

Distributed Computing



MODULE CODE: CIS435

MODULE NAME: CLOUD COMPUTING

MODULE TEACHER: MD. MINHAJ HOSEN

What is a Distributed System?

2

Definition: A *distributed system* is one in which **components** located at networked computers communicate and **coordinate** their actions only by passing **messages**. This definition leads to the following characteristics of distributed systems:

- Concurrency of components
- Lack of a global 'clock'
- Independent failures of components

Centralized System Characteristics

3

- One component with non-autonomous parts
- Component shared by users all the time
- All resources accessible
- Software runs in a single process
- Single point of control
- Single point of failure

Distributed System Characteristics

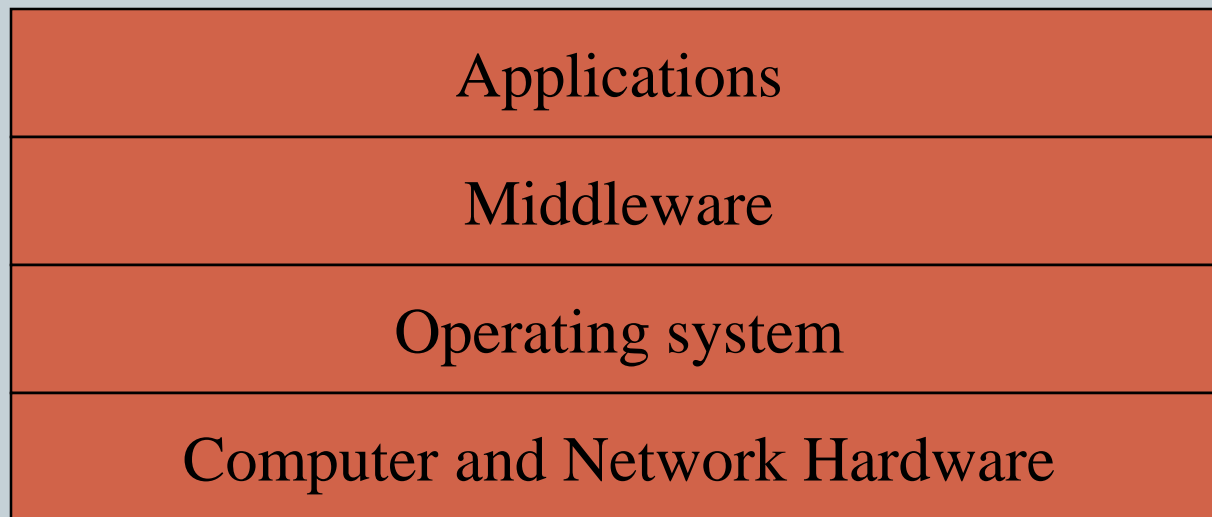
4

- Multiple autonomous components
- Components are not shared by all users
- Resources may not be accessible
- Software runs in concurrent processes on different processors
- Multiple points of control
- Multiple points of failure

