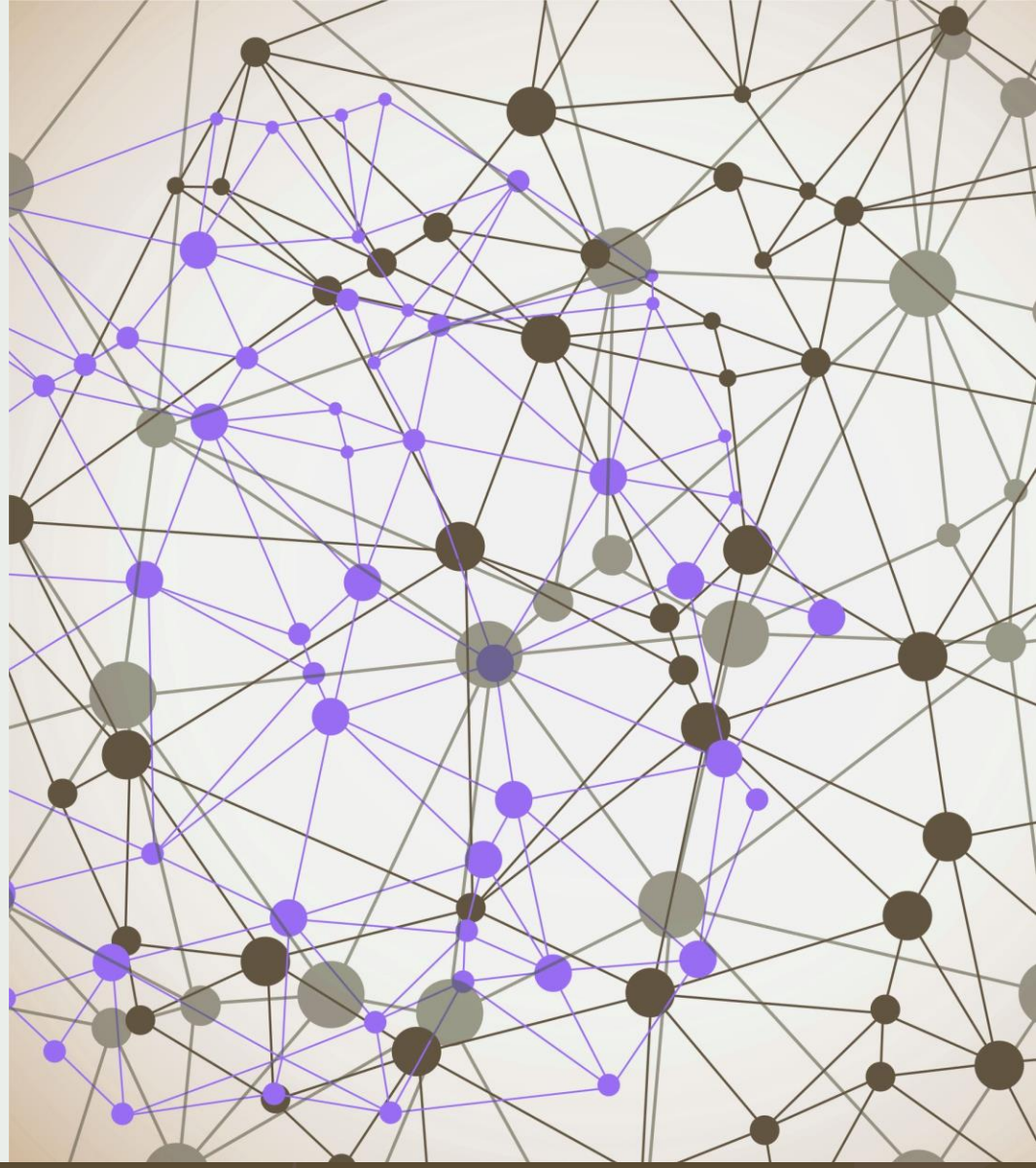
Introduction

Chapter 1



Distributed Systems

Definition: A *distributed system* is one in which components located at networked computers communicate and coordinate their actions only by passing the messages.

