Distributed Database Systems

Hoa Dinh Nguyen, PhD.
Department of Information Technology

Content

- General information about distributed database systems
- Architecture of distributed database management systems
- Design of distributed database systems
- Distributed query processing
- Distributed transaction management
- Advance topics

Related Information

References

1. TS. Phạm Thế Quế, **Cơ sở dữ liệu phân tán,** Học viện CNBCVT.

2. M.Tamer Ozsu And Patricle Valduriez, "**Principles of Distributed Database Systems**", Third Edition, Prentive Hall Upper Saddle River, New Jersy, 2011.

Course Organization

- Theory: 23 hours (Midterm exam 2 hours)
- Examples and Homework: 6 hours
- Self-study: 1 hour

Evaluation

- In-class attendance: 10%
- Homework/Course project: 20%
- Midterm exam: 10%
- Final exam: 60%

Course Project

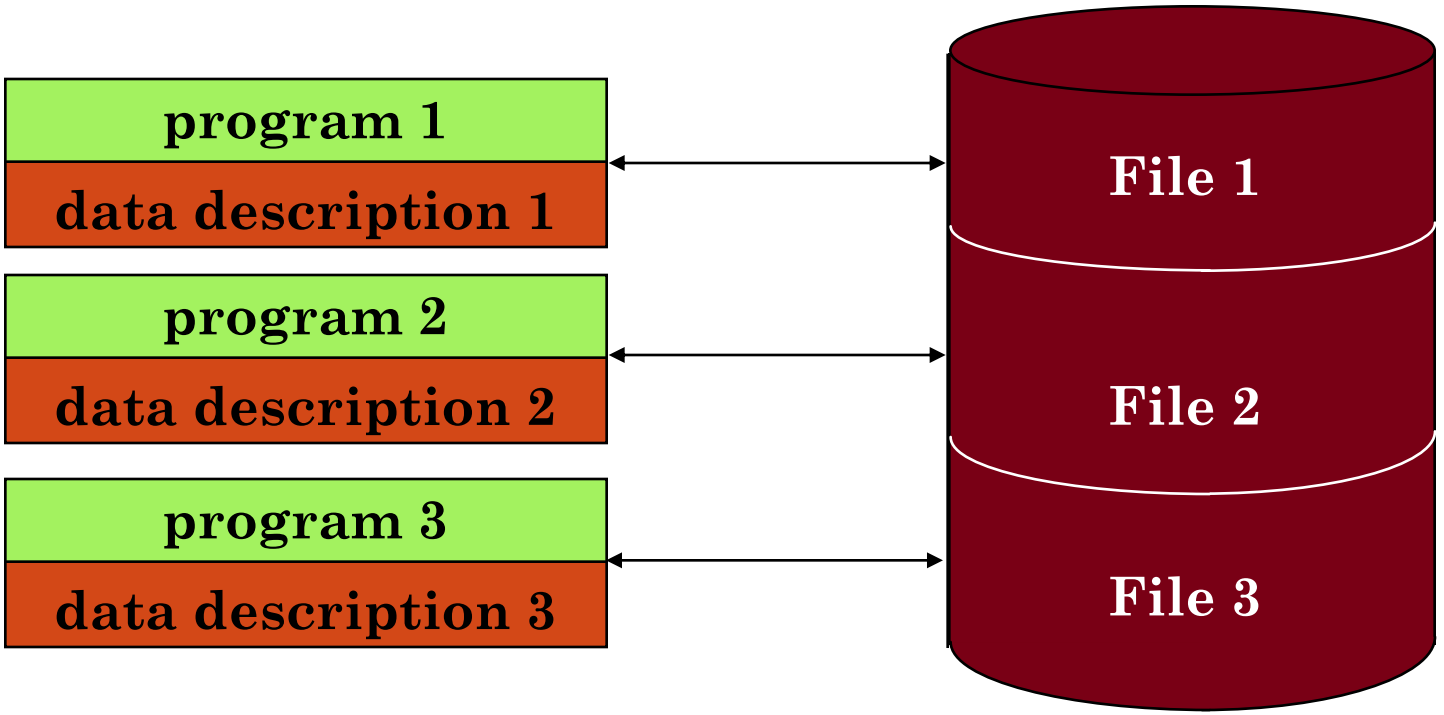
- Design and implement a distributed database system:
  - Work in group to design a distributed database system assigned by the lecturer
  - Implement and manage a distributed database system including:
    - Setup fundamental functions for the system
    - Database Input/output
    - Making queries and reports
  - Report all your work in details.

Lecture 1: Introduction to distributed database systems

Scope:

- Concept of distributed systems
- Classification of distributed systems
- Essential of distributed systems
- Characteristics of distributed systems
- Goals and objectives in designing a distributed system