Centralized Software Structure

5

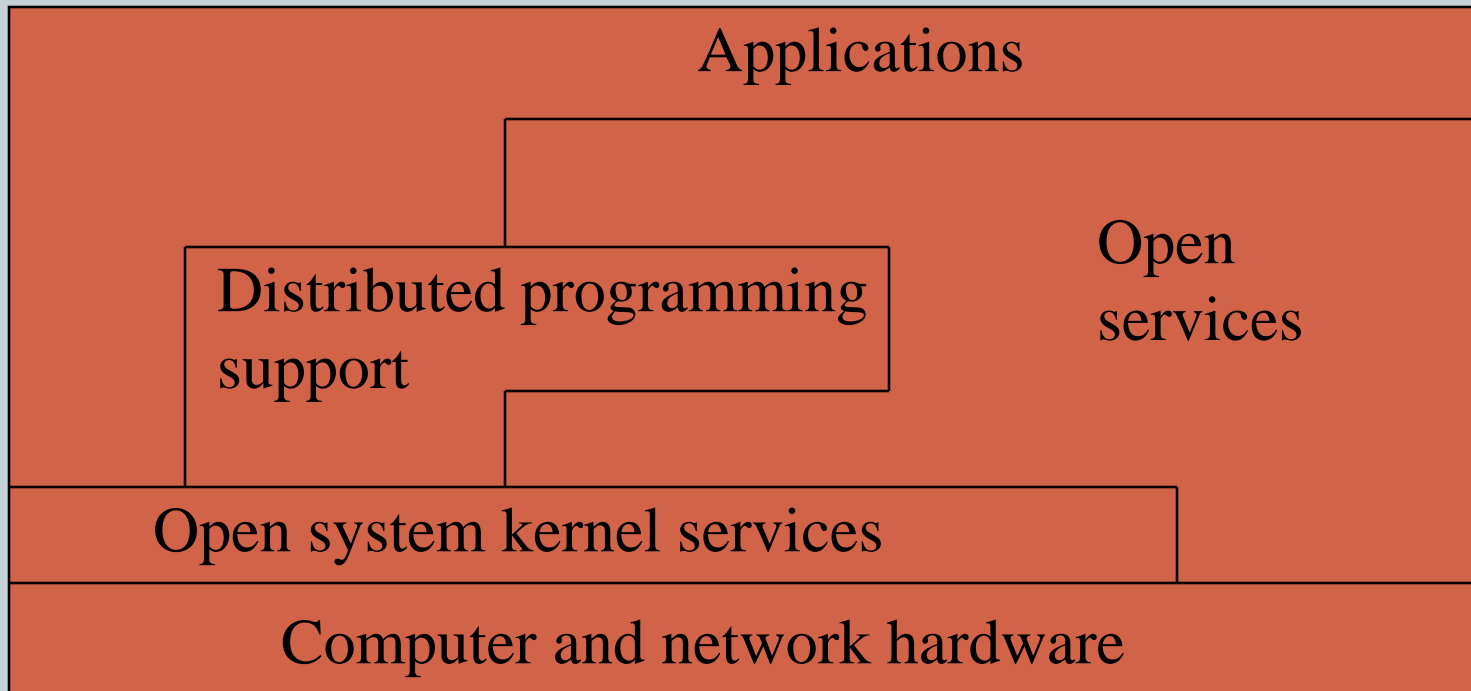
- Layers in centralized computer systems:



Distributed Software Structure

6

- Layers and dependencies in distributed systems:



Common Characteristics

7

Certain common characteristics can be used to assess distributed systems:

- Heterogeneity
- Openness
- Security
- Scalability
- Failure Handling
- Concurrency
- Transparency

Heterogeneity

8

- Variety and differences in
 - Networks
 - Computer hardware
 - Operating systems
 - Programming languages
 - Implementations by different developers
- *Middleware* as software layers to provide a programming abstraction as well as masking the heterogeneity of the underlying networks, hardware, OS, and programming languages (e.g., CORBA).
- *Mobile Code* to refer to code that can be sent from one computer to another and run at the destination (e.g., Java applets and Java *virtual machine*).

Openness

9

- Openness is concerned with extensions and improvements of distributed systems.
- Detailed interfaces of components need to be published.
- New components have to be integrated with existing components.
- Differences in data representation of interface types on different processors (of different vendors) have to be resolved.

Security

10

- In a distributed system, clients send requests to access data managed by servers, resources in the networks:
 - Doctors requesting records from hospitals
 - Users purchase products through electronic commerce
- Security is required for:
 - Concealing the contents of messages: security and privacy
 - Identifying a remote user or other agent correctly (authentication)
- New challenges:
 - Denial of service attack
 - Security of mobile code

4.4 Scalability

11

- Adaptation of distributed systems to
 - accommodate more users
 - respond faster (this is the hard one)
- Usually done by adding more and/or faster processors.
- Components should not need to be changed when scale of a system increases.
- Design components to be scalable!

