DISTRIBUTED SYSTEMS
Principles and Paradigms
Second Edition
ANDREW S. TANENBAUM
MAARTEN VAN STEEN

Chapter 1
Introduction

A. A. Pourhaji Kazem,   Fall 2009

Definition of a Distributed System (1)

A distributed system is:

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

Important Characteristics of Distributed Systems

- One important characteristic is that differences between the various computers and the ways in which they communicate are mostly hidden from users.

- Another important characteristic is that users and applications can interact with a distributed system in a consistent and uniform way, regardless of where and when interaction takes place.

Important Characteristics of Distributed Systems (cont.)

- In principle, distributed systems should also be relatively easy to expand or scale.

- Scalability characteristic is a direct consequence of having independent computers, but at the same time, hiding how these computers actually take part in the system as a whole.

Definition of a Distributed System (2)

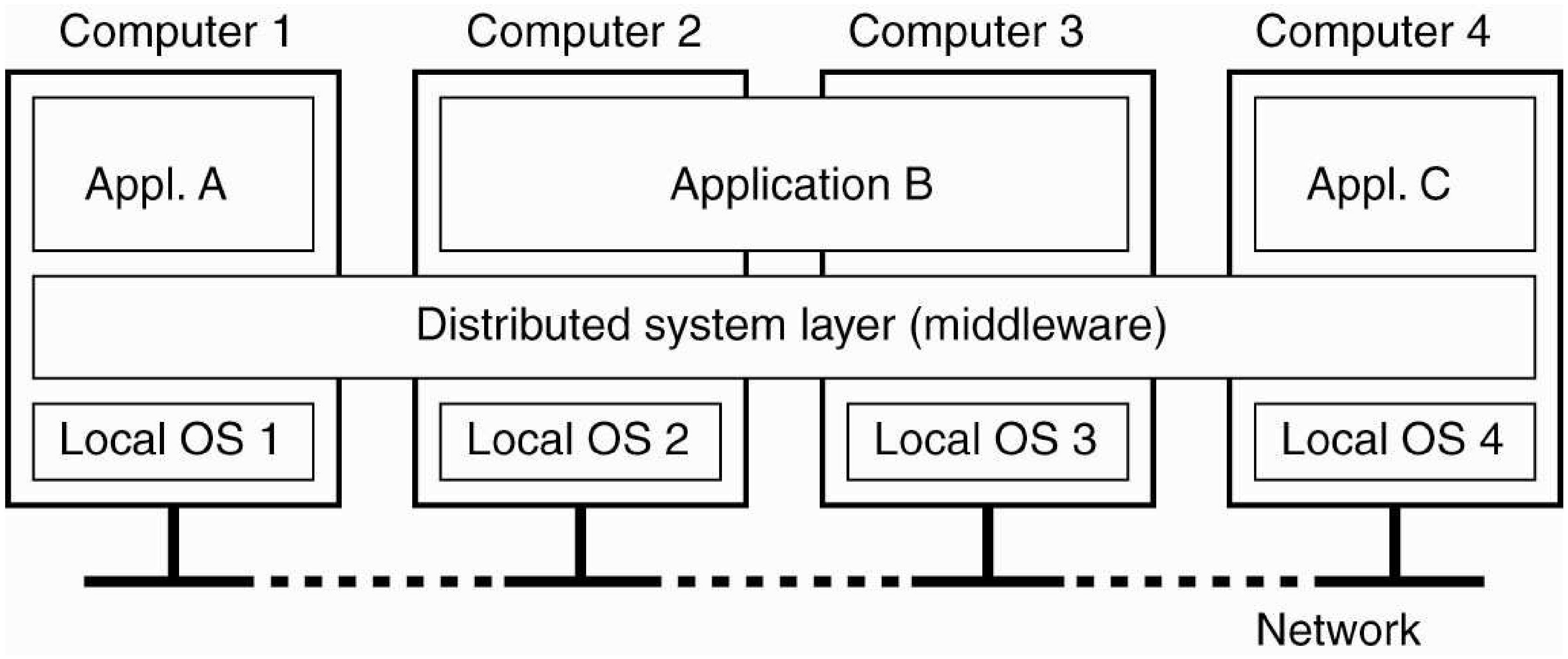


Figure 1-1. A distributed system organized as middleware. The middleware layer extends over multiple machines, and offers each application the same interface.

Goals of a Distributed System

- Making Resources Accessible
- Distribution Transparency
- Openness
- Scalability

Making Resources Accessible

- The main goal of a distributed system is to make it easy for the users (and applications) to access remote resources, and to share them in a controlled and efficient way

- Connecting users and resources also makes it easier to collaborate and exchange information

- However, as connectivity and sharing increase, security is becoming increasingly important

Distribution Transparency

- An important goal of a distributed system is to hide the fact that its processes and resources are physically distributed across multiple computers

- A distributed system that is able to present itself to users and applications as if it were only a single computer system is said to be transparent

Types of Transparency

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

Figure 1-2. Different forms of transparency in a distributed system (ISO, 1995).

Degree of Transparency

- Although distribution transparency is generally considered preferable for any distributed system, there are situations in which attempting to completely hide all distribution aspects from users is not a good idea

- There is also a trade-off between a high degree of transparency and the performance of a system

Openness

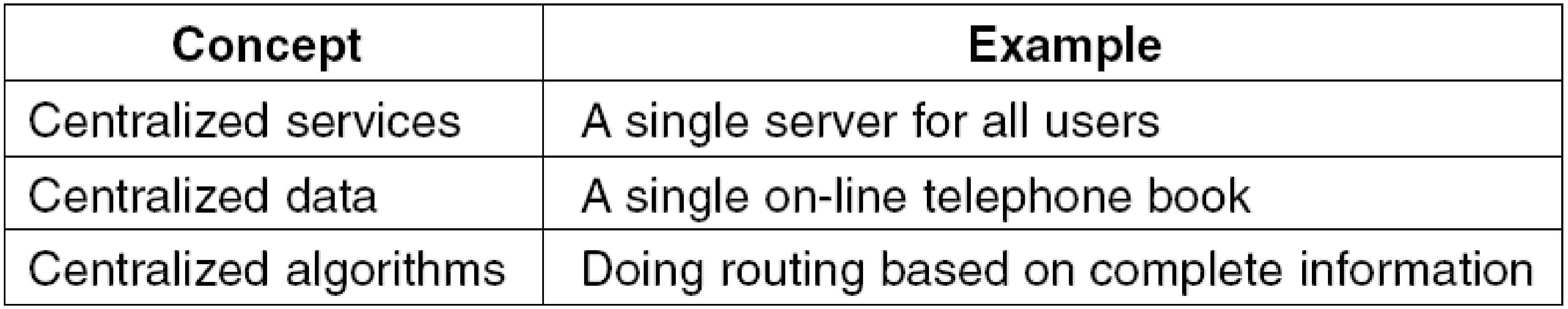
- An open distributed system is a system that offers services according to standard rules that describe the syntax and semantics of those services

