Selected Topics in Software Engineering - 1

[Software Engineering for Distributed Systems]

Soha Makady

s.makady@fci-cu.edu.eg

About Me

- Dr. Soha Makady: s.makady@fci-cu.edu.eg
  - Office hours: Right after the lecture, or by appointment (through email)
  - Office: 2nd floor in the New Student Building, in the main campus, right beside the electronics lab.

Outline

- Introduction to Distributed Systems
- Course Organization

Outline

- Introduction to Distributed Systems
- **Course Organization**

Course Learning Objectives

- The course teaches several state-of-the-art topics about large, distributed software systems and how they are developed.
- Concepts to be covered (**tentatively**) include:
  - Networking and communication basics
  - Concurrency (threading and its pros/cons)
  - Inter-process communication
  - Remote invocation
  - Indirect communication (e.g. messaging and publish/subscribe systems)
  - Distributed objects and components
  - Web services and Micro services
  - Distributed caching
  - Blockchain architecture and consensus algorithms

(Tentative) Evaluation

- Midterm (20 marks)
- Assignments / Lab exam (20 marks)
- Final exam (60 marks)

Evaluation (Cont’d)

- Cheating Policy
  - There will be ZERO tolerance for any sort of cheating.
  - COPYING your code from online resources IS CHEATING
  - You are expected to submit your OWN ORIGINAL work for the graded course work.
  - Discussing the details of your solution with your colleague is CHEATING
  - When in doubt, then it is probably cheating!

Course Material

- Textbooks
  - **George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair**. "Distributed systems: concepts and design“. Fifth Edition, published by Addison Wesley, May 2011.
  - Gorton, Ian. *Foundations of Scalable Systems*. " O'Reilly Media, Inc.", 2022.
- Additional readings may be added during each lecture.

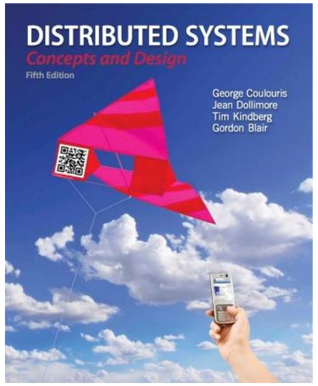
Outline

- **Introduction to Distributed Systems**
- Course Organization

Introduction to Distributed Systems and Characterisation

Dr. Rajkumar Buyya

Cloud Computing and **D**istributed **S**ystems (CLOUDS) Laboratory
School of Computing and Information Systems
The University of Melbourne, Australia
http://www.buyya.com

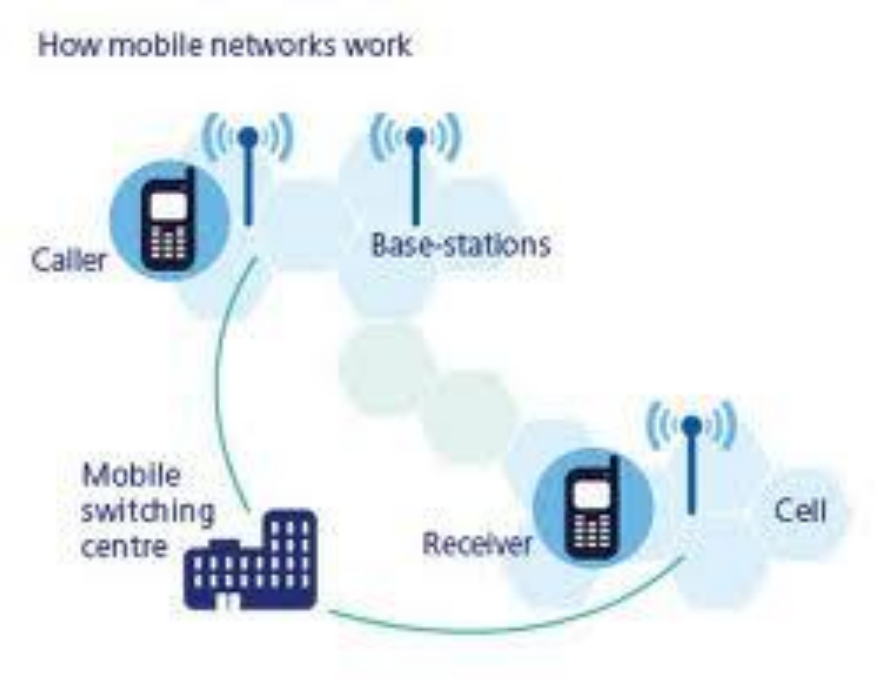


Most concepts are drawn from Chapter 1

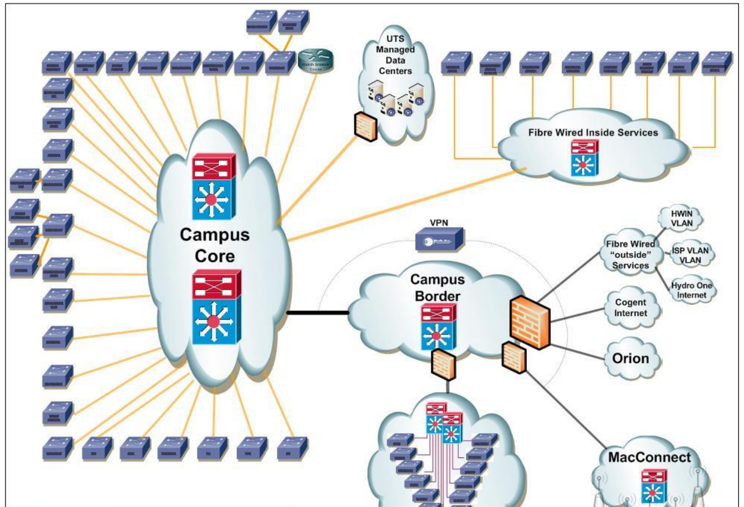
Presentation Outline

- Introduction
- Defining Distributed Systems
- Characteristics of Distributed Systems
- Example Distributed Systems
- Challenges of Distributed Systems
- Summary

