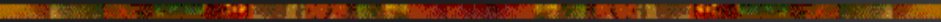


# Evolution of Distributed Database Management System

- During the 1970s, corporations implemented centralized database management systems to meet their structured information needs.
- Structured information is usually presented as regularly issued formal reports in a standard format.

- The use of centralized database required that corporate data be stored in a single central site, usually a mainframe computer.
- Data access was provided through serially connected **dumb terminals**.
- The centralized approach worked well to fill the structured information needs of corporations, but it fell short when quickly moving events required faster response times and equally quick access to information.



➤ The 1980s gave birth to a series of crucial social and technological changes that affected database development and design, such as:

- ✓ Business operations became more decentralized geographically.
- ✓ Competition increased at the global level.
- ✓ Customer demands and market needs favored a decentralized management style.



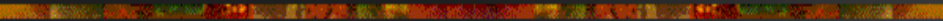
✓ Rapid technological change – **low cost mainframe.**

✓ The large number of applications based on DBMS and the need to protect investments in centralized DBMS software made the notion of data sharing attractive.



During 1990s the factors we have just described became even more firmly pressured. However, how these factors were addressed was strongly influenced by:

- The growing acceptance of **internet** and, particularly, the **WWW**, as the platform for data access and distribution. The WWW is, in effect, the repository for distributed data.
- The increased focus on data analysis that led to data mining and warehousing. Although a data warehouse is not usually a distributed database, it does rely on techniques-such as data replication and distributed queries-that facilitate its data extraction and integration.



The decentralized database is especially desirable because centralized database management is subject to problems such as:

- **Performance degradation** due to a growing number of remote locations over greater distance.
- **High Cost** associated with maintaining and operating large central (mainframe) database systems.
- **Reliability problems** created by dependence on central site.

# Advantages of DDBMS

- **Data are located near the “greatest demand” site:**

The data in a distributed database system are dispersed to match business requirements.



# •Faster Data Processing:

A distributed database system makes it possible to process data at several sites, thereby spreading out the system's workload.



## • **Faster Data Access:**

End users often work with only a subset of the company's data. If such data subsets are locally stored and accessed, the database system will deliver faster data access than is possible with remotely located centralized data.



# •Growth Facilitation:

New sites can be added to the network without affecting the operations of other sites. Such flexibility enables the company to expand relatively easily and rapidly.



# **Improved Communications:**

Because local sites are smaller and located closer to customers, local sites foster better communications among departments and between customers and company staff. Quicker and better communication often help to improve information systems.

# • **Reduced operating costs:**

It is much more cost-effective to add workstations to a network than to update a mainframe system. The cost of dedicated data communication lines and mainframe software is reduced proportionally. Development work is done more cheaply and more quickly on low-cost PCs than on mainframe.



# •User Friendly Interface:

PCs and workstations are usually endowed with an easy-to-use GUI. The GUI simplifies use and training for end users.

• **Less danger of a single-point failure:** In a centralized system, the mainframe's failure brings down all the system's operations. In contrast, a distributed system is able to shift operations when one of the computer fails. The system workload is picked up by other workstations, because one of the distributed system's features is that data exist at multiple sites.

# • **Processor Independence:**

The end user is able to access any available copy of the data and an end user's request is processed by any processor at the data location. In other words, requests do not depend on a specific processor; any available processor can handle the user's request.

# Disadvantages of DDBMS

- **Complexity of Management and control:**

Management of distributed data is a more complex task than management of centralized data. Application must recognize data location, and they must be able to stitch together data from different sites. DBA must have the ability to coordinate database activities to prevent database degradation due to data anomalies. Transaction management, concurrency control, security, backup recovery, query optimization, access path selection, and so on, must be addressed and resolved.





- **Security:**

The probability of security lapses increases when data are located at multiple sites. The responsibility of data management will be shared by different people at several sites, and LANs do not yet have the sophisticated security of centralized mainframe installation.

## • **Lack of Standards:**

Although distributed databases depend on effective communication, there are no standard communication protocols. In fact, few official standards exist in any of the distributed database protocols, whether they deal with communication or data access control. Consequently, distributed database users must wait for the definitive emergence of standard protocols before distributed databases can deliver all their potential goods.

