Distributed Database

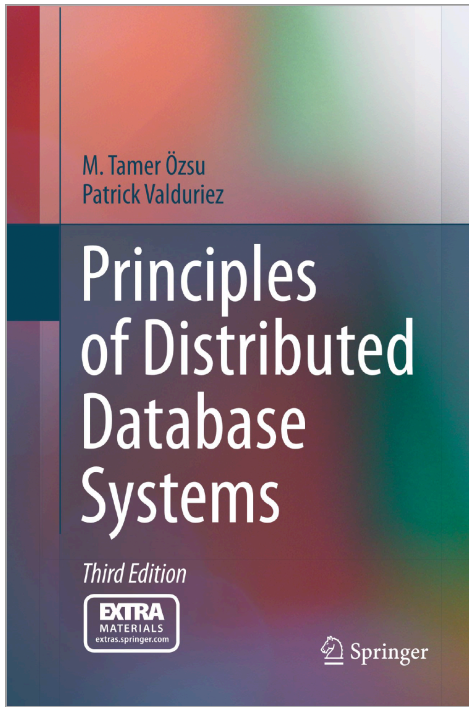
Lecture 1

By
Dr. Mohamed Elhoseny
2019

Outlines

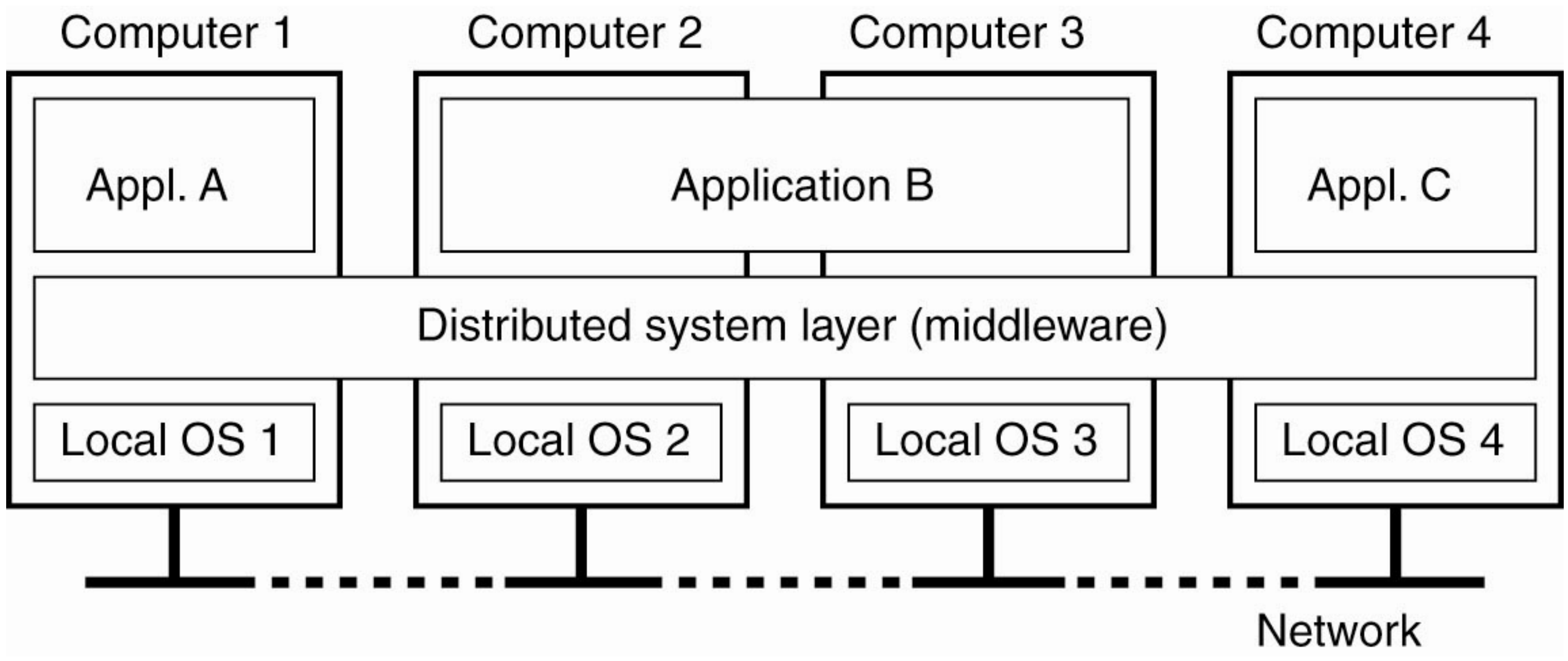
- Book

  Principles of Distributed Database Systems, Third Edition
  Authors, M. Tamer Özsu and Patrick Valduriez

