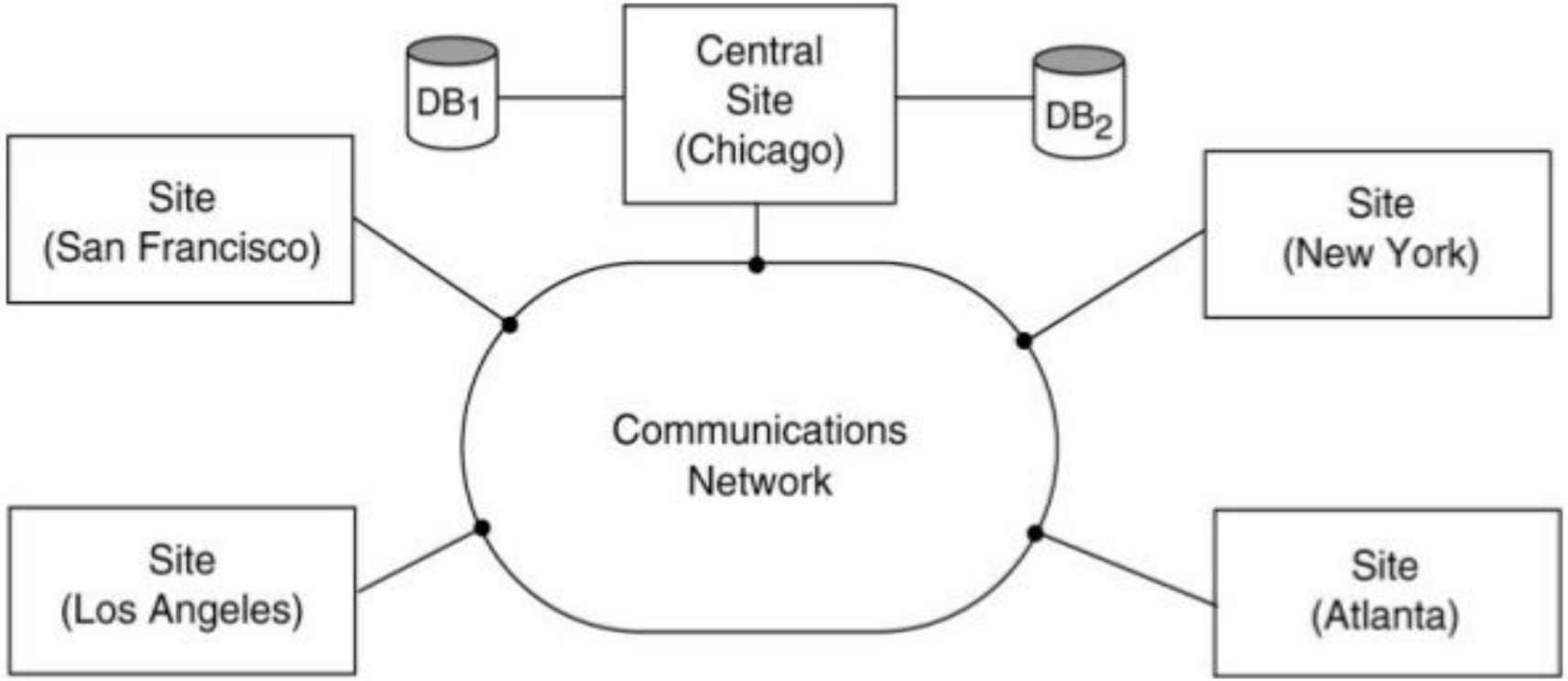
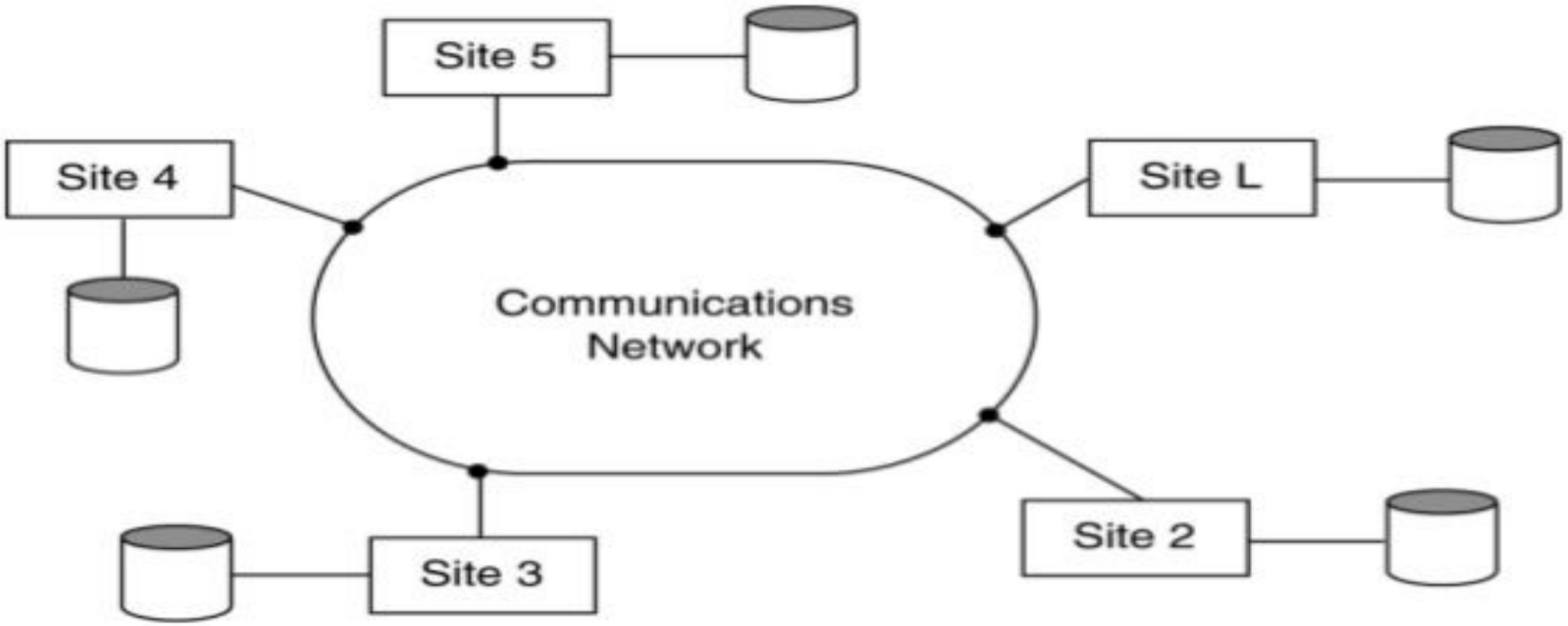
# Distributed Database Management Systems

