

CS6103 - APPLICATION DEVELOPMENT PRACTICES

C-L-T-P-EL-C pattern: 5 -1 -0- 4 -3- 4

Web Application Ecosystem

Web standards (W3C)	Unicode, JSON, XML, CSS, Internationalization	https://www.w3.org/
Client-side	HTML, CSS, JavaScript	5.0, 3.0, ECMAScript 2022
Server-side	PHP, Node.JS, Python, ASP.NET etc.,	8.2, 19.8.1,
Databases	MySQL, MongoDB, Oracle	
Libraries	jQuery, React	
Frameworks	Express, Angular, Zend/Laminas, BootStrap	
CMS	WordPress, Drupal, Joomla	
TOOLS	HTML checker, Visual Studio Code, Notepad++, Bluefish	



Some jargon

Browsers, Scripts, Client-side, server-side, 3-tier architecture, full-stack, libraries, frameworks, APIs, apps, standards, protocols, validation, security, web of things, IOT, CPS, web services, cloud, servers, databases, editors, XML, JSON, Microservices, bigdata, Web app, Mobile app, no code, low code, deep web, dark web, Responsive, URL, Cookies, Wiki, CMS, Interpreters, Templates, Themes, Analytics, Accessibility, Open Source, meta data, Search Engines, Hypertext, SPA, SaaS, host, domain, ipaddress, request, response, web engine, Mobile Web, Cheat sheet, Static / Dynamic web pages, Rendering, Best Practices, Front-End, Back-End, Business Logic