This definition leads to the following consequences of distributed systems:

Concurrency of components	Synchronization of a global 'clock'	Failure handling of independent components

Distributed Systems

- **Aim: A piece of software that ensure that, a collection of independent computers appears to its users as a single coherent system**
- **Two aspects:**
  - **Independent Computers**
  - **Single system**
- **Communication Way: Message**

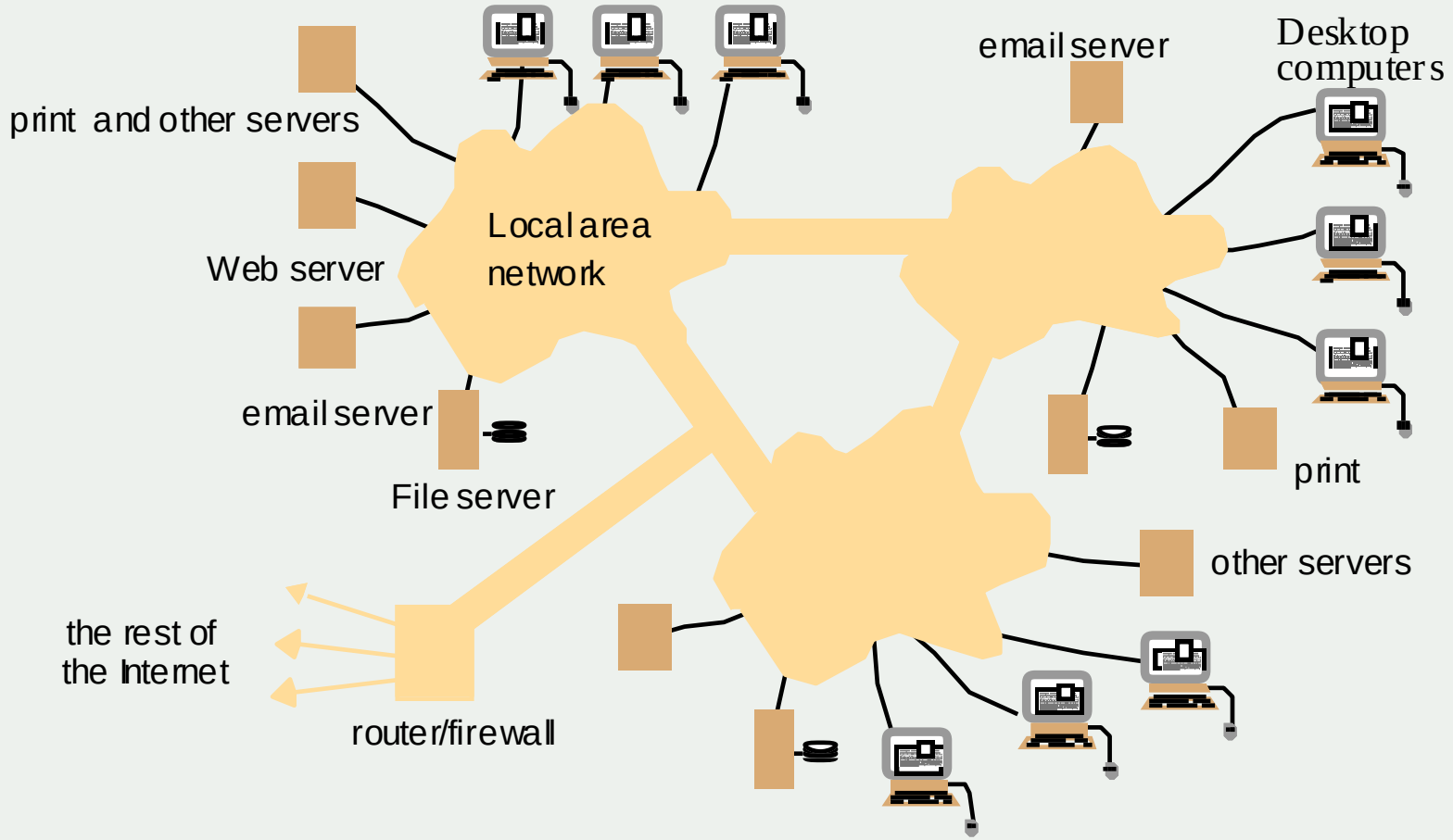
Examples of Distributed Systems

Intranet

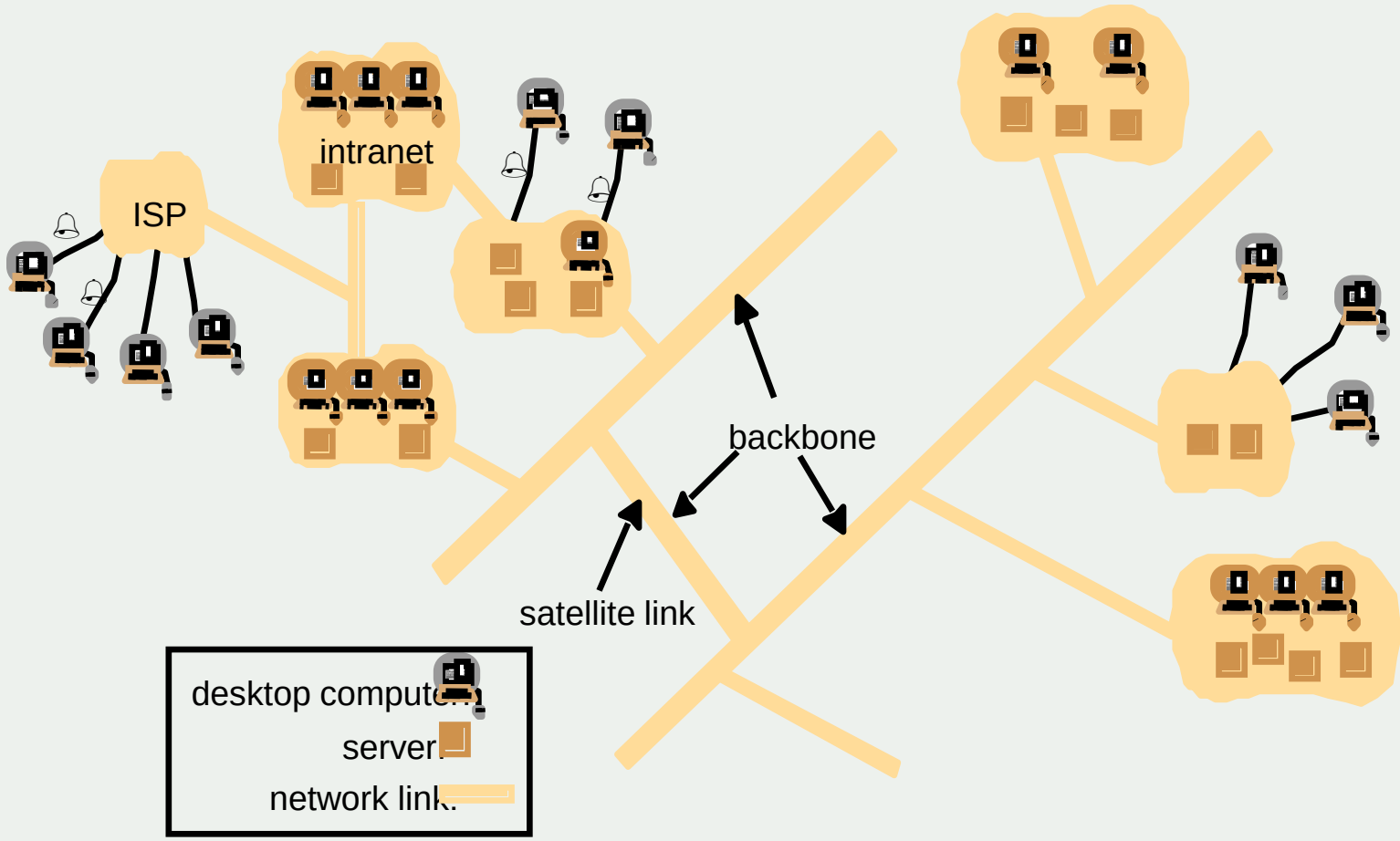
