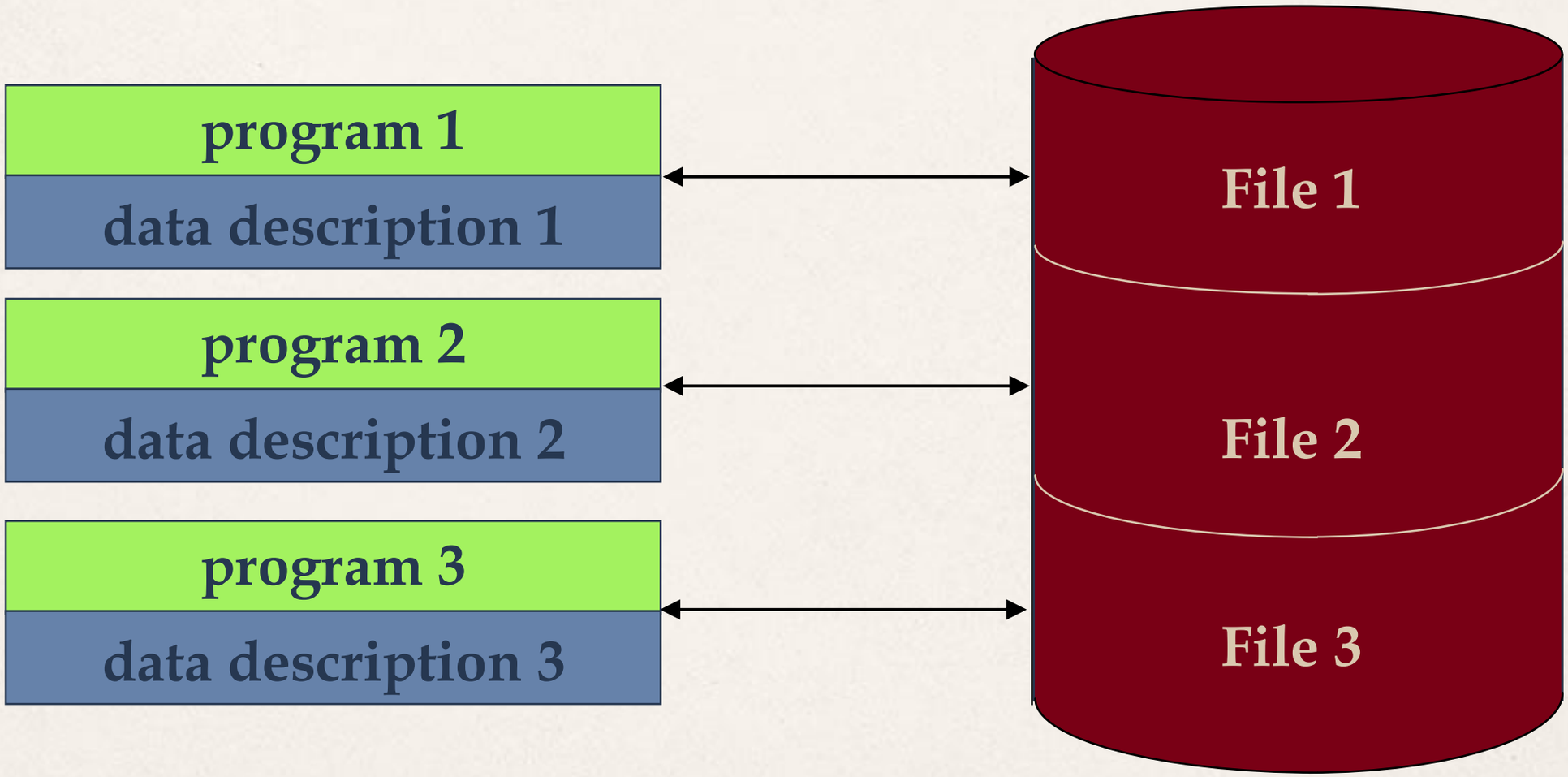
