

Web 2.0 Technology Overview

Lecture 8
GSL Peru 2014

Overview

What is Web 2.0?

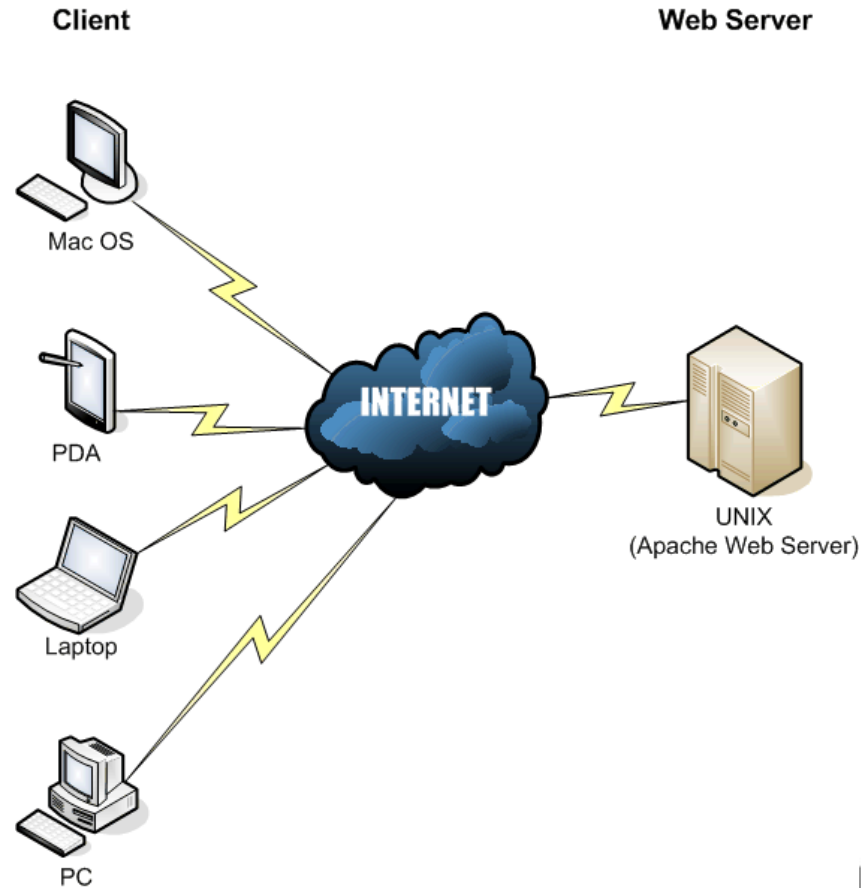
Sites use technologies beyond static pages of earlier websites.

- ❖ Users interact and collaborate with one another
 - Rich user experience with dynamic content
 - Users participate and contribute
 - Social networking sites, blogs, wikis, video sharing sites

