Outline

- Announcements
- Review Alibris Case Study
- Information Technology
- Student Presentations
- Client-Server Architecture
- The SUN N-Tier Architecture
Announcements

- Assignment 3 will be posted next week
- Review Sessions
  - 2/4/2011, 1:00-3:00 p.m. (E2-280)
  - 2/7/2011, 5:00 - 7:00 p.m. (social Sciences 2 room 071)
Review - Alibris

- A start-up to sell used books on the Internet.
- Interloc, Alibris’ predecessor, functioned like a classified ads page for book dealers
- Alibris changing Interloc’s model
  - Actually sell the books
  - Charge a fee per sale (instead of per listing)
  - Intermediary strategy
    - Buy books from dealers
    - Ship to warehouse
    - Re-pack, consolidate order, ship to customer
Architecture
What is Architecture?

How do you architect a solution?
A system is decomposed into interacting subsystems.

Each subsystem may have a similar internal decomposition.
System Architecture

- **System**: A composition of subsystems that cooperate to accomplish some purpose

- **Sub-system**: An element within the system that performs some well-defined action on behalf of the system
Time sharing

ASCII terminal (no graphics)

Mainframe (database and application server)

Point-to-point wire (no network)
Two-tier client/server

Local-area network

Server/Mainframe
Email application

Email client sends message to server

Message is stored on POP server

Later, recipient’s email client retrieves message from server

Slide adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice
Chat application

Chat clients send user’s typing to server

Chat server aggregates typing from all users and sends to all clients

Other user’s clients display aggregated typing from chat server

Slide adapted from slides for *Understanding Networked Applications*
By David G. Messerschmitt. Copyright 2000. See copyright notice
System integration

Architecture
-> subsystem implementation
    -> system integration
Bring together subsystems and make them cooperate properly to achieve desired system functionality

- Always requires testing
- May require modifications to architecture and/or subsystem implementation
Why system decomposition in subsystems?

- Divide and conquer approach to reduce complexity
- Reuse components
- In accordance with industry structure
- Others?
Infrastructure Layering

- Infrastructure decomposed into layers
- Each layer
  - depends on the layer below
  - provides services to the layer above
  - Only interacts with layers immediately above or below

- E.g software is “riding on top of” equipment
  - Software itself is also layered
### Simplified infrastructure layering

<table>
<thead>
<tr>
<th>Application</th>
<th>Middleware</th>
<th>Operating System</th>
<th>Equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed object management</td>
<td>Database management</td>
<td>File system</td>
<td>Storage peripherals</td>
</tr>
<tr>
<td>Network software</td>
<td></td>
<td></td>
<td>Storage</td>
</tr>
<tr>
<td>Network equipment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communications</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Communications: Storage peripherals
- Network software: File system
- Distributed object management: Database management
- Network equipment: Operating system
- Application: Middleware
- Equipment: Storage
## Simplified infrastructure layering

<table>
<thead>
<tr>
<th>Application</th>
<th>Middleware</th>
<th>Equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed object management</td>
<td>Database management</td>
<td>Operating system</td>
</tr>
<tr>
<td>Network software</td>
<td>File system</td>
<td>Equipment</td>
</tr>
<tr>
<td>Network equipment</td>
<td>Storage peripherals</td>
<td>Storage</td>
</tr>
<tr>
<td>Communications</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Operating system functions

- Graphical user interface (client only)
- Hide details of equipment from the application
- Multitasking
- Resource management
  - Processing, memory, storage, etc
  etc
File system (OS)

- Hides details of storage equipment from applications
- Enables services such as creating/accessing files

A File is:

- Collection of data managed for the benefit of the application
  - E.g. word document, excel spreadsheet
  - Size known, but unspecified structure and interpretation
- Name
- Location in naming hierarchy
Network equipment

- Hosts
- Switches
- Backbone links
- Access links
Messages and packets

Simplest network communication service is the message

- Smallest unit of communicated data meaningful to application
- Size, but unknown structure and interpretation
- Analogous to file in storage

Internally, the network may fragment a message into packets, and reassemble those packets back into a message
## Simplified infrastructure layering

