Web Databases

Lecture 13

Learning Outcomes

- Define the Client/Server architecture
- Introduction to web database, database connectivity and web-to-database middleware
- What is Extensible Markup Language (XML) and why it is important for Web database development
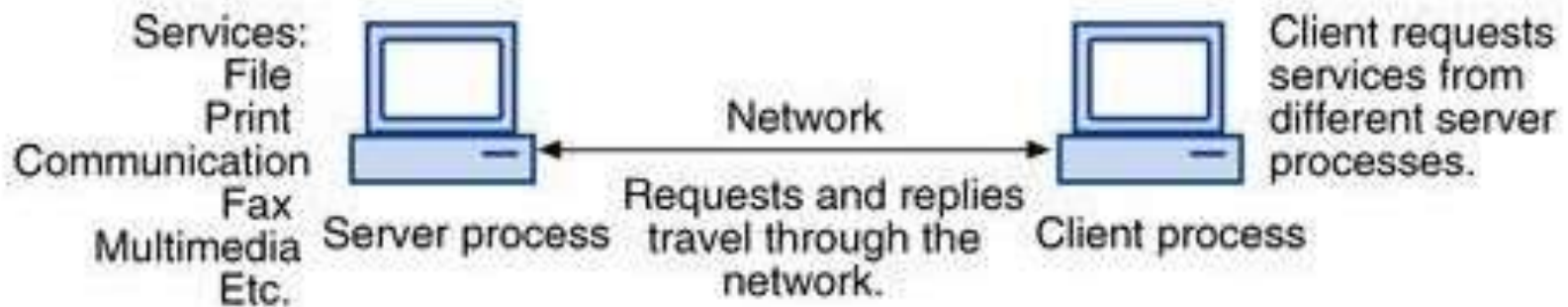- Cloud computing
- Types of cloud services

What Is Client/Server Computing?

- Client/server is based on the distribution of functions between two types of independent and autonomous processes; servers and clients.

- A client is any process that requests specific services from server processes.

- A server is a process that provides requested services for clients.

- Both clients and servers **can reside** in the same computer or in different computers connected by a network.

What Is Client/Server Computing?

- The key to client/server power is **where the requested processing takes place.**
  - In **mainframe** systems, all processing takes place on the mainframe, and the terminal is used to display the data screens.
  - The client/server environment provides a clear separation of server and client processes.
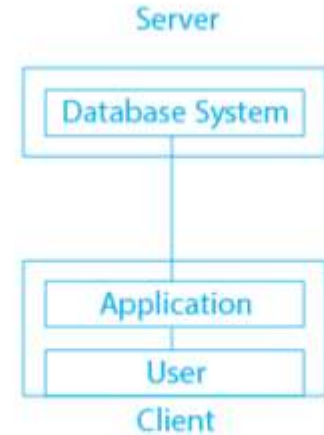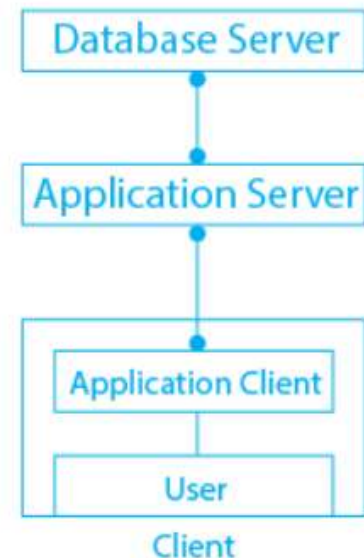
A Basic Client/Server Computing Model

Services:
File
Print
Communication
Fax
Multimedia
Etc.

Server process

Network

Requests and replies travel through the network.

Client requests services from different server processes.

Client process

A BASIC CLIENT/SERVER COMPUTING MODEL

CIT6114 – Database Fundamentals

What Is Client/Server Computing?

- Client/Server Systems may also classified as:
  - *2-Tier Client/Server System*, a client requests services directly from the server.
  - *3-Tier Client/Server System*, the client's requests are handled by intermediate servers.

Server

Database System

Application

User

Client

In a 2- tier architecture, client connects directly to Database

Database Server

Application Server

Application Client

User

Client

3- tier Architecture

CIT6114 – Database Fundamentals

| S.NO	Two-Tier Database Architecture	Three-Tier Database Architecture
1	It is a Client-Server Architecture.	It is a Web-based application.
2	In two-tier, the application logic is either buried inside the user interface on the client or within the database on the server (or both).	In three-tier, the application logic or process resides in the middle-tier, it is separated from the data and the user interface.
3	Two-tier architecture consists of two layers : Client Tier and Database (Data Tier).	Three-tier architecture consists of three layers : Client Layer, Business Layer and Data Layer.
4	It is easy to build and maintain.	It is complex to build and maintain.
5	Two-tier architecture runs slower.	Three-tier architecture runs faster.
6	It is less secured as client can communicate with database directly.	It is secured as client is not allowed to communicate with database directly.
7	It results in performance loss whenever the users increase rapidly.	It results in performance loss whenever the system is run on Internet but gives more performance than two-tier architecture.
8	Example – Contact Management System created using MS-Access or Railway Reservation System,	Example – Designing registration form which contains text box, label, button or a large

Client/Server Architecture

- Three Components of Client/Server Architecture
    - The *client* is any computer process that requests services from the server. It is also known as the front-end application.
    - The *server* is any computer process providing services to the clients. The server is also known as the back-end application.
    - The *communication middleware* is any computer process(es) through which clients and servers communicate. It is also known as middleware or communications layer.

How Client And Server Components Interact

Client process sends SQL request through communications middleware.   Communications middleware routes SQL request to database server process.   Database server process receives request, validates it, and executes it.