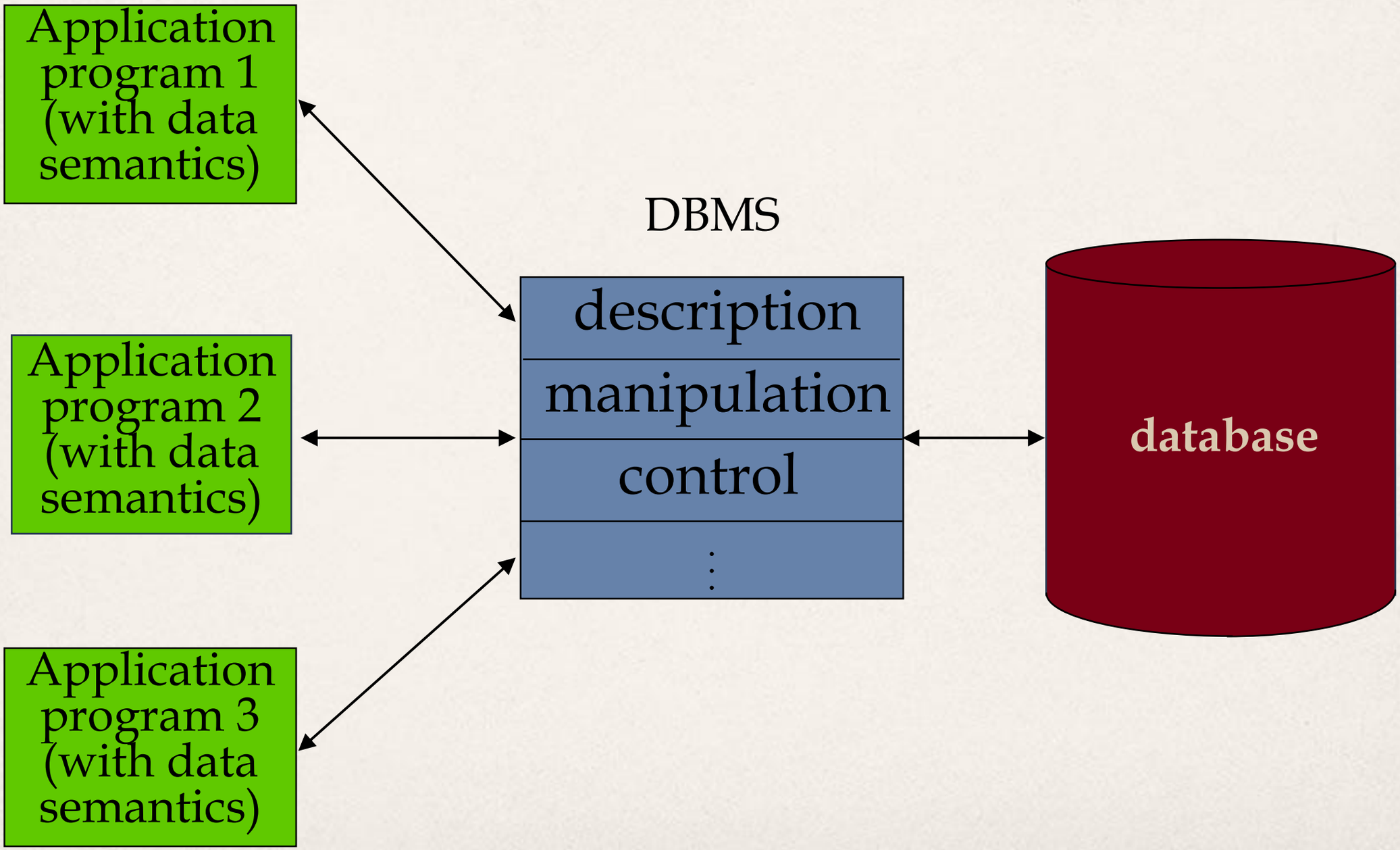
Distributed DBMS Architecture