Failure Handling (Fault Tolerance)

12

- Hardware, software and networks fail!
- Distributed systems must maintain *availability* even at low levels of hardware/software/network *reliability*.
- Fault tolerance is achieved by
 - recovery
 - redundancy

Concurrency

13

- Components in distributed systems are executed in concurrent processes.
- Components access and update shared resources (e.g. variables, databases, device drivers).
- Integrity of the system may be violated if concurrent updates are not coordinated.
 - Lost updates
 - Inconsistent analysis

Transparency

14

- Distributed systems should be perceived by users and application programmers as a whole rather than as a collection of cooperating components.
- Transparency has different aspects.
- These represent various properties that distributed systems should have.

Transparency

15

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 may be shared by several competitive users
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or on disk

6. Advantages

16

- **Sharing Data** : There is a provision in the environment where user at one site may be able to access the data residing at other sites.
- **Autonomy** : Because of sharing data by means of data distribution each site is able to retain a degree of control over data that are stored locally.
- **Availability** : If one site fails in a distributed system, the remaining sites may be able to continue operating. Thus a failure of a site doesn't necessarily imply the shutdown of the System.

Disadvantages

17

- **Software Development Cost**
- **Greater Potential for Bugs**
- **Increased Processing Overhead**

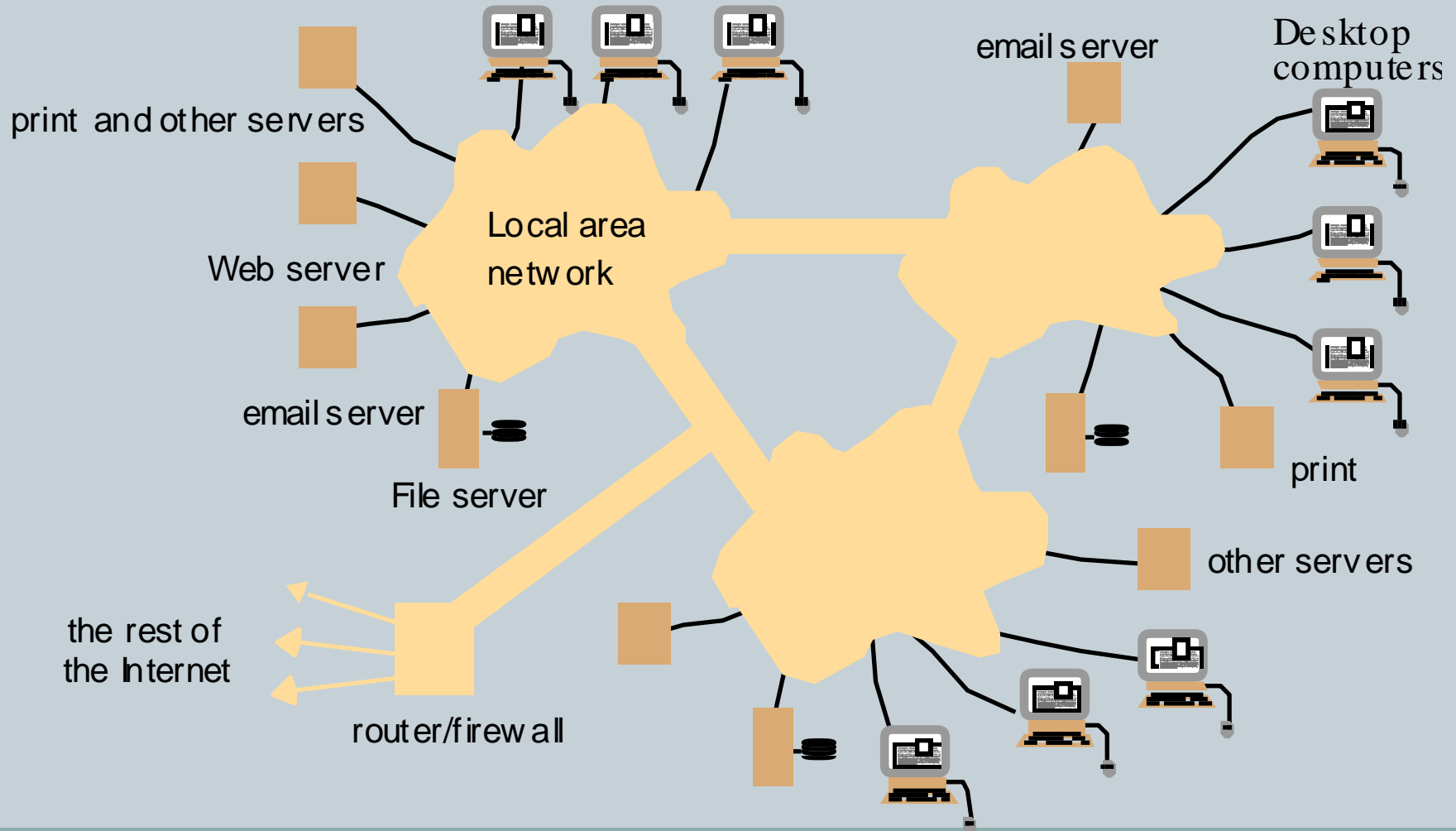
Examples of Distributed Systems

18

- Local Area Network and Intranet
- Database Management System
- Automatic Teller Machine Network
- Internet/World-Wide Web
- Mobile and Ubiquitous Computing