- For example, in computer networks, standard rules govern the format, contents, and meaning of messages sent and received

Scalability

- Scalability of a system can be measured along at least three different dimensions:
  - First, a system can be scalable with respect to its size, meaning that we can easily add more users and resources to the system.
  - Second, a geographically scalable system is one in which the users and resources may lie far apart.
  - Third, a system can be administratively scalable, meaning that it can still be easy to manage even if it spans many independent administrative organizations.

Scalability Problems



Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

Figure 1-3. Examples of scalability limitations.

Scalability Problems

Characteristics of decentralized algorithms:

- No machine has complete information about the system state.
- Machines make decisions based only on local information.
- Failure of one machine does not ruin the algorithm.
- There is no implicit assumption that a global clock exists.

Scaling Techniques

- Hiding communication latencies
- Distribution
- Replication

Scaling Techniques (cont.)

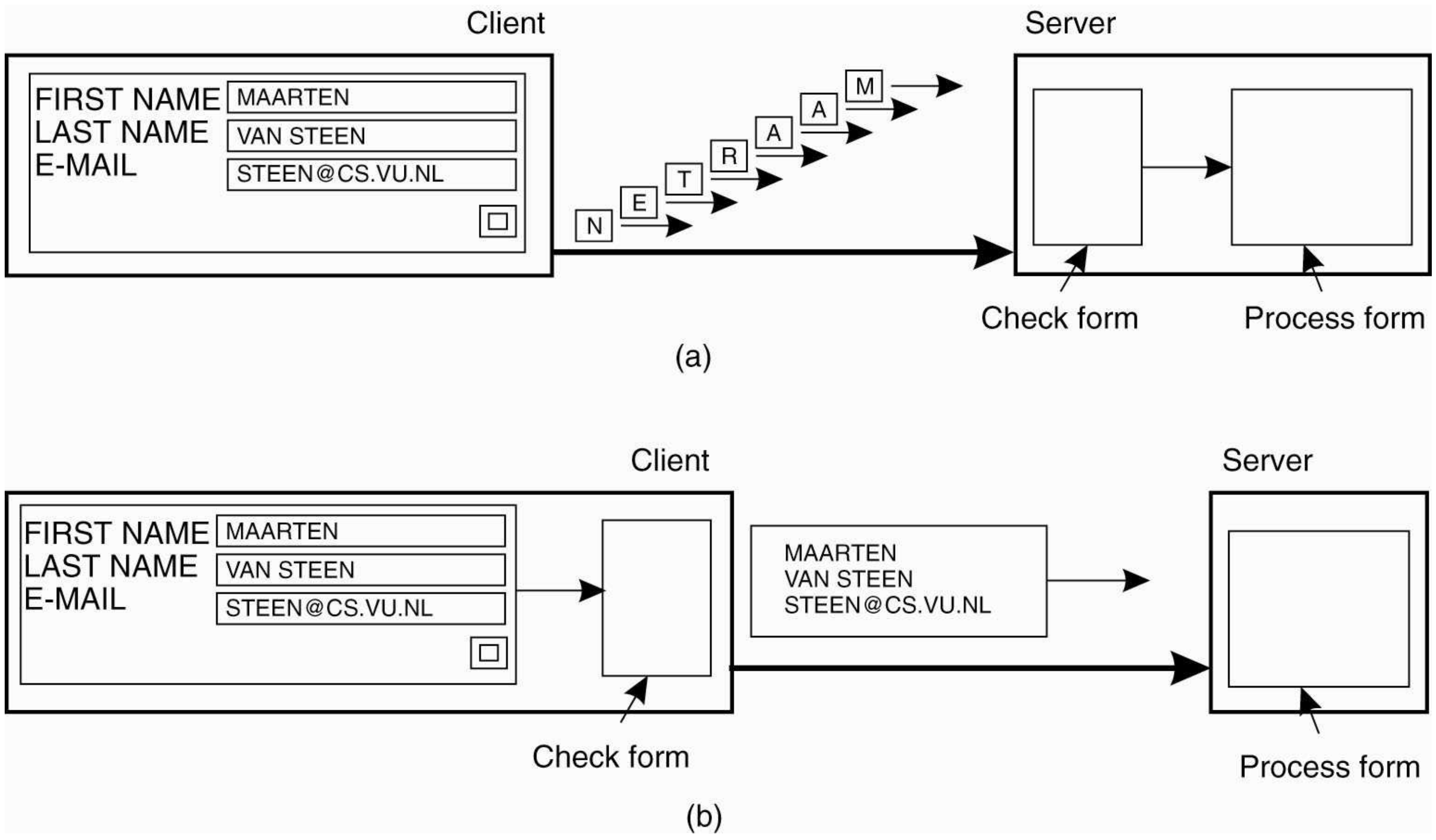


Figure 1-4. The difference between letting (a) a server or (b) a client check forms as they are being filled.

Scaling Techniques (cont.)