File System

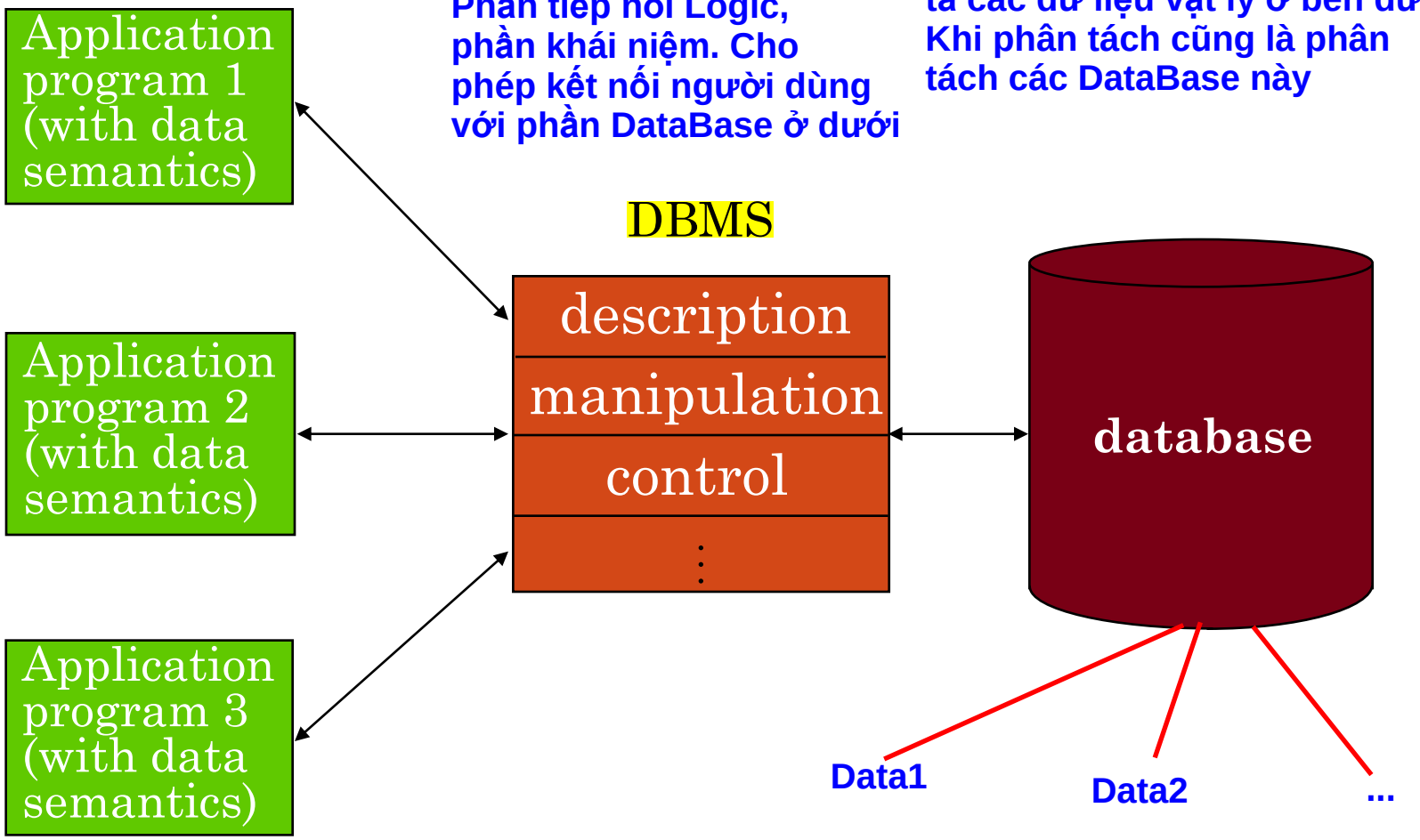


Database Management

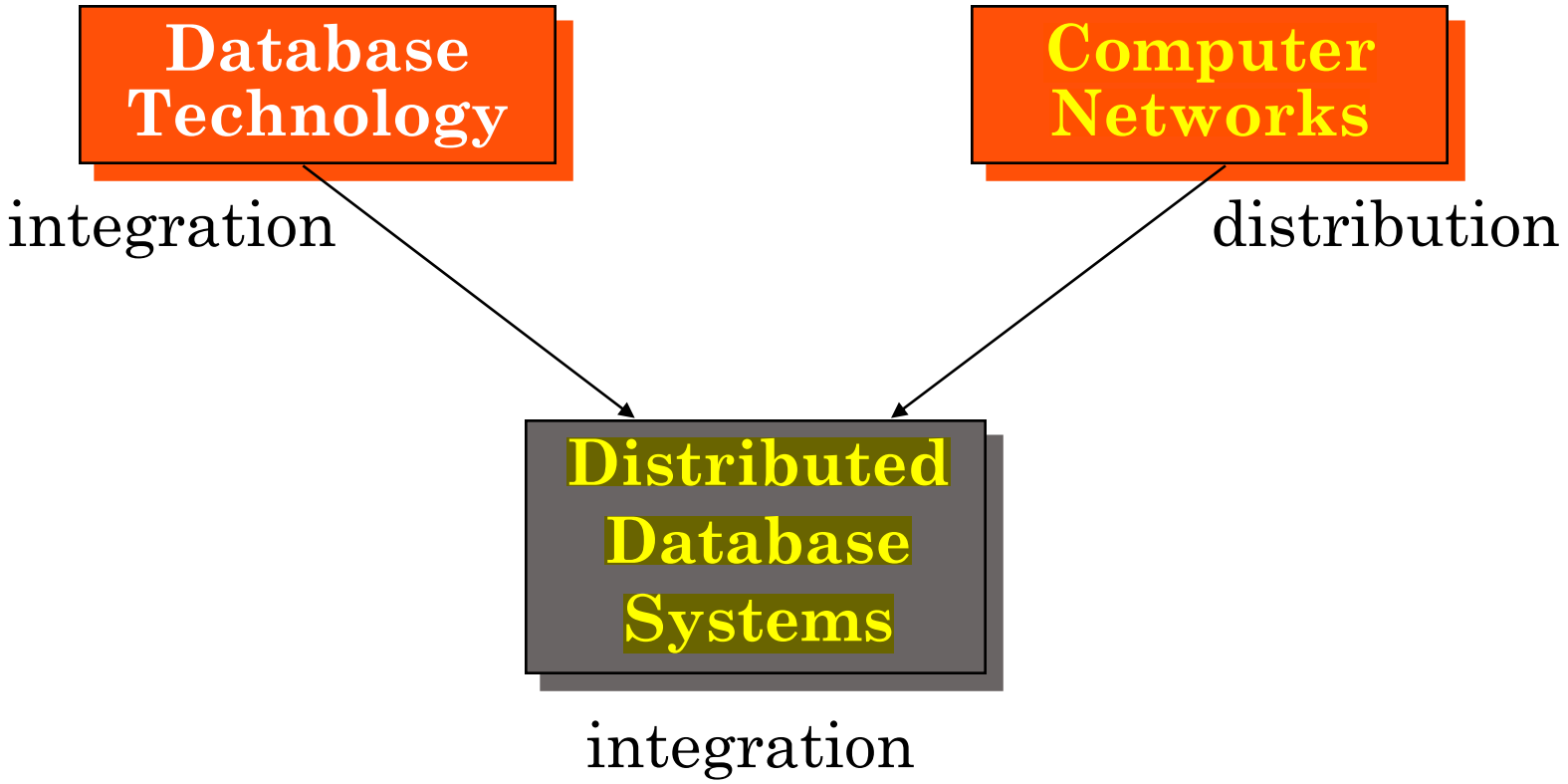
Ứng dụng người dùng
(Lớp User ở bên trên)

Phần tiếp nối Logic, phần khái niệm. Cho phép kết nối người dùng với phần DataBase ở dưới

Về cơ bản là các Meta-data để mô tả các (dữ liệu về mô tả các dữ liệu vật lý ở bên dưới. Khi phân tách cũng là phân tách các DataBase này

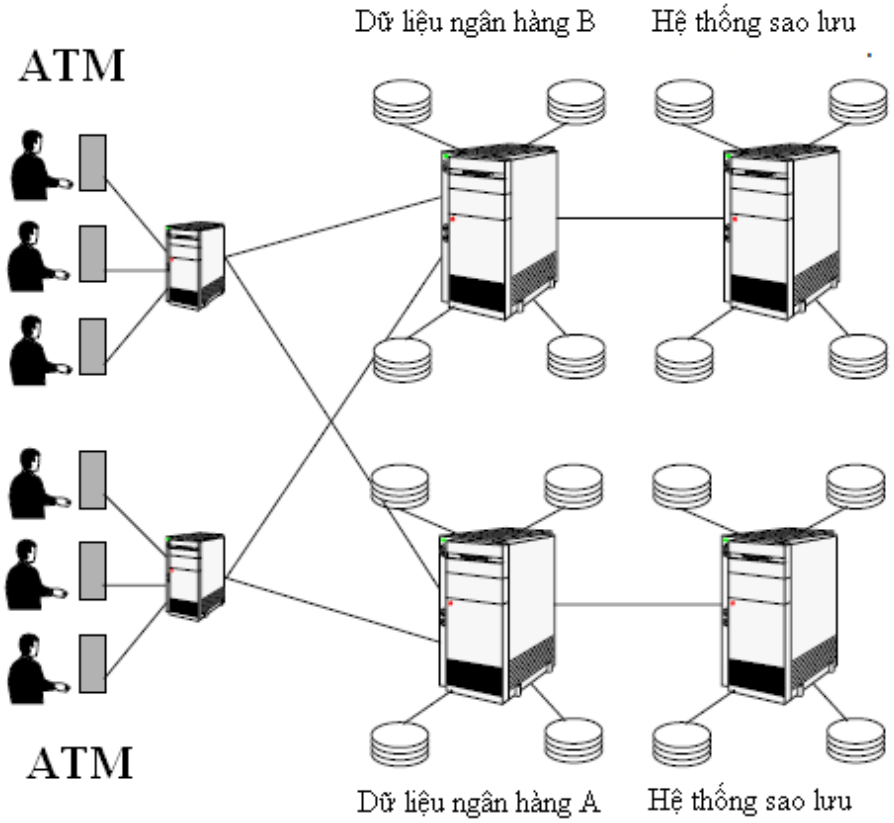


Database and Communication Integration



Distributed Systems

A distributed computing system is a collection of autonomous processing elements that are interconnected by a computer network.



What is distributed?

- Processing logic
- Functions
- Database
- Control

What is distributed database?

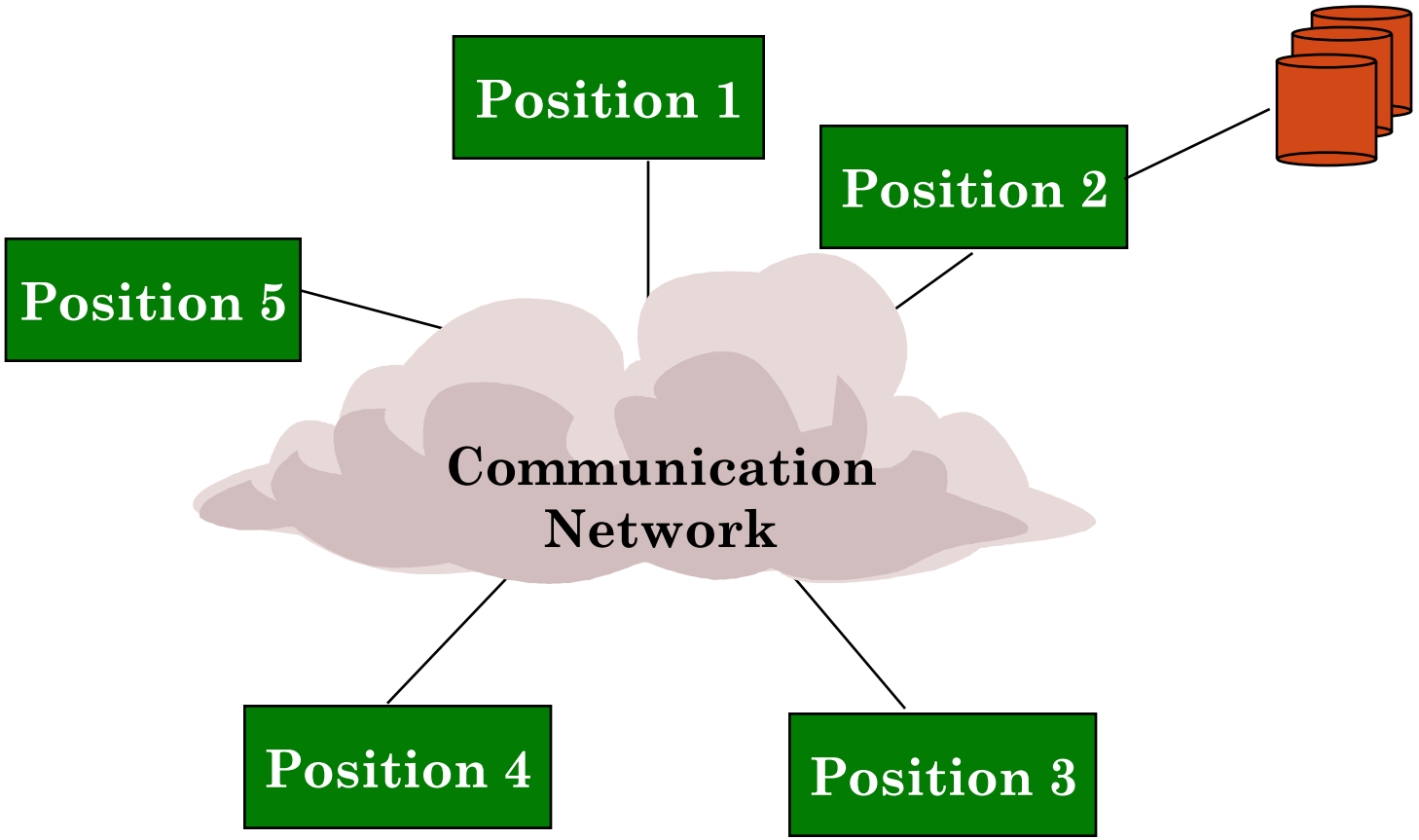
- A distributed database (DDB) is a collection of multiple, logically interrelated databases distributed over a computer network
- A distributed database management system (D–DBMS) is the software that manages the DDB and provides an access mechanism that makes this distribution transparent to the users.

