

Computer Science E-259

XML with Java

Lecture 7: HTTP 1.1, JavaServer Pages 2.1, and Java Servlet 2.5

14 March 2007

David J. Malan
`malan@post.harvard.edu`

Computer Science E-259

Last Time

- Namespaces in XML 1.1 (Second Edition)
- SVG 1.1
- XSL (XSL-FO) 1.1

Computer Science E-259

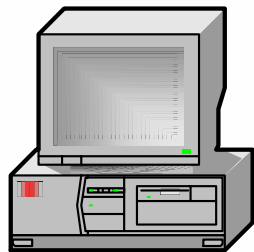
This Time

- HTTP 1.1
- n -Tier Enterprise Applications
- JavaServer Pages 2.1
- Java Servlet 2.5
- Project 3

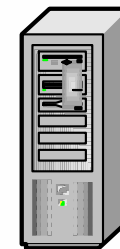
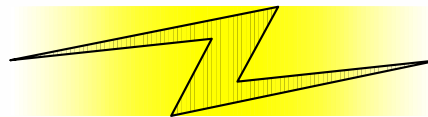
HTTP 1.1

Overview

- Request Methods
 - GET
 - POST
- TCP Ports



Client



Server

HTTP 1.1

Cookies

set-cookie	=	"Set-Cookie:" cookies
cookies	=	1#cookie
cookie	=	NAME "=" VALUE *("; " cookie-av)
NAME	=	attr
VALUE	=	value
cookie-av	=	"Comment" "=" value
		"Domain" "=" value
		"Max-Age" "=" value
		"Path" "=" value
		"Secure"
		"Version" "=" 1*DIGIT%

HTTP 1.1

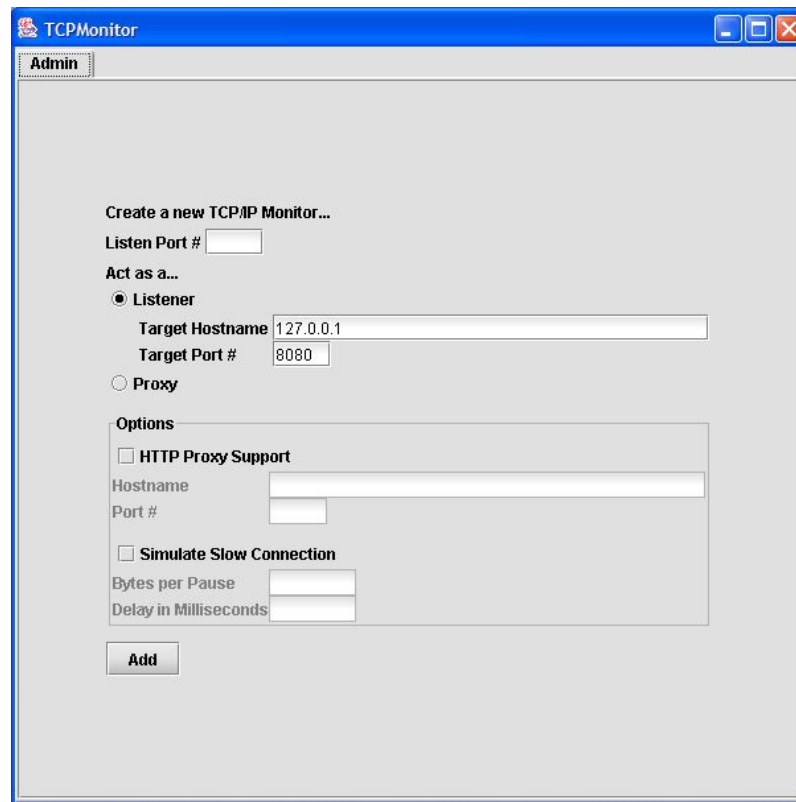
Cookies

```
cookie           = "Cookie:" cookie-version
                  1*((";" | ",") cookie-value)
cookie-value     = NAME "=" VALUE [";" path] [";" domain]
cookie-version   = "$Version" "=" value
NAME             = attr
VALUE            = value
path             = "$Path" "=" value
domain           = "$Domain" "=" value
```