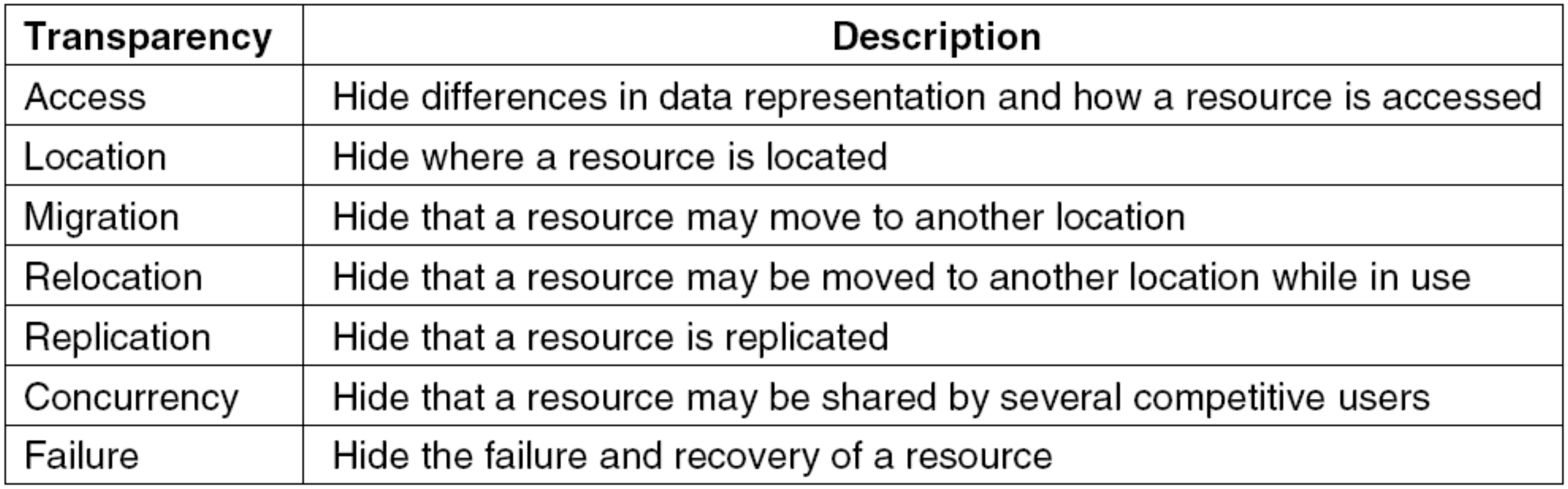

M. Tamer Özsu
Patrick Valduriez
Principles of Distributed Database Systems
Third Edition
EXTRA MATERIALS extras.springer.com
Springer

Definition of a Distributed System

A distributed system is: A collection of independent computers that appears to its users as a single coherent system.



Transparency in a Distributed System



Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource is replicated
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource

Distributed Database Concepts

It is a system to process a Unit of execution (a transaction) in a distributed manner. That is, a transaction can be executed by multiple networked computers in a unified manner.

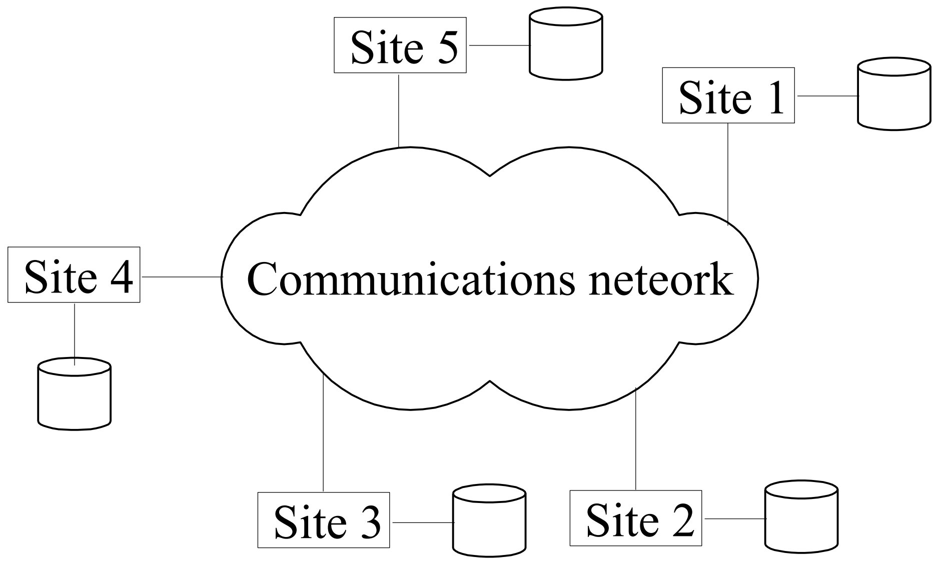
It can be defined as

A distributed database (DDB) is a collection of multiple logically interrelated databases distributed over a computer network, and a distributed database management system as a software system that manages a distributed database while making the distribution transparent to the user.