<table>
<thead>
<tr>
<th>Application</th>
<th>Middleware</th>
<th>Operating system</th>
<th>Equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed object management</td>
<td>Database management</td>
<td>File system</td>
<td>Storage peripherals</td>
</tr>
<tr>
<td>Network software</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network equipment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communications</td>
<td></td>
<td></td>
<td>Storage</td>
</tr>
</tbody>
</table>

- Communications Storage
- Network equipment Storage peripherals
- Network software File system
- Distributed object management Database management
- Middleware
- Operating system
Middleware Functions

- **Capabilities that can be shared by many applications, but that is not part of OS**
  - Example: Database Management System (DBMS)

- **Hide details of OS from application**
  - Java Virtual Machine

- **More purposes we’ll talk about later.**
Communication middleware

- Location independence
  - makes distributed application look similar to centralized

- Many possible other functions
Storage middleware

- **Database**
  - File with specified structure
  - Example: relational table
  - Oriented toward business applications

- **Database management system (DBMS)**
  - Manage multiple databases
  - Basis of online transaction processing (OLTP)
### A Database

<table>
<thead>
<tr>
<th>Year</th>
<th>City</th>
<th>Accommodation</th>
<th>Tourists</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>Oakley</td>
<td>Bed&amp;Breakfast</td>
<td>14</td>
</tr>
<tr>
<td>2002</td>
<td>Oakley</td>
<td>Resort</td>
<td>190</td>
</tr>
<tr>
<td>2002</td>
<td>Oakland</td>
<td>Bed&amp;Breakfast</td>
<td>340</td>
</tr>
<tr>
<td>2002</td>
<td>Oakland</td>
<td>Resort</td>
<td>230</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Camping</td>
<td>120000</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Bed&amp;Breakfast</td>
<td>3450</td>
</tr>
<tr>
<td>2002</td>
<td>Berkeley</td>
<td>Resort</td>
<td>390800</td>
</tr>
<tr>
<td>2002</td>
<td>Albany</td>
<td>Camping</td>
<td>8790</td>
</tr>
<tr>
<td>2002</td>
<td>Albany</td>
<td>Bed&amp;Breakfast</td>
<td>3240</td>
</tr>
<tr>
<td>2003</td>
<td>Oakley</td>
<td>Bed&amp;Breakfast</td>
<td>55</td>
</tr>
<tr>
<td>2003</td>
<td>Oakley</td>
<td>Resort</td>
<td>320</td>
</tr>
<tr>
<td>2003</td>
<td>Oakland</td>
<td>Bed&amp;Breakfast</td>
<td>280</td>
</tr>
<tr>
<td>2003</td>
<td>Oakland</td>
<td>Resort</td>
<td>210</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Camping</td>
<td>115800</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Bed&amp;Breakfast</td>
<td>4560</td>
</tr>
<tr>
<td>2003</td>
<td>Berkeley</td>
<td>Resort</td>
<td>419000</td>
</tr>
<tr>
<td>2003</td>
<td>Albany</td>
<td>Camping</td>
<td>7650</td>
</tr>
<tr>
<td>2003</td>
<td>Albany</td>
<td>Bed&amp;Breakfast</td>
<td>6750</td>
</tr>
</tbody>
</table>

- The DBMS enables updating and searching the database.

- **QUERIES**
  - E.g. “How many B&B are there in Berkeley?”
  - E.g. “What accommodation did most tourists visiting Oakley preferred?”
Some DBMS functions

- Logical structure separated from physical structure
- Platform independence
- Implement standard queries
- Access from multiple users/applications
- Manage data as asset separate from applications
The Internet
Intranet

Private internet

Often connected to Internet

- Firewall creates a protected enclave

Global Internet
Extranet

An Extranet is composed of

- Intranets connected through an unprotected domain (typically the Internet)

- Encryption and other security technologies used to
  - protect proprietary information
  - prevent imposters, vandals, etc
Communication between intranets encrypted.
What is the Internet?

- An **internet** is a “network of networks”
  - Interconnect standard for LAN’s, MAN’s, and WAN’s
- **Internet** = the major global internet
- A private internet is called an **intranet**
- An **extranet** is an interconnection of intranets through the Internet
Extranet

Consumers, field workers, etc.