Internet

Mobile and Ubiquitous Computing

Example 1: Intranet



Example 2: Internet

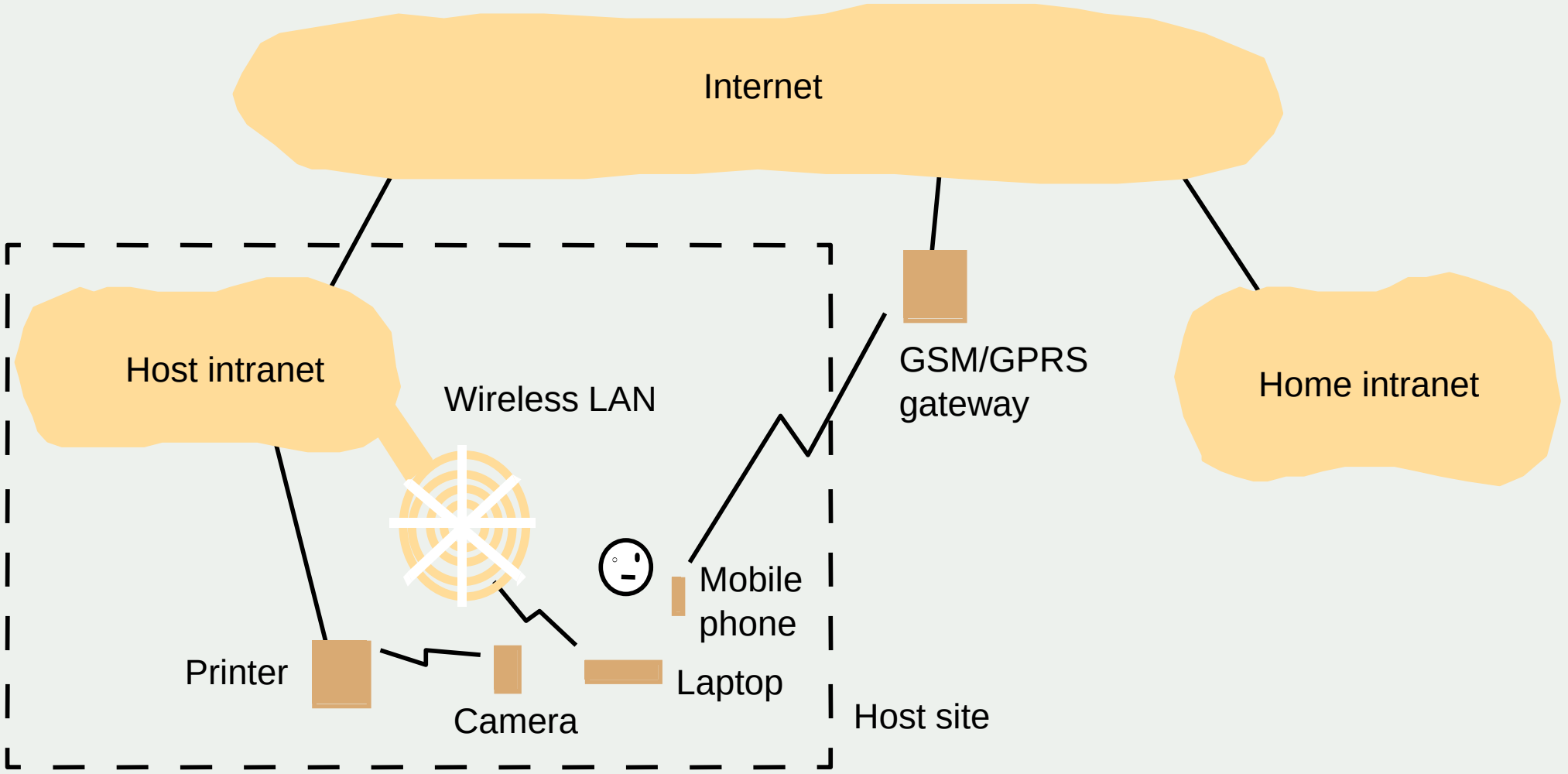


Example 3

Mobile computing

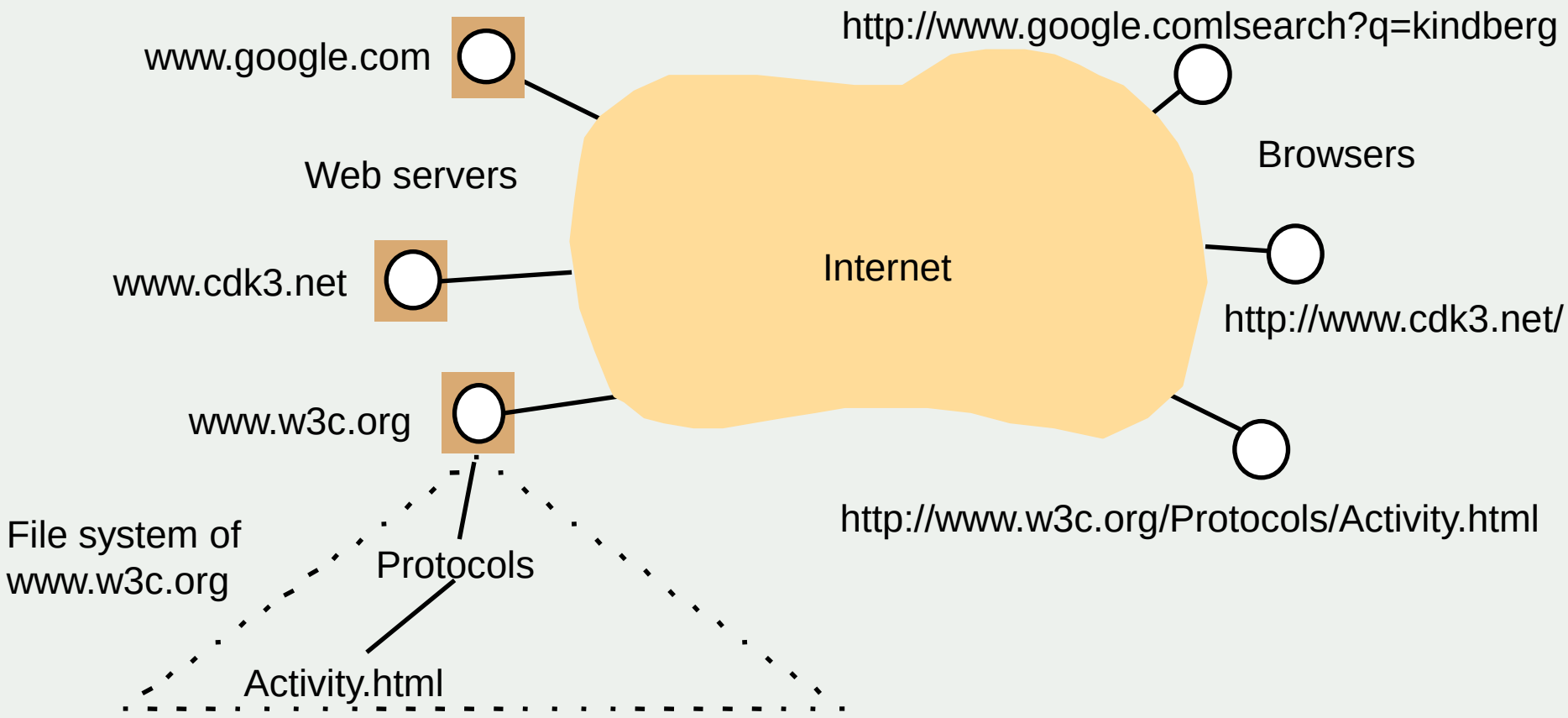
Ubiquitous computing

Nomadic Computing: performance of computing task while the user is on the move, or visiting places other than their usual environment
Location-aware or context-aware computing


Internet
Host intranet
Wireless LAN
GSM/GPRS
gateway
Home intranet
Mobile
phone
Printer
Laptop
Camera
Host site

Resource Sharing and Web

- Web Search
- Computer Supported Cooperative Working (CSCW)
- Service
- Server
- Client
- Remote Invocation



Web Server and Browsers

World Wide Web (WWW)

- Web is open system to publish and share the resources
- Main Components
  - HTML
  - URLs : *http:// servername [:port] [/pathName] [?query] [#fragment]*
  - HTTP: Request-Reply, Content types, One resource per request, Simple access control
- Dynamic Pages
- Web Services
- Semantic Web

```
<IMG SRC = "http://www.cdk5.net/WebExample/Images/earth.jpg">          1
<P>                                                                    2
Welcome to Earth! Visitors may also be interested in taking a look at the  3
<A HREF = "http://www.cdk5.net/WebExample/moon.html">Moon</A>.         4
</P>                                                                   5
```

http://www.cdk5.net
http://www.w3.org/standards/faq.html#conformance
http://www.google.com/search?q=obama

These can be broken down as follows:

Server DNS name	*Path name*	*Query*	*Fragment*
www.cdk5.net	(default)	(none)	(none)
www.w3.org	standards/faq.html	(none)	intro
www.google.com	search	q=obama	(none)

