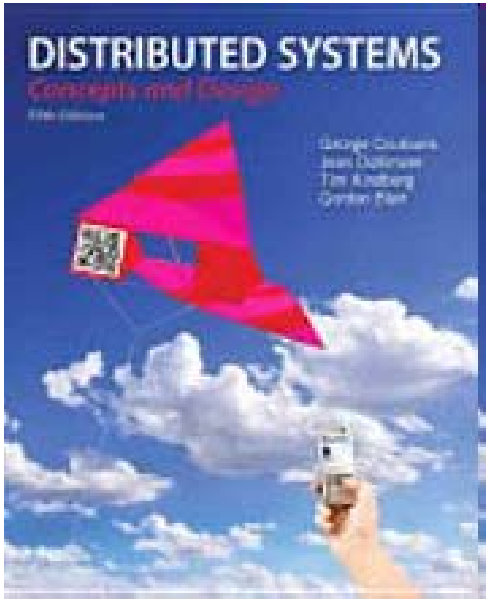
CS666 Advanced Distributed Systems (Spring 2018)

- Instructor: Tao Xie
- Office hours: MW 11 am – 12 pm; or by appointment
- Office: GMCS 535, phone extension 2014
- Email: txie@mail.sdsu.edu
- http://taoxie.sdsu.edu/cs666/index.html (class web page)
- Prerequisite: CS570 - Operating System (strictly enforced, show the grader your record by Jan. 24)
- Chapters 1,2,4,5,6,10,12,14,15,16,17,18

Text Book



- G. Coulouris, J. Dollimore and T. Kindberg, "Distributed Systems: Concepts and Design" fifth edition, by Addison Wesley, 2011, ISBN: 0-13-214301-1

Grading Policy

- Three Assignments ...15%
- Five Quizzes ...10%
- Class Participation ...5%
- Midterm Exam ...30%
- Research Project Proposal (two-page) ...5%
- Research Project Intermediate Report (two-four page) ...5%
- Research Project Presentation (10 minutes each group) ...10%
- Research Project Technical Report (8-page, IEEE format)...20%

The Grader (Prashant Joshi)

- Office hours: Monday&Wednesday 11am to noon at GMCS 557

- Please send a Hello email to the grader. In the email, tell the grader your RedID and full name by Jan. 24 Wednesday.

- Also, please email him a screenshot of your transcript to show the grader that you passed CS 570 Operating System in the Hello email.

Class Guidelines

- **Prerequisite**: The prerequisite for this course is CS570. Those who fail to provide this proof by **Jan. 24** will be dropped off.
- There will be **NO make-up** midterm exam, quizzes, and presentations without a verified excuse.
- I will **not sign late drop slips**!
- Failure to appear for the **Midterm Exam** at the indicated date and time will result in a grade of "**F**" in the course. No make-up exam will be given.
- **Incompletes**: To receive a grade of Incomplete ("I") in this class, you must meet all of the following criteria: a. You must have extenuating circumstances beyond your control for not completing the course, and I will be the sole judge as to whether the circumstances warrant withdrawal from the class. Official verification is required to corroborate your circumstances. b. You must have completed the Midterm Exam. c. You must have a grade of "C" or better in the Midterm Exam.
- **No late submission** will be accepted.
- Any questions about grading must be brought to the attention of the grader or the instructor within **one week** after the item in question is returned. No argument on grading after this one week period.

Project

- Note that only project is group-oriented.

- Each group has 2 students.

- Project takes 40% of the final grade in this class. Click the link below and start your project ASAP.

- http://taoxie.sdsu.edu/cs666/projects.htm

Consequences of Cheating

- **Any one** caught cheating/collaborating on **an** exam or **any** assignment will receive an **F** in the course.
- The incident will be reported to the **Office of Judicial Affairs** for disciplinary proceedings. Note: If, for instance, you allow your assignment to be copied by a classmate, you are considered as guilty as the copier.
- Do NOT take chances!

Objectives Today

- Definition of distributed system and its characteristics: concurrency, independent failure of components and lack of a global clock.

- Realistic examples of distributed systems: the Internet, an intranet and mobile computing.

- The benefits of resource sharing using the Web as an example.

- A good understanding of the challenges related to heterogeneity, openness, security, scalability, failure handling, concurrency and transparency.