# Centralized database



# Distributed database




(c) A distributed database on a network

# Distributed Database Management Systems

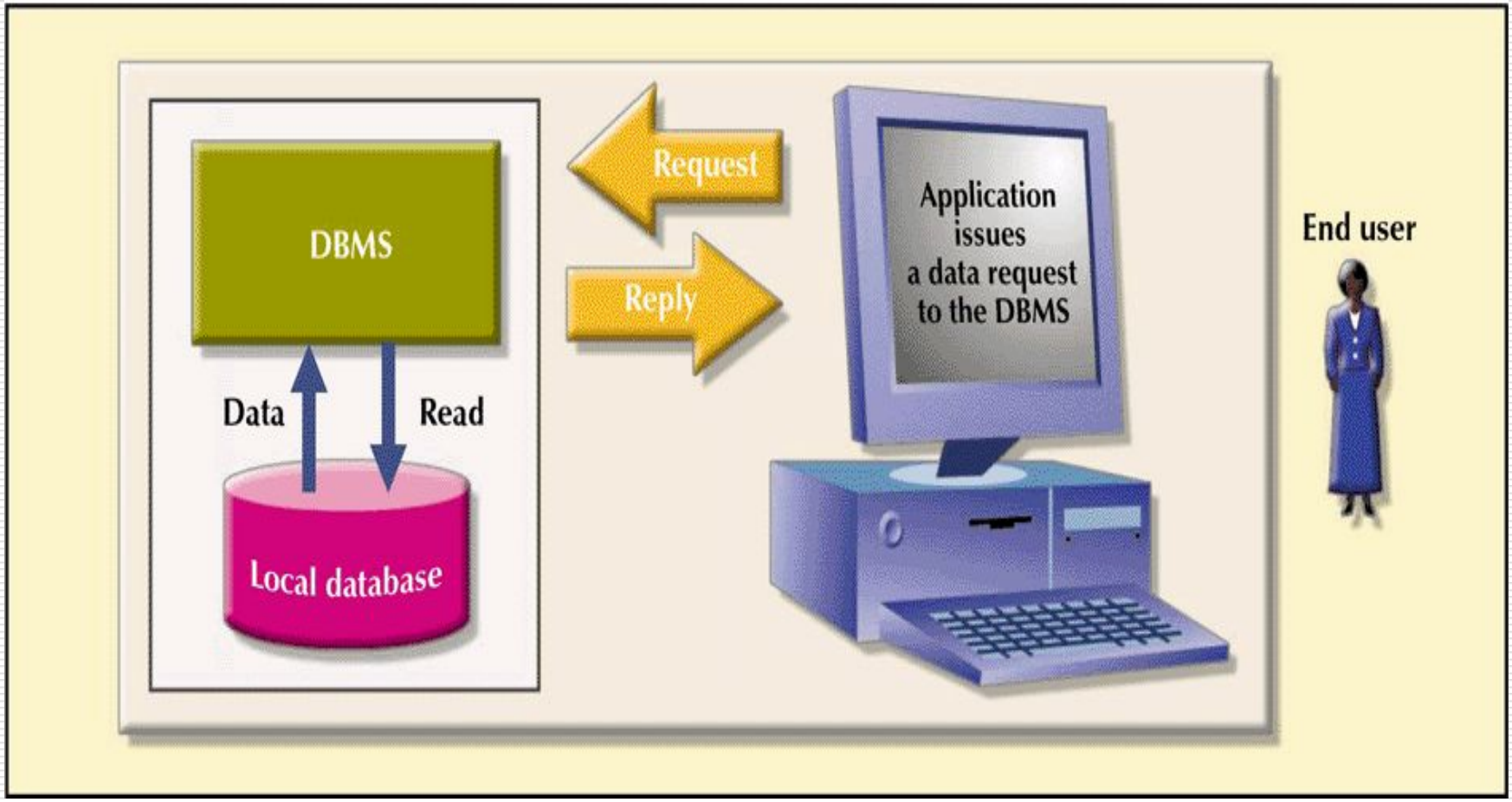
- Distributed database management system (DDBMS) :
  - Governs storage and processing of logically related data over interconnected computer systems in which both data and processing functions are distributed among several sites