STANDARDS for Interoperability, accessibility, usability

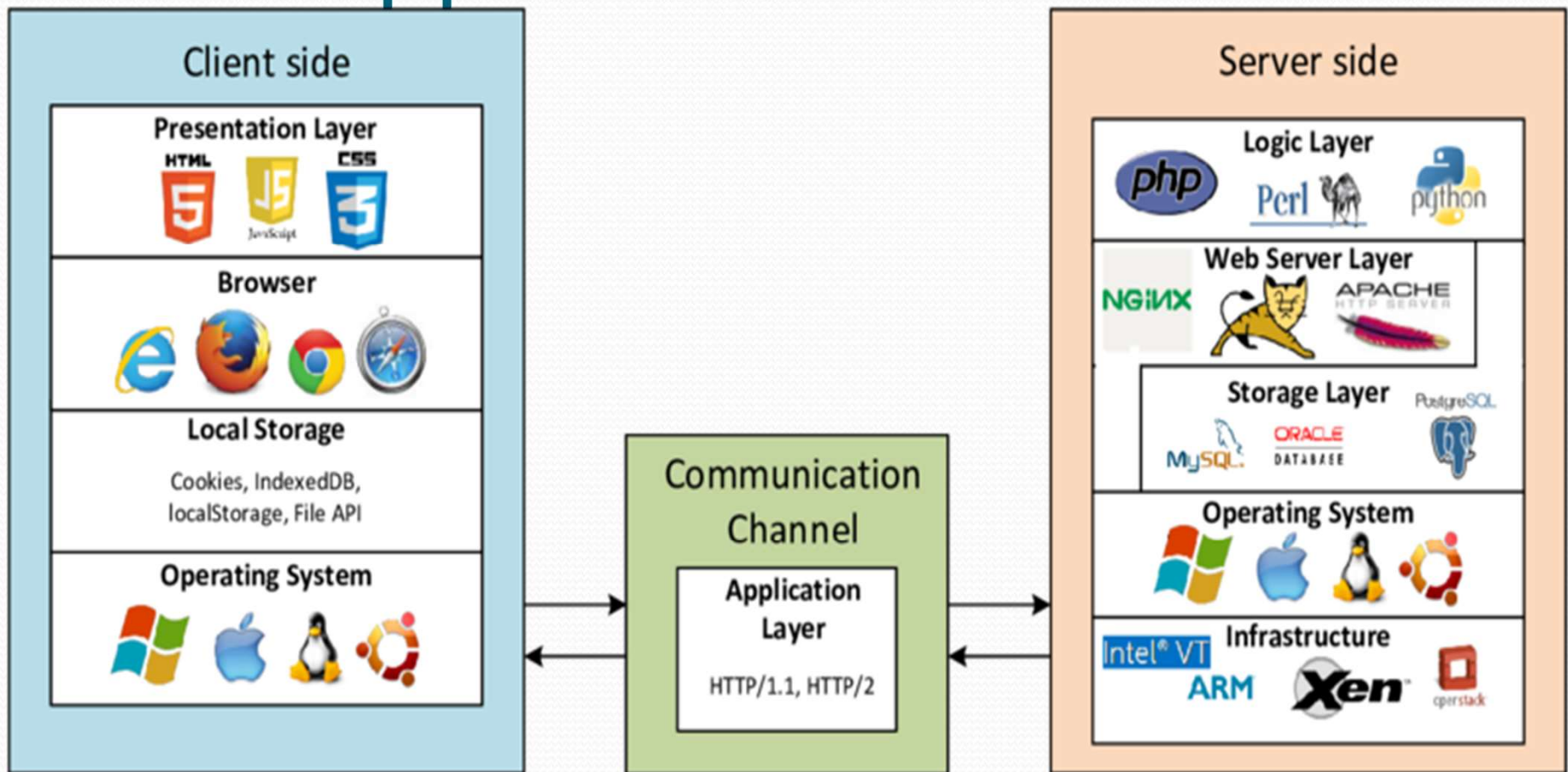
Various ways of Developing Web Apps

- Pure and traditional **Open Source** (PHP, Python, Cold Fusion etc..)
- **Java Based** (Servlets, JSP, JAVA EE, ...)
- **Windows** platform - **.NET based** (ASP.NET)
- Pure JavaScript based - **Node.js**
- Using Languages like Google **Go** (meant only for Google's platform)
- Google's Cloud based web apps (Google App Engine)

