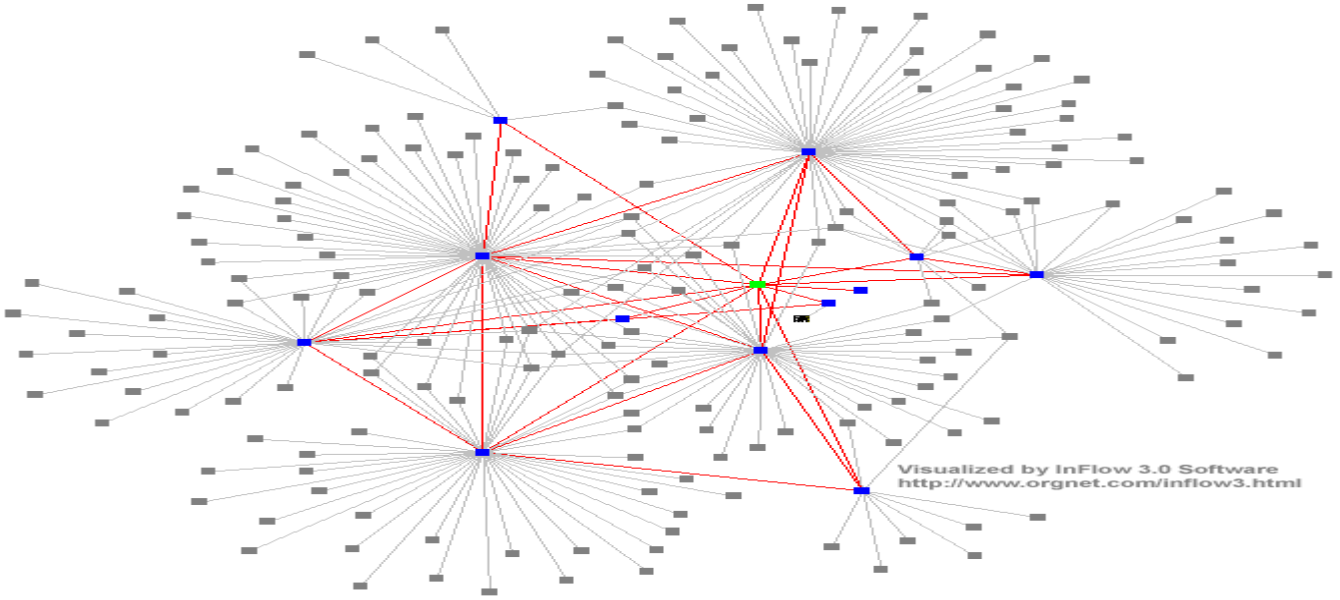
Introduction to Distributed Systems



Dr. Sunny Jeong. spjeong@uic.edu.hk

Mr. Bill Junru ZHONG billjrzhong@uic.edu.hk

With Thanks to Prof. G. Coulouris, Prof. A.S. Tanenbaum and Prof. S.C Joo

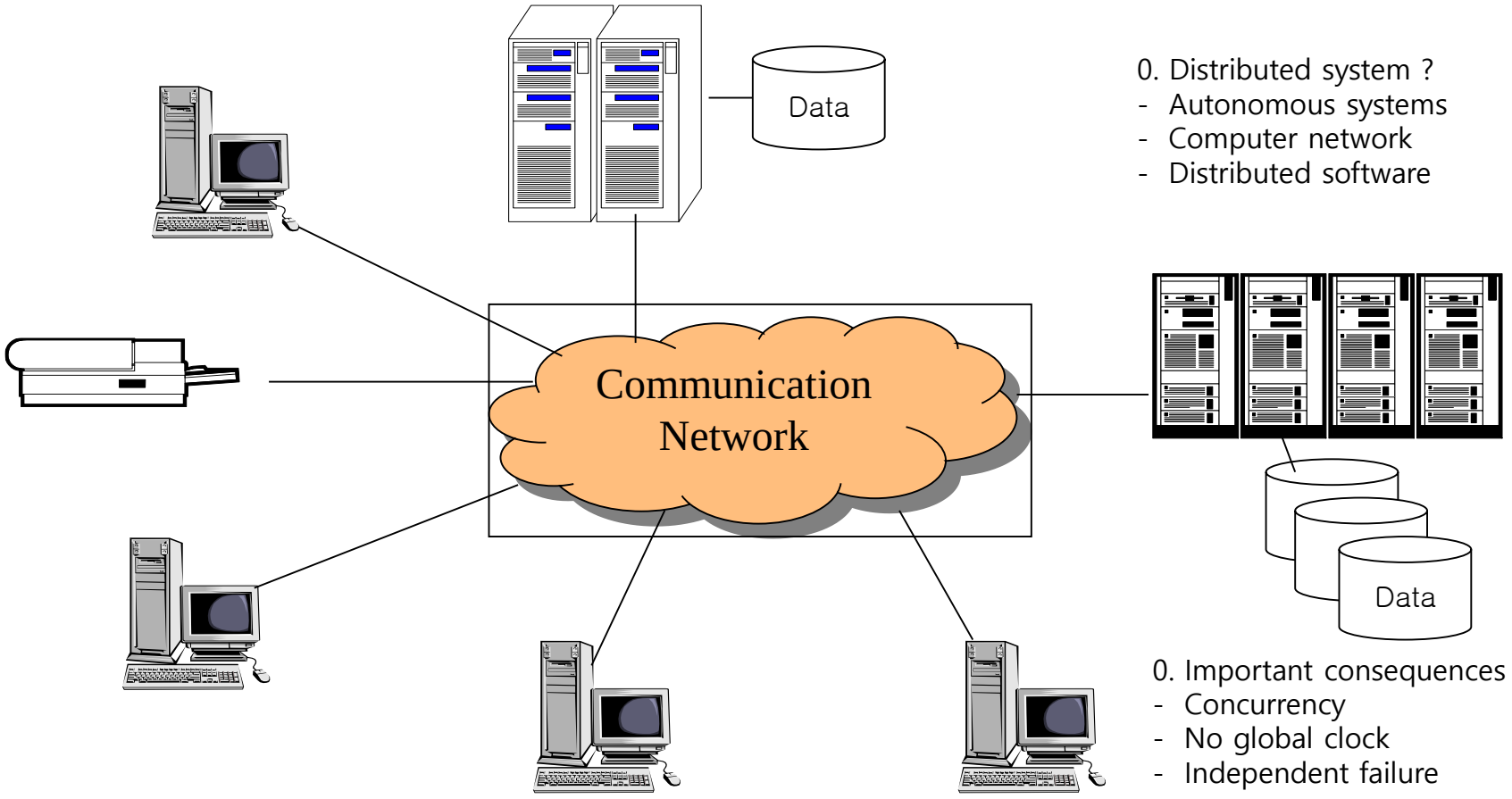
Distributed Systems

- A distributed system is:
  - Distributed system is one in which components located at networked computers communicated and coordinate their actions only by passing message **–G. Coulouris**
  - A collection of independent computers that appears to its users as a single coherent system. **- S. Tanenbaum**
- It leads to concurrency of components, lack of a global clock and independent failures of components
- - Characteristics of distributed systems
  - Concurrency
  - No Global clock
  - Independent failures

Why Distributed Systems?

- Main features
  - Geographical distribution of autonomous computers
  - Communication through cable/fiber/wireless/...connections
  - A collection of independent computers that appears to its users as **a single coherent system logically(called a single view system).**
  - Advantages
    - interaction, co-operation, and sharing of resources
  - Benefits
    - reduced costs, improved availability and performance
    - Scalability, resource sharing, fault tolerance.