- DDBS = DDB + DDBMS

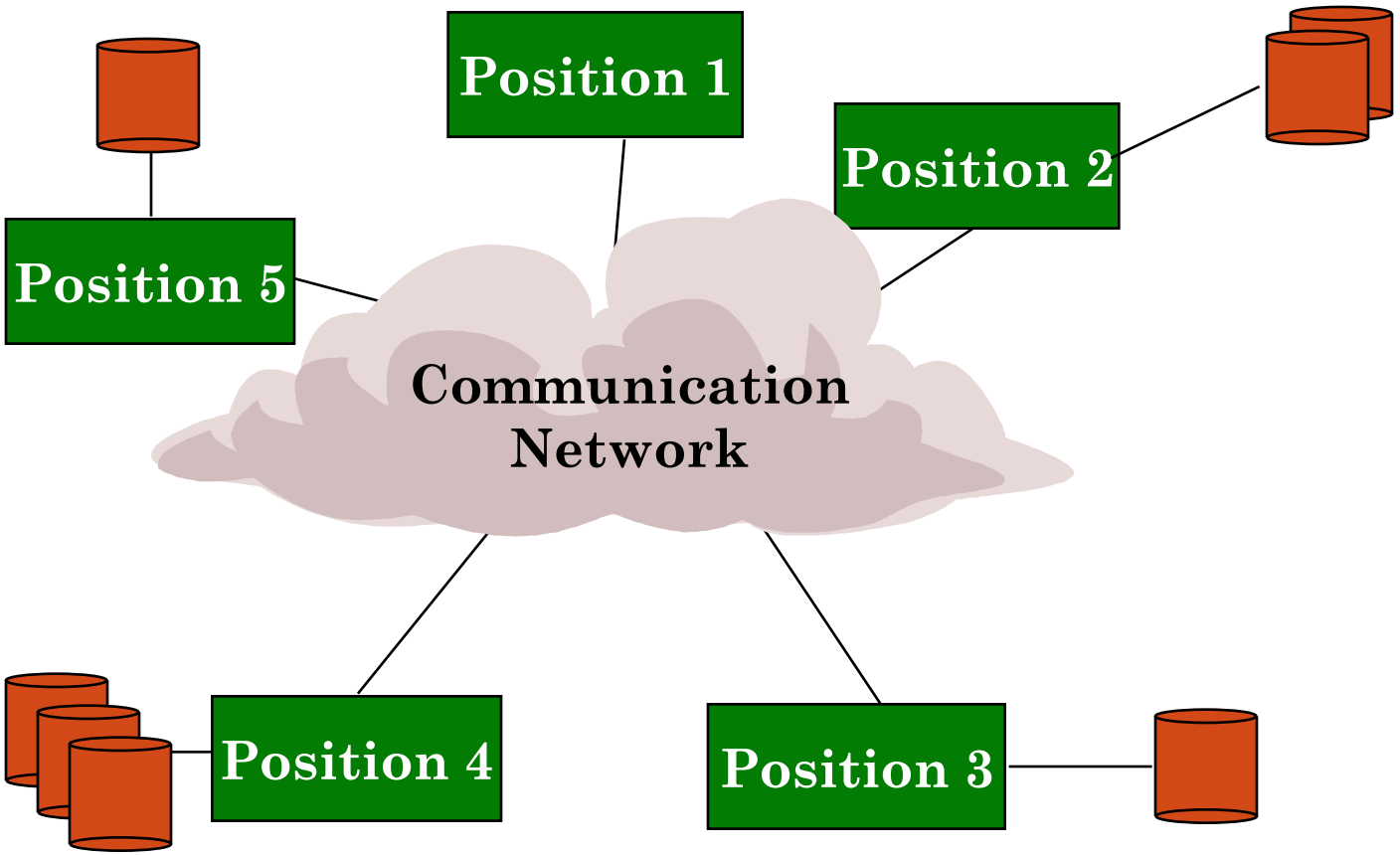
Assumptions

- Data stored at a number of sites, each site logically consists of a single processor
- Processors at different sites are interconnected by a computer network
- DDB is a database, not a collection of files. Data is logically related; structured into multiple files; accessed through common interface
- DDBMS is a collection of DBMSs

