CS436/636 Database Application Development, Spring 2021

INSTRUCTOR: BETTY O'NEIL

CLASS MEETS MW 4:00-5:15 ONLINE WITH ZOOM

Plan for course (more details on syllabus)

1. Introduction (today)
2. Java apps using a database (client-server)
   also, mainly in homework, HTML and web request basics
3. Java database-backed web applications using Spring Boot 2

Prereqs: CS430/630 or equivalent DB course, Java including Collection classes such as HashMap and TreeSet

HW1: get started as soon as possible! Get a class account!

Password for protected files…
Call roll…
Reading: Murach, Chap. 1. You could also start reading Chap. 11-12 on databases (specifically MySql), starting at pg. 364.

Introduction: Internet vs. World Wide Web

Internet is an interconnected ***network*** of thousands of networks and millions of computers (sometimes called host computers or just hosts).

The **World Wide Web**, or **Web** for short, is one of the Internet’s most popular ***services***, providing access to over one billion Web pages.

The Web runs over the Internet, using its ability to transfer data reliably (using TCP).

How the Internet evolved

The Internet, then known as the ARPANET (Advanced Research Projects Agency Network), began in 1969, soon file transfers, emails were flowing.

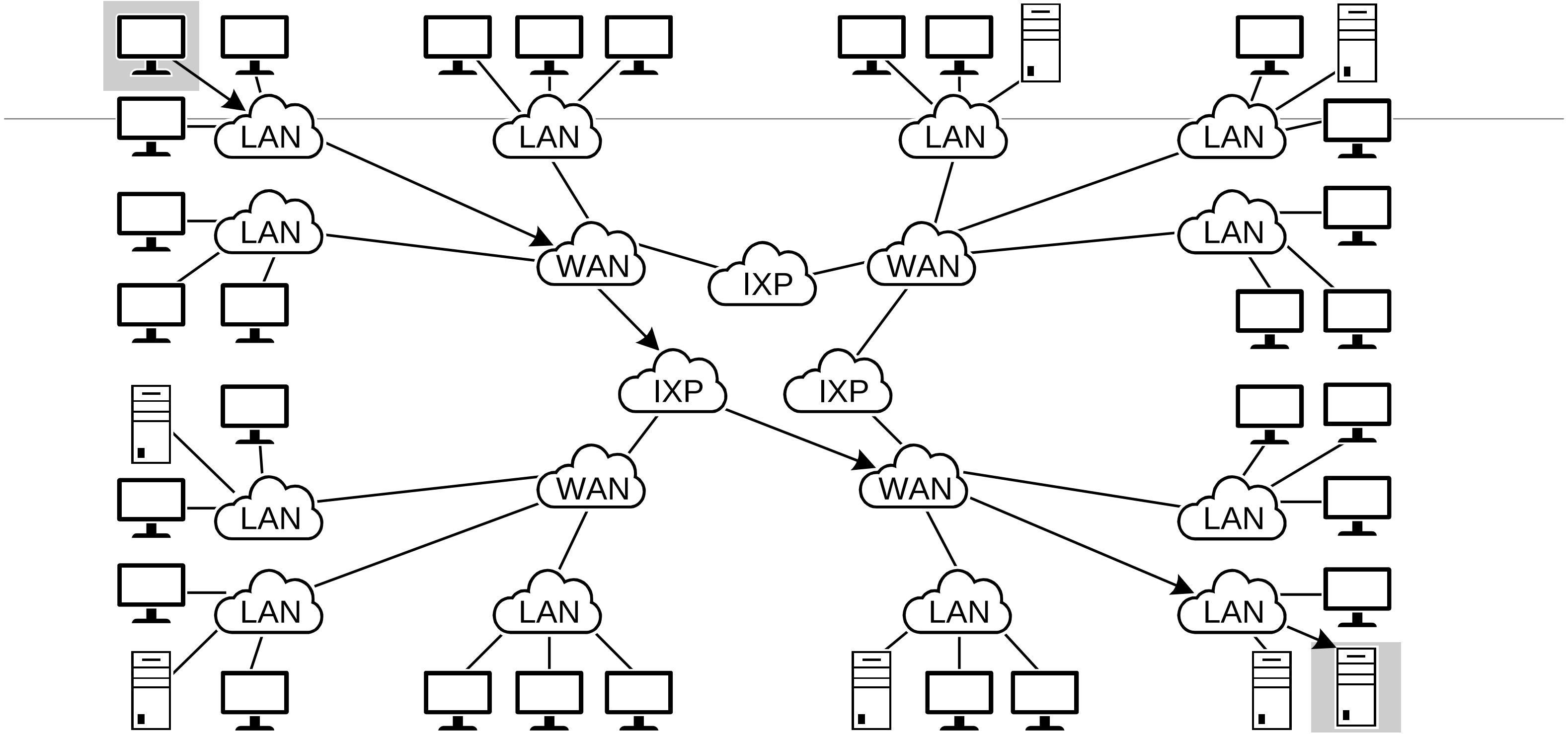
The current TCP/IP protocols date from 1983. The Internet was already international then, but restricted to non-commercial use.

In 1990, it was opened to commercial use….