Additional Examples

MASSIVELY MULTIPLAYER ONLINE GAMES

FINANCIAL TRADING

MASSIVE OPEN ONLINE COURSE

Trends in DS

- Pervasive Networking and Modern Internet
- Distributed Multimedia Systems
- Distributed Computing as a Utility and Grid

Goals of DS

Resource Sharing

Openness

Transparency

Scalability

Concurrency

Challenges of Distributed Systems

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

Heterogeneity

- 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).
- It also provides a uniform computational model for use by the programmers of servers and distributed applications.
- *Mobile Code* Mobile code is used to be run on heterogeneous computers. To get it done, the virtual machine approach provides a way of making code executable on any hardware: the compiler for a particular language generates code for a virtual machine instead of a particular hardware order code. (e.g., Java applets and Java *virtual machine*).

Openness

Openness is concerned with extensions and improvements of distributed systems.

Detailed interfaces of components 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

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

Scalability

- 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 need to be scalable!
- Major Challenges:
  - Controlling the cost of Physical Resources
  - Controlling the performance loss
  - Preventing software resources running out
  - Avoiding performance bottlenecks

Failure Handling (Fault Tolerance)

- Hardware, software and networks fail!
- Distributed systems must maintain *availability* even at low levels of hardware/software/network *reliability*.
- Fault tolerance is achieved by
  - Detecting failures
  - Masking failures
  - Tolerating failures
  - Recovery from failures
  - Redundancy

Concurrency

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

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: e.g., complexity

These represent various properties that distributed systems should have.

Access Transparency

Enables local and remote information objects to be accessed using identical operations.

Example: File system operations in NFS.

Example: Navigation in the Web.

Example: SQL Queries

Location Transparency

Enables information objects to be accessed without knowledge of their location.

Example: File system operations in NFS

Example: Pages in the Web

Example: Tables in distributed databases

Concurrency Transparency

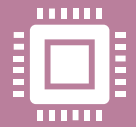
Enables several processes to operate concurrently using shared information objects without interference between them.

Example: NFS

Example: Automatic teller machine network

Example: Database management system

Replication Transparency



Enables multiple instances of information objects to be used to increase reliability and performance without knowledge of the replicas by users or application programs

Example: Distributed DBMS

Example: Mirroring Web Pages.

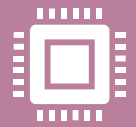
Failure Transparency

Enables the concealment of faults

Allows users and applications to complete their tasks despite the failure of other components.

Example: Database Management System

Mobility Transparency



Allows the movement of information objects within a system without affecting the operations of users or application programs

Example: NFS

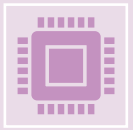
Example: Web Pages

Performance Transparency

Allows the system to be reconfigured to improve performance as loads vary. (If it can choose from a set of processors, it will delegate it to the fastest processor that has the lowest load. In this way it achieves an even better performance.)

Example: Distributed make.

Scaling Transparency



Allows the system and applications to expand in scale without change to the system structure or the application algorithms.

Example: World-Wide-Web

Example: Distributed Database

Advantages of DS over CS

Economics

Reliability

Speed

Incremental Growth

Data Sharing

Resource Sharing

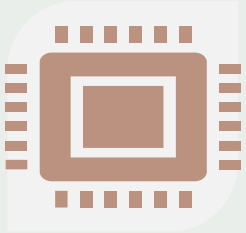
Communication

Flexibility

Major Facts

DIFFERENCE BETWEEN DISTRIBUTED SYSTEMS AND DISTRIBUTED COMPUTING

DIFFERENCE BETWEEN DISTRIBUTED AND PARALLEL COMPUTING



DISTRIBUTED APPLICATIONS (D-APPS)

DISTRIBUTED OPERATING SYSTEMS

DISTRIBUTED DATABASE

- **Distributed system**: a collection of independent **computers** that are connected with an interconnection network.