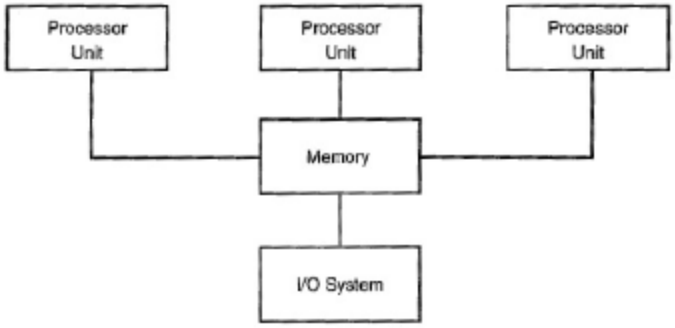
Centralized Database Enviroment



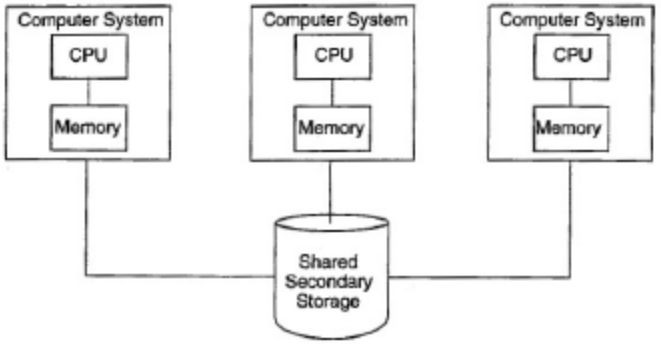
Distributed Database Enviroment



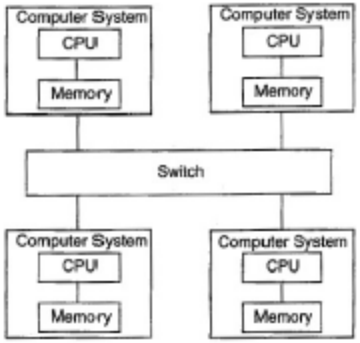
Non-distributed Database Systems



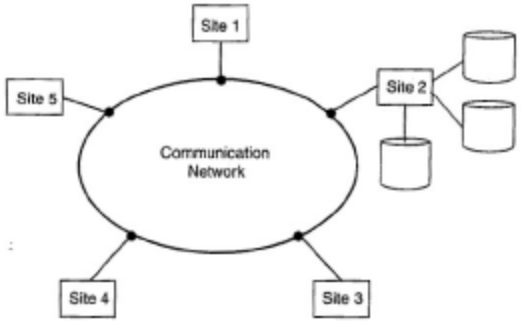
Shared Memory



Shared Disk

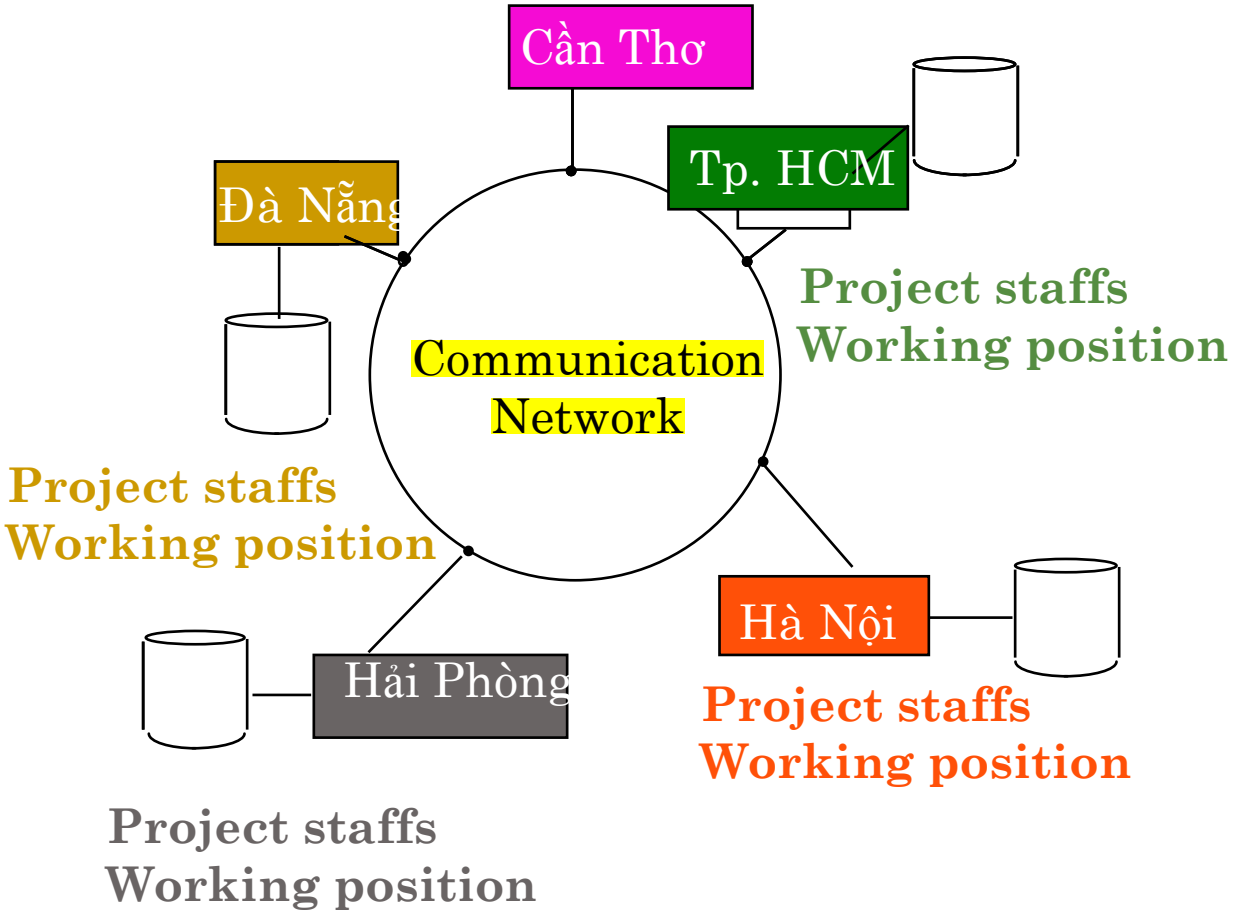


Shared Nothing



Central Databases

Example



Applications of DDBS

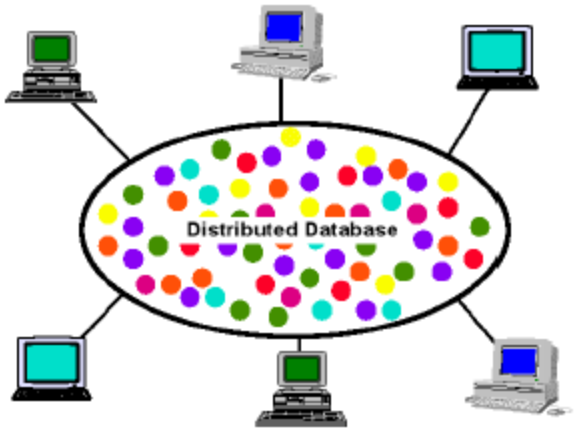
- Manufacturing - especially multi-plant manufacturing
- Military command and control
- Electronic fund transfers and electronic trading
- Corporate MIS
- Airline restrictions
- Hotel chains
- Any organization which has a decentralized organization structure

Advantages of DDBS

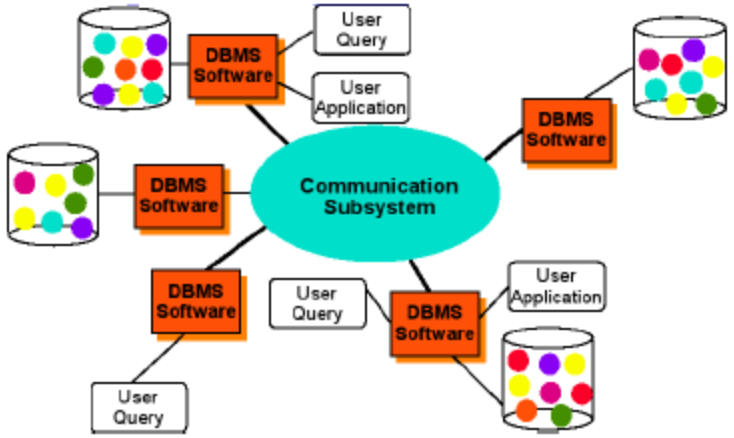
- Transparency of distributed and replicated data
- Higher reliability
- Improved performance
- Easier system expansion

Advantages of DDBS- Transparency (1)

- Separation of the higher-level semantics of the system from the lower-level implementation issues
- A transparent system "hides" implementation details from users
- A fully transparent DBMS provides high-level support for the development of complex applications.



(a) User wants to see one database



(b) Programmer sees many databases

Advantages of DDBS- Transparency (2)

- The goal of transparency is to provide data independence
- Various forms of transparency can be distinguished for DDBSs:
  - Network transparency (also called distribution transparency)
    - Location transparency
    - Naming transparency
  - Replication transparency
- Fragmentation transparency
  - Horizontal fragmentation
  - Vertical fragmentation

Advantages of DDBS- Reliability

- Higher reliability
- Replication of components and data should make DDBMS more reliable
- No single points of failure (SPOF)
- Distributed transaction processing guarantees the consistency of the database and concurrency

Advantages of DDBS- Performance

Improved performance

- Proximity of data to its points of use
  - Reduces remote access delays
  - Requires some support for fragmentation and replication
- Parallelism in execution
  - Inter-query parallelism
  - Intra-query parallelism
- Update and read-only queries influence the design of DDBSs substantially
  - If mostly read-only access is required, as much as possible of the data should be replicated
  - Writing becomes more complicated with replicated data

Advantages of DDBS- Expansion

Easier system expansion

- Issue is database scaling
- Emergence of microprocessor and workstation technologies
- Network of workstations much cheaper than a single mainframe computer
- Increasing database size

Technical Issues of DDBS (1)

- Distributed database design
  - How to fragment the data?
  - Partitioned data Vs. replicated data?
- Distributed query processing
  - Design algorithms that analyze queries and convert them into a series of data manipulation operations
  - Distribution of data, communication costs, etc. has to be considered
  - Find optimal query plans

Technical Issues of DDBS (2)

- Concurrency Control
  - Synchronization of concurrent accesses
  - Consistency and isolation of transactions' effects
  - Deadlock management
- Reliability
  - How to make the system resilient to failures
  - Atomicity and durability

Technical Issues of DDBS (3)