# The Evolution of Distributed Database Management Systems (DDBMS)

- Centralized database required that corporate data be stored in a single central site
  - Performance degradation as number of remote sites grew
  - High cost to maintain large centralized DBs
  - Reliability problems with one, central site.
  - Organization rigidity i.e organization might not support the flexibility and agility by modern global organization.
  - Scalability problems i.e physical space , temperature conditioning and power consumption.

# Centralized Database Management System

FIGURE 10.1 CENTRALIZED DATABASE MANAGEMENT SYSTEM



# DDBMS Advantages

- Data are located near "greatest demand" site.
- Faster data access.
- Faster data processing .
- Growth facilitation – new sites can be added to the network without affecting the operation of other sites.
- Improved communications : better communication among department and between customers and company staff.

- Reduced operating costs : It is more ost effective to add nodes to a network than to update a mainframe system.
- User-friendly interface .
- Less danger of a single-point failure .
- Processor independence : end user can access any available copy of the data and request is processed by any processor at the data location.

# DDBMS Disadvantages

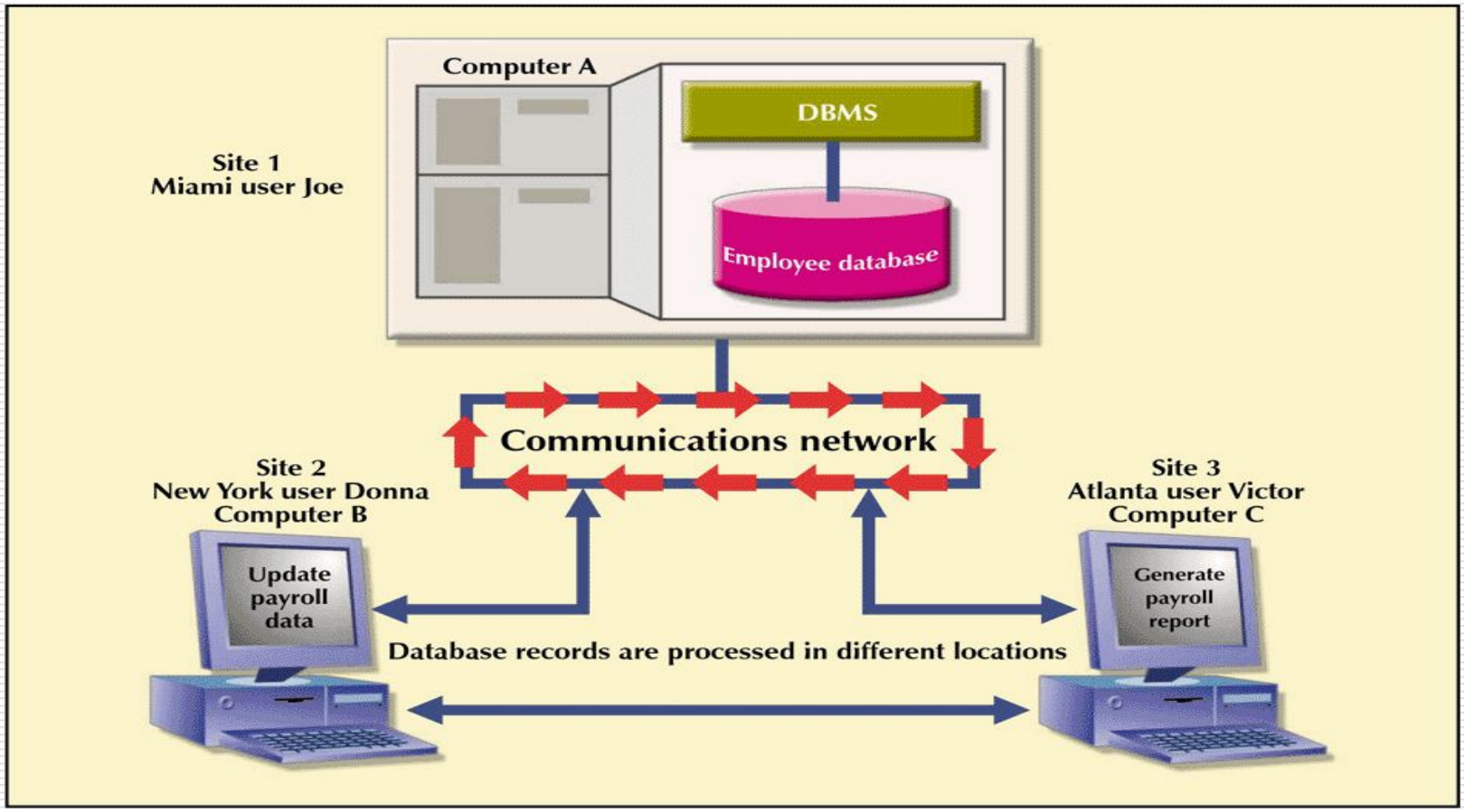
- Complexity of management and control
- Security
- **Lack of standards :** no standard communication protocol.
- **Increased storage and infrastructure requirements :** multiple copies of data are required at different sites , thus requiring additional storage space.
- Greater difficulty in managing the data environment / Technological difficulty
- **Increased cost :** both in training and in infrastucture.

