CPS530

Web Systems Development

Lesson #1

Introduction to Web Systems Development

Origins of the Internet

- **Arpanet:** late 1960s and early 1970s: U.S. Department of Defense
- **Bitnet:** late 70s and early 80s: email, ftp, telnet, gopher.
- **NSFNet:** 1990. Arpanet now becomes civilian and later renamed **Internet**.

What is the Internet?

- A world-wide network of computer networks.

- All connections use TCP/IP. TCP/IP hides the differences among devices connected to the Internet .

- Every node has a unique numeric address: A 32-bit binary number (IPv6, has 128 bits). Organizations are assigned groups of IPs for their computers.

World-Wide-Web

- Fully qualified domain name ex: *www.ryerson.ca*.
- DNS servers convert fully qualified domain names to IP addresses.
- Tim Berners-Lee at CERN proposed the Web in 1989.
- The Web uses one of the protocols, **http**, that runs on the Internet. There are several others.

Web Browsers

- Mosaic is the first to use a GUI. It led to the explosion of Web use.
- Browsers are clients. They always initiate communication.
- Servers react (sometimes servers require responses).
- Most requests are for existing documents, using *HyperText Transfer Protocol* (HTTP), but some requests are for program execution with the output being returned as a document.

Web Browser Engines

- There are major families of web browsers based on layout or rendering engines.

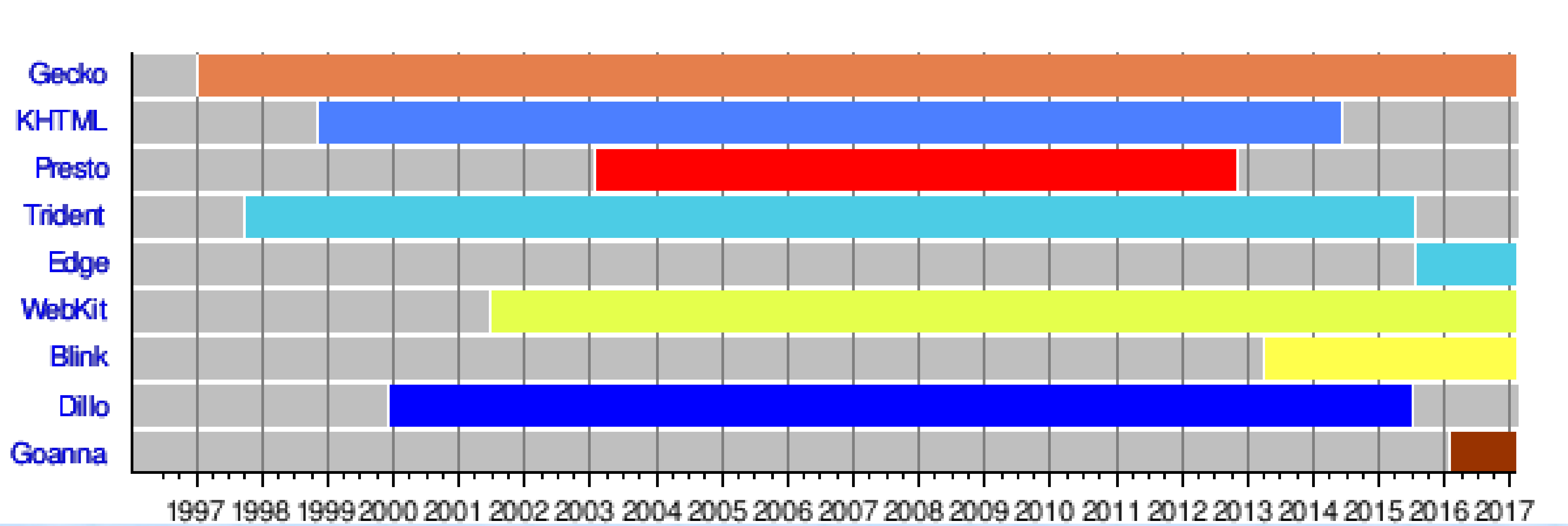
- **Blink** powers the Chrome and Opera browsers as well as Kindle and Android devices.