Networking and Parallel Computing

- Computer networking
  - Hardware that connects computers
  - Software that sends/receives messages from one computer to another, which might be on different networks (end to end delivery)
  - Goal is to transmit messages reliably and efficiently
- Parallel Computing
  - Multiple homogeneous processors in “one” computer
  - Shared or distributed memory
  - Goal is to execute a program faster by division of labor

Distributed Computing

- Networked computers that could be far apart
  - rely on computer networking
- Communicate and coordinate by sending messages
- Goal is to share (access/provide) *distributed* resources
- Issues:
  - Concurrent execution of processes
  - No global clock for coordination
  - More components, more independent failures

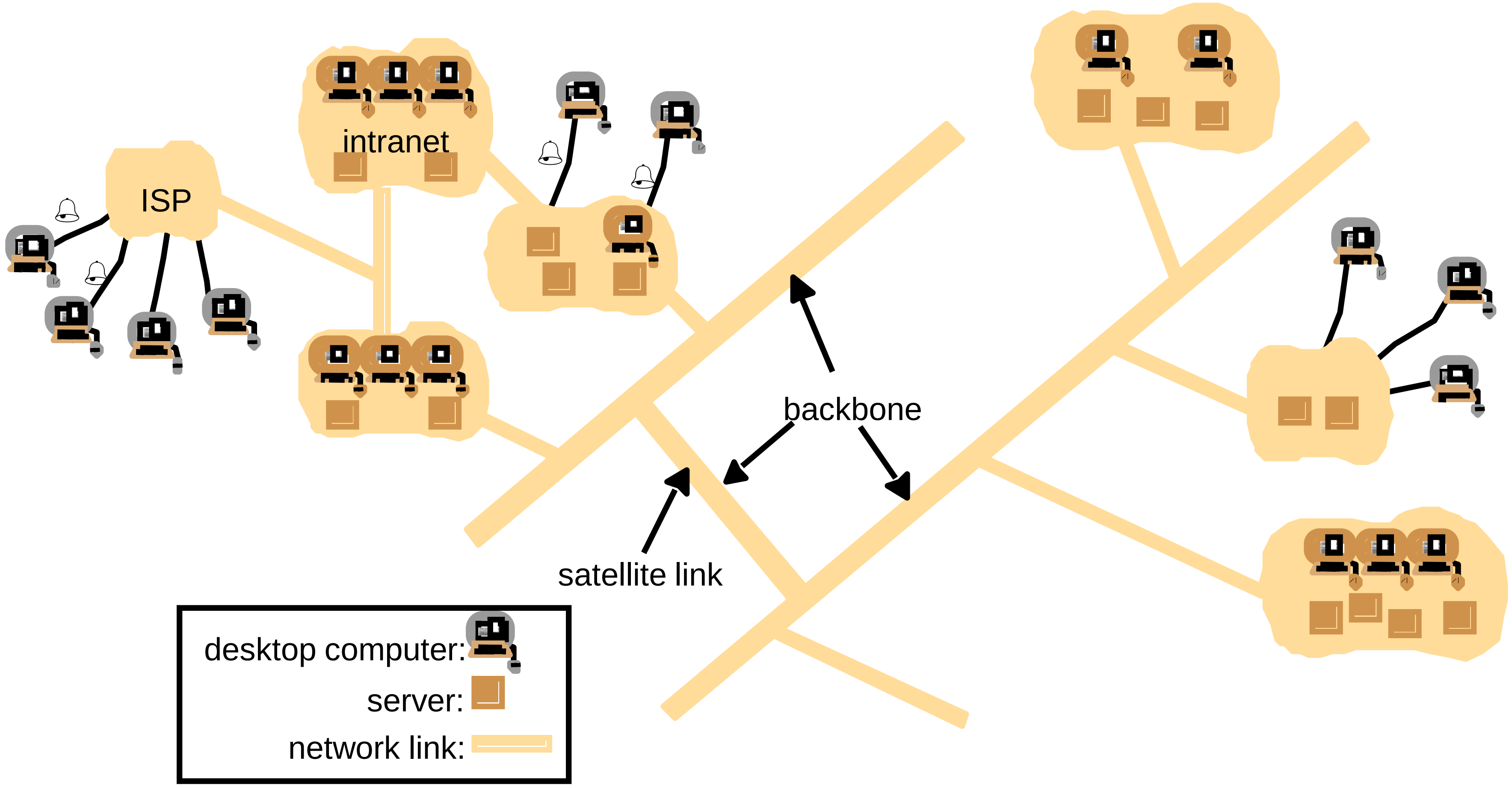
Definition of Distributed System

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

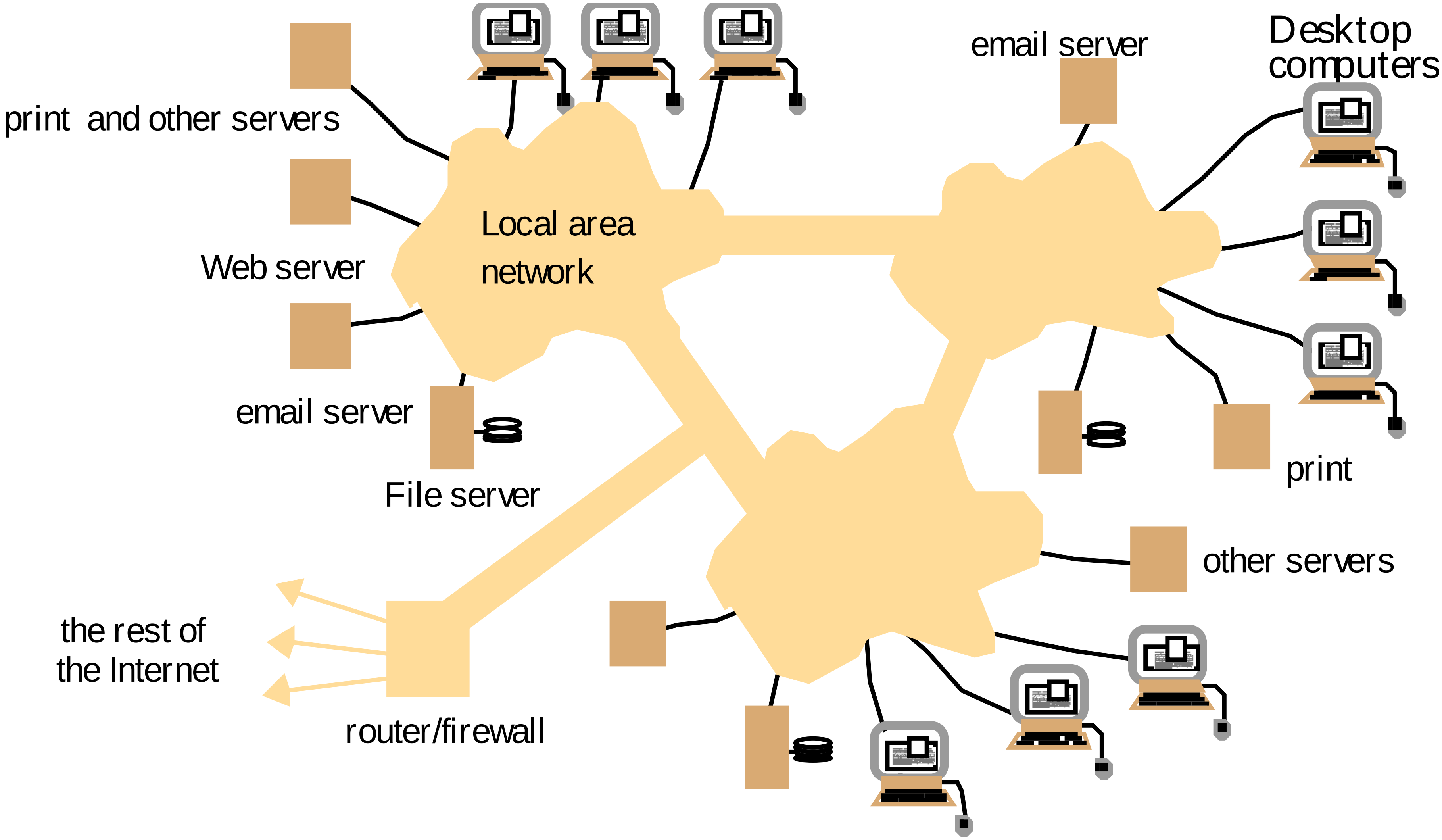
Examples of Distributed Systems

- Global Internet
- Organizational Intranets--behind router/firewall
- Mobile Computing -- computers move
- Ubiquitous Computing -- computers embedded everywhere
- Issues:
  - discovery of resources in different host environments
  - dynamic reconfiguration
  - limited connectivity
  - privacy and security guarantees to the user and the host environment

