IS-456: Database Management Systems

Module 10: Client/Server & Internet database / Blockchain Databases

Summer 2022

CityU

of Seattle

Topics

- Database Connectivity
- ODBC, DAO, and RDO
- Java Database Connectivity
- Database Internet Connectivity
- Web Server Interfaces
- Client-Side Extensions
- Web Application Servers
- Extensible Markup Language (XML)
- Cloud Computing Services
- Blockchain Database Technology
- Summary
- References

Database Connectivity

- *Database connectivity* refers to the mechanisms through which application programs connect and communicate with data repositories.
- Database management system (DBMS) functions as an intermediary between the data (stored in the database) and the end-user's applications.
- Database connectivity software works in a client/server architecture, by which processing tasks are split among multiple software layers. In this model, the multiple layers exchange control messages and data.
- **Basic functionality** of database connectivity software:

1. A data layer where the data resides. You could think of this layer as the actual data repository interface. This layer resides closest to the database itself and normally is provided by the DBMS vendor.
2. Middle layer that manages multiple connectivity and data transformation issues. This layer is in charge of dealing with data logic issues, data transformations, ways to "talk" to the database below it, and so on. This would also include translating multiple data manipulation languages to the native language supported by the specific data repository.
3. A top layer that interfaces with the actual external application. This mostly comes in the form of an application programming interface that publishes specific protocols for the external programs to interact with the data.

Database Connectivity Continue…

- Database connectivity software is also known as database middleware—because it provides an interface between the application program and the database or data repository.
- The data repository, also known as the *data source*, represents the data management application, such as Oracle, SQL Server, IBM DB2, or NoSQL that will be used to store the data generated by the application program.
- There are many ways to achieve database connectivity such as:
- Native SQL connectivity (vendor provided)
- Microsoft's Open Database Connectivity (ODBC), Data Access Objects (DAO), and
- Remote Data Objects (RDO)
- Microsoft's Object Linking and Embedding for Database (OLE-DB)
- Microsoft's ActiveX Data Objects (ADO.NET)
- Oracle's Java Database Connectivity (JDBC)

ODBC, DAO, and RDO

- Open Database Connectivity (ODBC) is Microsoft's implementation of a superset of the SQL Access Group Call Level Interface (CLI) standard for database access.
- ODBC allows any Windows application to access relational data sources, using SQL via

a standard application programming interface (API).

- The Webopedia online dictionary (*www.webopedia.com*) defines an API as "a set of routines, protocols, and tools for building software applications."
- A good API makes it easy to develop a program by providing all of the building blocks; the programmer puts the blocks together.
- Most operating environments, such as Windows, provide an API so that programmers can write applications consistent with the operating environment.
- Although APIs are designed for programmers, they are ultimately good for users because they guarantee that all programs using a common API will have similar interfaces.
- ODBC was the first widely adopted database middleware standard, and it enjoyed rapid adoption in Windows applications.

ODBC, DAO, and RDO Cont.

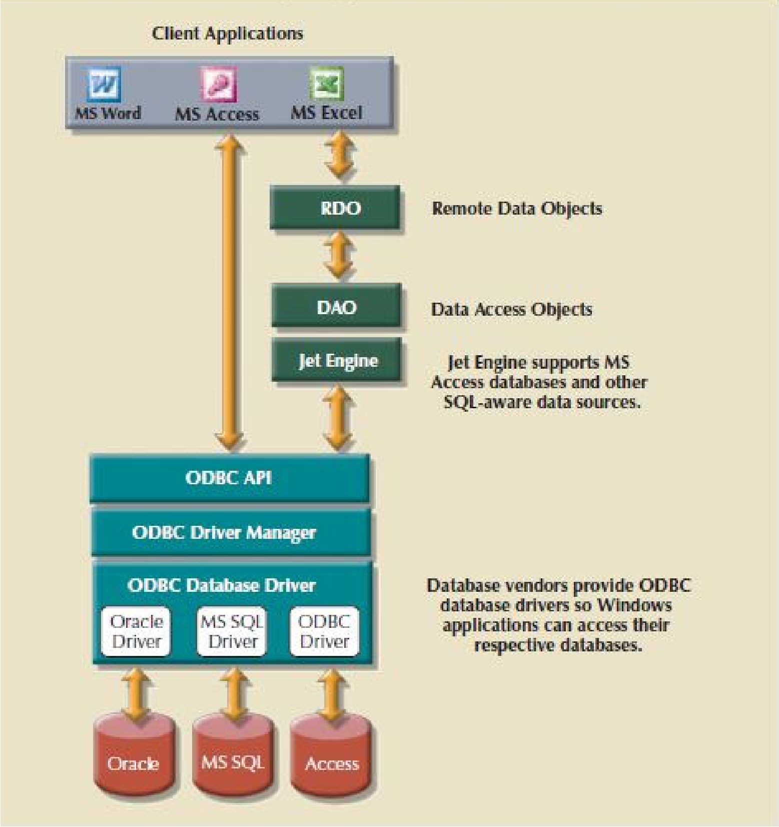
- ODBC did not provide significant functionality beyond the ability to execute SQL to manipulate relational-style data.
- Microsoft developed two other data access interfaces in order to answer the need:
- **Data Access Objects (DAO):** is an object-oriented API used to access desktop databases, such as MS Access and FileMaker Pro. DAO provides an optimized interface that exposes programmers to the functionality of the Jet data engine.
- **Remote Data Objects (RDO):** is a higher-level, object-oriented application interface

