: data analysis is separate from user interface, so changing one can be done independently of the other

Approaches to Designing Client/Server Architectures

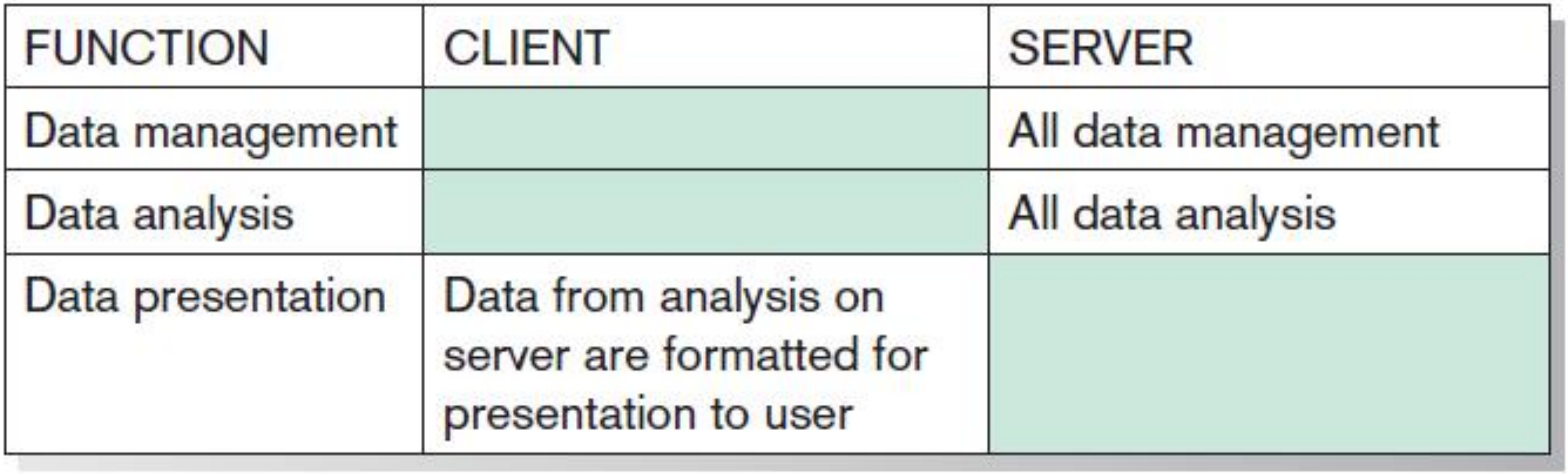
- Distributed Presentation
- Remote Presentation
- Remote Data Management
- Distributed Function
- Distributed Database
- Distributed Processing

Distributed Presentation



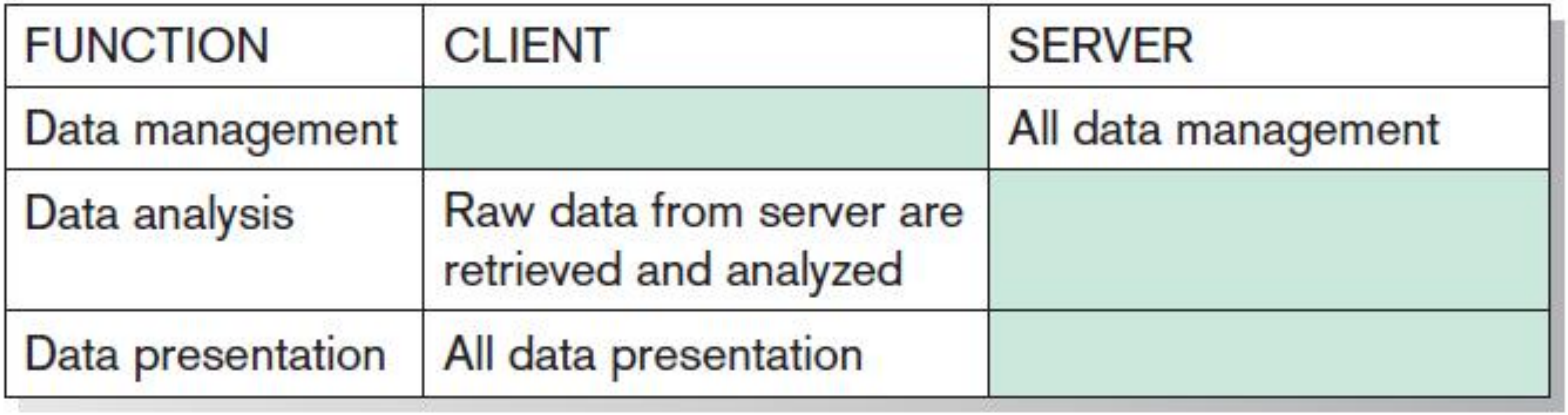
FUNCTION	CLIENT	SERVER
Data management		All data management
Data analysis		All data analysis
Data presentation	Data for presentation on server are reformatted for presentation to user	Data delivered to client using server presentation technologies

