CS454/654 Distributed Systems

Bernard Wong
(based on notes from Tamer Ozsu)
DC 3514
bernard@uwaterloo.ca

Course Modules

- Module 1: Fundamental Architectures and Models of Distributed Systems
- Module 2: Brief Overview of Computer Networks
- Module 3: Distributed Objects and Remote Invocation
- Module 4: Distributed Naming
- Module 5: Distributed File Systems
- Module 6: Synchronization
- Module 7: Data Replication
- Module 8: Fault Tolerance
- Module 9: Security

Course Information

- Intended Audience:
  - CS 454 is a course for CS major students and is normally completed in the fourth year.
- Related Courses:
  - Prerequisites: CS 350 (CS 354) or ECE 354 and fourth-year standing in a CS major program.
  - Antirequisites: CS 436, E&CE 454.
  - CS 456 is not a prerequisite but provides information about the underlying facilities assumed in this course.

Course Information

- Lectures:
  - TR 2:30-3:50pm in MC 2017
- Office Hours:
  - Best to send e-mail to set-up appointment
  - TBA
- Assignments:
  - 3 assignments, a combination of answering questions and implementation work.
- Exams:
  - A Midterm and a Final

Course Documents

- Textbook:
  - A.S. Tanenbaum and M. van Steen, *Distributed Systems: Principles and Paradigms*, Pearson/Prentice-Hall, 2007 (2nd Edition).
- Other References:
  - G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*, *4th edition*, Addison-Wesley, 2005.
- Course Home Page:
  - **`http://www.cs.uwaterloo.ca/~bernard/cs454`**
  - Home page will include all the slides used in the course.
- Course Newsgroup:
  - **`uw.cs.cs454`**

Administrivia

- Grading

	CS 454	CS 654
Assignments	30%	22% (Assignments 2&3 only)
Midterm	30%	20%
Final	40%	40%
Project		18%

  - Assignments 1& 3 are 8%, 2 is 14%
  - Students have to pass the exams to obtain a passing grade in the course.
- Announcements
  - In class, on the course Web page, and on the newsgroup; material will also be available electronically

Plagiarism and Academic Offenses

- I will be very strict with academic offenses.
- Nice explanation of plagiarism on-line
  `http://watarts.uwaterloo.ca/~sager/plagiarism.html`
- Plagiarism applies to both text and code
- You are free (even encouraged) to exchange ideas, but ***no sharing code***
- Possible penalties
  - First offense
    - → -100% for that part of the course
  - Second offense
    - → Expulsion is possible

What’s a Distributed System?

A distributed system is a collection of independent computers that appear to the users of the system as a single computer

- Example:
  - a network of workstations allocated to users
  - a pool of processors in the machine room allocated dynamically
  - a single file system (all users access files with the same path name)
  - user command executed in the best place (user workstation, a workstation belonging to someone else, or on an unassigned processor in the machine room)

Why Distributed?

Economics	Microprocessors offer a better price/ performance than mainframes
Speed	A distributed system may have more total computing power than a mainframe
Inherent distribution	Some applications involve spatially separated machines
Reliability	If one machine crashes, the system as a whole can still survive
Incremental growth	Computing power can be added in small increments