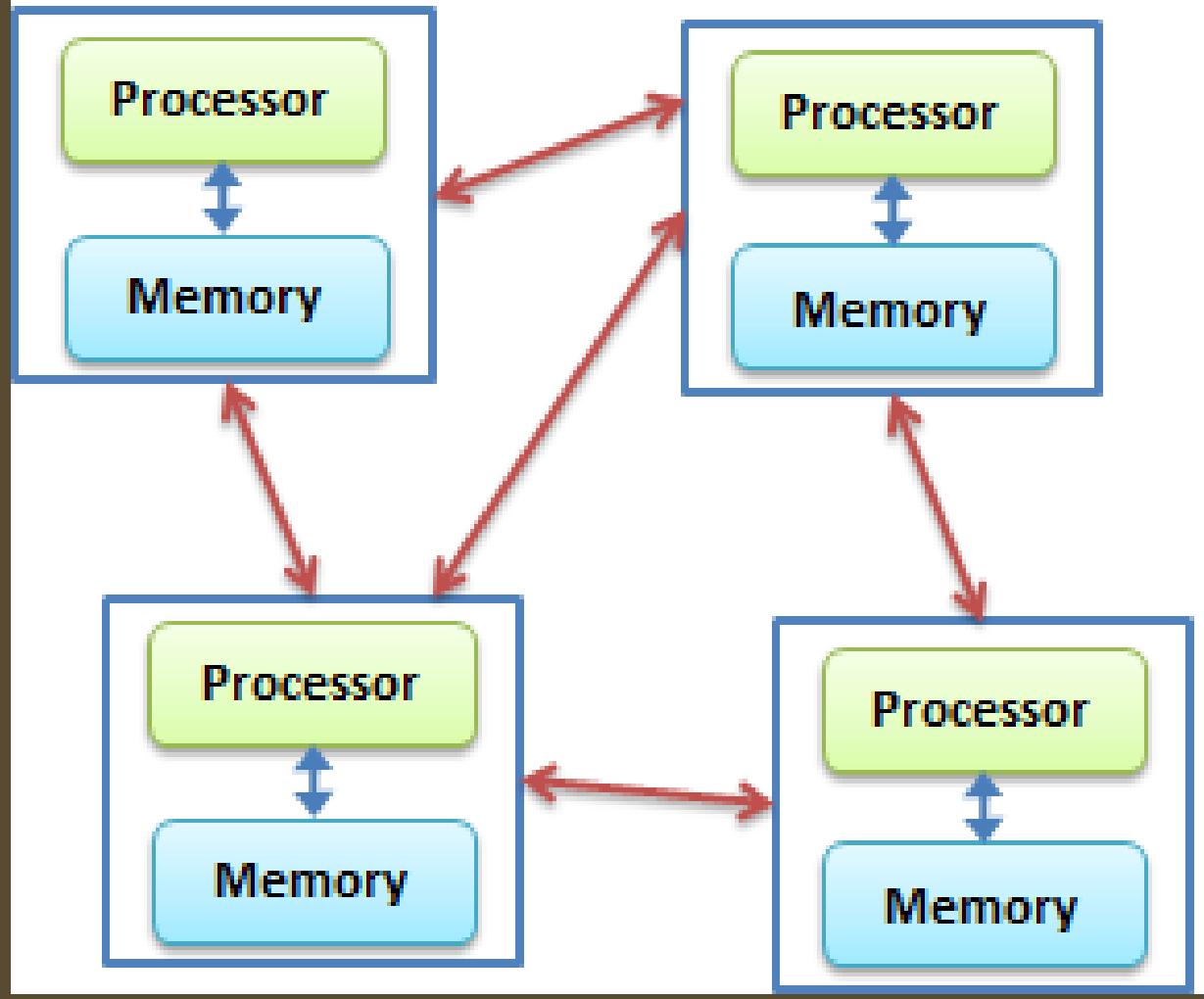
- **Distributed computing**: a method of computer processing in which different parts of a computer program are run on two or more **computers** that are communicating with each other over a network.

- **Distributed Applications** are software **applications** that are stored mostly on cloud computing platforms and that run on multiple systems simultaneously. The systems run on the same network and communicate with each other in an effort to complete a specific task or command.

- A **distributed operating system** is a software over a collection of independent, networked, communicating, and physically separate computational nodes. They handle jobs which are serviced by multiple CPUs. Each individual node holds a specific software subset of the global aggregate **operating system**.

- The main **difference between parallel and distributed computing** is that **parallel computing** allows multiple processors to execute tasks simultaneously while **distributed computing** divides a single task **between** multiple computers to achieve a common goal.

Distributed Computing



Parallel Computing