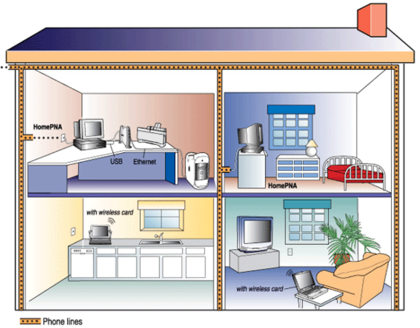
Introduction



- Networks of computers are everywhere!
  - Mobile phone networks
  - Corporate networks
  - Factory networks
  - Campus networks
  - In-car networks
  - **Internet of Things (IoT)**
  - On board networks in planes and trains



WiFi INFLIGHT™



Defining Distributed Systems

- *“A system in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing.”* [Coulouris]
- *“A distributed system is a collection of independent computers that appear to the users of the system as a single computer.”* [Tanenbaum]
    - Example?

Defining Distributed Systems

- Example Distributed Systems:
  - Example 1:
    - Consider a network of workstations in a university or company department.
    - In addition to each user's personal workstation, there is a pool of processors in the machine room that are not assigned to specific users but are allocated dynamically as needed.
    - When a user types a command, the system could look for the best place to execute that command, possibly on the user's own workstation, and possibly on one of the unassigned processors in the machine room.
    - If the system as a whole looked and acted like a classical single-processor timesharing system, it would qualify as a distributed system.

Defining Distributed Systems

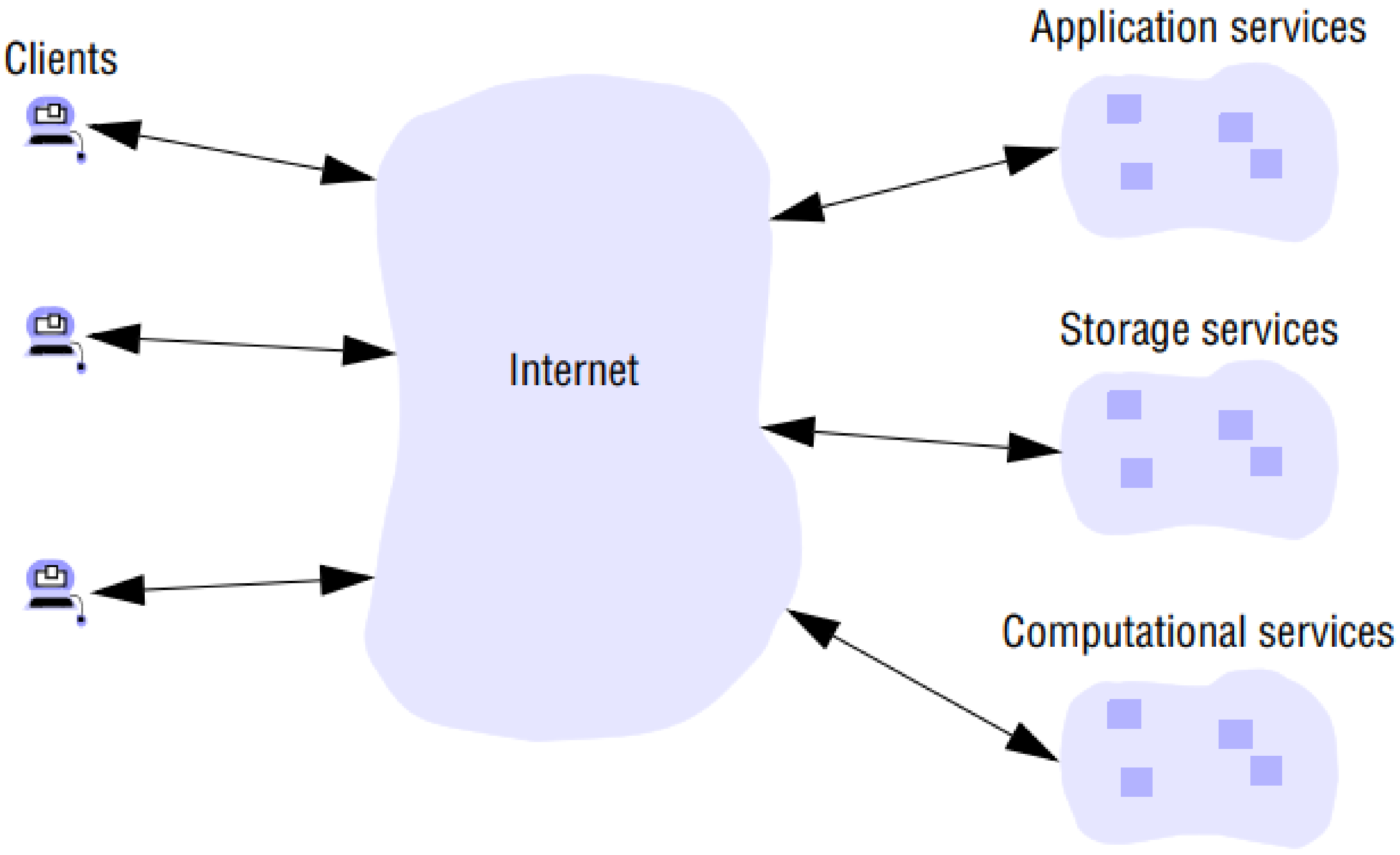
- Example Distributed Systems:
  - Example 2:
    - Consider a large bank with hundreds of branch offices all over the world.
    - Each office has a master computer to store local accounts and handle local transactions.
    - Each computer has the ability to talk to all other branch computers and with a central computer at headquarters.
    - If transactions can be done without regard to where a customer or account is, and the users do not notice any difference between this system and the old centralized mainframe that it replaced, it too would be considered a distributed system.

