ITIS 313
Data and Information Management

Fall 2021

Distributed Database Management Systems

Learning Objectives

- About distributed database management systems (DDBMSs) and their components
- How database implementation is affected by different levels of data and process distribution
- How transactions are managed in a distributed database environment

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Distributed Database Management Systems

Learning Objectives

- How distributed database design draws on data partitioning and replication to balance performance, scalability, and availability
- About the trade-offs of implementing a distributed data system

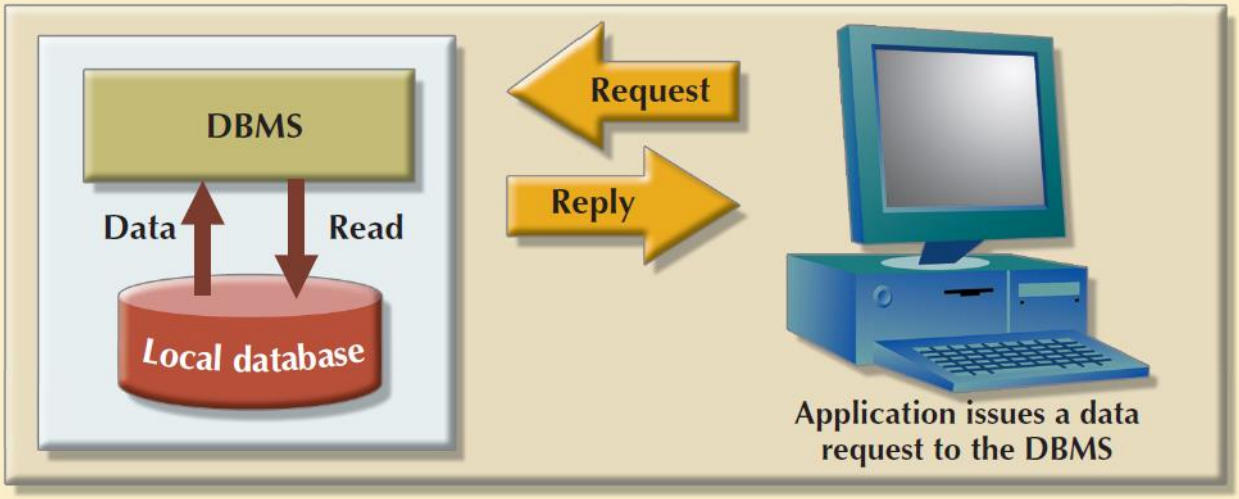
©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Evolution Database Management Systems

- **Distributed database management system (DDBMS)**: Governs storage and processing of logically related data over interconnected computer systems
  - Data and processing functions are distributed among several sites
- Centralized database management system
  - Required that corporate data be stored in a single central site
  - Data access provided through dumb terminals

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Figure 12.1 - Centralized Database Management System



Cengage Learning © 2015

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Factors Affecting the Centralized Database Systems

- Globalization of business operation
- Advancement of web-based services
- Rapid growth of social and network technologies
- Digitization resulting in multiple types of data
- Innovative business intelligence through analysis of data

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Factors That Aided DDBMS to Cope With Technological Advancement

- Acceptance of Internet as a platform for business
- Mobile wireless revolution
- Usage of application as a service
- Focus on mobile business intelligence

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Advantages and Disadvantages of DDBMS

Advantages	Disadvantages
• Data are located near greatest demand site • Faster data access and processing • Growth facilitation • Improved communications • Reduced operating costs • User-friendly interface • Less danger of a single-point failure • Processor independence	• Complexity of management and control • Technological difficulty • Security • Lack of standards • Increased storage and infrastructure requirements • Increased training cost • Costs incurred due to the requirement of duplicated infrastructure

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Distributed Processing and Distributed Databases

- **Distributed processing**: Database's logical processing is shared among two or more physically independent sites via network
- **Distributed database**: Stores logically related database over two or more physically independent sites via computer network
  - **Database fragments**: Database composed of many parts in distributed database system

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Characteristics of Distributed Management Systems

Application interface	Validation	Transformation	Query optimization
Mapping	I/O interface	Formatting	Security
Backup and recovery	DB administration	Concurrency control	Transaction management

