

Com S 417

Software Testing

Fall 2017 – Week 8, Lecture 14

Announcements

- Lab 3 is available and due Oct. 17.
- We *will* have 5 labs.

Topics

- Understanding Web Apps

Client/Server Systems

- Server-Client
 - A (typically remote) computer functioning as “server” waits for clients to connect at a specified port.
 - Client computers connect to the server computer and avail themselves of the services (remote applications) available at the server computer.
 - Many Examples: telnet, ftp, ssh,svn

Generic Client/Server Interaction

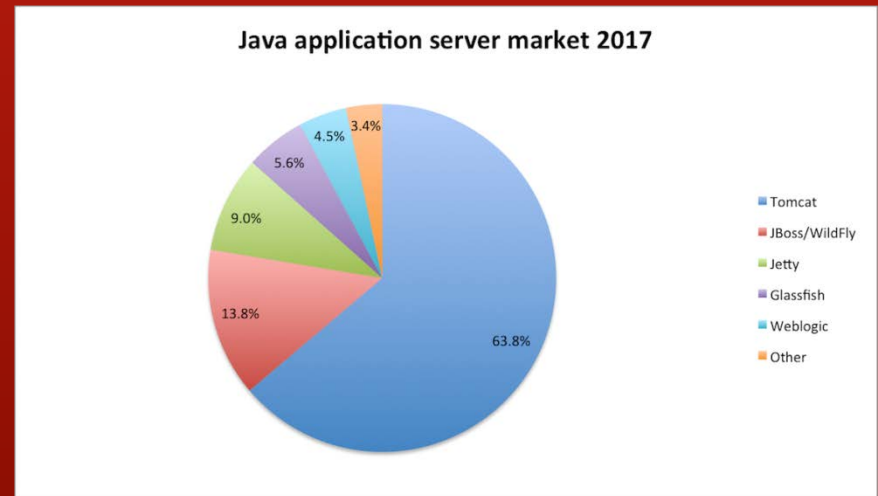
- A **service** (here the page server) is associated with a specific **port**
- A **server** waits for a connection from a client at a specific **port**.
- A **client** tries to connect to server at a **port**.
- After connection is made – a **socket** represents the **established connection** between a client and a server.
- A socket is a tuple
 - <Local-ip: port#, Remote-ip: port#>
- The client and server each have a socket.

Ports

- Ports are a TCP/IP concept.
 - Ports help the receiving O.S. dispatch messages to the appropriate listener.
 - Typical (standard) port numbers:
 - 21—FTP, File Transfer Protocol
 - 22—SSH, Secure Shell
 - 25—SMTP, Simple Mail Transfer Protocol
 - 53—DNS, Domain Name Service
 - 80—HTTP, Hypertext Transfer Protocol
 - 8080—HTTP (used for testing HTTP)