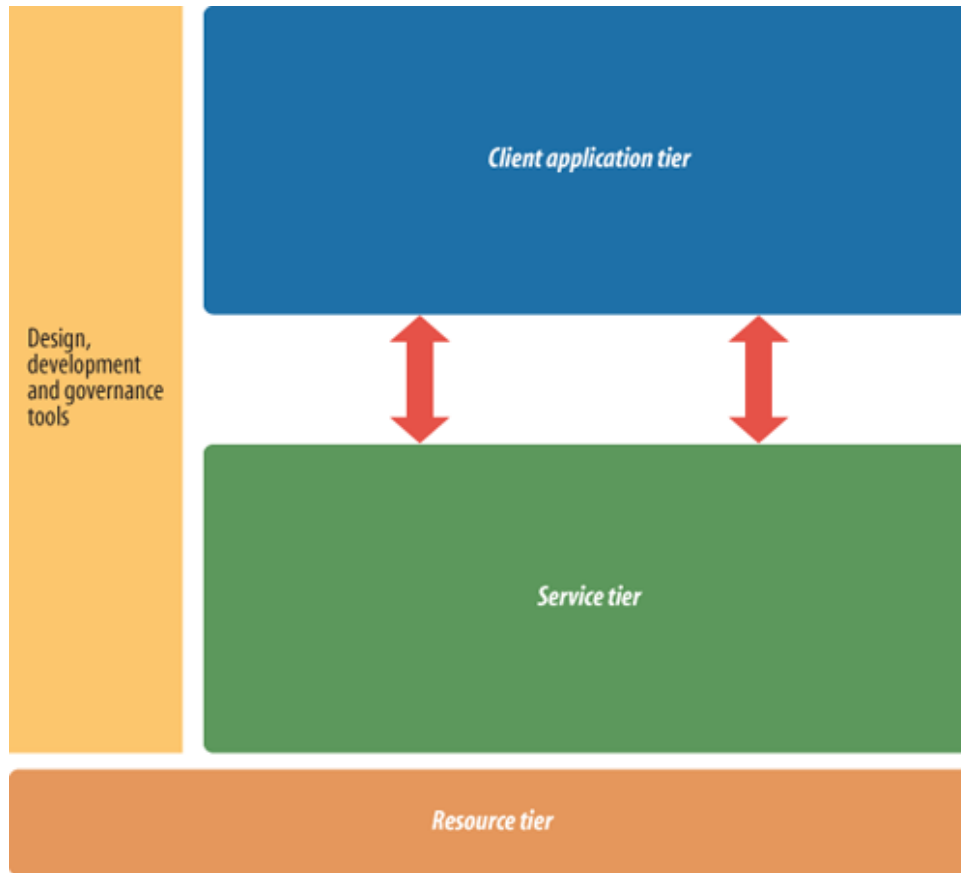
Examples

- ❖ Social Networking
 - Facebook, Twitter, LinkedIn
- ❖ Photo and Video Sharing
 - Flickr, Youtube
- ❖ Other
 - Google Docs, Google Maps, Google Calendar