# Distributed Processing vs Distributed Database

- Distributed processing – a database's logical processing is shared among two or more physically independent sites that are connected through a network
  - One computer performs I/O, data selection and validation while second computer creates reports
  - Uses a single–site database but the processing chores are shared among several sites
- Distributed database – stores a logically related database over two or more physically independent sites. The sites are connected via a network
  - Database is composed of database fragments which are located at different sites and may also be replicated among various sites

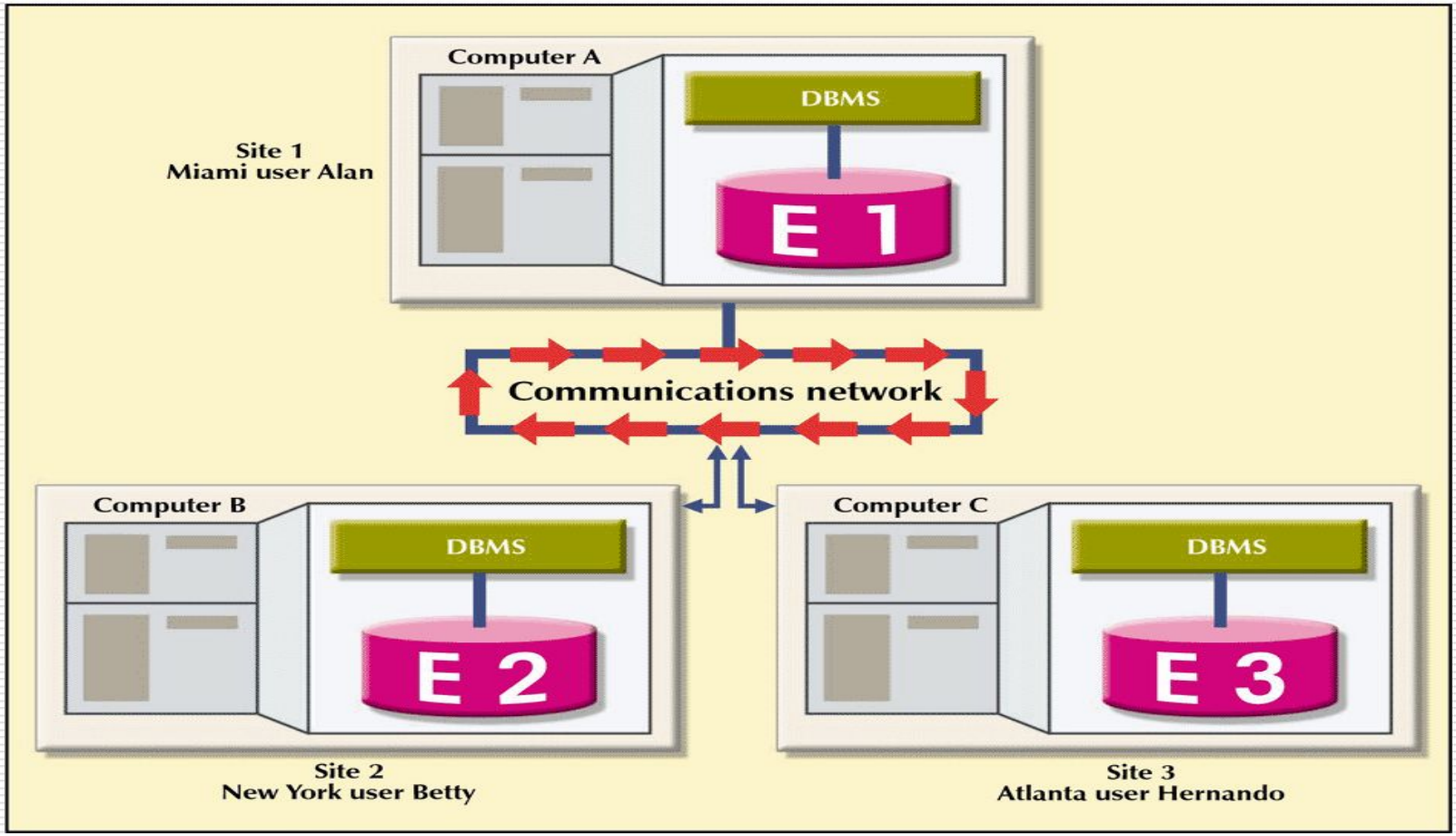
# Distributed Processing Environment

FIGURE 10.2 DISTRIBUTED PROCESSING ENVIRONMENT



# Distributed Database Environment

FIGURE 10.3 DISTRIBUTED DATABASE ENVIRONMENT