Distributed Systems



0. Distributed system ?
- Autonomous systems
- Computer network
- Distributed software

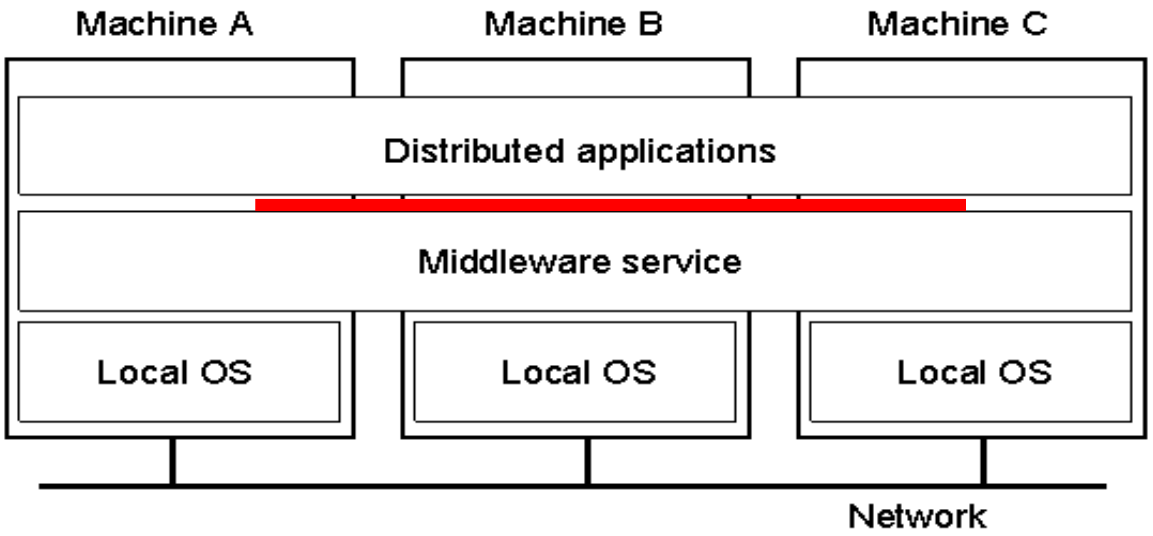
0. Important consequences
- Concurrency
- No global clock
- Independent failure

Distributed Systems- consequences

- Concurrency
  - web pages /
  - sharing resources
- No Global clock
  - No single global notion of the correct time
- Independent failures
  - slow network /unexpected termination
  - each component of the system can fail independently

Introduction of Distributed Systems

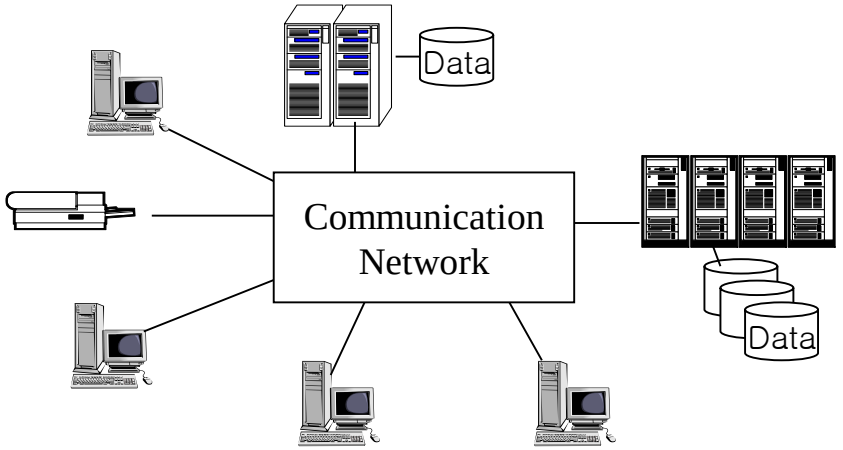
- Definitions of Distributed Systems
  - A collection of independent computers that appears to its users as **a single coherent system**.



  - A distributed system organized as **middleware**.
    - Note that the middleware layer extends over multiple machines.

D.S and Processes

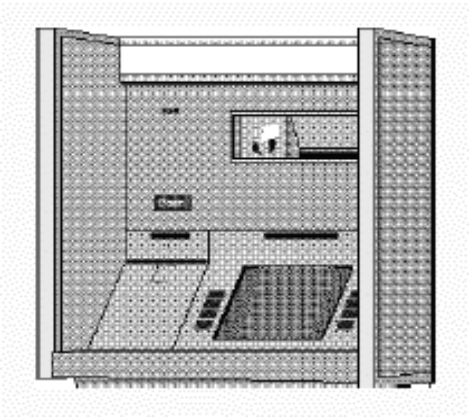
- A **distributed system** is a collection of autonomous computers interconnected by a computer network and equipped with distributed system software to form an integrated computing facility.



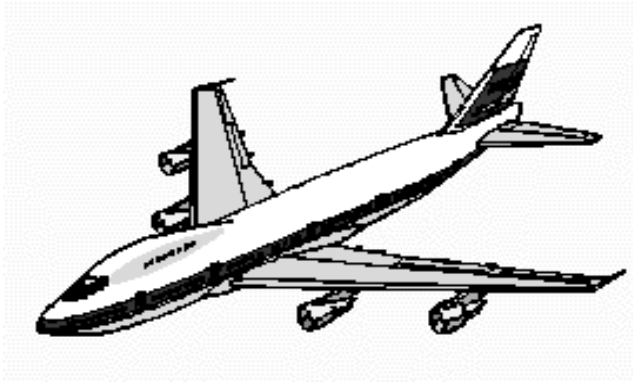
- Processes
  - execute **concurrently**
  - interact in order to **co-operate to achieve a common goal**
  - **co-ordinate their activities and exchange information** by means of messages transferred over a communication network

Importance of Distributed Computing

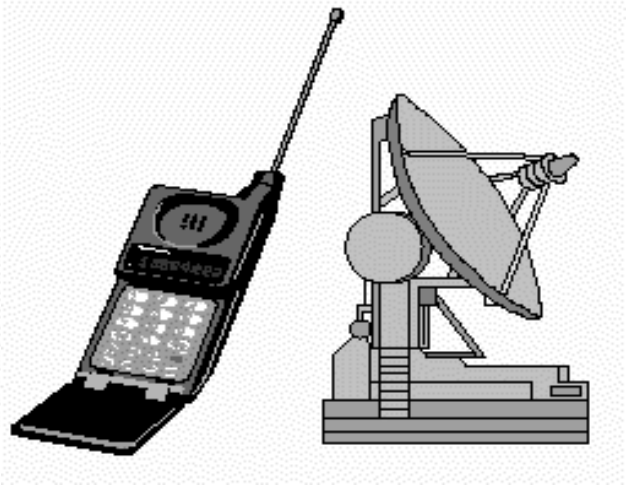
- Distributed (computer) systems are **critical for** functioning of many organizations



