Distributed Data Processing

- Introduction
- **Distributed DBMS Architecture**
- Distributed DB Design
- Query Processing
- Transaction Management

2. Distributed DBMS Architecture

- DBMS Standardization
- Architectural Models for DDBMS
- Distributed DBMS Architecture
- Global Directory Issues
- Conclusion

2.1 DBMS Standardization

Important events in DB

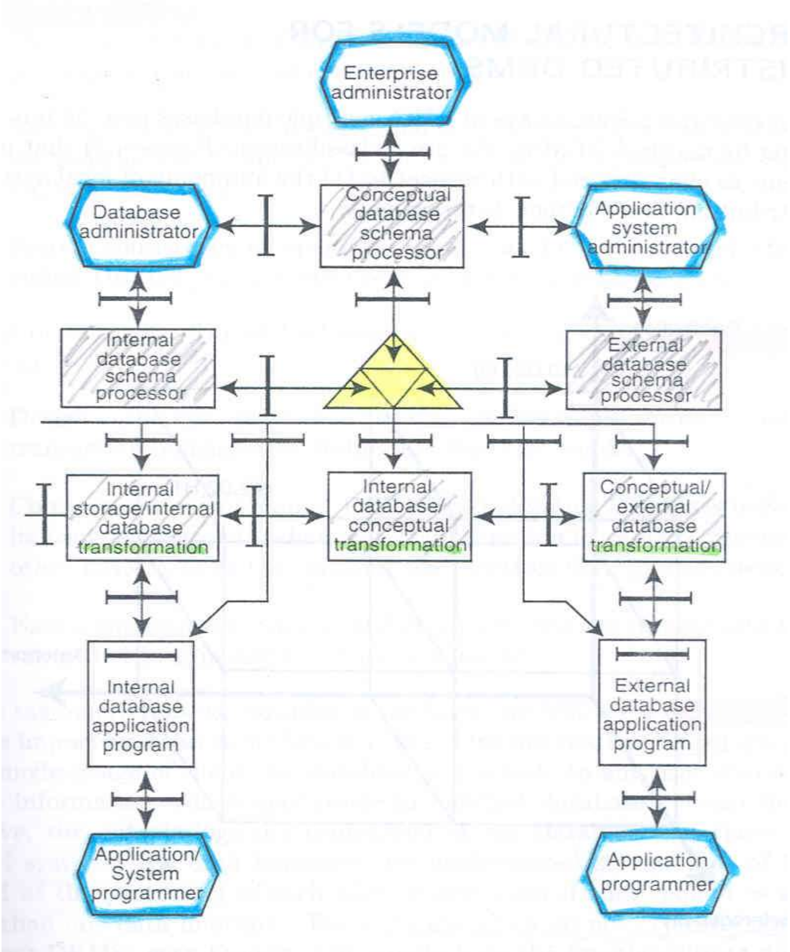
- IBM
  - IMS – hierarchical DB (1969)
- DBTG of CODASYL
  - Survey of DBMS (1969)
  - Network DBMS (1971)
- E. F. Codd
  - Relational Data model (1970)

CODASYL: Conference On Data SYstem Language
DBTG: Database Task Group

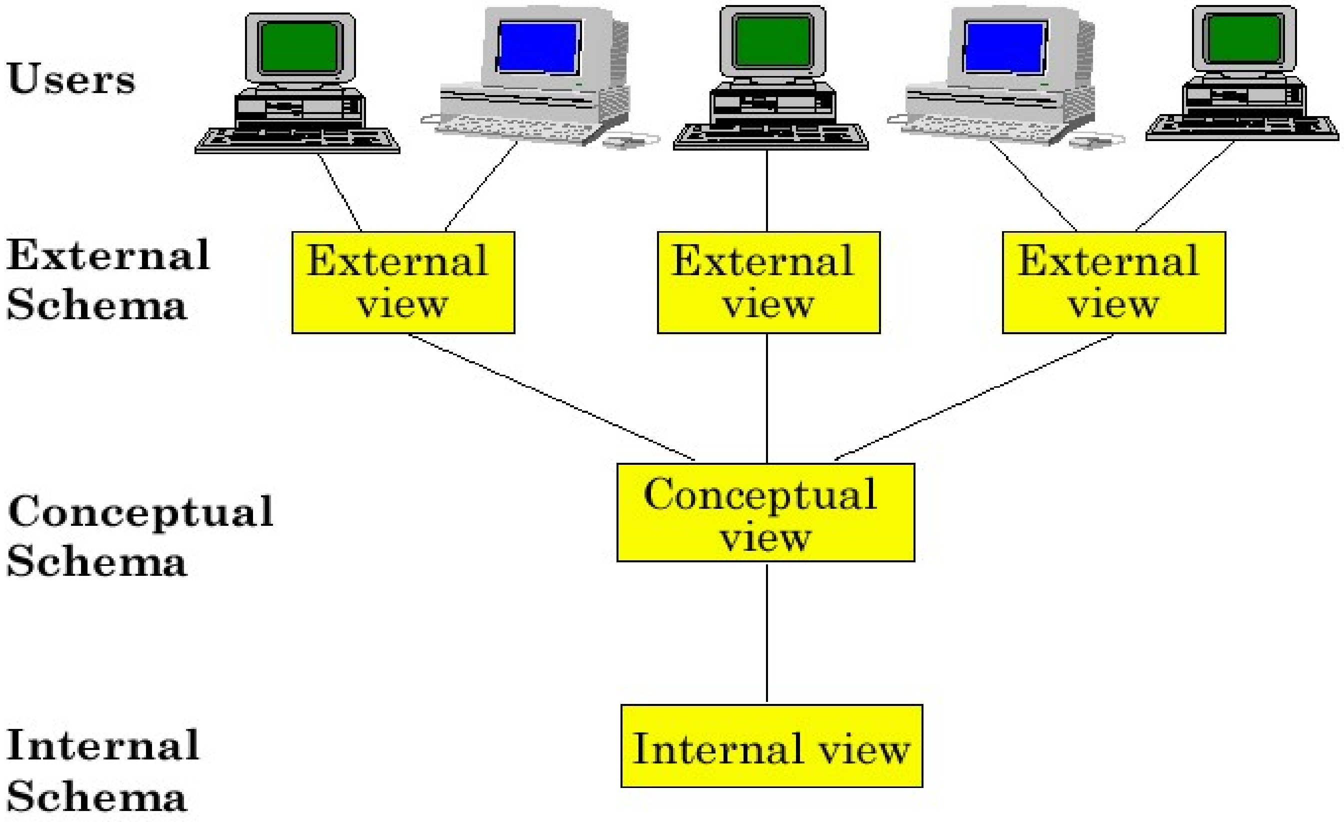
Standardization

- **In late 1972,** the computer and Information Processing Committee (X3) of the American National Standards Institute (ANSI) established a Study Group on Database Management Systems under the auspices of its Standards Planning and Requirements Committee (SPARC)
- **Mission:** to study the **feasibility** of setting up standards in this area, as well as determining which aspects should be **standardized** if it was feasible.
- **Interim report 1975 & Final report 1977**
  - **ANSI/SPARC architecture – containing 43 interfaces (ANSI/X3/SPARC DBMS Framework)**