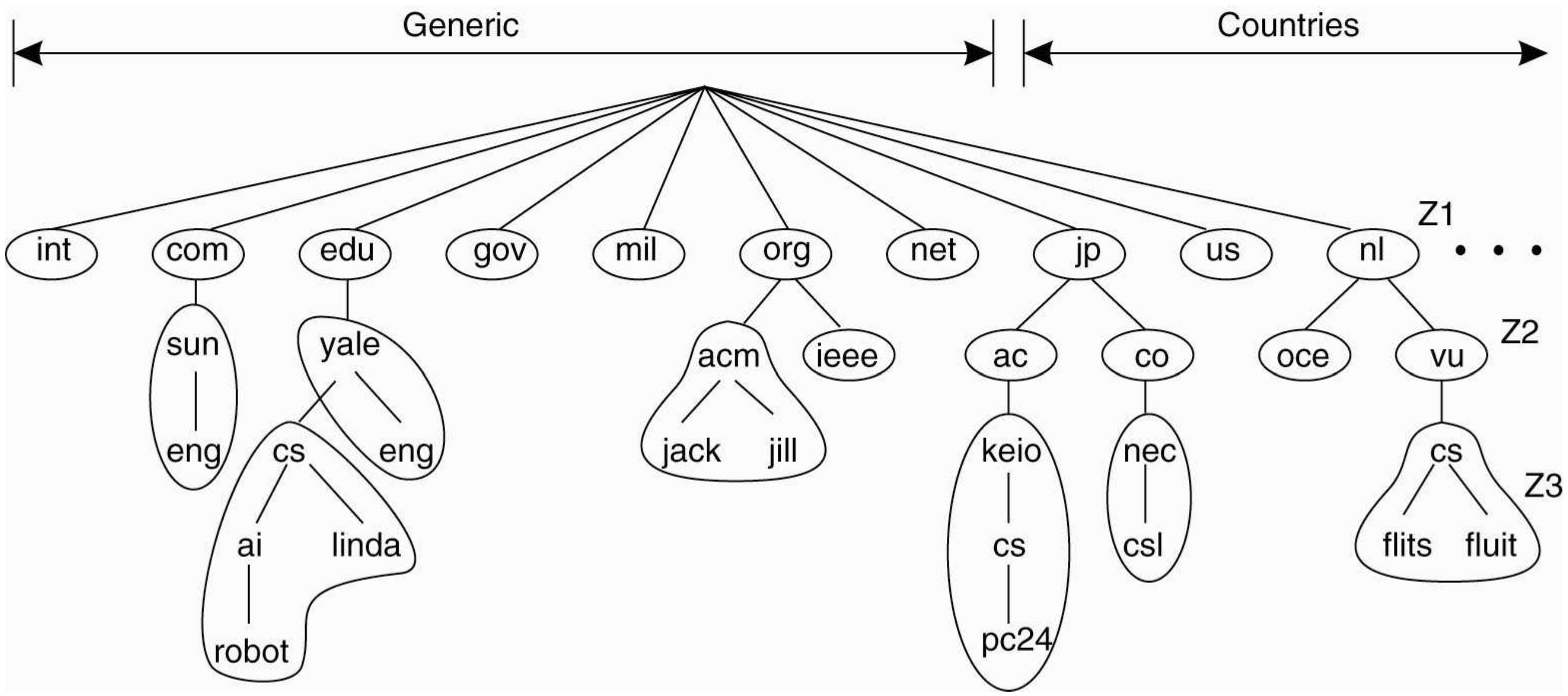


Figure 1-5. An example of dividing the DNS name space into zones.

Pitfalls when Developing Distributed Systems

False assumptions made by first time developer:

- The network is reliable.
- The network is secure.
- The network is homogeneous.
- The topology does not change.
- Latency is zero.
- Bandwidth is infinite.
- Transport cost is zero.
- There is one administrator.

Types of Distributed Systems

- Distributed Computing Systems
  - Cluster Computing System
  - Grid Computing System
- Distributed Information Systems
  - Transaction Processing Systems
  - Enterprise Application Integration
- Distributed Pervasive Systems
  - Home Systems
  - Electronic Health Care Systems
  - Sensor Networks

Cluster Computing Systems

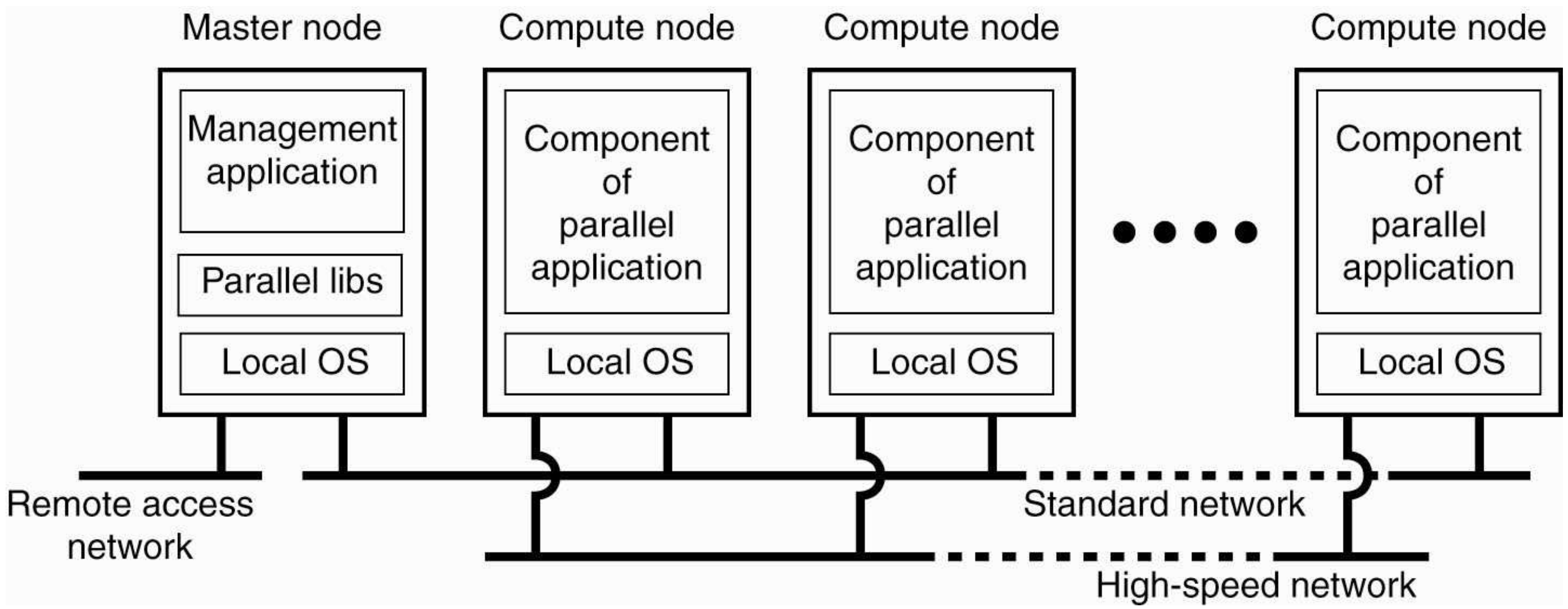


Figure 1-6. An example of a cluster computing system.