Remote Presentation



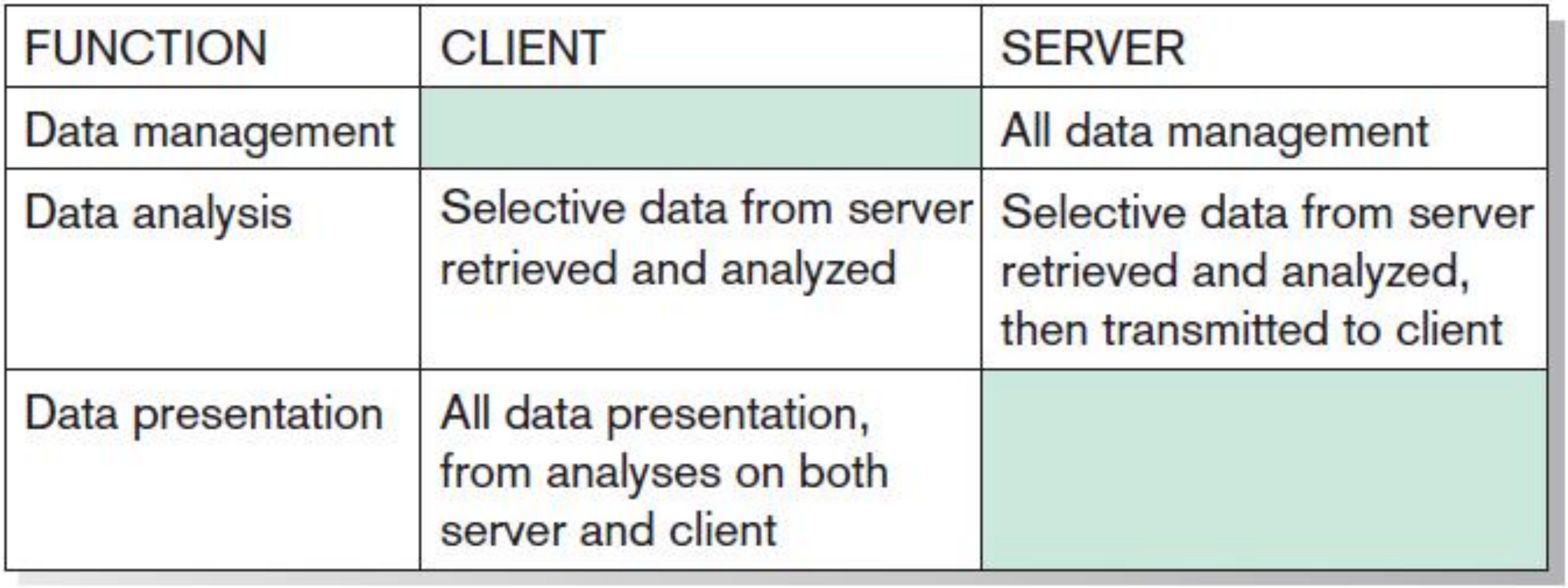
FUNCTION	CLIENT	SERVER
Data management		All data management
Data analysis		All data analysis
Data presentation	Data from analysis on server are formatted for presentation to user	

Remote Data Management



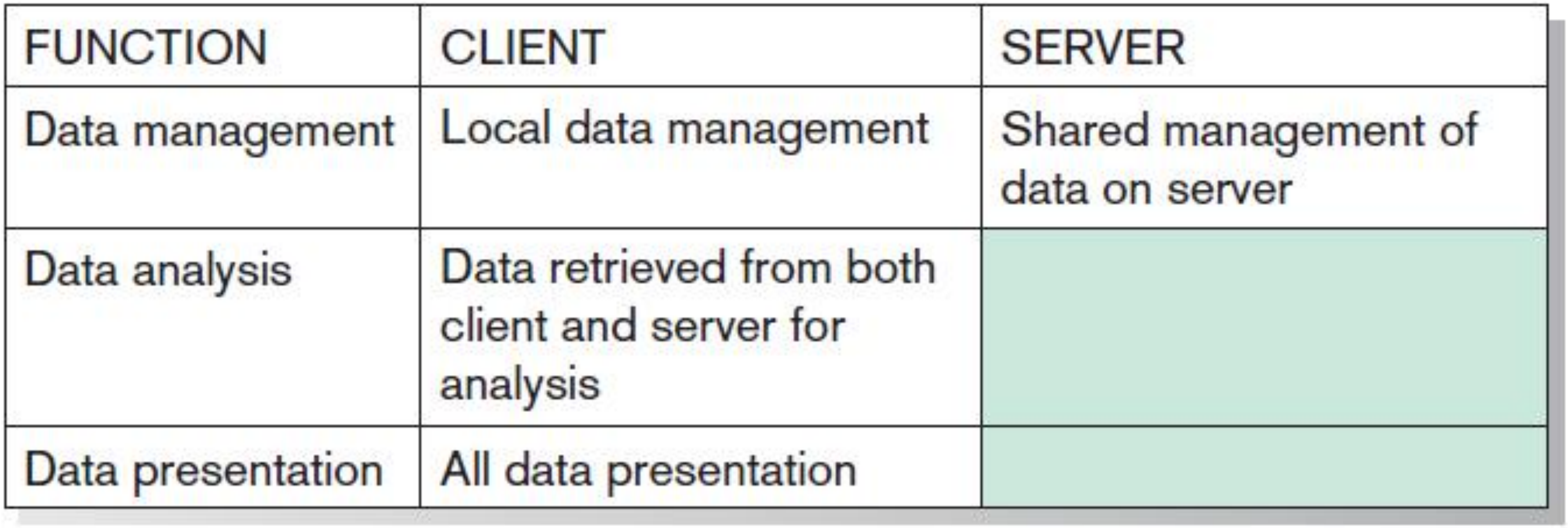
FUNCTION	CLIENT	SERVER
Data management		All data management
Data analysis	Raw data from server are retrieved and analyzed	
Data presentation	All data presentation	