Partial Schematic of the ANSI/SPARC Architectural Model



ANSI/SPARC Architecture



Example

Conceptual Schema Definition

```
RELATION EMP [
    KEY = {ENO}
    ATTRIBUTES = {
        ENO     : CHARACTER(9)
        ENAME   : CHARACTER(15)
        TITLE   : CHARACTER(10)
    }
]
RELATION PAY [
    KEY = {TITLE}
    ATTRIBUTES = {
        TITLE   : CHARACTER(10)
        SAL     : NUMERIC(6)
    }
]
```

Conceptual Schema Definition

```
RELATION PROJ [
    KEY = {PNO}
    ATTRIBUTES = {
        PNO     : CHARACTER(7)
        PNAME   : CHARACTER(20)
        BUDGET  : NUMERIC(7)
    }
]
RELATION ASG [
    KEY = {ENO,PNO}
    ATTRIBUTES = {
        ENO     : CHARACTER(9)
        PNO     : CHARACTER(7)
        RESP    : CHARACTER(10)
        DUR     : NUMERIC(3)
    }
]
```

Internal Schema Definition

```
RELATION EMP [
    KEY = {ENO}
    ATTRIBUTES = {
        ENO      : CHARACTER(9)
        ENAME    : CHARACTER(15)
        TITLE    : CHARACTER(10)
    }
]


INTERNAL_REL EMPL [
    INDEX ON E# CALL EMINX
    FIELD = {
        HEADER   : BYTE(1)
        E#       : BYTE(9)
        ENAME    : BYTE(15)
        TIT      : BYTE(10)
    }
]
```

External View Definition – Example 1

Create a BUDGET view from the PROJ relation

```
CREATE  VIEW    BUDGET(PNAME, BUD)
 AS     SELECT  PNAME, BUDGET
        FROM    PROJ
```

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

External View Definition – Example 2

Create a Payroll view from relations EMP and TITLE_SALARY

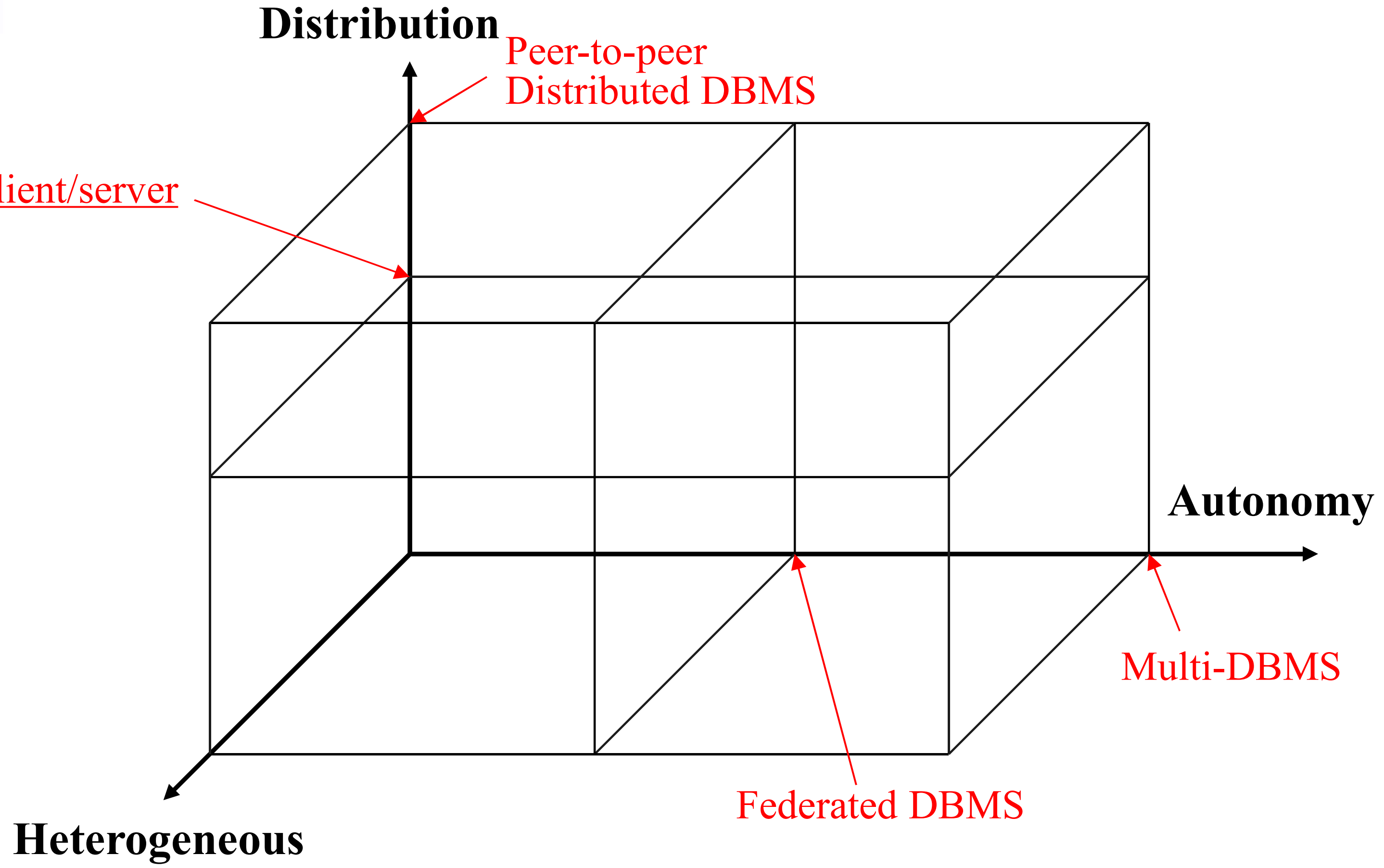
```
CREATE   VIEW    PAYROLL (ENO, ENAME, SAL)
AS       SELECT  EMP.ENO,EMP.ENAME,PAY.SAL
         FROM    EMP, PAY
         WHERE   EMP.TITLE = PAY.TITLE
```

2.2 Architectural Models for DDBMS

Dimensions of the Problem

- Distribution
- Heterogeneity
- Autonomy

DBMS Implementation Alternatives



Distribution

- Whether the components of the system are located on the same machine or not
- Classification:
  - Non-distributed
  - Client/server
  - Peer-to-peer

Heterogeneity

- In general
  - hardware
  - communications
  - operating system
- DBMS important one – data management
  - data model
  - query language
  - transaction management algorithms
- Classification
  - Homogeneous
  - heterogeneous

Autonomy

- Refers to the distribution of control – indicates the degree to which individual DBMSs can operate independently
- Not well understood and most troublesome

- Requirements
- Dimension
- Classification

Requirements

- Local
  - The local operations of the individual DBMSs are not affected by their participation in the multidatabase system
- Manner
  - The manner in which the individual DBMSs process queries and optimize them should not be affected by the execution of global queries that access multiple databases
- Join/leave
  - System consistency or operation should not be compromised when individual DBMSs join or leave the multidatabase confederation

Dimension

- **Design autonomy**: Ability of a component DBMS to decide on issues related to its own design.
- **Communication autonomy**: Ability of a component DBMS to decide whether and how to communicate with other DBMSs.
- **Execution autonomy**: Ability of a component DBMS to execute local operations in any manner it wants to.