Defining Distributed Systems

- Example Distributed Systems:
  - Cloud:
    - *“a type of parallel and distributed system consisting of a collection of interconnected and* ***virtualised computers*** *that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers” [Buyya].*

Defining Distributed Systems

Cloud computing

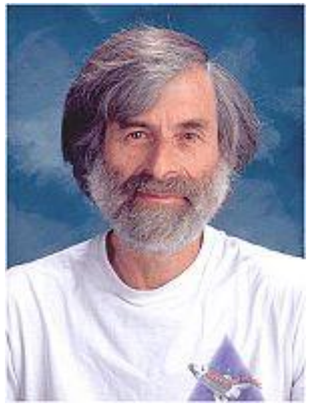


Defining Distributed Systems

- Example Distributed Systems:
  - Cluster:
    - *“A type of parallel or distributed processing system, which consists of a collection of interconnected stand-alone computers cooperatively working together as a single, integrated computing resource” [Buyya].*

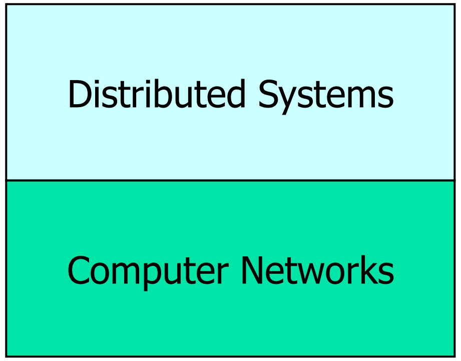
Leslie Lamport’s Definition

- *"A distributed system is one on which I cannot get any work done because some machine I have never heard of has crashed.“*
  - Leslie Lamport – a famous researcher on timing, message ordering, and clock synchronization in distributed systems.
  - Example?
  - Our Zoom meetings during the Corona time!



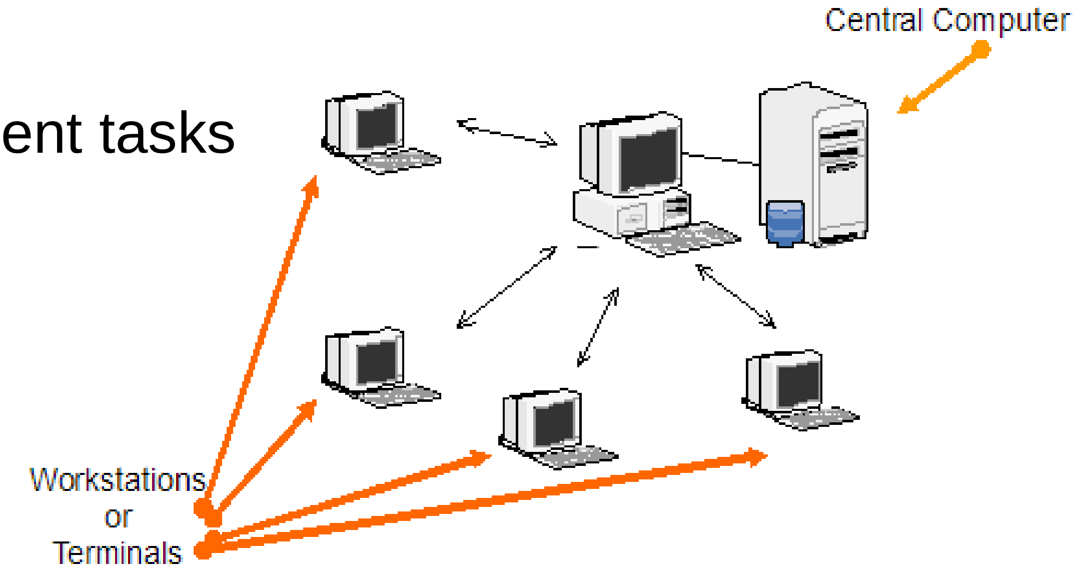
Networks vs. Distributed Systems

- Networks: A media for interconnecting local and wide area computers and exchange messages based on protocols. Network entities are visible and they are explicitly addressed (IP address).
- Distributed System: existence of multiple autonomous computers is transparent
- However,
  - many problems (e.g., openness, reliability) in common, but at different levels.
    - Networks focuses on packets, routing, etc., whereas distributed systems focus on applications.
    - Every distributed system relies on services provided by a computer network.