Distributed Database System

Advantages

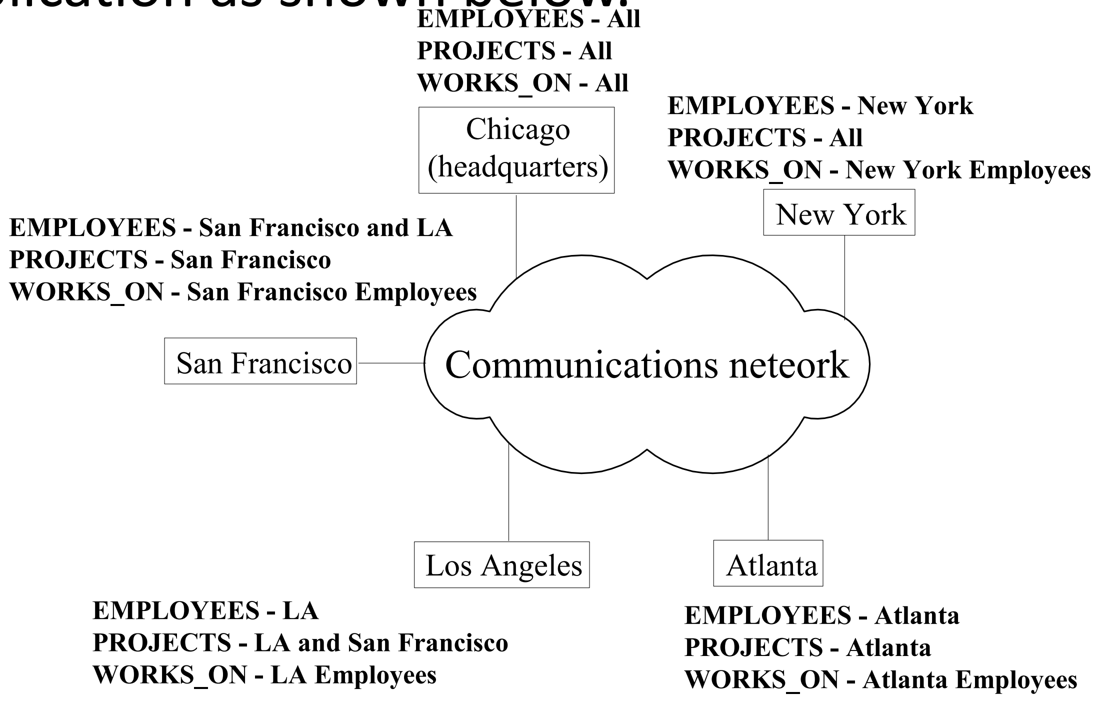
1. **Management of distributed data with different levels of transparency**: This refers to the physical placement of data (files, relations, etc.) which is not known to the user (distribution transparency).



Distributed Database System

Advantages

The EMPLOYEE, PROJECT, and WORKS_ON tables may be fragmented horizontally and stored with possible replication as shown below.



Distributed Database System

Advantages

- **Distribution and Network transparency**: Users do not have to worry about operational details of the network. There is **Location transparency**, which refers to freedom of issuing command from any location without affecting its working. Then there is Naming transparency, which allows access to any names object (files, relations, etc.) from any location.
- **Replication transparency**: It allows to store copies of a data at multiple sites as shown in the above diagram. This is done to minimize access time to the required data.
- **Fragmentation transparency**: Allows to fragment a relation horizontally (create a subset of tuples of a relation) or vertically (create a subset of columns of a relation).

Distributed Database System

Advantages

2. **Increased reliability and availability**: Reliability refers to system live time, that is, system is running efficiently most of the time. Availability is the probability that the system is continuously available (usable or accessible) during a time interval. A distributed database system has multiple nodes (computers) and if one fails then others are available to do the job.
3. **Improved performance**: A distributed DBMS fragments the database to keep data closer to where it is needed most. This reduces data management (access and modification) time significantly.
4. **Easier expansion (scalability):** Allows new nodes (computers) to be added anytime without chaining the entire configuration.

Data Fragmentation, Replication and Allocation

Data Fragmentation

Split a relation into logically related and correct parts. A relation can be fragmented in two ways:

<u>Horizontal fragmentation</u>

- It is a horizontal subset of a relation which contain those of tuples which satisfy selection conditions.
- Consider the Employee relation with selection condition (DNO = 5). All tuples satisfy this condition will create a subset which will be a horizontal fragment of Employee relation.
- A selection condition may be composed of several conditions connected by AND or OR.
- Derived horizontal fragmentation: It is the partitioning of a primary relation to other secondary relations which are related with Foreign keys.

Data Fragmentation, Replication and Allocation

Vertical fragmentation

- It is a subset of a relation which is created by a subset of columns. Thus a vertical fragment of a relation will contain values of selected columns.
- Consider the Employee relation. A vertical fragment of can be created by keeping the values of Name, Bdate, Sex, and Address.
- Because there is no condition for creating a vertical fragment, each fragment must include the primary key attribute of the parent relation Employee. In this way all vertical fragments of a relation are connected.

Data Fragmentation, Replication and Allocation

Representation of Fragmentation

Horizontal fragmentation

- Each horizontal fragment on a relation can be specified by a σ_{Ci} (R) operation in the relational algebra.
- Complete horizontal fragmentation
  - A set of horizontal fragments whose conditions C1, C2, …, Cn include all the tuples in R- that is, every tuple in R satisfies (C1 OR C2 OR … OR Cn).
- Disjoint complete horizontal fragmentation: No tuple in R satisfies (Ci AND Cj) where $i \neq j$.
- To reconstruct R from horizontal fragments a UNION is applied.

Data Fragmentation, Replication and Allocation

Vertical fragmentation

- A vertical fragment on a relation can be specified by a $\Pi_{Li}(R)$ operation in the relational algebra.
- Complete vertical fragmentation
  - A set of vertical fragments whose projection lists L1, L2, ..., Ln include all the attributes in R but share only the primary key of R. In this case the projection lists satisfy the following two conditions:

    $L1 \cup L2 \cup ... \cup Ln = ATTRS(R)$

    $Li \cap Lj = PK(R)$ for any i j, where ATTRS (R) is the set of attributes of R and PK(R) is the primary key of R.
- To reconstruct R from complete vertical fragments a OUTER UNION is applied.