Classification

- Tight integration
  - A single-image of the entire database is available to any user who wants to share the information, which may reside in multiple databases
- Semiautonomous
  - The individual systems can (and usually do) operation independently, but have decided to participate in a federation to make their local data sharable
- Total isolation
  - The individual systems are stand-alone DBMSs, which know neither of the existence of other DBMSs nor how to communicate with them

2.3 Distributed DBMS Architecture

Architecture

- Three reference architecture
  - client/server system
    - (Ax,D1,Hy)
  - Peer-to-peer distributed DBMS
    - (A0,D2,H0)
  - multidatabase system
    - (A2,Dx,Dy)

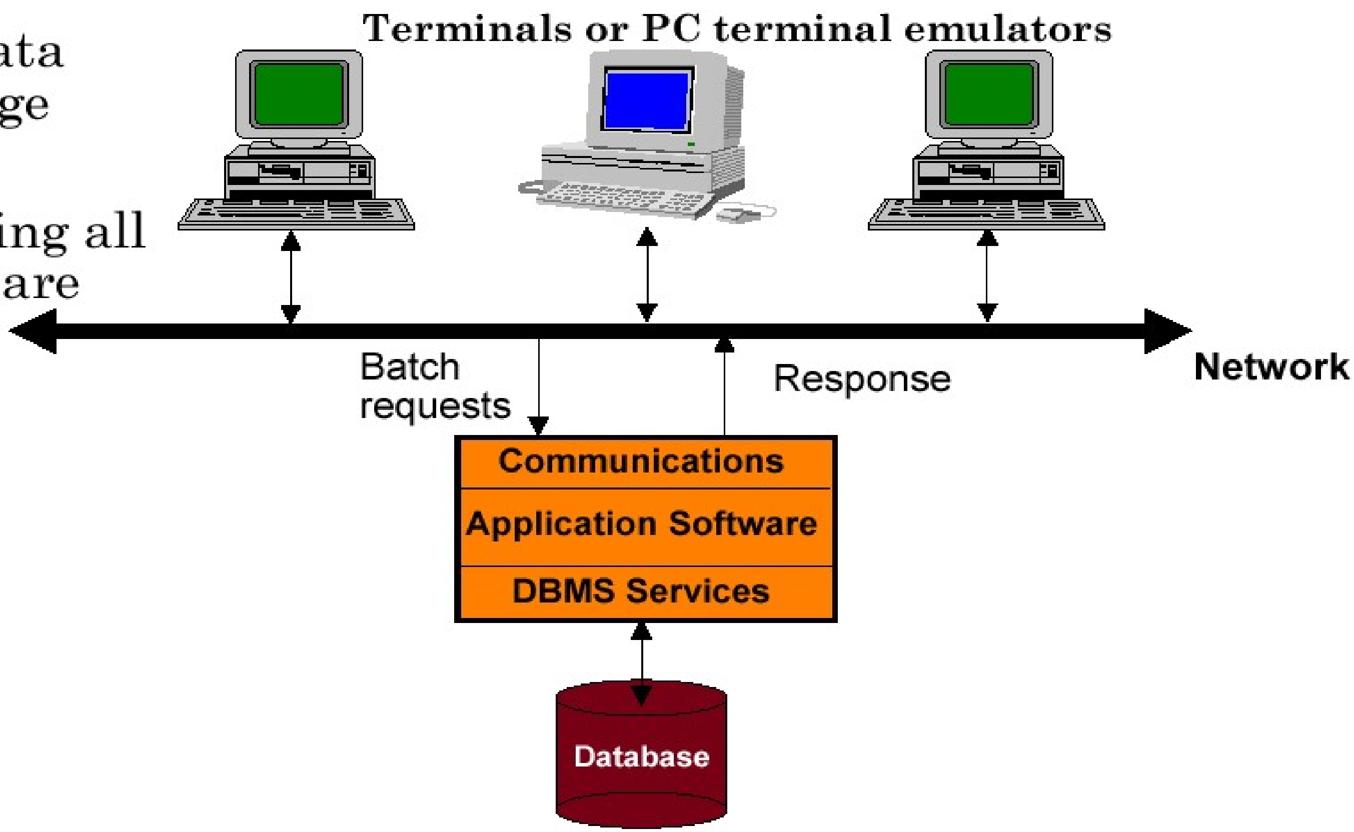
2.3.1 Client/Server System

(Ax,D1,Hy)

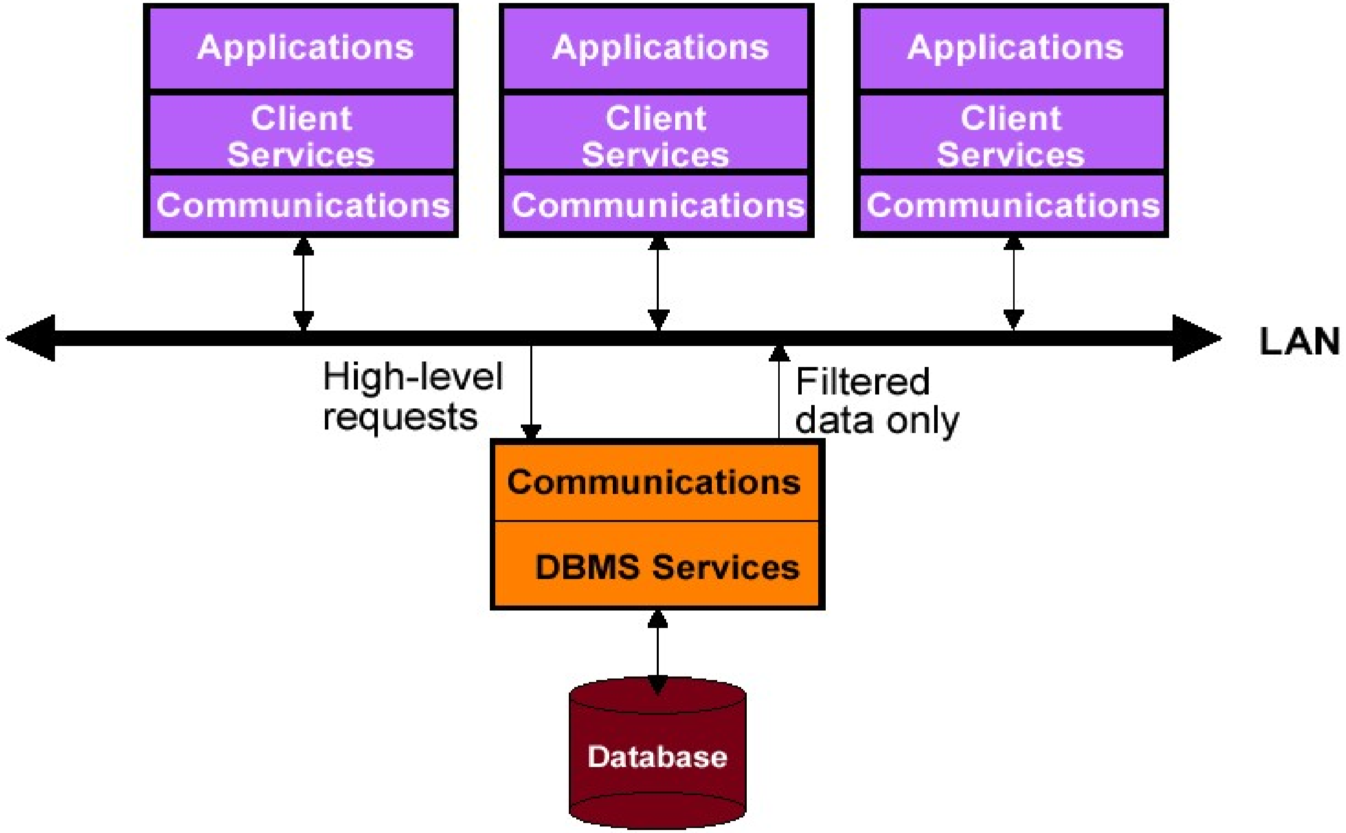
- Client/Server vs. peer-to-peer
  - Architectural paradigm