HTTP 1.1

Requests and Responses

`org.apache.axis.utils.tcpmon`



The screenshot shows the 'Admin' tab of the 'TCPMonitor' application. It features a section titled 'Create a new TCP/IP Monitor...' with the following fields and options:

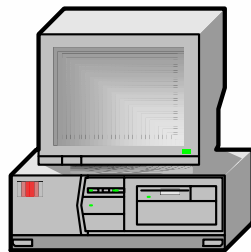
- Listen Port #**: A text input field.
- Act as a...**: A radio button group with two options:
 - ☒ **Listener**: This option is selected. It includes sub-fields for **Target Hostname** (containing '127.0.0.1') and **Target Port #** (containing '8080').
 - ☐ **Proxy**
- Options**: A group box containing:
 - ☐ **HTTP Proxy Support**: Includes sub-fields for **Hostname** and **Port #**.
 - ☐ **Simulate Slow Connection**: Includes sub-fields for **Bytes per Pause** and **Delay in Milliseconds**.
- Add**: A button at the bottom of the configuration area.

Image from <http://ws.apache.org/axis/java/user-guide.html>.

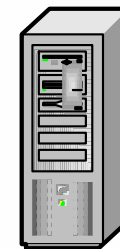
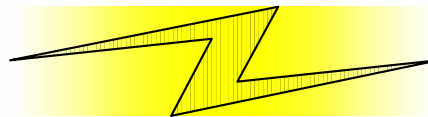
n -Tier Enterprise Applications

$n = 2$

- client tier
 - presents data to user, gathers data from user
- server tier
 - hosts application logic, databases, and data services



Client

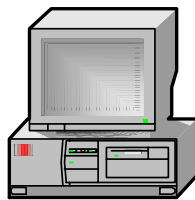


Server

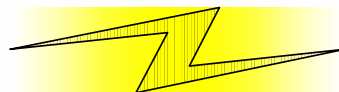
n -Tier Enterprise Applications

$n = 3$

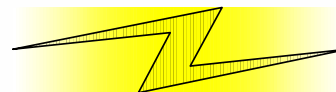
- Client Tier
 - presents data to user, gathers data from user
- Middle Tier
 - hosts application logic
- Back-End Tier
 - hosts databases and data services