HOW CLIENT AND SERVER COMPONENTS INTERACT

CIT6114 – Database Fundamentals
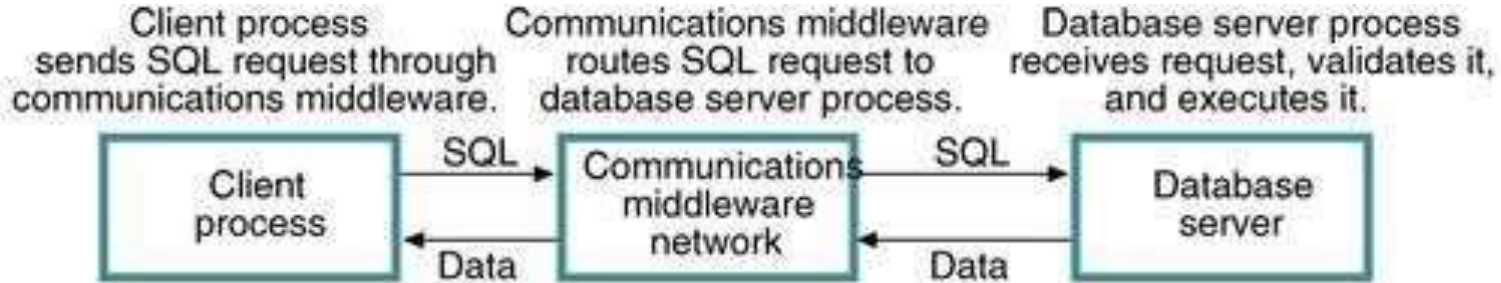
An Example Of Client/Server Architecture
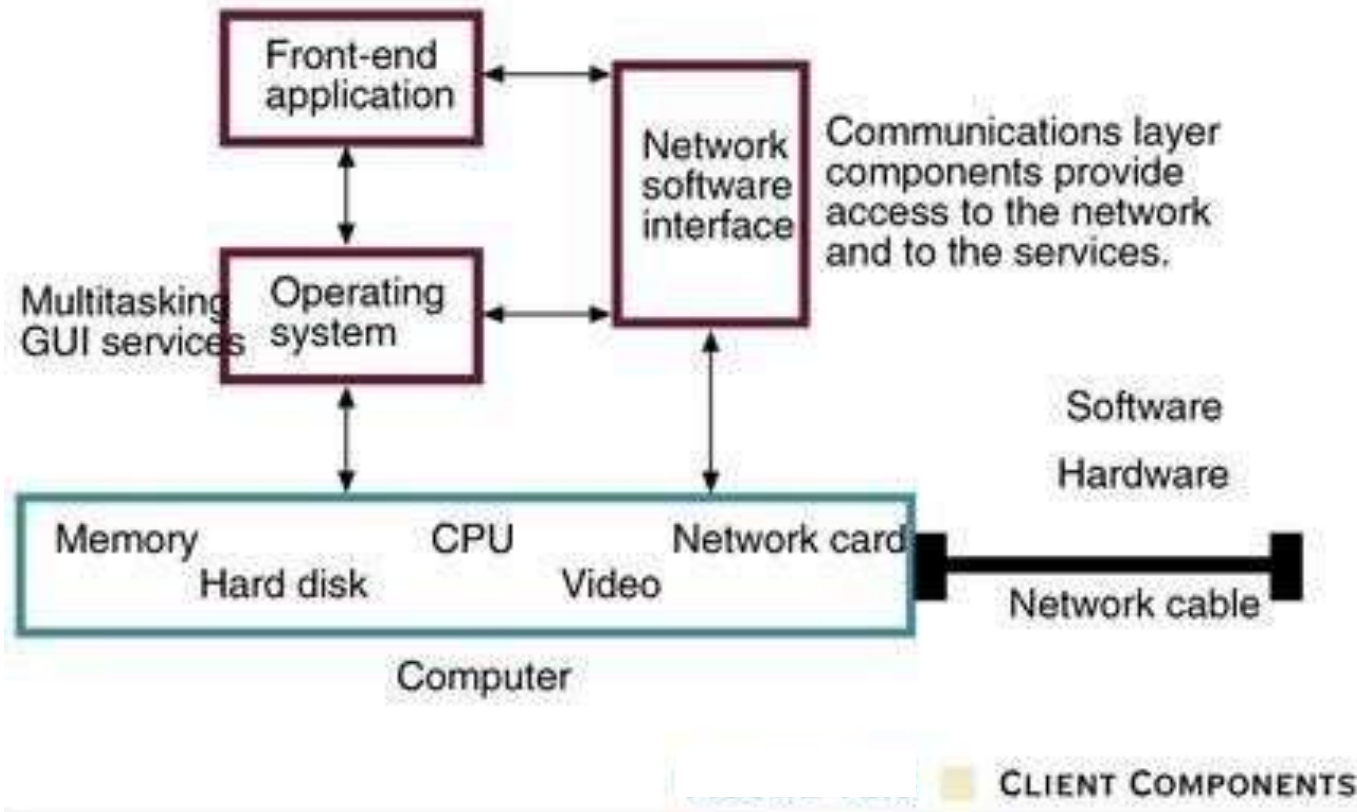

AN EXAMPLE OF CLIENT/SERVER ARCHITECTURE

CIT6114 – Database Fundamentals
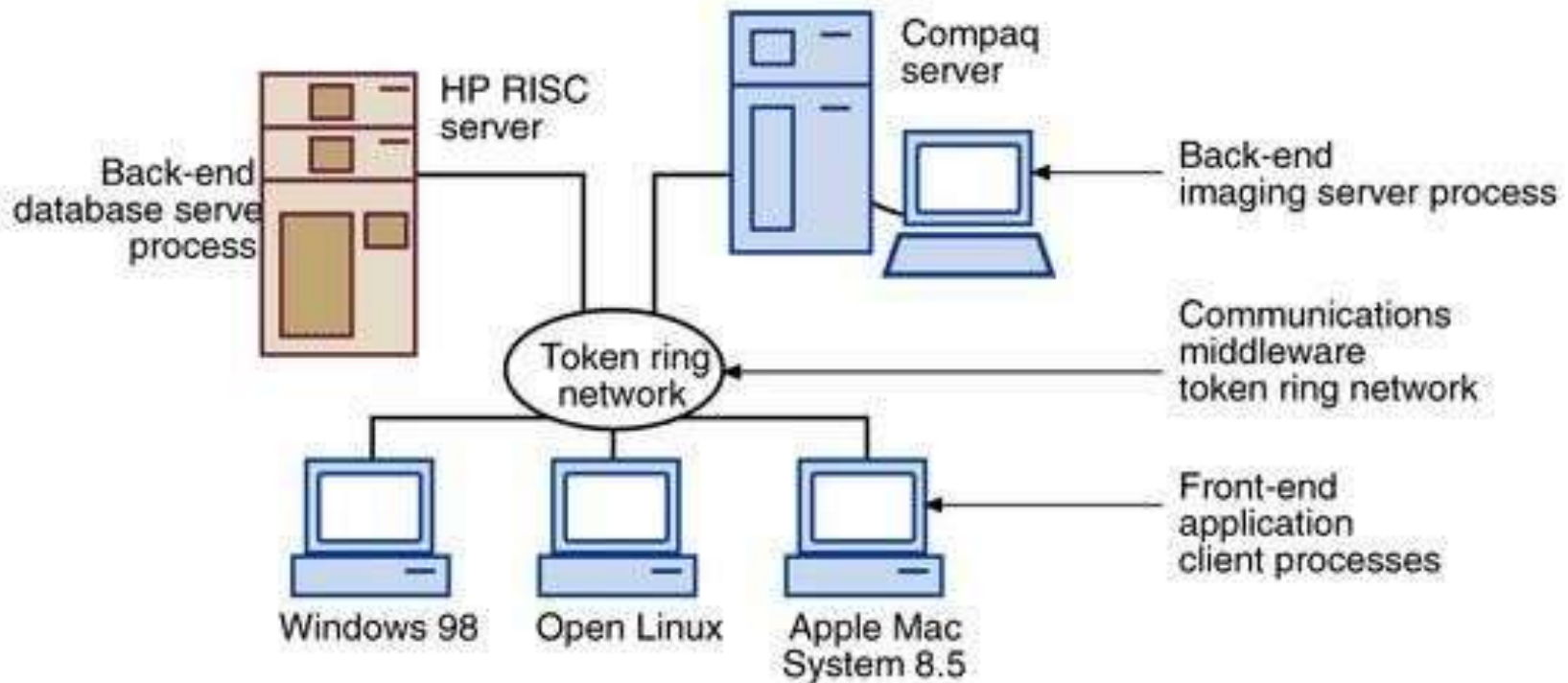
Client/Server Architecture