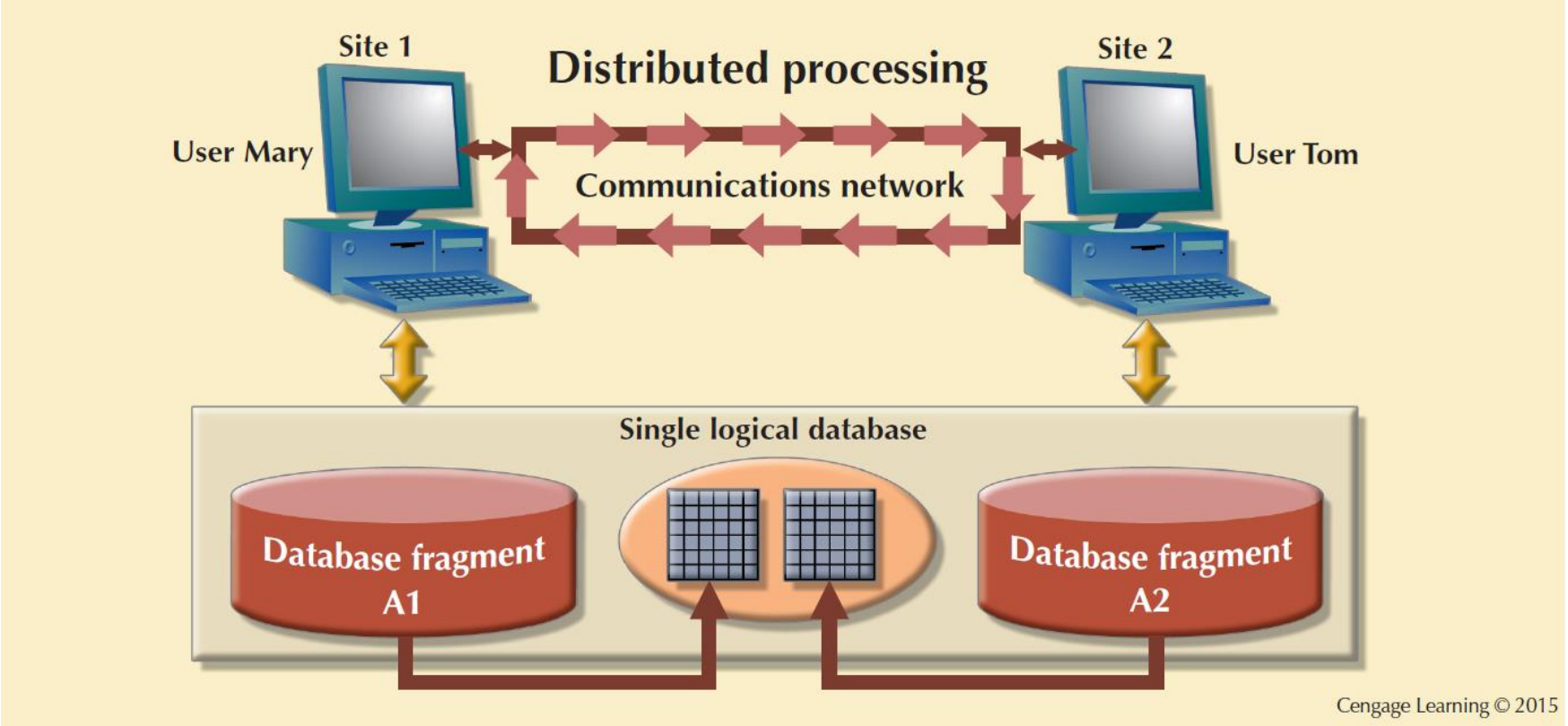
©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Functions of Distributed DBMS

- Receives the request of an application
- Validates analyzes, and decomposes the request
- Maps the request
- Decomposes request into several I/O operations
- Searches and validates data
- Ensures consistency, security, and integrity
- Validates data for specific conditions
- Presents data in required format

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Figure 12.4 - A Fully Distributed Database Management System



©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

DDBMS Components

- Computer workstations or remote devices
- Network hardware and software components
- Communications media

- **Transaction processor (TP):** Software component of a system that requests data
  - Known as **transaction manager (TM)** or **application processor (AP)**
  - **Data processor (DP)** or **data manager (DM)**
    - Software component on a system that stores and retrieves data from its location