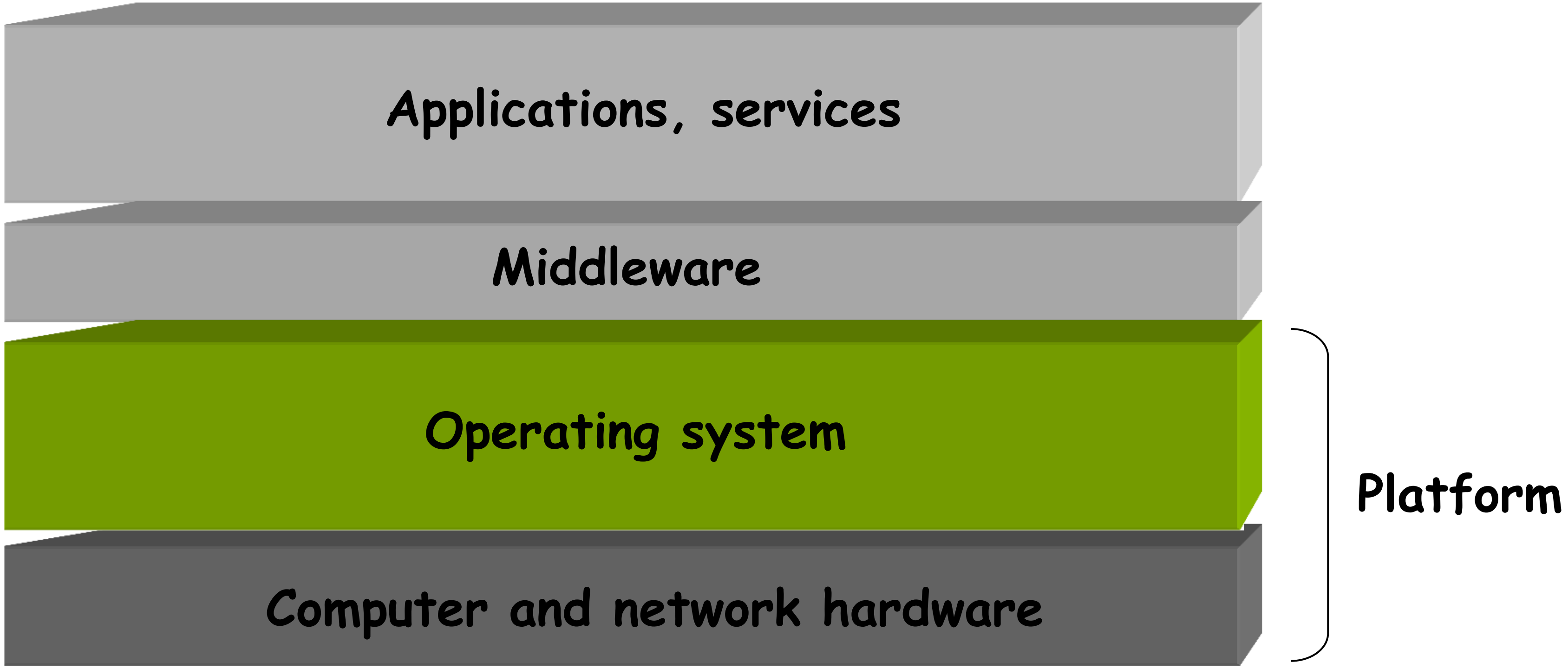
Primary Features

- Multiple computers
  - Concurrent execution
  - Independent operation and failures
- Communications
  - Ability to communicate
  - No tight synchronization (no global clock)
- “Virtual” Computer
  - Transparency