Banks



Transport



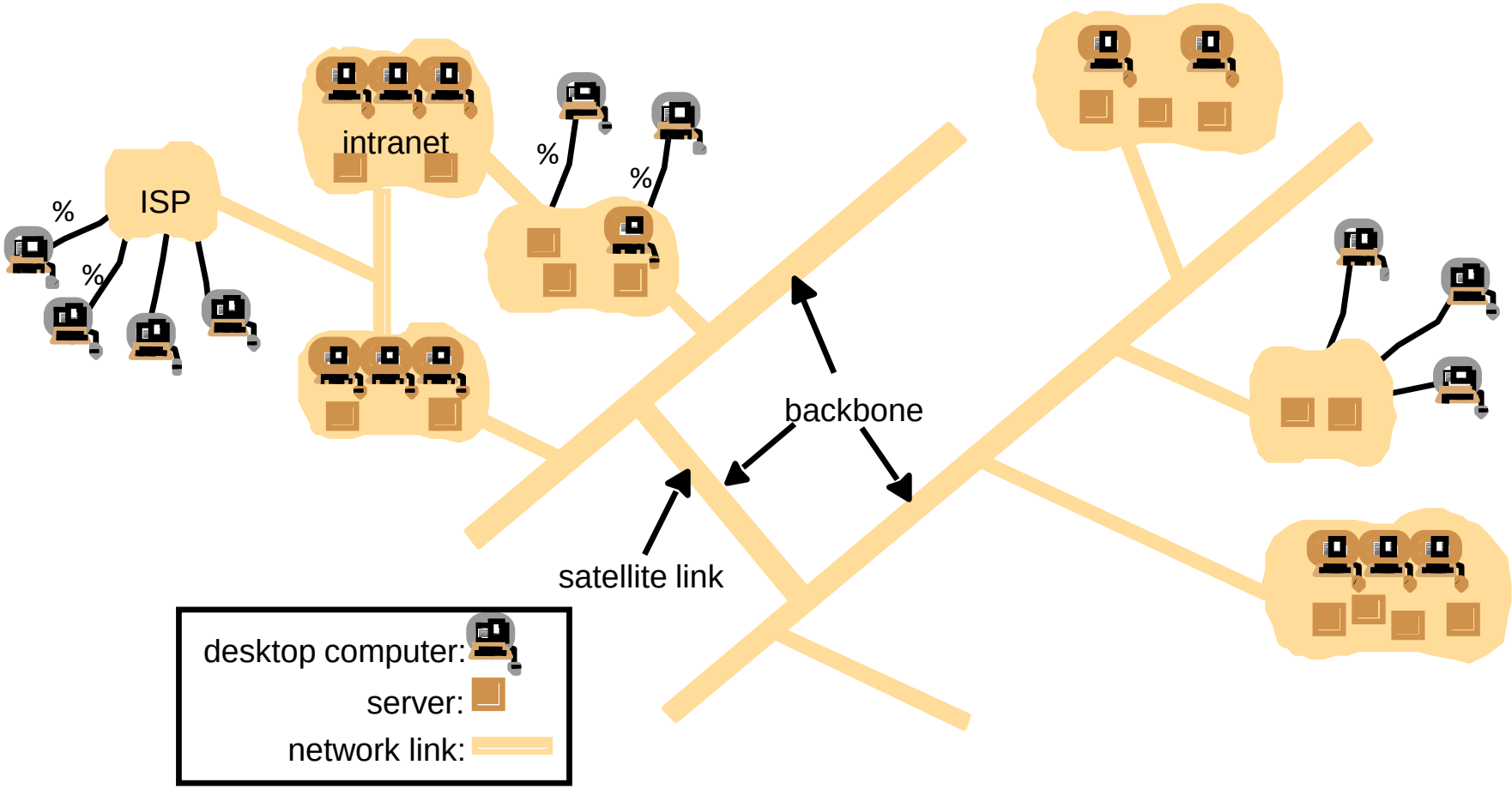
Telecommunications

- Distributed Application
  - A **set of processes** that are distributed across a network of machines and work together as an ensemble to solve a common problem

Typical examples

- **The Internet**
  - global network of interconnected computers which communicate through IP protocols
  - A vast interconnected collection of computer networks of many different types.
- **An Intranets**
  - a separately **administered network with a boundary** that allows to enforce local security policies
  - A portion of the Internet. Router/firewall exclusive File services/ Impeding Firewalls/The cost of installation
- **Mobile and ubiquitous computing**
  - laptops, PDAs, mobile phones, printers, home devices, ...
  - technological advance in device miniaturization and wireless networking.
- **World-Wide Web**
  - **system for publishing and accessing resources and services across the Internet**
  - **HTML/URLs/HTTP/Dynamic Pages**

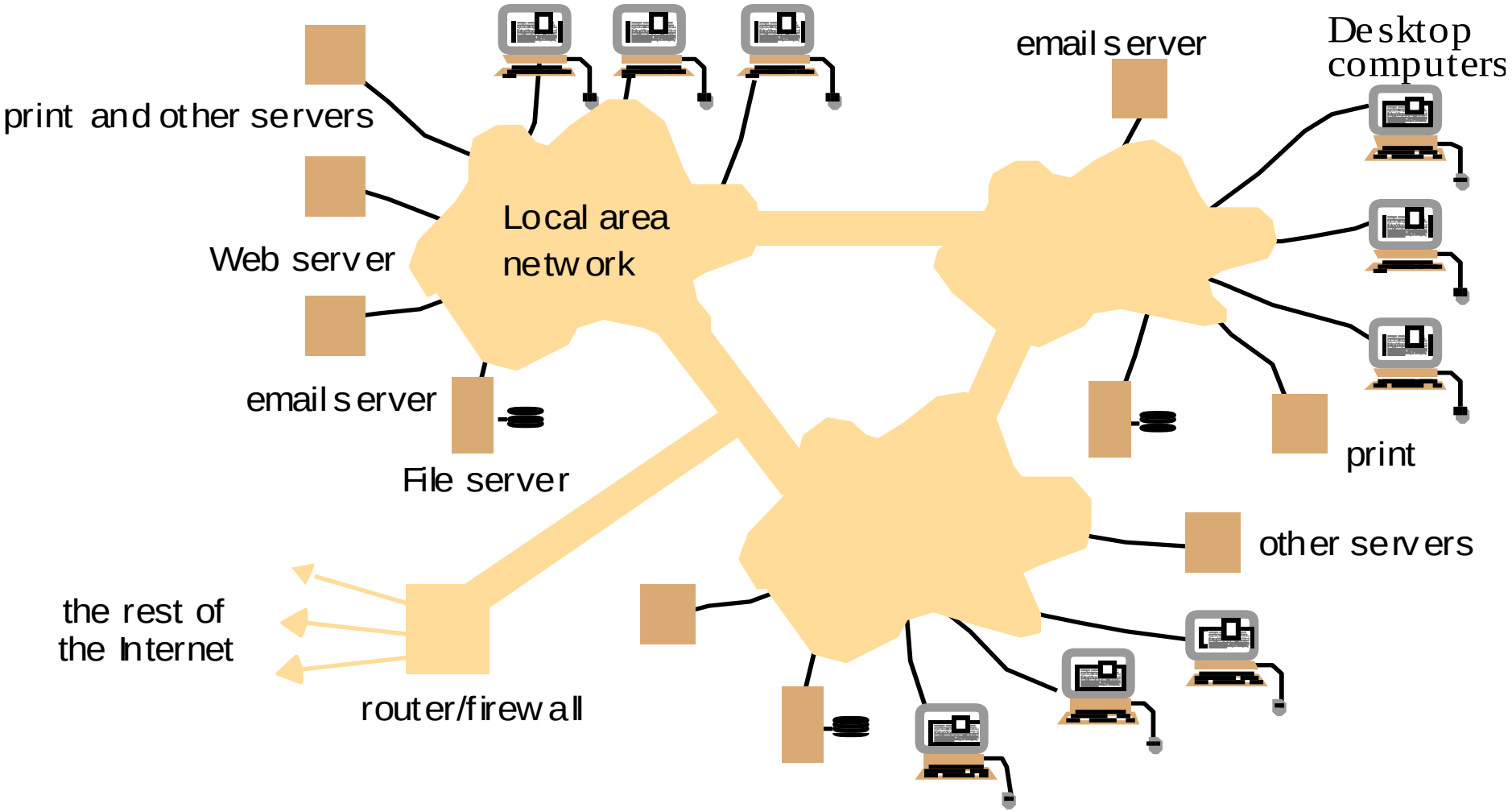
A typical portion of the Internet



Characteristics of Internet

- very **large and heterogeneous**
- enables email, file transfer, multimedia communications, WWW,...
- open-ended
- connects intranets (via backbones) with home users (via modems, ISPs)