Intranet

Internet
Lock icon indicates this is an extranet.
Certificate is the server’s credential
Questions

What business purposes do nomadic workers serve?
Mobile?
What advantage does direct Internet access have over long distance telephony?
Ideas and examples (Chapters 4-5)

by

David G. Messerschmitt
Peer to peer

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Three-tier client/server

Presentation

Note: many clients per application server, several application servers per data server

Local-area network

Application logic

Shared data

Slide adapted from slides for *Understanding Networked Applications* by David G Messerschmitt. Copyright 2000. See copyright notice.
Book merchant

Web browser

Web server

Customer logic

Fullfillment logic

Databases

Customers

Merchandise

Orders

Inter-enterprise e-commerce

Consumer e-commerce

Outside links

Acquirer bank

Book distributors

Slide adapted from slides for *Understanding Networked Applications*
By David G Messerschmitt. Copyright 2000. See copyright notice
Three-tier client/server

Client

Enterprise data server

Application server
Inter-organizational computing
E-commerce
Emergence

Subsystems are more specialized and simpler functionality
Higher-level system functionality arises from the interaction of subsystems

Emergence includes capabilities that arise purely from that interaction (desired or not)

- e.g. airplane flies, but subsystems can’t
Major subsystems

- Presentation
- Application software
- Logic Data
- Infrastructure software
- Network
- Infrastructure equipment
- Client host
- Server host
Layering builds capability incrementally by adding to what exists.

Elaboration or specialization

Existing layers
Layering

- **Major Categories**
  - Consumer (B2C)
  - Inter-consumer (C2C)
  - Inter-enterprise (B2B)

- Elaboration or specialization
- Services

Existing layers
Sun Case
Java Applets

- Key feature of Java

- **Applets**: chunks of Java code
  - Initially enabled animations on web pages
  - Later used to facilitate e-commerce applications, in cellular phones, etc.

- Applets are downloaded through the browser
  - Only what and when was needed
  - No need to keep a copy on client!

- **Servlets**: Applets that run on Webtop servers
**Sun N-Tier**

**Step 1:** The user logs into his client and calls down an application. This message is sent to the Application Server.

**Step 2:** An initial applet is sent to the client. At the same time a servlet is sent to the Webtop Server.

**Step 3:** The applet talks back and forth with the Webtop Server via the LAN. At the same time the servlet talks back and forth with the App Server via a WAN.

**Step 4:** As new data is received (i.e., a new customer’s name) the App Server communicates with the database to update that information.

**Local:** The Webtop Server and client communicate via a LAN.

**Remote:** The database and App Server communicate with the Webtop Server via a WAN.

**Exhibit 3** How the N-tier Architecture Works
What would you do...

- If you were in the executive board of Microsoft?
Sun N-tier case

- What does Sun make?
  - Workstations
  - Servers
  - Software
How Successful had Sun been up to 1998?

- Founded in 1982
- Open Standards Workstation
  - Unix Operating System (Solaris)
- 1988 - Revenues $1 billion
- 1993 - Market value $3.6 billion
  - Known for its workstations (addressing engineers)
- 1997 - Jumped from 3rd to 1st in Unix Server Market.
  - Achieved a 75% year-over-year increase in total server shipments
How Successful had Sun been up to 1998?

- “Last standing, fully integrated computing company”*
- Designed its own hardware/software:
  - SPARC microprocessor
  - Solaris OS
  - Networking capabilities (TCP/IP)

- Scalable, Reliable network computing for large corporations
  - ERP systems
  - Intranets (SCM, email, file directories etc)

* Brent Schlender, Fortune Magazine (Oct 1997)
How Successful had Sun been up to 1998?

- 1993 - Sun’s CEO: “The network is the computer.”
  - Sun adapts Solaris OS to coordinate data on networks
- 1994 - Internet explodes in popularity
- 1998 - Sun is the leader in network computing
Java

- 1995 - Sun introduces Java
- Programming Language
  - Portable between computers with different operating systems (did not have to run Windows OS)
  - JVM: Java Virtual Machine
  - Easy to write programs in
  - Easier re-use: “Write once, run everywhere”

- But, programs were slow
- Constant updates on JDK library
  - Programmers had to update to current versions
- Some felt that language was not yet mature
In the meantime, Microsoft...

- Dominated Desktop software (mid 90’s)
  - Users familiar with Windows, Office, etc.

- Windows NT servers
  - Fine for small intranets, “not industrial strength”