File Systems



Database Management



What is a Distributed Database System?

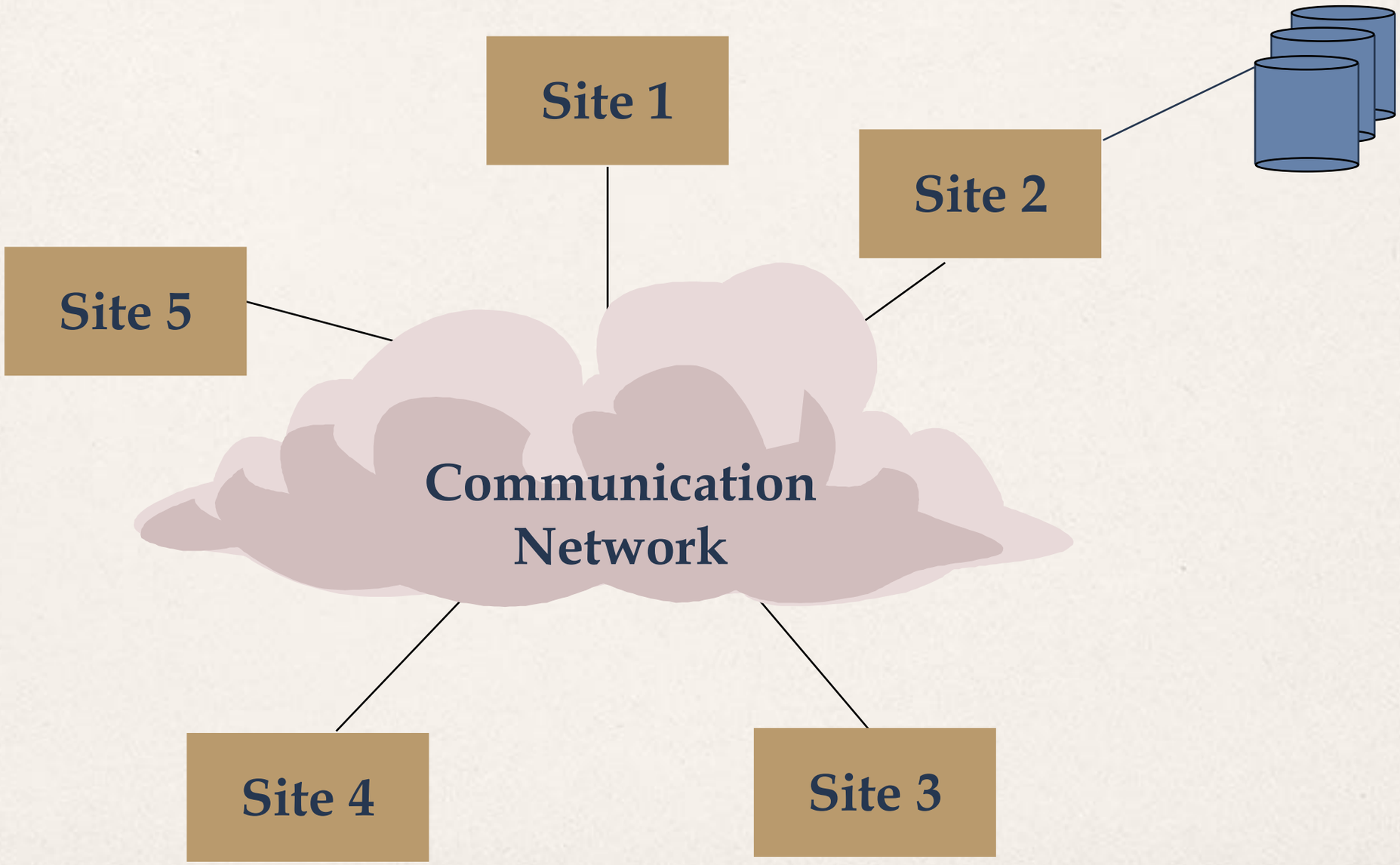
A distributed database (DDB) is a collection of multiple, *logically interrelated* databases distributed over a *computer network*.

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.