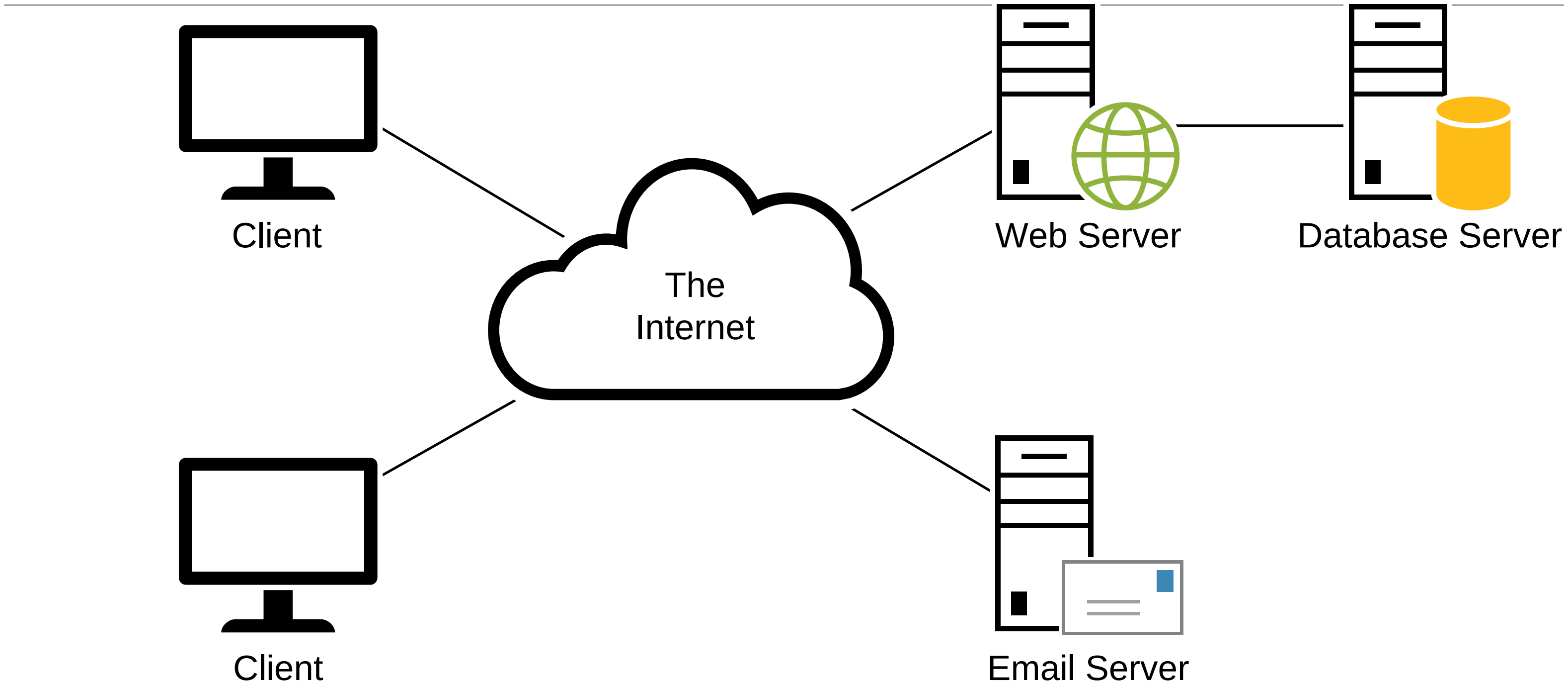
A Hierarchy of Networks

- Every computer that is connected to the Internet is part of a network.
- You may use a cable or DSL modem to an **Internet Service Provider (ISP)**. At work, you may be part of a **local area network (LAN)** using an ISP that your company has contracted with.
- When you connect to your ISP, you become part of their network. The ISP may then connect to a larger network and become part of their network. The Internet is simply a *network of networks*.

The architecture of the Internet



Luckily, the Internet can be viewed like this for our purposes…



Internet Domain Naming System

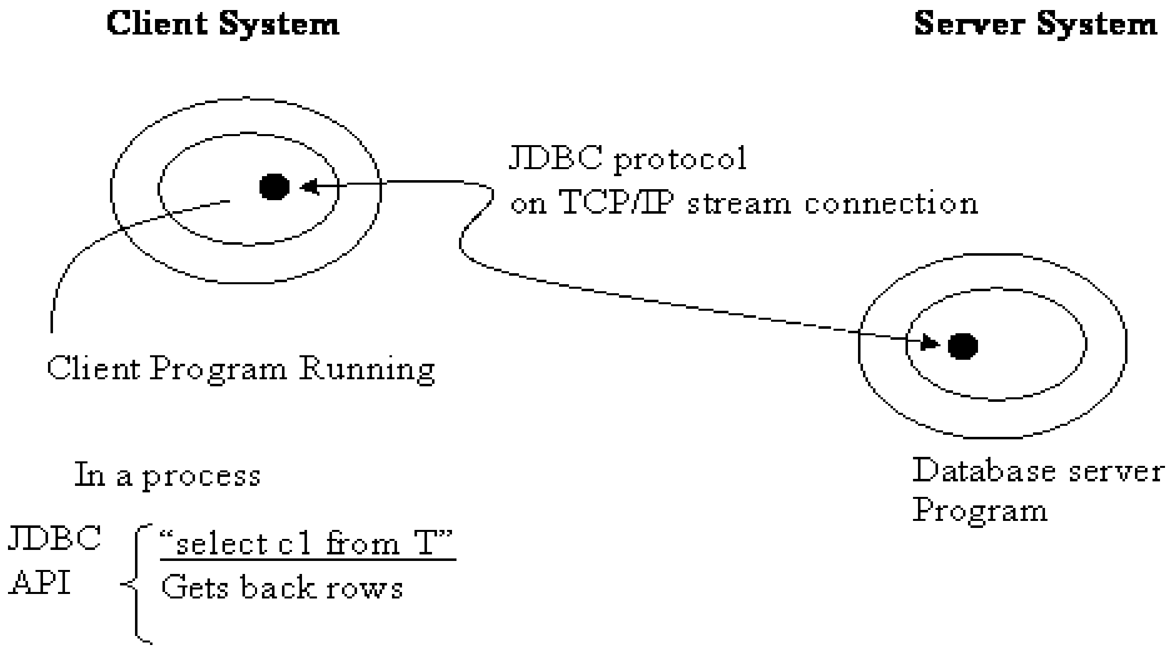
- Provides unique ids for all the hosts directly connected to the Internet
  - Example hostname = www.cs.umb.edu, a unique hostname across the Internet.