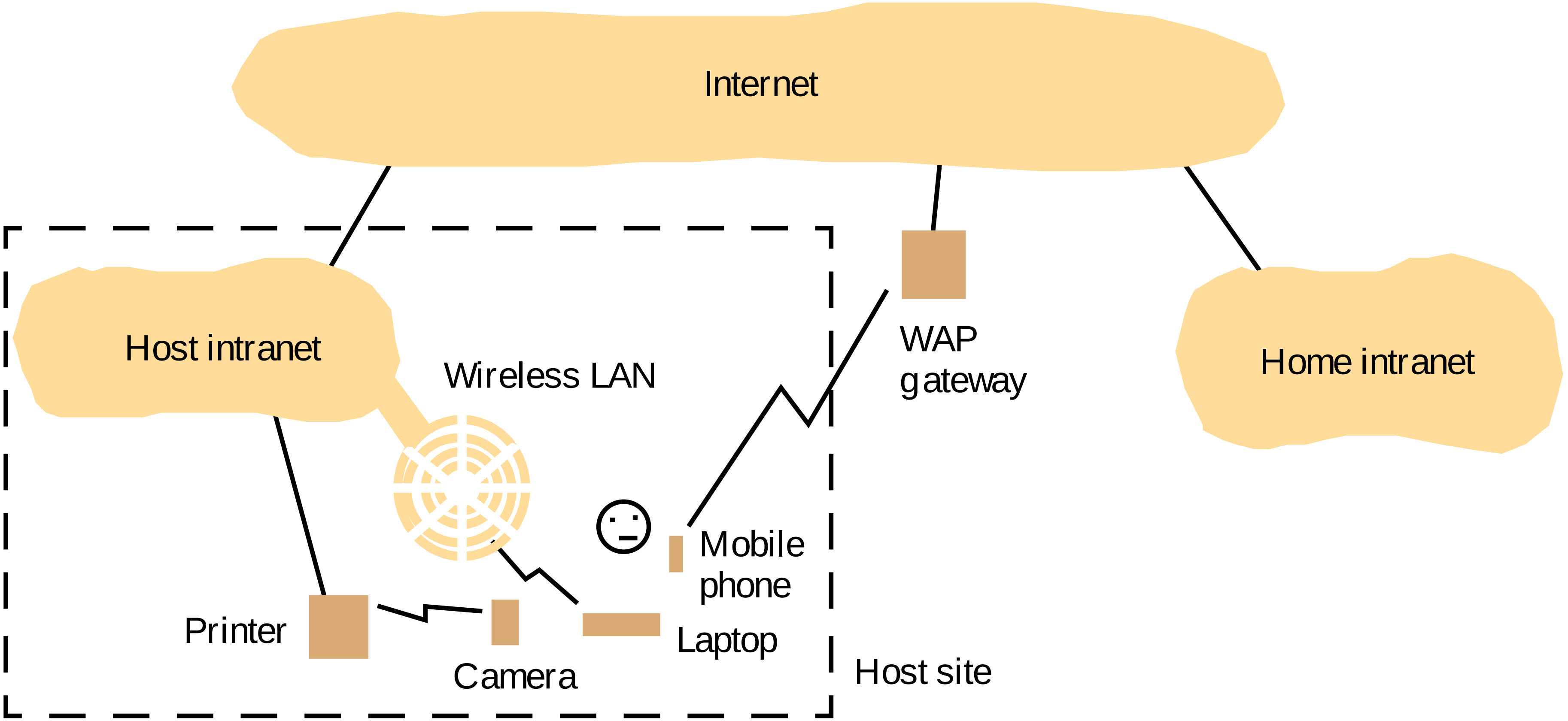
A Typical Portion of the Internet



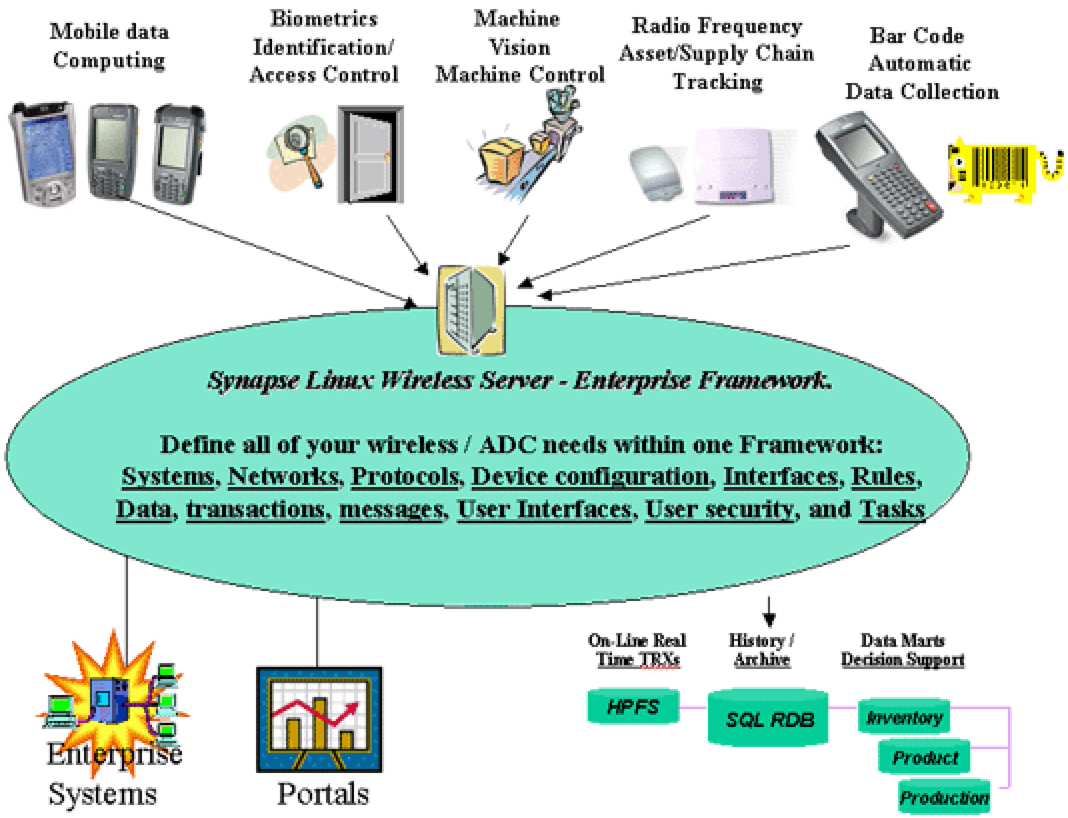
A Typical Intranet



Portable and handheld devices



Mobile Computing


Mobile data Computing
Biometrics Identification/ Access Control
Machine Vision Machine Control
Radio Frequency Asset/Supply Chain Tracking
Bar Code Automatic Data Collection

Synapse Linux Wireless Server - Enterprise Framework.