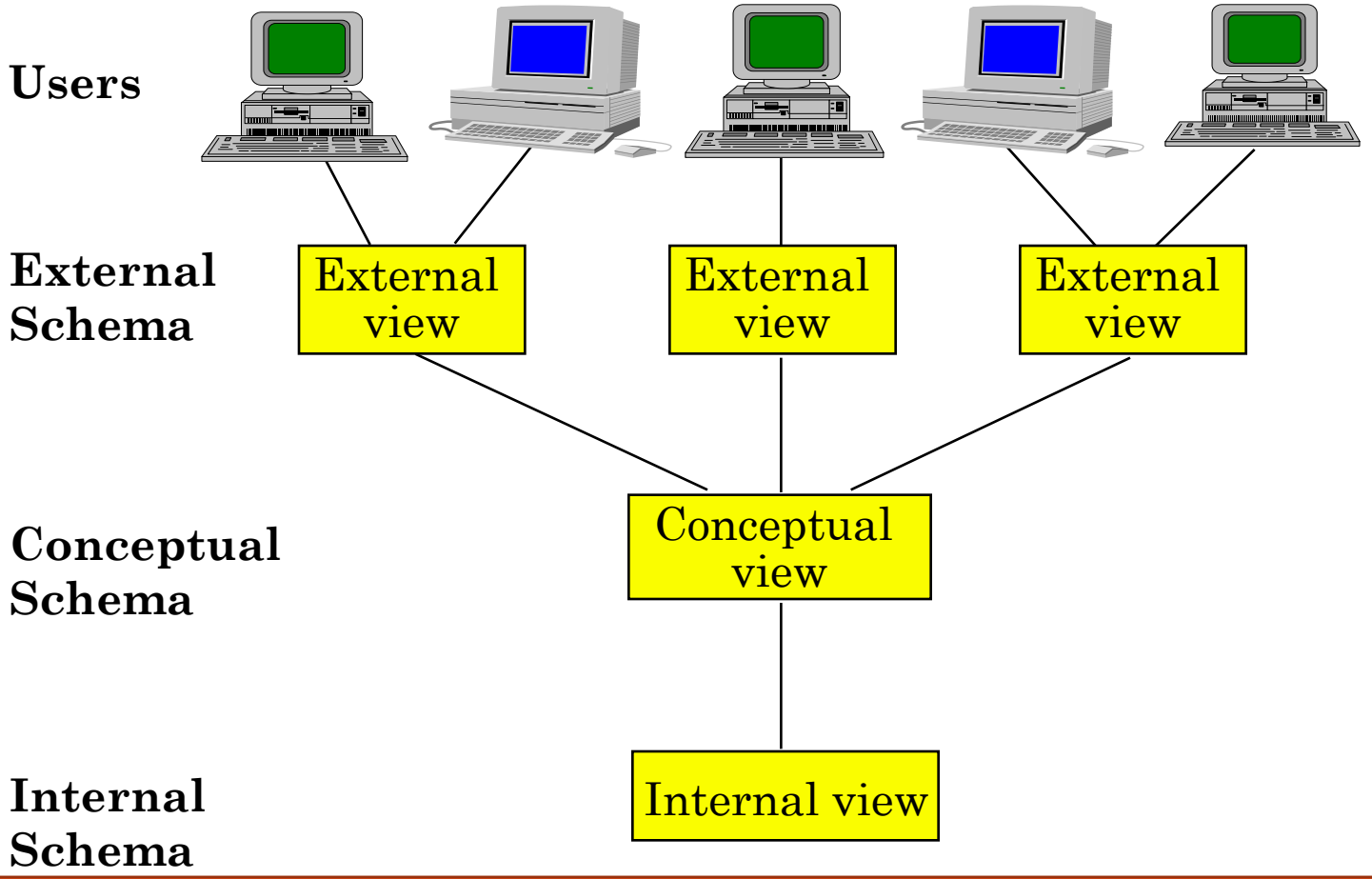
- Privacy/Security
  - Keep database access private
  - Protect against malicious activities
- Trusted Collaborations (Emerging requirements)
  - Evaluate trust among users and database sites
  - Enforce policies for privacy
  - Enforce integrity

DDBS Architecture

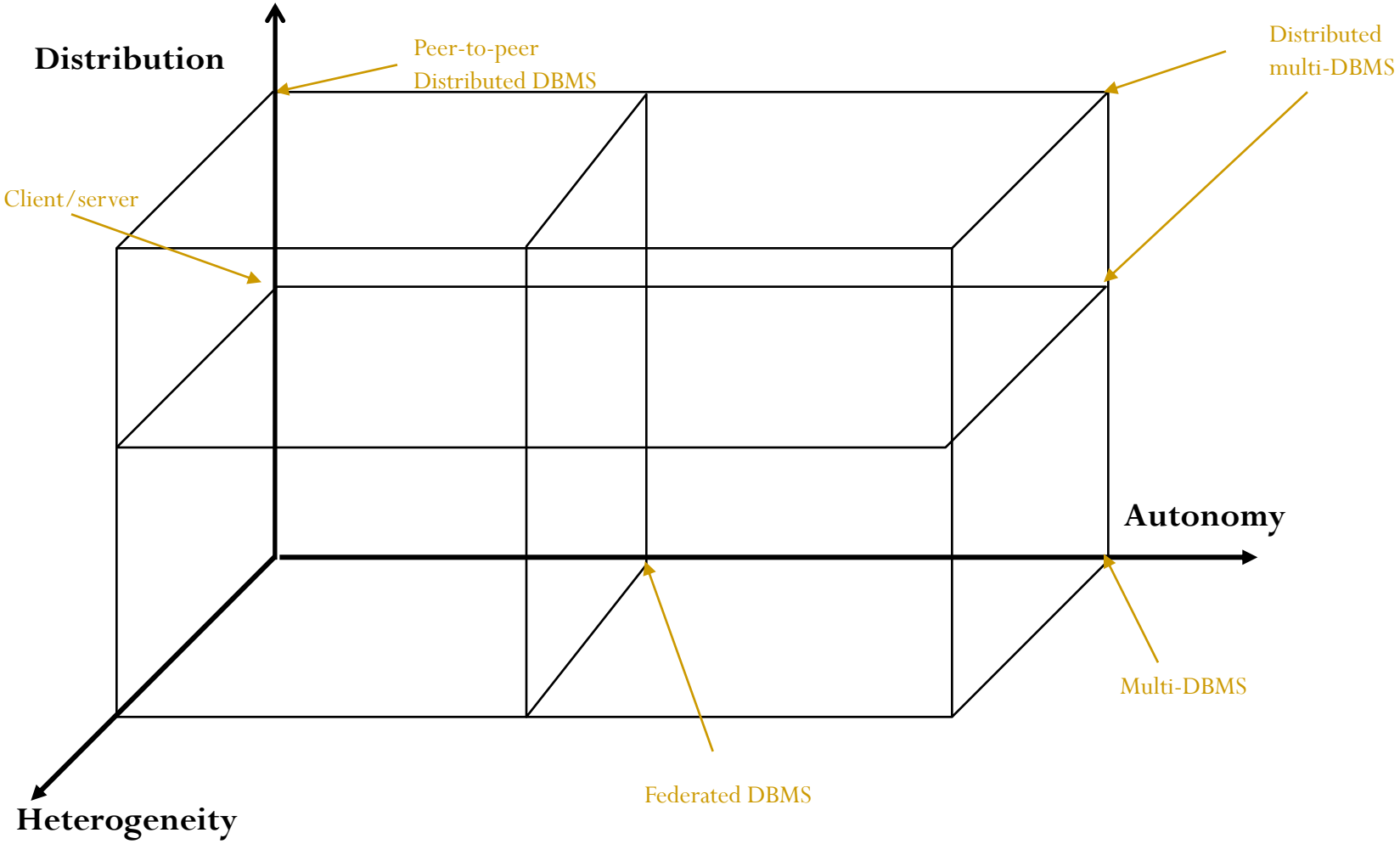
- Defines the structure of the system
  - Components identified
  - Functions of each component defined
  - Interrelationships and interactions between components defined
- Applies both for computer systems as well as for software systems
- There is a close relationship between the architecture of a system, standardization efforts, and a reference model

ANSI/SPARC Architecture

- ANSI-SPARC stands for ***American National Standards Institute, Standards Planning And Requirements Committee***



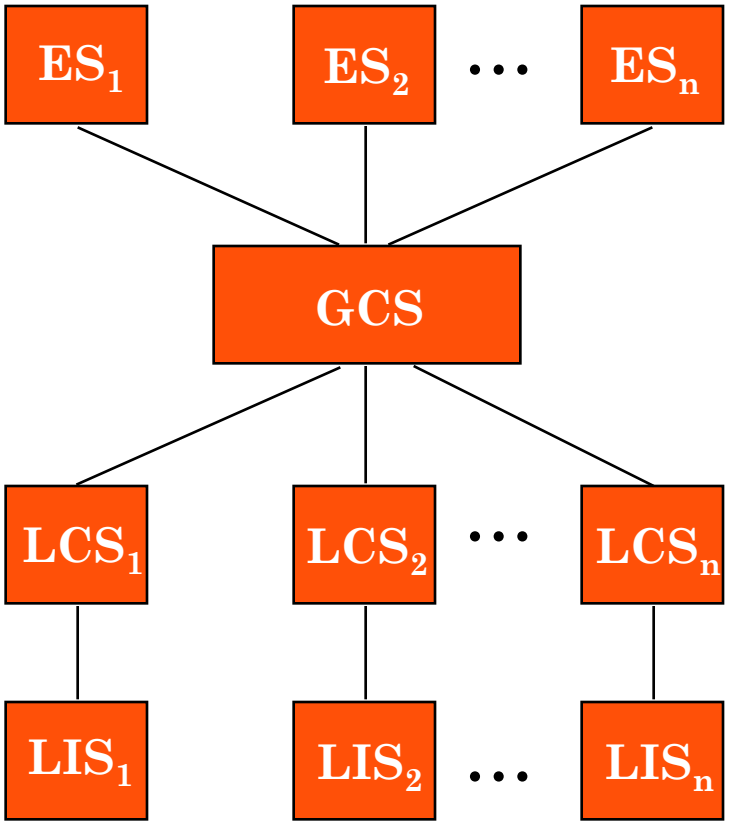
DDBS Architecture Models



Architectural Dimensions

- Distribution
  - Whether the components of the system are located on the same machine or not
- Heterogeneity
  - Various levels (hardware, communications, operating system)
  - DBMS important one data model, query language, transaction management algorithms
- Autonomy
  - Design autonomy: each individual DBMS is free to use the data models and transaction management techniques that it prefers.
  - Communication autonomy: each individual DBMS is free to decide what information to provide to the other DBMSs
  - Execution autonomy: each individual DBMS can execute the transactions that are submitted to it in any way that it wants to.