Data Fragmentation, Replication and Allocation

Mixed (Hybrid) fragmentation

- A combination of Vertical fragmentation and Horizontal fragmentation.
- This is achieved by SELECT-PROJECT operations which is represented by $\Pi_{Li}(\sigma_{Ci}(R))$.
- If C = True (Select all tuples) and L ≠ ATTRS(R), we get a vertical fragment, and if C ≠ True and L ≠ ATTRS(R), we get a mixed fragment.
- If C = True and L = ATTRS(R), then R can be considered a fragment.

Data Fragmentation, Replication and Allocation

Fragmentation schema

A definition of a set of fragments (horizontal or vertical or horizontal and vertical) that includes all attributes and tuples in the database that satisfies the condition that the whole database can be reconstructed from the fragments by applying some sequence of UNION (or OUTER JOIN) and UNION operations.

Allocation schema

It describes the distribution of fragments to sites of distributed databases. It can be fully or partially replicated or can be partitioned.

Data Fragmentation, Replication and Allocation

Data Replication

Database is replicated to all sites. In full replication the entire database is replicated and in partial replication some selected part is replicated to some of the sites. Data replication is achieved through a replication schema.

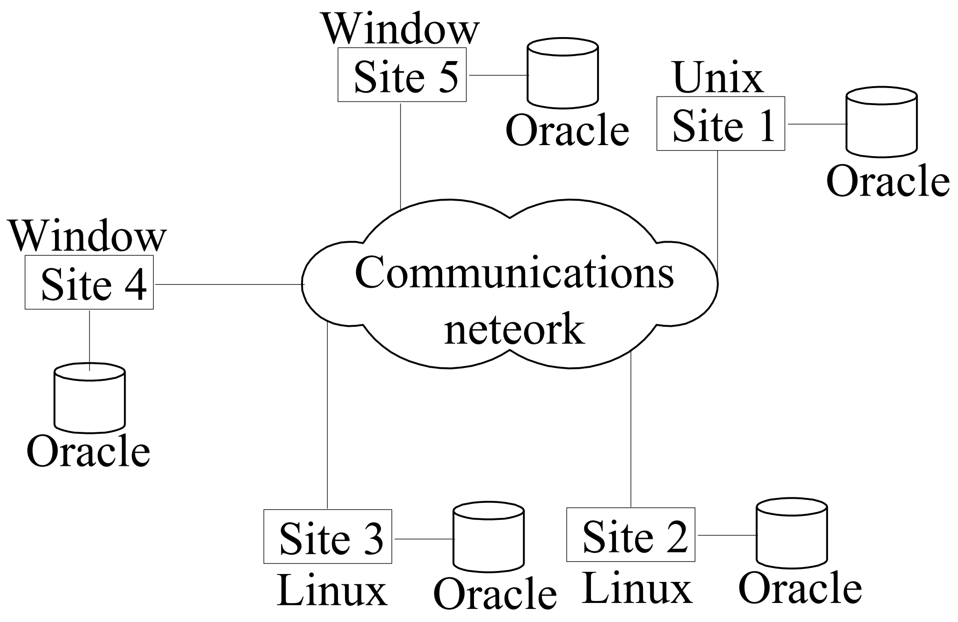
Data Distribution (Data Allocation)

This is relevant only in the case of partial replication or partition. The selected portion of the database is distributed to the database sites.

Types of Distributed Database Systems

Homogeneous

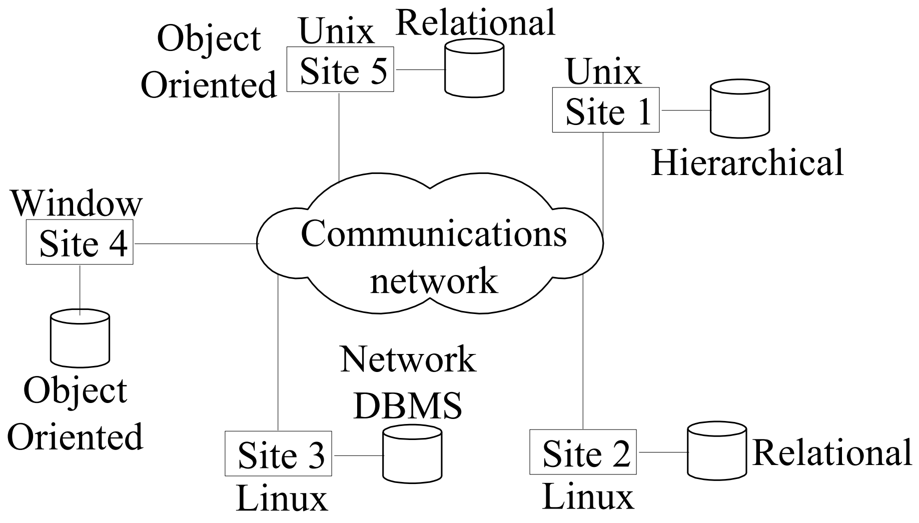
All sites of the database system have identical setup, i.e., same database system software. The underlying operating system may be different. For example, all sites run Oracle or DB2, or Sybase or some other database system. The underlying operating systems can be a mixture of Linux, Window, Unix, etc. The clients thus have to use identical client software.