- **Client Components**
  - Powerful hardware
  - An operating system capable of multitasking
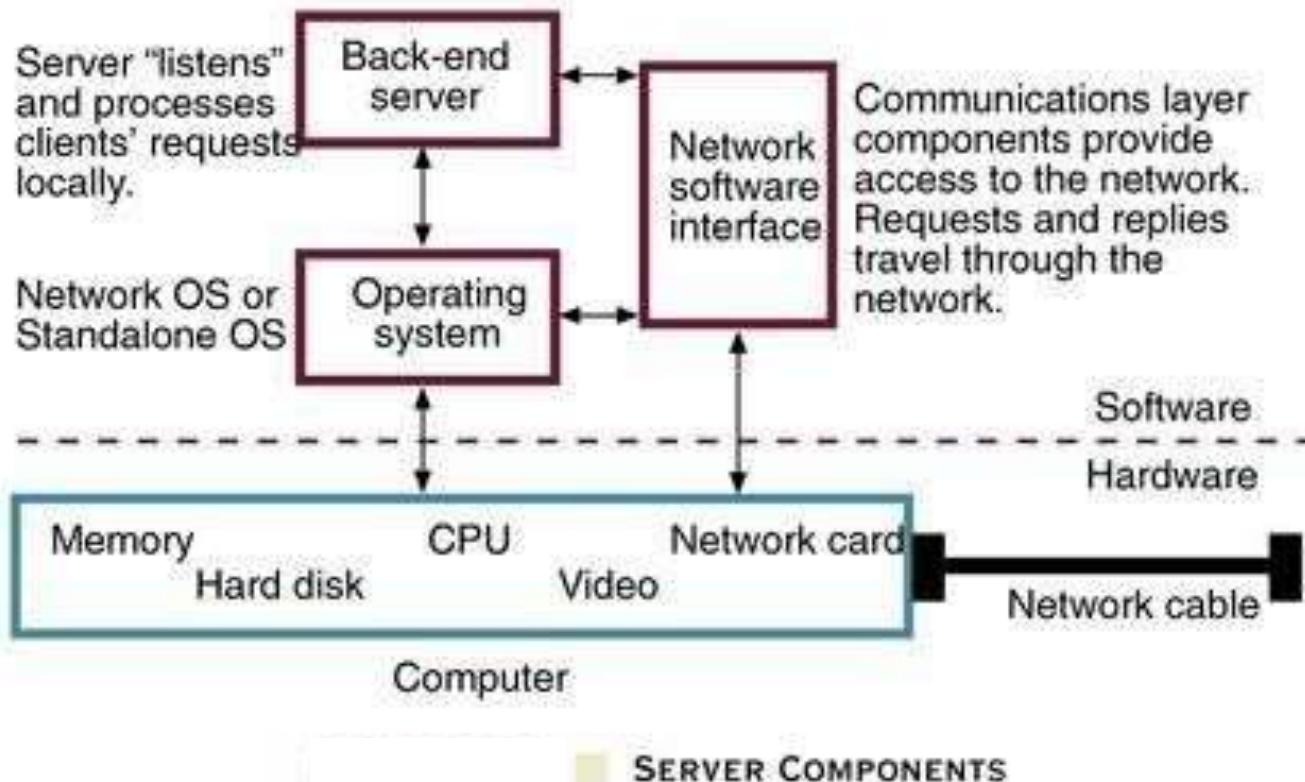 - A graphical user interface (GUI)
  - Communications capability

CIT6114 – Database Fundamentals

Client Components

		Software
		Hardware

Front-end application ↔ **Network software interface**

Communications layer components provide access to the network and to the services.

Multitasking GUI services — **Operating system**

Memory   CPU   Network card
Hard disk   Video

Network cable

Computer

CLIENT COMPONENTS

Client/Server Architecture

- ## Server Components

  - *File services*

  - *Print services*

  - *Fax services*

  - *Communications services*

  - *Database services*

Server Components

Server "listens" and processes clients' requests locally.

Network OS or Standalone OS

Back-end server

Operating system

Network software interface

Communications layer components provide access to the network. Requests and replies travel through the network.

Software

Hardware

Memory   CPU   Network card

Hard disk   Video

Network cable

Computer

SERVER COMPONENTS

Client/Server Advantages

- **Less expensive** than alternate minicomputer or mainframe solutions

    - Considerable cost advantage to offloading applications development to PCs

- **More** people in job market have PC **skills** than mainframe skills

- Data analysis and query **tools** facilitate interaction with DBMSs

Client/Server Disadvantages

- More **complex** environment

- Increase in number of users and processing sites causes **security** problems

- Possible to spread data access to much wider circle of users

  - Increases **demand** for people with broad knowledge of computers and software

 - Increases burden of **training** and **cost** of maintaining the environment

The Web

- Information presented on Web pages can contain:
  - Text
  - Graphics
  - Pictures
  - Audio/Video

- It can also contain hyperlinks to other Web pages, which allow users to navigate in a non-sequential way through information

- Web pages are written using HTML

The Web

- Web consists of network of computers that can act in two roles:
  - as **servers**, providing information
  - as **clients** (browsers), requesting information

- Protocol that governs exchange of information between Web server and browser is **HyperText Transfer Protocol (HTTP)**

- Locations within documents identified as **Uniform Resource Locator (URL)**

- Much of Web's success is due to its *simplicity* and *platform-independence*

CIT6114 – Database Fundamentals

Basic Components of Web Environment

CIT6114 – Database Fundamentals
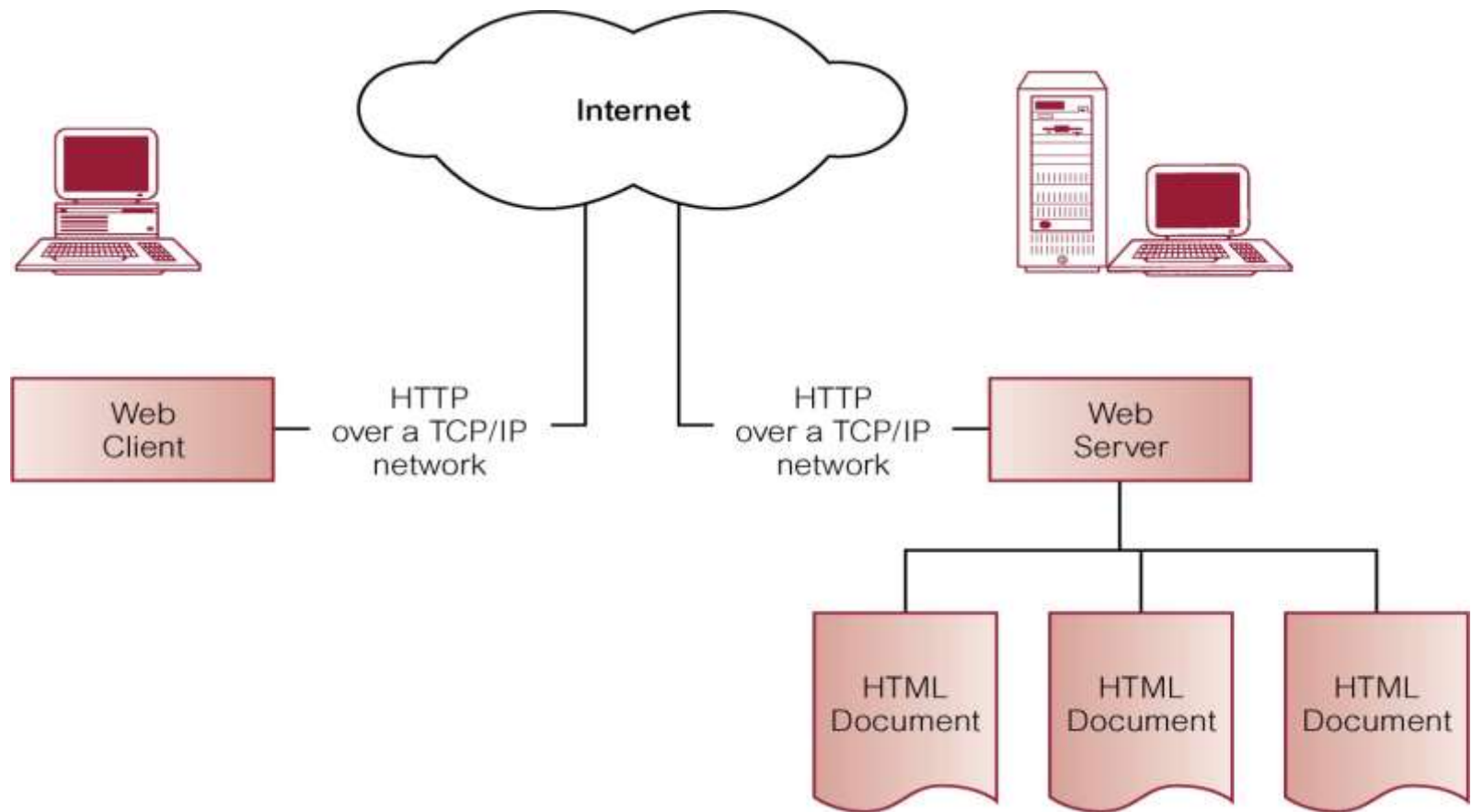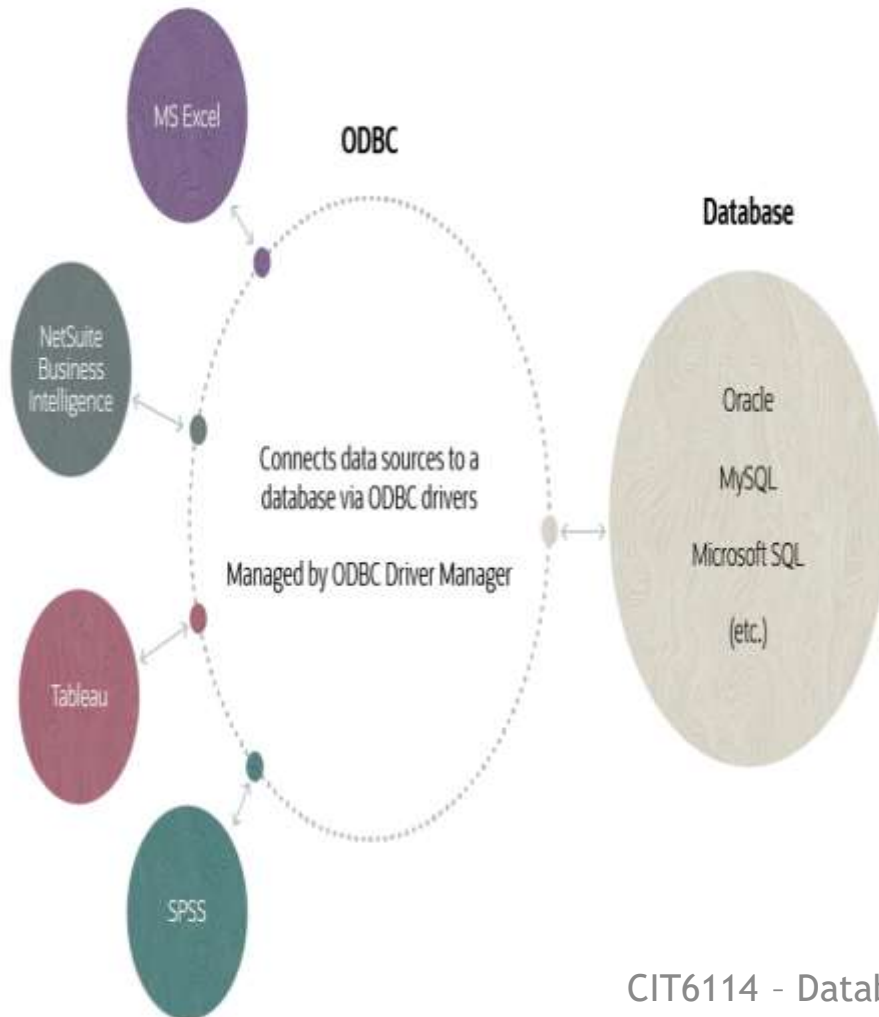
Database Connectivity