FULL STACK WEB DEVELOPMENT

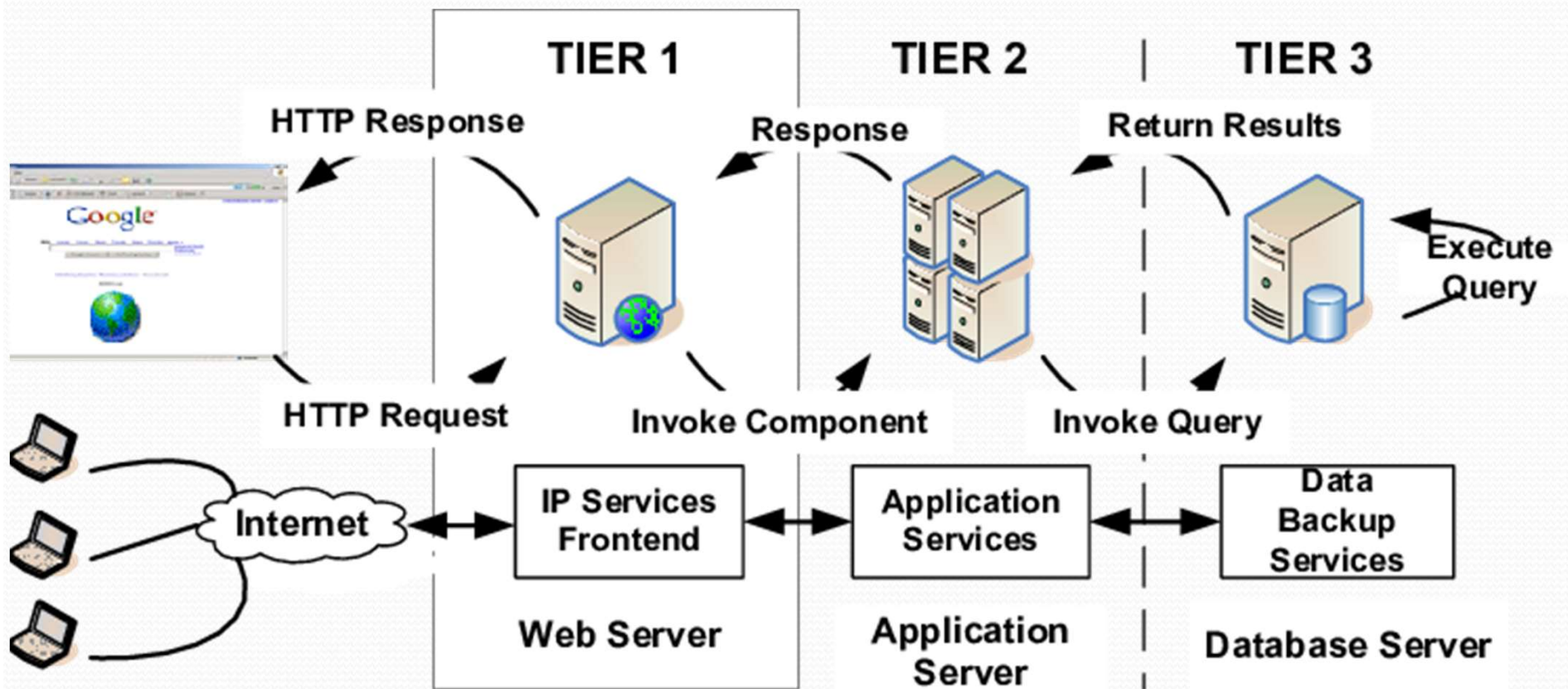
- **Popular Stacks**
 - **LAMP/XAMPP stack**: JavaScript - Linux - Apache - MySQL – PHP / **Apache – MariaDB – PHP-PERL**
 - **MEAN** stack: JavaScript - MongoDB - Express - AngularJS - Node.js
 - **MERN** stack: MongoDB – Express – ReactJS – Node.js
 - **Django stack**: JavaScript - Python - Django - MySQL
 - **Ruby on Rails**: JavaScript - Ruby - SQLite - Rails
- (source: w3schools.com)