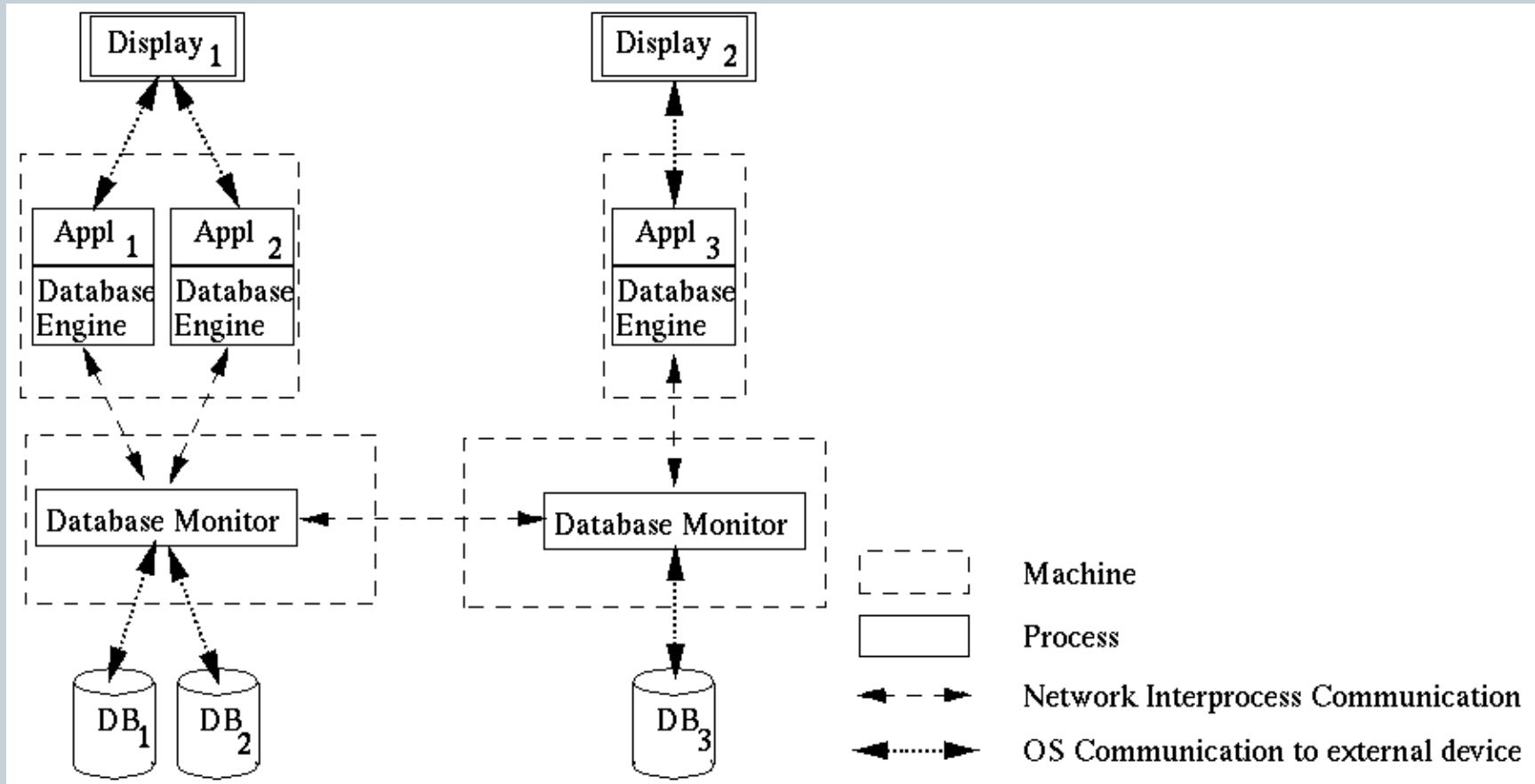
Local Area Network

19



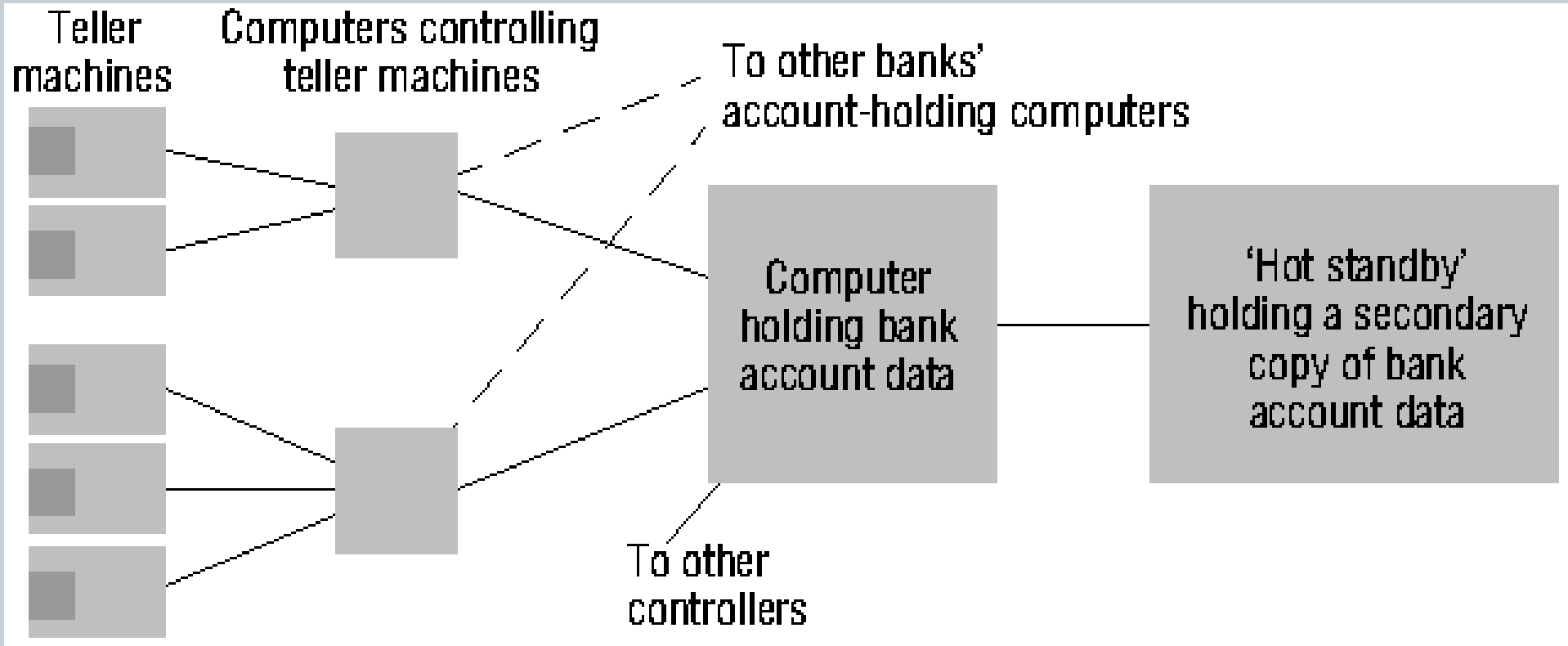