- **WebKit** powers the Apple Safari browser and its variants. It is also the layout engine that Blackberry devices.

- **Gecko** powers the Firefox and SeaMonkey browsers (Mozilla family).

- **Trident** powers Internet Explorer, Real Player and Winamp. The new Edge browser is powered by a fork engine named **EdgeHTML**.

Web Browser Engines



Web Servers

- A web server runs in the background of the operating system. It monitors a communications port on the host, accepting HTTP messages when they appear.
- All current Web servers have a common ancestry from either CERN or NCSA. The most common server configuration is Apache running on Unix.
- Web servers have two main directories: *Document root* (servable documents) and *Server root* (server system software).
- Document root is accessed indirectly by clients. Its actual location is set by the server configuration file.

Web Servers

- Virtual document trees permits some documents to be stored outside the documents root.
- Many servers can support more than one site on a computer. These are called *virtual hosts*.
- A *proxy server* is used to serve documents located on other machines on the web.

URL Paths

- protocol + domain + port + doc.
- Ex: http://www.ryerson.ca:80/index.html.
- URLs cannot include spaces or any of a collection of other special characters (semicolons, colons, ...)
- The doc path may be abbreviated as a partial path. The rest is furnished by the server configuration (you may omit index.html for example). The port is optional (if default 80).
- If the doc path ends with a slash, it is a directory.

MIME Types

- Multipurpose Internet Mail Extensions (MIME).

- Originally developed for email. Used to specify to the browser the form of a file returned by the server (attached by the server to the beginning of the document).

- Type specifications: type/subtype

- Examples: `text/plain, text/html, image/gif, image/jpeg`

- Server gets type from the requested file name's
- suffix (.html implies text/html)

Web Programming Overview

- Web programming can be either server-side or client-side. Static or dynamic.
- Client-side: HTML, JavaScript, jQuery, VBScript, XML, RSS.
- Server-side: Perl, PHP, ASP, Java Servlets, JSP, TCL, Python, Ruby.

Client-side (HTML)

- HTML, (HyperText Markup Language), is the predominant markup language for Web pages. It provides a means to describe the structure of text-based information in a document.

- HTML is written in the form of tags, surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code (such as JavaScript) which can affect the behavior of Web browsers and other HTML processors.

Client-side (JavaScript)

- JavaScript is a scripting language most often used for client-side web development.
- It is a dynamic, weakly typed, prototype-based language with first-class functions.
- JavaScript was influenced by many languages and was designed to look like Java, but be easier for non-programmers to work with.
- Jquery and Ajax provide new methods of using JavaScript, server side languages (eg ASP.Net or PHP) and other languages to improve the user experience.

Client-side (VBScript)

- VBScript (short for Visual Basic Scripting Edition) is an Active Scripting language developed by Microsoft.
- When employed in Microsoft Internet Explorer, VBScript is similar in function to JavaScript, as a language to write functions that are embedded in or included from HTML pages and interact with the Document Object Model (DOM) of the page, to perform tasks not possible in HTML alone. Other web browsers such as Firefox, and Opera do not have built-in support for VBScript.

Client-side (Flash and Silverlight)

- Flash is commonly used to create animation, advertisements, and various web page Flash components, to integrate video into web pages, and to develop rich Internet applications.

- Silverlight is a web application framework developed by Microsoft. Initially released as a video streaming plugin, later versions brought additional interactivity features and support for .NET languages and development tools.

Server-side (Perl)

- Perl has quietly and without much hype, provided a better case-study of extensible and reusable software design than almost anything that has gone before it. It is very hard to find a computer system that hasn't some form of support for Perl available.