Distributed Function



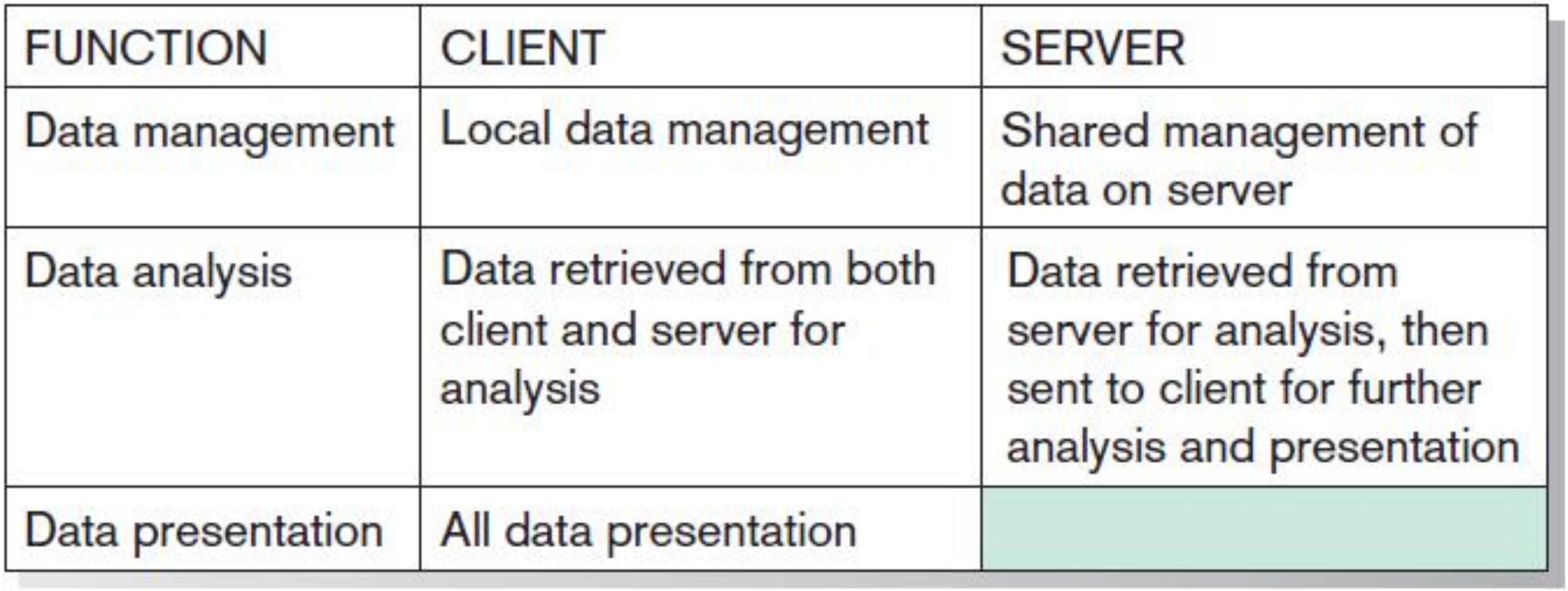
FUNCTION	CLIENT	SERVER
Data management		All data management
Data analysis	Selective data from server retrieved and analyzed	Selective data from server retrieved and analyzed, then transmitted to client
Data presentation	All data presentation, from analyses on both server and client	

Distributed Database



FUNCTION	CLIENT	SERVER
Data management	Local data management	Shared management of data on server
Data analysis	Data retrieved from both client and server for analysis	
Data presentation	All data presentation	

Distributed Processing



FUNCTION	CLIENT	SERVER
Data management	Local data management	Shared management of data on server
Data analysis	Data retrieved from both client and server for analysis	Data retrieved from server for analysis, then sent to client for further analysis and presentation
Data presentation	All data presentation	

Designing Internet Systems

- Most new system development focuses on Internet-based applications (for internal processing, business-to-business, and business-to-consumer).
- Main design issues: standards, separating content from display, future evolution, site consistency, site management and online data management

Standards Drive the Internet

- Internet design is simpler than client/server due to proliferation of standards.