- Domain naming system (in use since 1983):
  - Top-level domain name: edu
  - UMB's domain name: umb.edu
  - Departmental subdomain: cs.umb.edu
  - Full hostname = www.cs.umb.edu, specifying the departmental web server host

- The Internet's Domain Naming Service (DNS) servers translate a human-readable domain name into the machine-readable IP address.
  - The IP address of www.cs.umb.edu is 158.121.106.222, also unique across the Internet

Client-server use of the Internet:

our first project will use this setup



Client System

Server System

JDBC protocol
on TCP/IP stream connection

Client Program Running

Database server
Program

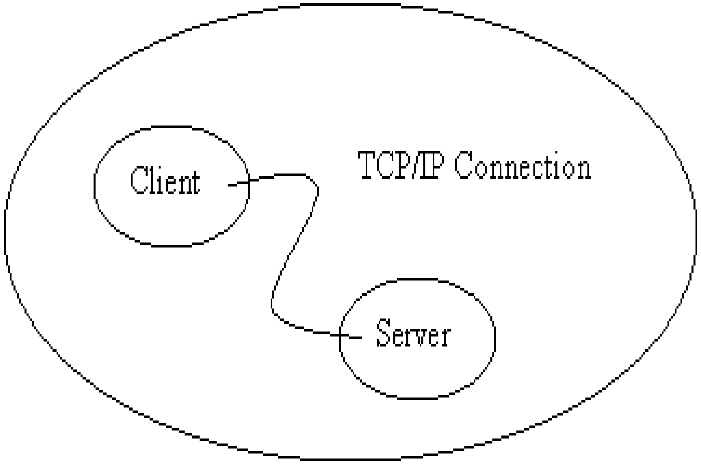
In a process

JDBC
API

"select c1 from T"

Gets back rows

Client-server case, all on one system



In this case, the Internet is not involved in client-server communication, but it’s still done with a TCP connection.

Client-server UI

- We’ll settle for a crummy user experience here, just line-oriented user input, knowing we’ll do better later with a web app.

- Of course, with more effort we could make a decent GUI for it.

The World Wide Web

- The Web runs over the Internet, using its ability to transfer data reliably (using TCP) and specify hosts with domain names and IP addresses.

- The crucial web protocol is HTTP (Hypertext Transfer Protocol)

- The Web dates from the early 90s.

- Since the Internet was already an international presence, so was the Web.

Web Accessibility

- Web accessibility means that people with disabilities can use the Web.
- HTML, especially HTML5, plays an important role by providing a standard way of describing what’s on the screen.
- Our web apps will send HTML5 to our users, and so provide base-level accessibility.

Uniform Resource Locator (URL)

URLs specify locations on the Internet, along with the protocol needed for accessing something there.

We say URLs specify “resources”.

A simple URL has the following format:

<protocol> :// <hostname or IP address> / <path>

Example: https://www.cs.umb.edu/cs636 (our class homepage)

Here hostname = www.cs.umb.edu

Using IP address: https://158.121.106.224/cs636

What is a web application?

App vs. web app:

An app (desktop or mobile) is a program that runs directly on the OS of the device.

- Thus has sub-species: Android app, ioS app, Linux app, Windows app, MacOS app, etc., and each works only on its own OS.
- An organization that wants all its users to use “its app” needs to implement all these versions.

A web app runs inside a web browser. The web browser itself is an app on the desktop or mobile device.

- Thus we would expect sub-species Chrome app, Safari app, Firefox app
- But browsers are much more standardized than OSs, so we just say “web app” and expect it to run on any of these modern browsers.
- An organization that wants all its users to use “its webapp” should only need to implement one version.