Browser



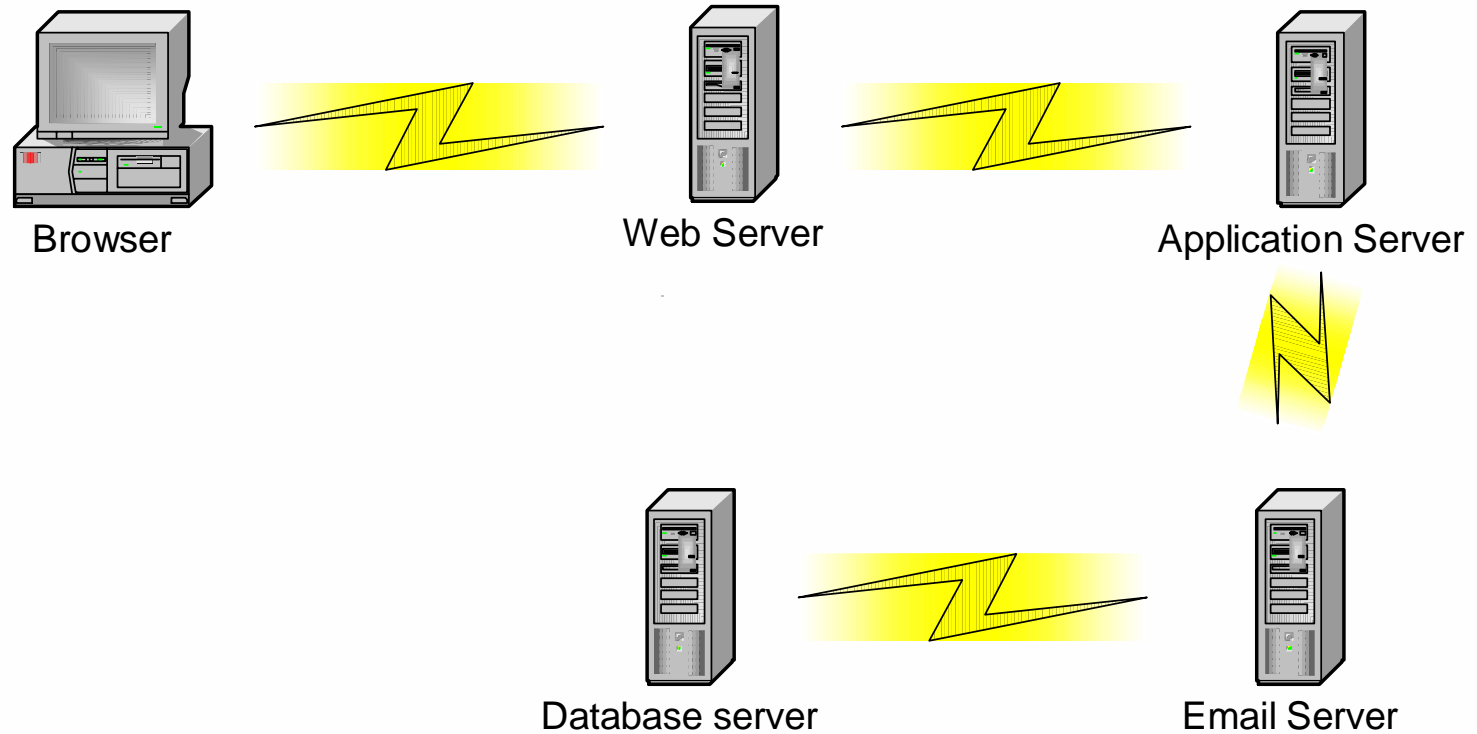
Web Server



Email Server

n -Tier Enterprise Applications

$n = 5$



n-Tier Enterprise Applications

XML

- XML was originally targeted for data publication to the client
- The rationale was that the client software
 - may know best how to display the data for this particular user
 - could benefit from the more semantically meaningful and self-expressive capabilities of XML over HTML
- Very quickly, however, everyone started realizing the potential of XML on the server side as well
- Today, XML can be deployed at all tiers

n-Tier Enterprise Applications

XML on the Client

- If the server communicates to the client self-describing, semantically tagged data in XML, then the client can
 - do more intelligent rendering with the help of stylesheets (*e.g.*, dynamic sorting, searching)
 - present multiple views on the same data
 - take some processing load off the server
 - save many roundtrips to the server by caching data
- Clients can even communicate data to other clients in XML!