Web as Client/Server System

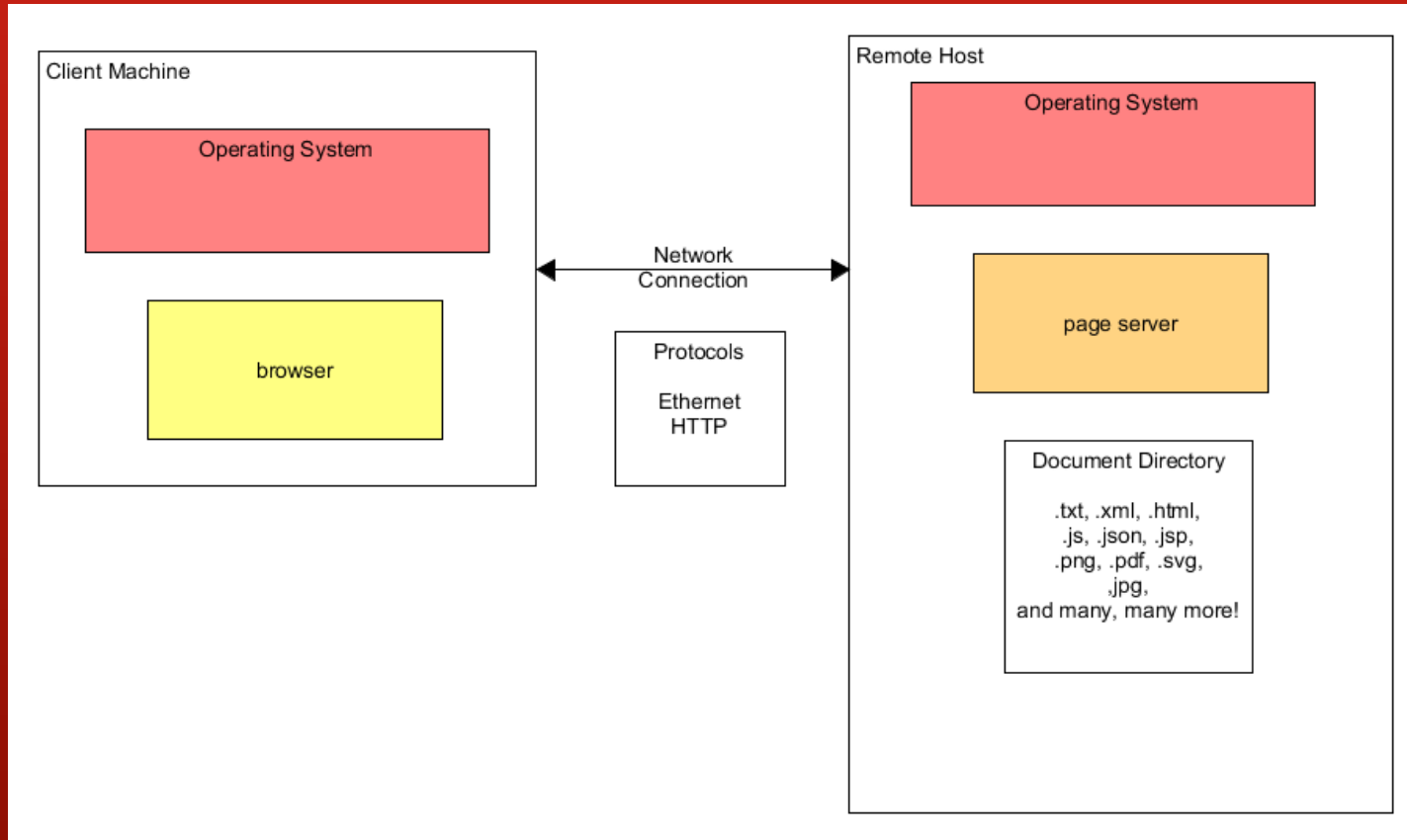
- Web Server
 - An important server application that typically sends html pages to the client and that uses port 80.
 - Port 443 for secure (SSL) connection.
 - Common Java-based web servers:
 - Tomcat, GlassFish, JBoss, Jetty, WebLogic, Websphere
 - Client uses a browser to view the html pages.



Adapted from slides by S Mitra

Pie Chart from: <https://plumbr.eu/blog/java/most-popular-java-application-servers-2017-edition>

A simple static web-site



Client/Server as Distributed Application

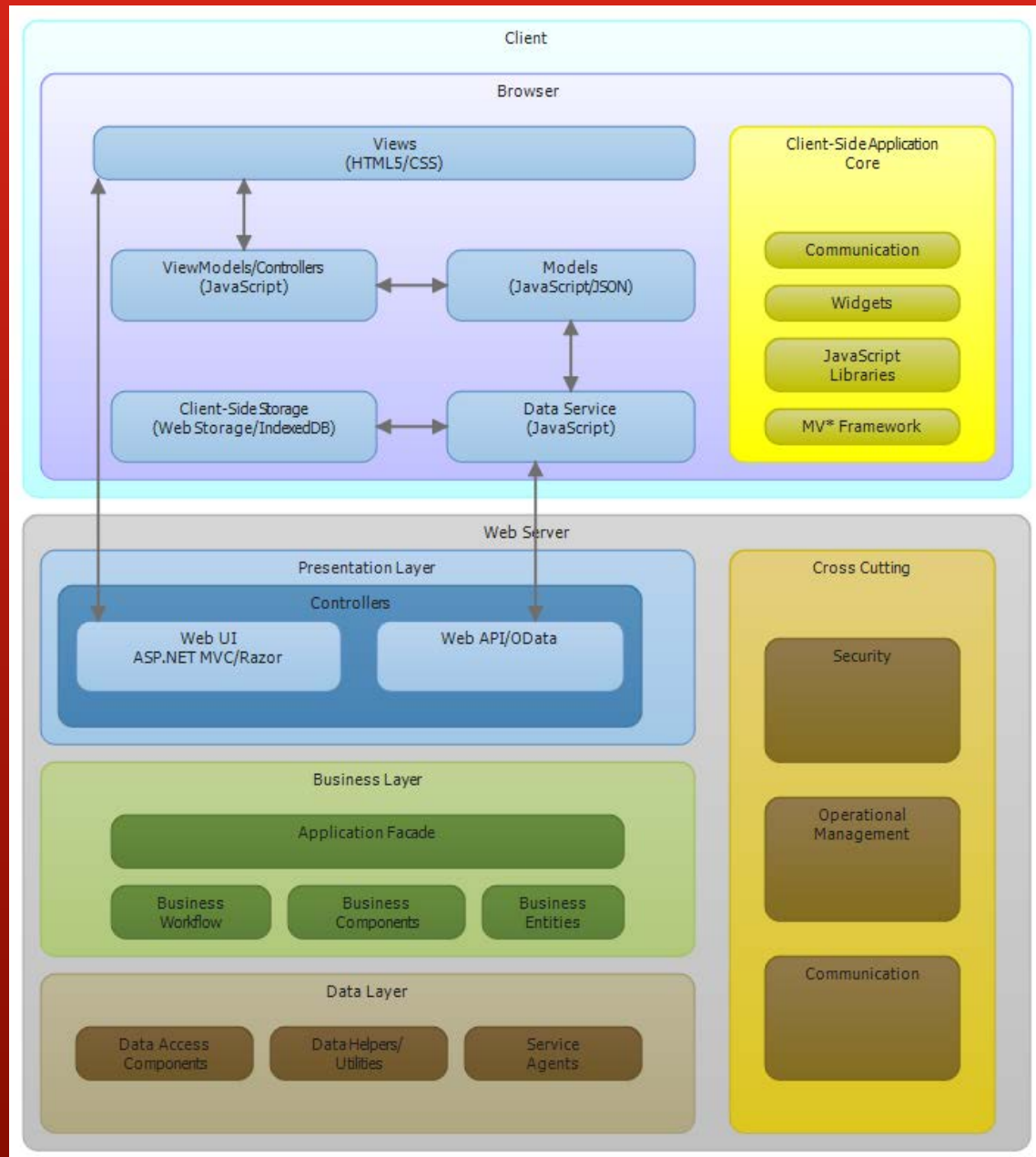
- Both the client and the server can do processing.
 - Examples of processing on Client side are javascripts and applets.
 - Examples of processing on Server side are php, c, java, asp, etc.