Standards Drive the Internet (Cont.)

- Types of Standards:
  - □ **Domain naming (BIND):** a method for translating domain names into Internet Protocol (IP) addresses
    - "B" refers to Berkeley, Ca. where first developed.
    - See *www.isc.org/products/BIND/bind-history.html*
  - □ **Hypertext Transfer Protocol** (**HTTP**): a communication protocol for exchanging information on the Internet
  - □ **Hypertext Markup Language** (**HTML**): the standard language for representing content on the Web via command tags

Separating Content and Display

- HTML has limitations due to format orientation of tags.
- **eXtensible Markup Language** (**XML**): an Internet-authoring language that allows designers to create customized tags, enabling the definition, transmission, validation, and interpretation of data between applications

Future Evolution

- **Thin client**: a client device designed so that most processing and data storage occur on the server
- Often using wireless mobile devices
  - *Wireless Markup Language* (WML): a wireless version of HTML

FIGURE 12-8
Thin clients used to access the Internet

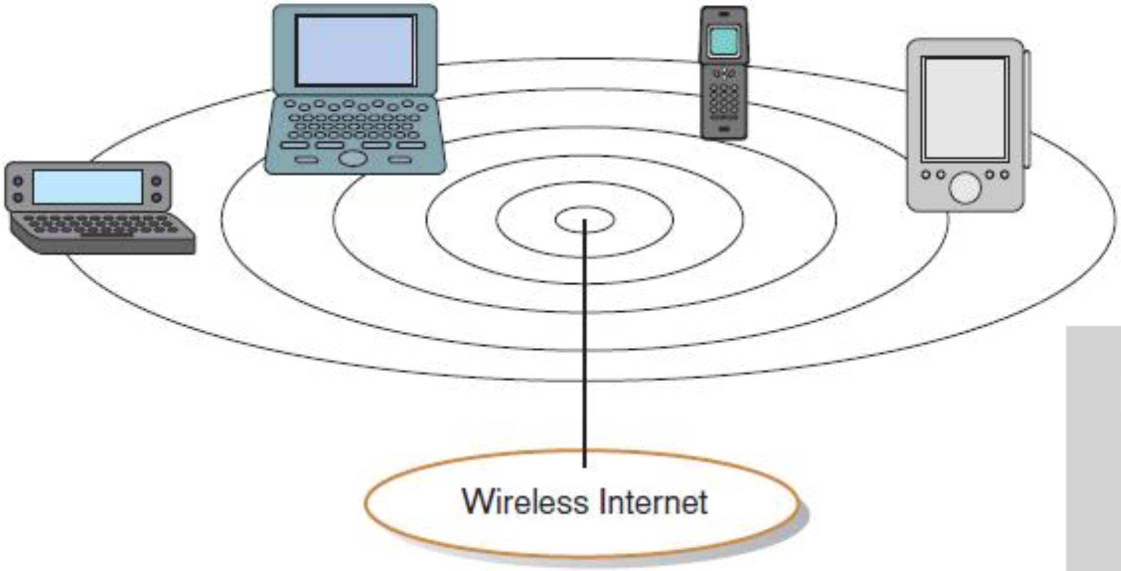


FIGURE 12-9
Thin clients typically have a limited screen size.
(*Source:* tele52 / Shutterstock)

Site Consistency

- Professionalism requires a consistent look-and-feel across all pages of a Web site.
- **Cascading Style Sheets** (**CSSs**): a set of style rules that tells a Web browser how to present a document

Site Consistency (Cont.)

- **eXtensible Style Language** (**XSL**): a specification for separating style from content when generating HTML documents
  - *Methods for transforming* XML documents into a generic comprehensive form
  - *Methods for formatting* the generic comprehensive form into a device-specific form