used to access remote database servers. RDO uses the lower-level DAO and ODBC for direct access to databases. RDO is optimized to deal with server-based databases such as MS SQL Server, Oracle, and DB2.

- ODBC, DAO, and RDO are implemented as shared code that is dynamically linked to the Windows operating environment through dynamic-link libraries (DLLs), which are stored as files with a .dll extension.

- Figure 15.2 illustrates how Windows applications can use ODBC, DAO, and RDO to access local and remote relational data sources.

FIGURE 15.2 USING ODBC, DAO, AND RDO TO ACCESS DATABASES



The basic **ODBC architecture** has three main components:

❑ A high-level *ODBC API* through which application programs access ODBC functionality

❑ A *driver manager* that is in charge of managing all database connections

❑ An *ODBC driver* that communicates directly to the DBMS

Java Database Connectivity

- Java is an object-oriented programming language developed by Sun Microsystems

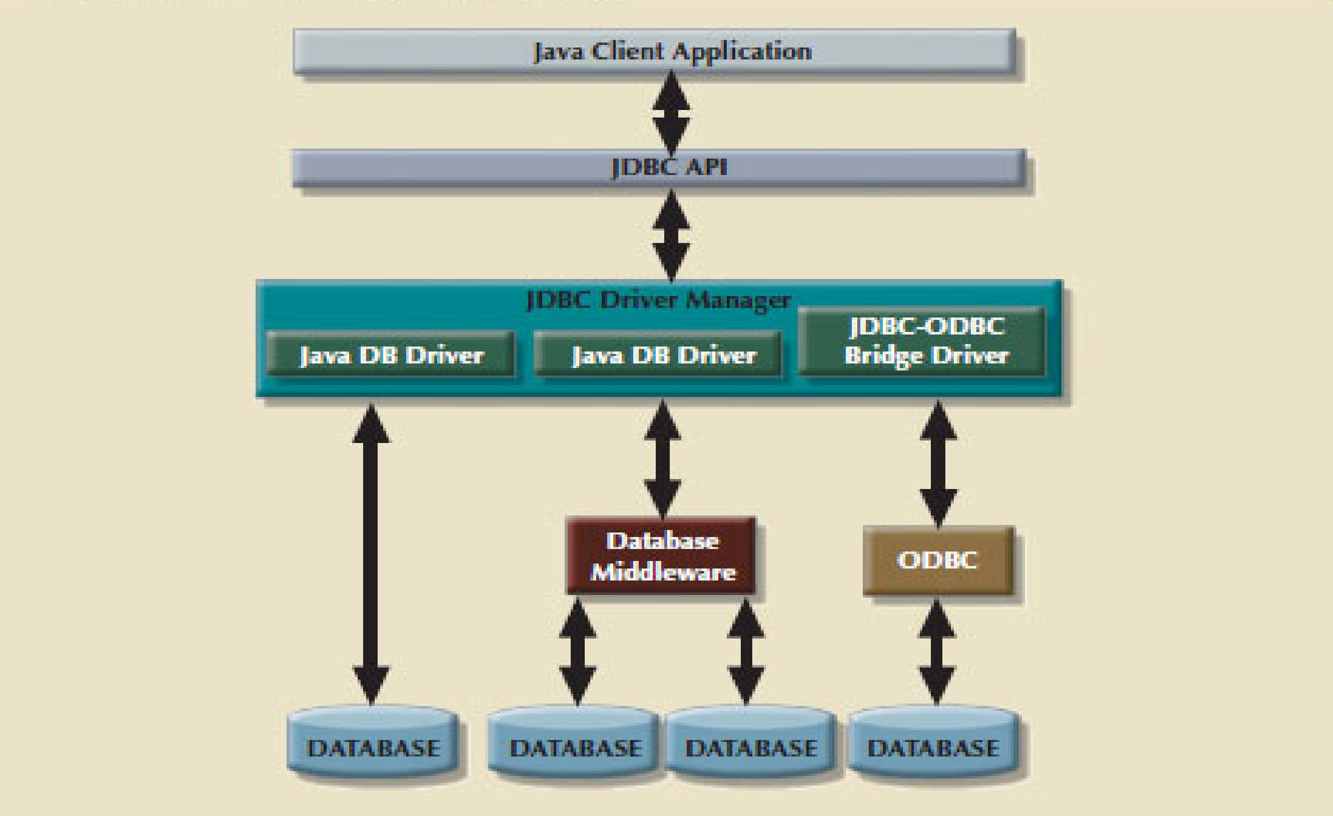
(acquired by Oracle in 2010) that runs on top of web browser software.