Timesharing Access to a Central Database

- No data storage
- Host running all software



Multiple Clients/Single Server



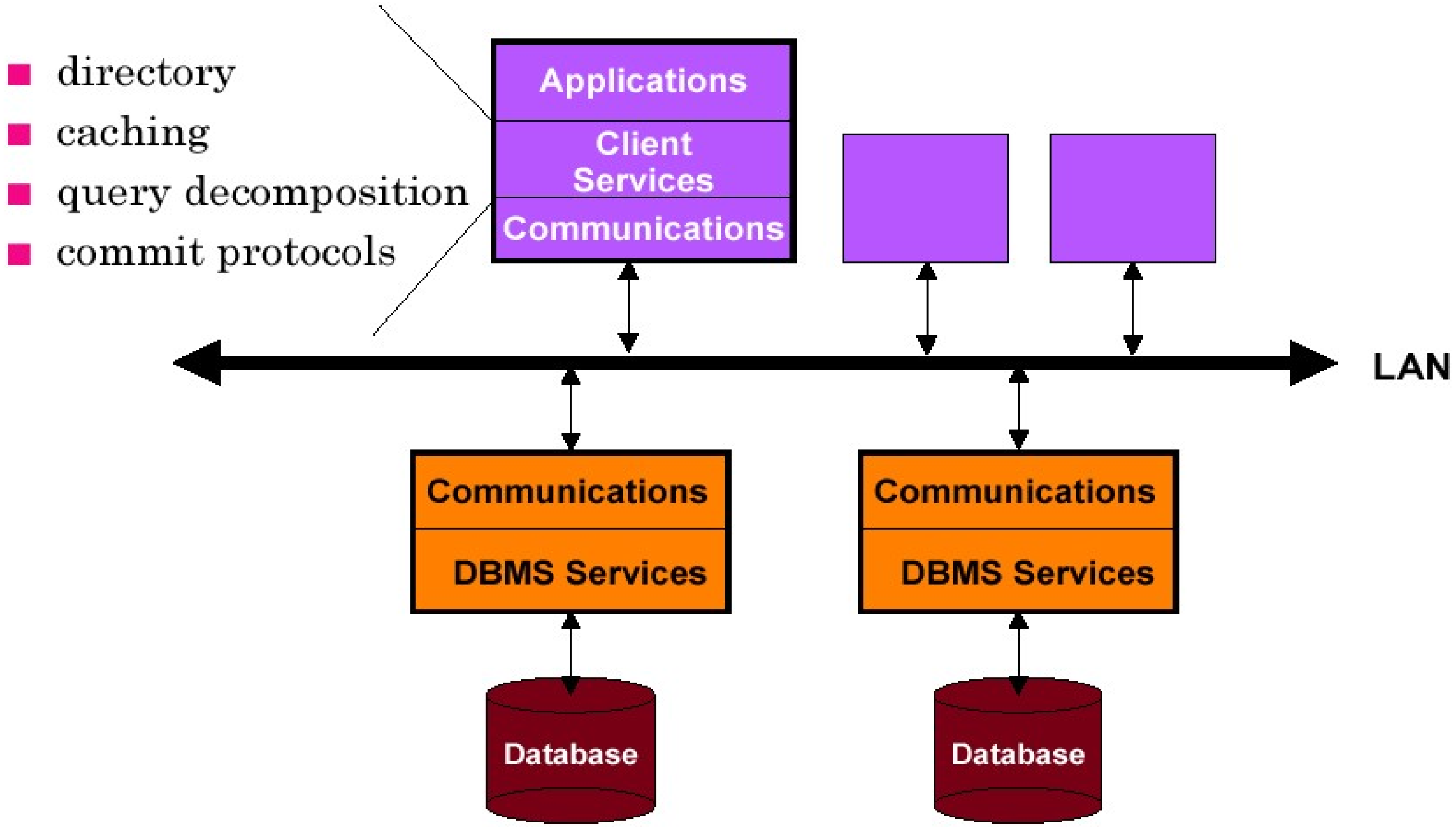
Advantages of Client-Server Architectures

- 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

Problems With Multiple- Client/Single Server

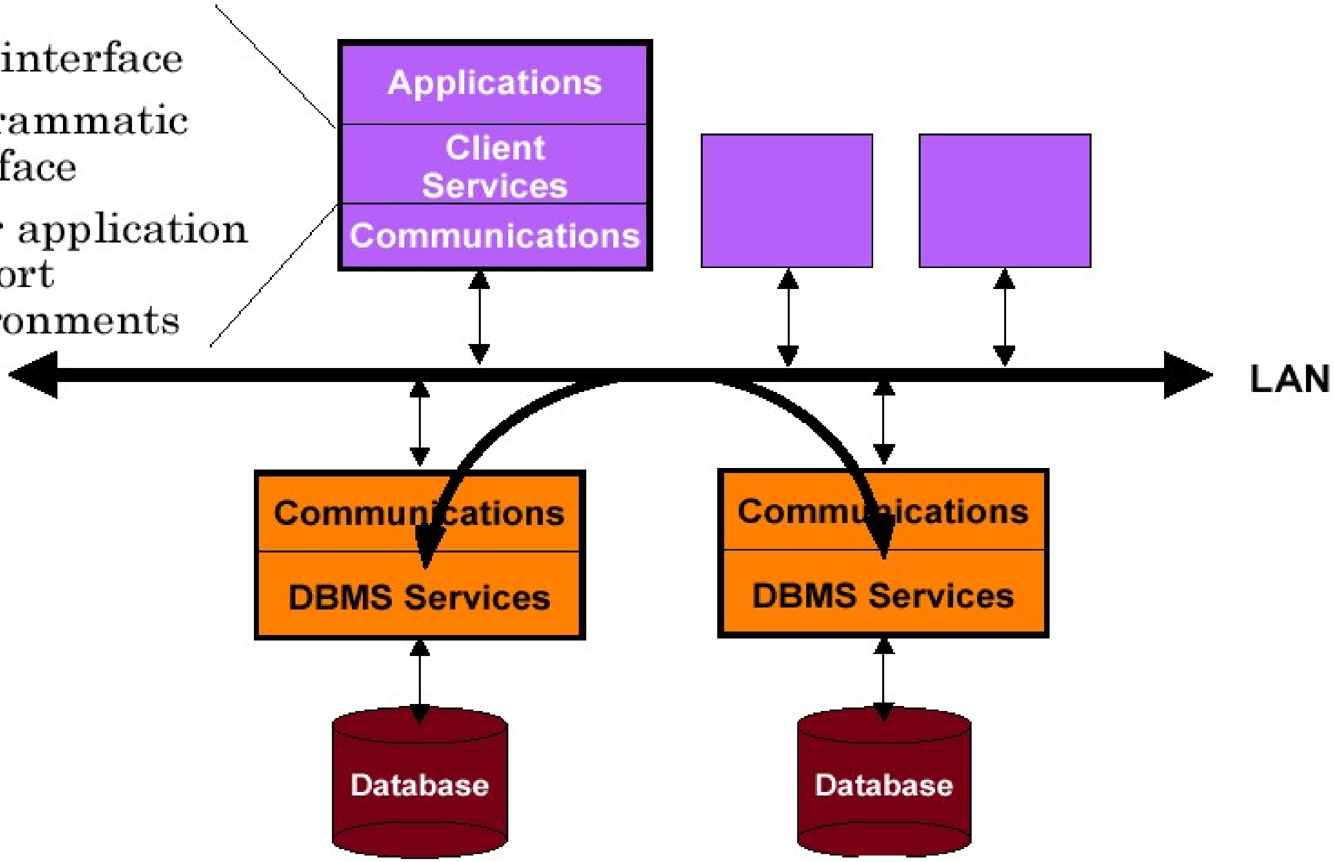
- Server forms bottleneck
- Server forms single point of failure
- Database scaling difficult