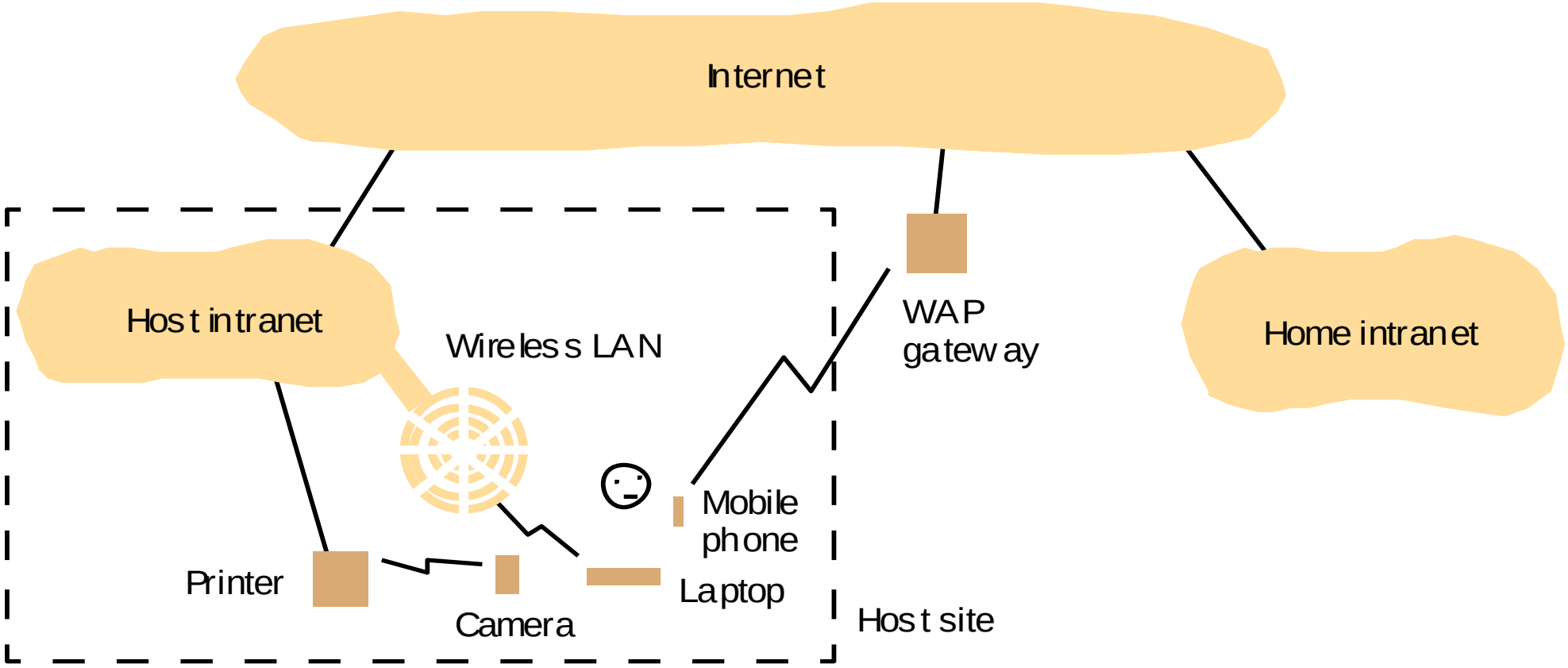
A typical Intranet



Characteristics of intranets

- **Several LANs linked by backbones**
- **Enables information flow within an organization**
  - **electronic data, documents, …**
- **Provides various services**
  - **email, file, print servers,...**
- **often connected to Internet via Router**
- **in/out communications protected by Firewall**

Portable and handheld devices

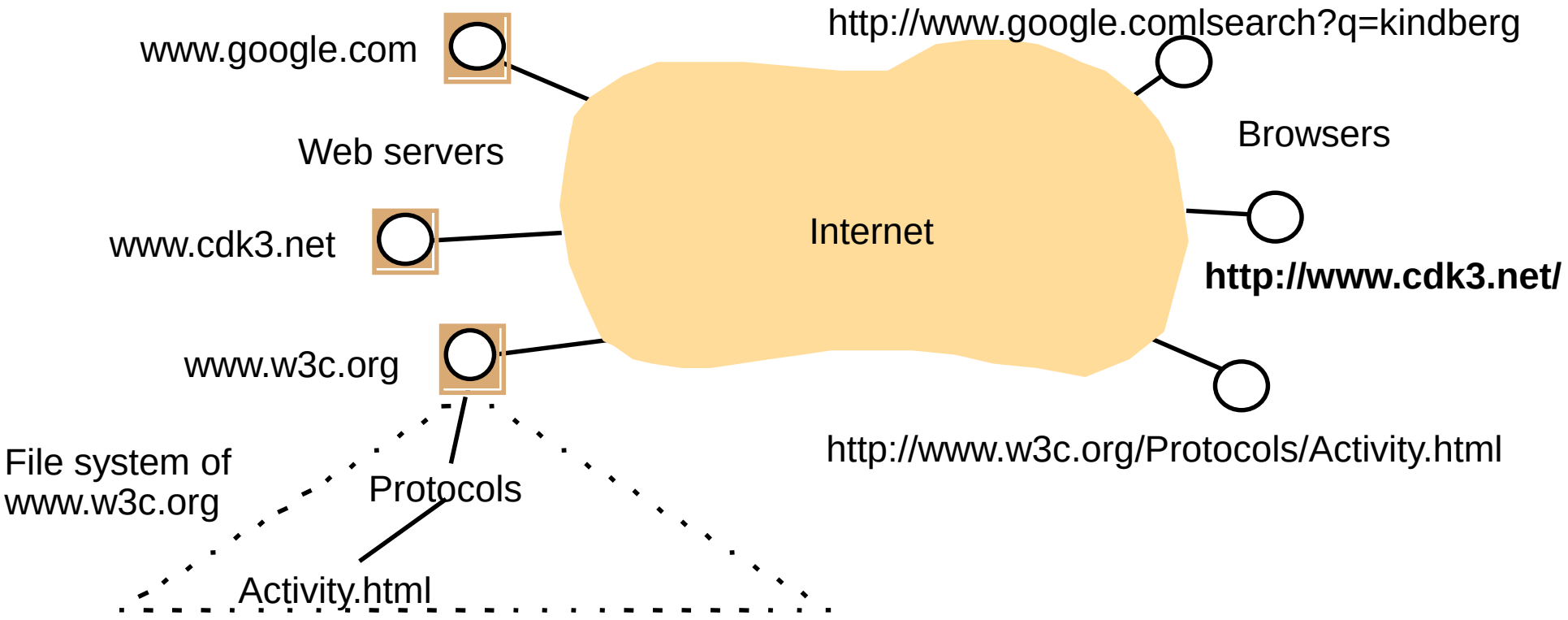


Mobile & ubiquitous computing

- Wireless LANs (WLANs)
  - connectivity for portable devices (laptops, PDAs, mobile phones, video/dig. cameras, …)
  - Uses WAP (Wireless Applications Protocol)

- Home intranet(= home network)
  - devices embedded in home appliances (hi-fi, washing machines, …)
  - universal **‘remote control’** + communication
  - future ➔ **environment for applying embedded systems, ubiquitous computing**

Web servers and web browsers



WWW

- world-wide resource sharing over Internet or Intranet
- based on the following technologies:
  - HTML (HyperText Markup Language)
  - URL (Uniform Resource Locator)
  - Client-Server architecture
- Open system
  - Open-ended
  - can be extended, re-implemented, ...