Client/Server Model



System Architecture

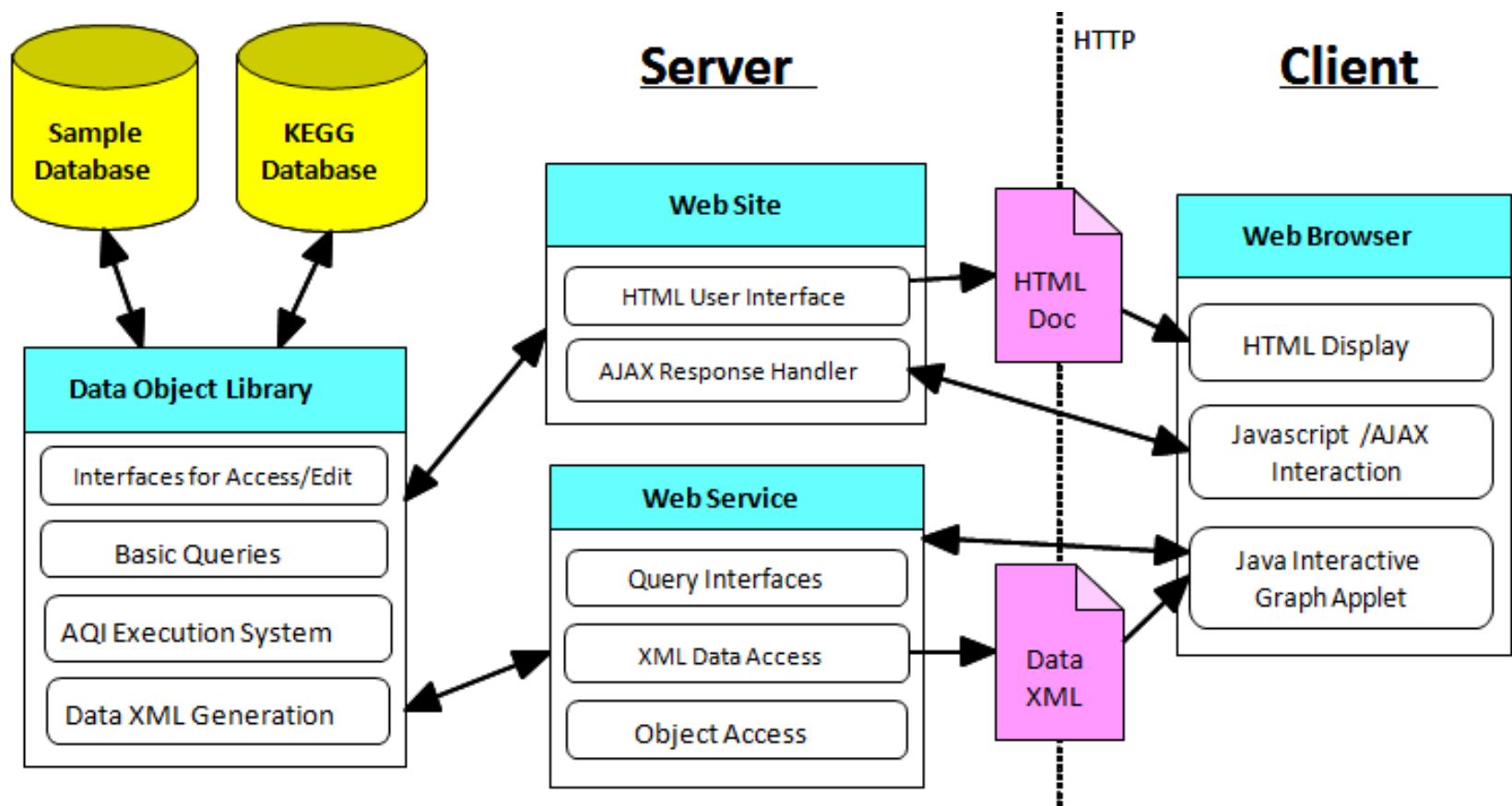


- ❖ Resource tier: backend systems, files, and databases
- ❖ Service tier: connects resources to web through frameworks including PHP, Rails, ASP
- ❖ Client application tier: client-side views such as web browsers
- ❖ Design, development, and governance tools: tools to build the web applications including IDEs, xCode, Adobe Dreamworks

Source: www.oreilly.com

System Architecture

Client-Server Model



Client-Server Model

Client and Database

- ❖ Client: web browser
 - Google Chrome, Internet Explorer, Mozilla Firefox
 - HTML5, CSS3, JavaScript, AJAX
- ❖ Server
 - Apache, Microsoft IIS, lighttpd
- ❖ Database: backend data
 - MySQL, Oracle Database

Client-Server Model

Server

- ❖ Server: web server
 - Apache, IIS
 - PHP
 - Python (Django)
 - Rails (Ruby)
 - JavaScript (jQuery, Node.js)
 - ASP (Asp.Net, C#)
 - JSP (Java/EJB)
 - Flash
 - CGI/Perl

Evolution of Web Technologies

Flash to HTML5

Flash: Adobe's product for website engine

- ❖ Search engines do not like Flash
- ❖ Ignores user needs
 - Splash sites, site intros
 - Disabled back-button
- ❖ Requires a lot of bandwidth
- ❖ Better uses than for website engine:
 - Ads & banners, games, video streaming
- ❖ Stores data on the client