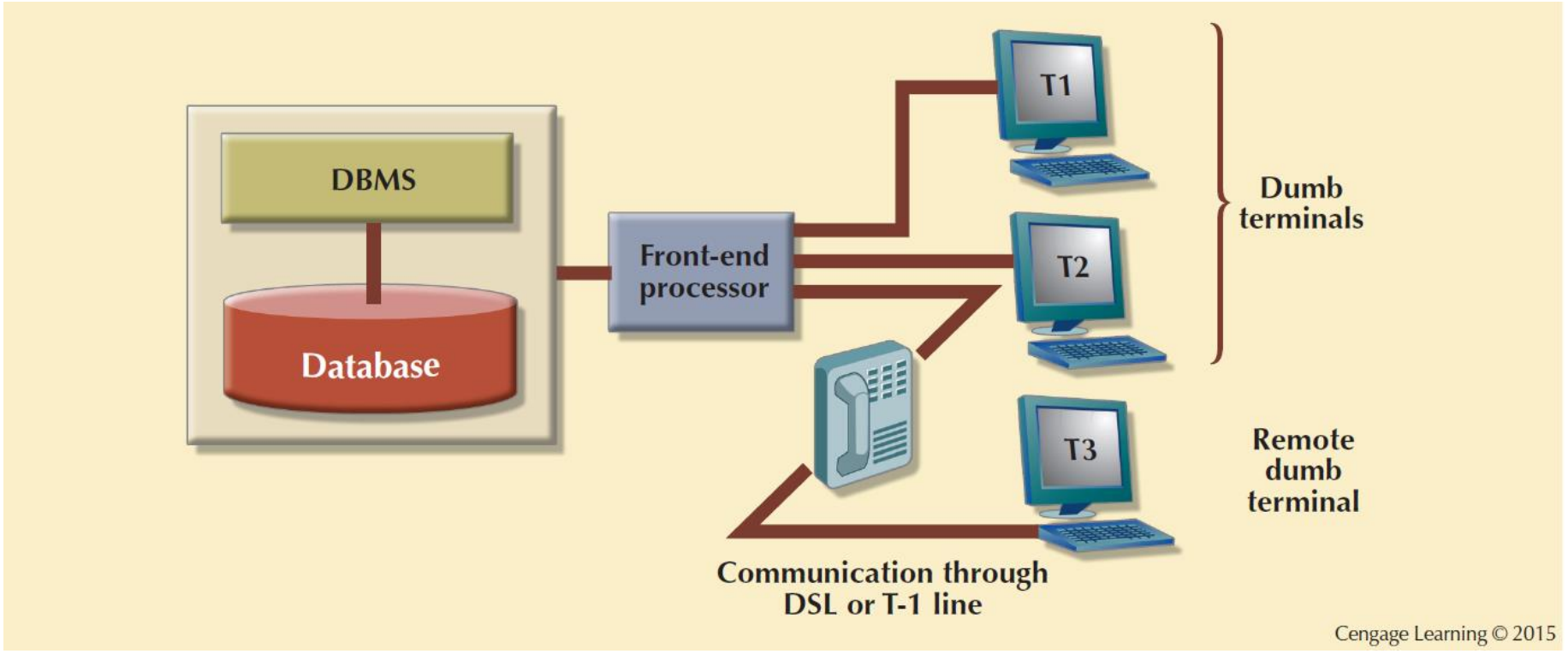
©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Single-Site Processing, Single-Site Data (SPSD)

- Processing is done on a single host computer
- Data stored on host computer's local disk
- Processing restricted on end user's side
- DBMS is accessed by dumb terminals

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Figure 12.6 - Single-Site Processing, Single-Site Data (Centralized)