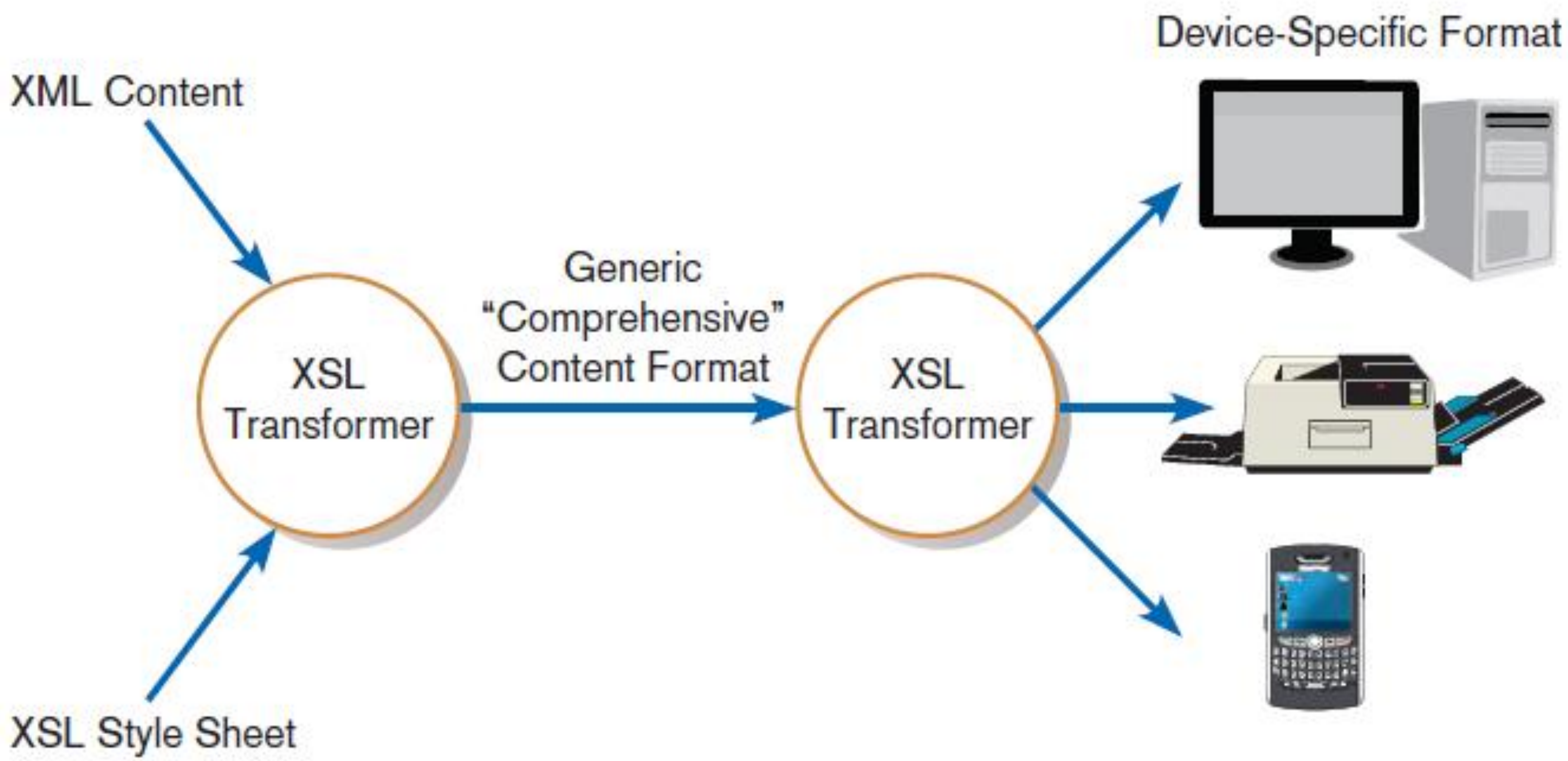


FIGURE 12-11
Combining XML data with XSL style sheet to format content

Other Site Consistency Issues

- Two key issues should be considered:
  - *Use unique titles.*
  - *Choose words carefully.*
- Major problem is users do not know where they are going when they follow a hyperlink.
  - You should have link titles to explain hyperlinks.

Design Issues Related to Site Management

- Customer Loyalty and Trustworthiness
  - *Design quality*
  - *Up-front disclosure*
  - *Comprehensive, correct, and current content*
  - *Connected to the rest of the Web*