A web application has a website

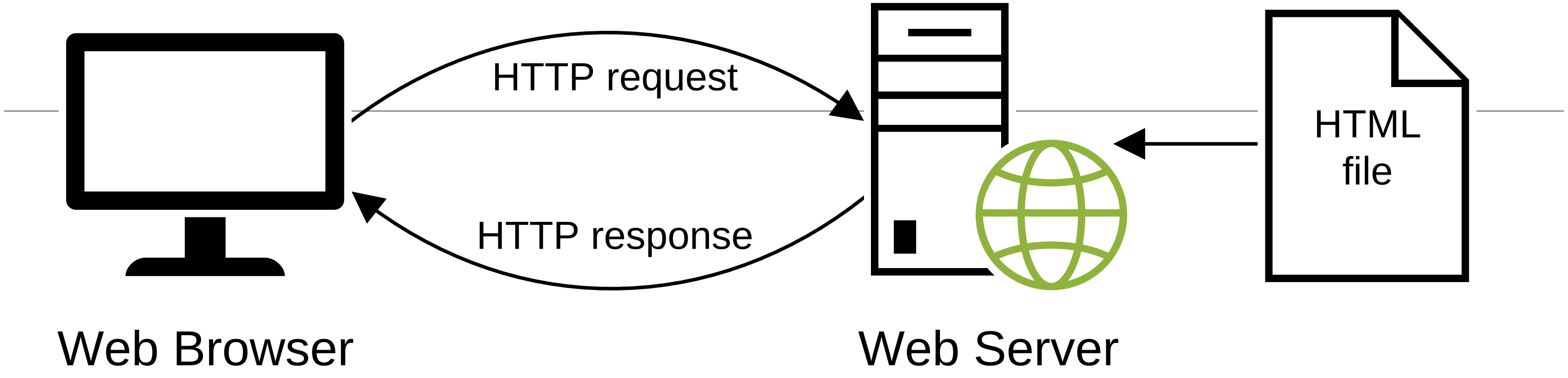
A web app pulls its pages, etc., from a website on a server connected to the Internet (or possibly a smaller network).

A web app can accept user input and send it back to its server for processing (the Java or PHP way), or process it right in the browser (the Javascript way).

Either way, a web app depends on its server for saving data persistently.

A web app UI is much like an app UI: buttons, forms, text input, etc., but may be slowed down by the network.

How static web pages are processed



- A *static* web page is a simple HTML file, no Java or PHP or other server-side processing involved.
- Its HTML specifies a whole page on the browser, say in one tab of Chrome or Firefox or …
- HTTP is the network protocol that delivers HTML and many other data formats to the client web browser.
- Note the web browser is very likely to be in a mobile device today, so this picture is a little dated.

A simple HTTP request

```
GET / HTTP/1.1
Host: www.example.com
```

A simple HTTP response

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 136
Server: Apache/2.2.3

<html>
<head>
    <title>Example Web Page</title>
</head>
<body>
    <p>This is a sample web page</p>
</body>
</html>
```

A simple HTTP request

```
GET / HTTP/1.1
Host: www.example.com
```

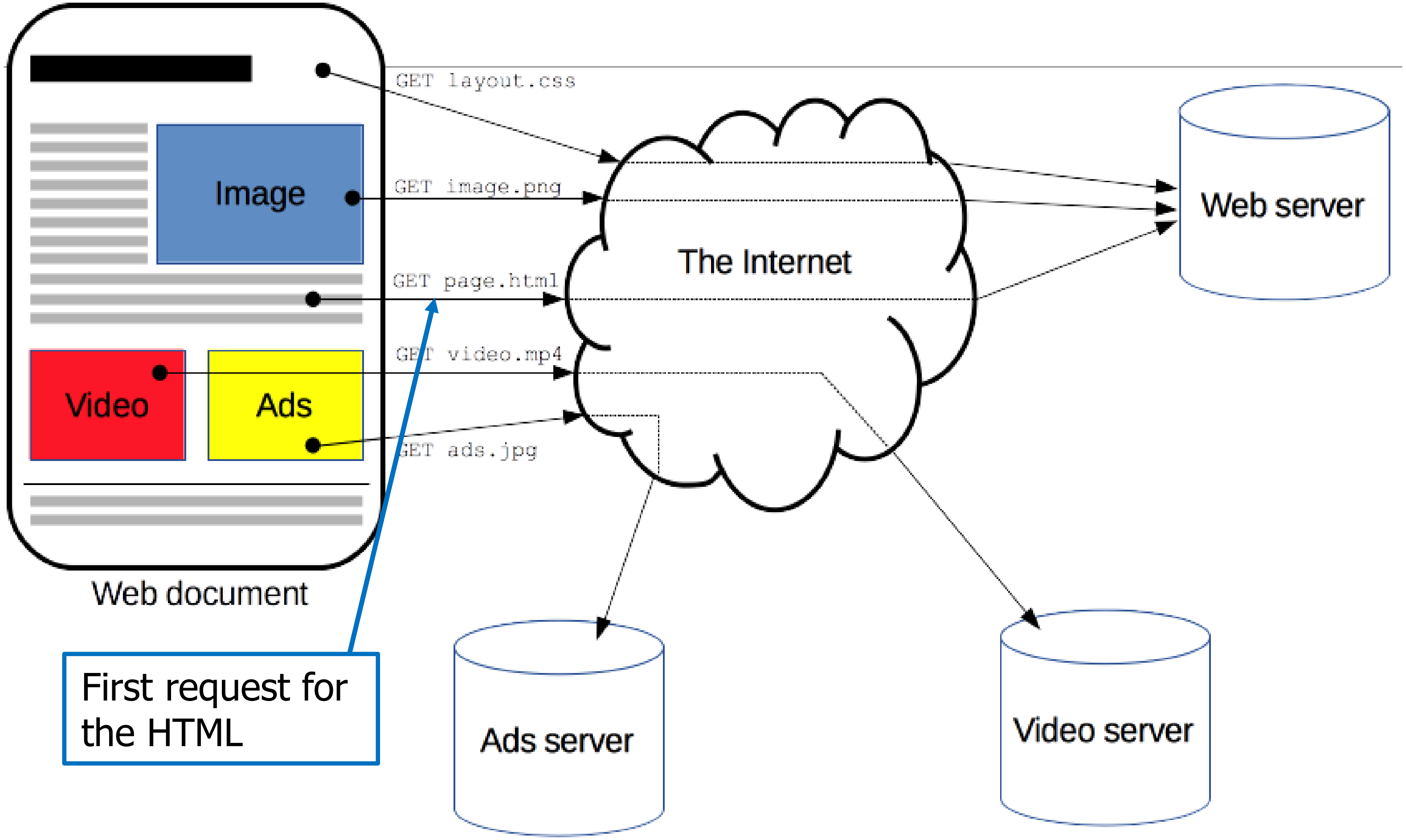
A simple HTTP response

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 136
Server: Apache/2.2.3

<html>
<head>
    <title>Example Web Page</title>
</head>
<body>
    <p>This is a sample web page</p>
</body>
</html>
```