- An astonishing range of free extensions (modules) exist for Perl, by far outweighing the impact of Perl simply as a vehicle for expressing programs.

Server-side (PHP)

- PHP is free for download and works very well on all of the Unix-like, Linux and Windows platforms.

- PHP has a thriving developer community, numerous contributors, a wide range of features (extensible through plug-ins) and is used in thousands of websites around the world.

Server-side (ASP)

- ASP (Active Server Pages) is a rounded proprietary product from Microsoft.
- Regrettably, it is only properly supported on Microsoft's IIS platform.
- Most ASP pages are written in VBScript, but any other Active Scripting engine can be selected instead.

Server-side (ASP.NET)

- ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

- ASP.NET encourages the programmer to develop applications using an event-driven GUI model, rather than in conventional web-scripting environments like ASP and PHP.

- Compiled code means applications run faster with more design-time errors trapped at the development stage.

Server-side (JSP/Servets)

- Java Server Pages is reminiscent in nature of the Microsoft ASP framework, but intended from the ground up for enthusiasts of Java, Enterprise Beans and the whole industry that has grown up around them.

- JSPs are compiled into Java Servlets by a JSP compiler. A JSP compiler may generate a servlet in Java code that is then compiled by the Java compiler, or it may generate byte code for the servlet directly. JSPs can also be interpreted on-the-fly.

Server-side (Python)

- Python is an interpreted, object-oriented language developed as a full-featured, but easy to use, scripting language developed in a Unix environment, Python is now available on PCs and Macs, and applications are portable across platforms.
- Python is more of a regular programming language, but simpler and easier to program than Java.

Server-side (Tcl)

- Tcl (tool command language) is a widely used scripting language generally used in conjunction with the Tk GUI library for building quick and easy X windows GUIs on Unix platforms, but also valued for the ease by which C libraries can be imported and referenced from Tcl.
- Tcl is higher level, quicker to program, and slower to execute; Java is lower level, harder to program, quicker to execute.

Server-side (Ruby)

- Ruby is a dynamic, reflective, general purpose object-oriented programming language that combines syntax inspired by Perl with Smalltalk-like features.
- Ruby supports multiple programming paradigms, including functional, object oriented, imperative and reflection. It also has a dynamic type system and automatic memory management making it similar in varying respects to Python or Perl.

RSS

- RSS is a family of Web feed formats used to publish frequently updated works such as blog entries, news headlines, audio, and video in a standardized format.

- An RSS document (which is called a "feed," "web feed," or "channel") includes full or summarized text plus metadata such as publishing dates and authorship.

HTTP

- HTTP (for HyperText Transfer Protocol) is the primary method used to convey information on the World Wide Web. The original purpose was to provide a way to publish and receive HTML pages.

- HTTP is a request/response protocol between clients and servers. An HTTP client, such as a web browser, typically initiates a request by establishing a TCP connection to a particular port on a remote host (port 80 by default).

HTTP

- An HTTP server listening on that port waits for the client to send a request string, such as "GET / HTTP/1.1" (which would request the default page of that web server), followed by an email-like MIME message which has a number of informational header strings that describe aspects of the request, followed by an optional body of arbitrary data.

HTTP

- Upon receiving the request string (and message, if any), the server sends back a response string, such as "200 OK", and a message of its own, the body of which is perhaps the requested file, an error message, or some other information.
- GET, by far the most common method, requests a resource by specifying a URL.
- POST, is similar to GET, except that a message body, typically containing key-value pairs from an HTML form submission, is included in the request.