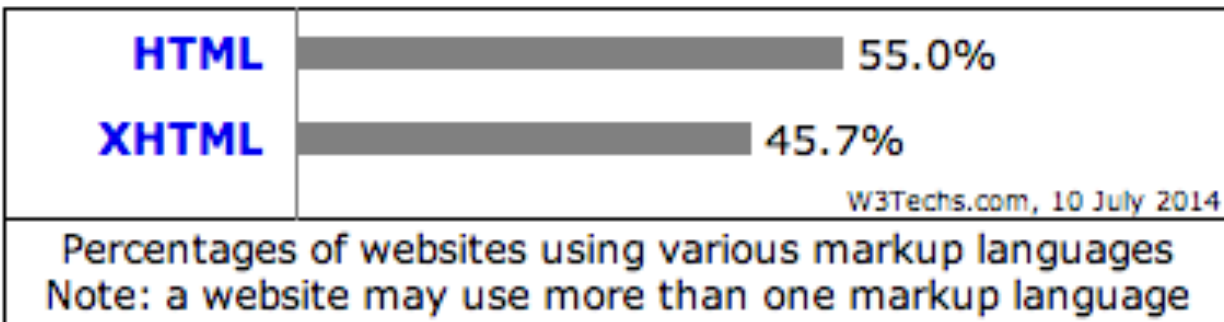
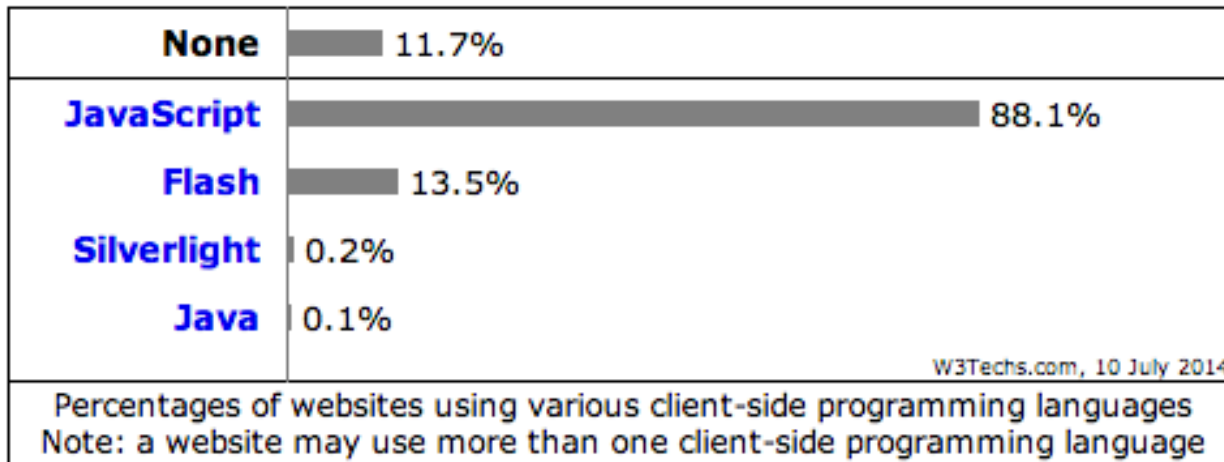
Evolution of Web Technologies