Web Application Architecture



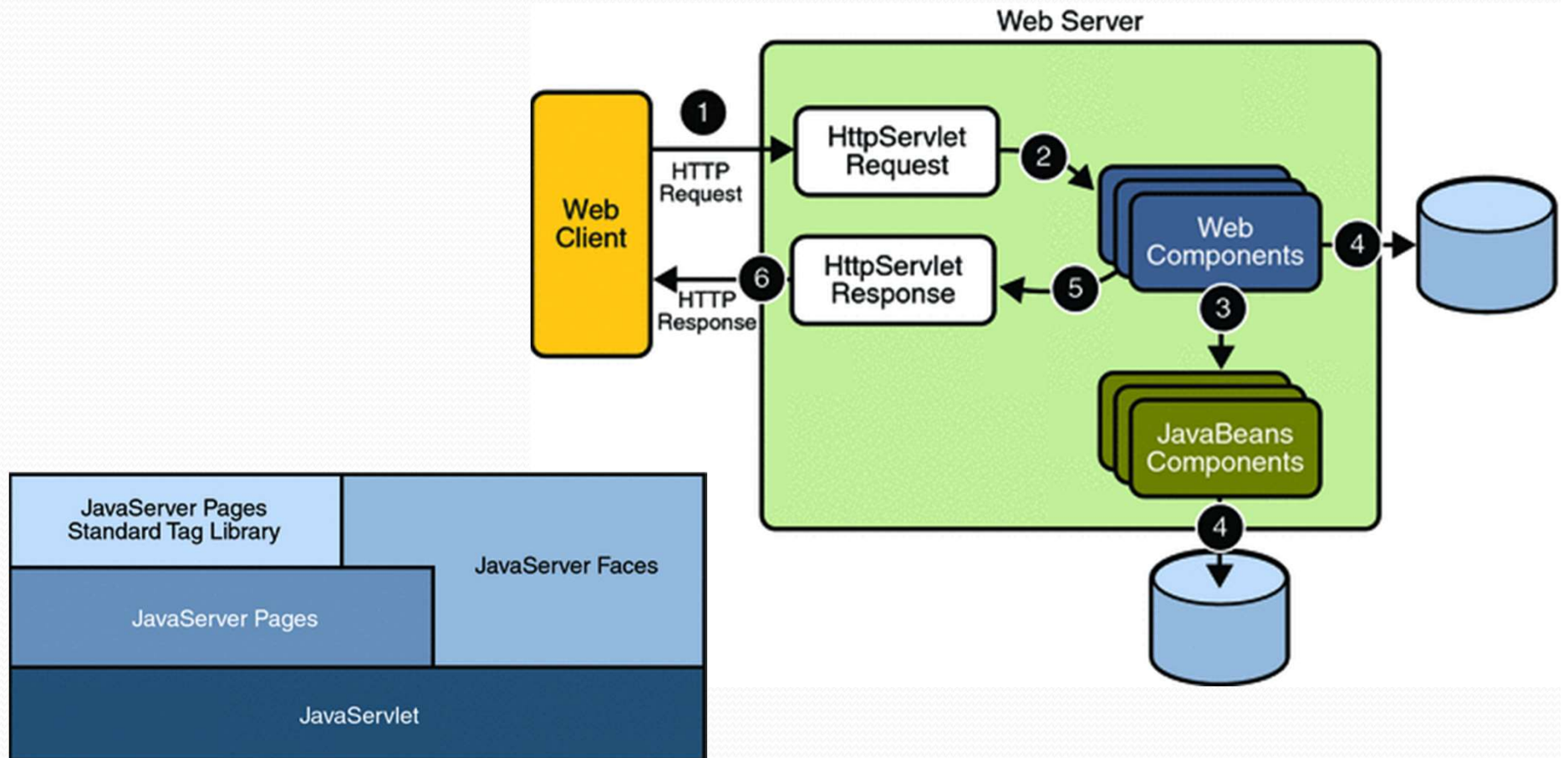
<https://www.researchgate.net/>

Web Application Architecture