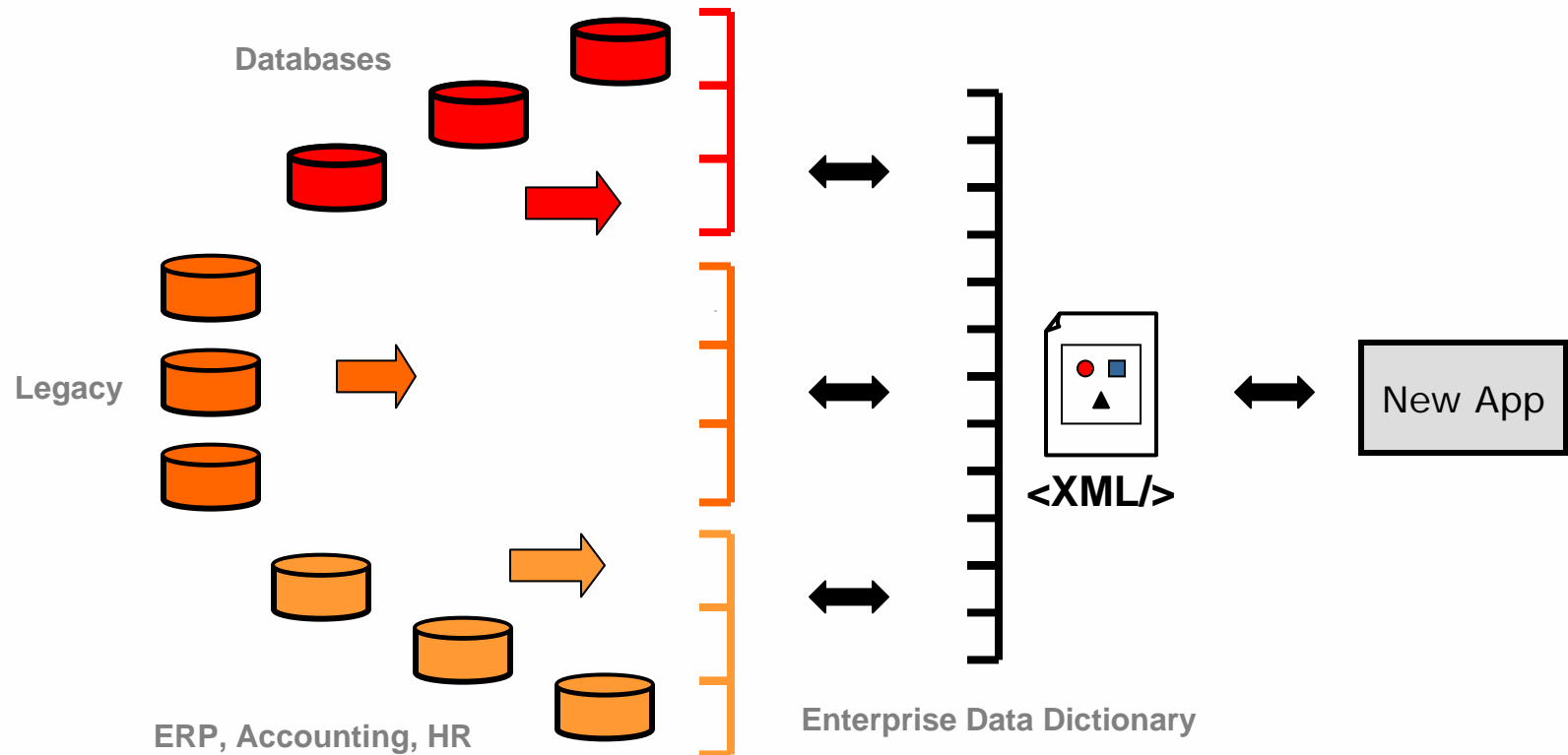
n-Tier Enterprise Applications

XML in the Middle Tier

- Middle-tier applications often intercommunicate by exposing remote objects
 - COM/DCOM objects expose COM interfaces but are bound to Microsoft platforms
 - EJBs expose Java interfaces via RMI but only allow Java clients
 - CORBA uses IDL interfaces and attempts to be platform-neutral
- Content aggregation via XML
 - completely platform neutral
 - easy to use
 - easy to send over the network