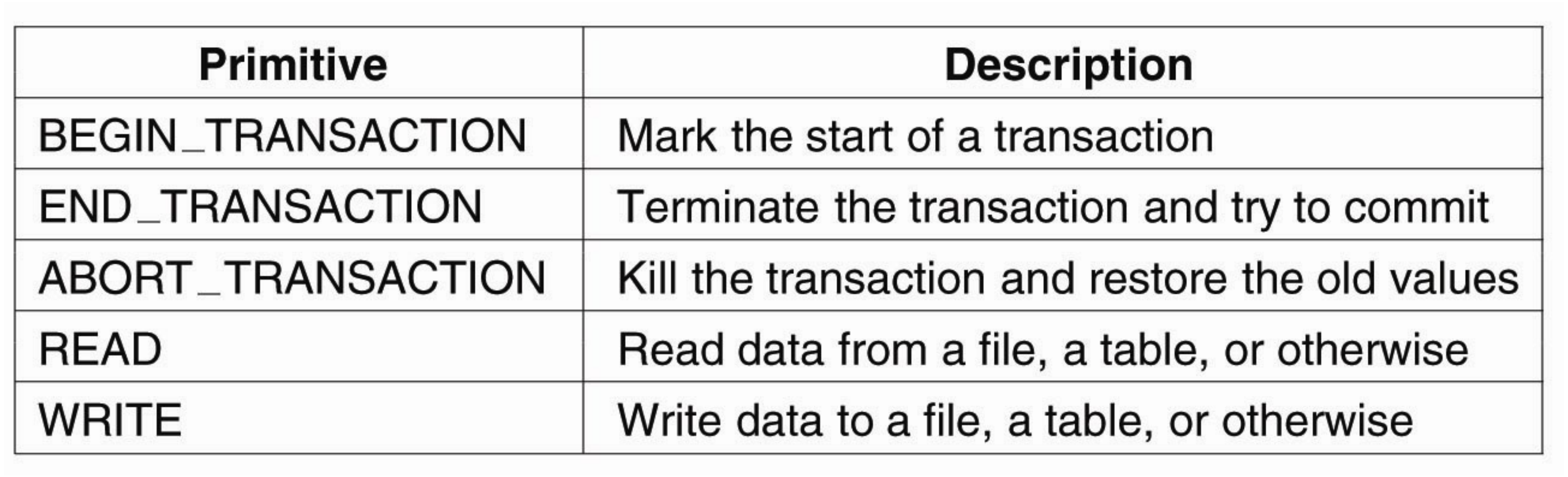
Types of Distributed Database Systems

Heterogeneous

- **Federated**: Each site may run different database system but the data access is managed through a single conceptual schema.
- This implies that the degree of local autonomy is minimum. Each site must adhere to a centralized access policy. There may be a global schema.
- **Multidatabase**: There is no one conceptual global schema. For data access a schema is constructed dynamically as needed by the application software.



Transaction Processing Systems

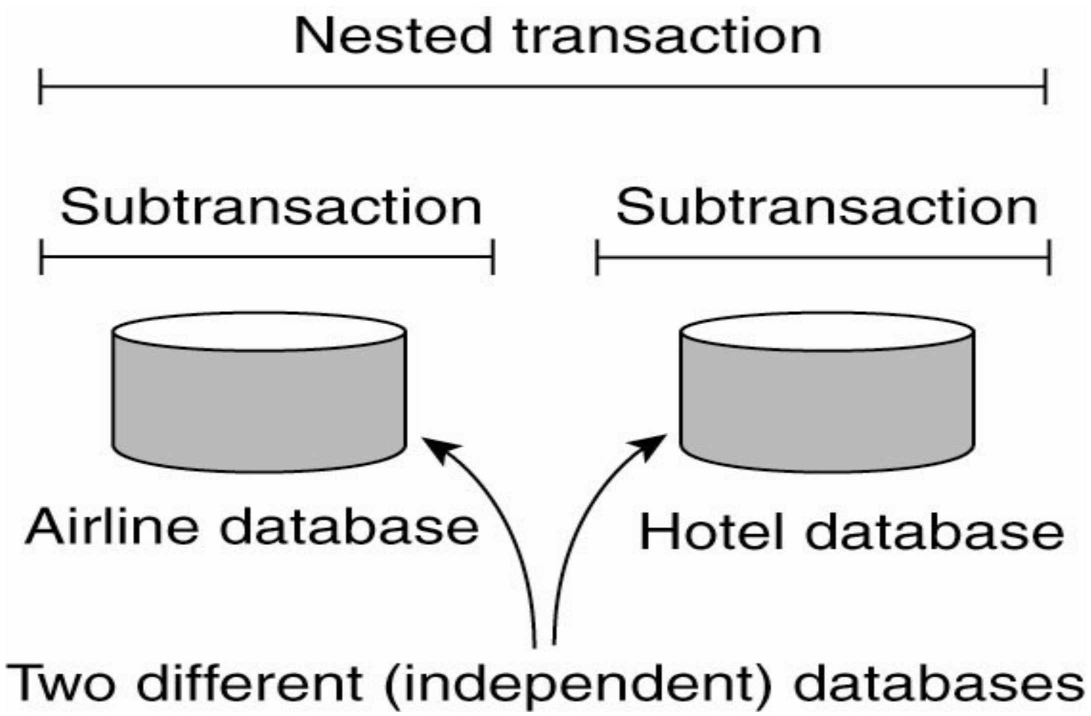


Primitive	Description
BEGIN_TRANSACTION	Mark the start of a transaction
END_TRANSACTION	Terminate the transaction and try to commit
ABORT_TRANSACTION	Kill the transaction and restore the old values
READ	Read data from a file, a table, or otherwise
WRITE	Write data to a file, a table, or otherwise