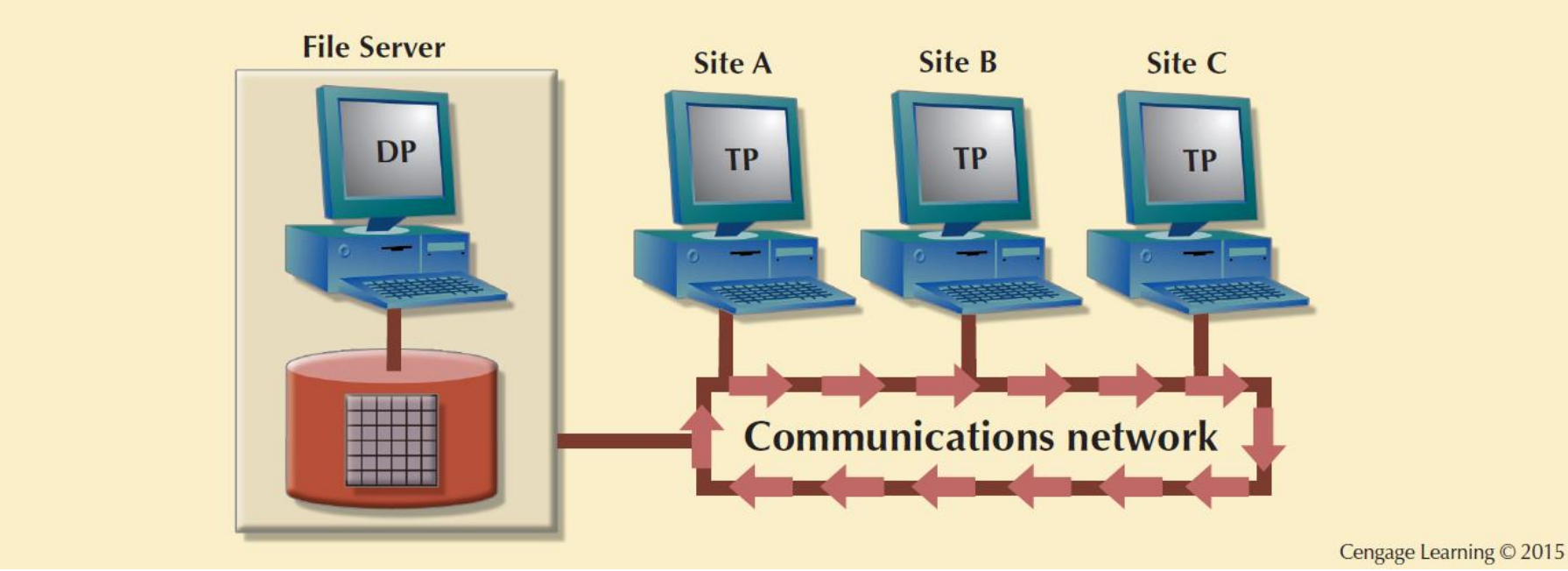
©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Multiple-Site Processing, Single-Site Data (MPSD)

- Multiple processes run on different computers sharing a single data repository
- Require network file server running conventional applications
  - Accessed through LAN
- **Client/server architecture**
  - Reduces network traffic
  - Processing is distributed
  - Supports data at multiple sites

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Figure 12.7 - Multiple-Site Processing, Single-Site Data



©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Multiple-Site Processing, Multiple-Site Data (MPMD)

- Fully distributed database management system
  - Support multiple data processors and transaction processors at multiple sites
- Classification of DDBMS depending on the level of support for various types of databases
  - **Homogeneous**: Integrate multiple instances of same DBMS over a network
  - **Heterogeneous**: Integrate different types of DBMSs
  - **Fully heterogeneous**: Support different DBMSs, each supporting different data model