n-Tier Enterprise Applications

XML in the Middle Tier



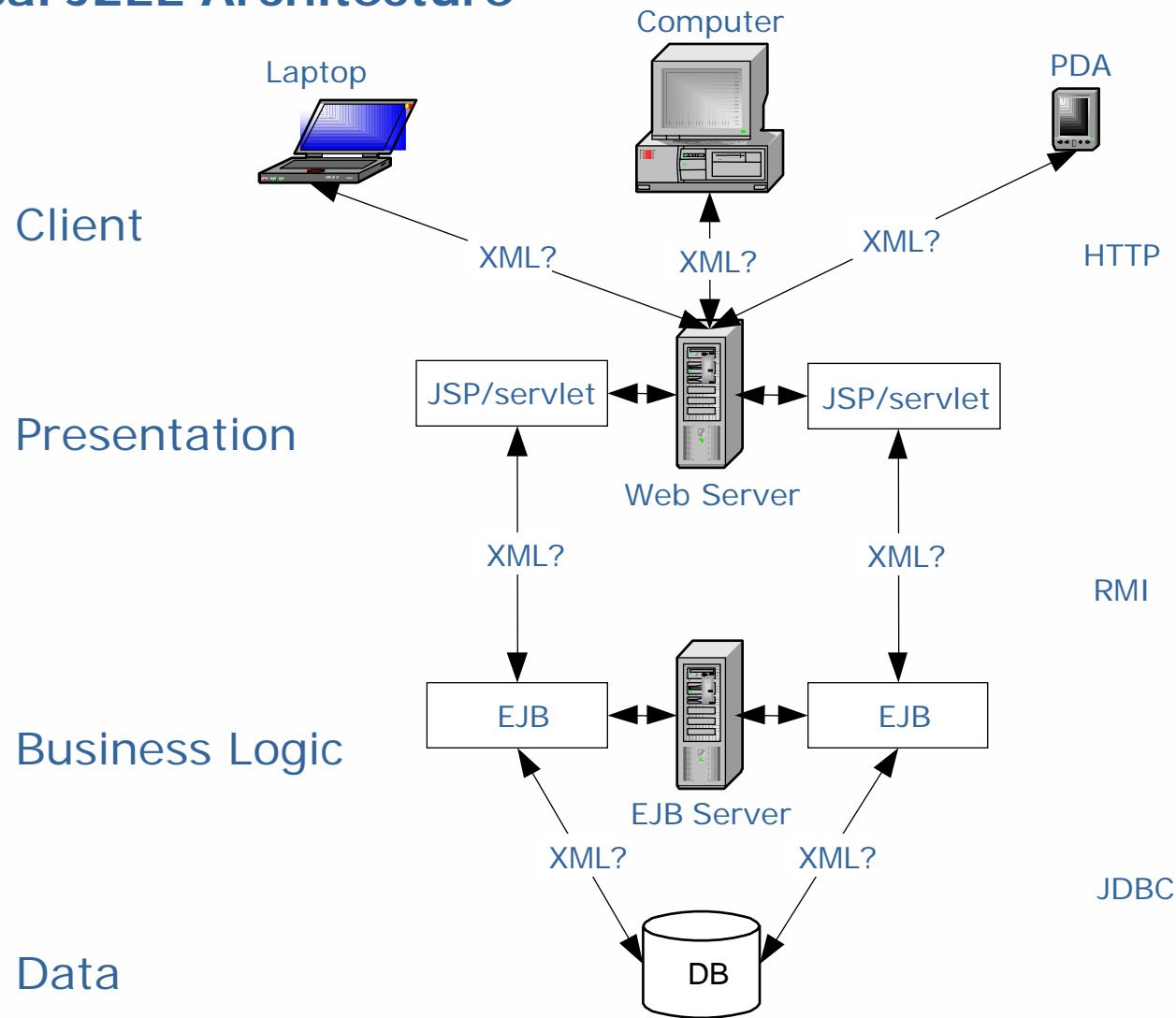
n-Tier Enterprise Applications

XML in the Back-End Tier

- The backend tier usually consists of databases and data services, including
 - relational databases
 - CRM (customer relationship management) applications
 - ERP (enterprise resource planning) applications
 - other legacy applications
- The challenge is to enable middle-tier applications to access information from these data sources

n-Tier Enterprise Applications

Typical J2EE Architecture



JavaServer Pages 2.1

The JavaServer Pages™ Technology

"JavaServer Pages™ is the Java™ 2 Platform, Enterprise Edition (J2EE) technology for building applications for generating dynamic web content, such as HTML, DHTML, XHTML and XML. JSP technology enables the easy authoring of web pages that create dynamic content with maximum power and flexibility."