DDBS Peer-to-Peer Architecture



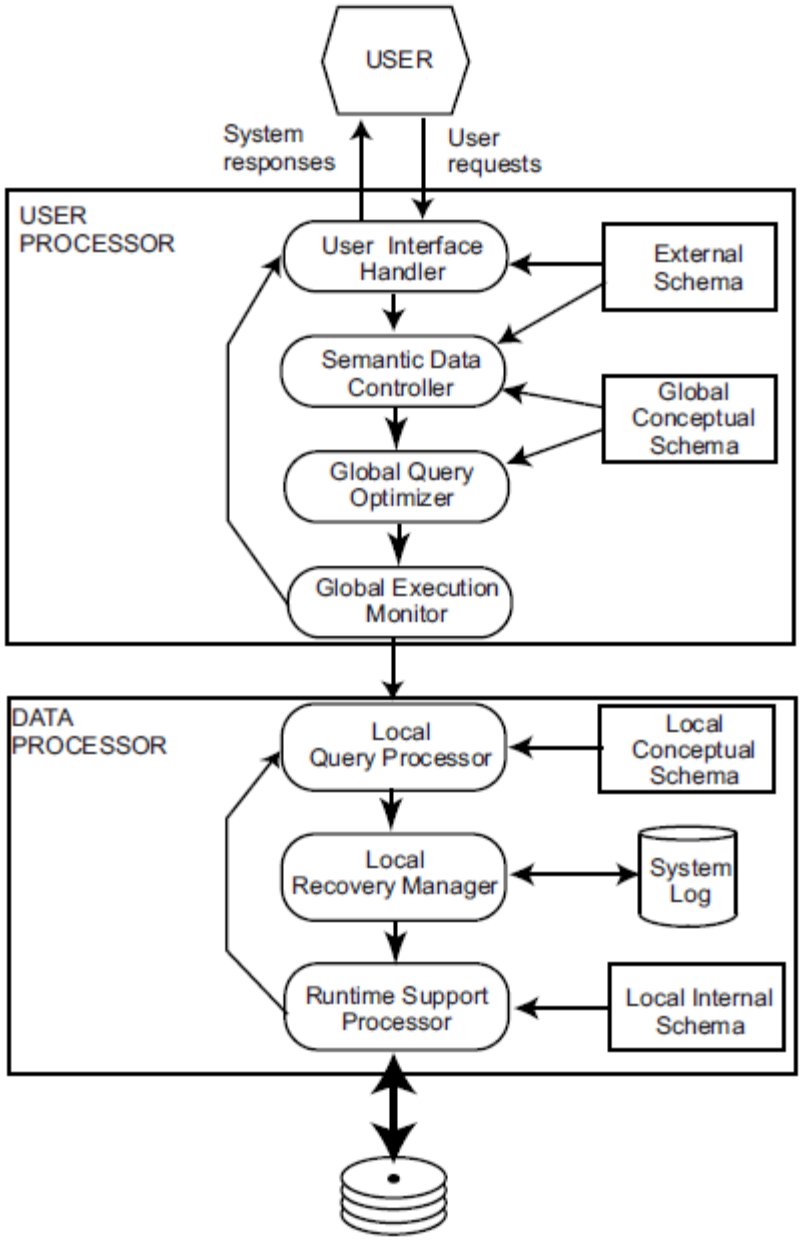
ES: External Schema

GCS: Global Conceptual Schema

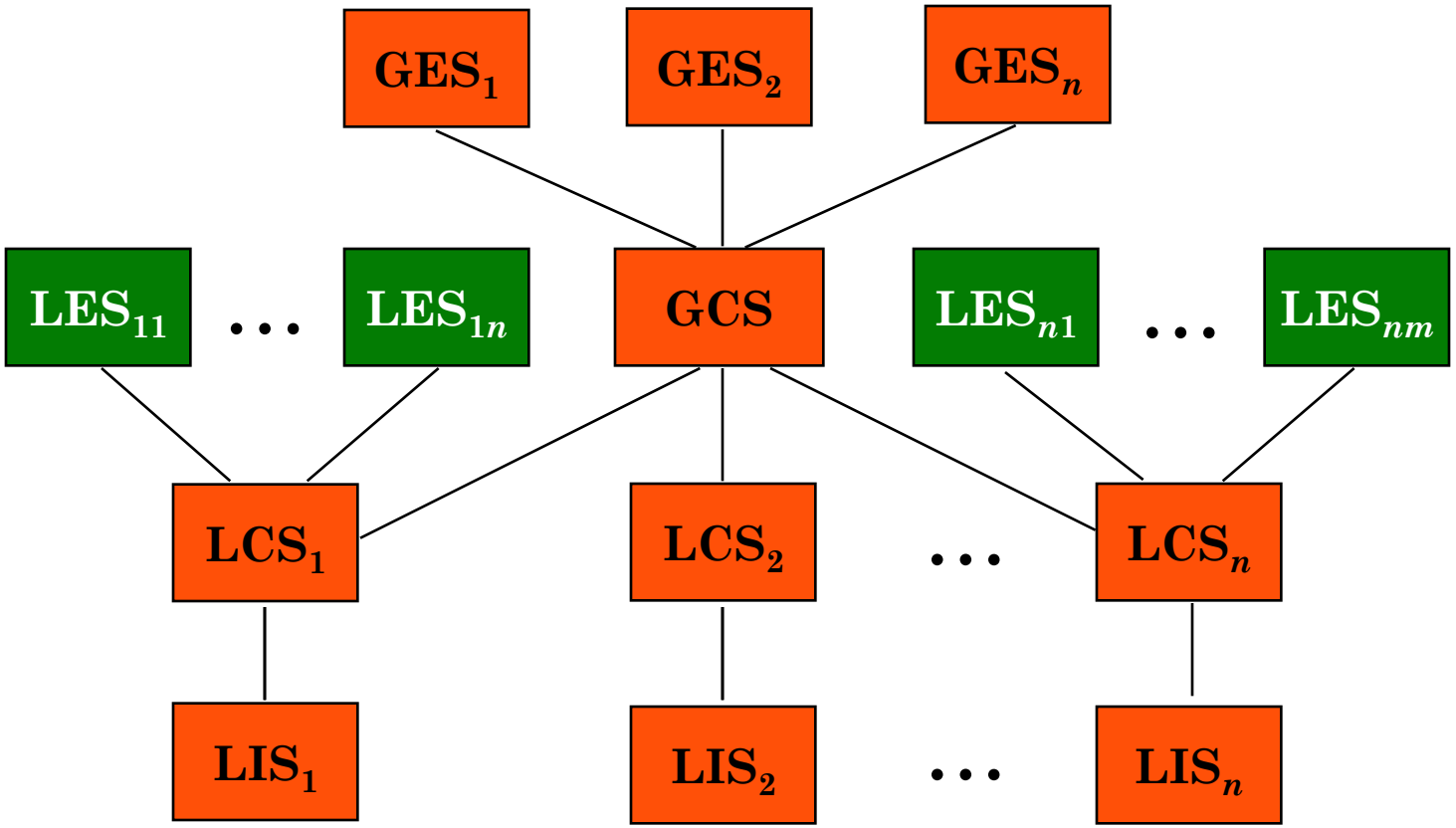
LCS: Local Conceptual Schema

LIS: Local Internal Schema

Component Peer-to-Peer Architecture

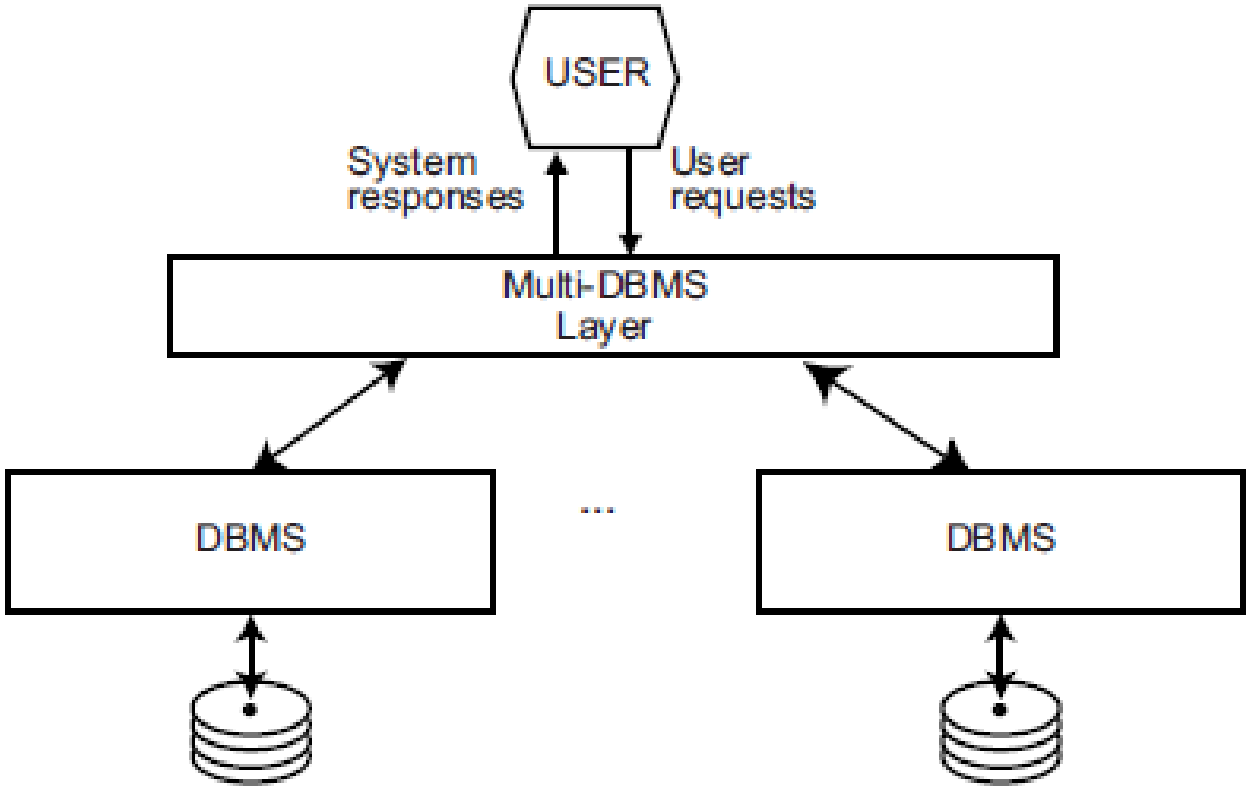