Distributed Systems' Challenges

- Challenges
  - Heterogeneity
  - Openness
  - Security
  - Scalability
  - Failure handling
  - Concurrency
  - Transparency

- Due to:
  - complexity
  - size
  - changing technologies
  - society's dependence

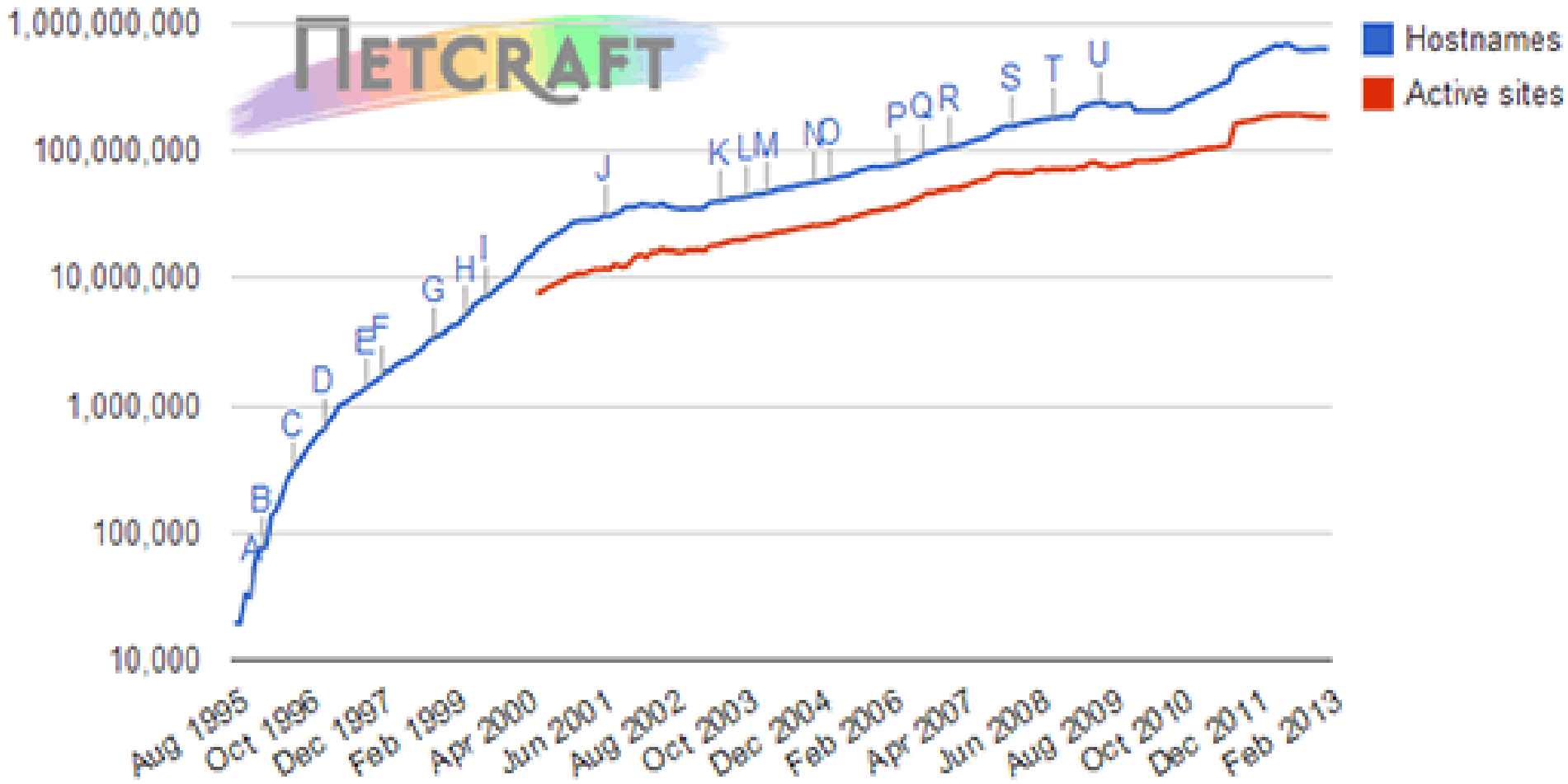
Computers in the Internet

Date	*Computers*	*Web servers*
1979, Dec.	188	0
1989, July	130,000	0
1999, July	56,218,000	5,560,866
2003, Jan.	171,638,297	35,424,956

Computers vs. Web servers in the Internet

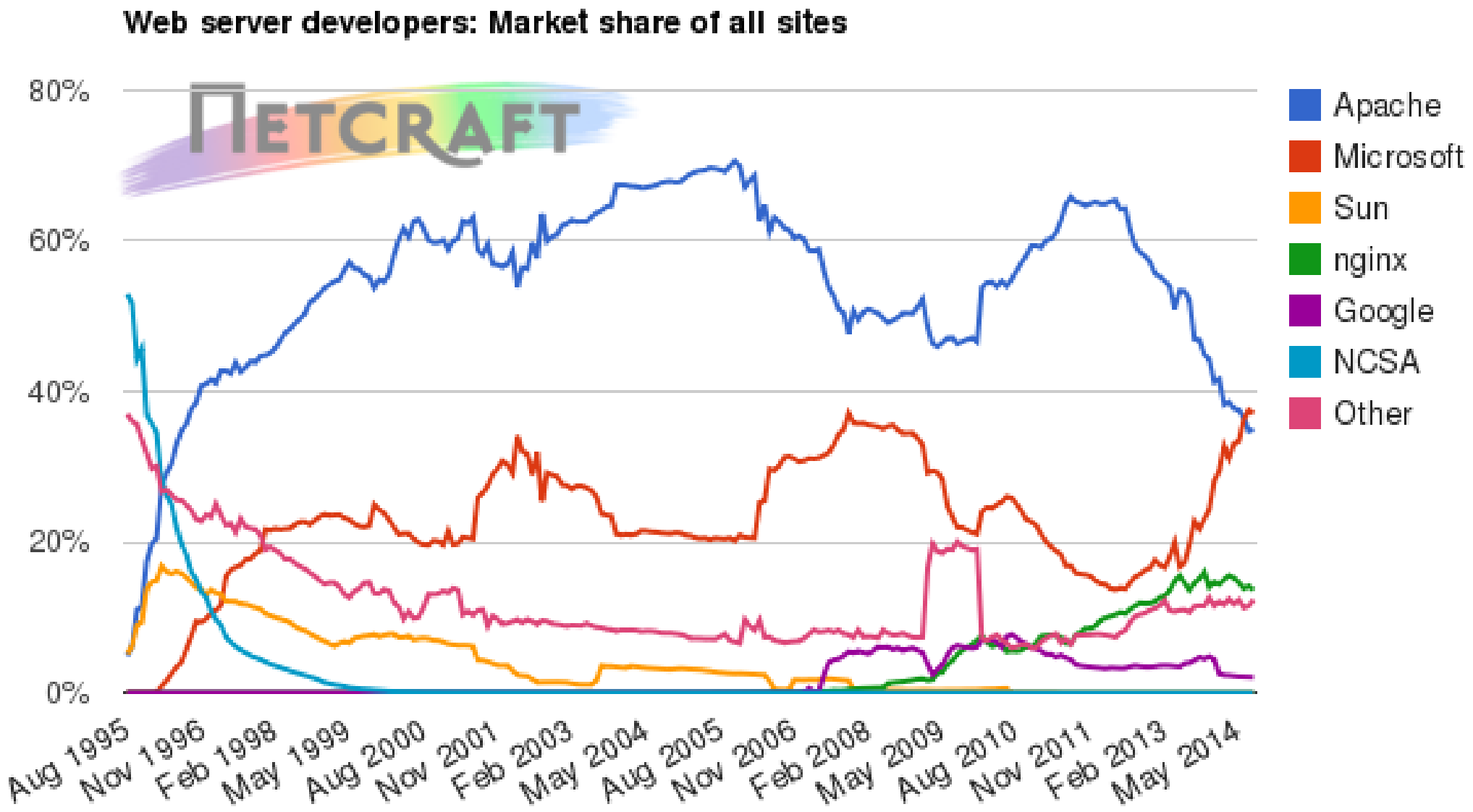
Date	*Computers*	*Web servers*	*Percentage*
1993, July	1,776,000	130	0.008
1995, July	6,642,000	23,500	0.4
1997, July	19,540,000	1,203,096	6
1999, July	56,218,000	6,598,697	12
2001, July	125,888,197	31,299,592	25
2003		42,298,371	
2014		**968,882,453**	
2015		**863,105,652**	

Total Sites Across All Domains
August 1995 - September 2013 -netcraft.com



Market Share for Top Servers Across All Sites

-netcraft.com



Distributed Systems' Challenges

- Heterogeneity
  - Networks, hardware, OSs, P-languages…etc.
  - *Solution-Protocol, middleware.*
- Openness
  - Cumbersome & slow moving - *follow standardization*
- Security
  - Hospital/E-commerce/banking
- Failure handling
  - Software/hardware - *proper policies*
- Concurrency
  - Sharing resource at the same time - *operation should be synchronized*
- Scalability / Transparency