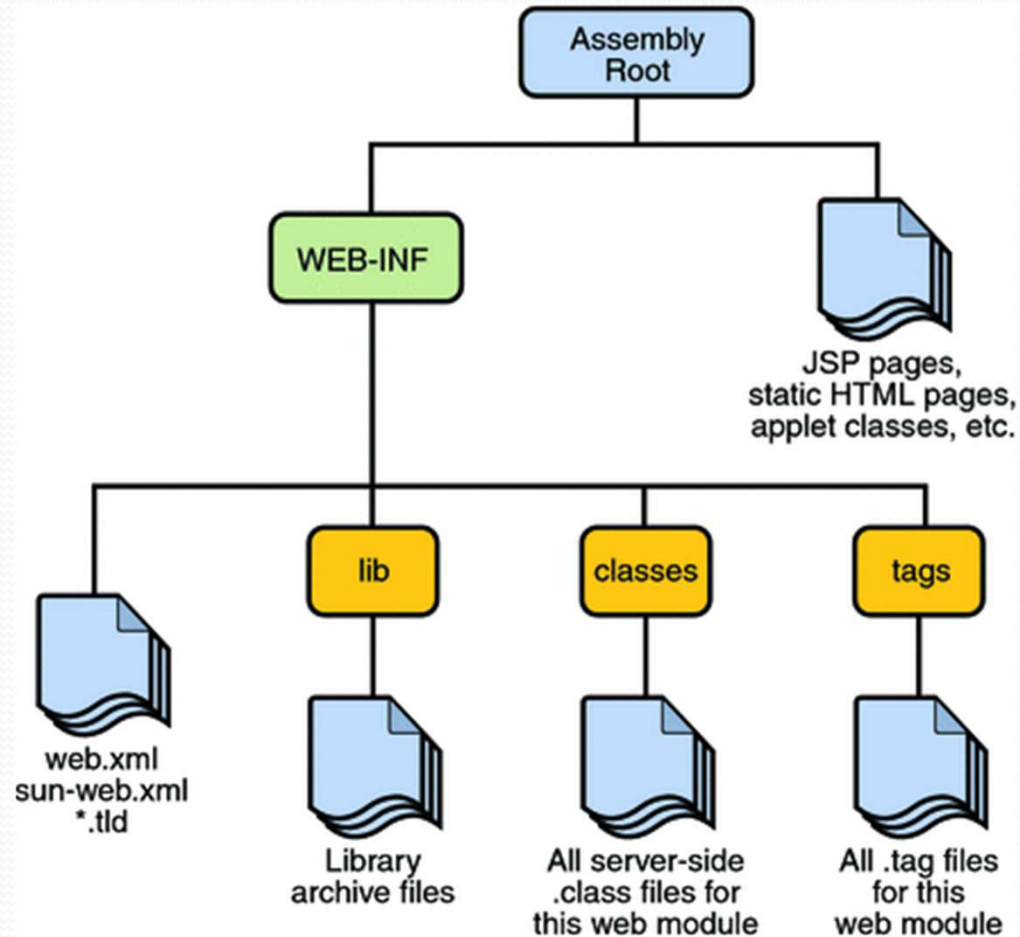


<https://www.researchgate.net/>

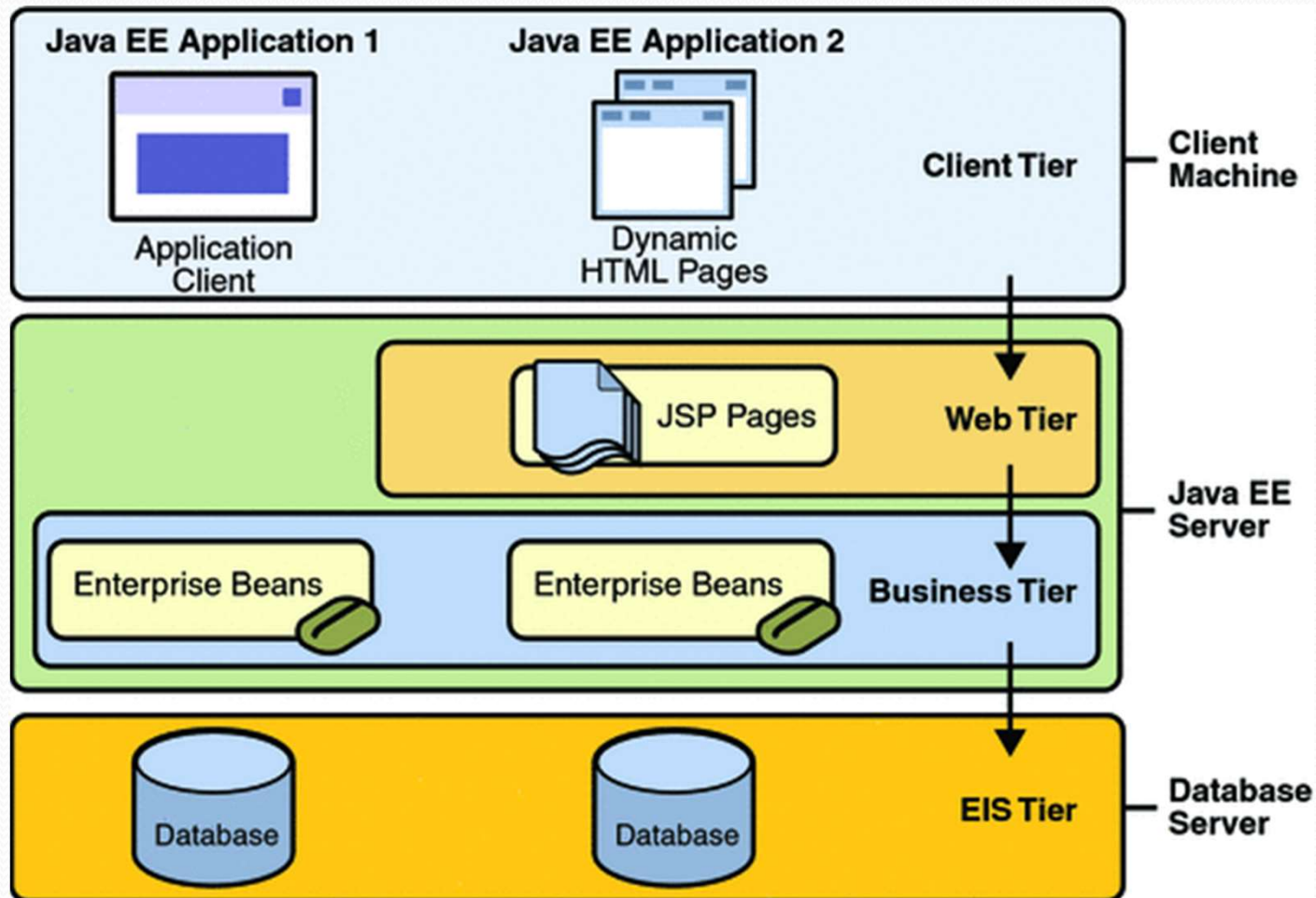
JAVA WEB APP ARCHITECTURES