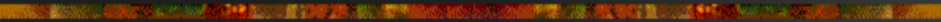
## • **Increased Storage Requirements:**

Data replication requires additional disk storage space. This disadvantage is minor one, because disk storage space is relatively cheap and it is becoming cheaper. However, disk access and storage management is a widely dispersed data storage environment are more complex than they would be in a centralized database.

# Distributed Processing and Distributed Databases

---

**Distributed processing** shares the database's logical processing among two or more physically independent sites that are connected through a network. For example, distributed processing might perform the data input/output, the data selection, and the data validation in one computer, and then create a report based on such data in another computer.



A **distributed database**, on the other hand, stores a logically related database over two or more physically sites. The sites are connected via a computer network. In contrast, the distributed processing system uses only a single-site database but shares the processing chores among the several sites. In a distributed database system, a database is composed of several parts known as database fragments. The database fragments are located at different sites.

# Points to remember:

---

- 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. In order to manage distributed data, copies or parts of the database processing functions must be distributed to all data storage sites.
- Both distributed processing and distributed databases require a network to connect all components.

# DDBMS

---

A distributed database management system governs the storage and processing of logically related data over interconnected computer systems in which both data and processing functions are distributed among several sites. A DBMS must have at least the following functions to be classified as distributed:

- Application interface to interact with end user or application programs and with other DBMSs within the distributed databases.
  - Validation to analyze data requests.
-

- Transformation to determine which data request components are distributed and which ones are local.
- Query optimization to find the best access strategy.
- 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.
- Formatting to prepare the data for presentation to the end user or an application program.



- Security to provide data privacy at both local and remote databases.
- Backup and recovery to ensure the availability and recoverability of the database in case of a failure.
- DB administration for the database administrator.
- Concurrency control to manage simultaneous data access and ensure data consistency across database fragments in the DDBMS.
- Transaction management to ensure that the data move from one consistent state to another.

# DDBMS Components

The DDBMS must include the following components:

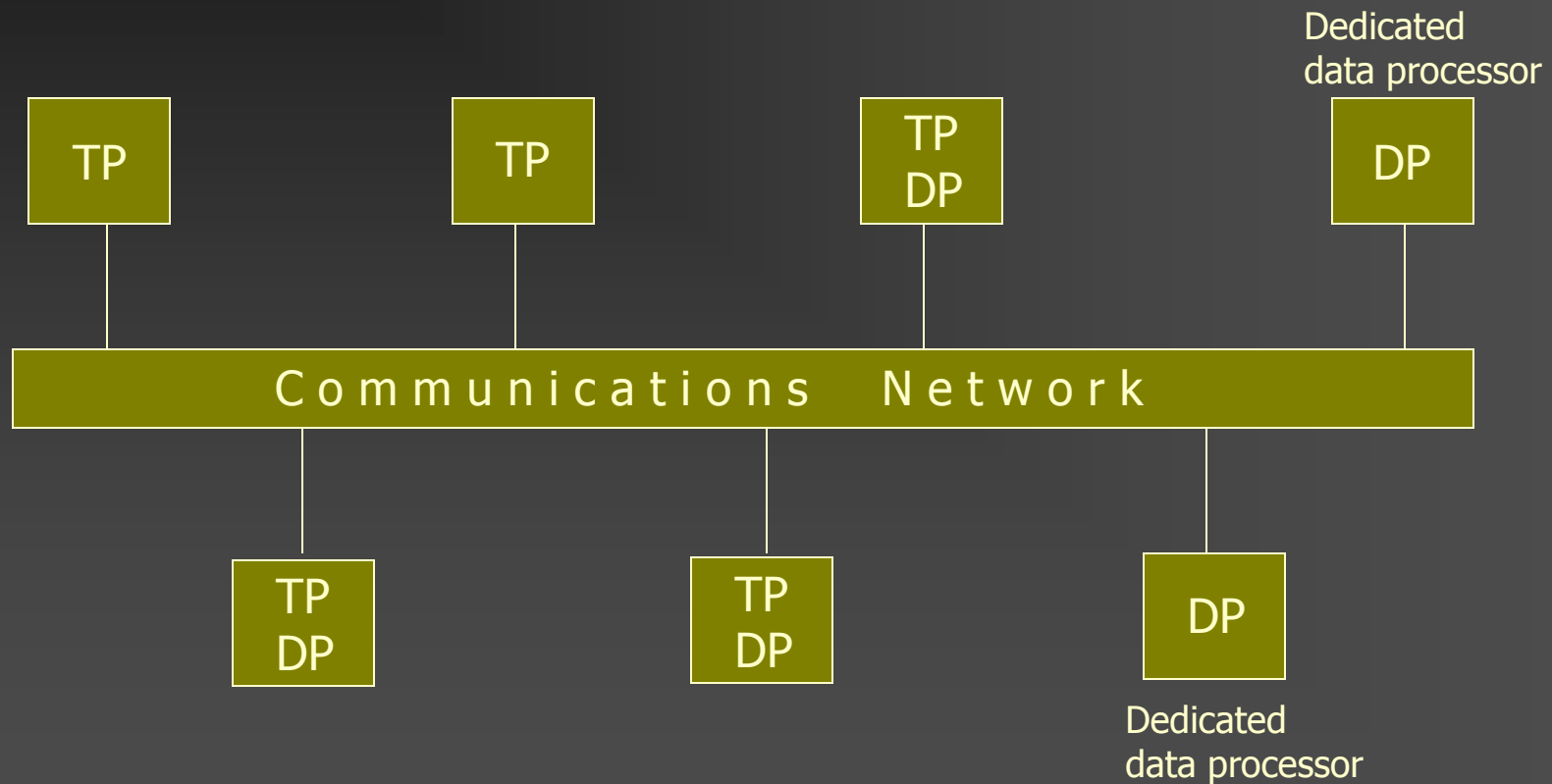
- **Computer Workstations** that form the network system. The distributed database system must be independent of the computer system hardware.
- **Network hardware and software** components that reside in each workstation. The network components allow all sites to interact and exchange data. Network system independence is a desirable distributed database system attribute.