- Mechanisms by which application programs connect and communicate with data sources

  - Also known as database middleware

- Database connectivity interface/middleware:

  - `ODBC`,`OLE-DB`, `ADO`

  - JDBC

Open Database Connectivity (ODBC)

- Microsoft's implementation of a superset of SQL Access Group Call Level Interface (CLI)

- Widely supported database connectivity interface

- Any Windows application can access relational data sources

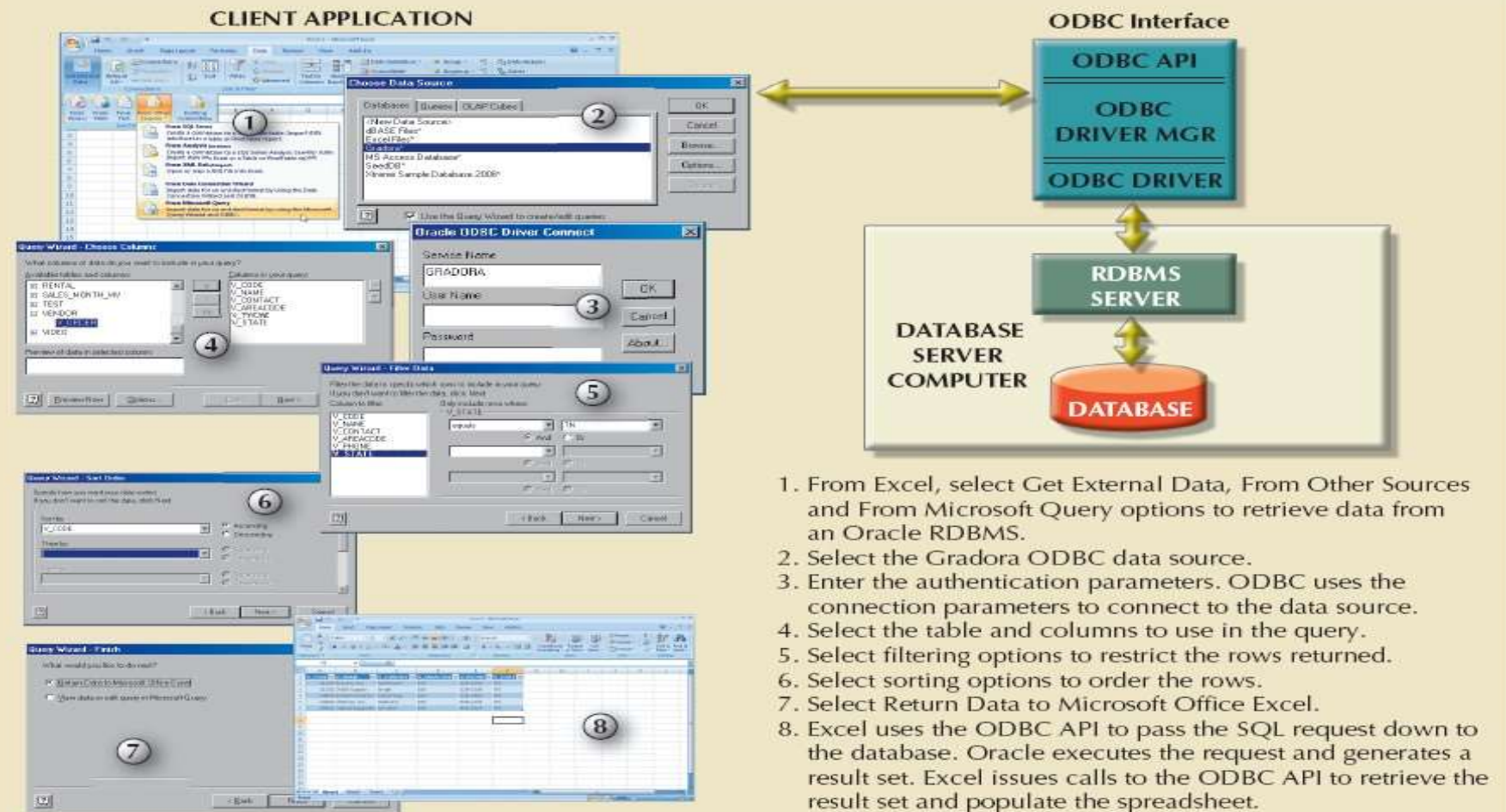- Uses SQL via standard application programming interface (API)

Open Database Connectivity (ODBC)

How ODBC Works

ODBC

Connects data sources to a database via ODBC drivers

Managed by ODBC Driver Manager

Database: Oracle, MySQL, Microsoft SQL (etc.)