Grid Computing Systems

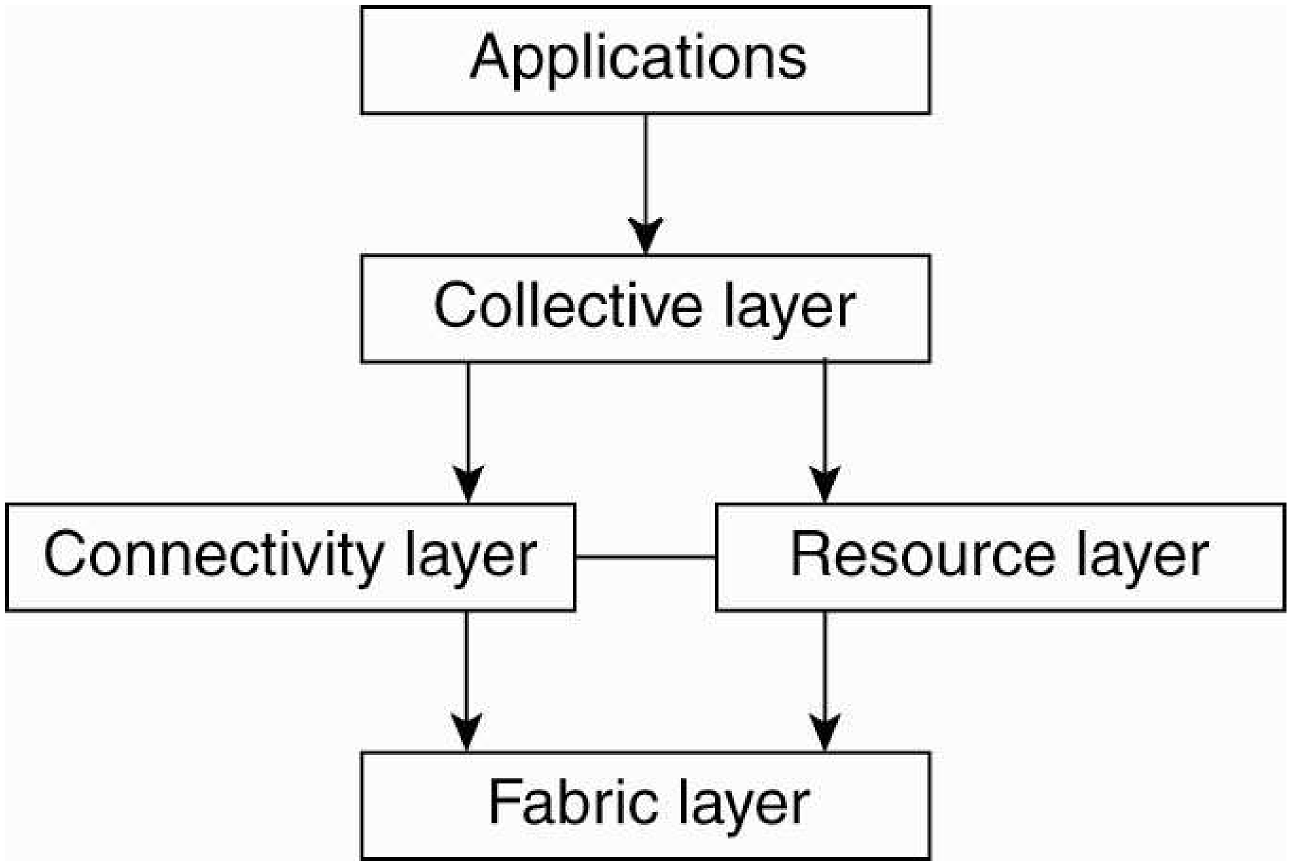


Figure 1-7. A layered architecture for grid computing systems.

Grid Computing Systems

- The lowest **fabric layer** provides interfaces to local resources at a specific site
- The **connectivity layer** consists of communication protocols for supporting grid transactions that span the usage of multiple resources
- The **resource layer** is responsible for managing a single resource. It uses the functions provided by the connectivity layer and calls directly the interfaces made available by the fabric layer
- Durable: Once a transaction commits, the