Flash to HTML5

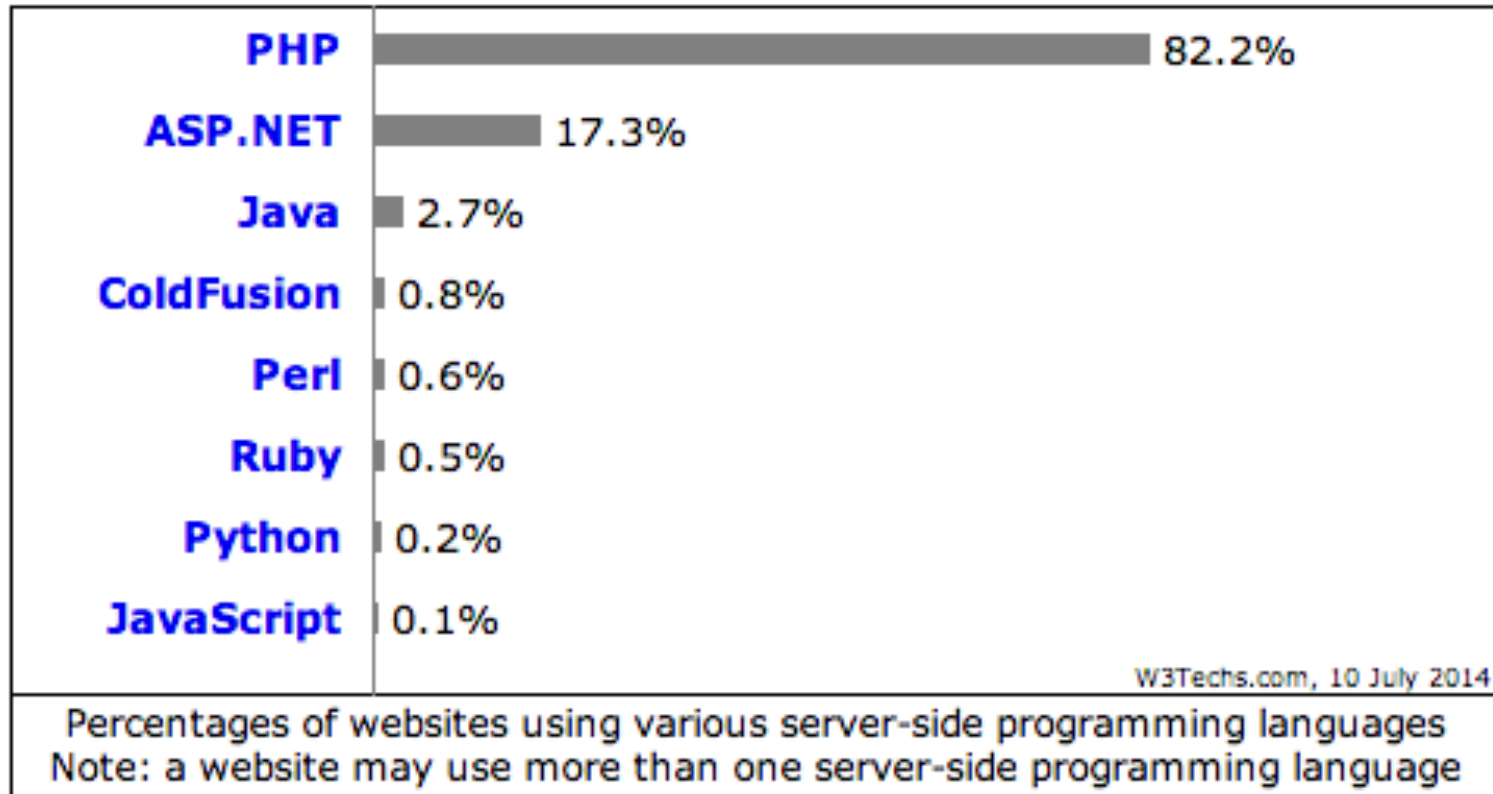
HTML5

- ❖ Mobile web application development
- ❖ All browsers can use it
- ❖ Game development
 - Alternative to Flash!
- ❖ Dynamic web applications
 - Drag and drop capabilities, browser history management, document editing
- ❖ Cleaner, descriptive semantics / code

Client-Side Languages *Popularity*



Server-Side Languages *Popularity*



Comparing Different Web Technologies

- ❖ **JavaScript:** absolutely necessary to know for front-end development.
 - Necessary to build the UI
 - Also should know HTML5/CSS3
 - Reference/tools: <http://www.w3schools.com/>
 - Not necessary to use Node.js or jQuery frameworks to use JavaScript (more on that later)

Comparing Different Web Technologies

- ❖ **Ruby On Rails:** Ruby is a programming language and Rails is the framework that uses Ruby
 - Popular blackbox platform today
 - Nice, clean language to use
 - Steep learning curve
 - Learn Ruby: <http://www.codecademy.com/en/tracks/ruby>
 - Getting started with Rails: http://guides.rubyonrails.org/getting_started.html

Comparing Different Web Technologies

- ❖ **PHP:** one of the most common back-end, server-side languages
 - Easiest to learn, especially for beginner programmers
 - Very common => several libraries and APIs already exist
 - Not a very good language though
 - Performance one of the slowest
 - Reputation for security issues
 - Reference/tools: <http://www.w3schools.com/>

Comparing Different Web Technologies

- ❖ **Django:** web development framework written in Python
 - Popular blackbox platform today
 - Python has strong support with non-web aspects
 - System administration, data analytics
 - Steep learning curve if not familiar with Python or the framework
 - Tools/References: <https://code.djangoproject.com/wiki/Tutorials>

Comparing Different Web Technologies

- ❖ **Node.js & jQuery:** JavaScript based
 - Good for strong background in JavaScript
 - Growing in popularity
 - Node.js isn't good for large CPU tasks
 - jQuery has simple architecture
 - Both are compatible with JavaScript, which is anyway used for the front-end
 - Tools/References:
 - jQuery: <http://www.w3schools.com/jQuery/>
 - Node.js: <http://code.tutsplus.com/tutorials/nodejs-for-beginners--net-26314>