Client Applications: MS Word, MS Access, MS Excel

ODBC API

ODBC Driver Manager

ODBC Database Driver: Oracle Driver, MS SQL Driver, ODBC Driver

Oracle, MS SQL, Access

Database vendors provide ODBC database drivers so Windows applications can access their respective databases.

SOURCE: Course Technology/Cengage Learning

Open Database Connectivity (ODBC)

FIGURE 11.4 MS Excel uses ODBC to connect to an Oracle database

1. From Excel, select Get External Data, From Other Sources and From Microsoft Query options to retrieve data from an Oracle RDBMS.
2. Select the Gradora ODBC data source.
3. Enter the authentication parameters. ODBC uses the connection parameters to connect to the data source.
4. Select the table and columns to use in the query.
5. Select filtering options to restrict the rows returned.
6. Select sorting options to order the rows.
7. Select Return Data to Microsoft Office Excel.
8. Excel uses the ODBC API to pass the SQL request down to the database. Oracle executes the request and generates a result set. Excel issues calls to the ODBC API to retrieve the result set and populate the spreadsheet.

SOURCE: Course Technology/Cengage Learning

Object Linking and Embedding for Database (OLE-DB)

- Database middleware that adds object-oriented functionality for access to data

- Series of objects provides low-level database connectivity for applications

FIGURE 11.5 OLE-DB architecture

SOURCE: Course Technology/Cengage Learning

CIT6114 – Database Fundamentals
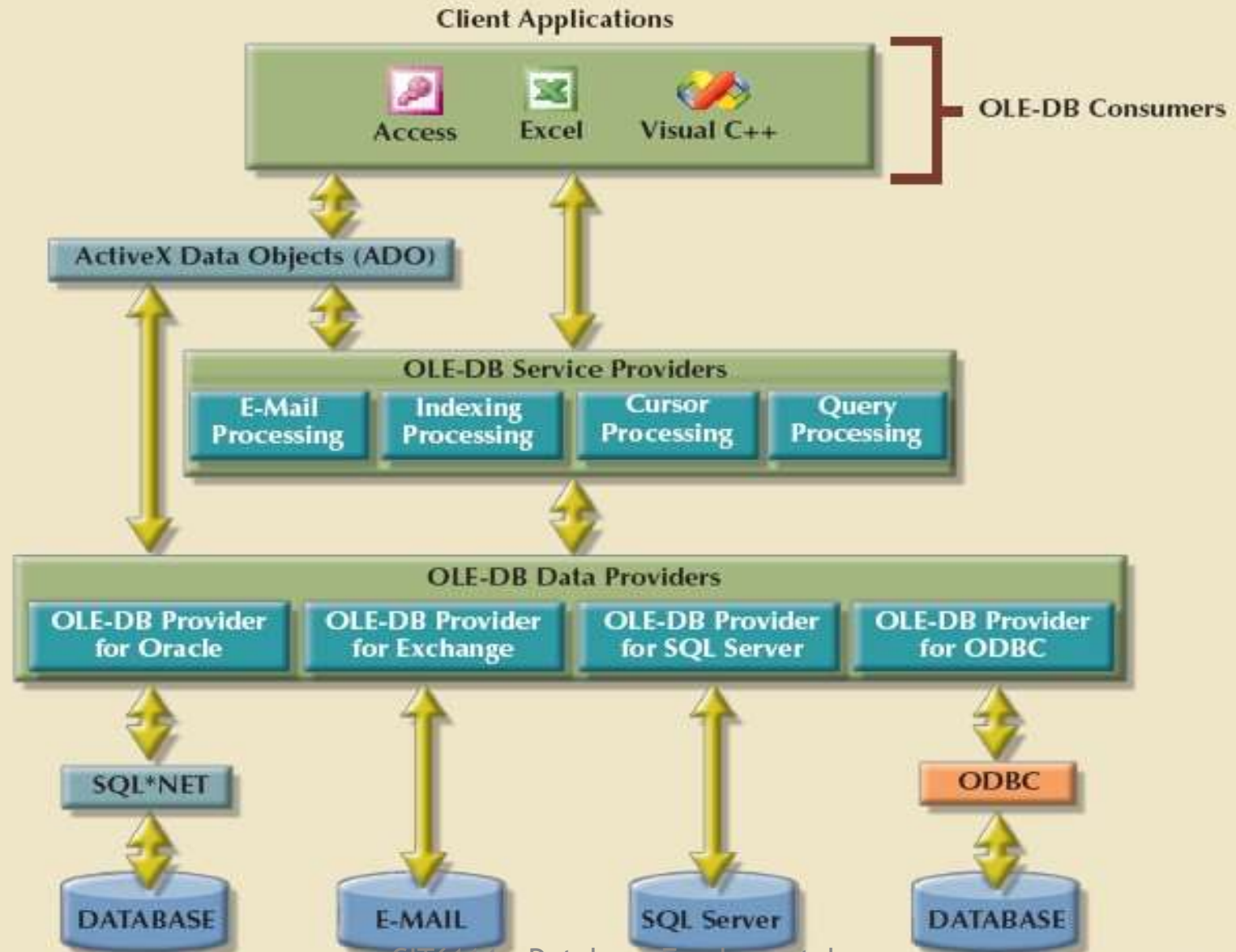
Java Database Connectivity (JDBC)

- JDBC – is an API for JAVA programming language

- It provides methods to query and update data in a database and is oriented towards relational databases

- A JDBC-ODBC bridge enables connections to any ODBC-accessible data source

Java Database Connectivity (JDBC)

FIGURE 11.7 — JDBC architecture
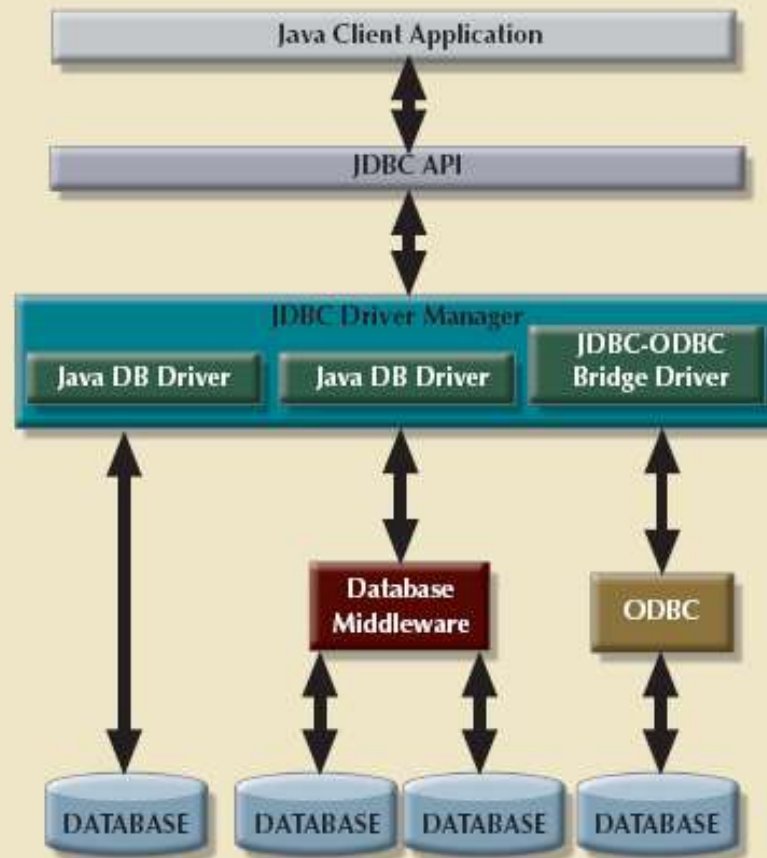
SOURCE: Course Technology/Cengage Learning

Interacting with DB using JDBC

- Loading driver classes
  - **sun.jdbc.odbc.JdbcOdbcDriver** → for JDBC-ODBC bridge
  - **oracle.thin.Driver** → for Oracle
  - **jdbc:z1MySQL** → for MySQL
  - **com.ibm.db2.jcc.DB2Driver** → for IBM DB2