• **Communication media** that carry the data from one workstation to another. The DDBMS must be communications-media-independent; that is, it must be able to support several types of communications media.

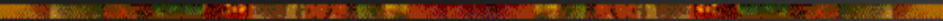
• **The Transaction Processor(TP)** which is the software component found each computer that requests data. The transaction processor receives and processes the application's data requests (remote and local). The TP is also known as the application processor(AP) or the transaction manager(TM).

- The **data processor(DP)**, which is the software component residing on each computer that stores and retrieves data located at the site. The DP is also known as the data manager (DM). A data processor may even be a centralized DBMS.

# Communication among TPs and DPs



**Note: Each TP can access data on any DP, and each DP handles all requests for local data from any TP**



The Protocols determine how the distributed database system will:

- Interface with the network to transport data and commands between DPs and TPs.
- Synchronize all data received from DPs and route retrieved data to the appropriate TPs.
- Ensure common database functions in a distributed system. Such functions include security, concurrency control, backup and recovery.

# Levels of Data and Process Distribution

---

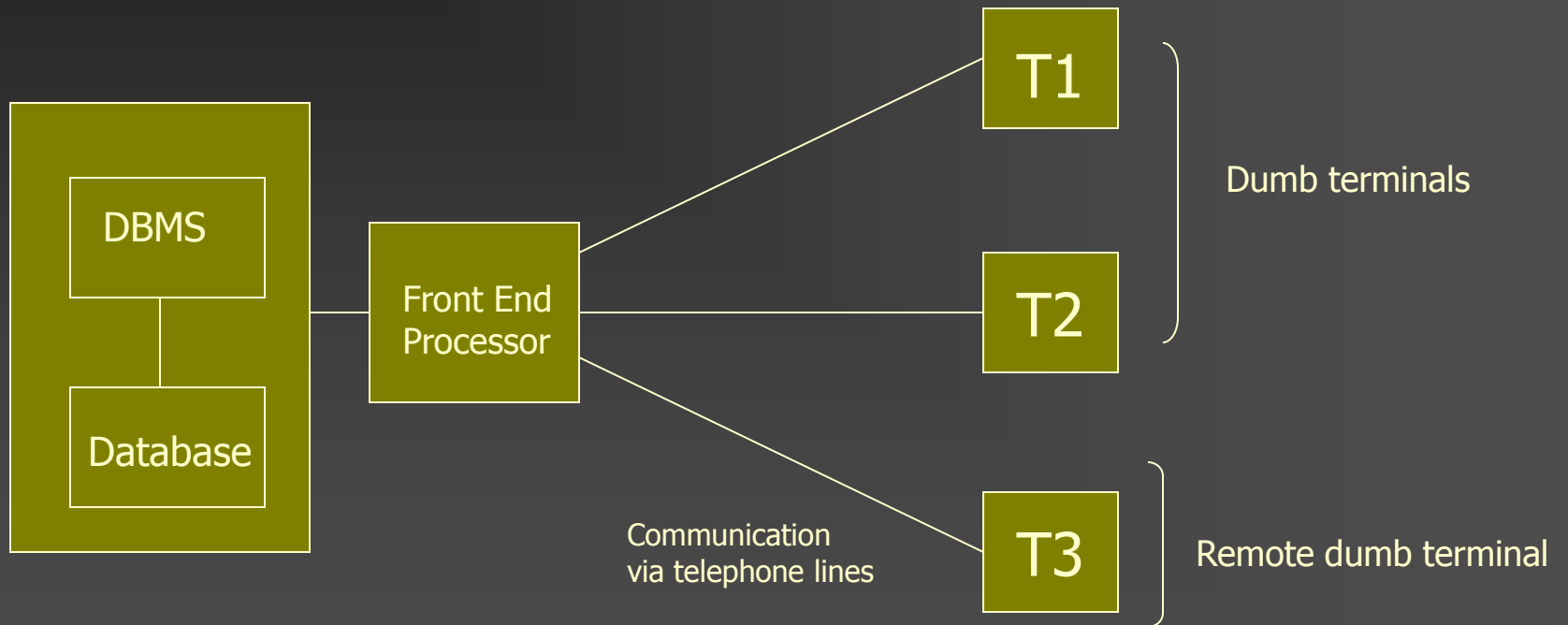
We can classify current database systems on the basis of how process distribution and data distribution are supported. For example, a DBMS may store data in a single site(Centralized DB) or in multiple sites(distributed DB) and may support data processing at a single site and multiple site.

# Single-site Processing, Single-site Data

In the single-site processing, single-site data (SPSD) scenario, all processing is done on a single CPU or host computer and all the data are stored on the host computer's local disk. Processing can be done on the end user's side of the system. Such a scenario is typical of most mainframe and minicomputer DBMSs. The DBMS is located on the host computer, which is accessed by dumb terminals connected to single-user microcomputer databases.