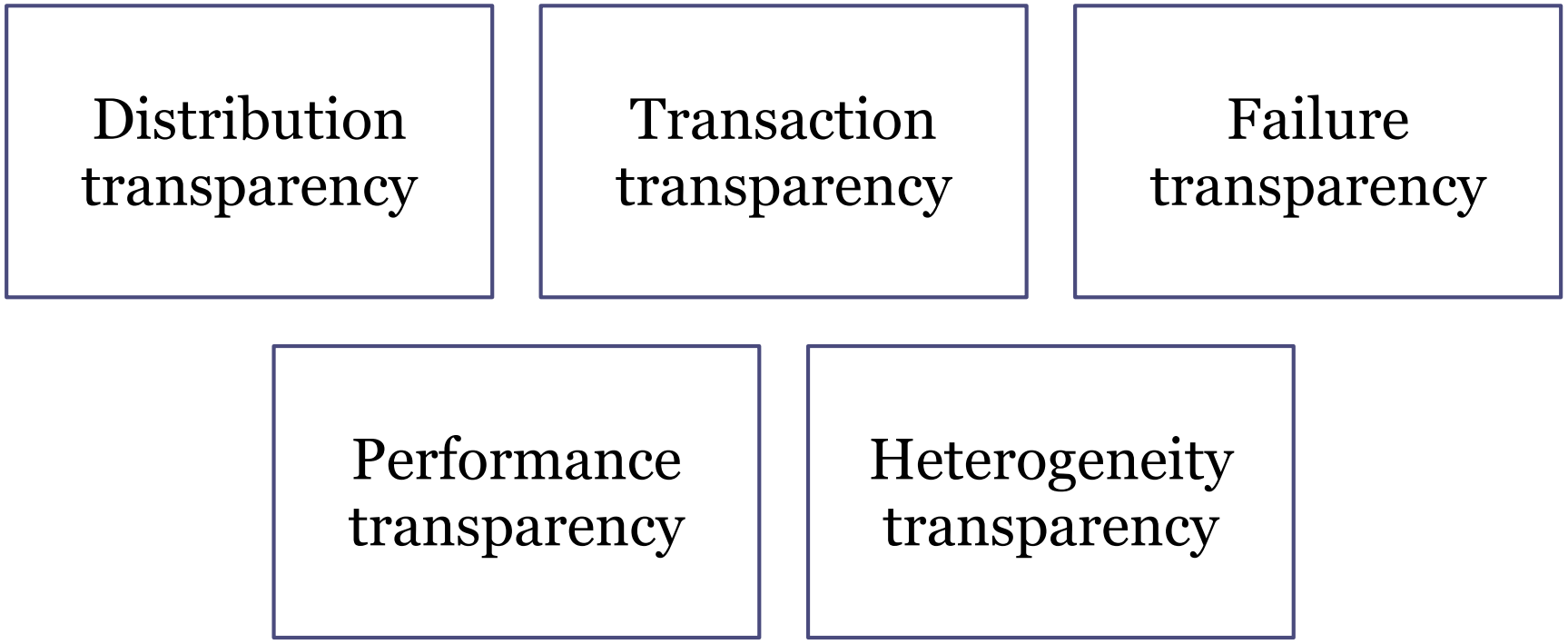
©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Restrictions of DDBMS

- Remote access is provided on a read-only basis
- Restrictions on the number of remote tables that may be accessed in a single transaction
- Restrictions on the number of distinct databases that may be accessed
- Restrictions on the database model that may be accessed

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Distributed Database Transparency Features



©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Distribution Transparency

- Allows management of physically dispersed database as if centralized
- Levels
  - **Fragmentation transparency**
  - **Location transparency**
  - **Local mapping transparency**

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Distribution Transparency

- **Unique fragment**: Each row is unique, regardless of the fragment in which it is located
- Supported by **distributed data dictionary (DDD)** or **distributed data catalog (DDC)**
  - DDC contains the description of the entire database as seen by the database administrator
- **Distributed global schema**: Common database schema to translate user requests into subqueries

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Transaction Transparency

- Ensures database transactions will maintain distributed database's integrity and consistency
- Ensures transaction completed only when all database sites involved complete their part
- Distributed database systems require complex mechanisms to manage transactions

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Distributed Requests and Distributed Transactions

Remote request

- Single SQL statement accesses data processed by a single remote database processor