- Database connection URL

  - jdbc:<subprotocol>:<subname>

    - jdbc:db2:myDB

    - jdbc:odbc:mydatabase

Database Internet Connectivity

- Web database connectivity allows new innovative services that:

    - Permit rapid response by **bringing new services** and **products** to market quickly

    - **Increase customer satisfaction** through creation of Web-based support services

    - Allow **anywhere, anytime data access** using mobile smart devices via the Internet

    - Yield **fast** and **effective information dissemination** through universal access

TABLE 11.3 Characteristics and Benefits of Internet Technologies

INTERNET CHARACTERISTIC	BENEFIT
Hardware and software independence	Savings in equipment and software acquisition Ability to run on most existing equipment Platform independence and portability No need for multiple platform development
Common and simple user interface	Reduced training time and cost Reduced end-user support cost No need for multiple platform development
Location independence	Global access through Internet infrastructure and mobile smart devices Reduced requirements (and costs!) for dedicated connections
Rapid development at manageable costs	Availability of multiple development tools Plug-and-play development tools (open standards) More interactive development Reduced development times Relatively inexpensive tools Free client access tools (Web browsers) Low entry costs; frequent availability of free Web servers Reduced costs of maintaining private networks Distributed processing and scalability using multiple servers

CIT6114 – Database Fundamentals

Web-to-Database Middleware: Server-Side Extensions

- **Web server** is the main hub through which Internet services are accessed

- Server-side extension: a program that interacts directly with the Web server

  - Also known as **Web-to-database middleware**

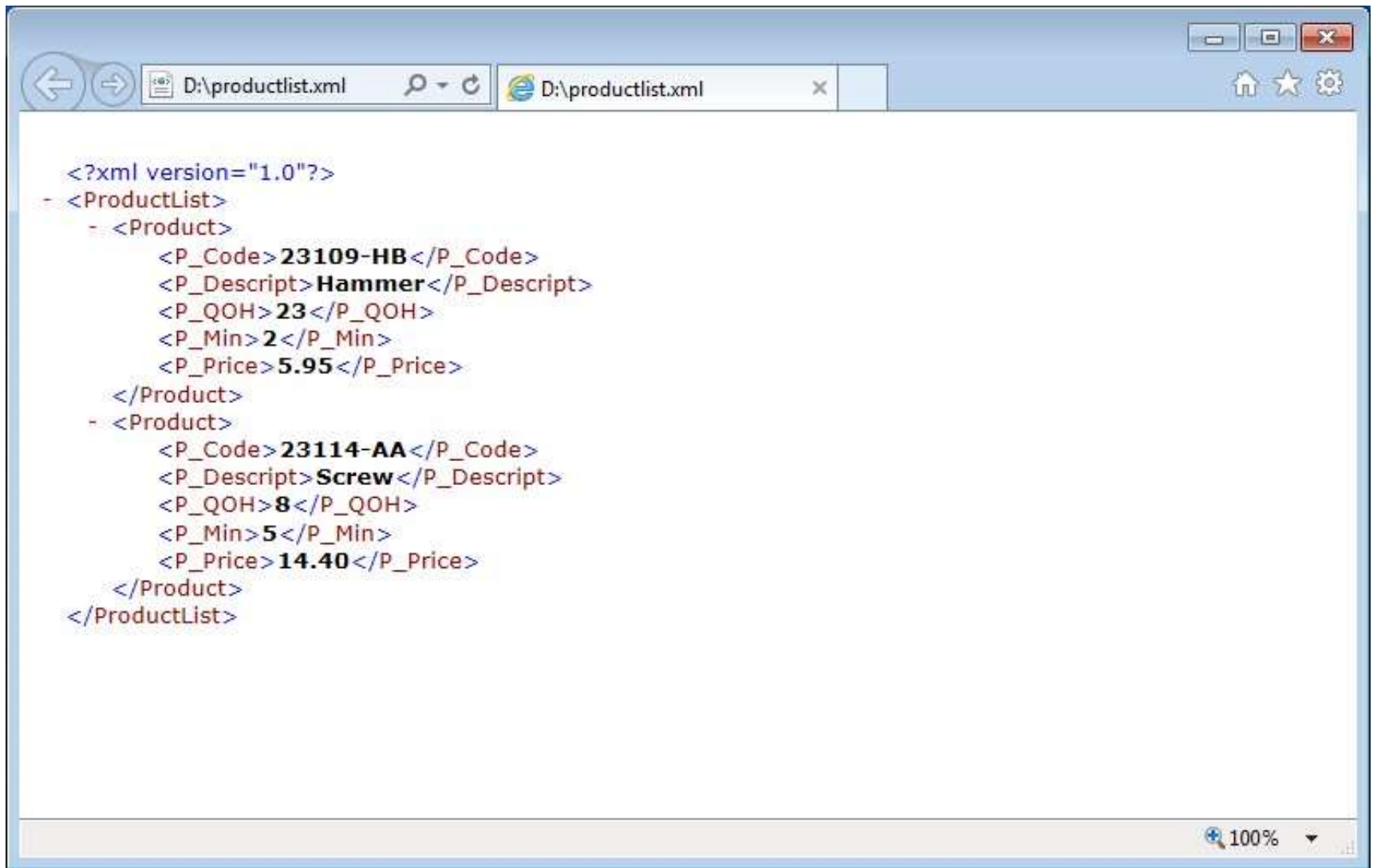- Middleware must be well integrated

FIGURE 11.8 Web-to-database middleware

CLIENT COMPUTER
① HTTP page request

TCP/IP NETWORK

② Web server receives request

WEB SERVER

SERVER COMPUTER

SCRIPT PAGE

③ Web server determines the page contains script language and passes the script page to the Web-to-database middleware

WEB-TO-DATABASE MIDDLEWARE ④

④ Web-to-database middleware connects to the database and passes query using database connectivity layer

HTML PAGE ⑧

The result of the database query is displayed in HTML format

Web server sends the HTML formatted page to the client

HTML PAGE

⑥

⑦ Web-to-database middleware passes the query results in HTML format back to the Web server

JDBC
ADO.NET
ADO
OLE-DB
ODBC

RDBMS Computer

Database server passes the query results back to the Web-to-database middleware

RDBMS SERVER ⑤

DATABASE

32 CIT6114 - Database Fundamentals

SOURCE: Course Technology/Cengage Learning

Extensible Markup Language (XML)

- Metalanguage to represent and manipulate data elements

- Facilitates exchange of structured documents (e.g., invoices, orders, etc.) over the Web

- Characteristics:
  - Allows definition of new tags to describe the data elements (e.g., <ProdPrice>)
  - Case sensitive (<ProductID> is not the same as <productID>)
  - Must be well-formed and properly nested
  - Comments indicated with < - and ->
  - XML and xml prefixes reserved for XML tags only

```xml
<?xml version="1.0"?>
<ProductList>
    <Product>
        <P_Code>23109-HB</P_Code>
        <P_Descript>Hammer</P_Descript>
        <P_QOH>23</P_QOH>
        <P_Min>2</P_Min>
        <P_Price>5.95</P_Price>
    </Product>
    <Product>
        <P_Code>23114-AA</P_Code>
        <P_Descript>Screw</P_Descript>
        <P_QOH>8</P_QOH>
        <P_Min>5</P_Min>
        <P_Price>14.40</P_Price>
    </Product>
</ProductList>
```

CIT6114 – Database Fundamentals

XML Data to Relational Data


```
<?xml version="1.0"?>
- <ProductList>
  - <Product>
      <P_Code>23109-HB</P_Code>
      <P_Descript>Hammer</P_Descript>
      <P_QOH>23</P_QOH>
      <P_Min>2</P_Min>
      <P_Price>5.95</P_Price>
    </Product>
  - <Product>
      <P_Code>23114-AA</P_Code>
      <P_Descript>Screw</P_Descript>
      <P_QOH>8</P_QOH>
      <P_Min>5</P_Min>
      <P_Price>14.40</P_Price>
    </Product>
</ProductList>
```