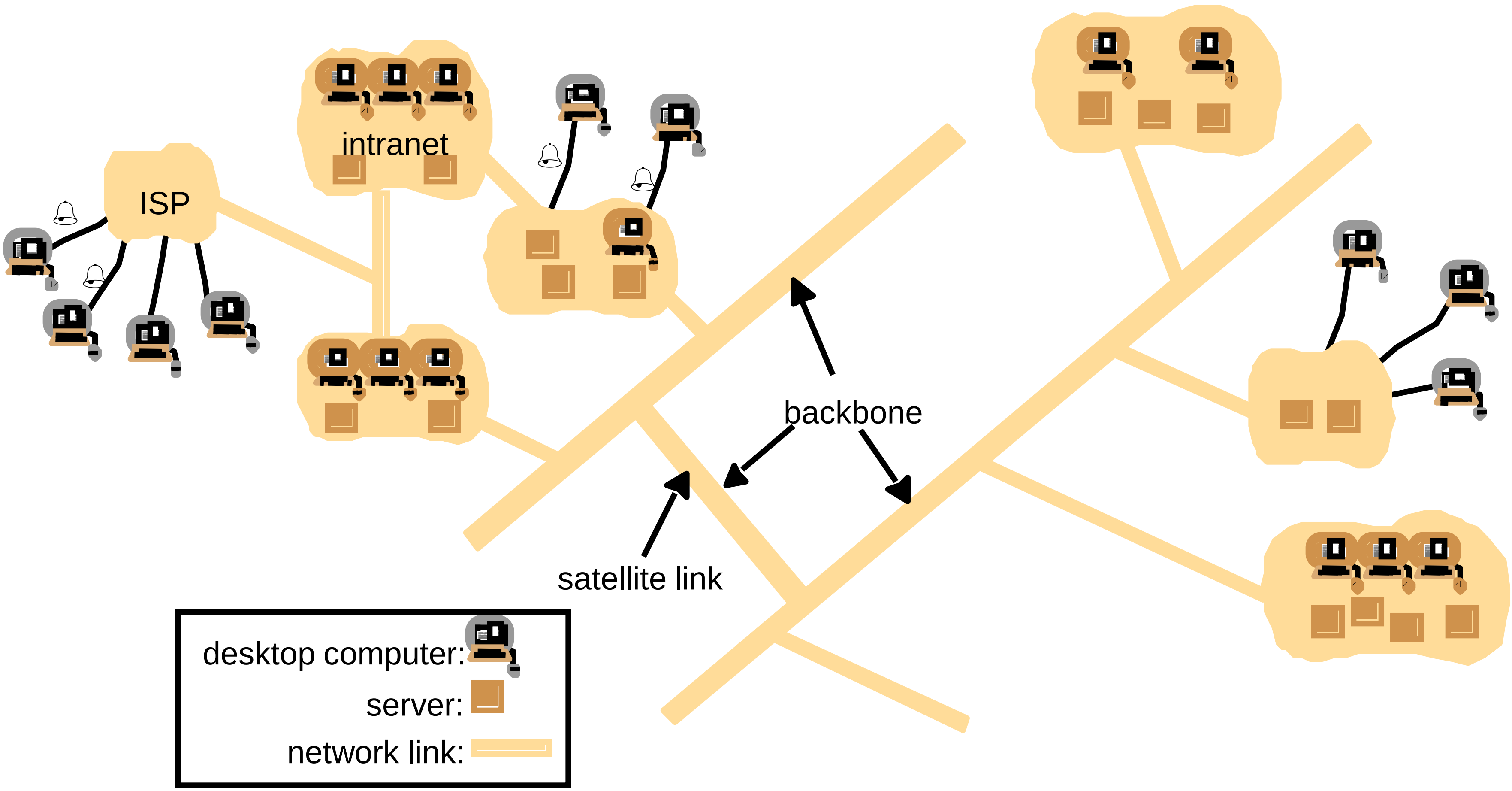
Types of Transparency

Access	local and remote resources are accessed using identical operations
Location	resources are accessed without knowledge of their location
Concurrency	several processes operate concurrently using shared resources without interference between them
Replication	multiple instances of resources appear as a single instance
Failure	the concealment of faults from users
Mobility	the movement of resources and clients within a system (also called migration transparency)
Performance	the system can be reconfigured to improve performance
Scaling	the system and applications can expand in scale without change to the system structure or the application algorithms