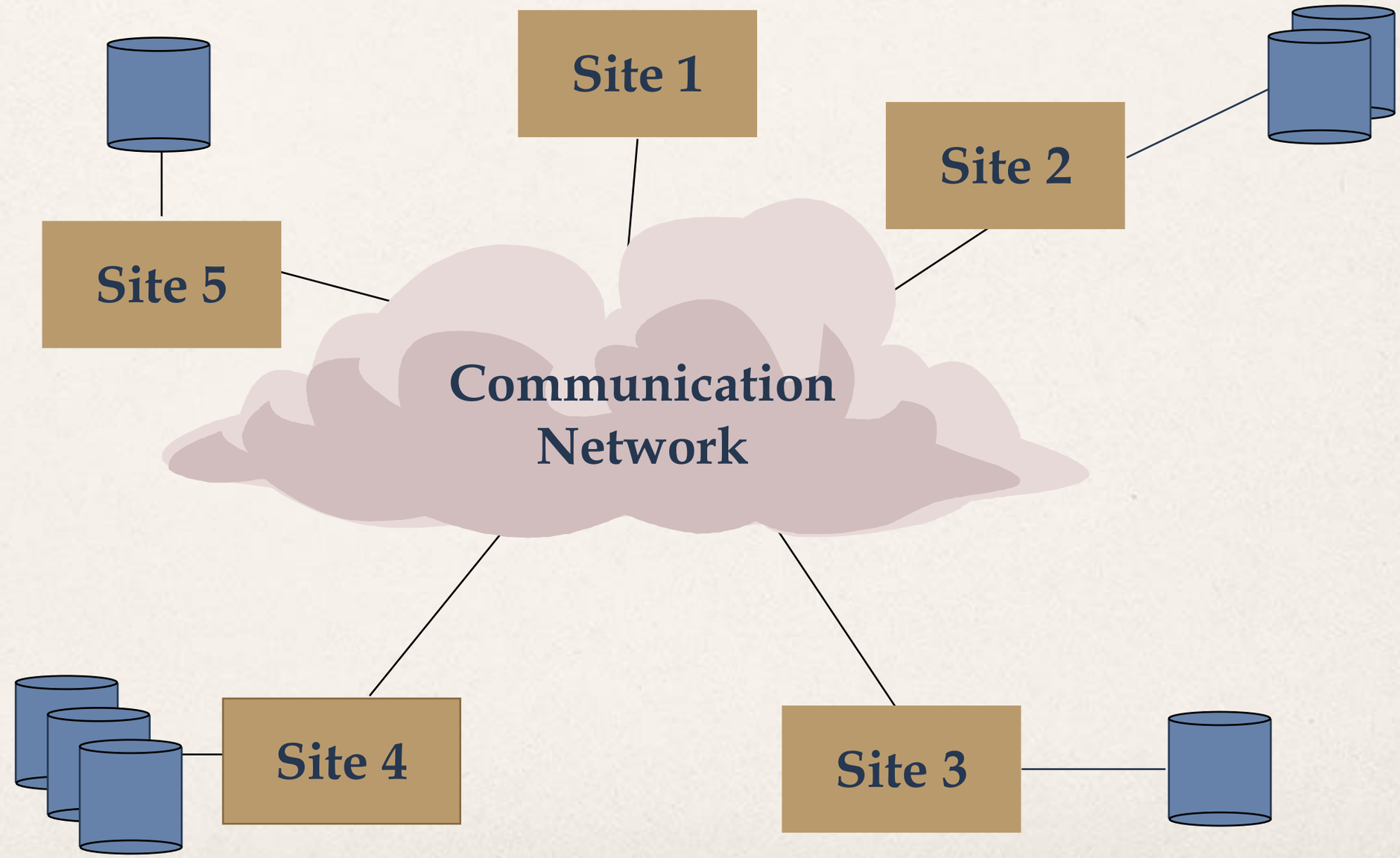
What is not a DDBS?

- A timesharing computer system
- A loosely or tightly coupled multiprocessor system
- A database system which resides at one of the nodes of a network of computers - this is a centralized database on a network node

Centralized DBMS on a Network



Distributed DBMS Environment



DistributeDBMS Promisesd

A **Distributed Database Management System (DDBMS)** is software that manages a collection of **interrelated databases** distributed across different sites (computers) and makes them appear to users as **one single database**.

1. Transparent management of distributed, fragmented, and replicated data
2. Improved reliability/availability through distributed transactions
3. Improved performance
4. Easier and more economical system expansion

Transparency

- Transparency is the separation of the higher level semantics of a system from the lower level implementation issues.
- Fundamental issue is to provide

  data independence

  in the distributed environment
  - Network (distribution) transparency
  - Replication transparency
  - Fragmentation transparency
    - horizontal fragmentation: selection
    - vertical fragmentation: projection
    - hybrid

Example

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.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

PAY

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

Transparent Access

Consider an engineering firm that has offices in Boston, Waterloo, Paris and San Francisco. We would like to maintain a **distributed** database for their data.

- Partition
  - ✓ Given the distributed nature of this firm's business, it is preferable to localize data such that data about the employees in Waterloo office are stored in Waterloo, those in the Boston office are stored in Boston, and so forth. Partition each of the relations => store each partition at a different site.