- Distributed processing does not require a distributed database, but a distributed database requires distributed processing.
- Distributed processing may be based on a single database located on a single computer.
- Both distributed processing and distributed databases require a network of interconnected components.

# Characteristics of a DDBMS

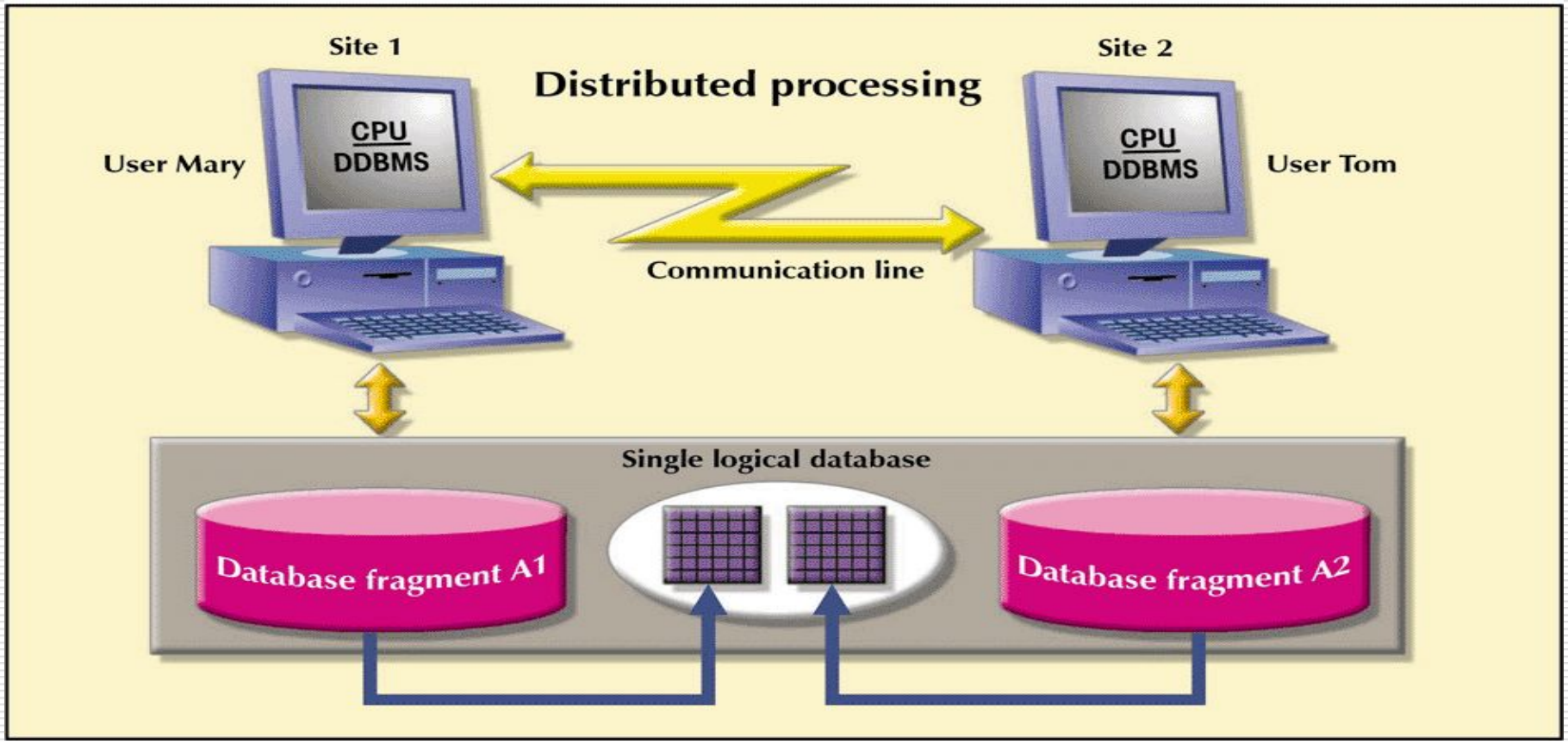
1. **Application interface** to interact with the end user, application programs, and other DBMSs within the distributed database.
2. **Validation** to analyze data requests for syntax correctness.
3. **Transformation** to decompose complex requests into atomic data request components.
4. **Query optimization** to find the best access strategy. (Which database fragments must be accessed by the query,and how must data updates, if any, be synchronized?)
5. **Mapping** to determine the data location of local and remote fragments.