DDBMS Architecture (Data-based)



- GES: Global External Schema
- LES: Local External Schema
- LCS: Local Conceptual Schema
- LIS: Local Internal Schema

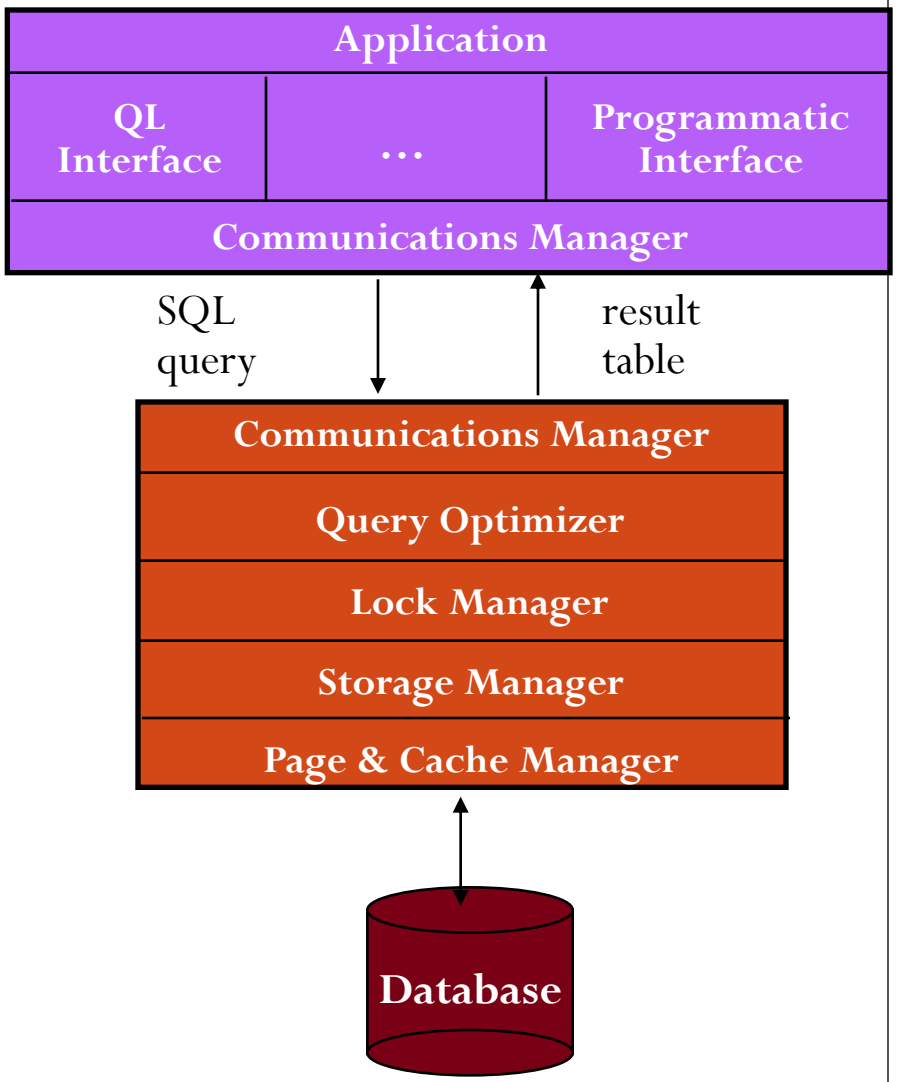
DDBMS Architecture (Component-based)



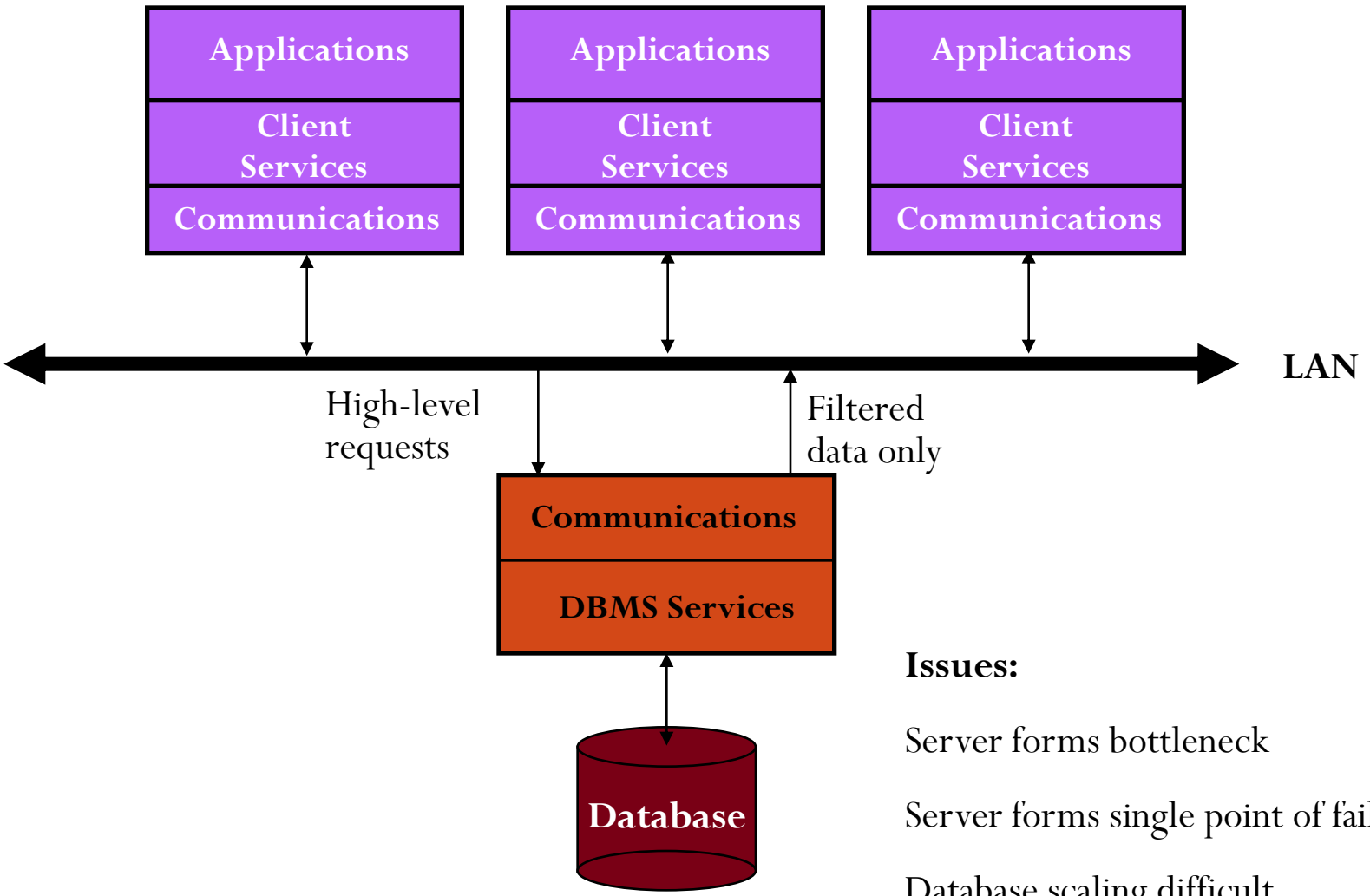
Client/Server Architecture (Data-based)