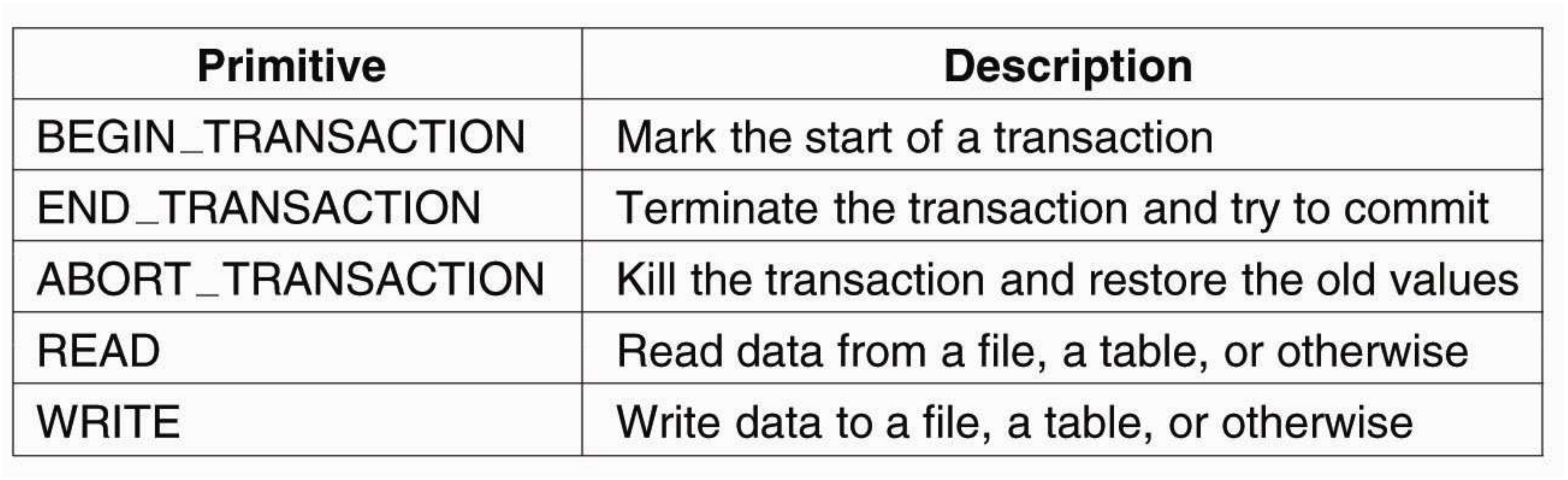
Grid Computing Systems

- The **resource layer** is responsible for managing a single resource.

  It uses the functions provided by the connectivity layer and calls directly the interfaces made available by the fabric layer

- **Collective layer** deals with handling access to multiple resources and typically consists of services for resource discovery, allocation and scheduling of tasks onto multiple resources, data replication, and so on

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

Figure 1-8. Example primitives for transactions.

Transaction Processing Systems (cont)

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 (cont)

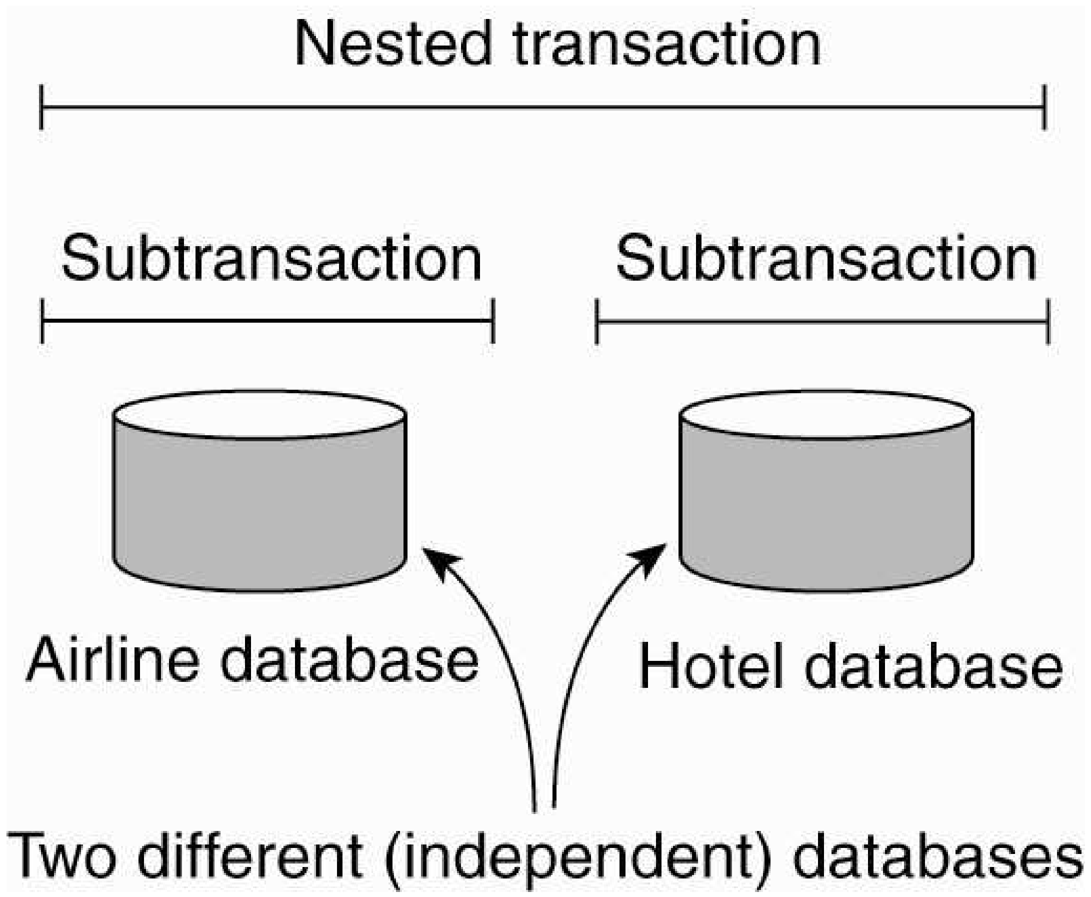


Figure 1-9. A nested transaction.

Transaction Processing Systems (cont)