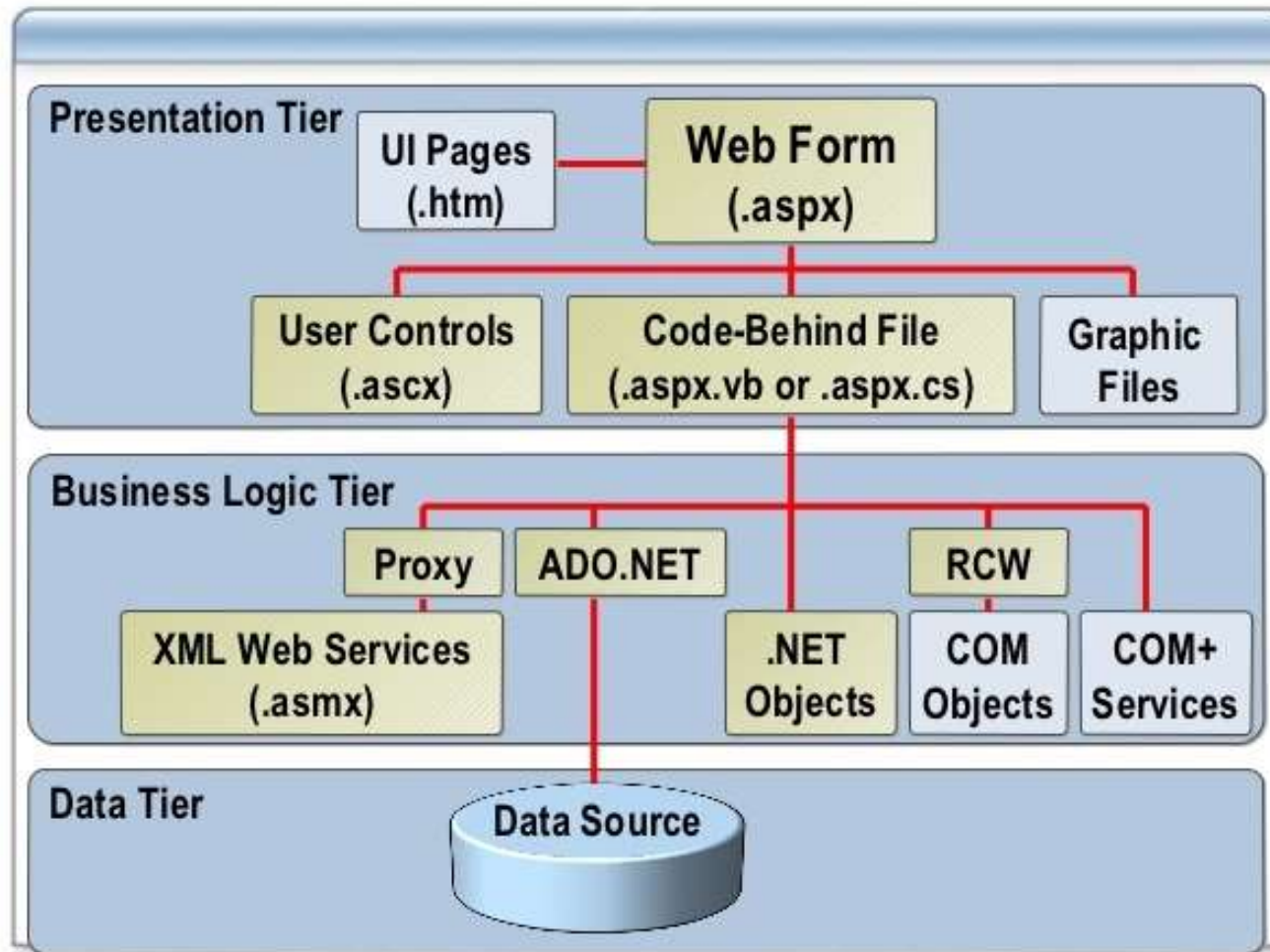
JAVA WEB APP ARCHITECTURES



JAVA WEB APP ARCHITECTURES



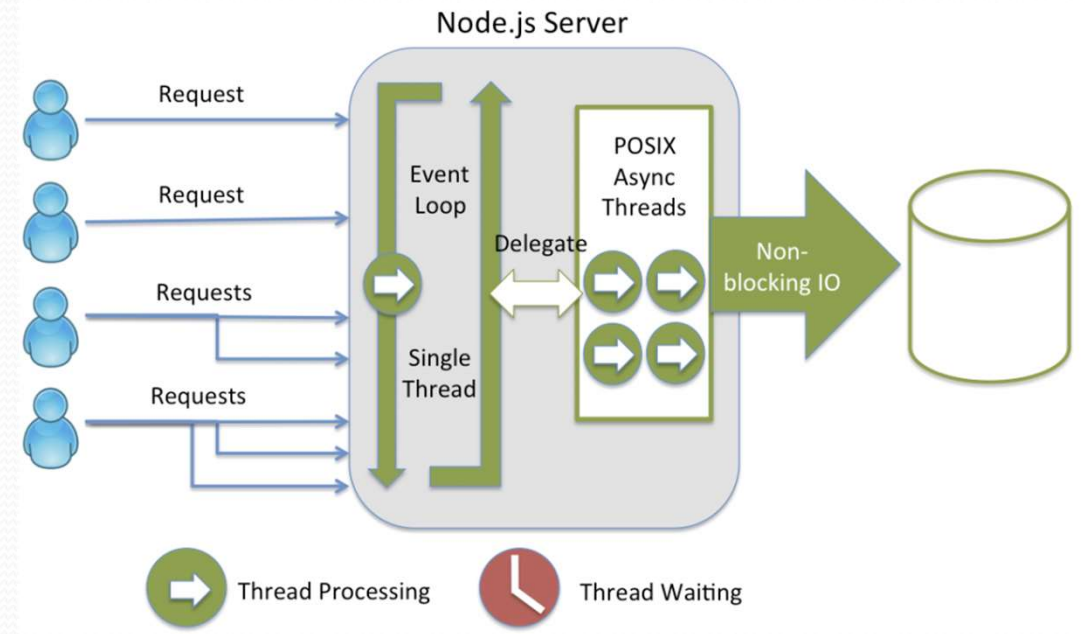