P-Code	P_Descript	P_QOH	P_Min	P_Price
23109-HB	Hammer	23	2	5.95
23114-AA	Screw	8	5	14.40

CIT6114 – Database Fundamentals

Cloud Computing Services

- *"A computing model for enabling **ubiquitous**, convenient, on-demand network access to a shared pool of configurable computer resources that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

- Potential to become a "game changer"

- Windows Azure, Google, Amazon

Cloud Computing Services

- It enables access to a shared pool of configurable computer resources that can be:
    - Rapidly provisioned
    - Released with minimal management effort or service provider interaction

- It eliminates financial and technological barriers

Cloud Services

Cengage Learning © 2015
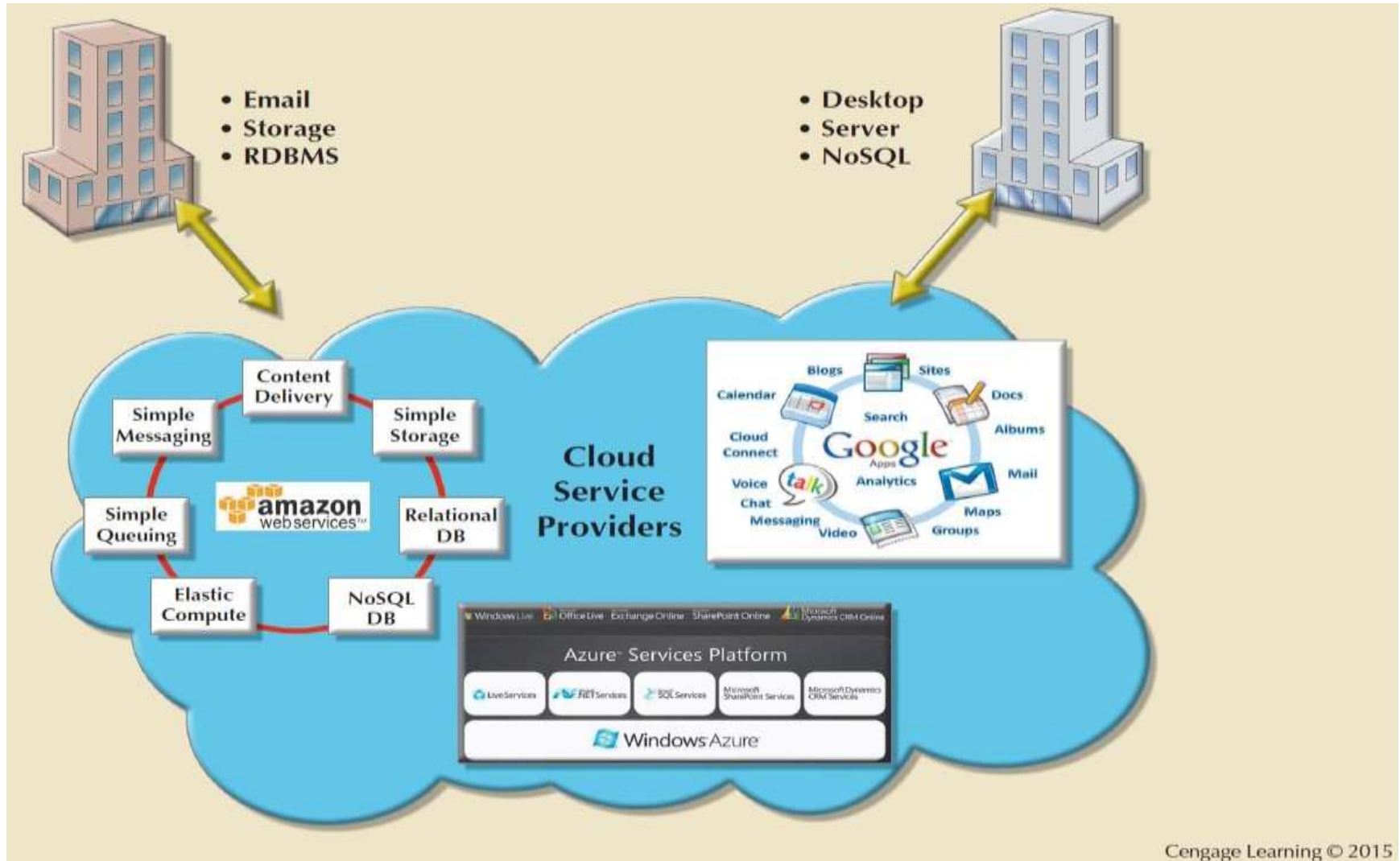
Cloud Services

- Managed by a cloud services provider

- **No hassle on own resources** as the service provider supplies the hardware and software

- E.g., online data storage and backup, web-based email services, hosted office suites, document collaboration services, etc.

Cloud Implementation Types

- **Public Cloud**
  - Built by a third-party organization to sell cloud services to the general public

- **Private Cloud**
  - Built by an organization for the sole purpose of servicing its own needs

- **Community Cloud**
  - Built by and for a specific group of organizations that share a common trade

Public Cloud

- Open for public
- Maybe free or offered on as pay-per-usage model
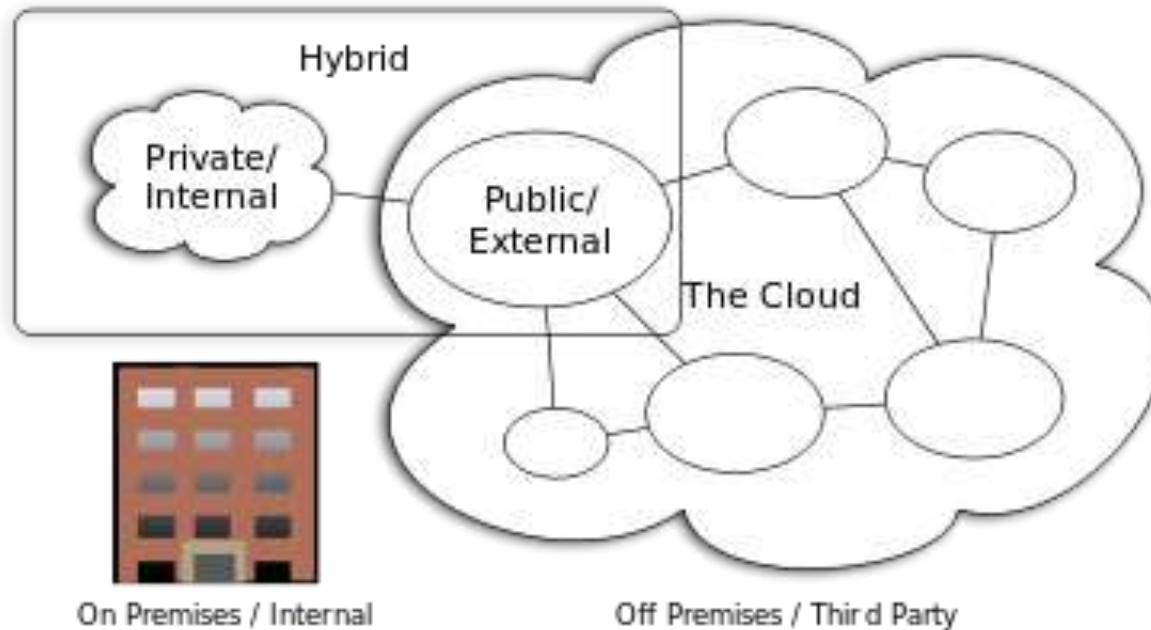
Cloud Computing Types

[Source from wikipedia]