   I/O interface to read or write data from or to permanent local storage.

6. **Formatting** to prepare the data for presentation to the end user or to an application program.
7. **Security** to provide data privacy at both local and remote databases.
8. **Backup and recovery** to ensure the availability and recoverability of the database in case of a failure.
9. **DB administration** features for the database administrator.
10. **Concurrency control** to manage simultaneous data access and to ensure data consistency across database fragments in the DDBMS.
11. **Transaction management** to ensure that the data moves from one consistent state to another. This activityincludes the synchronization of local and remote transactions as well as transactions across multiple distributed segments.

# A Fully Distributed Database Management System

FIGURE 10.4 A FULLY DISTRIBUTED DATABASE MANAGEMENT SYSTEM

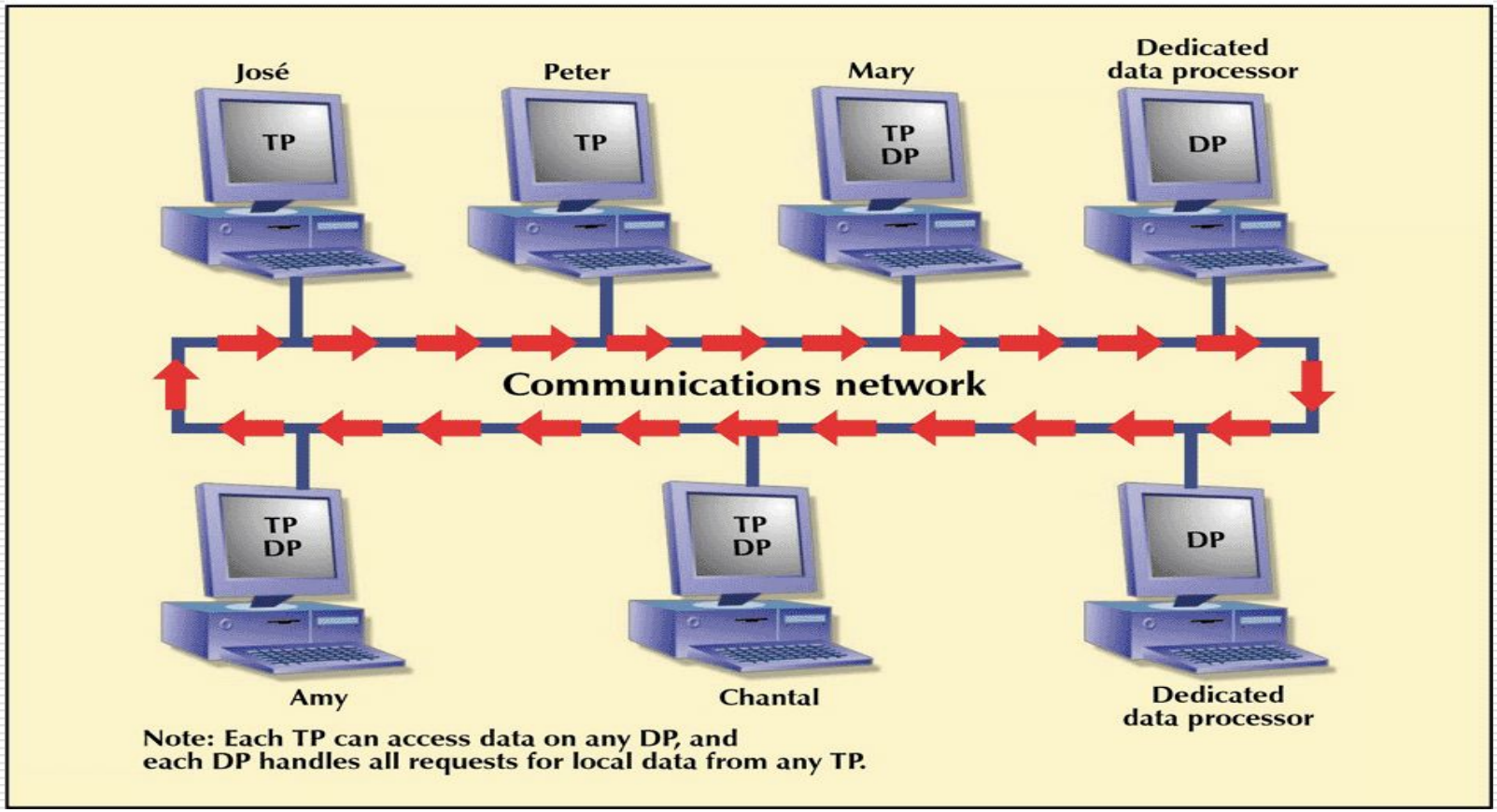


# DDBMS Components

- Must include (at least) the following components:
  - Computer workstations
  - Network hardware and software
    - Allows all sites to interact and exchange data