JavaServer Pages 2.1

What is a JSP Page?

"A JSP page is a textual document that describes how to create a response object from a request object for a given protocol. The processing of the JSP page may involve creating and/or using other objects.... HTTP is the default protocol for requests and responses. Additional request/response protocols may be supported by JSP containers.... The default request and response objects are of type **HttpServletRequest** and **HttpServletResponse** respectively."

JavaServer Pages 2.1

hello, world

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
  <head>
    <title><% out.print("hello, world"); %></title>
  </head>
  <body>
    <% out.print("hello, world"); %>
  </body>
</html>
```

JavaServer Pages 2.1

hello, world

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<jsp:root xmlns="http://java.sun.com/JSP/Page" version="1.2">
  <jsp:directive.page contentType="text/html"/>
  <![CDATA[<?xml version="1.0" encoding="iso-8859-1"?>]]>
  <![CDATA[<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">]]>
  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
    <head>
      <title>
        <jsp:scriptlet>
          out.print("hello, world");
        </jsp:scriptlet>
      </title>
    </head>
    <body>
      <jsp:scriptlet>
        out.print("hello, world");
      </jsp:scriptlet>
    </body>
  </html>
</jsp:root>
```

Java Servlet 2.5

What is a Servlet?

"A servlet is a Java™ technology-based Web component, managed by a container, that generates dynamic content. Like other Java-based components, servlets are platform-independent Java classes that are compiled to platform-neutral byte code that can be loaded dynamically into and run by a Java technology-enabled Web server."

Java Servlet 2.5

What is a Servlet Container?

“The servlet container is a part of a Web server or application server that provides the network services over which requests and responses are sent, decodes MIME-based requests, and formats MIME-based responses. A servlet container also contains and manages servlets through their lifecycle.”

Java Servlet 2.5

Java Servlet Containers

- Apache Tomcat 6.0.10
- BEA WebLogic Server 9.2
- IBM WebSphere Application Server 6.1
- JBoss Application Server 4.2.0 CR1
- Oracle Application Server 10g
- Sun Java System Application Server Platform Edition 9.0 Update 1
- ...

Java Servlet 2.5

Typical Sequence of Events

1. A client (*e.g.*, a Web browser) accesses a Web server and makes an HTTP request.
2. The request is received by the Web server and handed off to the servlet container. The servlet container can be running in the same process as the host web server, in a different process on the same host, or on a different host from the web server for which it processes requests.
3. The servlet container determines which servlet to invoke based on the configuration of its servlets, and calls it with objects representing the request and response.
4. The servlet uses the request object to find out who the remote user is, what HTTP POST parameters may have been sent as part of this request, and other relevant data. The servlet performs whatever logic it was programmed with, and generates data to send back to the client. It sends this data back to the client via the response object.
5. Once the servlet has finished processing the request, the servlet container ensures that the response is properly flushed, and returns control back to the host Web server.

Java Servlet 2.5

hello, world

```
javax.servlet.*  
javax.servlet.http.*
```

Java Servlet 2.5

Sessions



Java Servlet 2.5

Synchronization



Project 3

Content Syndication

Main Entry: syn·di·cate

Pronunciation: 'sin-d&-"kAt

Function: verb

Inflected Form(s): -cat·ed; -cat·ing

Date: 1882

transitive senses

1 : to subject to or manage as a syndicate

2 a : to sell (as a cartoon) to a syndicate or for publication in many newspapers or periodicals at once b : to sell (as a series of television programs) directly to local stations