Private Cloud

- Solely for a single organization, managed internally or by third-party
- Self-run data centers
- Need to buy, build and manage own resources
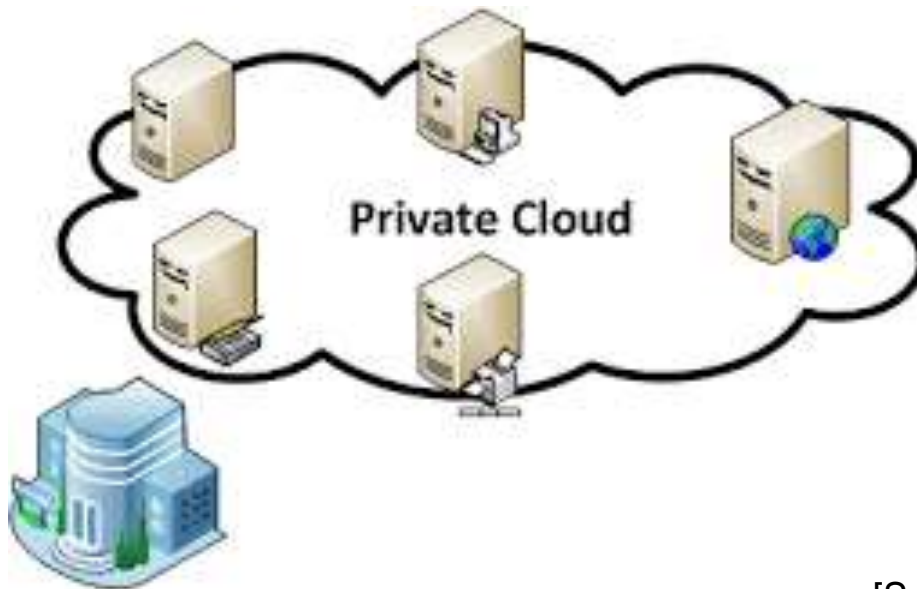
[Source from falconITservices.com]

CIT6114 – Database Fundamentals

Community Cloud

- Also known as **hybrid cloud**
- Composition of two or more clouds
- Runs on a private cloud, 'burst' to a public cloud when the demand for computing capability increases (e.g., storage/memory)

Community Cloud Model

Organization 1

Organization 2
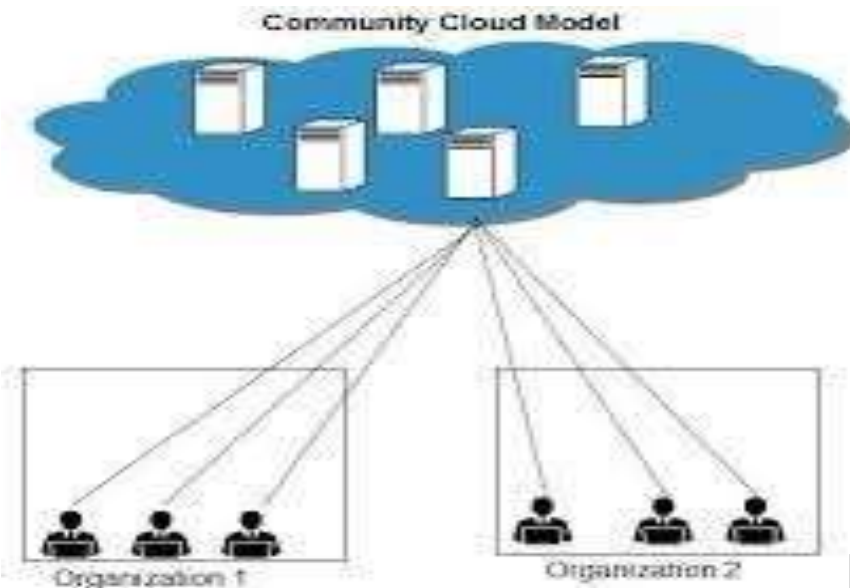
[Source from bsnlldce.ngntraining.com]

Characteristics of Cloud Services

- Ubiquitous access via Internet technologies
- Shared infrastructure
- Lower costs and variable pricing
- Flexible and scalable services
- Dynamic provisioning
- Service orientation
- Managed operations

Types of Cloud Services

Servers

Laptops

Desktops

Tablets

Smartphones

Internet

Software as a Service
- MS Office Live, MS Exchange Online
- Google Docs, Google Email
- Salesforce CRM Online
- SAP Business ByDesign

Platform as a Service
- Amazon Web Services, Amazon Relational Data Service, Amazon Simple DB
- MS Azure Platform, MS SQL Service
- Google Application Engine

Infrastructure as a Service
- Amazon Web Services Elastic Computing Cloud 2 (EC2)
- Amazon Elastic MapReduce Service
- Amazon Simple Storage Service (S3)
- Amazon Elastic Load Balancing Service

Cengage Learning © 2015

CIT6114 – Database Fundamentals

Software as a Service (SaaS)

- Also known as 'On-demand software'

- Based on a **pay-per-use** basis (subscription fee)

- Cloud users access the software from cloud clients

  - Eliminates the needs to install and run the application on own computers

  - Simplified maintenance and support

Platform as a Service (PaaS)

- Delivers a **computing platform**, including operating system, programming language execution environment, database, and web server

- Underlying computer and storage resources scale automatically to match application demand

Infrastructure as a Service (IaaS)

- Provides access to **computing resources** in a virtualized environment, usually the Internet

  - Such virtual server space, network connections, bandwidths, IP addresses, load balancers, etc.

- The client is given access to the virtualized components in order to build their own IT platforms

Database as a Service (DaaS)

- Cloud computing data management service

- Provides relational data management to companies of any size

- Avoids high cost of personnel/maintenance

- Leverages Internet to provide:
  - Hosted data management
  - Standard protocols
  - A common programming interface

- Assist businesses with limited information technology resources

FIGURE 11.22

Provisioning an RDBMS in the cloud

Provisioning a MySQL RDBMS instance in Amazon Web Services (AWS)

Windows Azure Pricing Calculator

Provisioning a MySQL Azure RDBMS instance in Microsoft Azure

CIT6114 – Database Fundamentals

Cloud Services: Advantages and Disadvantages

ADVANTAGE	DISADVANTAGE
Low initial cost of entry. Cloud computing has lower costs of entry when compared with the alternative of building in house.	*Issues of security, privacy, and compliance.* Trusting sensitive company data to external entities is difficult for most data-cautious organizations.
Scalability/elasticity. It is easy to add and remove resources on demand.	*Hidden costs of implementation and operation.* It is hard to estimate bandwidth and data migration costs.
Support for mobile computing. Cloud computing providers support multiple types of mobile computing devices.	*Data migration is a difficult and lengthy process.* Migrating large amounts of data to and from the cloud infrastructure can be difficult and time-consuming.
Ubiquitous access. Consumers can access the cloud resources from anywhere at any time, as long as they have Internet access.	*Complex licensing schemes.* Organizations that implement cloud services are faced with complex licensing schemes and complicated service-level agreements.
High reliability and performance. Cloud providers build solid infrastructures that otherwise are difficult for the average organization to leverage.	*Loss of ownership and control.* Companies that use cloud services are no longer in complete control of their data. What is the responsibility of the cloud provider if data are breached? Can the vendor use your data without your consent?
Fast provisioning. Resources can be provisioned on demand in a matter of minutes with minimal effort.	*Organization culture.* End users tend to be resistant to change. Do the savings justify being dependent on a single provider? Will the cloud provider be around in 10 years?
Managed infrastructure. Most cloud implementations are managed by dedicated internal or external staff. This allows the organization's IT staff to focus on other areas.	*Difficult integration with internal IT system.* Configuring the cloud services to integrate transparently with internal authentication and other internal services could be a daunting task.

TABLE 11.4 Advantages and Disadvantages of Cloud Computing