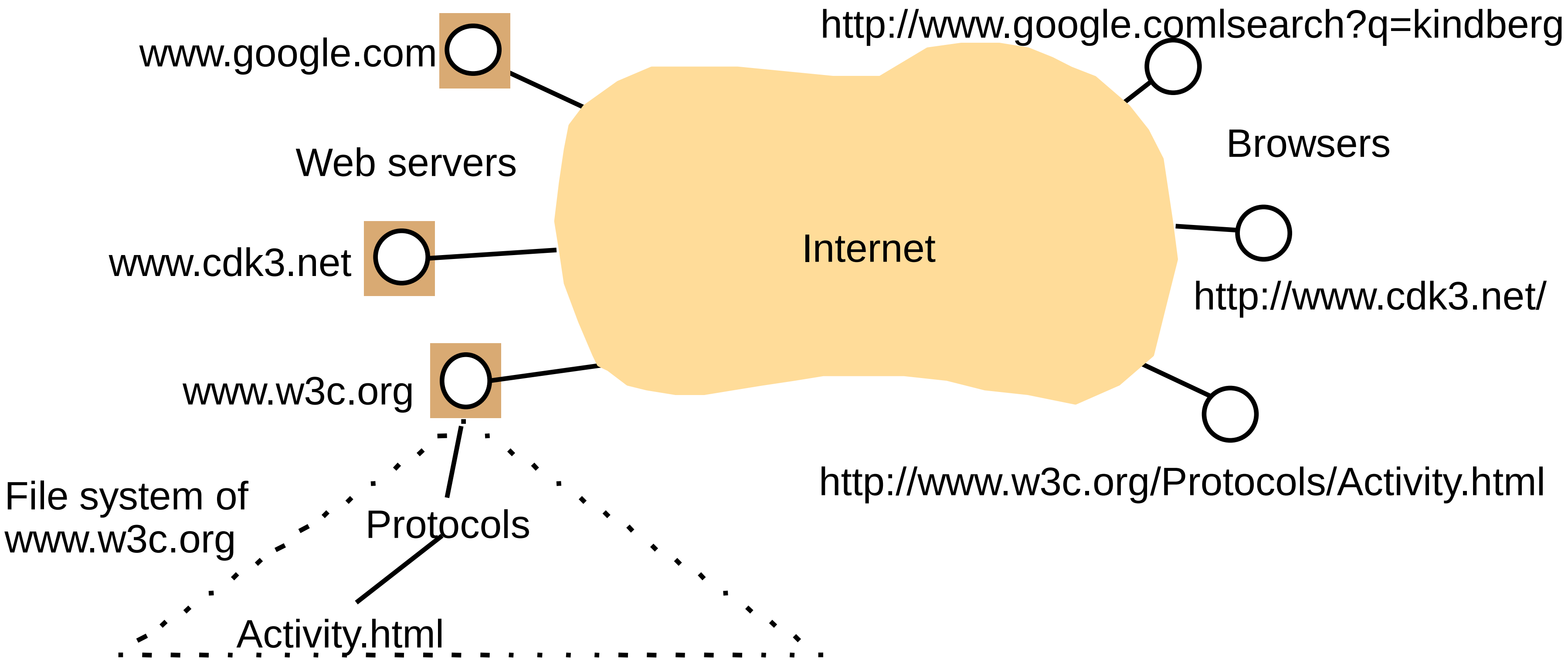
Define all of your wireless / ADC needs within one Framework: Systems, Networks, Protocols, Device configuration, Interfaces, Rules, Data, transactions, messages, User Interfaces, User security, and Tasks

Enterprise Systems
Portals
On-Line Real Time TRXs
History / Archive
Data Marts Decision Support
HPFS
SQL RDB
Inventory
Product
Production

Resources Sharing and the Web

- HTML, Hyper Text Markup Language
- URL, Uniform Resource Locator
  - http://servername[:port] [/pathname] [?arguments]
- HTTP, HyperText Transfer Protocol
  - request-reply protocol (client-server)
  - content types--MIME types, multipurpose internet mail extensions
  - one resource per request
  - simple access control (mostly public)

Web Servers and Web Browsers



Other Web Technologies

- web forms
- CGI programs, common gateway interface, run on the server
- applets, run on the client
- RDF, resource description framework, vocabulary for meta-data
- XML, extensible markup language, allow meta-data information to be included

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
		42,298,371	

Challenges and Issues (1)

- Heterogeneity
  - networks, hardware, OS, languages...
  - middleware—corba (Common Object Request Broker Architecture)
  - mobile code, virtual machines
- Openness
  - extended and re-implemented in various ways
  - standard published interfaces
  - RFC, request for comments
- Security
  - confidentiality
  - integrity
  - availability

Challenges and Issues (2)

- Scalability
  - effective with significant increase in resources
  - cost
  - performance
- Failure handling
  - detecting
  - masking—hide, less severe (retransmit)
  - tolerating--ignore, timeout
  - recovery--logs, rollback
  - Redundancy
- Concurrency

Challenges and Issues (3)

- Transparency
  - *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.

Challenges and Issues (4)

- Transparency
  - *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.

Assignment1 (chapter1)

- Exercise (Page 34)
- 1.1
- 1.2
- 1.6

- Assignment1 will be due one week after we finish Chapter 5.