Multiple Clients/Multiple Servers



Server-to-Server

- SQL interface
- programmatic interface
- other application support environments



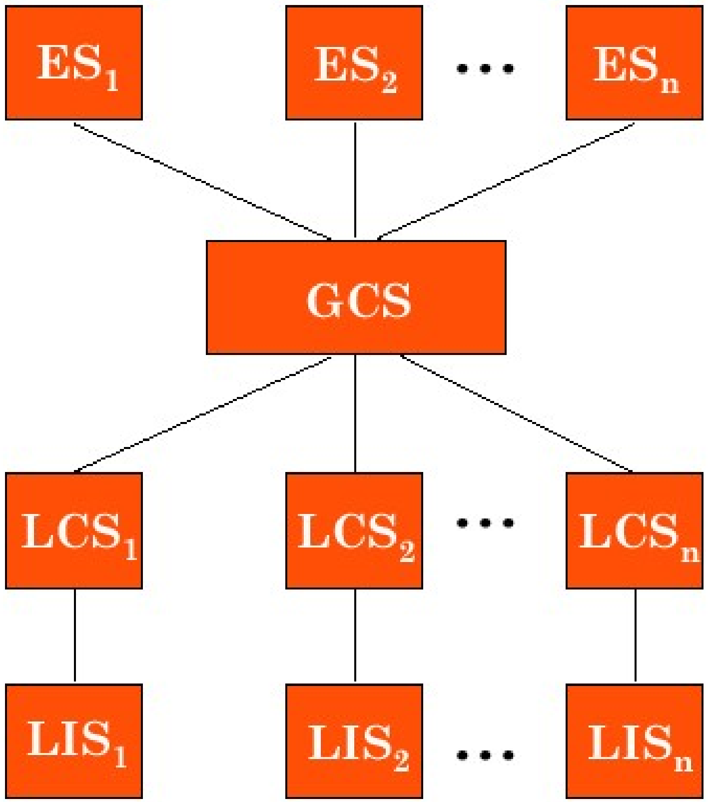
2.3.2 Distributed DBMS Architecture

(A0,D2,H0)

Peer-to-peer distributed DBMS

- LIS: **Local Internal Schema**
- GCS: **Global Conceptual Schema**
- LCS: **Local Conceptual Schema**
- ES: **External Schema**

Distributed DBMS Architecture



GCS -> LCS

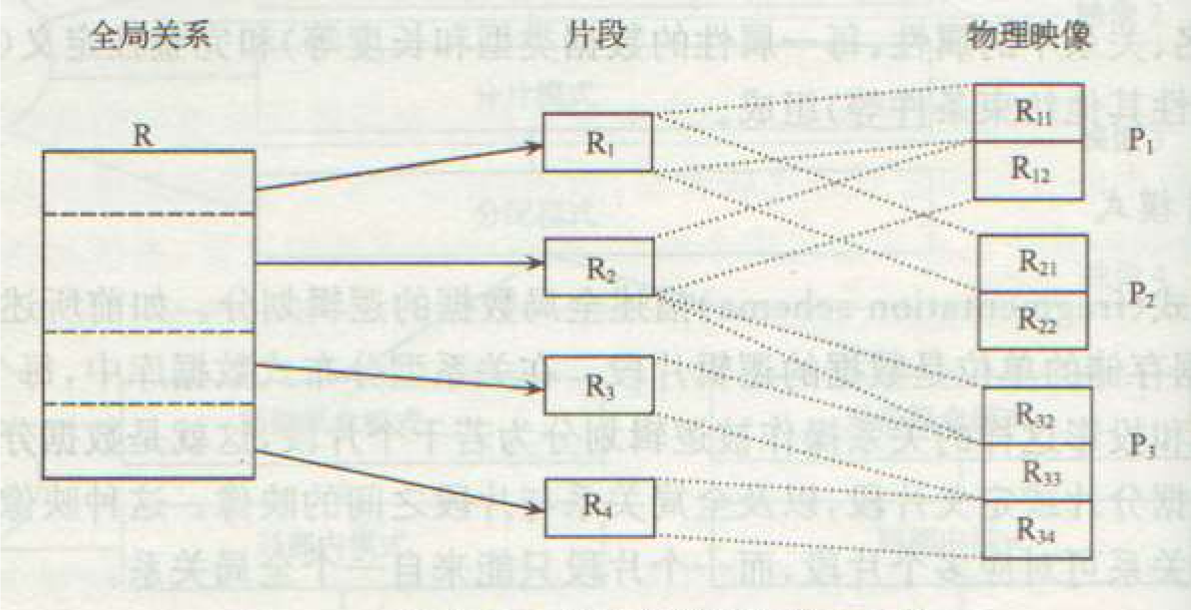
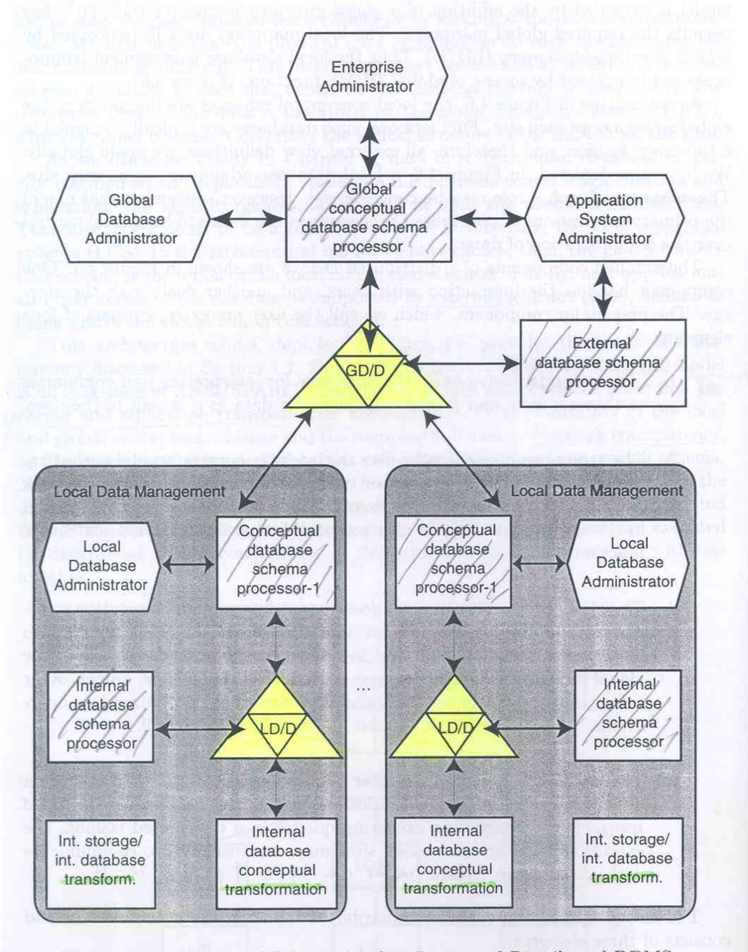