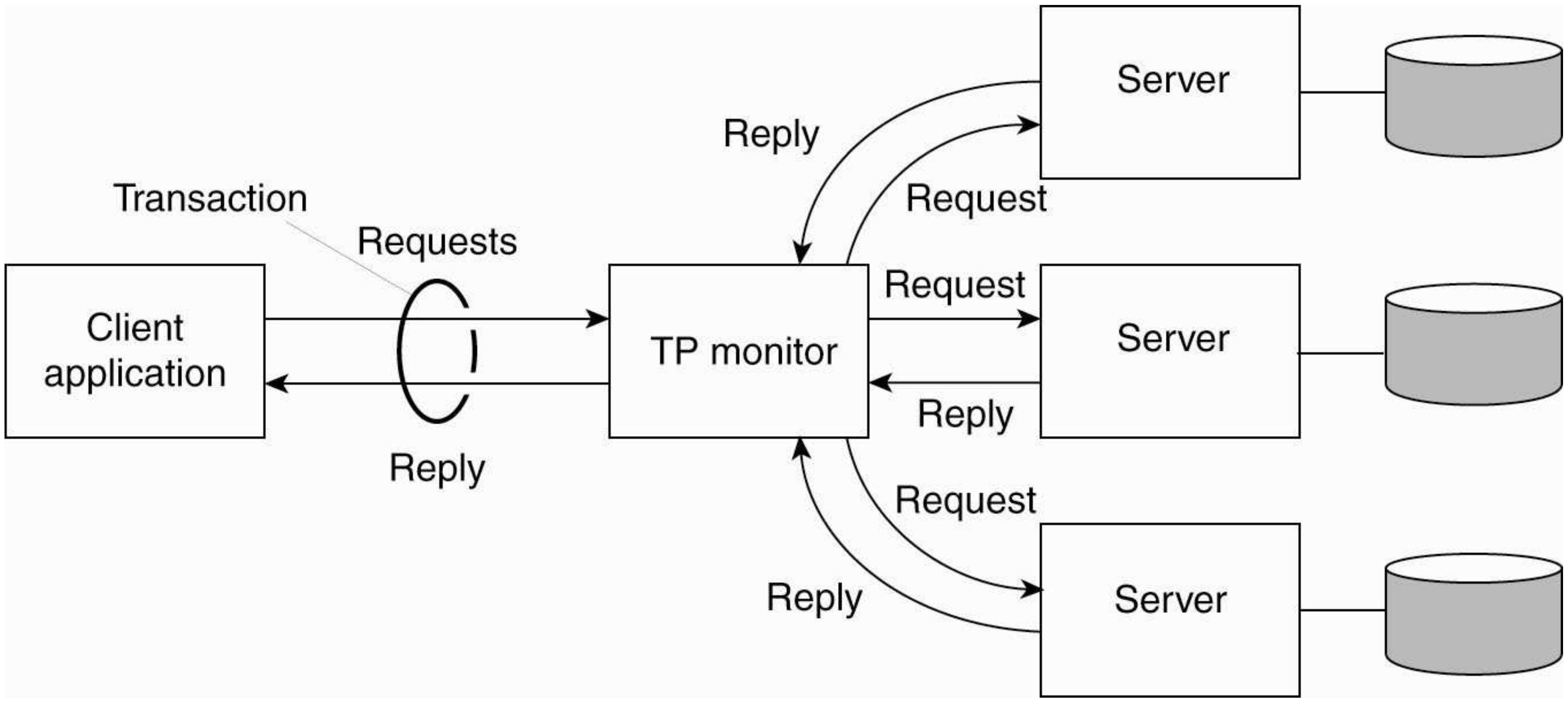


Figure 1-10. The role of a TP monitor in distributed systems.

Enterprise Application Integration

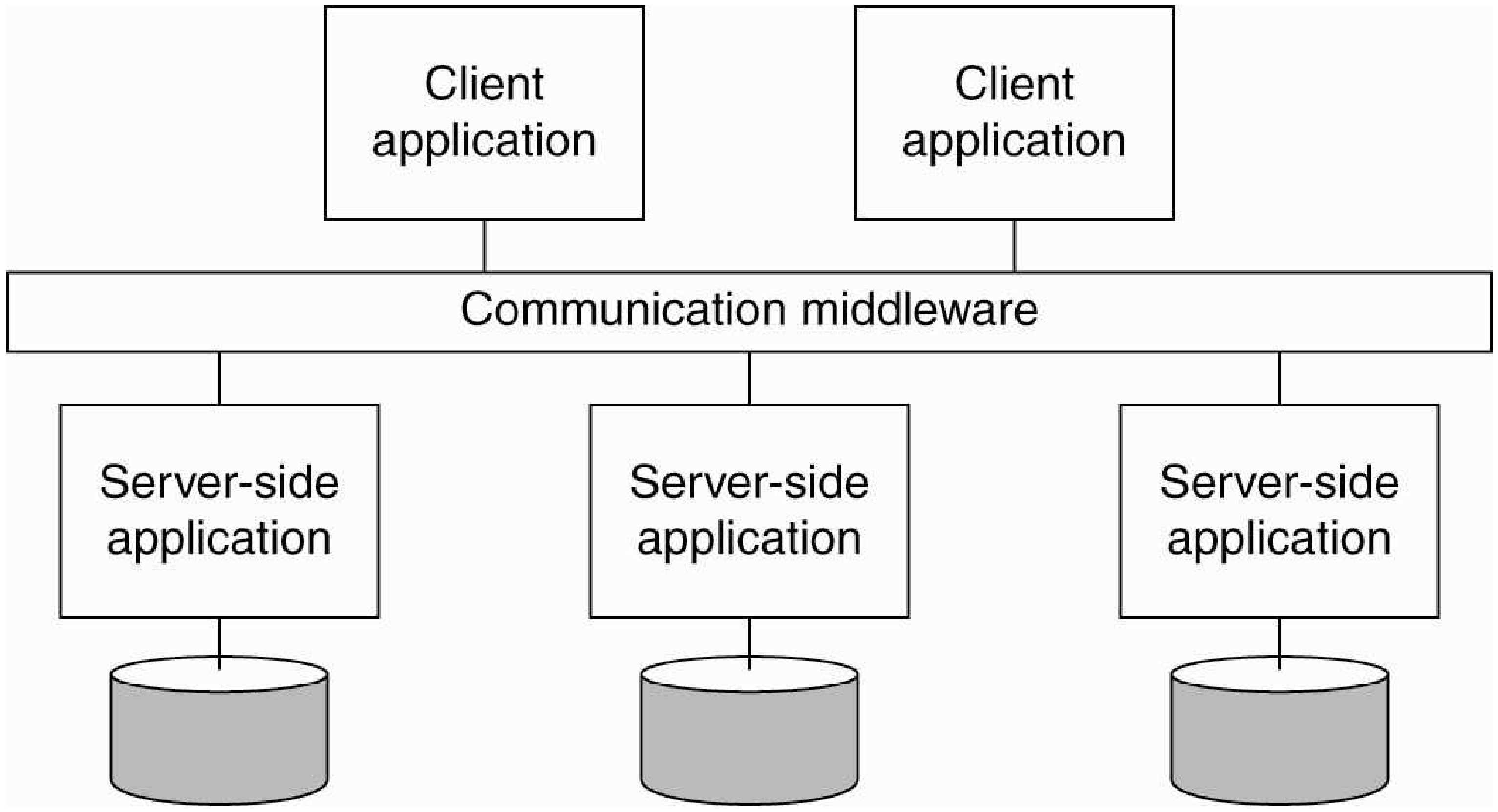


Figure 1-11. Middleware as a communication facilitator in enterprise application integration.

