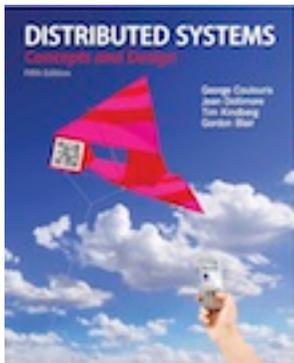


# Slides for Chapter 1

## Characterization of Distributed Systems

---



*From* **Coulouris, Dollimore, Kindberg and Blair**  
**Distributed Systems:**  
**Concepts and Design**

Edition 5, © Addison-Wesley 2012

Note: not lecturing on all of Chapter 1, but you must read it

## Figure 1.1 (see book for the full text)

### Selected application domains and associated networked applications

<i>Finance and commerce</i>	eCommerce e.g. Amazon and eBay, PayPal, online banking and trading
<i>The information society</i>	Web information and search engines, ebooks, Wikipedia; social networking: Facebook and MySpace.
<i>Creative industries and entertainment</i>	online gaming, music and film in the home, user-generated content, e.g. YouTube, Flickr
<i>Healthcare</i>	health informatics, on online patient records, monitoring patients
<i>Education</i>	e-learning, virtual learning environments; distance learning
<i>Transport and logistics</i>	GPS in route finding systems, map services: Google Maps, Google Earth
<i>Science</i>	The Grid as an enabling technology for collaboration between scientists
<i>Environmental management</i>	sensor technology to monitor earthquakes, floods or tsunamis