图 1.4　全局关系 R 的逻辑片段与物理映像

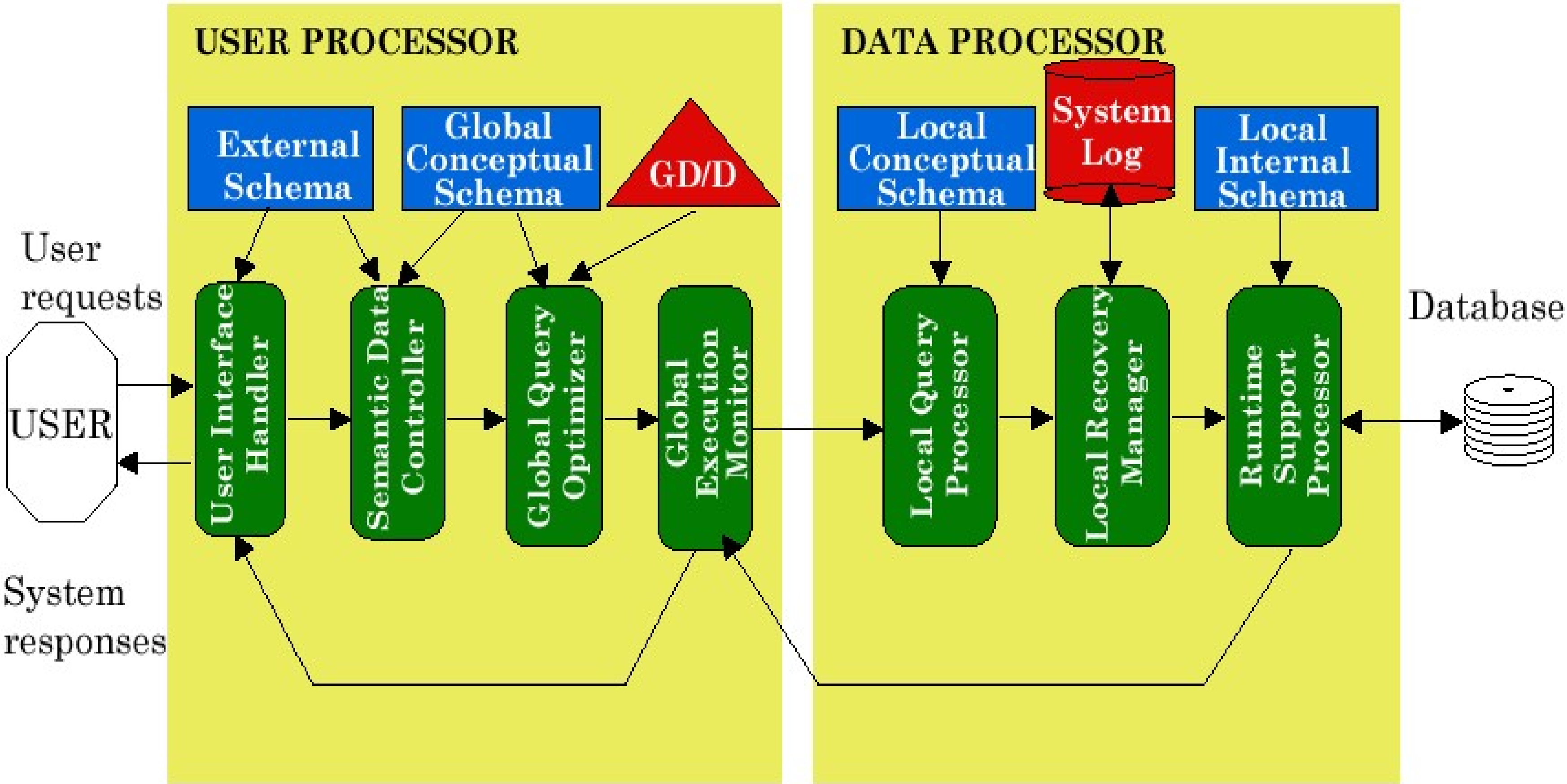
Functional Schematic of an Integrated DDBMS



Some Issues on DDBMS

- About transparent support
  - Data independence
  - Fragmentation and Replication transparencies
  - Network transparency
- Global Directory/Dictionary (GD/D)
  - Global mapping
- Components of a DDBMS
  - User processor: to deal with interaction with users
  - Data processor: to deal with storage