Typical Layering in DSs

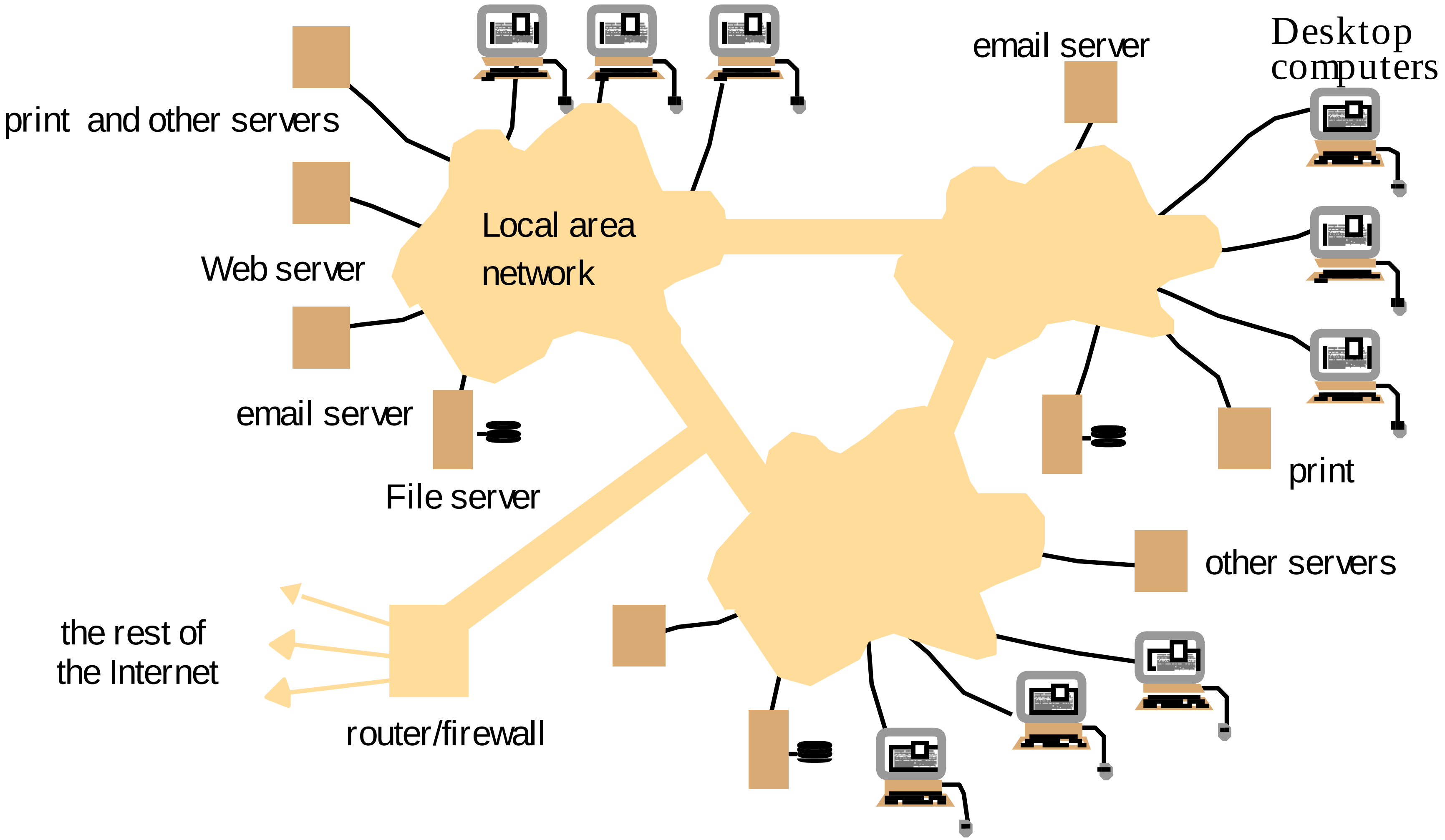


Example – Internet (Portion of it)