Heterogeneity

- varying software and hardware
  - OSs, networks, computer hardware, program languages, implementations by different developers
  - need for standards of protocols, middleware

- Heterogeneity and mobile code support
  - virtual machine approach (cf, Java applets)

Openness

- independence of vendors
- publishable key interfaces
  - CORBA(Common Object Request Broker Architecture)
- publishable communication mechanisms
  - Java RMI(Remote Method Invocation)

Security

- confidentiality (protect against disclosure)
  - cf, medical records
- integrity (protect against alteration and interference)
  - cf, financial data

⇒ Need encryption and knowledge of identity

1. Denial of Service attacks->Distributed DoS
2. Security of mobile code

Scalability

- Design of scalable distributed systems
  - Controlling the cost of physical resource
  - Controlling the performance loss Preventing software resource running out
  - Avoiding performance bottleneck
  - Examples of scalability limitations

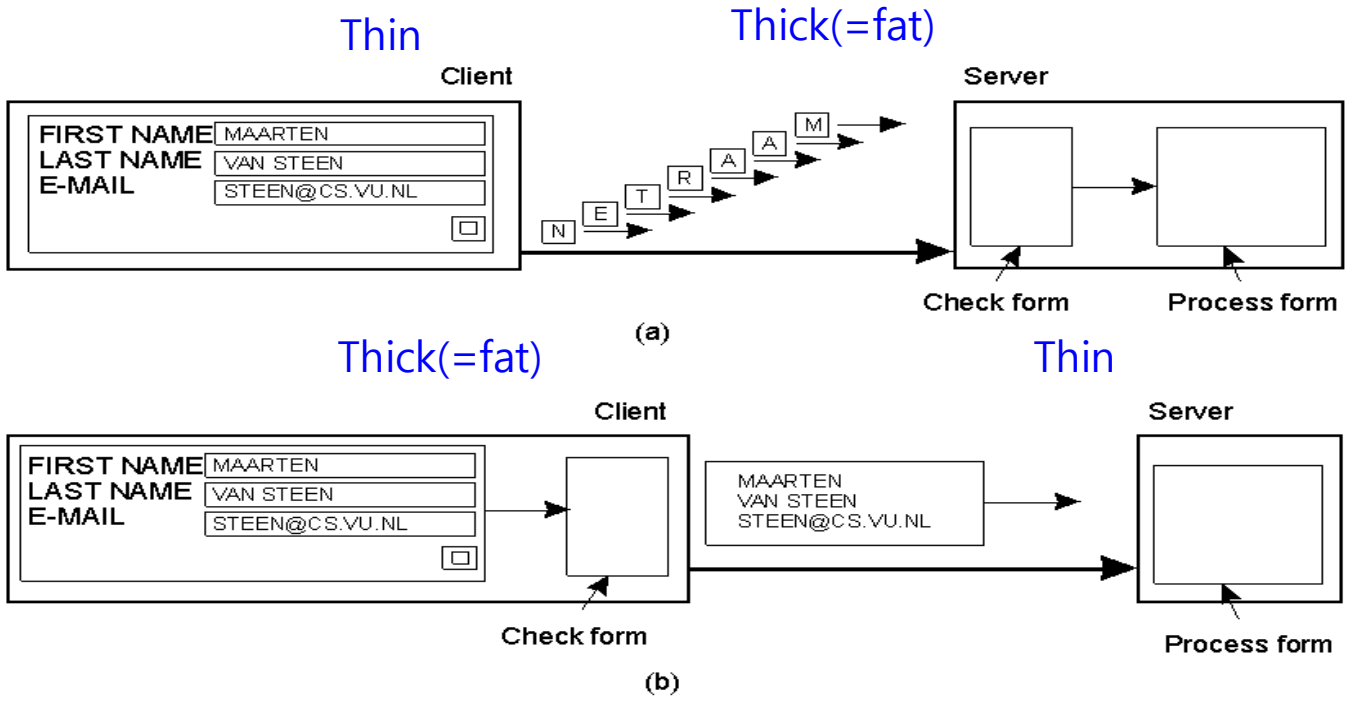
Concept	Example
Centralized services	**A single server for all users**
Centralized data	**A single on-line telephone book**
Centralized algorithms	**Doing routing based on complete information**

Scalability Techniques (1)

- The difference between letting: No. of servers or clients

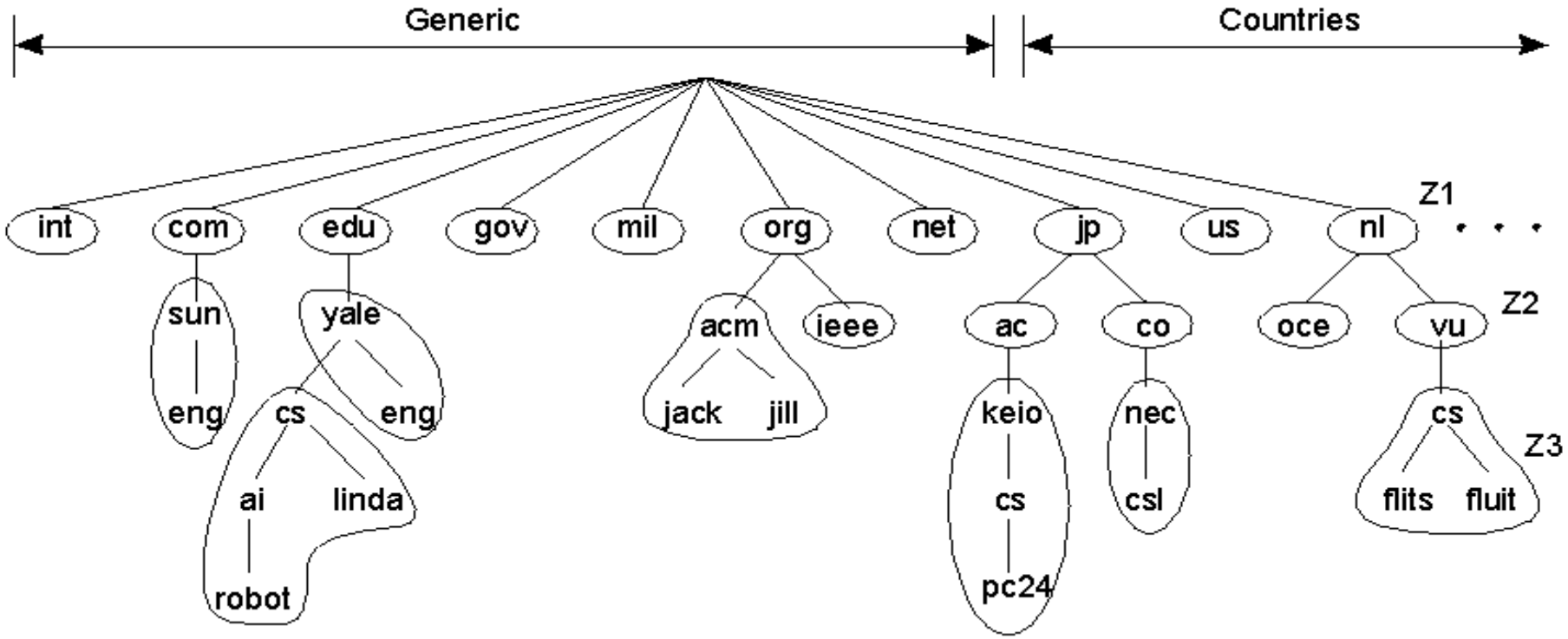
a) a server or

b) a client check forms as they are being filled



Scalability Techniques (2)

- An example of dividing the DNS name space into zones



Scalability

- 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.

Failure handling

Ability to continue computation in the presence of failures.

- **Detecting failures**
- **Masking failures(= hiding failure)**
- **Tolerate failures**
- **Recovery from failures**
- **Redundancy**

Concurrency

Processes execute **simultaneously** and share resources.