HTTP client request

- Other methods are HEAD(Fetch just the header of the document), PUT(store a new document on the server) and DELETE (remove a document from the server).
- `GET / HTTP/1.1`
- `Host: ryerson.ca`

HTTP server response

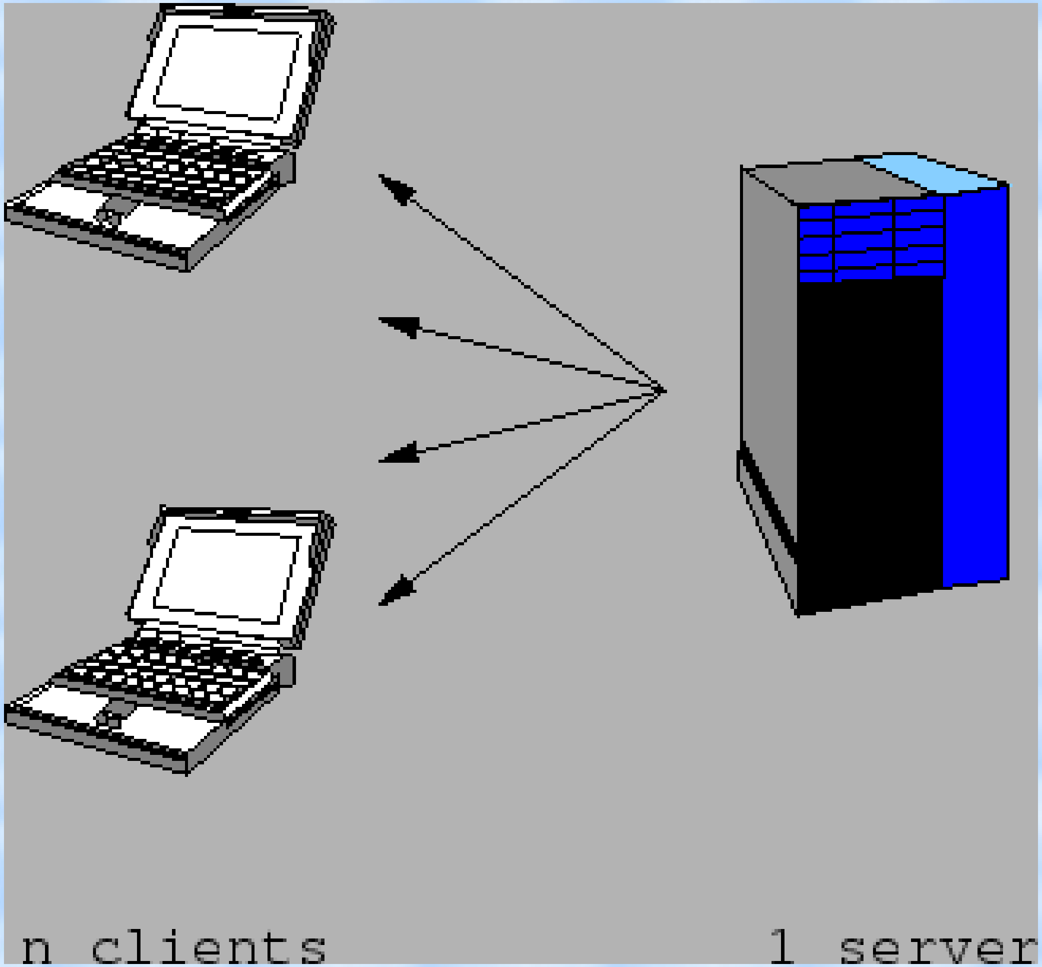
```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (Unix)  (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
```