Figure 1.2  
An example financial trading system

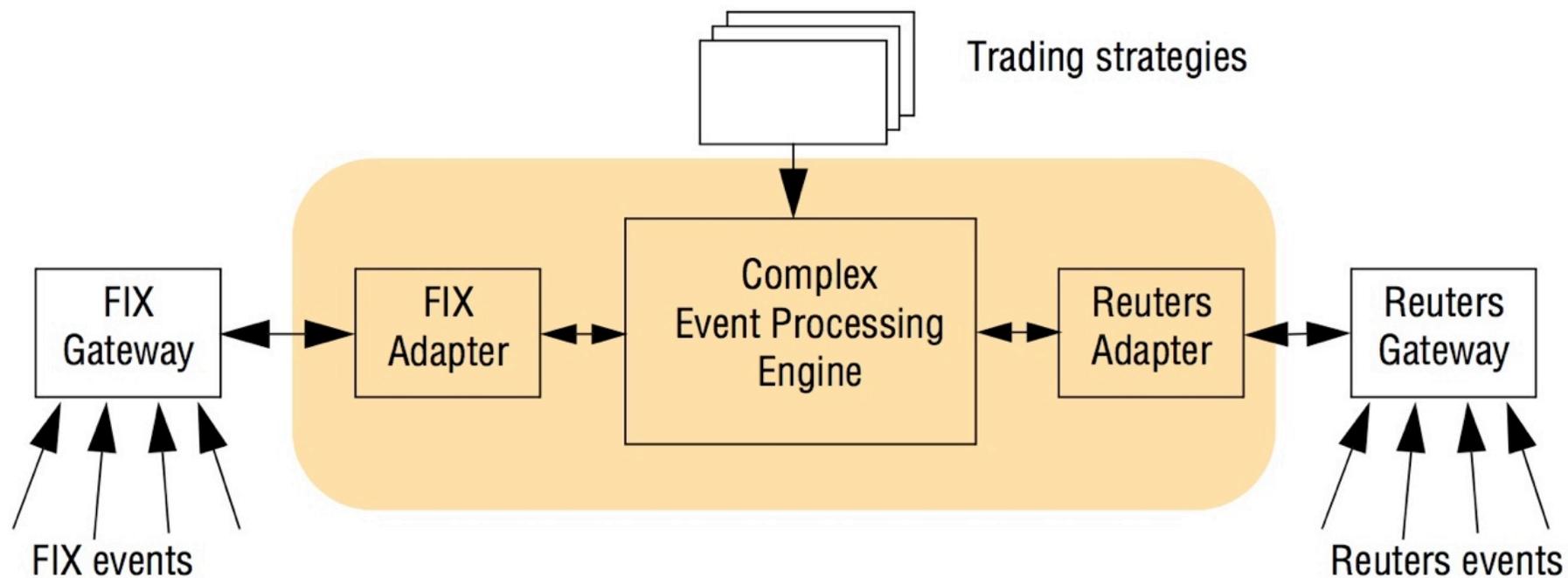
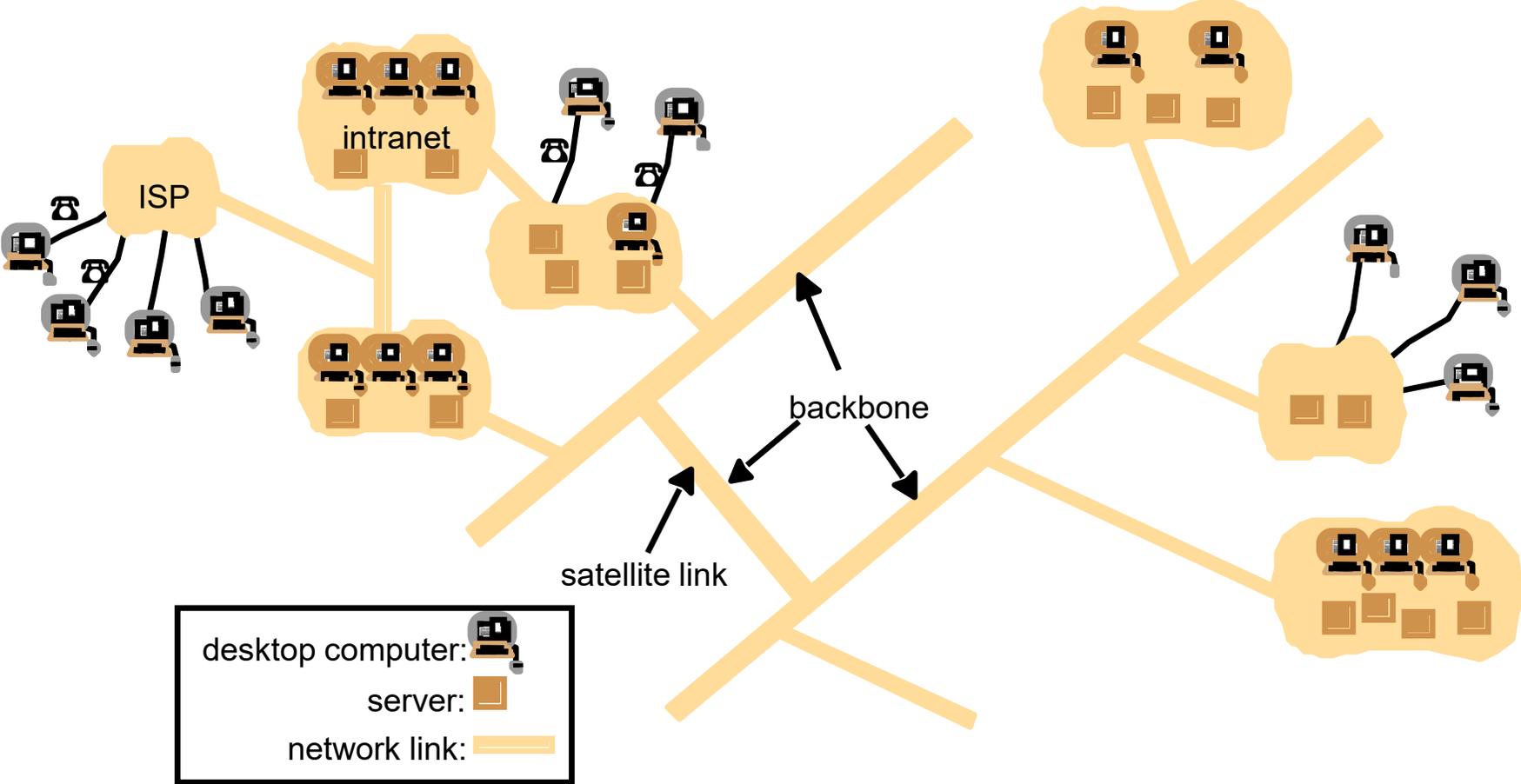


Figure 1.3  
A typical portion of the Internet



## 1.3 Mobile and Ubiquitous Computing

**Mobile Computing**: a computer that can move around with you.

- Laptops
- Tablets
- Wearable devices: watches, clothes (??)
- Smart phones

Sometimes are **Location Aware**: can seamlessly access local printers or other resources

**Context Awareness**: a more logical/abstract mapping onto location awareness

## Ubiquitous Computing

**Ubiquitous Computing**: harnessing many small, cheap computational devices present in users' physical environments.

**Q: hot term for this lately?**

- No UI: it blends into the environment
- Text (2011) is a bit dated on this area: “it *may* be convenient for users to control their washing machine or their entertainment system from their phone or a ‘universal remote control’ device in the home. Equally, the washing machine *could* notify the user via smart badge or phone when the washing is done. (*emphasis mine*)