Database Management System

20



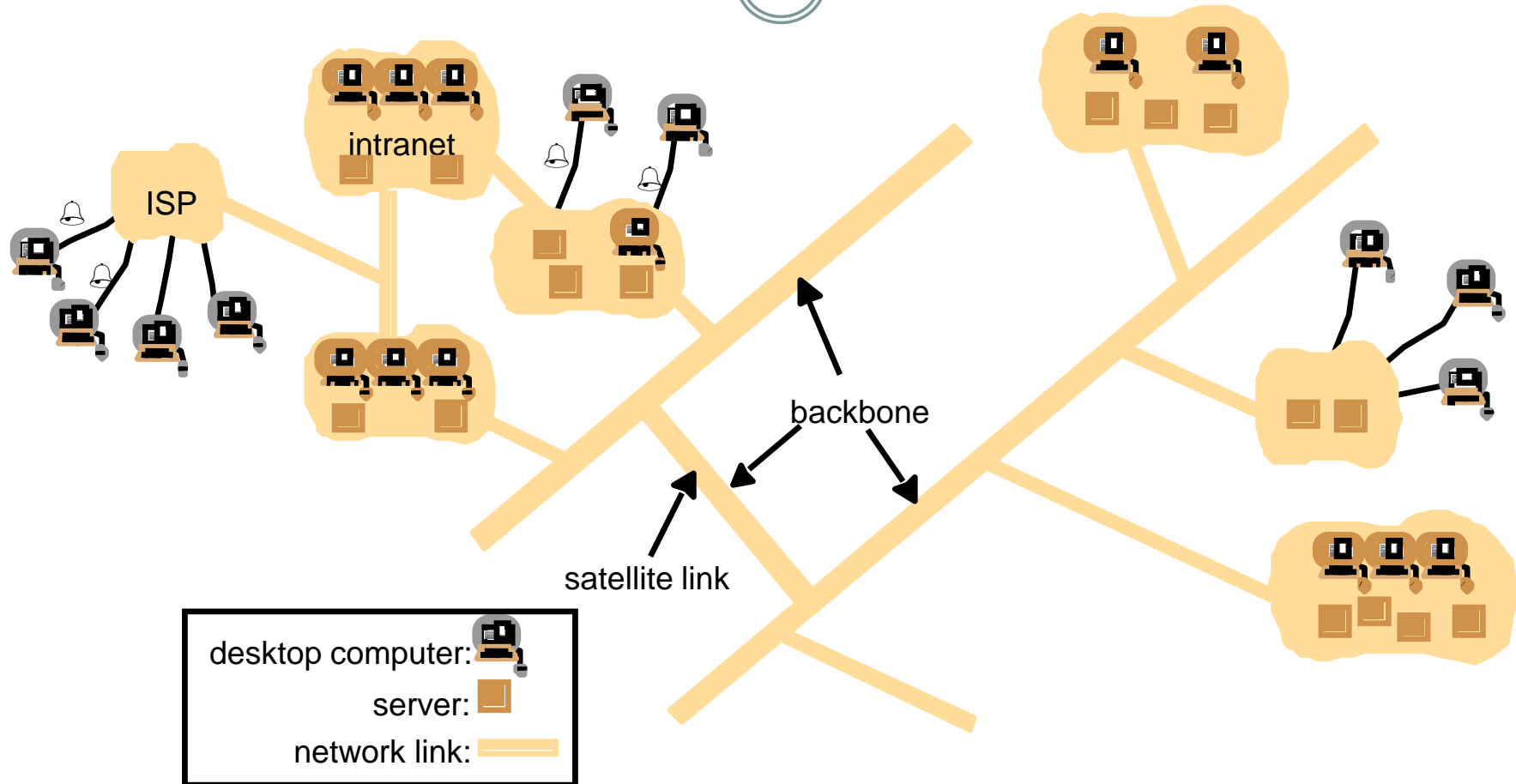