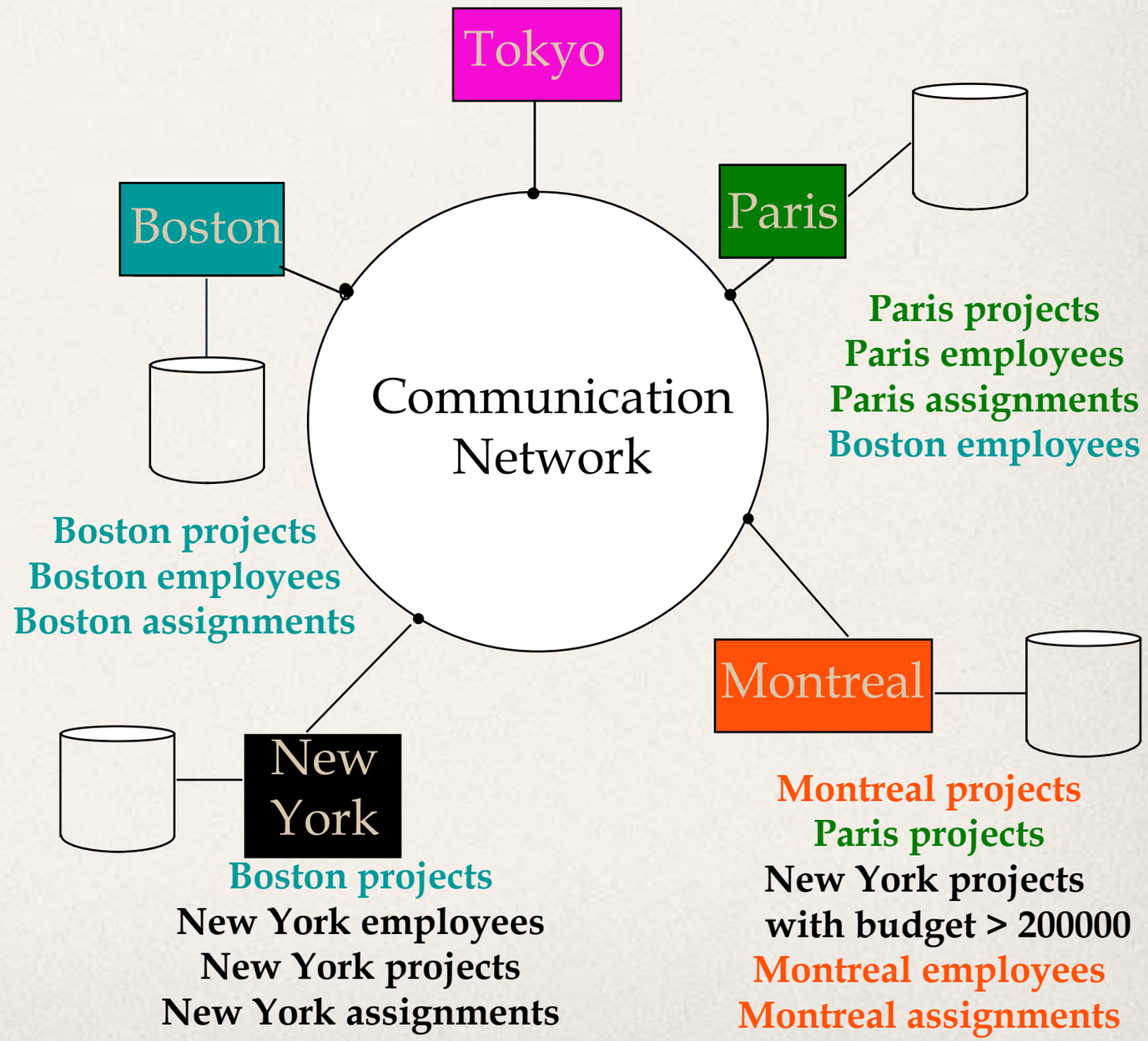
- Duplication
  - ✓ Duplicate some of this data at other sites for performance and reliability reasons.

Fully transparent access means that the users can still pose the query as specified above, without paying any attention to the fragmentation, location, or replication of data, and let the system worry about resolving these issues.