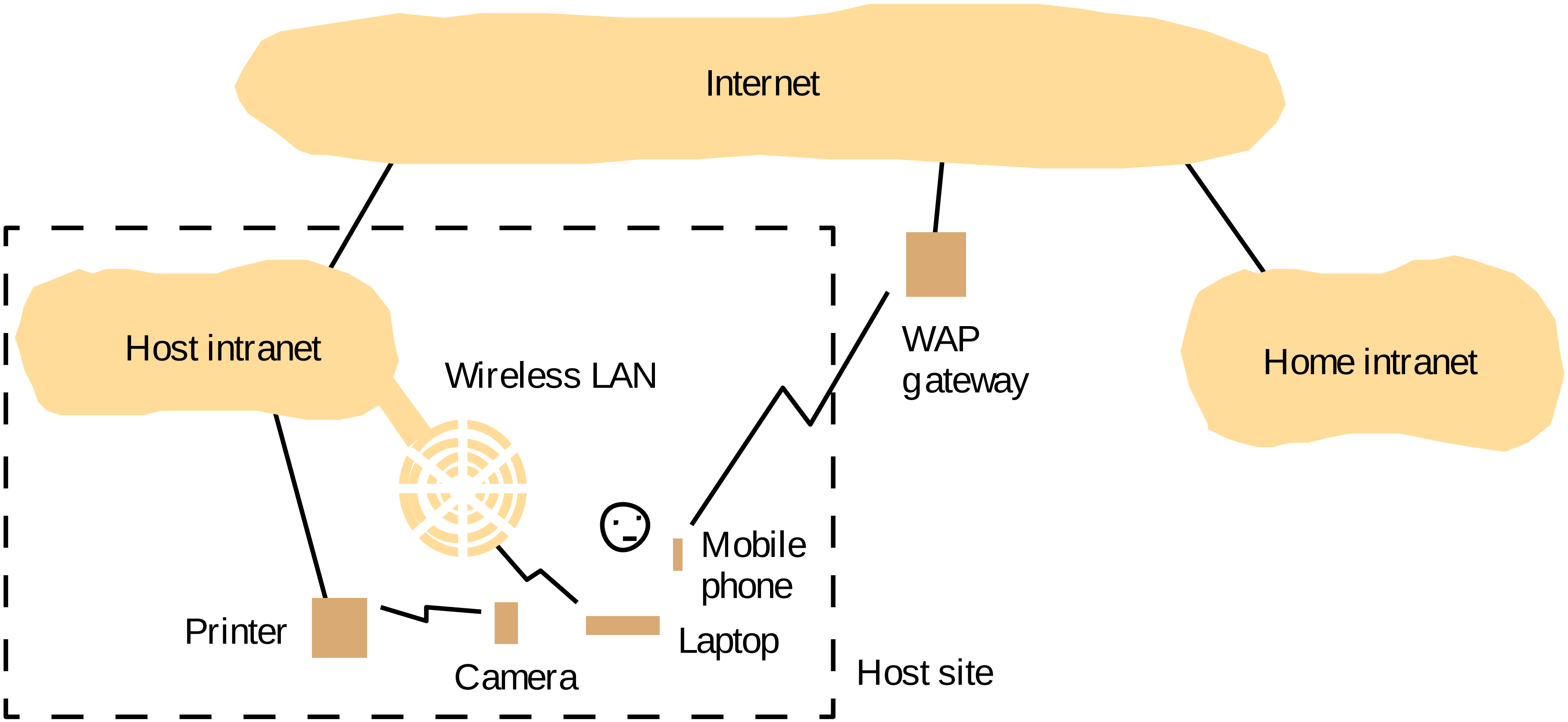
From Couloris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