- If this HTML text was just read from a file on the server, this is *static HTML.*
- If a program helped generate it, it is *dynamic HTML*.

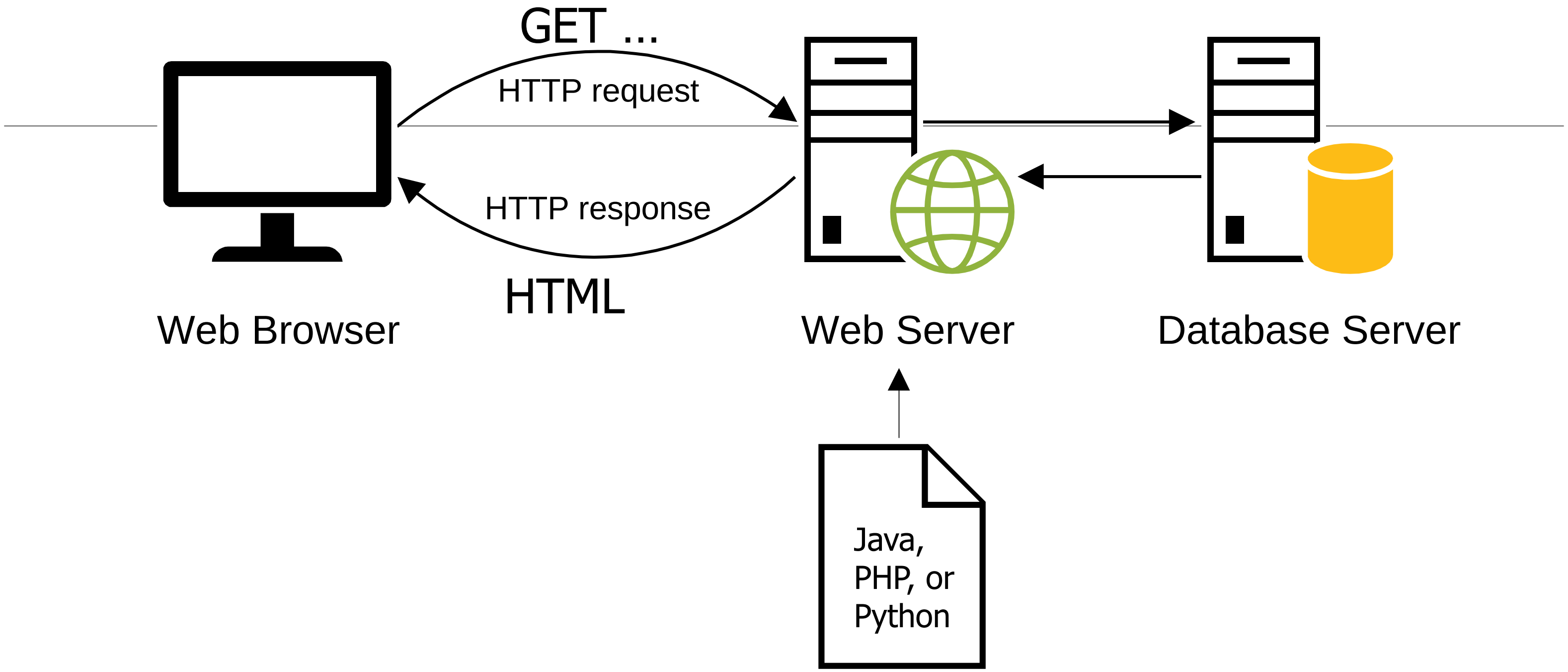
One static page can involve many HTTP requests



First request for the HTML

From https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

How dynamic web pages are processed with Java or PHP or Python



- The server-side code is not seen by the client, only the *server-generated* HTML.
- Any user click/tap causes another HTTP request to the server for processing.

Clients: Desktops and Mobile Devices

Similarities

- Both have rectangular hi-res color screens
- Both can accept keyboard input
- Both can accept click/tap selection of buttons, etc.
- Both can accept drag and drag-and-drop user gestures

➢ Based on these similarities, we want to be able to write webapps that run on both using the same codebase.