Transaction Processing Systems

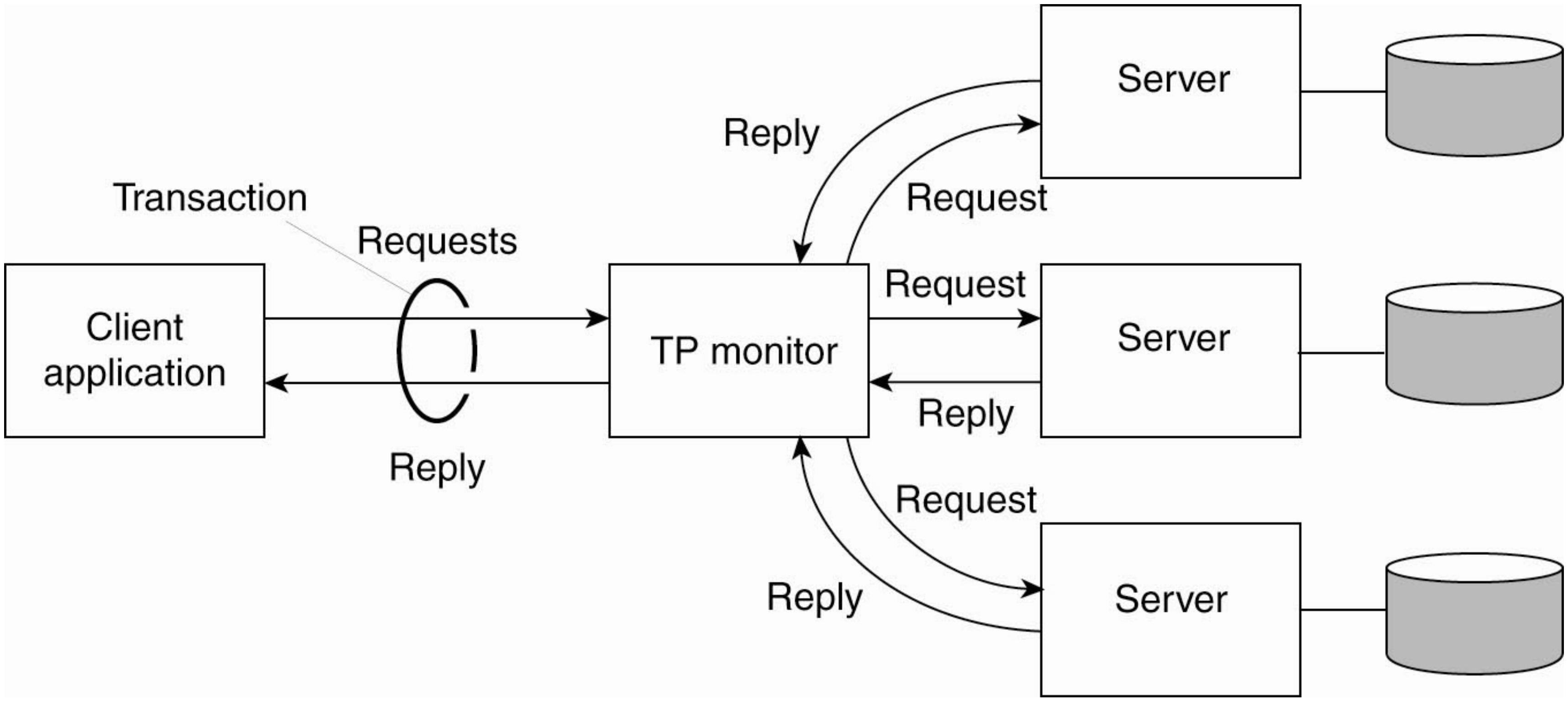
- **Characteristic properties of transactions:**
  - **Atomic**: To the outside world, the transaction happens indivisibly.
  - **Consistent**: The transaction does not violate system invariants.
  - **Isolated**: Concurrent transactions do not interfere with each other.
  - **Durable**: Once a transaction commits, the changes are permanent.

Transaction Processing Systems



Transaction Processing Systems

- The role of a TP monitor in distributed systems.