Example – An Intranet



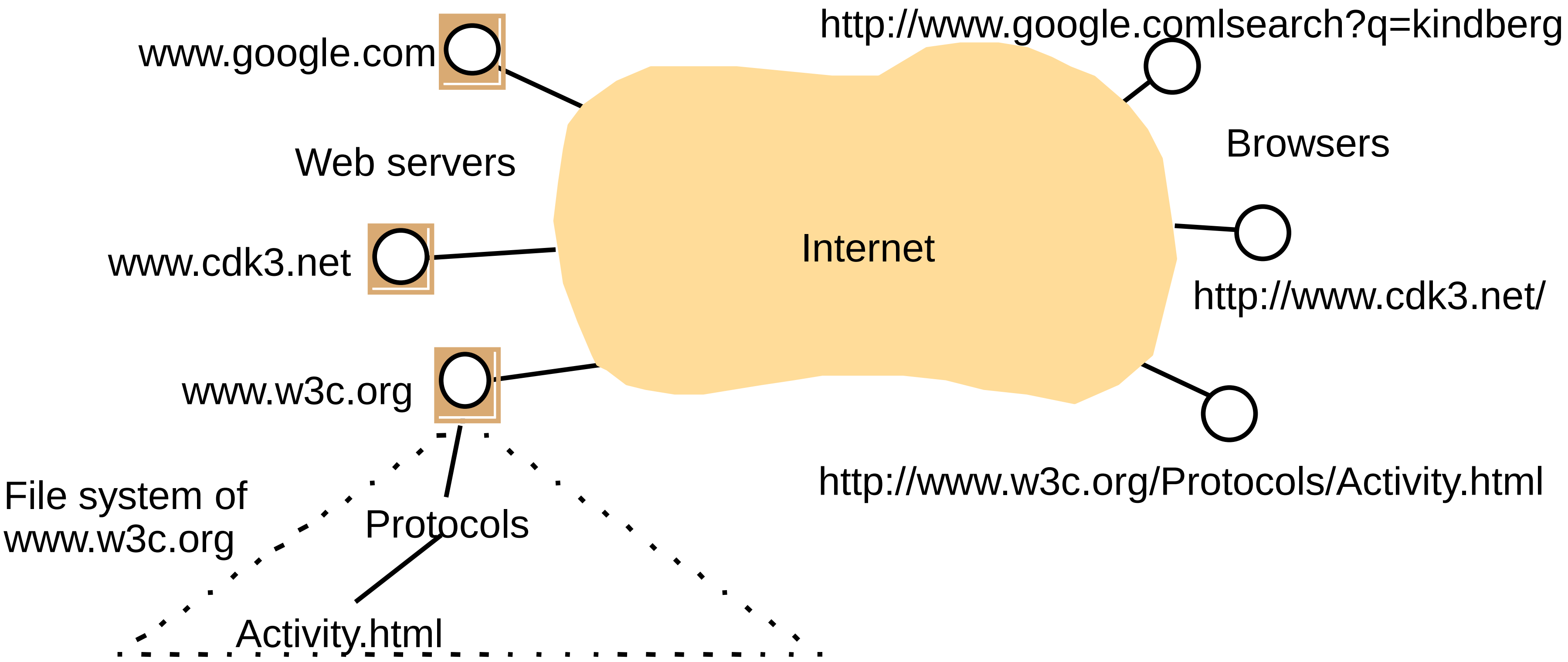
From Coulouris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

Example – Mobile Environment



From Coulouris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

Example - Web



From Coulouris, Dollimore and Kindberg, Distributed Systems: Concepts and Design, 3rd ed.

Challenges (I)

- Heterogeneity
  - Networks
  - Hardware
  - OS
  - Programming Languages
  - Implementations
- Openness
  - Can you extend and re-implement the system?
  - Public and published interfaces
  - Uniform communication mechanism

Challenges (II)

- Scalability
  - Can the system behave properly if the number of users and components increase?
    - → Scale-up: increase the number of “users” while keeping the system performance unaffected
    - → Speed-up: improvement in the system performance as system resources are increased
  - Impediments
    - → Centralized data
      - A single file
    - → Centralized services
      - A single server
    - → Centralized algorithms
      - Algorithms that “know it all”

Challenges (III)

- Failure handling
  - Partial failures
    - → Can non-failed components continue operation?
    - → Can the failed components easily recover?
  - Failure detection
  - Failure masking
  - Failure tolerance
  - Recovery
  - An important technique is replication
    - → Why does the space shuttle have 3 on-board computers?

Challenges (IV)