Remote transaction

- Accesses data at single remote site composed of several requests

Distributed transaction

- Requests data from several different remote sites on network

Distributed request

- Single SQL statement references data at several DP sites

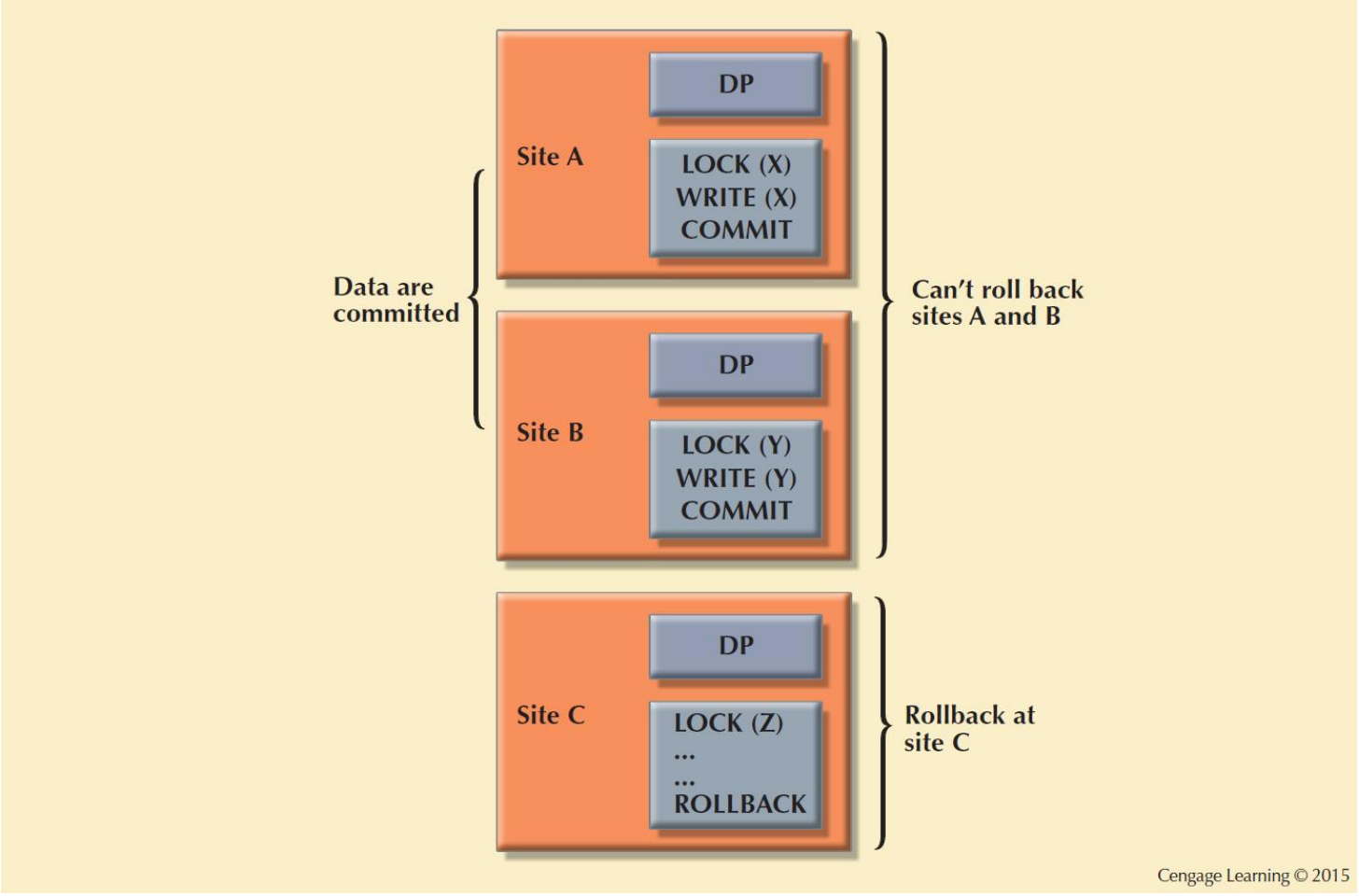
©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Distributed Concurrency Control

- Concurrency control is important in distributed databases environment
  - Due to multi-site multiple-process operations that create inconsistencies and deadlocked transactions

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Figure 12.14 - The Effect of Premature COMMIT



©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Two-Phase Commit Protocol (2PC)

- Guarantees if a portion of a transaction operation cannot be committed, all changes made at the other sites will be undone
  - To maintain a consistent database state
- Requires that each DP's transaction log entry be written before database fragment is updated
- **DO-UNDO-REDO protocol**: Roll transactions back and forward with the help of the system's transaction log entries

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Two-Phase Commit Protocol (2PC)

- **Write-ahead protocol**: Forces the log entry to be written to permanent storage before actual operation takes place
- Defines operations between **coordinator** and **subordinates**
- Phases of implementation
  - Preparation
  - The final COMMIT

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Performance and Failure Transparency

- **Performance transparency**: Allows a DDBMS to perform as if it were a centralized database
- **Failure transparency**: Ensures the system will operate in case of network failure
- Considerations for resolving requests in a distributed data environment
  - Data distribution
  - Data replication
    - **Replica transparency**: DDBMS's ability to hide multiple copies of data from the user

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Performance and Failure Transparency

- Network and node availability
  - **Network latency**: delay imposed by the amount of time required for a data packet to make a round trip
  - **Network partitioning**: delay imposed when nodes become suddenly unavailable due to a network failure

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Distributed Database Design

Data fragmentation

- How to partition database into fragments

Data replication

- Which fragments to replicate

Data allocation

- Where to locate those fragments and replicas

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Data Fragmentation

- Breaks single object into many segments
  - Information is stored in distributed data catalog (DDC)
- Strategies
  - **Horizontal fragmentation**: Division of a relation into subsets (fragments) of tuples (rows)
  - **Vertical fragmentation**: Division of a relation into attribute (column) subsets
  - **Mixed fragmentation**: Combination of horizontal and vertical strategies