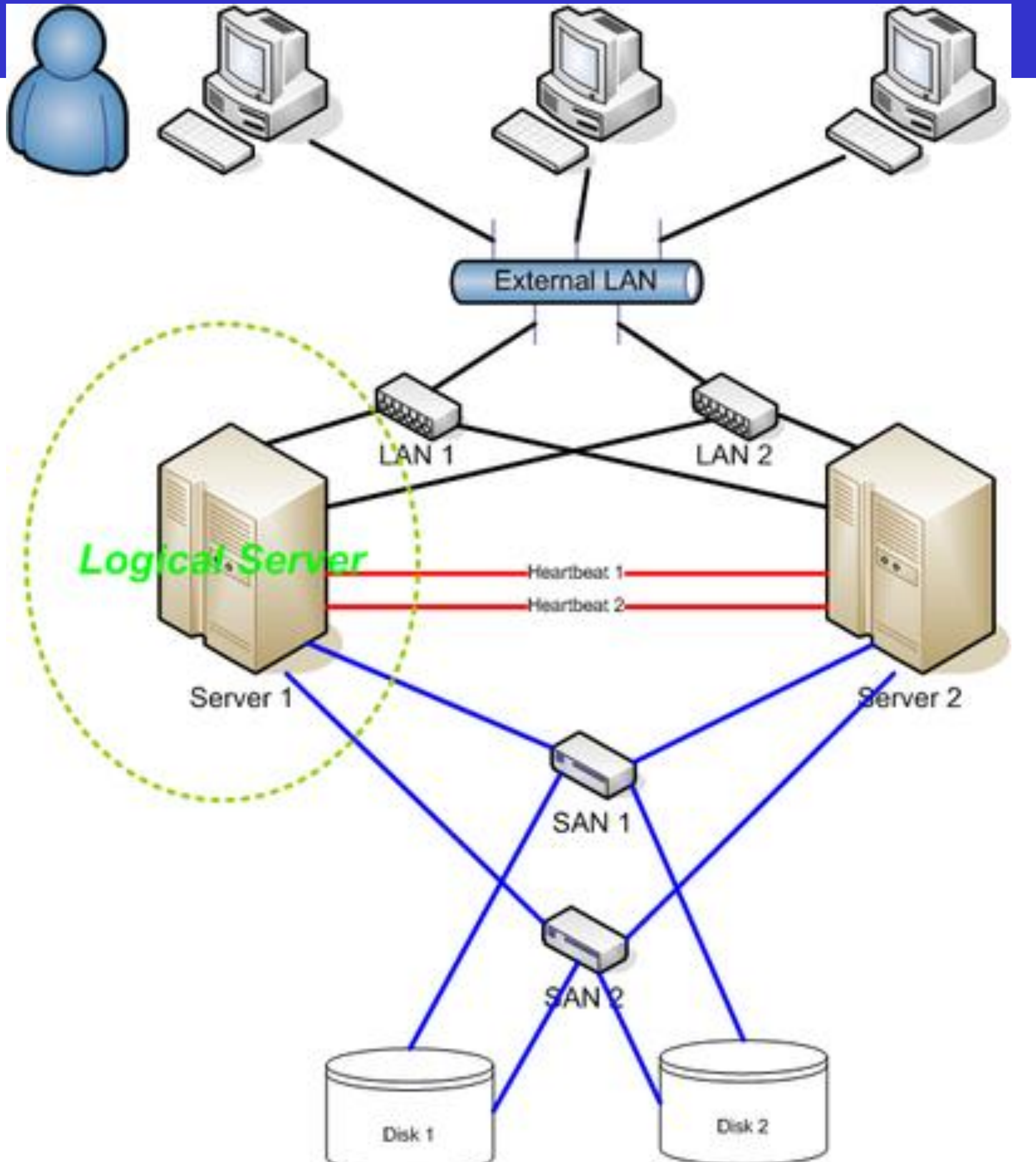
Characteristics of Distributed Systems

- Parallel activities
  - Autonomous components executing concurrent tasks
- Communication via message passing
  - No shared memory
- Resource sharing
  - Printer, database, other services
- No global state
  - No single process can have knowledge of the current global state of the system
- No global clock
  - Only limited precision for processes to synchronize their clocks



Goals of Distributed Systems

- Connecting Users and Resources
- Transparency
- Openness
- Scalability
- Enhanced Availability