Peer-to-Peer Component Architecture



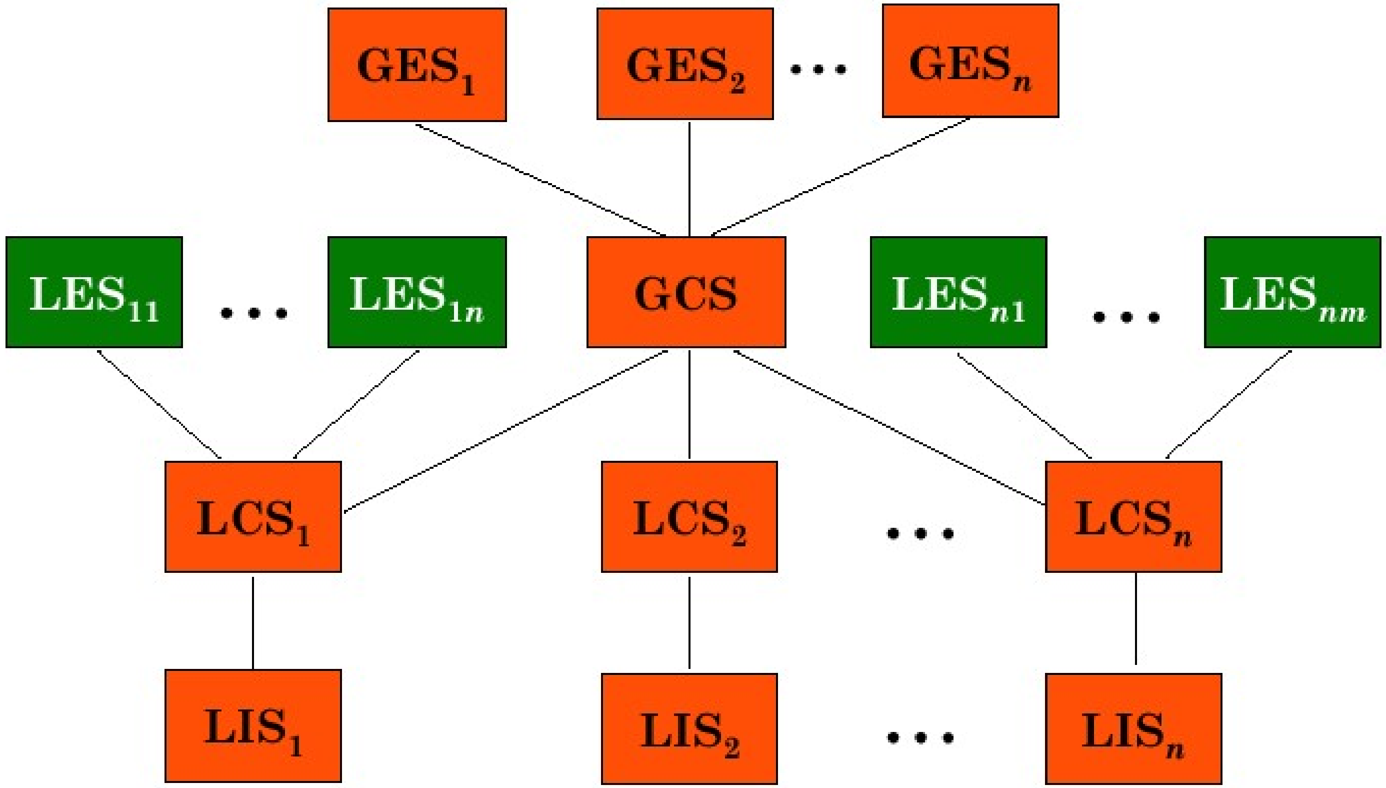
2.3.3 Multi-DBMS

(A2,Dx,Dy)

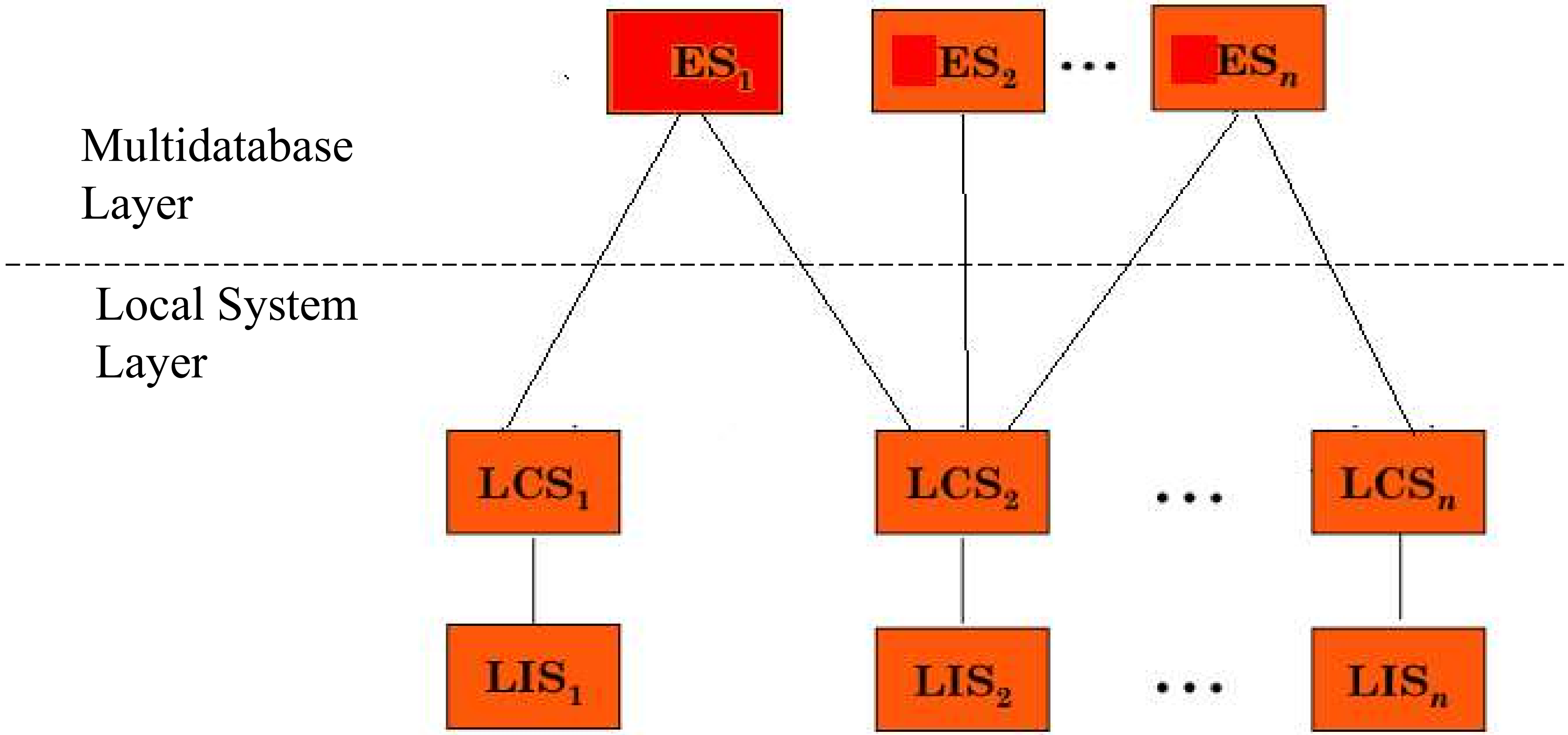
- Multi-DBMS vs Distributed DBMS
  - Autonomy

The differences in the level of autonomy are also reflected in their architectural models: the fundamental difference relates to the definition of the global conceptual schema