Servlets & Server-side frameworks

- The Java “Servlet” Interface is the basis for many, many server side frameworks.
 - A Servlet is a Java object
 - You can consider each servlet as a “request handler.”
 - The servlet interface insulates the Server Java code from lower-level network details (i.e. sockets, ports, etc.)
 - Many patterns and frameworks build on top of the servlet mechanism:
 - JSP, Struts, Spring MVC Framework, Tapestry, JSF (JavaServer Faces), Tiles
 - Some of these are action-oriented, some view-oriented.

A Modern Dynamic Web App Architecture

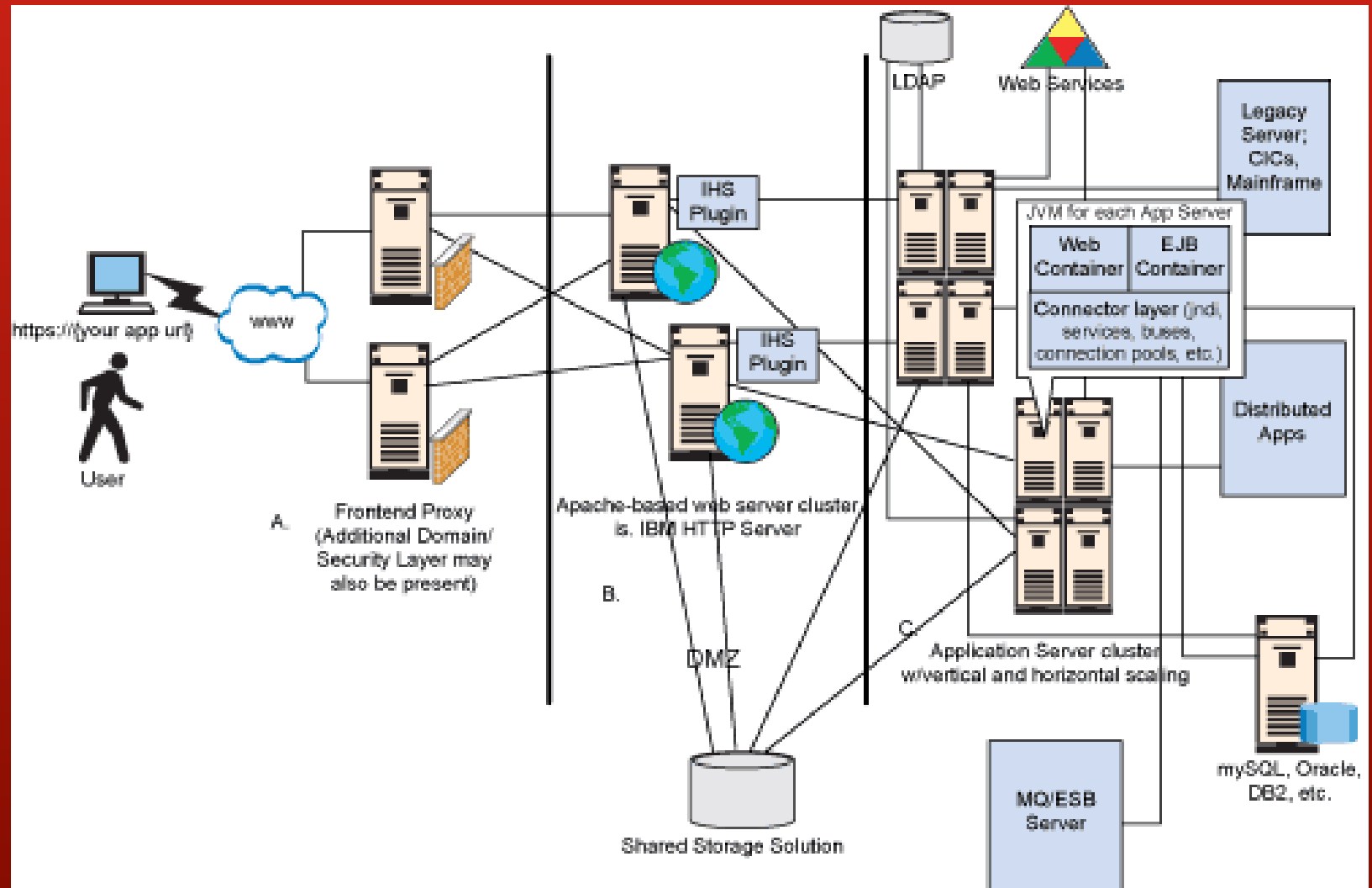
- Controllers act as the “front door” for requests.
- Business layer implements actions.
- Data layer persists state.