- Java is one of the most common programming languages for web development.
- Sun Microsystems created Java as a “write once, run anywhere” environment, which means that a programmer can write a Java application once and then run it in multiple environments without any modification.
- When Java applications need to access data outside the Java runtime environment, they

use predefined application programming interfaces.

- ❑ Java Database Connectivity (JDBC) is an application programming interface that allows a Java program to interact with a wide range of data sources, including relational databases, tabular data sources, spreadsheets, and text files.
- ❑ JDBC allows a Java program to establish a connection with a data source, prepare and send the SQL code to the database server, and process the result set.
- ❑ JDBC allows programmers to use their SQL skills to manipulate the data in the company’s databases.
- ❑ JDBC allows direct access to a database server or access via database middleware.

FIGURE 15.7 JDBC ARCHITECTURE



Database Internet Connectivity

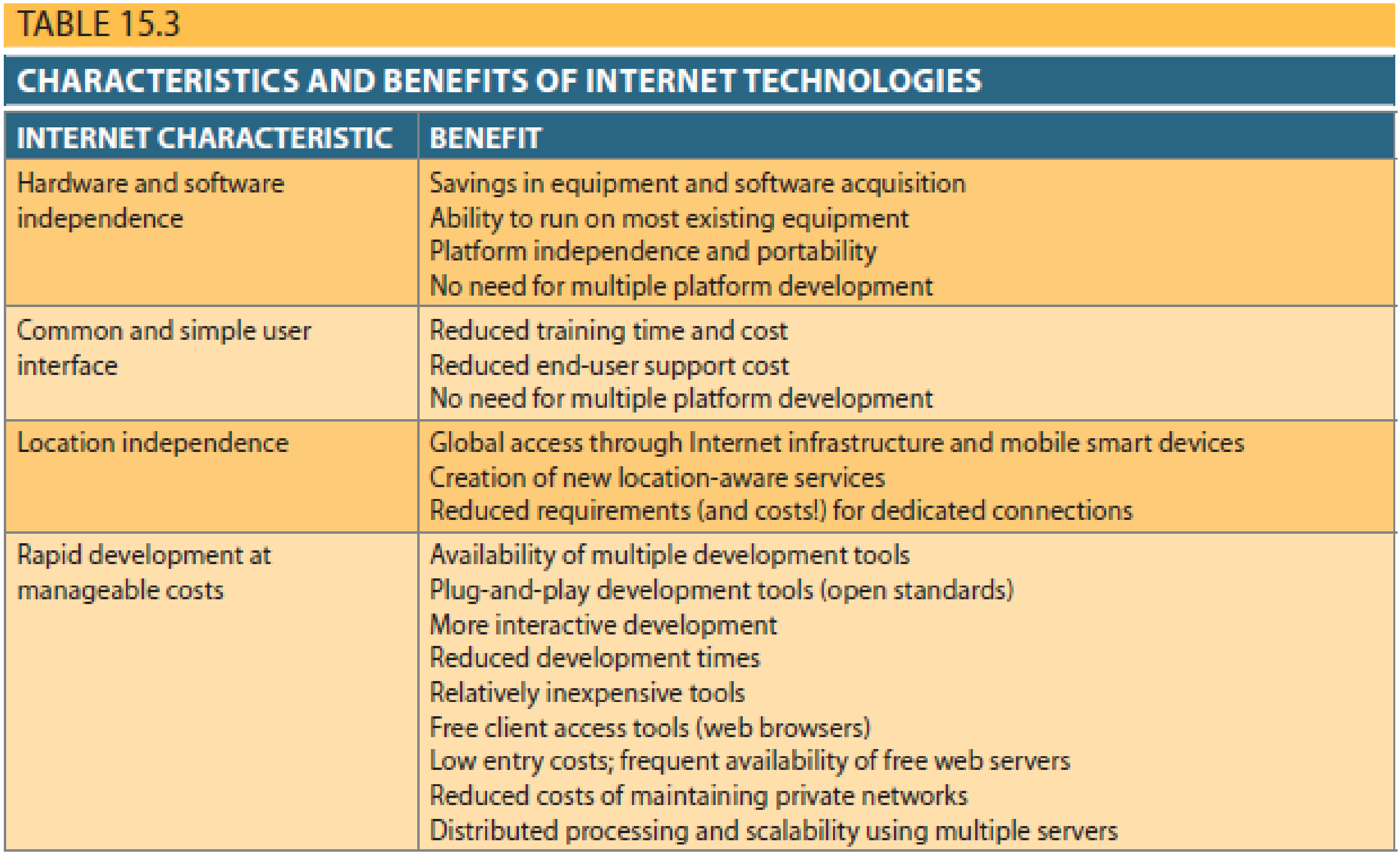


TABLE 15.3

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 Creation of new location-aware services 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

Web server Interfaces

- A web server interface defines a standard way to exchange messages with external programs.
- There are two well-defined web server interfaces:
- Common Gateway Interface (CGI)
- Application programming interface (API)
- The Common Gateway Interface (CGI) uses script files that perform specific functions

based on the client’s parameters that are passed to the web server.

- The main disadvantage of using CGI scripts is that the script file is an external program

that executes separately for each user request and therefore causes a resource bottleneck.

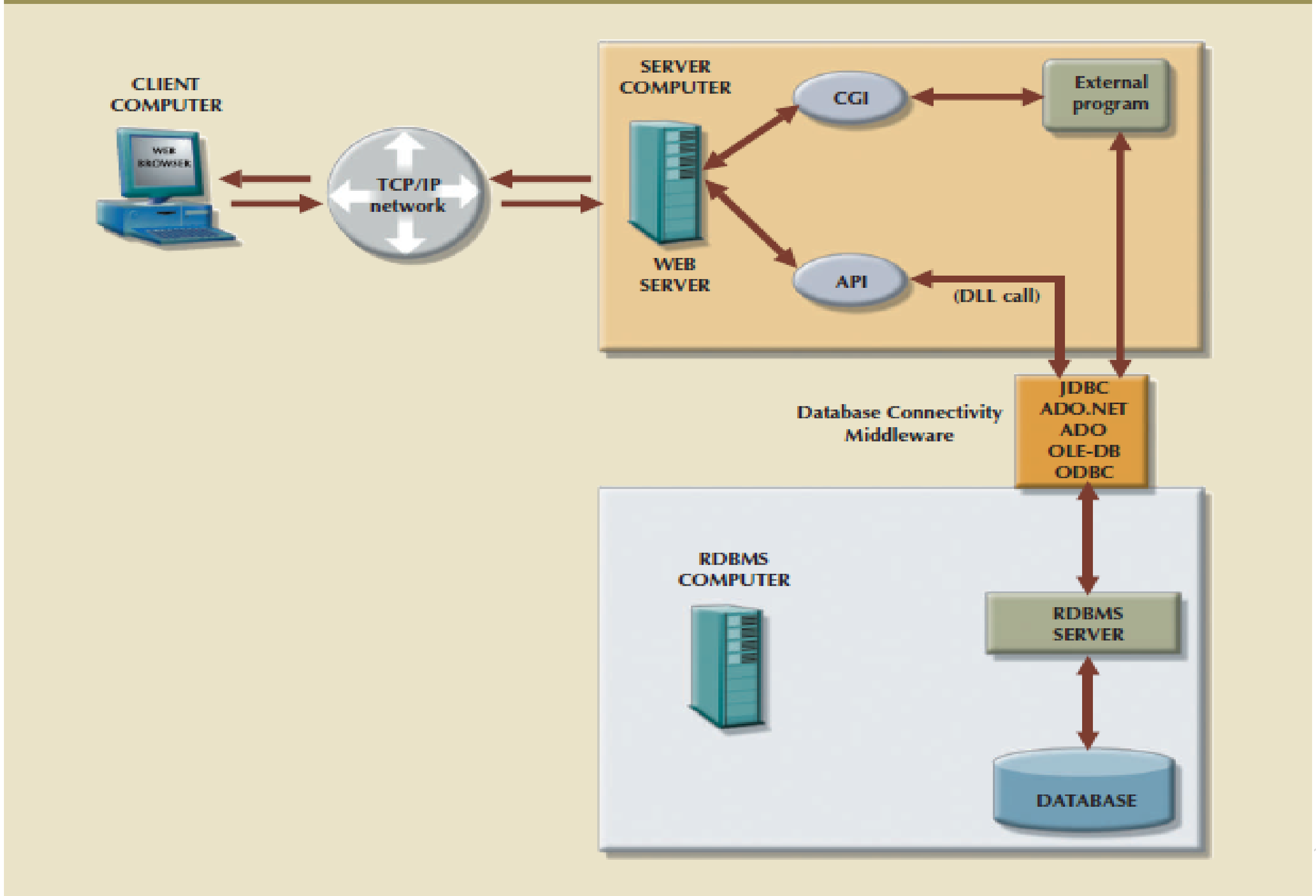
Performance also could be degraded by using an interpreted language or by writing the script inefficiently.