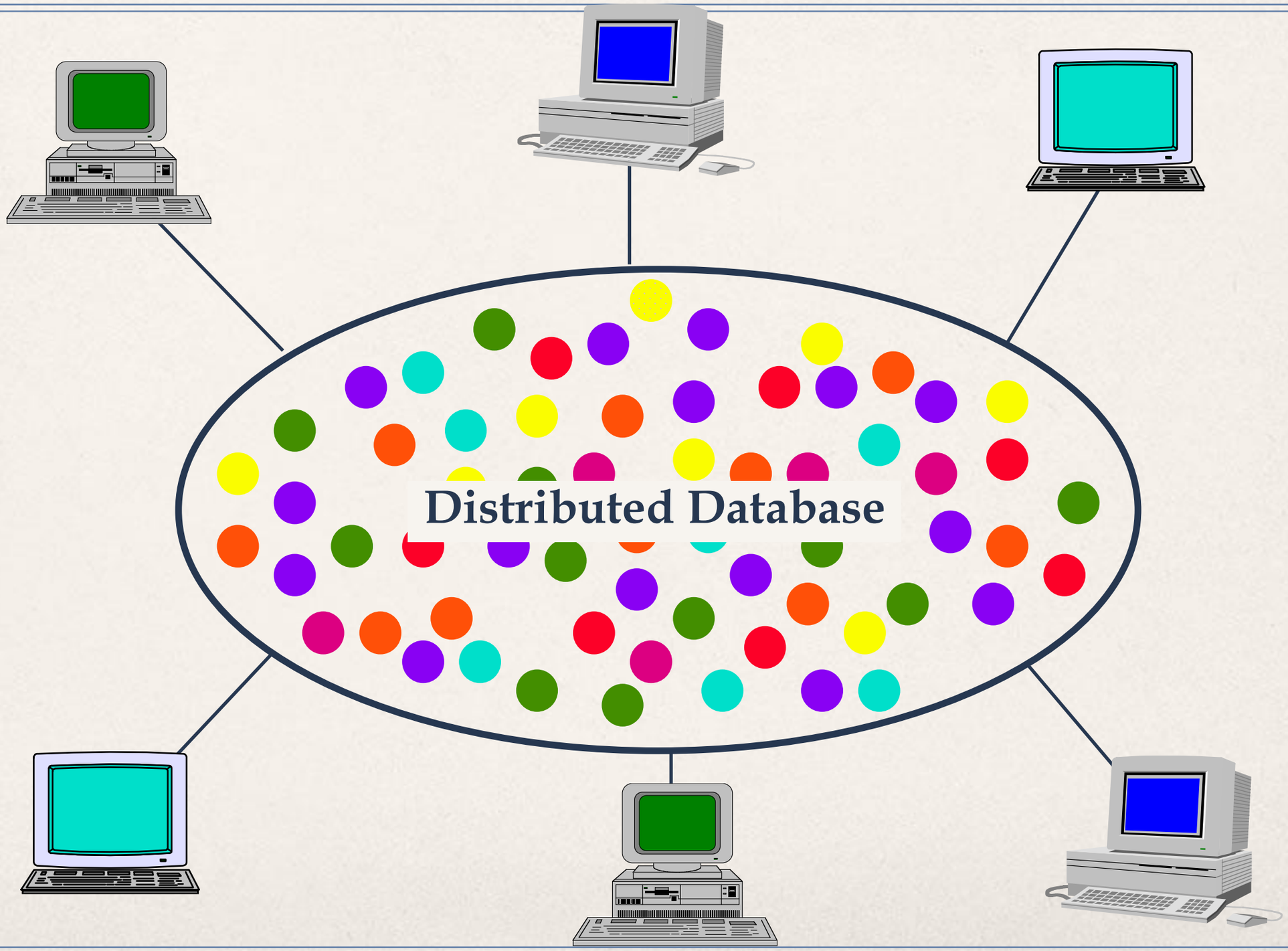
Transparent Access

Find out the names and employees who worked on a project for more than 12 months:

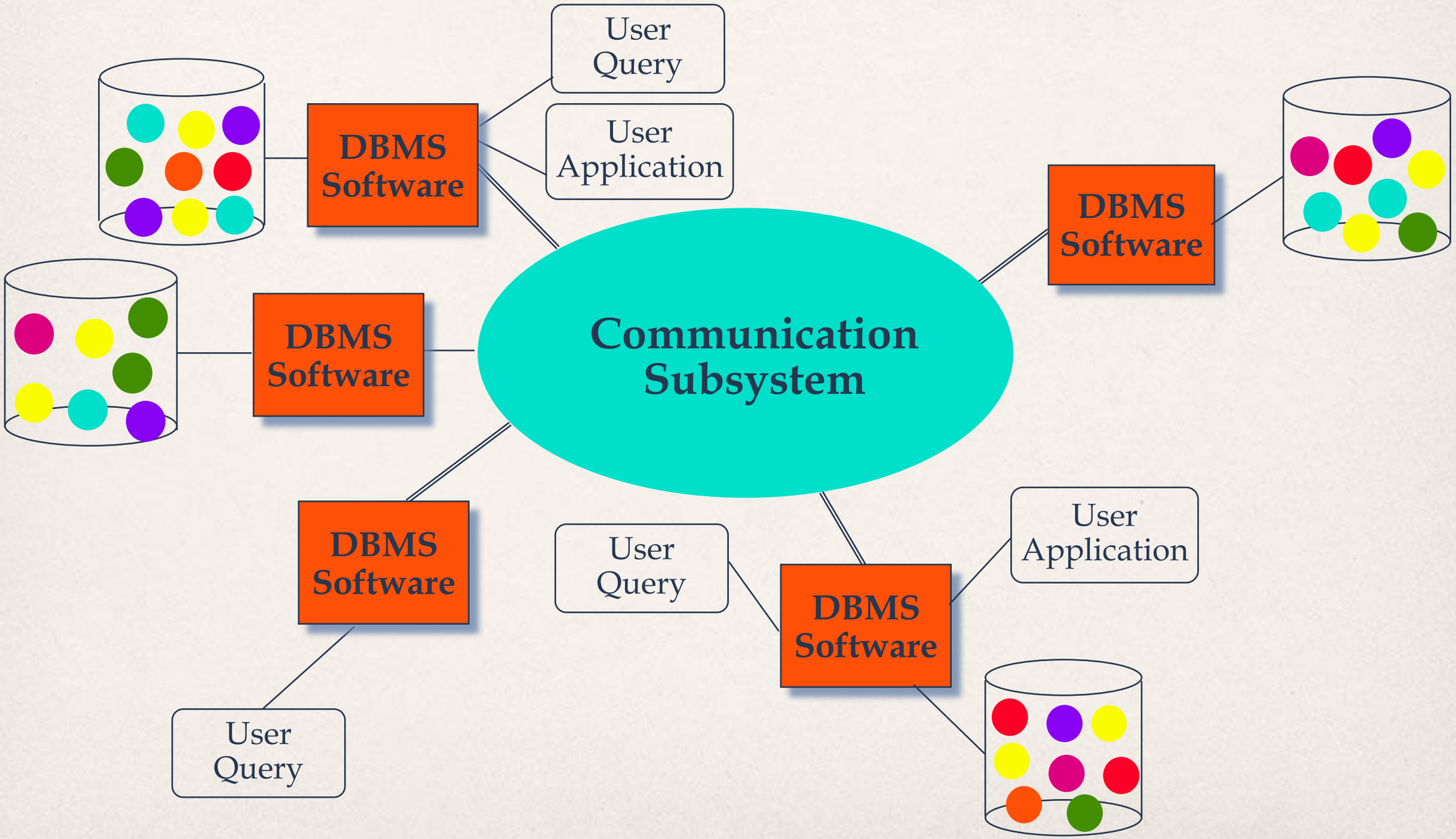
```
SELECT  ENAME,SAL
FROM    EMP,ASG,PAY
WHERE   DUR > 12
AND     EMP.ENO = ASG.ENO
AND     PAY.TITLE = EMP.TITLE
```



Distributed Database - User View



Distributed DBMS - Reality



Types of Transparency

- Data independence
- Network transparency (or distribution transparency)
  - Location transparency
  - Naming transparency
- Replication transparency
  - For performance, reliability, and availability reasons, it is usually desirable to be able to distribute data in a replicated fashion across the machines on a network.
- Fragmentation transparency
  - This is commonly done for reasons of performance, availability, and reliability. Furthermore, fragmentation can reduce the negative effects of replication. Each replica is not the full relation but only a subset of it; thus less space is required and fewer data items need be managed

Distributed DBMS Issues

- **Distributed Database Design**
  - How to distribute the database
  - Replicated & non-replicated database distribution
  - A related problem in directory management
- **Query Processing**
  - Convert user transactions to data manipulation instructions
  - Optimization problem
    - min{cost = data transmission + local processing}
  - General formulation is NP-hard

Distributed DBMS Issues

- **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