Comparing Different Web Technologies

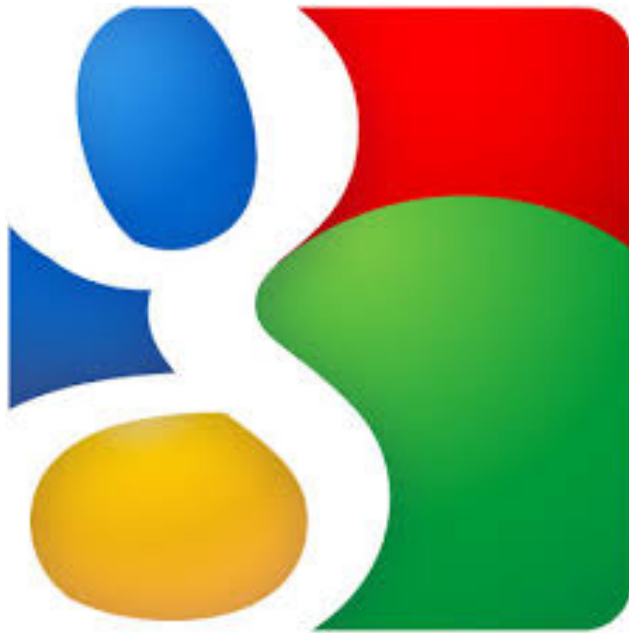
- ❖ **ASP.NET:** language for Microsoft's ASP technology.
 - Can also use C# and Visual Basic
 - Class library system => good maintainability
 - Uses large amount of web server resources than PHP and other languages
 - Runs on IIS
 - Documented bugs and vulnerabilities
 - Tools/Resources: <http://asp.net-tutorials.com/>

**Where are these
technologies used today?**



Web Technologies Used

Google



- ❖ **Server-side:**
 - Mainly Python
 - Also Java, C++
- ❖ **Client-side:**
 - JavaScript
 - HTML, CSS
- ❖ **Database:**
 - BigTable
- ❖ **Web Server:**
 - Google Web Server

Web Technologies Used

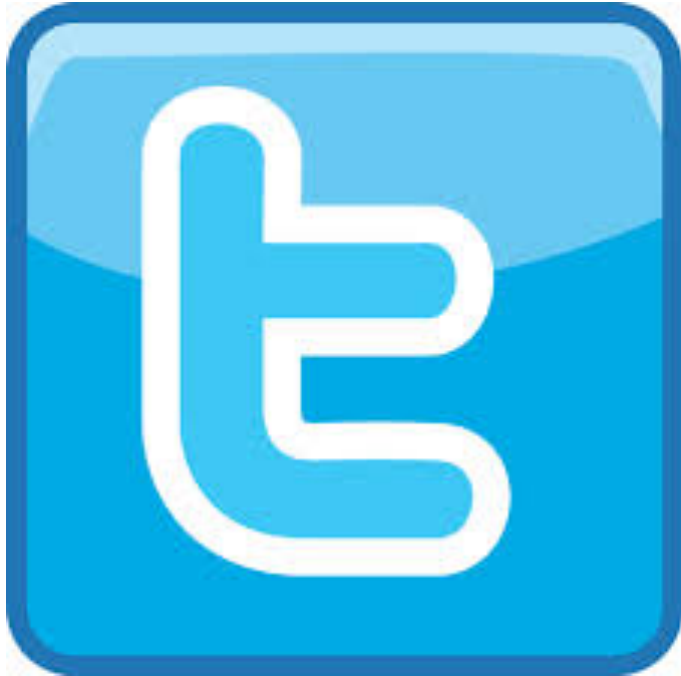
Facebook



- ❖ **Server-Side:**
 - Mainly PHP
 - Also C++, Java, Python
- ❖ **Client-Side:**
 - JavaScript
 - HTML/CSS
- ❖ **Database:**
 - MySQL, HBase

Web Technologies Used

Twitter



- ❖ **Server-Side:**
 - C++, Java, Ruby on Rails
 - Also Scala
- ❖ **Client-Side:**
 - JavaScript
 - HTML, CSS
- ❖ **Database:**
 - MySQL

Web Technologies Used

Youtube



- ❖ **Server-Side:**
 - C/C++
 - Also Java, Python
- ❖ **Client-Side:**
 - **Flash**
 - JavaScript
- ❖ **Database:**
 - MySQL
 - BigTable

Web Technologies Used

Bing



- ❖ Server-Side:
 - ASP.NET
- ❖ Client-Side:
 - JavaScript
 - HTML, CSS
- ❖ Database:
 - Microsoft SQL Server

Now, for selecting your web technology

Here are some things to consider...