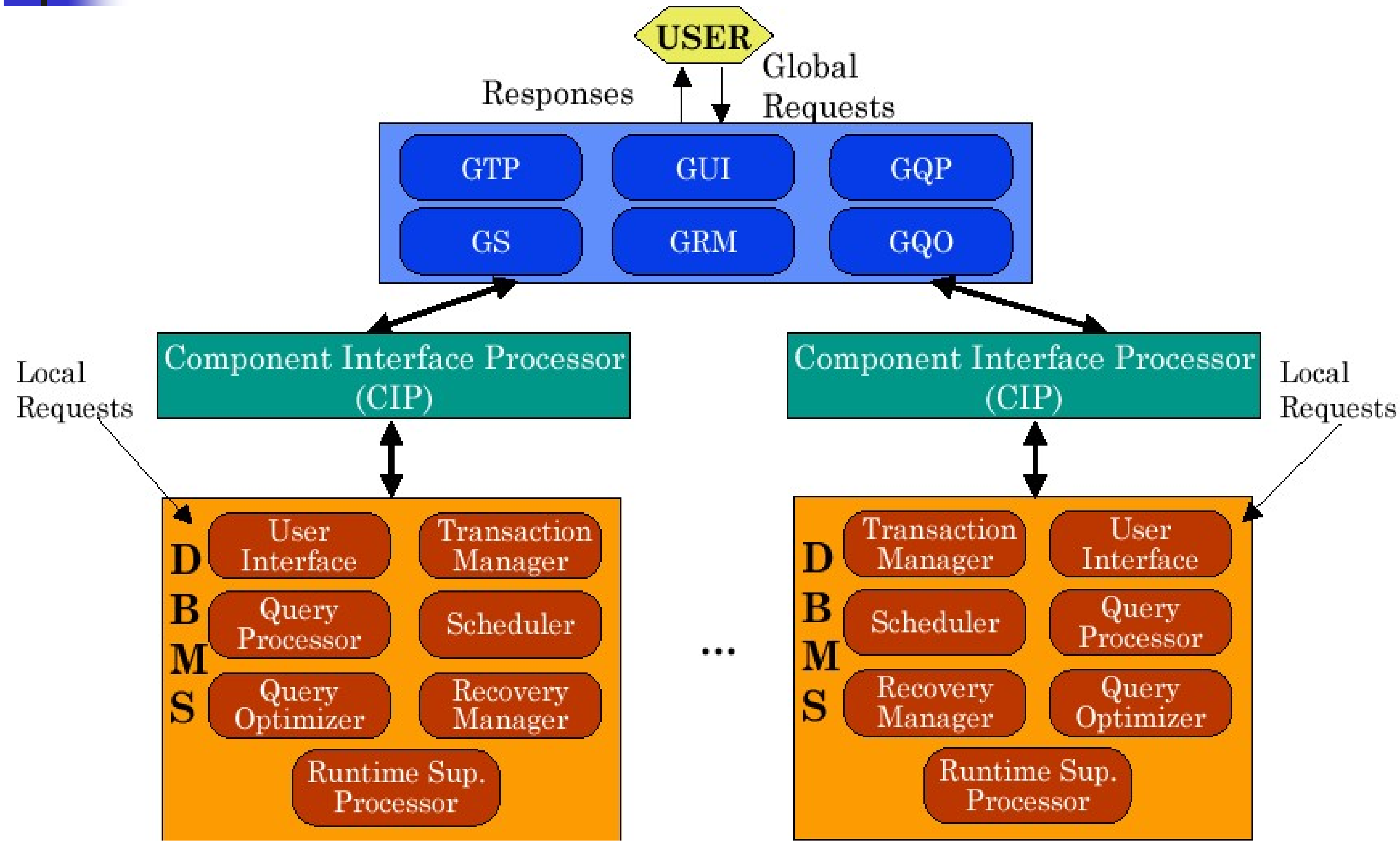
Multi-DBMS Architecture with GCS



Multi-DBMS Architecture without GCS



Components of a Multi-DBMS

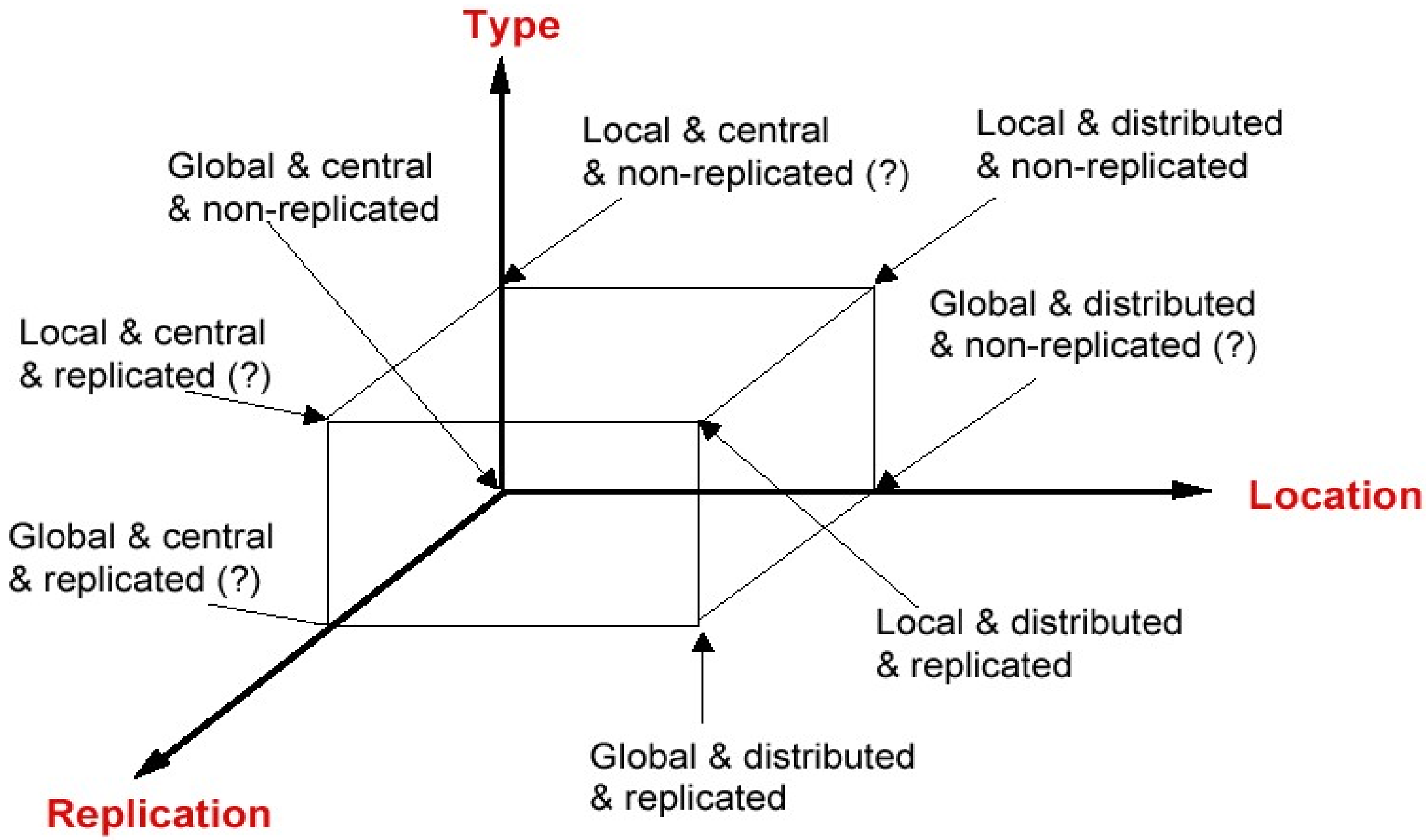


2.4 Global Directory Issues

Global Directory Issues

- If the global directory exists, it is an ***extension*** of the directory as described in the ANSI/SPARC report.
- It includes information about the location of the fragments as well as the makeup of the fragments
- The directory is itself a database that contains ***meta-data*** about the actual data stored in the database

Alternative Directory Management Strategies



2.5 Conclusion

Conclusion

- Heterogeneity
- Distribution
  - Client/Server
  - Peer-to-peer
- Autonomy
  - Multi-DBS
  - Federated DBS
  - Distributed DBS

References

- D. Tsichritzis and A. Klug. The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems. Inf. Syst. (1978), 1: 173-191
- C. Mohan and R. T. Yeh. Distributed Data Base Systems: A Framework for Data Base Design. In Distributed Data Bases, Infotech State-of-the-Art Report, London: Infotech, 1978
- F. Schreiber. A Framework for Distributed Database Systems. In Proc. Int. Computing Symposium, 1977, 475-482
- M. Adiba, J. C. Chupin, R. Demolombe, et al., Issues in Distributed Data Base Management Systems: A Technical Overview. In Proc. 4th Int. Conf. on Very Large Data Bases, Sept. 1978, 89-110.