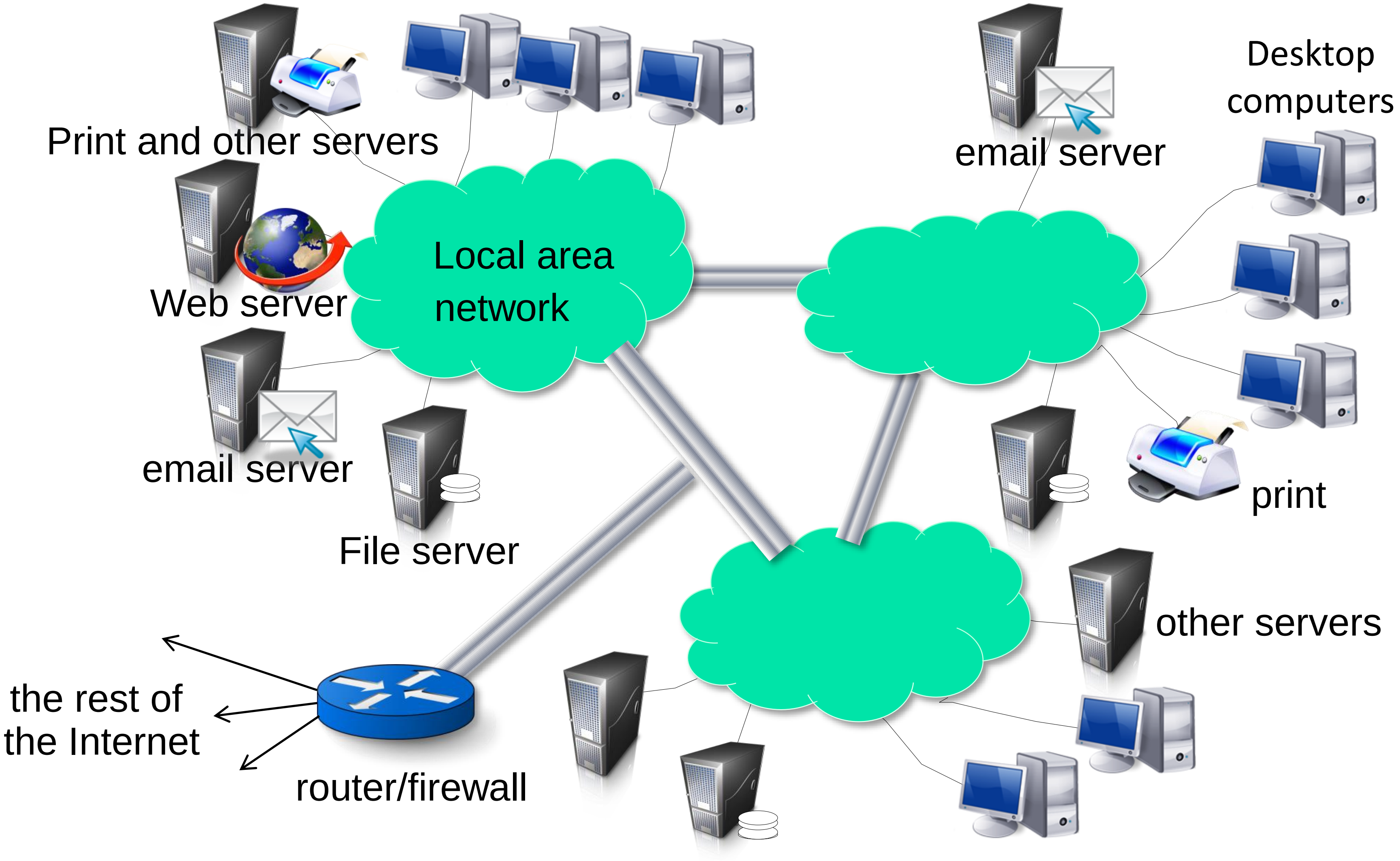


Selected application domains and associated networked applications

Finance and Commerce	eCommerce e.g. Amazon and eBay, PayPal, online banking and trading
The information Society	Web information and search engines, ebooks, Wikipedia; social networking: Facebook and Twitter.
Creative Industries and Entertainment	Online gaming, music and film in the home, user-generated content, e.g. YouTube, Flickr
Healthcare	Health informatics, on online patient records, monitoring patients (Metro South Health hospital trial in Queensland)
Education	e-learning, virtual learning environments; distance learning. e.g., Coursera
Transport and Logistics	GPS in route finding systems, map services: Google Maps, Google Earth
Science and Engineering	Cloud computing as an enabling technology for collaboration between scientists (LHC, LIGO)
Environmental Management	Sensor networks to monitor earthquakes, floods or tsunamis (Bureau of Meteorology flood warning system)

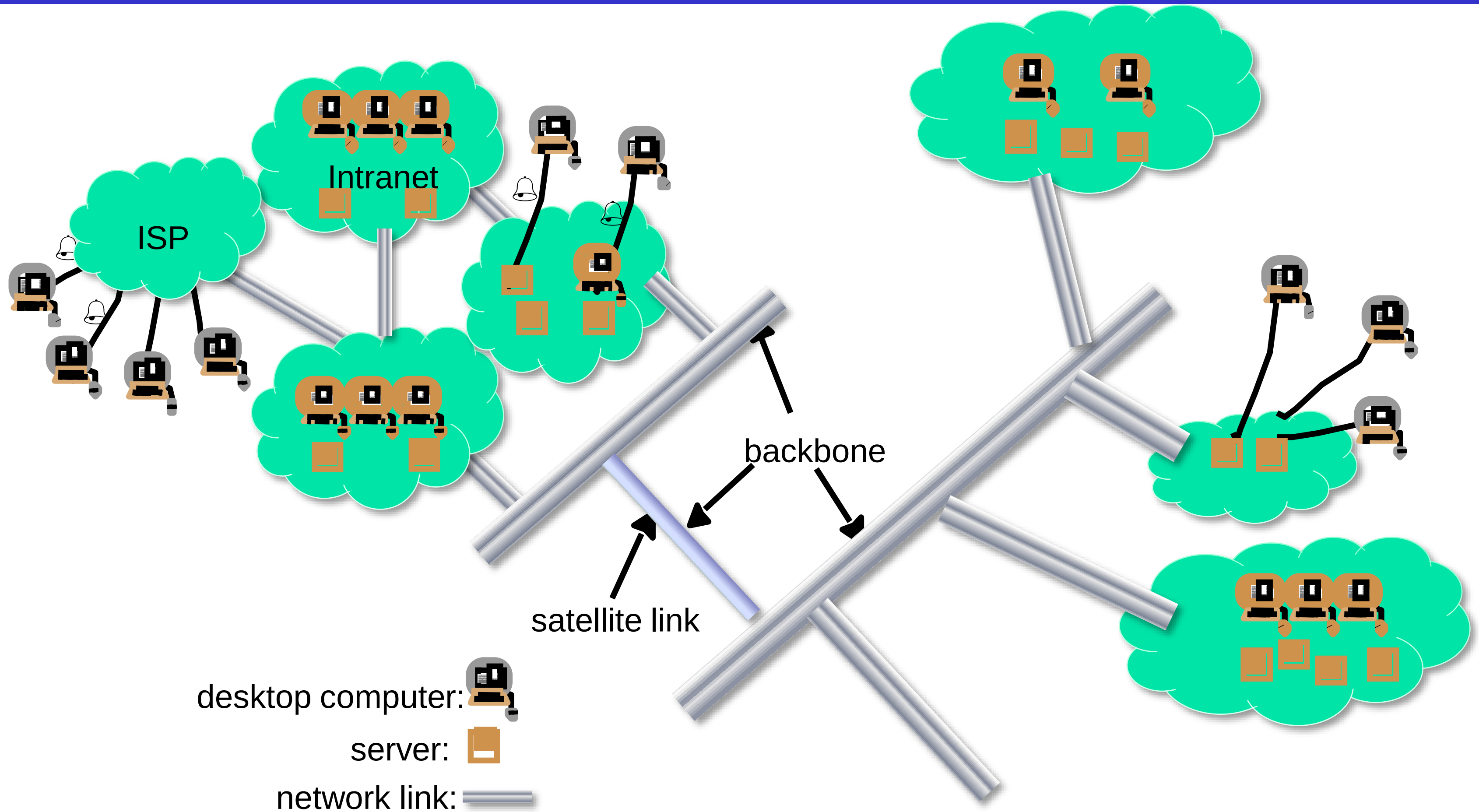
A typical Intranet:

A portion of Internet that is separately administered & supports internal sharing of resources (file/storage systems and printers)



A typical portion of the Internet and its services:

Multimedia services providing access to music, radio, TV channels, and video conferencing supporting several users.