- An application programming interface (API) is a newer web server interface standard

that is more efficient and faster than a CGI script. APIs are more efficient because they are implemented as shared code or as dynamic-link libraries (DLLs).

- APIs are faster than CGI scripts because the code resides in memory, so there is no need to run an external program for each request. Instead, the same API serves all requests.
- an API can use a shared connection to the database instead of creating a new one every time, as is the case with CGI scripts.

FIGURE 15.9 WEB SERVER CGI AND API INTERFACES



Client-Side Extensions

- Client-side extensions add functionality to the web browser. Although client-side

extensions are available in various forms, the most common are:

- Plug-ins
- Java and JavaScript
- ActiveX and VBScript

- A plug-in is an external application that is automatically invoked by the browser when

needed.

- JavaScript is a scripting language (one that enables the execution of a series of commands or macros) that allows web authors to design interactive sites.
- ActiveX is Microsoft's alternative to Java. ActiveX is a specification for writing programs

that run inside the Microsoft client browser, Internet Explorer.

- **ActiveX** extends the web browser by adding controls to webpages, including drop-down lists, a slider, a calendar, and a calculator.
- VBScript is another Microsoft product that is used to extend browser functionality. VBScript is derived from Microsoft Visual Basic. Like JavaScript, VBScript code is embedded inside an HTML page and is activated by triggering events such as clicking a link.

Web Application Servers

- A web application server is a middleware application that expands the functionality of web servers by linking them to a wide range of services, such as databases, directory systems, and search engines.
- The web application server also provides a consistent runtime environment for web applications.
- Web application servers can be used to perform the following:
- Connect to and query a database from a webpage.
- Present database data in a webpage using various formats.
- Create dynamic web search pages.
- Create webpages to insert, update, and delete database data.
- Enforce referential integrity in the application program logic.
- Performance and fault-tolerant features
- Database access with transaction management capabilities
- Access to multiple services, such as file transfers (FTP), database connectivity, email, and directory services

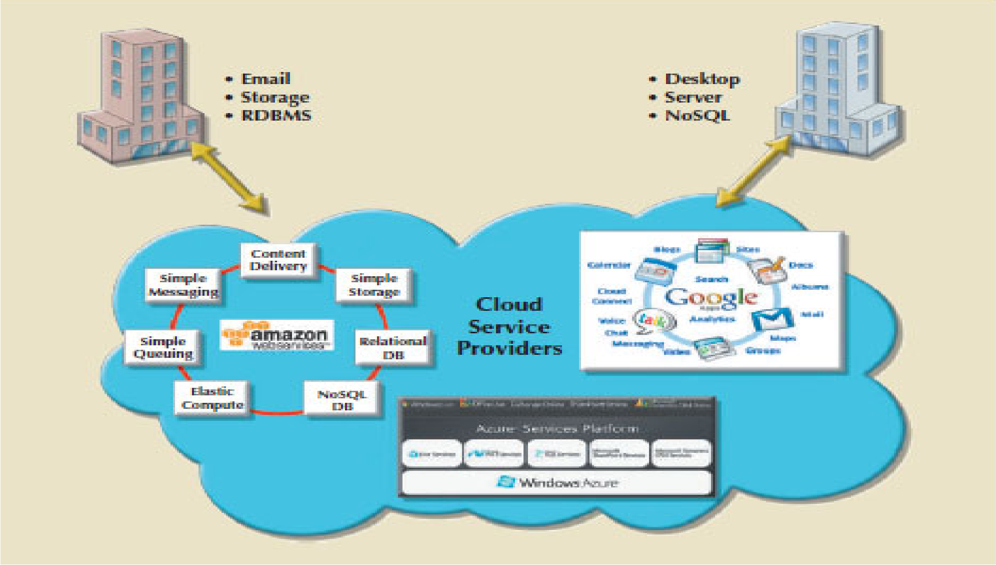
Extensible Markup Language (XML)

- Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
- Extensible Markup Language (XML) is a meta-language used to represent and manipulate data elements.
- **XML** has a few important additional characteristics:
- XML allows the definition of new tags to describe data elements.
- XML is case sensitive: <ProductID>is not the same as <Productid>.
- XML must be well formed; that is, tags must be properly formatted. Most openings also have a corresponding closing. For example, a product's identification would require the format <ProductId>2345-AA</ProductId>.
- XML must be properly nested. For example, properly nested XML might look like this: <Product><ProductId>2345-AA</ProductId></Product>.
- You can use the <-- and --> symbols to enter comments in the XML document.
- The *XML* and *xml* prefixes are reserved for XML only.
- XML is *not* a new version or replacement for HTML. XML is concerned with the description and representation of the data, rather than the way the data is displayed.

Cloud Computing Services

- According to the National Institute of Standards and Technology (NIST), cloud computing is “a computing model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computer resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”
- The term cloud services is used to refer to the services provided by cloud computing.

FIGURE 15.21 CLOUD SERVICES



more and more organizations are tapping into cloud services to secure advanced database services (relational or NoSQL)
for their organizations.

Cloud Implementation Types

- Cloud computing has different types of implementations based on who the target customers are:

- ❑ Public cloud: This type of cloud infrastructure is built by a third-party organization

to sell cloud services to the general public. The public cloud is the most common type of cloud implementation.

- ❑ Private cloud: This type of internal cloud is built by an organization for the sole purpose of servicing its own needs. Private clouds are often used by large, geographically

dispersed organizations to add agility and flexibility to internal IT services.

- ❑ Community cloud: This type of cloud is built by and for a specific group of organizations that share a common trade, such as agencies of the federal government, the military, or higher education.

- Regardless of the implementation an organization uses, most cloud services share a

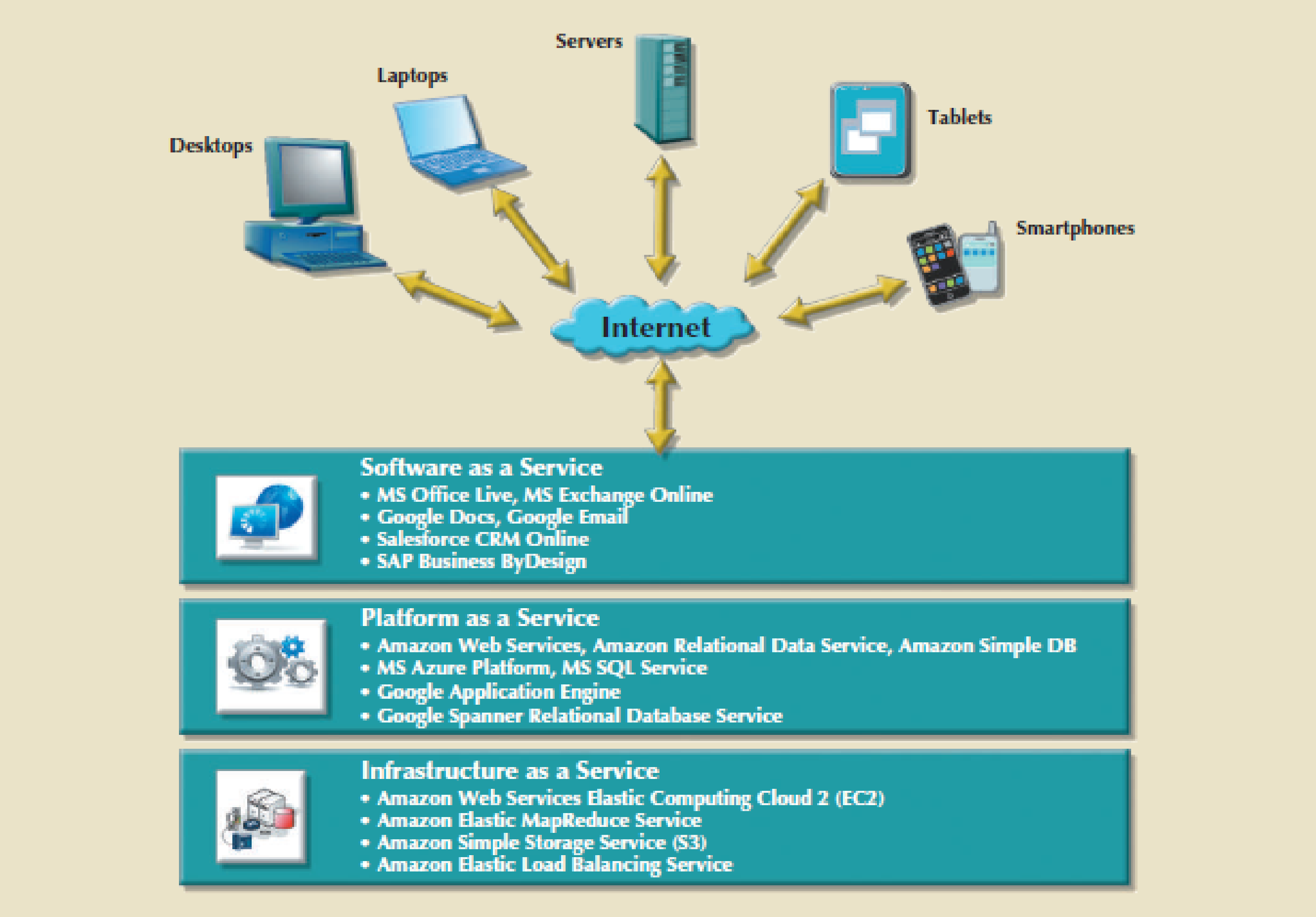
common set of core characteristics.