Distributed Pervasive Systems

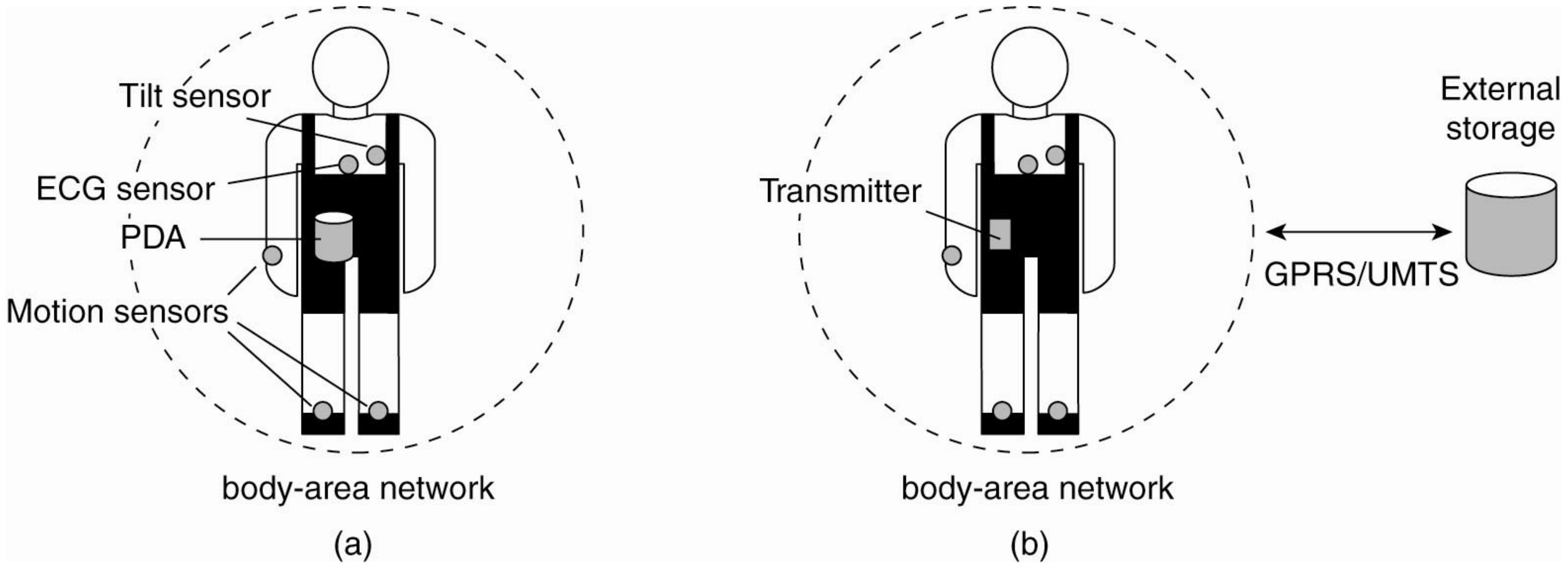
- Requirements for pervasive systems
    - Embrace contextual changes.
    - Encourage ad hoc composition.
    - Recognize sharing as the default.

Electronic Health Care Systems

- Questions to be addressed for health care systems:

- Where and how should monitored data be stored?
- How can we prevent loss of crucial data?
- What infrastructure is needed to generate and propagate alerts?
- How can physicians provide online feedback?
- How can extreme robustness of the monitoring system be realized?
- What are the security issues and how can the proper policies be enforced?

Electronic Health Care Systems



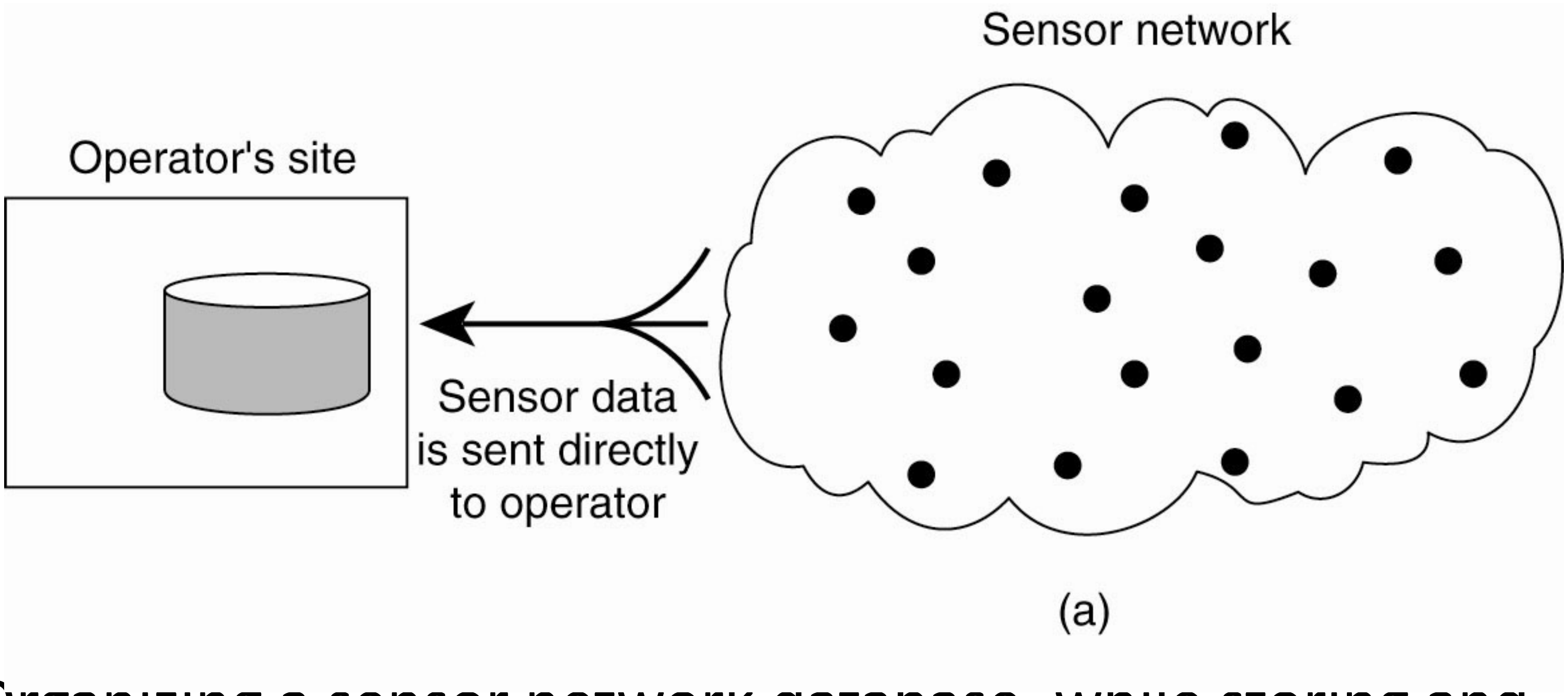
- Monitoring a person in a pervasive electronic health care system, using (a) a local hub or (b) a continuous wireless connection.