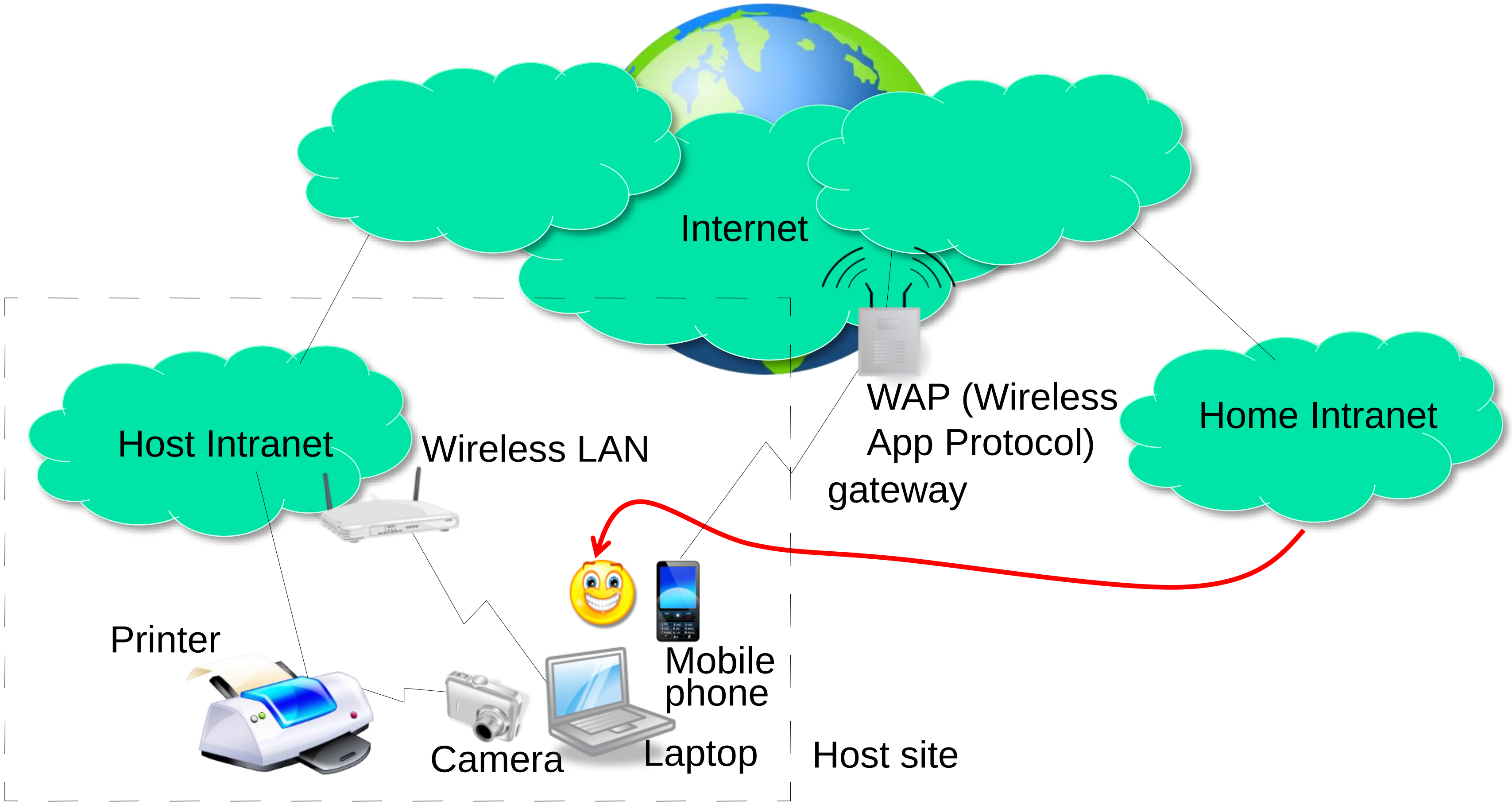


- The Internet is a vast collection of computer networks of many different types and hosts various types of services.

Mobile and Ubiquitous Computing

- Technological advances in device miniaturization and wireless networking have led increasingly to the integration of small and portable computing devices into distributed systems.
- These devices include:
  - Laptop computers.
  - Handheld devices, including mobile phones, smart phones, GPS-enabled devices, pagers, personal digital assistants (PDAs), video cameras and digital cameras.
  - Wearable devices, such as smart watches with functionality similar to a PDA.
  - Devices embedded in appliances such as washing machines, hi-fi systems, cars and refrigerators.
- Mobile computing?
- Any challenges with mobile computing?

Mobile and ubiquitous computing: portable and handheld devices in a distributed system



- Supports continued access to Home intranet resources via wireless and provision to utilise resources (e.g., printers) that are conveniently located (location-aware computing).