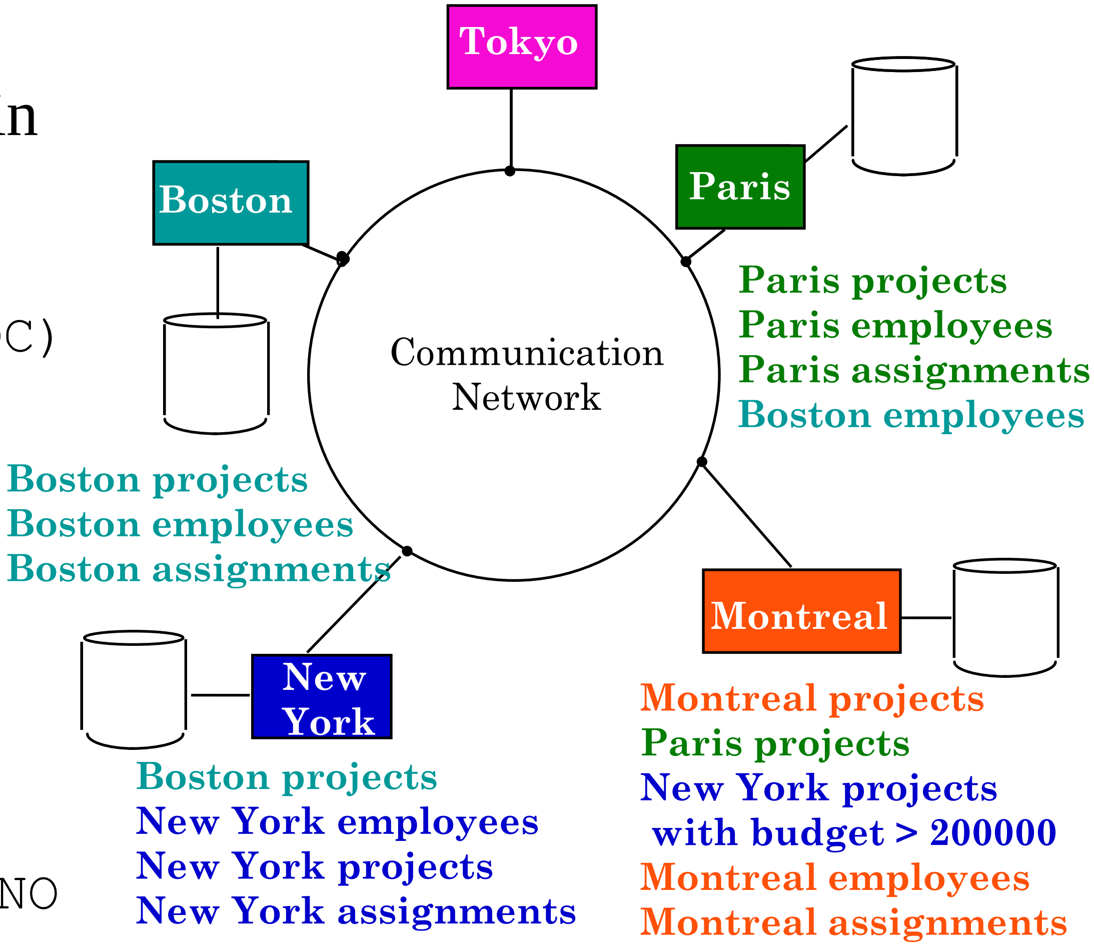
- Concurrency
  - Multiple “clients” sharing a resource ⇒ how do you maintain integrity of the resource and proper operation (without interference) of these clients?
  - Example: bank account
    - → Assume your account has a balance of $100
    - → You are depositing from an ATM a cheque for $50 into that account
    - → Someone else is withdrawing from the same account $30 using another ATM but at the same time
    - → What should the final balance of the account be?
      - $120
      - $70
      - $150

Challenges (V)

- Transparency
  - Not easy to maintain
  - Example:

```
EMP(ENO,ENAME,TITLE,LOC)
PROJECT(PNO,PNAME,LOC)
PAY(TITLE,SAL)
ASG(ENO,PNO,DUR)

SELECT  ENAME,SAL
FROM    EMP,ASG,PAY
WHERE   DUR > 12
AND     EMP.ENO = ASG.ENO
AND     PAY.TITLE = EMP.TITLE
```



Tokyo

Paris

Paris projects
Paris employees
Paris assignments
Boston employees

Boston

Communication Network

Boston projects
Boston employees
Boston assignments

Montreal

Montreal projects
Paris projects
New York projects with budget > 200000
Montreal employees
Montreal assignments

New York

Boston projects
New York employees
New York projects
New York assignments