Sensor Networks

- Questions concerning sensor networks:

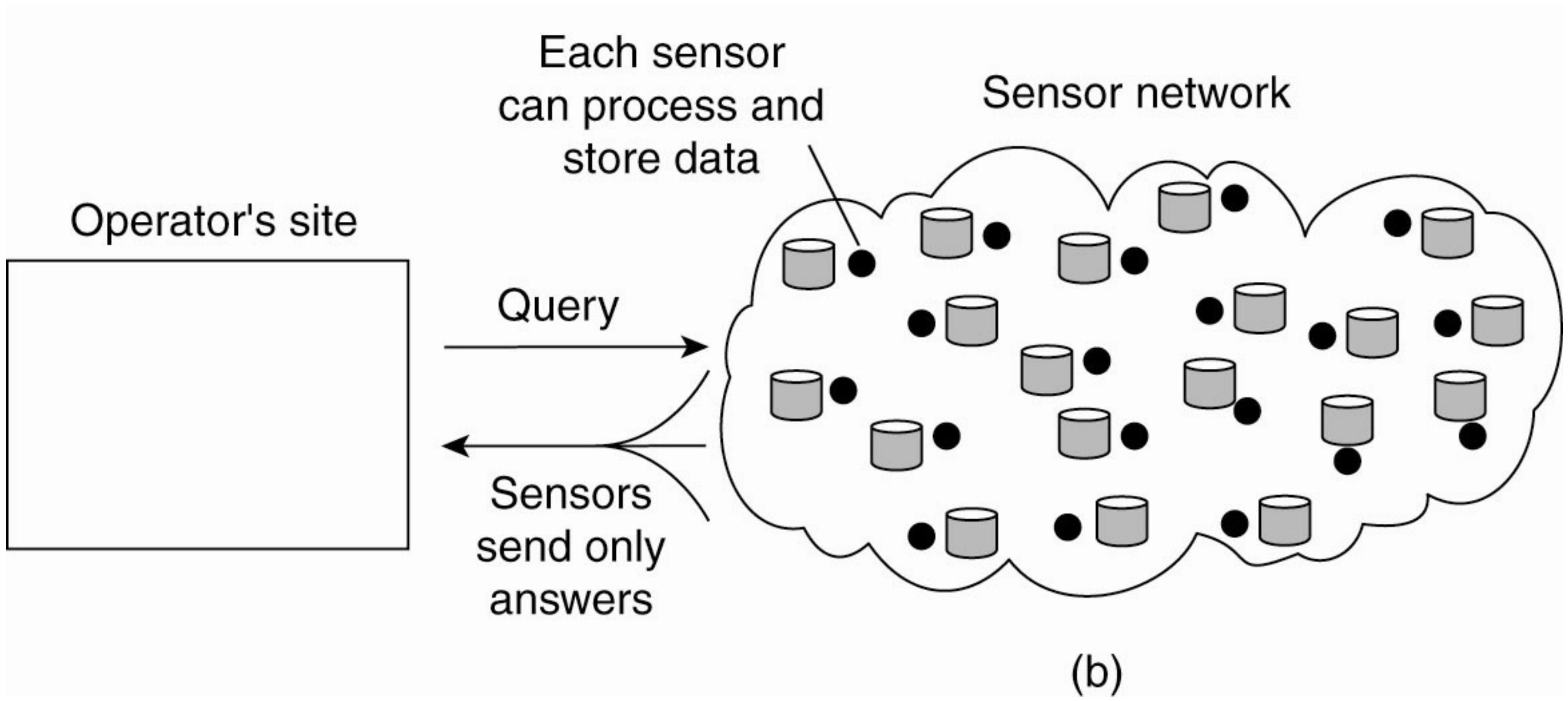
- How do we (dynamically) set up an efficient tree in a sensor network?
- How does aggregation of results take place? Can it be controlled?
- What happens when network links fail?

Sensor Networks



- Organizing a sensor network database, while storing and processing data (a) only at the operator’s site or …

Sensor Networks (3)


Each sensor can process and store data
Sensor network
Operator's site
Query
Sensors send only answers
(b)

- Organizing a sensor network database, while storing and processing data … or (b) only at the sensors.

End of Lecture 1

Thank You