  - Communications media
    - Carry the data from one workstation to another
  - Transaction processor (application processor or transaction manager)
    - Software component found in each computer that receives and processes the application's requests data
  - Data processor or data manager
    - Software component residing on each computer that stores and retrieves data located at the site
    - May even be a centralized DBMS
    - Communications between the TPs and DPs is made possible through a set of protocols used by the DDBMS

# Distributed Database System Components

FIGURE 10.5 DISTRIBUTED DATABASE SYSTEM COMPONENTS



# Database Systems: Levels of Data and Process Distribution

TABLE 10.1 DATABASE SYSTEMS: LEVELS OF DATA AND PROCESS DISTRIBUTION

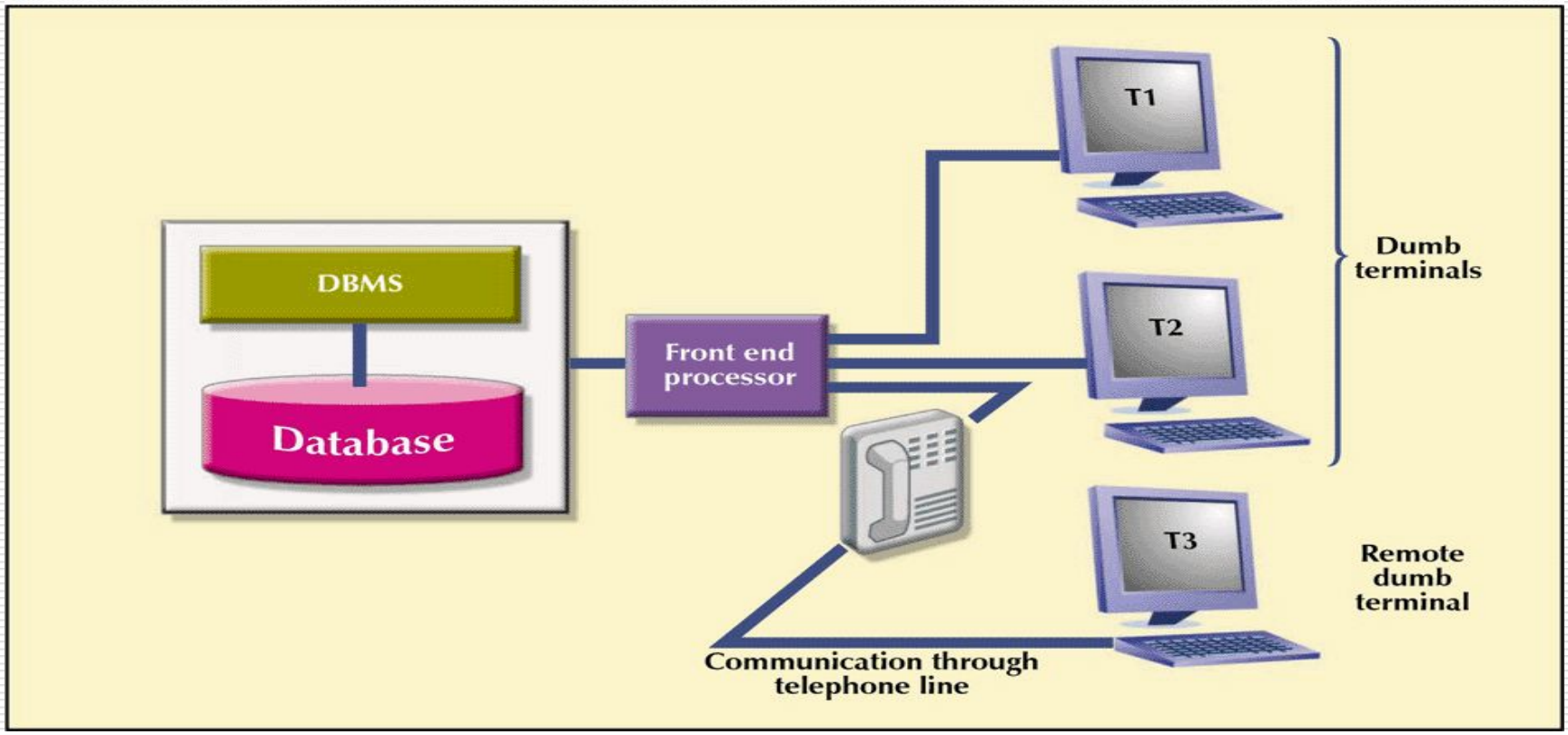
| | SINGLE-SITE DATA | MULTIPLE-SITE DATA |
|---|---|---|
| **Single-site process** | Host DBMS (Mainframes) | Not applicable<br>(Requires multiple processes) |
| **Multiple-site process** | File server<br>Client/server DBMS (LAN DBMS) | Fully distributed<br>Client/server DDBMS |