Distributed Pervasive Systems

- Distributed pervasive systems are systems in which instability is the default behavior of them.

- The devices in these systems, are often characterized by being small, battery-powered, mobile, and having only a wireless connection, although not all these characteristics apply to all devices

Distributed Pervasive Systems

Requirements for pervasive systems

- Embrace contextual changes.
- Encourage ad hoc composition.
- Recognize sharing as the default.

Home Systems

- These systems generally consist of one or more personal computers, but more importantly integrate typical consumer electronics such as TVs, audio and video equipments, gaming devices, (smart) phones, PDAs, and other personal wearables into a single system.

Electronic Health Care Systems

- Another important and upcoming class of pervasive systems are those related to (personal) electronic health care.

- With the increasing cost of medical treatment, new devices are being developed to monitor the well-being of individuals and to automatically contact physicians when needed. In many of these systems, a major goal is to prevent people from being hospitalized

Electronic Health Care Systems (cont.)

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 (cont.)

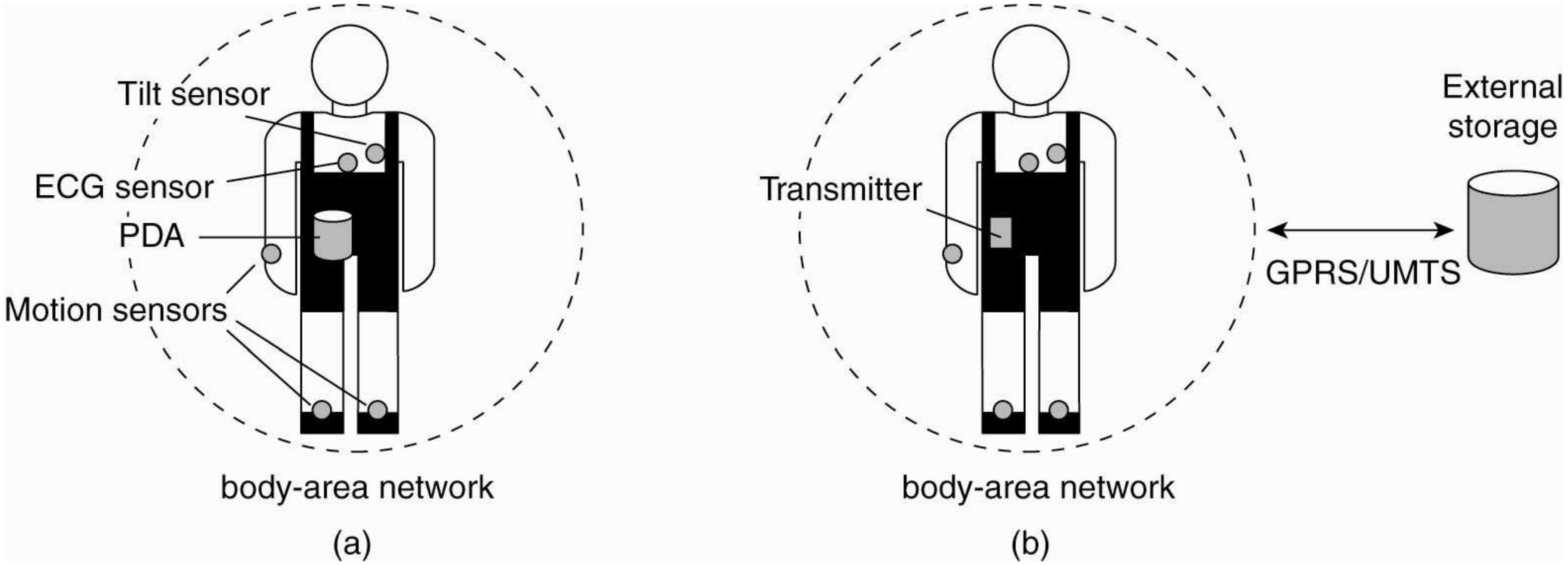


Figure 1-12. Monitoring a person in a pervasive electronic health care system, using (a) a local hub or (b) a continuous wireless connection.

Sensor Networks

- A sensor network typically consists of tens to hundreds or thousands of relatively small nodes, each equipped with a sensing device.
- Most sensor networks use wireless communication, and the nodes are often battery powered.
- Their limited resources, restricted communication capabilities, and constrained power consumption demand that efficiency be high on the list of design criteria.

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 (cont.)