Selecting Your Web Technology

Factors to Consider

❖ Needs vs. Strengths

- Figure out what strengths you need most; ignore the other strengths
- Decide whether it saves time or not

❖ Testing

- Testable with unit testing and integration testing?
 - Lecture on testing to come in future...

❖ Documentation

- Choose technology with well-written and easy-to-understand documentation
- Sample code and tutorials available

Selecting Your Web Technology

Factors to Consider

❖ Servability

- Speed, hosting, and operation costs matter!

❖ Security

- Check track record of common vulnerabilities, such as database injections
- Check track record of maintainance

❖ Longevity

- Compare how long technology has been around or will be around moving forward

Selecting Your Web Technology

Factors to Consider

- ❖ Learning Curve and Hireability
 - Know if the technology is difficult to learn at first and easy to master later, or vice versa
 - Know if the skills needed are easy to find today

How do we implement?



Hardware



Server Hardware



Compute Server

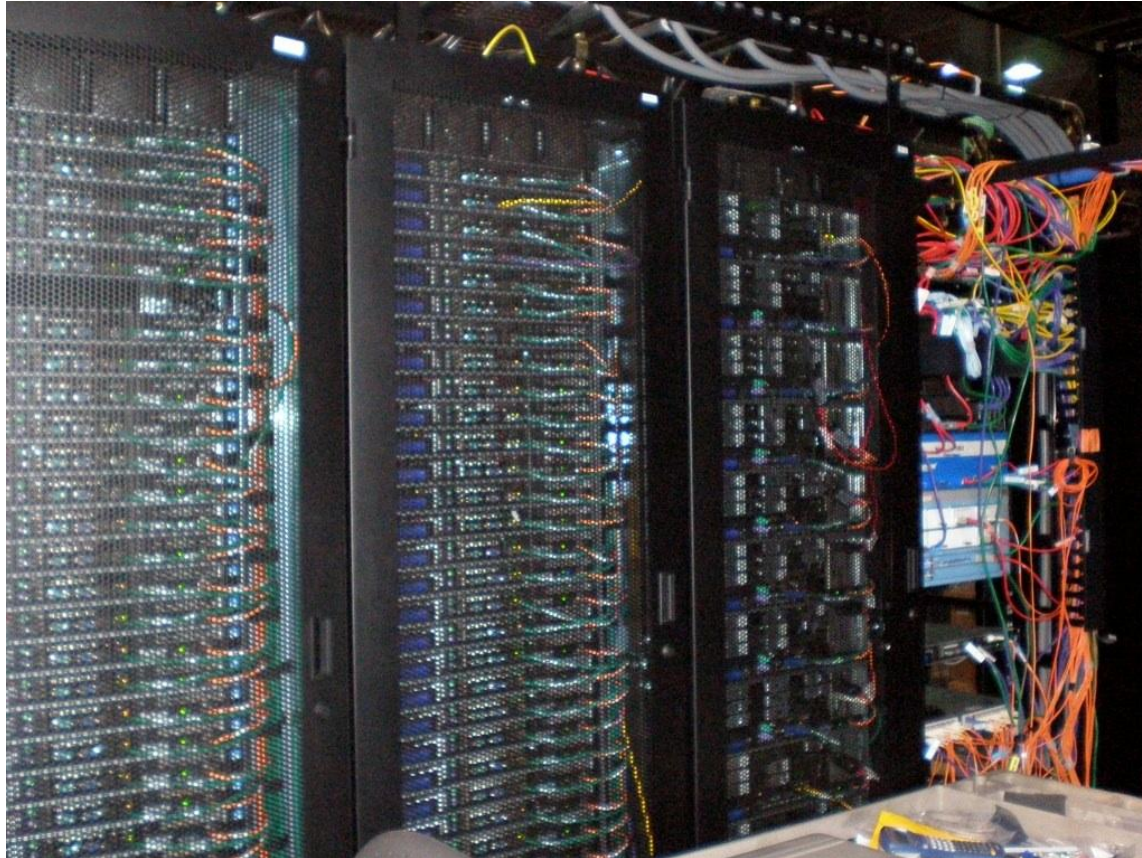
Storage - SAN



Data Center

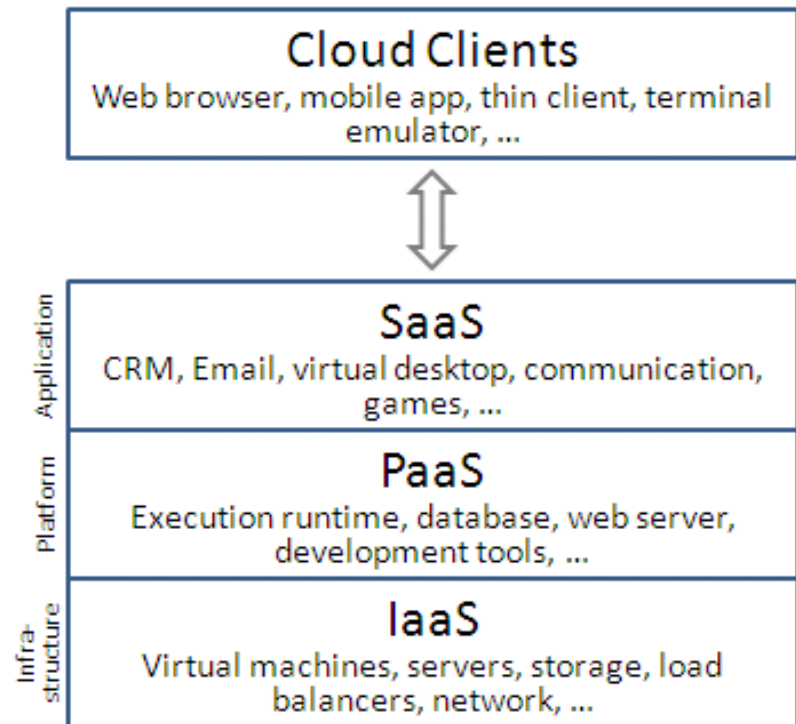


Data Center



Cloud Services

Can Provide
Scalability without
capital investment



Platform as a Service (PaaS)

- Provides computing platforms as a subscription service
- No need to know how to maintain/administer the platform - OS, Development Environment, Database, Server
- Automatic provisioning
- Example - Heroku
 - Originally, Ruby
 - Now, Java, Node.js, Scala, Clojure, Python and PHP