Link Titles

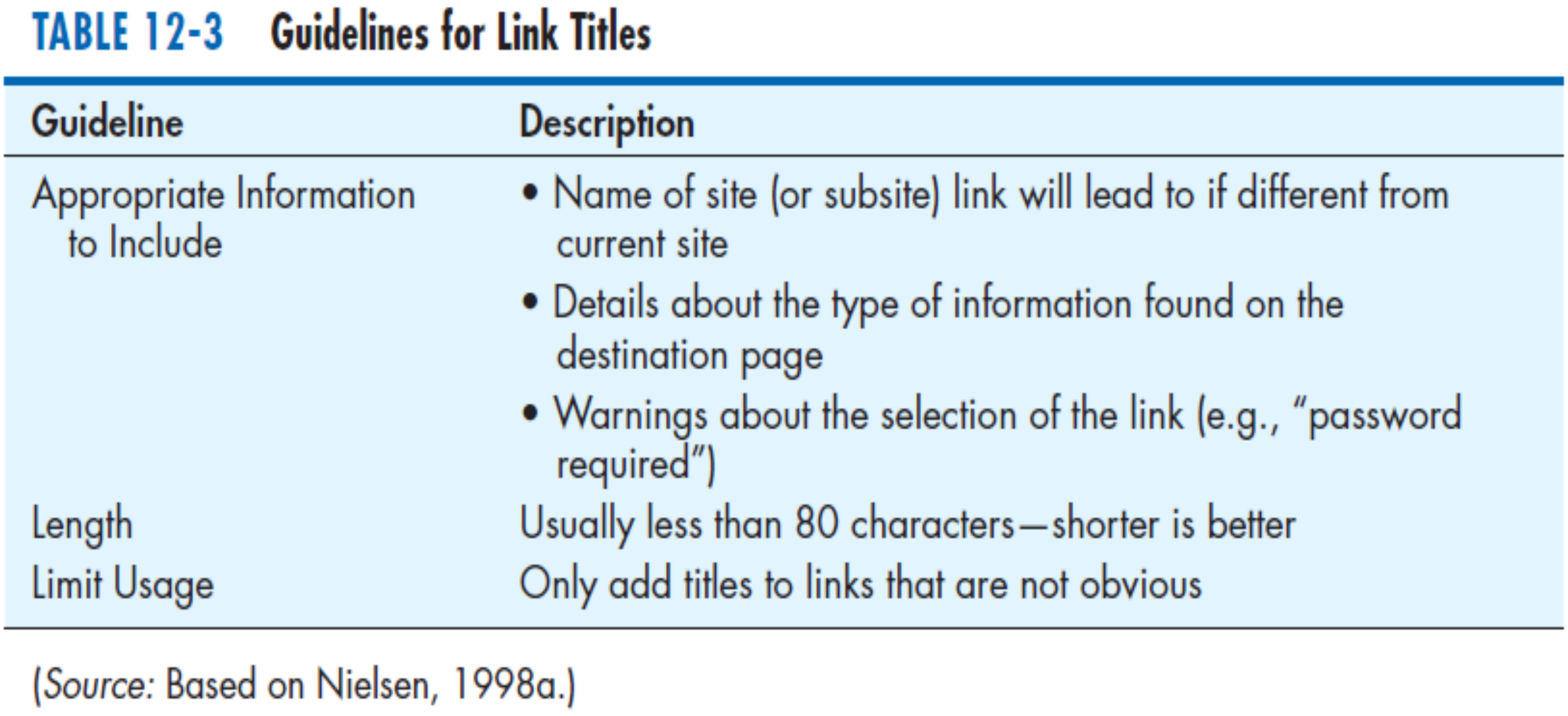


TABLE 12-3 Guidelines for Link Titles

Guideline	Description
Appropriate Information to Include	• Name of site (or subsite) link will lead to if different from current site • Details about the type of information found on the destination page • Warnings about the selection of the link (e.g., “password required”)
Length	Usually less than 80 characters—shorter is better
Limit Usage	Only add titles to links that are not obvious

(*Source:* Based on Nielsen, 1998a.)

Customer Loyalty and Trustworthiness

- **Personalization**: providing Internet content to a user based upon knowledge of that customer
- **Customization**: Internet sites that allow users to customize the content and look of the site based on their personal preferences

Web Pages Must Live Forever

- *Customer Bookmarks*
- *Links from Other Sites*
- *Search Engine Referrals*
- *Old Content Adds Value*
- System Security vs. Ease of Use
  - "Remember my password"
  - Use of cookies

Online Data Management

- **Context development**: a method that helps analysts to better understand how a system fits within the existing business activities and data
- **Integration depth**: a measurement of how far into the existing technology infrastructure a system penetrates
- **Organizational breadth**: a measurement that tracks the core business functions affected by a system

Online Transaction Processing (OLTP)

- **Online transaction processing** (**OLTP**): the immediate automated responses to the requests of users
- Designed to handle multiple concurrent transactions
- Plays a large role in electronic commerce applications

Online Analytical Processing (OLAP)

- **Online analytical processing** (**OLAP**): the use of graphical software tools that provide complex analysis of data stored in a database
- OLAP server is the chief component
- Good for time series and trend analysis
- Enables user to “drill-down” into the data

Merging Transaction and Analytical Processing

- **Operational systems**: systems that are used to interact with customers and run a business in real time
- **Informational systems**: systems designed to support decision making based on a stable point-in-time or historical data

Data Warehousing

- **Data warehouse**: a subject-oriented, integrated, time-variant, nonvolatile collection of data used in support of management decision making

Data Warehousing (Cont.)

- Key features
  - *Subject-oriented*: organized around key subjects
  - *Integrated*: data are collected from many operational systems and made to conform to standards
  - *Time-variant*: data contains a time dimension
  - *Nonvolatile*: data cannot be updated by users

Data Warehousing (Cont.)

- Four basic building steps:
  - Extract data from various source system files and databases.
  - Transform, integrate, and load the data.
  - Data warehouse is a read-only environment.
  - Users access via query languages and analytical tools.

Data Warehousing (Cont.)