Clients: Desktops and Mobile Devices

Differences

- Not all desktop screens are touch screens
- Can't "hover" with touch screens
- Can't "pinch" or "spread" with non-touch screens
- Other less common touch-screen gestures: long press, flick

- ➢ Using just Java/PHP/Python on the server, we can't detect any of these, or drag or double-click either, so we will concentrate on click/tap, and text input
- ➢ Using Javascript, we could detect the others…

Java Advantages

- General-purpose language, not just web apps
- A compiled language, with strong typing, good for avoiding bugs
- Has open-source version
- Has JDBC, providing access to all databases with one API.
- Provides excellent performance compared to PHP, Python, almost as good as C/C++
- Available on many platforms including Android mobile devices

Java disadvantages

- Considered hard to learn, needlessly complicated
- Wordy, needing lots of curly braces
- Shared JVM in web server can cause unwanted data leakage between requests unless the programmer follows rules.

Java Uses related to the web

- Server code for medium to very large sites
- Server-side code for Javascript running in web pages
- Server-side code for providing data for programs running in mobile devices
- Android apps.

Java vs. Javascript

Javascript (JS for short) has become very important for web apps, especially ones that need to work nicely on smartphones.

JS runs in the browser, so is closer to the user than Java or PHP or Python, which run only on the server (other than separate use of Java in Android apps).

JS can detect the advanced user gestures like pinch/spread.

JS now can run on the server too (with nodejs), providing a single language for both sides.

JS is a completely different language from Java or PHP or Python, themselves in the same family. It does use Java-like syntax for conditionals, loops, etc.

Javascript and Java

Websites often use a combination of JS and Java (or JS and PHP or Python) along with HTML and CSS.

- The actions that need fast response to user requests, like game playing or device control, are in JS
- Ancillary support, like UI for changing a password, or finding documentation, are in server-side code.
- Server-side support, like accessing a database, can be in Java/PHP/Python or server-side JS.

The user often doesn't know what language is in use.

- All these languages can do buttons, forms, links, etc.
- But if there's a bug running around the screen, that's JS.
  - Or the restaurant menu folds up for you (this also uses advanced CSS).
- On the other hand, you don't need JS for a photo carousel, so motion by itself doesn't necessitate JS.

Three approaches for developing Java web apps:

(Text pages 12-13)

Servlet/JSP: we'll cover

- Is a lower-level API that does less work for the programmer.
- Provides a high degree of control over the HTML/CSS/JavaScript that's returned to the browser.