Automatic Teller Machine Network

21



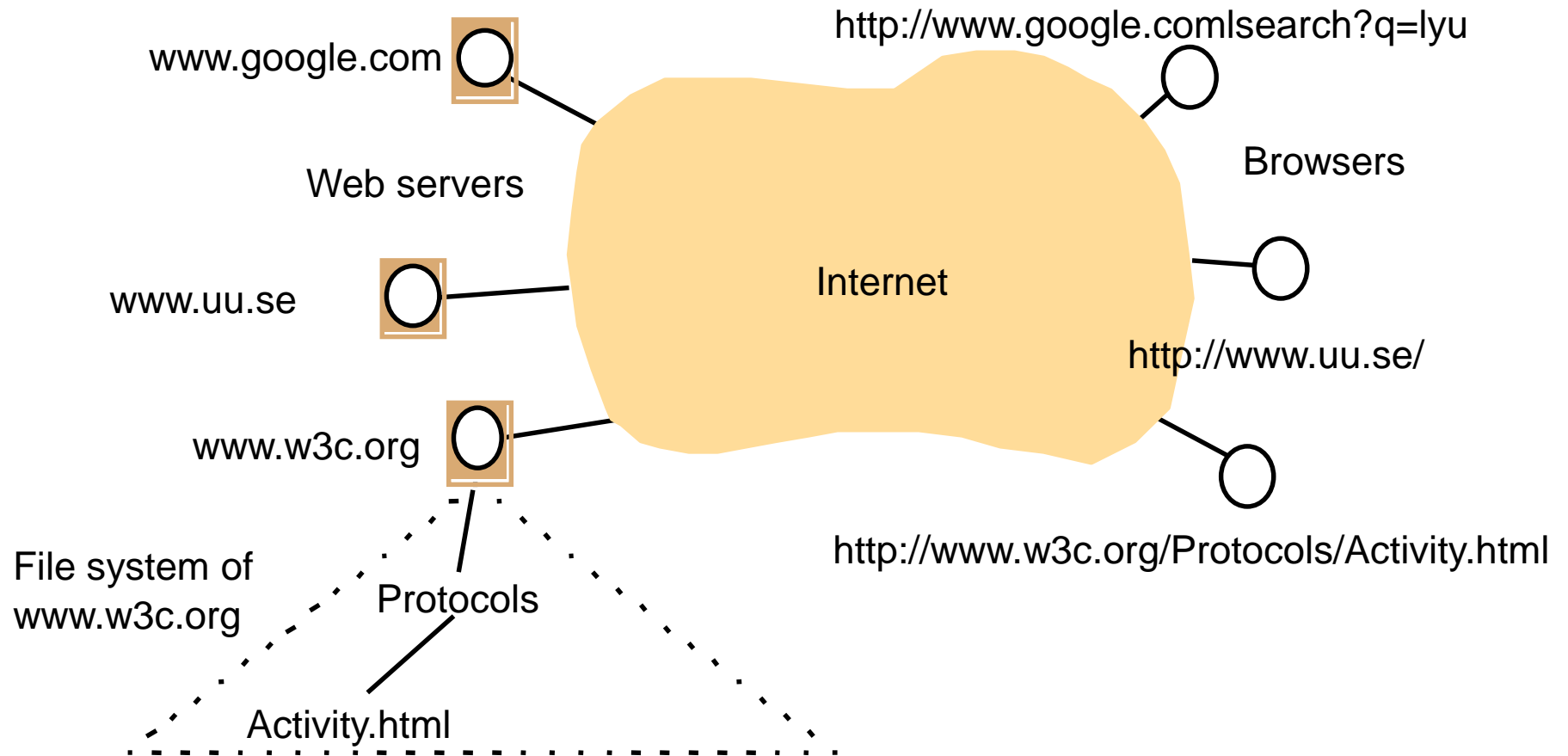
Internet

22



Web Servers and Web Browsers

23



Mobile and Ubiquitous Computing

24