Type of Cloud Services

- Cloud services come in different shapes and forms; no single type of service works for all consumers.
- Cloud services can be classified by the following categories:

- ❑ Software as a Service (SaaS): The cloud service provider offers turnkey applications that run in the cloud
- ❑ Platform as a Service (PaaS). The cloud service provider offers the capability to build and deploy consumer-created applications using the provider's cloud infrastructure.
- ❑ Infrastructure as a Service (IaaS). In this case, the cloud service provider offers consumers the ability to provision their own resources on demand

FIGURE 15.23 TYPES OF CLOUD SERVICES



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
- Google Spanner Relational Database Service

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

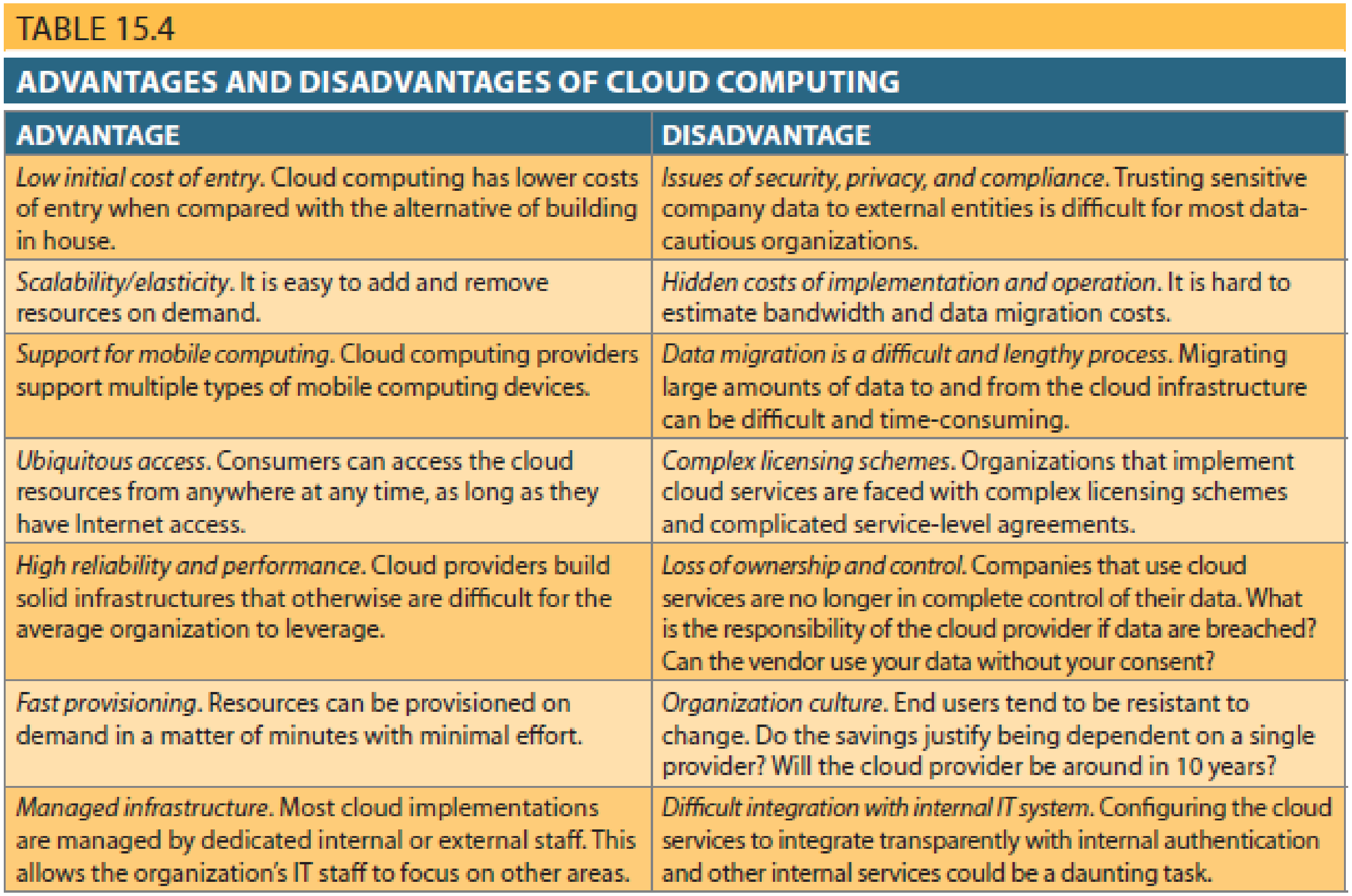


TABLE 15.4

ADVANTAGES AND DISADVANTAGES OF CLOUD COMPUTING

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.

Blockchain database Technology

- If you have been following banking, investing, or cryptocurrency over the last ten years, you may be familiar with “blockchain,” the record-keeping technology behind the Bitcoin network.
- Blockchain is literally just a chain of blocks; **digital information** (the “block”) stored in a **public database** (the “chain”).
- ~~D eorfnfkdlq vwruhv lqirupdwlrq lq eorfnv wkdw duh xqlirup lq vl}h1 Hdfk eorfn frqwdlqv wkh kdvkhg lqirupdwlrq ru kdvk frgh iurp wkh suhylrxv eorfn wr surylgh fu|swrjudsklf vhfxulw|1~~
- ~~Xqolnh gdwdedvhv/ wklv dgghg vhfxulw| ihdwxuh hqjurvvhg zlwk lq eorfnfkdlqv pdnhv wkhp h{wuhpho| gliilfxow wr kdfn dqg wdpshu1~~