- Two level Architecture:
  - Data warehouse and decision support environment
- Three-level architecture:
  - Operational systems and data
  - An enterprise data warehouse
  - Data marts

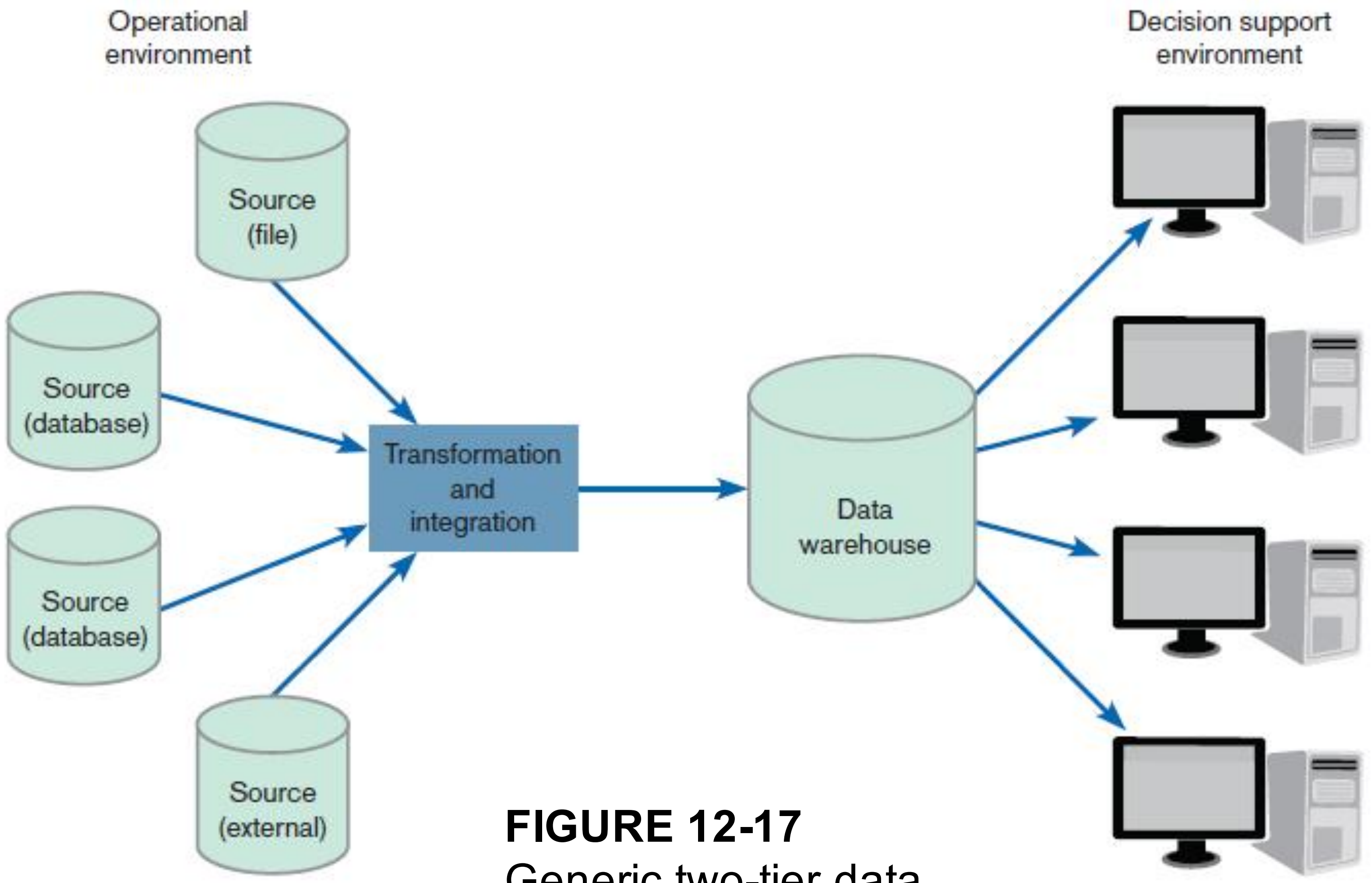


FIGURE 12-17
Generic two-tier data warehouse architecture

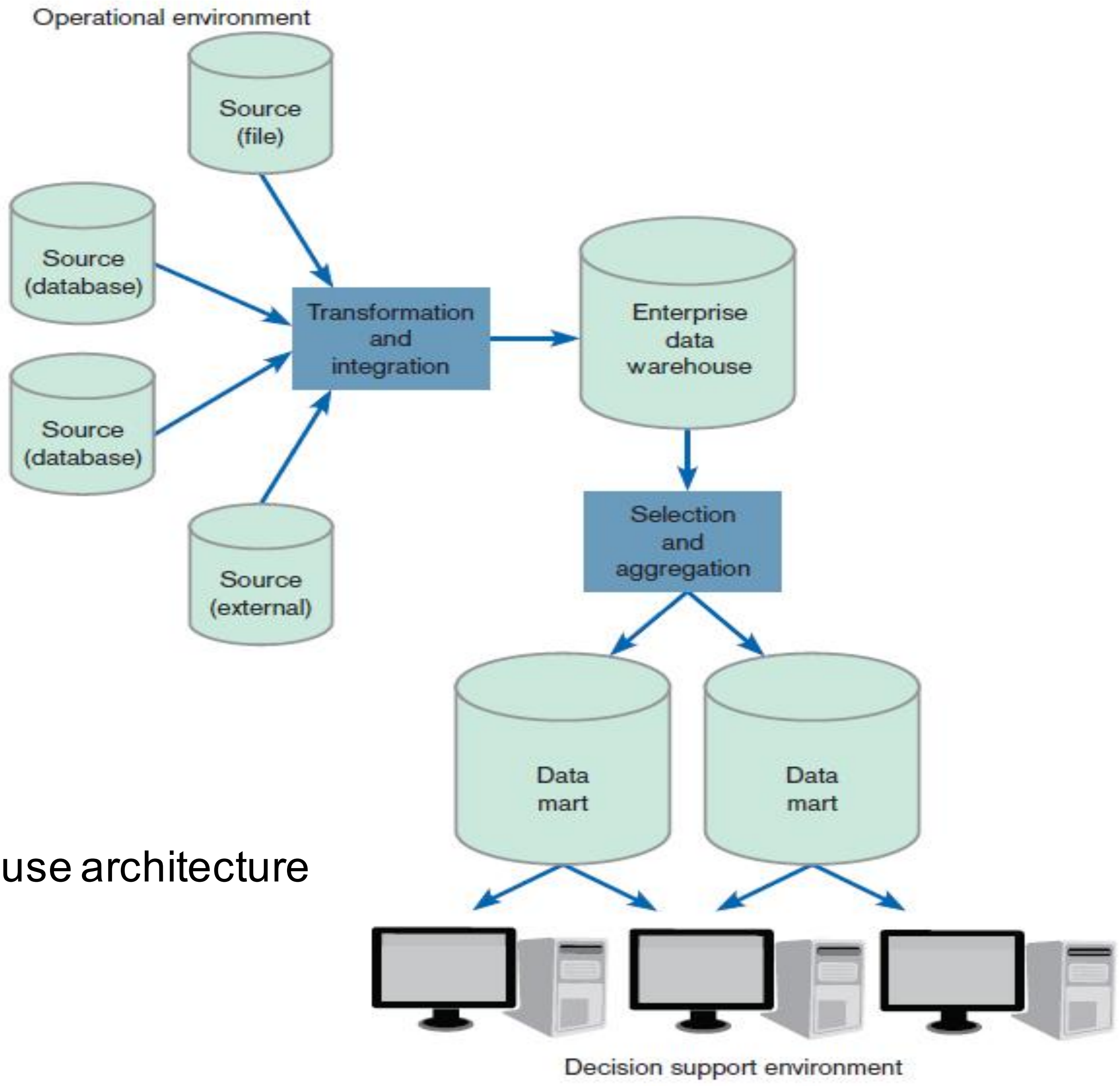

Operational environment
Source (file)
Source (database)
Source (database)
Source (external)
Transformation and integration
Enterprise data warehouse
Selection and aggregation
Data mart
Data mart
Decision support environment

FIGURE 12-18
Three-tier warehouse architecture

Data Warehousing (Cont.)

- **Enterprise data warehouse** (**EDW**): a centralized, integrated data warehouse that is the control point and single source of all data made available to end users for decision support applications throughout the entire organization
- **Data mart**: a data warehouse that is limited in scope; its data are obtained by selecting and (where appropriate) summarizing data from the enterprise data warehouse

Web Site Content Management

- **Content management system (CMS)**: a special type of software application for collecting, organizing, and publishing Web site content

Electronic Commerce Application: Designing a Distributed Advertisement Server for PVF’s WebStore

- Benefits for including advertising:
  - Potential to increase revenue generated from the WebStore
  - Potential to create cross-promotions and alliances with other online commerce systems
  - Potential to provide customers with improved service when looking for additional products that accessorize PVF’s product line

Advertising on PVF’s WebStore

- List of advertisement system concerns:
  - Advertisement must be served quickly so that site performance is not affected.