- synchronization
- inter-process communication(IPC)

Transparency

Concealment of the separated nature of system from user/programmer

=>Network transparency

- Access transparency + Location Transparency
- cf .log on, email, …. on network

- Transparencies=> **By ANSA Reference Manual & ISO Reference Model for Open Distributed Processing (RM-ODP)**

Transparencies(1) G. Coulouris

•*Access transparency*: enables local and remote resources to be accessed using identical operations.

•*Location transparency*: enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).

•*Concurrency transparency*: enables several processes to operate concurrently using shared resources without interference between them.

•*Replication transparency*: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.

Transparencies(2) G. Coulouris

•*Failure transparency*: enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.

•*Mobility transparency*: allows the movement of resources and clients within a system without affecting the operation of users or programs.

•*Performance transparency*: allows the system to be reconfigured to improve performance as loads vary.

•*Scaling transparency*: allows the system and applications to expand in scale without change to the system structure or the application algorithms.

Transparency (A. Tanenbaum)

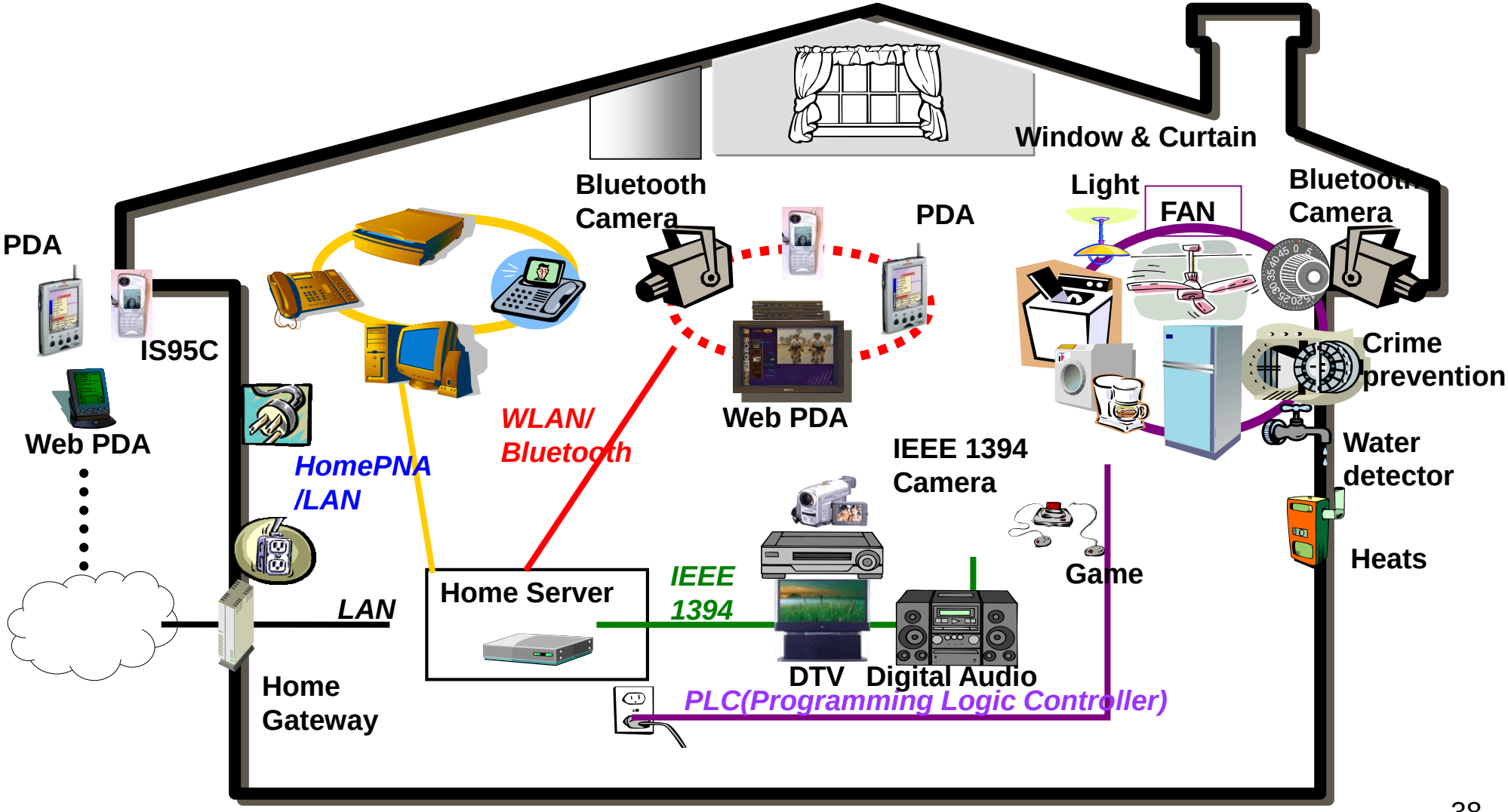
Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation(mobility)	Hide that a resource may be moved to another location while in use
Replication	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
Persistence	Hide whether a (software) resource is in memory or on disk

Summary

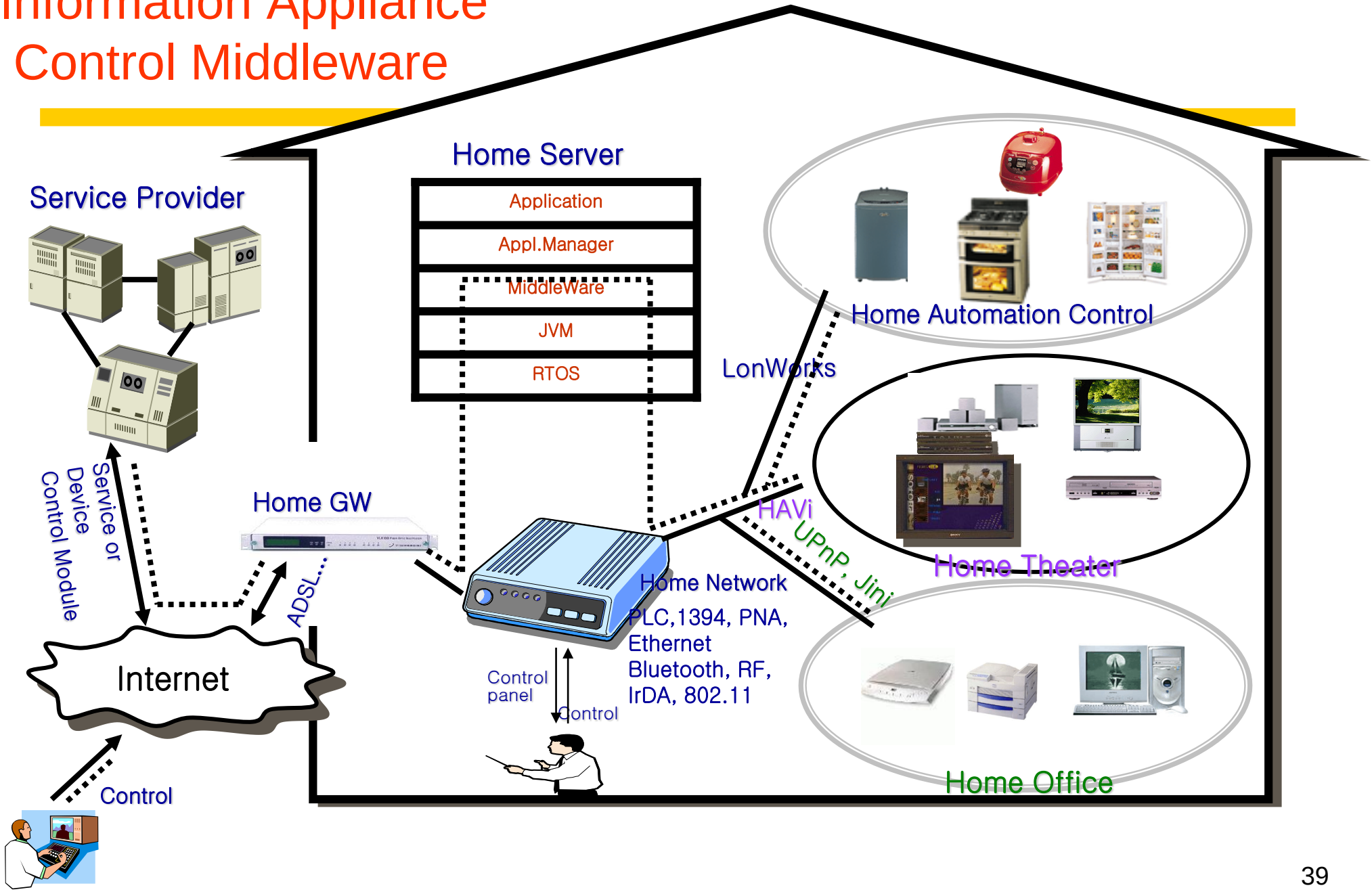
Introduction to Distributed Computing Systems