JSF: on the decline, so not covered

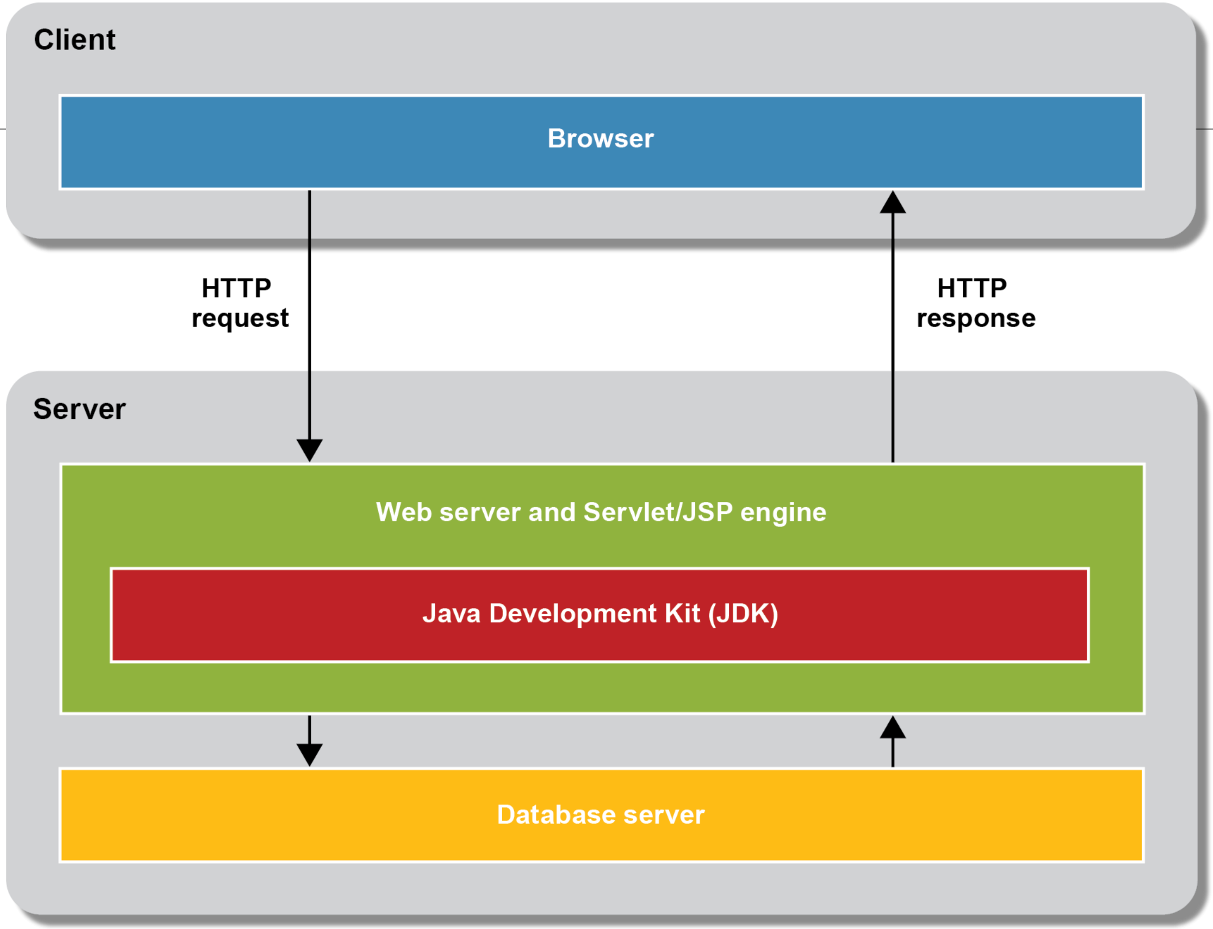
Spring Framework: we'll cover

- Is a higher-level API that does more work for the programmer.
- Provides a high degree of control over the HTML/CSS/JavaScript that's returned to the browser.
- Can be used with JSP for defining dynamic web pages, as we will do.

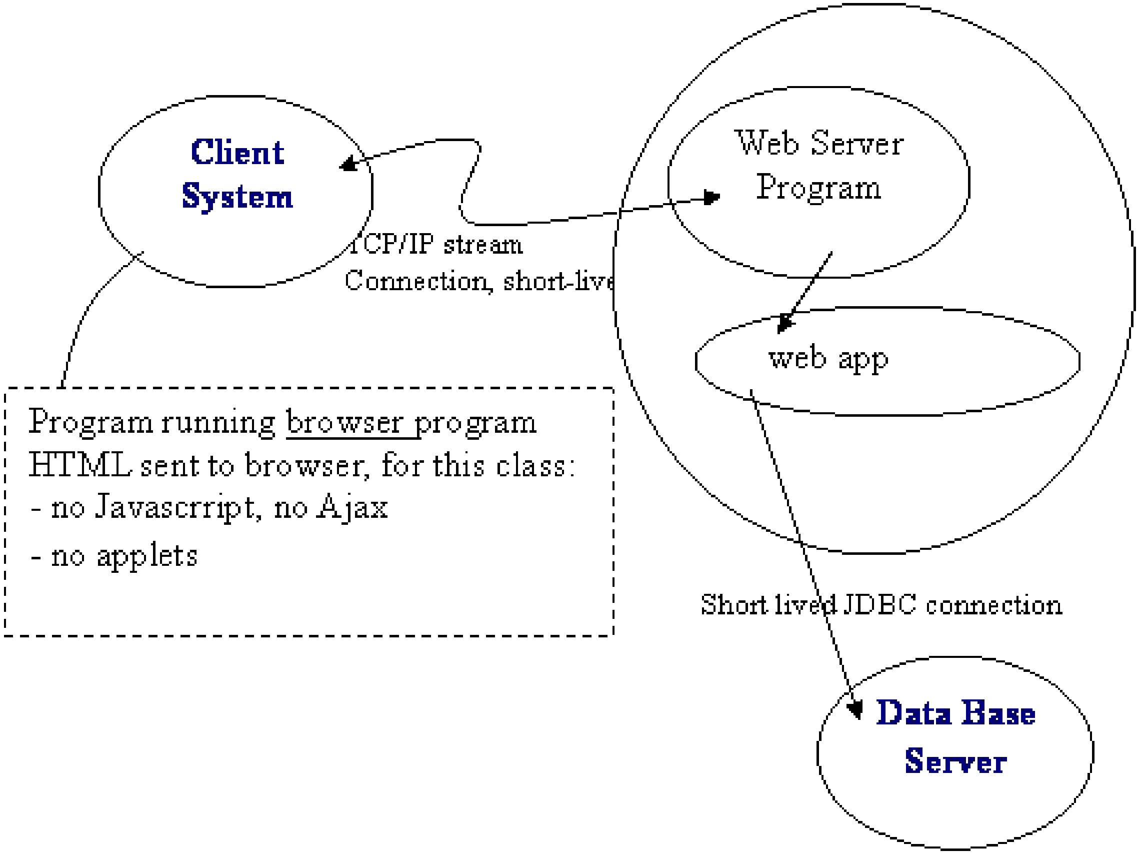
Terminology

- The *Java Standard Edition* (*Java SE*) includes the *Java Development Kit* (*JDK*) and the *Java Runtime Environment* (*JRE*).
- The *Java Enterprise Edition* (*Java EE*) specification describes how web servers can interact with all Java web technologies.
- *Servlets* store the Java code that does the server-side processing.
- *JavaServer Pages* (*JSPs*) specify the HTML that defines the user interface.
  - *Thymeleaf* is an alternative way to specify the HTML
- *Java Persistence API* (*JPA*) is an API for working with databases.