Commercial Web-App Complexity

- To address scalability, security, reliability, and performance, enterprise-grade commercial web-apps often require very complex deployments.
 - multiple app servers,
 - sophisticated caching systems,
 - load balancers,
 - proxies to prevent direct access to sensitive resources,
 - high-performance, dedicated data base services
 - fail-over to redundant or alternate sites.

An Enterprise Deployment



Testing Challenges

- Multiple execution environments
- Many, many protocols, subsystems, technologies
- Sometimes complex distributed state
- Modern Ajax apps amount to parallel distributed apps
- Usually not designed for automated test
 - How do you send the browser input?
 - How do you monitor the response?
- Instrumentation may need to run on separate host (e.g., wireshark, mediation system)
- How do you select test sets?

Servlet Engines for our Labs

Tomcat

- **Tomcat** is the Servlet Engine than handles servlet requests for Apache (a generic network server)
 - Tomcat is a “helper application” for Apache
- Apache can handle many types of web services
 - Apache can be installed without Tomcat
 - Tomcat can be installed without Apache
 - It is easier to install Tomcat standalone than as part of Apache
 - By itself, Tomcat can handle web pages, servlets, and JSP
- Apache and Tomcat are open source (and therefore free)

Servlet Engines for our Labs

GlassFish

- **GlassFish** is Sun/Oracle's reference implementation for the Java EE (enterprise) 6 specification.
 - GlassFish is open source.
 - Because GlassFish was created "from scratch" to support the extensions in Java EE, the relationship of GlassFish artifacts to the specification is a little more natural.
- By default, Eclipse does *not* support integrated control of GlassFish.
 - A good thing, because Eclipse prefer's to use the embedded version of Tomcat, which can cause significant confusion for beginners.

Adapted from slides by S Mitra

Background Reading

Network Basics

- TCP/IP
 - Connections
 - Ports
 - Headers
 - Packets
- <http://www.fujitsu.com/downloads/TEL/fnc/pdfservices/TCPIPTutorial.pdf>
 - p.2
 - p.6
 - p. 16

Background Reading

HTTP: the web protocol

- Basics of HTTP messages:
 - https://www.tutorialspoint.com/http/http_messages.htm
 - https://www.tutorialspoint.com/http/http_requests.htm
 - https://www.tutorialspoint.com/http/http_responses.htm
 - https://www.tutorialspoint.com/http/http_methods.htm
 - https://www.tutorialspoint.com/http/http_status_codes.htm
 - https://www.tutorialspoint.com/http/http_url_encoding.htm
- Examples & network level tools
 - https://wiki.wireshark.org/Hyper_Text_Transfer_Protocol
 - <https://tools.ietf.org/html/rfc2616>