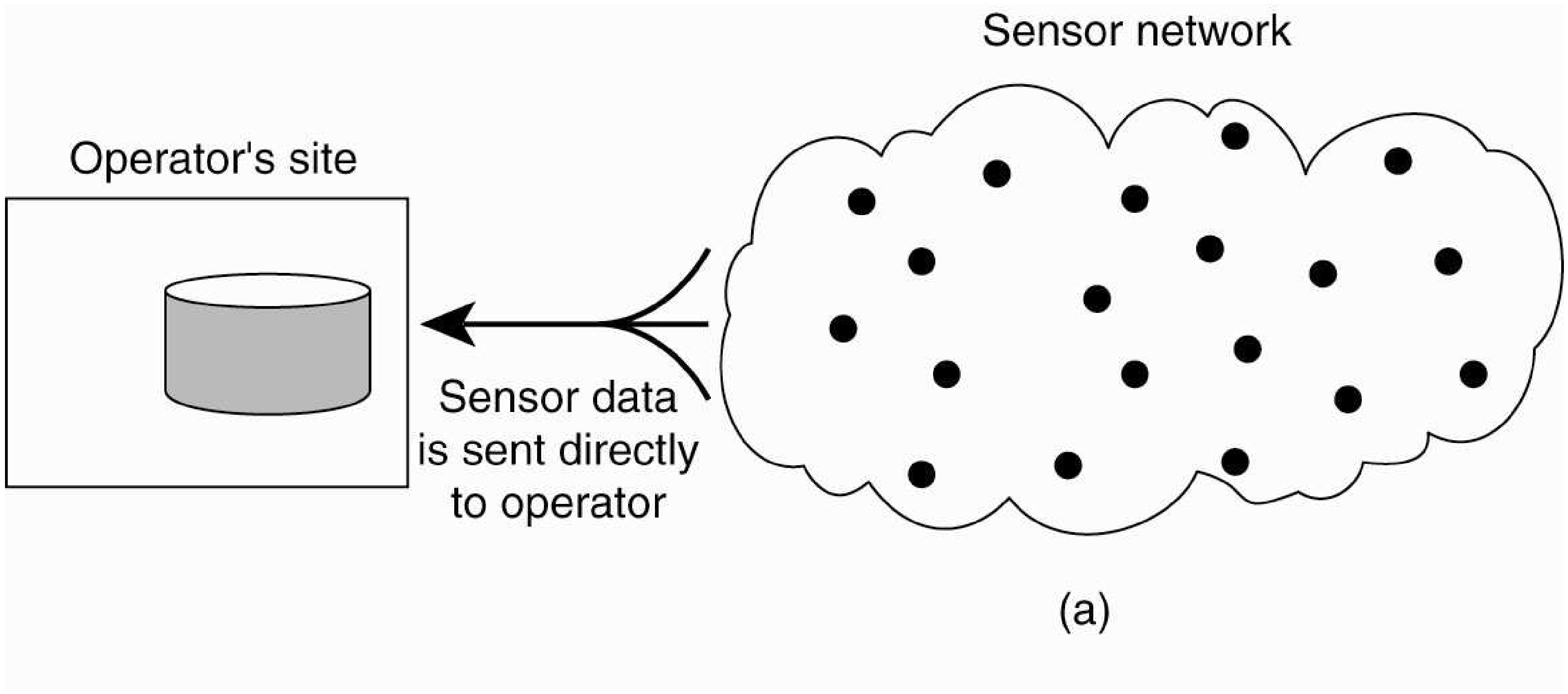


Figure 1-13. Organizing a sensor network database, while storing and processing data (a) only at the operator’s site or …

Sensor Networks (cont.)

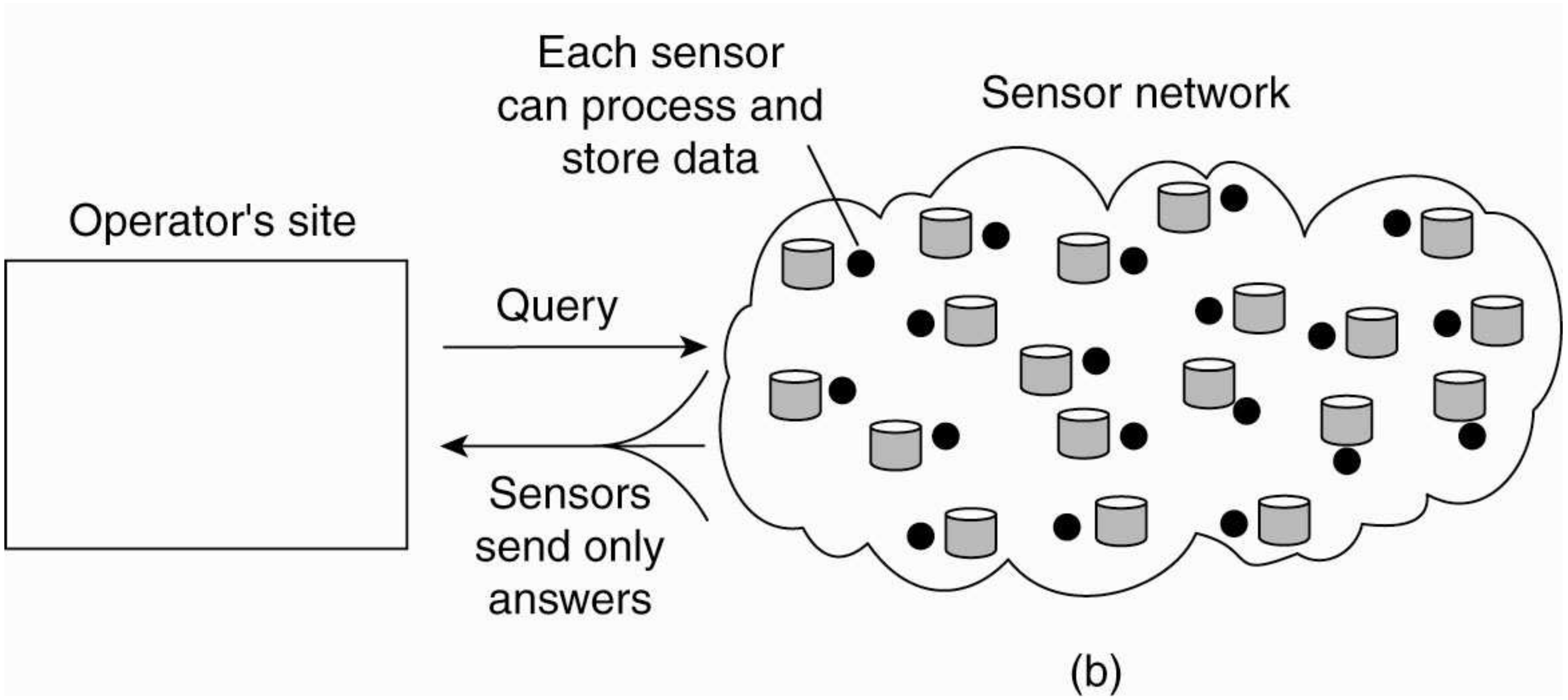

Each sensor can process and store data
Sensor network
Operator's site
Query
Sensors send only answers
(b)

Figure 1-13. Organizing a sensor network database, while storing and processing data … or (b) only at the sensors.