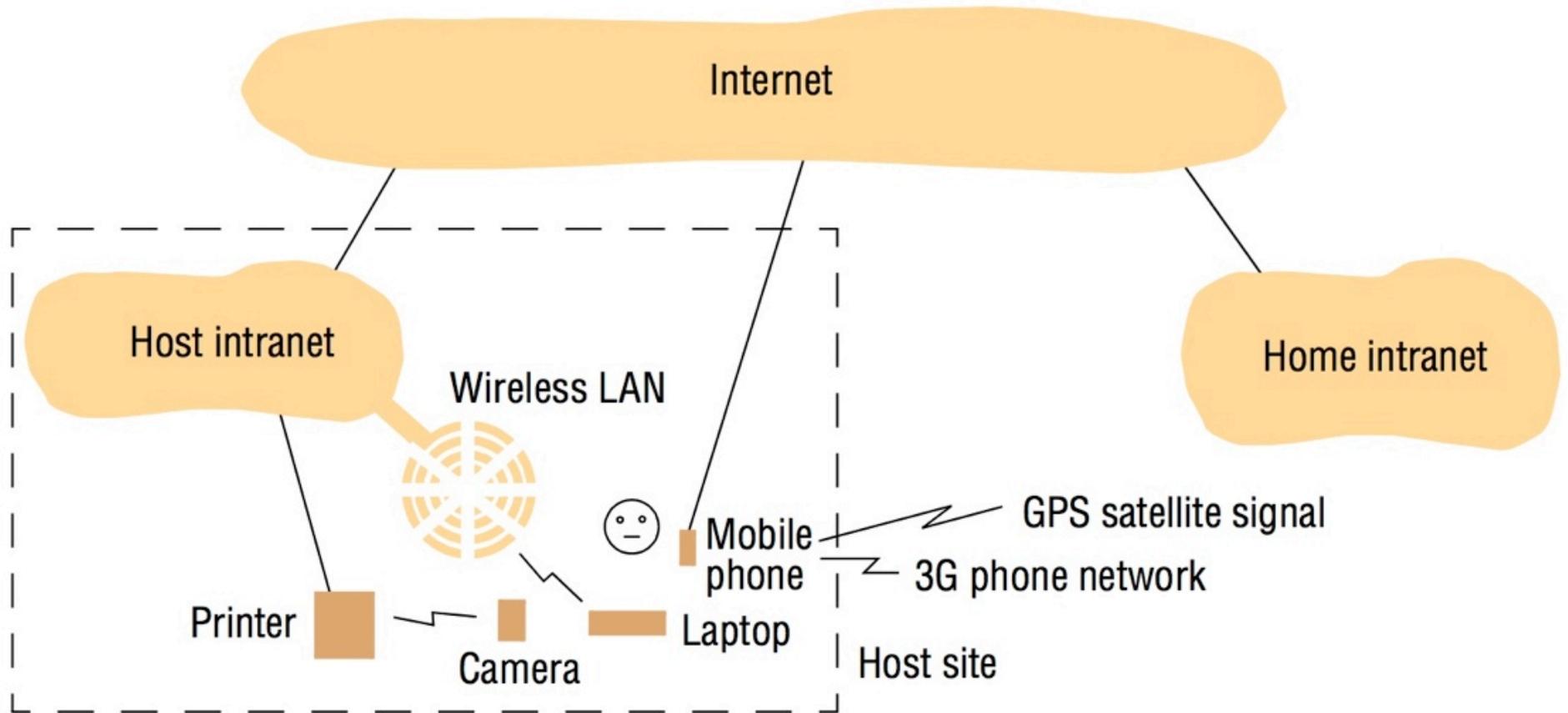
## Ubiquitous Computing (cont.)

**Spontaneous interoperation**: associations between devices are routinely created and destroyed

- Need to support this for location/context-aware ubiquitous computing
- **Service discovery**: associating a device with suitable local services.

Note: mobile and ubiquitous have some overlap.