Business Example and Challenges

- Online bookstore (e.g. in World Wide Web)
  - Customers can connect their computer to your computer (web server):
    - Browse your inventory
    - Place orders
    - …

This example has been adapted from **Torbin Weis**, Berlin University of Technology

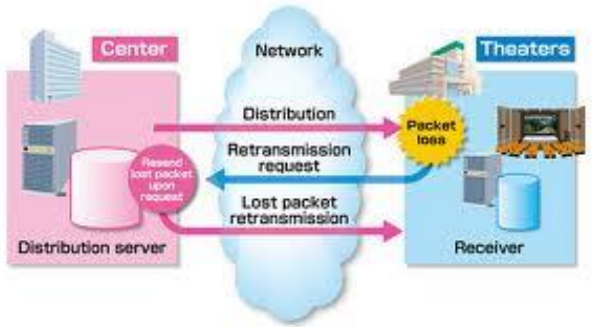
Business Example – Challenges I

- What if
  - Your customer uses a completely different hardware? (PC, MAC, iPad, Mobile…)
  - … a different operating system? (Windows, Unix,…)
  - … a different way of representing data? (ASCII, EBCDIC,…)
  - **Heterogeneity**
- Or
  - You want to move your business and computers to the Caribbean (because of the weather or low tax)?
  - Your client moves to the Caribbean (more likely)?
  - **Distribution transparency**

Business Example – Challenges II

- What if
  - Two customers want to order the same item at the same time?
  - **Concurrency**
- Or
  - The database with your inventory information crashes?
  - Your customer’s computer crashes in the middle of an order?
  - **Fault tolerance**

Business Example – Challenges III



Reliable and secure distribution

- What if
  - Someone tries to break into your system to steal data?
  - … sniffs for information?
  - … your customer orders something and doesn’t accept the delivery saying he didn’t?
  - **Security**
- Or
  - You are so successful that millions of people are visiting your online store at the same time?
  - **Scalability**

Business Example – Challenges IV

- When building the system…
  - Do you want to write the whole software on your own (network, database,…)?
  - What about updates, new technologies?
  - **Reuse** and **Openness** (Standards)

Overview Challenges I

- Heterogeneity
  - Heterogeneous components must be able to interoperate
- Distribution transparency
  - Distribution should be hidden from the user as much as possible
- Fault tolerance
  - Failure of a component (partial failure) should not result in failure of the whole system
- Scalability
  - System should work efficiently with an increasing number of users
  - System performance should increase with inclusion of additional resources

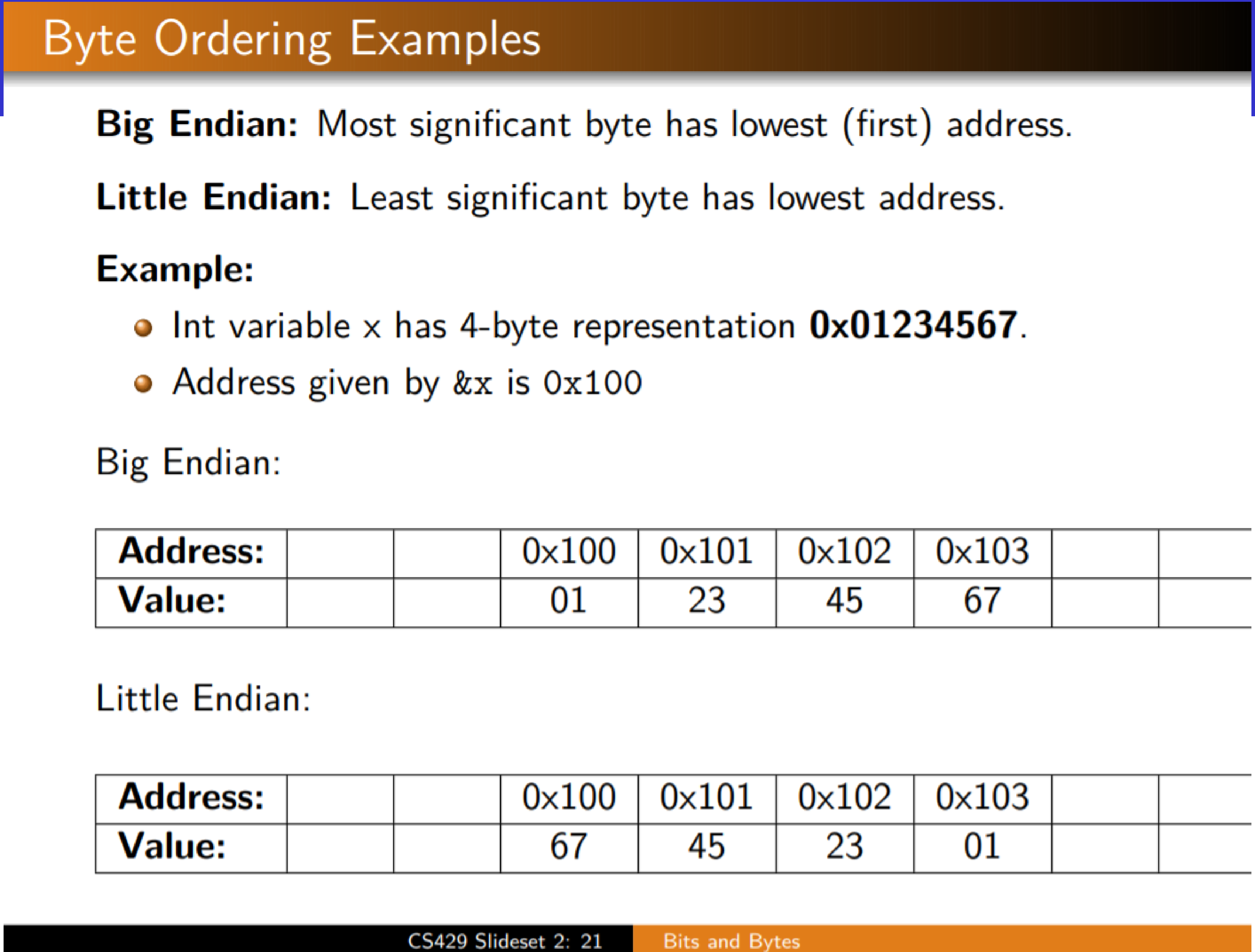
Overview Challenges II

- Concurrency
  - Shared access to resources must be possible
- Openness
  - Interfaces should be publicly available to ease inclusion of new components
- Security
  - The system should only be used in the way intended

Heterogeneity

- Heterogeneous components must be able to interoperate across different:
  - Operating systems
  - Hardware architectures (Marshalling!)
  - Communication architectures
  - Programming languages
  - Software interfaces
  - Security measures
  - Information representation

Heterogeneity – Hardware Architectures



Byte Ordering Examples

Big Endian: Most significant byte has lowest (first) address.