The components of a servlet/JSP application



Systems view in web case



Client System

TCP/IP stream

Connection, short-live

Web Server Program

web app

Short lived JDBC connection

Data Base Server

Program running browser program

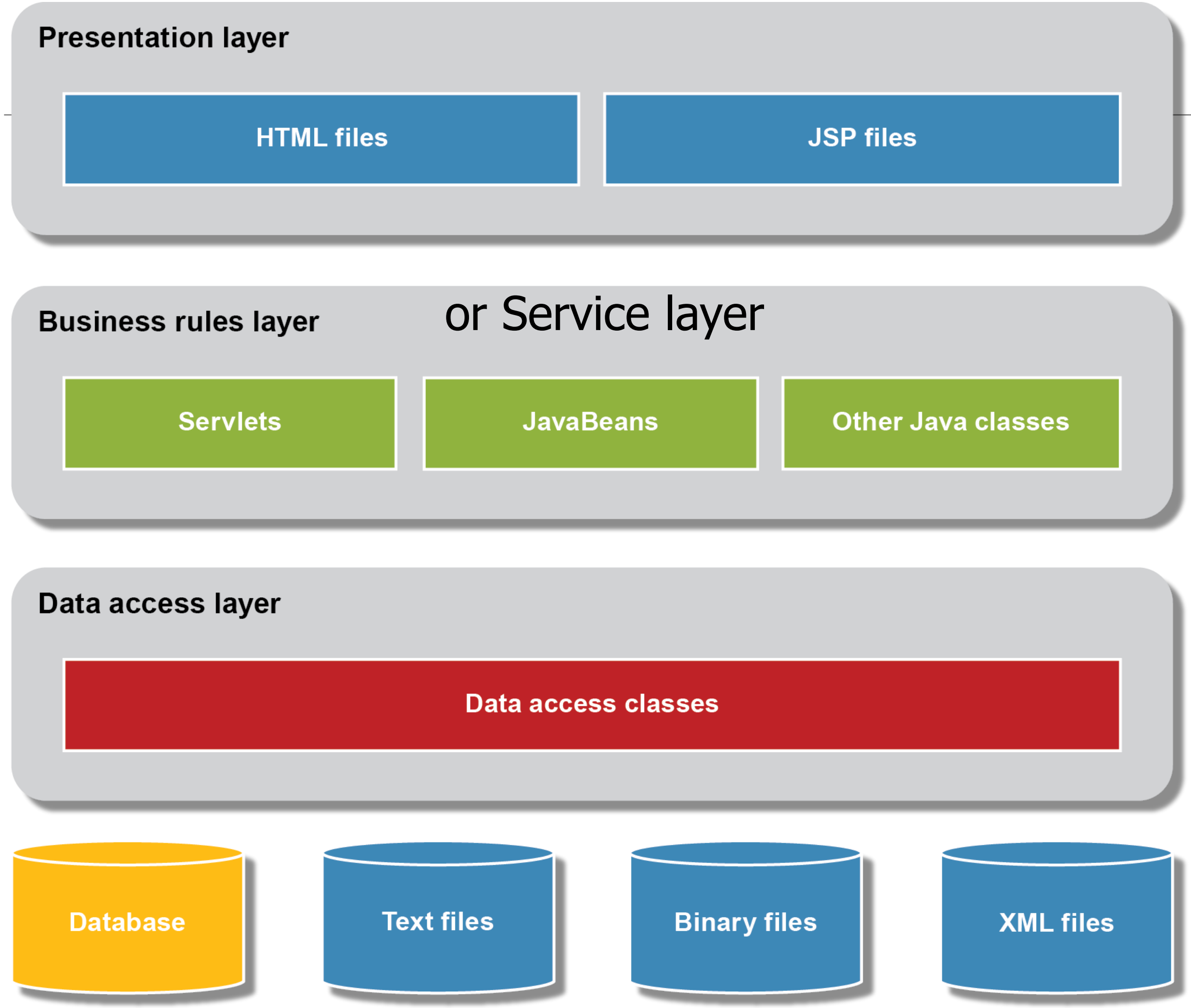
HTML sent to browser, for this class:

- no Javascrript, no Ajax

- no applets

The Three Layers in the server

--and source for the icon image for our website:



or Service layer

A “full stack developer” can develop for any of these layers.

Layers for client-server too

We'll use same three layers for the client-server case too. We'll be able to move the lower two layers over to the web app case with very few mods.

Presentation: UI code using line-oriented Java
Service/Business Rules: top level actions of the app, implemented by calling the data access layer as appropriate
Data Access: all the SQL code is here, to run the database access and provide convenient methods to the service layer.

Two popular web servers with servlet support

Tomcat: we'll use

- Is a servlet/JSP engine that includes a web server.
- Is free, open-source, and runs on all modern operating systems.
- Is a popular web server for Java web applications.

GlassFish

- Is a complete Java EE application server.
- Is free, open-source, and runs on all modern operating systems.
- Provides more features than Tomcat.
- Requires more system resources than Tomcat.

Database servers

- I'll create a mysql database for you on pe07.cs.umb.edu and an Oracle account, after you obtain a class account.

MySQL: we’ll use

- Is a relational database server that implements most of the SQL standard.
- Is free, open-source, and runs on all modern operating systems.
- Is the most popular database server for Java web applications.

Other popular database servers

- PostgreSQL
- Oracle: the premier enterprise DB. --we’ll use it
- H2 embedded database: we’ll use, esp. for testing

Development Tools

We will be working in two environments, our personal systems and the server pe07.cs.umb.edu See DevelopmentSetup.

- We'll use Java and Maven in both environments: they are very portable
  - We'll cover Maven from basics.

- We'll use the eclipse IDE on our own systems to do most of the coding. If you are expert in and prefer another IDE, that's OK too.