Figure 1.4  
Portable and handheld devices in a distributed system



# Figure 1.5 Cloud computing

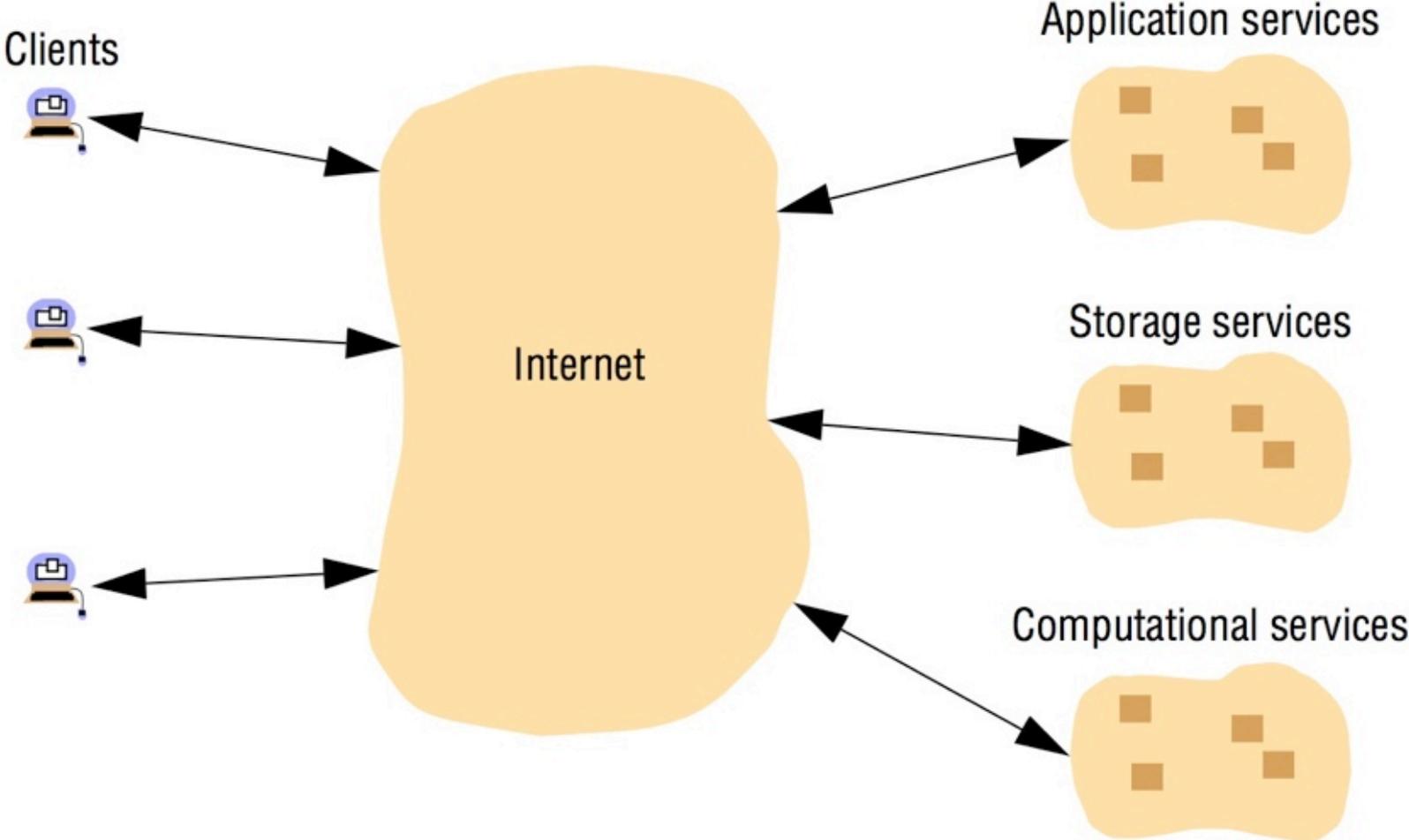


Figure 1.6  
Growth of the Internet (computers and web servers)

---

<i>Date</i>	<i>Computers</i>	<i>Web servers</i>	<i>Percentage</i>
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, July	~200,000,000	42,298,371	21
2005, July	353,284,187	67,571,581	19

---

## 1.5.1 Heterogeneity

### **Heterogeneity** (variety/difference) in DSs

- Network technology
- CPU architecture / computer hardware
- Operating systems
- Programming languages
- Implementations by different developers

**Q: what is the antonym for heterogeneity?**

Middleware helps us with the above heterogeneties

## Section 1.5.7

### Transparencies

*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.

*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.

## 1.5.8 Quality of Service

**Quality of Service (QoS)**: the “non-functional” properties (beyond the API)

- Reliability
- Security
- Performance
- Adaptability

Text: “In fact, the abbreviation QoS has effectively been commandeered to refer to the ability of systems to meet such deadlines.”

You will get some modest exposure to QoS in your programming assignments

Figure 1.7  
Web servers and web browsers