General idea: Divide the functionality into two classes:

- server functions
  - mainly data management, including query processing, optimization, transaction management, etc.
- client functions
  - might also include some data management functions (consistency checking, transaction management, etc.) not just user interface
- Provides a two-level architecture
- More efficient division of work
- Different types of client/server architecture:
  - Multiple client/single server
  - Multiple client/multiple server



Multiclients / Single Server



Issues:

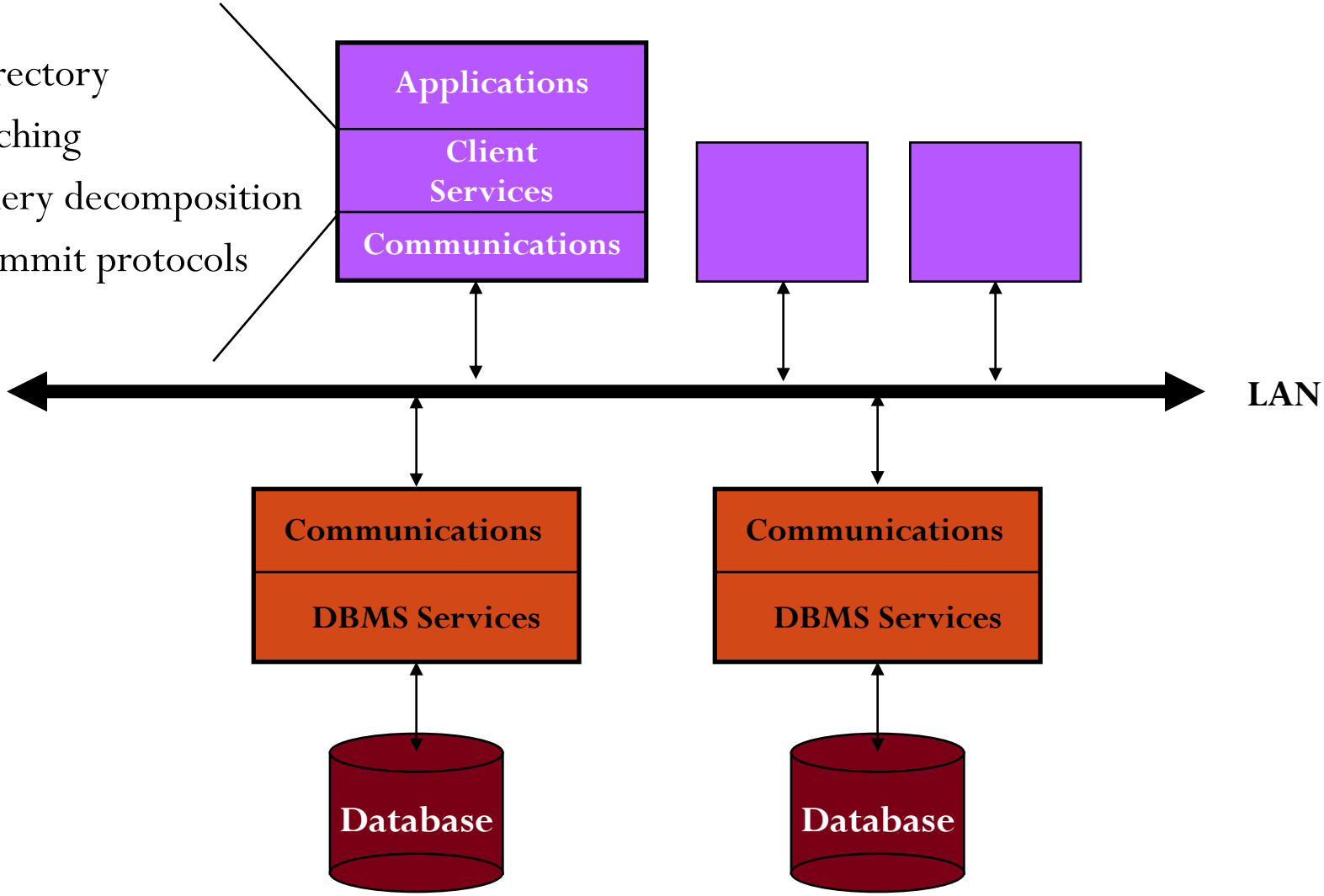
Server forms bottleneck

Server forms single point of failure

Database scaling difficult

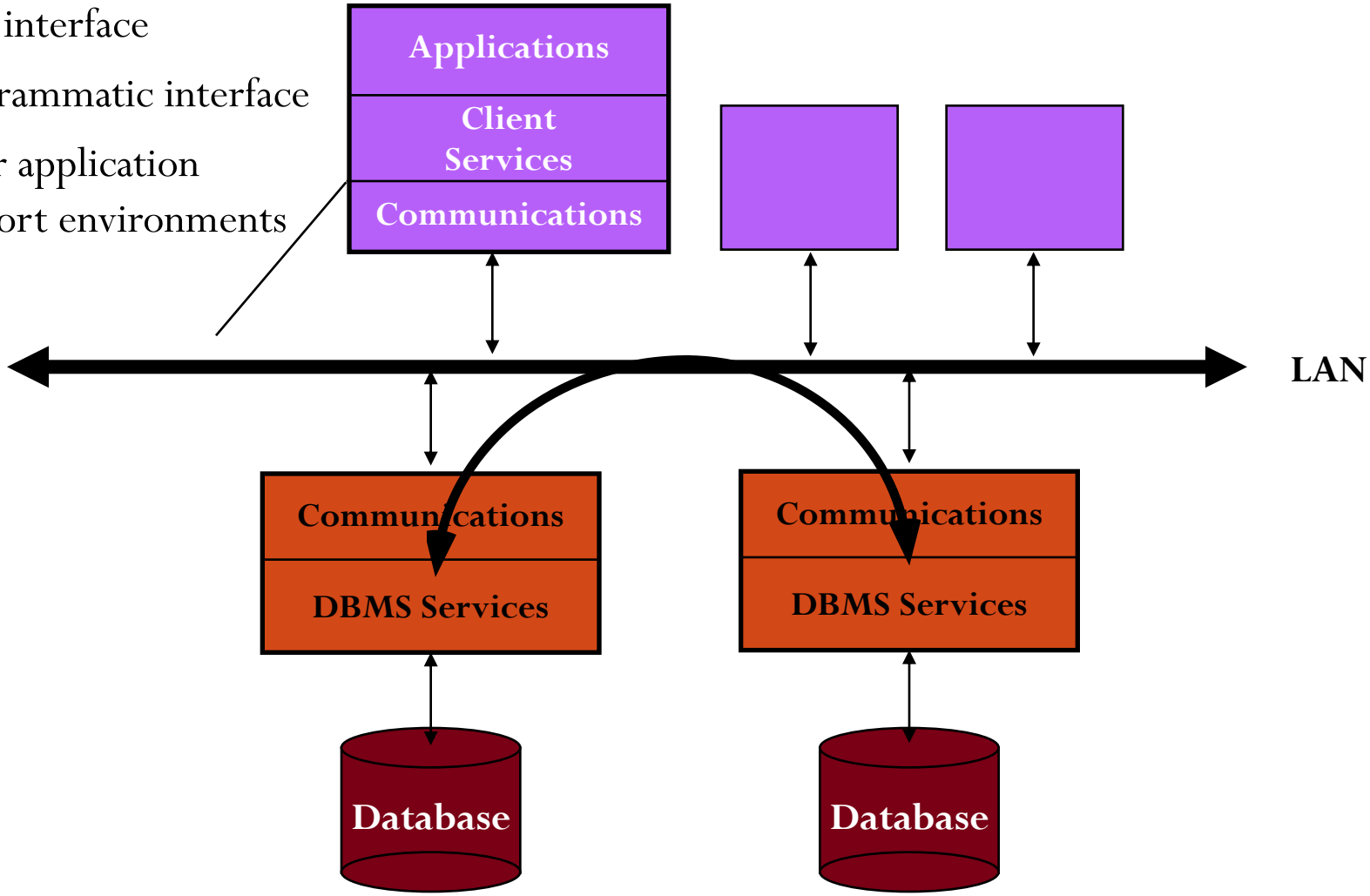
Multi Clients/ Multi Servers (1)

- directory
- caching
- query decomposition
- commit protocols



Multi Clients/ Multi Servers (2)

- SQL interface
- programmatic interface
- other application support environments



Advantages of Client/Server Architecture

- More efficient division of labor
- Horizontal and vertical scaling of resources
- Better price/performance on client machines
- Ability to use familiar tools on client machines
- Client access to remote data (via standards)
- Full DBMS functionality provided to client workstations
- Overall better system price/performance

Summarize

- General information of distributed database
- Development of DDB
- Research issues in DDB
- System architecture
- There are three orthogonal implementation dimensions for DDBS: level of distribution, autonomity, and heterogeinity
- Different architectures
  - Client-Server Systems
  - Peer-to-Peer Systems