  ~~Wkh pdlq gliihuhqfhv ehwzhhq eorfnfkdlqv dqg gdwdedvhv duh olvwhg lq wkh wdeoh~~ ehorz

~~Database~~	~~Blockchain~~
~~Centralized~~	~~Decentralized~~
~~Permissioned~~	~~Permission-less~~
~~Requires administrator~~	~~No administrator~~

Blockchain vs. Bitcoin

- The goal of blockchain is to allow digital information to be recorded and distributed, but not edited.
- Blockchain technology was first outlined in 1991 by Stuart Haber and W. Scott Stornetta, two researchers who wanted to implement a system where document timestamps could not be tampered with.
- The Bitcoin protocol is built on the blockchain.
- Bitcoin's pseudonymous creator Satoshi Nakamoto referred to it as "a new electronic cash system that's fully peer-to-peer, with no trusted third party."
- Transactions made in bitcoin are verified by a network of computers. This is what is meant by the Bitcoin network and blockchain being "decentralized."
- Mining
- Wallet
- Cryptographic keys
- Blockchain technology is considered confidential

Summary

- **Database Connectivity**
- **ODBC, DAO, and RDO**
- **Java Database Connectivity**
- **Database Internet Connectivity**
- **Web Server Interfaces**
- **Client-Side Extensions**
- **Web Application Servers**
- **Extensible Markup Language (XML)**
- **Cloud Computing Services**
- **Blockchain Database Technology**

Resources

- Mukesh Negi, (2019): Fundamental of Database Management System
- Coronel C., Morris S., (2019): Database Systems, Design, Implementation, & Management
- Adham Saeed, (2017): Role of Database Management Systems in Supporting Information Technology