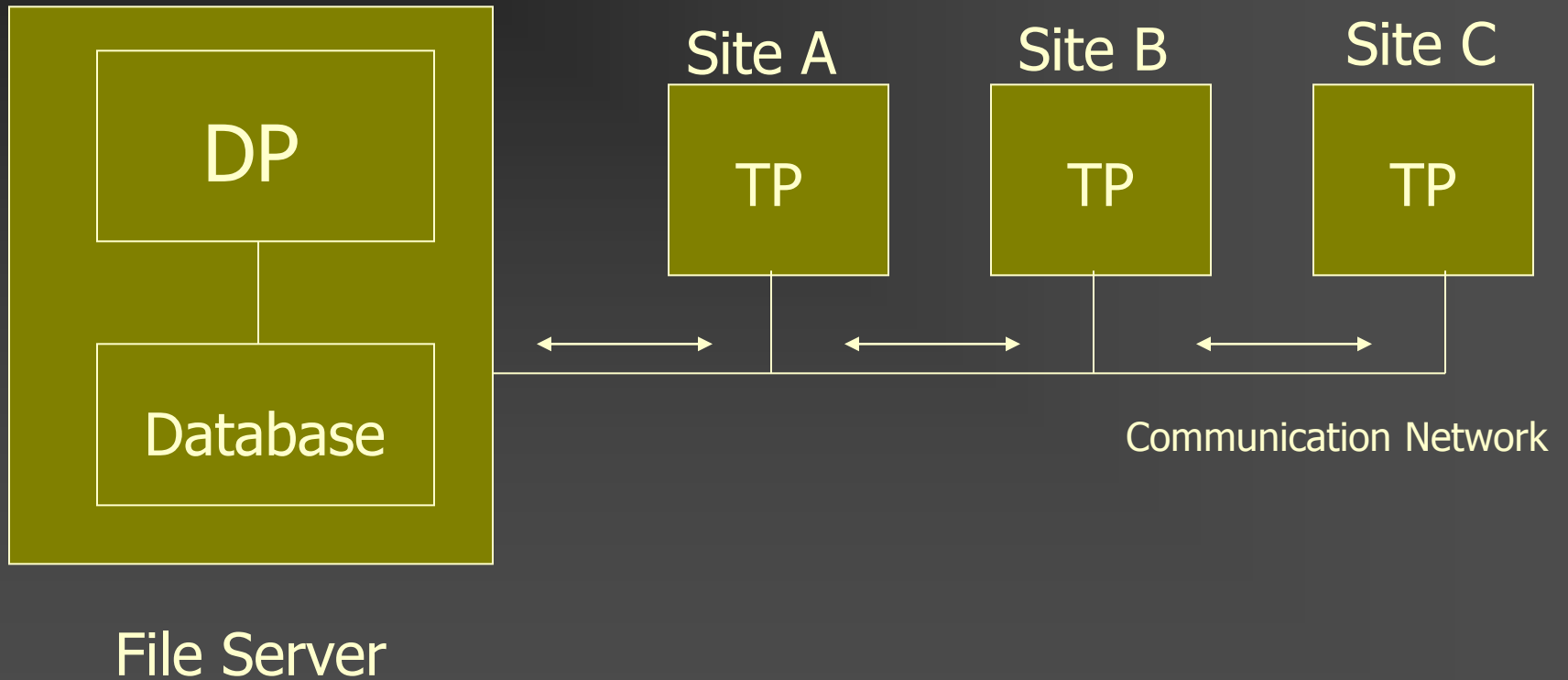
# NONDISTRIBUTED (CENTRALIZED) DBMS



# Multiple-site Processing, Single-site Data

Under the multiple-site processing, single-site data (MPSD) scenario processes run on different sharing a single data repository. Typically, the MPSD scenario requires a network file server on which conventional applications are access through a LAN. Many multi user accounting applications, running under a personal computer network, fit such a description.

# MULTIPLE-SITE PROCESSING, SINGLE-SITE DATA



# MULTIPLE-SITE PROCESSING, MULTIPLE-SITE DATA (MPMD)

MPMD scenario describe a fully distributed database management system with support for multiple data processors and transaction processors at multiple sites. Depending on the level of support for different types of centralized DBMSs, distributed database management systems(DBMSs) are classified as either homogeneous or heterogeneous.

# Homogeneous DDBMS

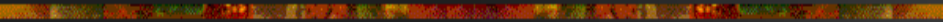
It integrate only one type of centralized DBMS over a network. Thus, the same DBMS will be running on different mainframes, minis, and microcomputers.

# Heterogeneous DDBMS

It integrate different types of central different type of centralized DBMSs over a network. A fully heterogeneous DDBMS will that may even support different data models (relational, hierarchical, or network) running under different computer systems, such as mainframes, minis, and microcomputers.

# Heterogeneous Distributed Database Scenario

Platform	DBMS	Op. system	N/w Comm. Prot.
IBM 3090	DB2	MVS	APPC LU 6.2
DEC VAX	VAX RDB	MVS	DECnet
IBM AS/400	SQL/400	OS/400	3270
RISC	INFORMIX	UNIX	TCP/IP
PENTIUM CPU	ORACLE	OS/2	NetBIOS



No DDBMS currently provides full support for the scenario depicted in the previous figure, nor for the fully heterogeneous environment. Some DDBMS implementations support several platforms, operating systems, and networks and allow remote data access to another DBMS. However, such DDBMSs still are subject to certain restrictions:



- Remote access is provided on a read-only basis and does not support write privileges.
- Restrictions are placed on the number of remote tables that may be accessed in a single transaction.
- Restrictions are placed on the number of distinct databases that may be accessed.
- Restrictions are placed on database model that may be accessed. Thus, access may be provided to relational databases but not to network or hierarchical databases.