- pervasive in society
- Consequences
  - Concurrency
  - No global clock
  - Independent failure
- use a variety of technologies
- understanding underlying concepts and issues important in their management, implementation, programming
- DS's challenges
  - Heterogeneity
  - Openness
  - Security
  - Scalability
  - Fault handling
  - Concurrency
  - Transparency

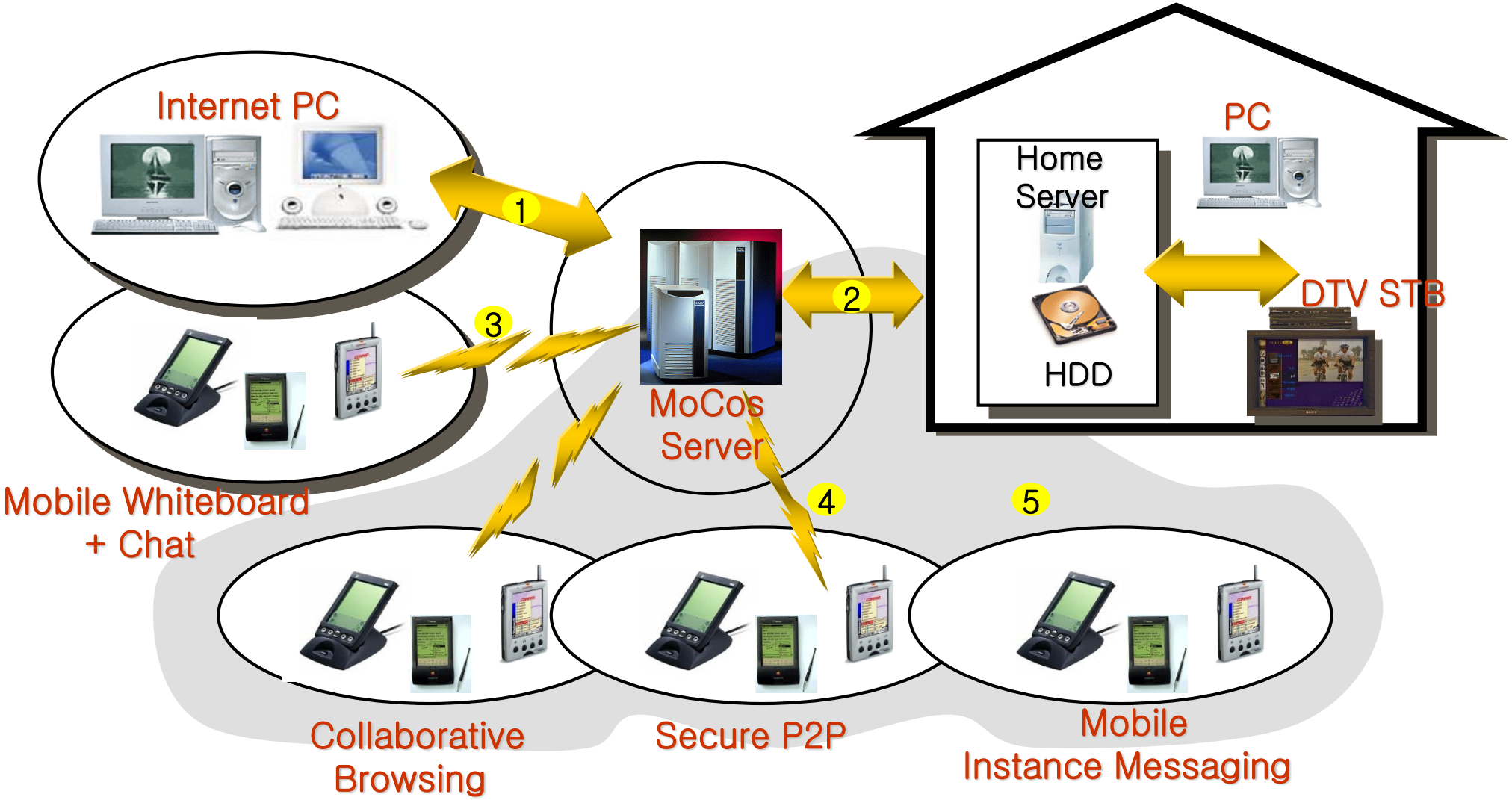
Overview of Internet Information Appliances



Information Appliance Control Middleware



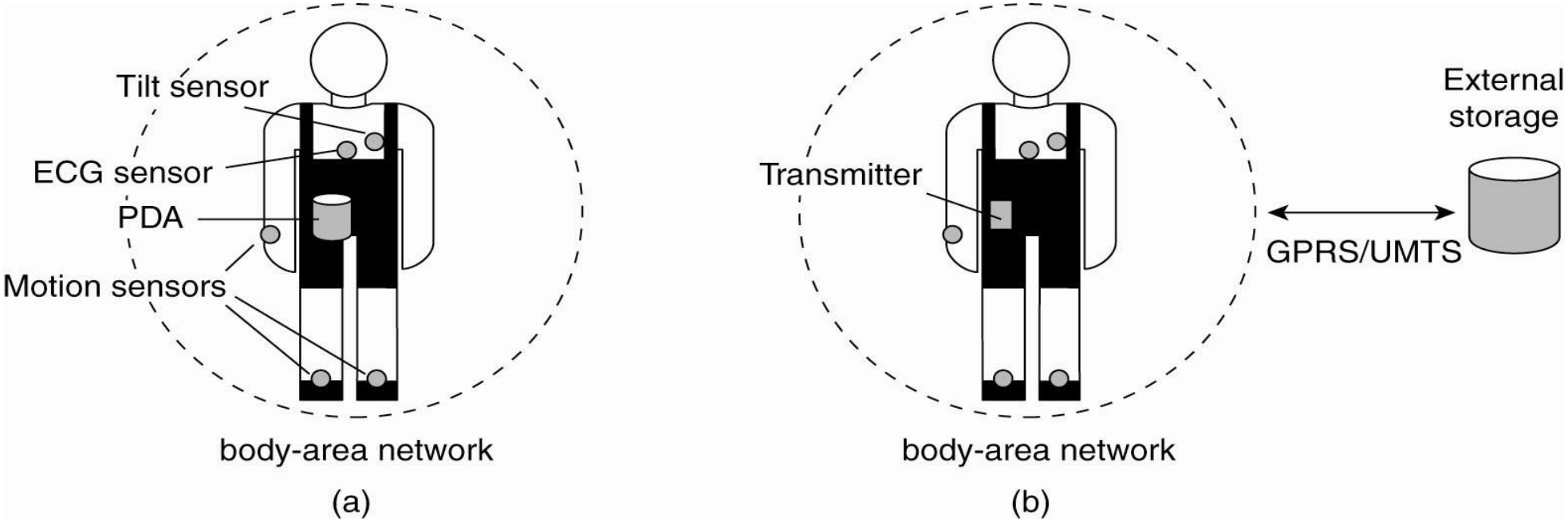
Mobile Collaboration for Information Appliance



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 (2)



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

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.