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Data Replication

- Data copies stored at multiple sites served by a computer network
- **Mutual consistency rule**: Replicated data fragments should be identical
- Styles of replication
  - Push replication
  - Pull replication
- Helps restore lost data

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Data Replication Scenarios

Fully replicated database

- Stores multiple copies of each database fragment at multiple sites

Partially replicated database

- Stores multiple copies of some database fragments at multiple sites

Unreplicated database

- Stores each database fragment at a single site

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Data Allocation Strategies

Centralized data allocation

- Entire database stored at one site

Partitioned data allocation

- Database is divided into two or more disjoined fragments and stored at two or more sites

Replicated data allocation

- Copies of one or more database fragments are stored at several sites

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

The CAP Theorem

- CAP stands for:
  - Consistency
  - Availability
  - Partition tolerance
- **Basically available, soft state, eventually consistent (BASE)**
  - Data changes are not immediate but propagate slowly through the system until all replicas are consistent

©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

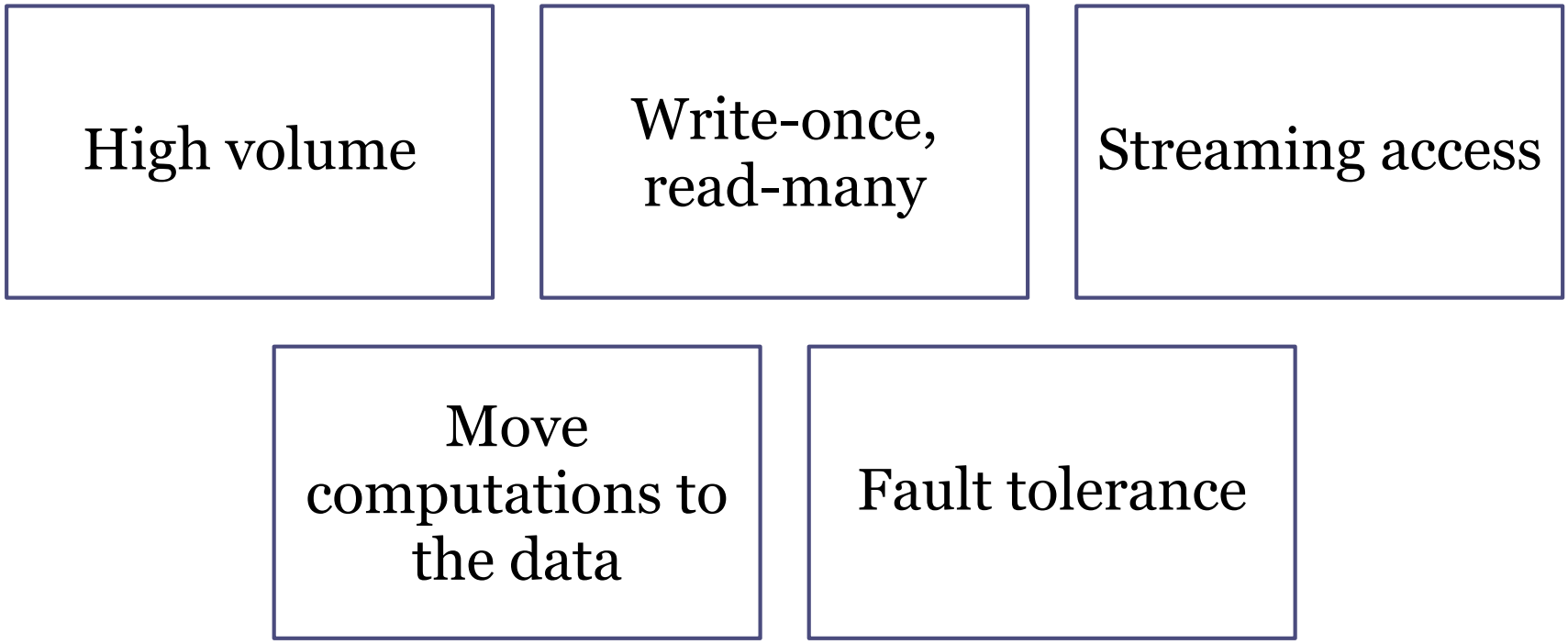
Table 12.8 - Distributed Database Spectrum