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

- All processing is done on single CPU or host computer (mainframe, midrange, or PC)
- All data are stored on host computer's local disk
- Processing cannot be done on end user's side of the system
- Typical of most mainframe and midrange computer DBMSs
- DBMS is located on the host computer, which is accessed by dumb terminals connected to it
- Also typical of the first generation of single-user microcomputer databases

# Single–Site Processing, Single–Site Data (Centralized)

FIGURE 10.6 SINGLE-SITE PROCESSING, SINGLE-SITE DATA (CENTRALIZED)



# Multiple–Site Processing, Single–Site Data (MPSD)

- Multiple processes run on different computers sharing a single data repository
- MPSD scenario requires a network file server running conventional applications that are accessed through a LAN
- Many multi–user accounting applications, running under a personal computer network, fit such a description

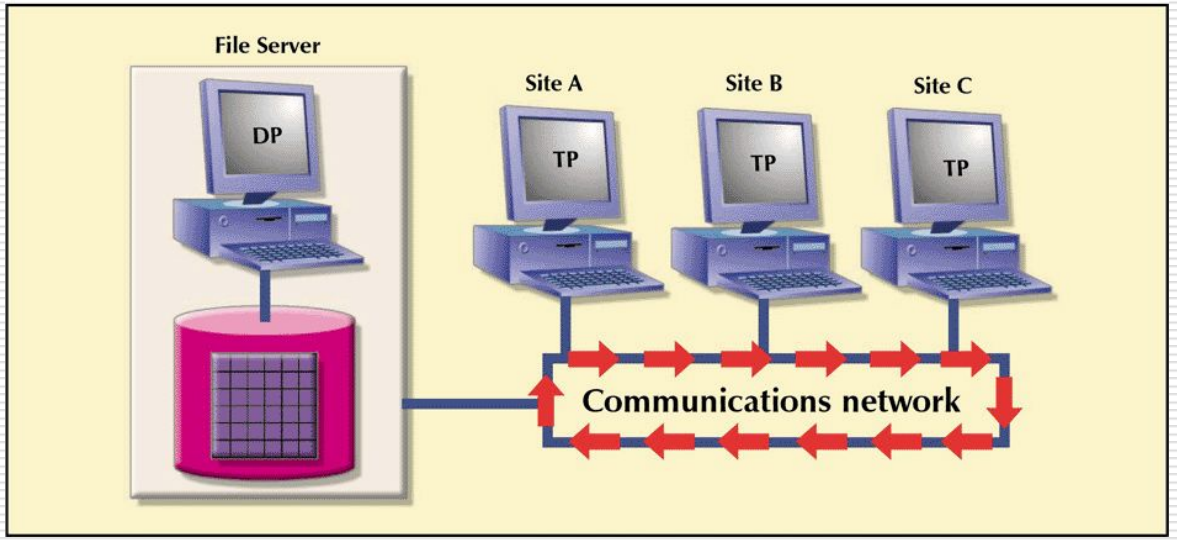
# Multiple-Site Processing, Single-Site Data (MPSD)

- TP at each workstation acts only as a redirector to route all network data requests to the file server
- All record and file locking activity occurs at the end-user location
- All data selection, search and update functions takes place at the workstation. This requires entire files to travel through the network for processing at the workstation. This increases network traffic, slows response time and increases communication costs
  - To perform SELECT that results in 50 rows, a 10,000 row table must travel over the network to the end-user

# Multiple–Site Processing, Single–Site Data (MPSD)

- In a variation of MPSD known as *client/server architecture*, all processing occurs at the server site, reducing the network traffic
- The processing is distributed; data can be located at multiple sites

FIGURE 10.7 MULTIPLE-SITE PROCESSING, SINGLE-SITE DATA



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

- Fully distributed database management system with support for multiple data processors and transaction processors at multiple sites
- Classified as either homogeneous or heterogeneous
- Homogeneous DDBMSs
  - Integrate only one type of centralized DBMS over a network
  - The same DBMS will be running on different mainframes, minicomputers and microcomputers
- Heterogeneous DDBMSs
  - Integrate different types of centralized DBMSs over a network
- Fully heterogeneous DDBMS
  - Support different DBMSs that may even support different data models (relational, hierarchical, or network) running under different computer systems, such as mainframes and microcomputers
- No DDBMS currently provides full support for heterogeneous or fully heterogeneous DDBMSs