Infrastructure as a Service (IaaS)

- Provides physical or virtual machines along with resources such as storage in a subscription model
- More control, but more administrative/maintenance overhead

Rackspace

- ❖ **Cloud Sites: web application hosting**
 - PaaS
 - Fixed, monthly payment
 - Supports most application frameworks, but not Java (server-side) at this time
- ❖ **Cloud Files: cloud storage**
 - Unlimited online storage
 - Online control panel to manage
- ❖ **Cloud Storage: virtual, private servers**
 - IaaS

Amazon AWS

- ❖ Amazon Elastic Cloud Compute (EC2) is central part of cloud computing platform
 - Users create, launch, and terminate server “instances” as needed (instances = VMs)
 - Pay by the hour (\$0.013/hr)
- ❖ Amazon Simple Storage Service (S3) is online file storage web service
 - Web hosting, image hosting, storage for backup systems
 - Stores more than 2 trillion objects as of 2013
 - Pay \$0.15 per gigabyte per month

Examples

- Netflix
- DropBox
- Reddit
- Foursquare

Microsoft Azure

- ❖ PaaS and IaaS services for Microsoft-specific and third party systems
 - Web hosting for PHP, ASP.NET, Node.js, and Python (PaaS)
 - Virtual Machines run Windows and some Linux distributions (IaaS)

Google

- ❖ Google Cloud Storage for developers, Google Drive for non-developers (personal)
 - Can integrate both together
 - PaaS: Google App Engine
 - IaaS: Google Compute Engine
- ❖ User-friendly GUI to manage projects and objects (all your data)
- ❖ Used by Snapchat, Khan Academy, Pulse, and more