DBMS TYPE	CONSISTENCY	AVAILABILITY	PARTITION TOLERANCE	TRANSACTION MODEL	TRADE-OFF
Centralized DBMS	High	High	N/A	ACID	No distributed data processing
Relational DDBMS	High	Relaxed	High	ACID	
(2PC)	Sacrifices availability to ensure con-sistency and isolation				
NoSQL DDBMS	Relaxed	High	High	BASE	Sacrifices consistency to ensure availability

Cengage Learning © 2015

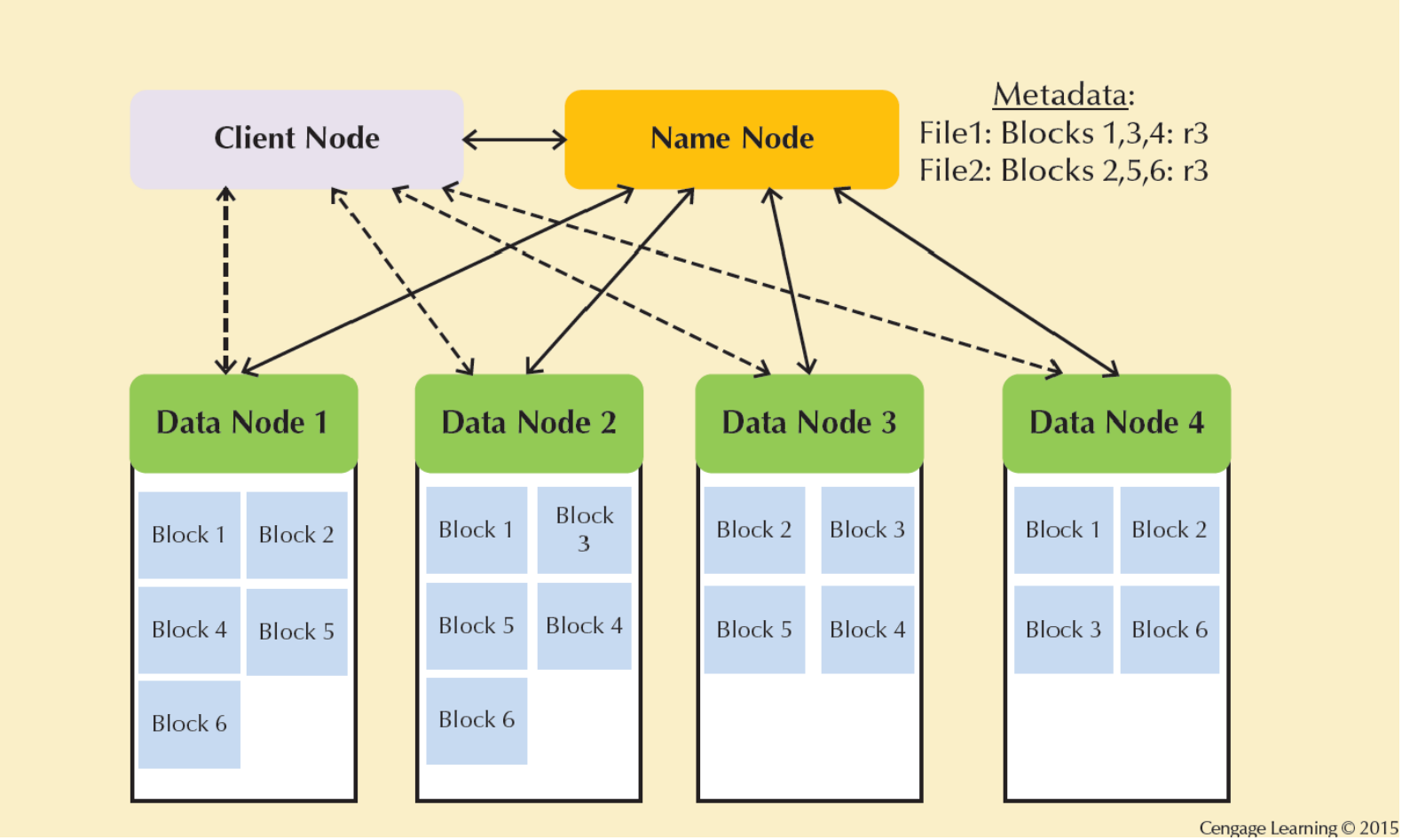
©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Key Assumptions of Hadoop Distributed File System



©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Figure 12.20 - Hadoop Distributed File System (HDFS)



©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.