Little Endian: Least significant byte has lowest address.

Example:

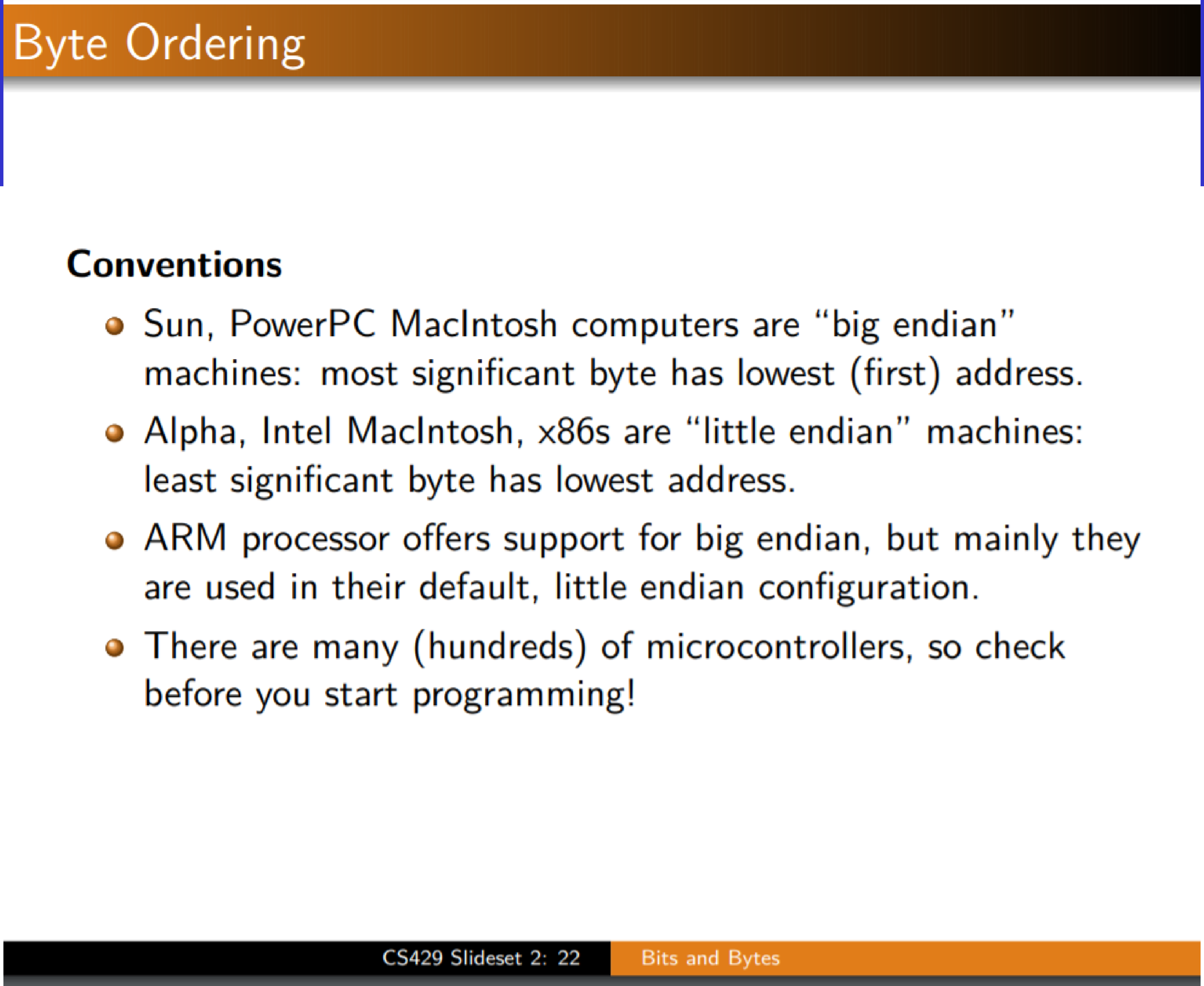
- Int variable x has 4-byte representation **0x01234567**.
- Address given by &x is 0x100

Big Endian:

Address:			0x100	0x101	0x102	0x103		
Value:			01	23	45	67		

Little Endian:

Address:			0x100	0x101	0x102	0x103		
Value:			67	45	23	01		



Byte Ordering

Conventions

- Sun, PowerPC MacIntosh computers are “big endian” machines: most significant byte has lowest (first) address.
- Alpha, Intel MacIntosh, x86s are “little endian” machines: least significant byte has lowest address.
- ARM processor offers support for big endian, but mainly they are used in their default, little endian configuration.
- There are many (hundreds) of microcontrollers, so check before you start programming!

Summary

- Distributed Systems are everywhere
- Internet enables users throughout the world to access its (application) services from anywhere
- Resource sharing is the main motivating factor for constructing distributed systems
- Construction of DS produces many challenges:
  - Heterogeneity, Openness, Security, Scalability, Failure handling, Concurrency, and Transparency
- Distributed systems enable globalization:
  - Community (Virtual teams, organizations, social networks)
  - Science (e-Science)
  - Business (..e-Banking..)
  - Entertainment (YouTube, e-Friends)

Required Readings

- Chapter 1 from the course’s textbook:

George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair. "Distributed systems: concepts and design“. Fifth Edition, published by Addison Wesley, May 2011.