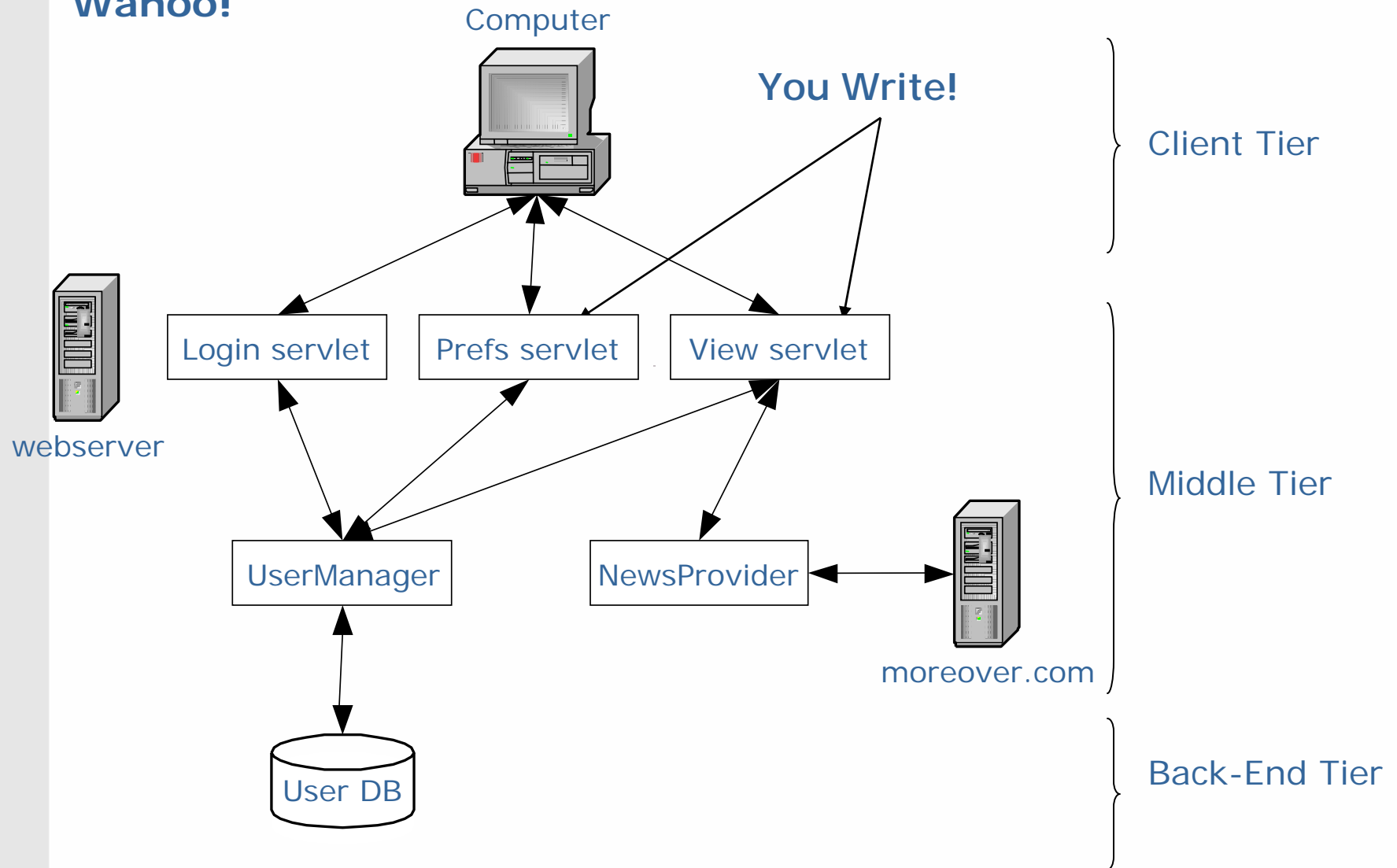
intransitive senses : to unite to form a syndicate

- syn·di·ca·tion /"sin-d&-'kA-sh&n/ noun

- syn·di·ca·tor /'sin-d&-"kA-t&r/ noun

Project 3

Wahoo!



Next Time

XQuery 1.0 and DTD

- XQuery 1.0
- DTD

Computer Science E-259

XML with Java

Lecture 7: HTTP 1.1, JavaServer Pages 2.1, and Java Servlet 2.5

14 March 2007

David J. Malan
`malan@post.harvard.edu`