HTTP server response

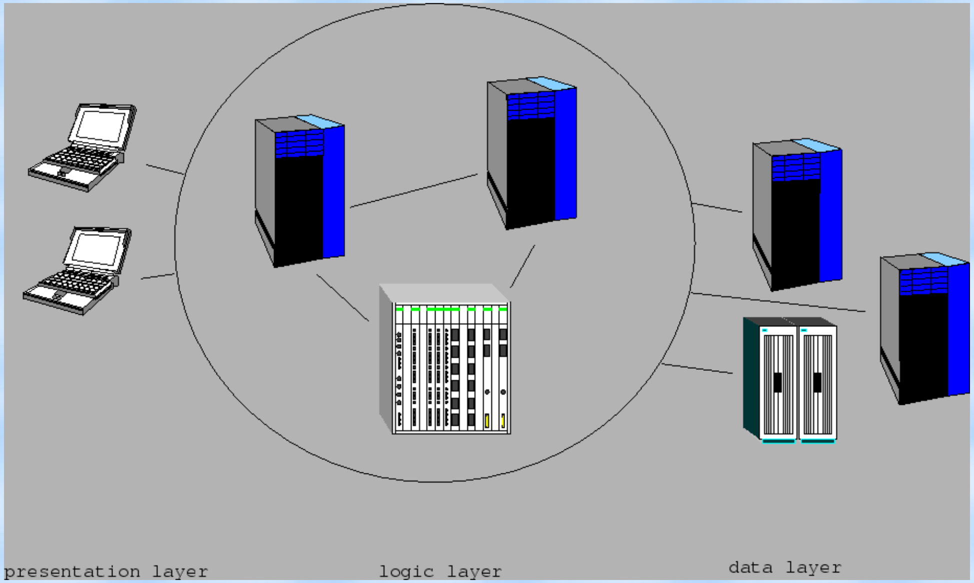
- Status code is a three-digit number; first digit specifies the general status:
- 1 => Informational
- 2 => Success
- 3 => Redirection
- 4 => Client error
- 5 => Server error

Client-server architecture

- Processing requirements are mostly on the client side.
- Each client needs multiple requests for data.
- Best for few clients.



Three-tiered architecture



- The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging.

Security

Security issues for a communication between a browser and a server are

1. Privacy: communication must not be intercepted.

2. Integrity: content must not be modified.

3. Authentication: identity must be confirmed.

4. Non-repudiation: communication must be legally proved as being taken place.

Security

Security issues for a communication between a browser and a server are:

The basic tool to support privacy and integrity is encryption.

Everyone uses your public key to encrypt messages sent to you. You decrypt them with your matching private key. It works because it is virtually impossible to compute the private key from a given public key.

RSA is the most widely used public-key algorithm.

Other Security Issues

Destruction of data on computers connected to the Internet.

Viruses and worms

Denial-of-Service (DoS): Created by flooding a Web server with requests.

Systems Development Options

1. Pure HTML and CSS

The base for all web systems development. Ultimately, almost the only possible technology for web deployment. Almost all the other approaches generate an and product in HTML and CSS.

Very versatile but time consuming. A must however for every serious web developer.

Systems Development Options

2. Dynamic server-side and client-side programming

An extension of option #1. Here a programming language generates the HTML and CSS codes to render web pages.

Also very versatile and time consuming. Offers dynamic content. Also a must for every serious web developer.

Systems Development Options

3. Web applications frameworks

A semi-versatile approach to web development. It is based on coding. If you are a programmer or a web developer that loves coding and do not want to go the pure HTML route. Some frameworks are back-end, others front-end.

Among the most popular are AngularJS, Bootstrap, Laravel, CodeIgniter, Symfony, CakePHP, Ruby on Rails, and Django.

Systems Development Options

4. Content management systems (CMS)

A CMS is a GUI management service to maintain and build a website. It features include creating pages and navigation, modifying objects on pages, analysing stats, and much more.

No need for any technical training or expertise but it is less versatile. The most popular are WordPress, Joomla and Drupal.

Systems Development Options

5. Flash

The only approach not to use HTML at all. Requires knowledge of the Flash development software and language.

Good for amazing graphic displays but weak on search engine optimization and not supported on the very popular iOS devices.

Systems Development Options

Recommendations:

This is personal opinion of course, but my recommendations are either the pure HTML/CSS with back-end server-side programming (#2) for its required knowledge (can't call yourself a web developer if you don’t know those) and extreme versatility and CMS like WordPress, Joomla, or Drupal (#4) for their ease of use and speed of development as well as nicer and more modern looking results.

End of lesson