  - Advertisement must be uniform in size and resolution, so as not to disrupt the site layout.
  - Advertisement links must not redirect the user’s browser away from the WebStore.

Designing the Advertising Component

- Transactional requirements are to
  - □ Determine which advertisements apply, based on where the user is in the WebStore.
  - □ Personalize the advertisement if the identity of user and preferences are known.
  - □ Check for any seasonal or promotional advertisements.
  - □ Log the transaction.

Designing the Management Reporting Component

- Queries for top-management:
  - □ “How many women, when shopping for desks, clicked on an advertisement for lamps?”
  - □ “How many advertisements were served to shoppers looking at filing cabinets?”
  - □ “How many people clicked on the first advertisement they saw?”
  - □ “How many people clicked on an advertisement and then purchased something from the WebStore?”

Summary

- In this chapter you learned how to:
- ✓ Define the key terms: client/server architecture, local area network LAN, distributed database, and middleware.
- ✓ Distinguish between file server and client/server environments and contrast how each is used in a LAN.
- ✓ Describe alternative designs for distributed systems and their trade-offs.
- ✓ Describe how standards shape the design of Internet-based systems.

Summary (Cont.)

- ✓ Describe options for ensuring Internet design consistency.
- ✓ Describe how site management issues can influence customer loyalty and trustworthiness as well as system security.
- ✓ Discuss issues related to managing online data, including context development, online transaction processing (OLTP), online analytical processing (OLAP), and data warehousing.