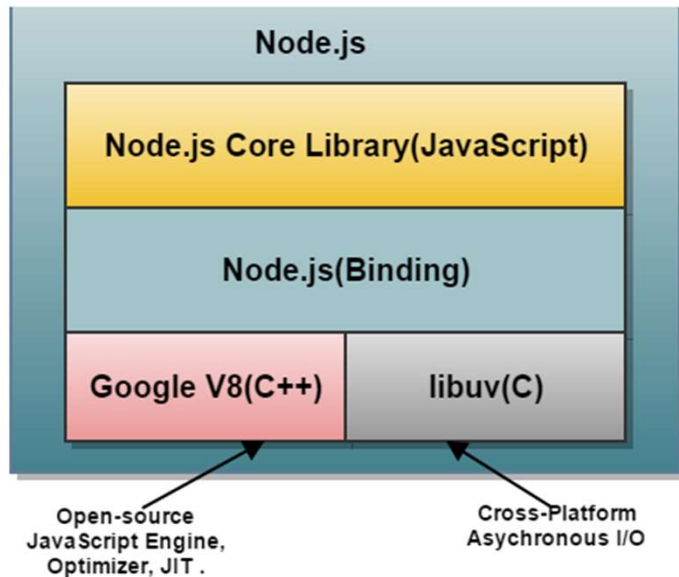
ASP.NET WEB APP ARCHITECTURE



node.js WEB APP (JavaScript everywhere)



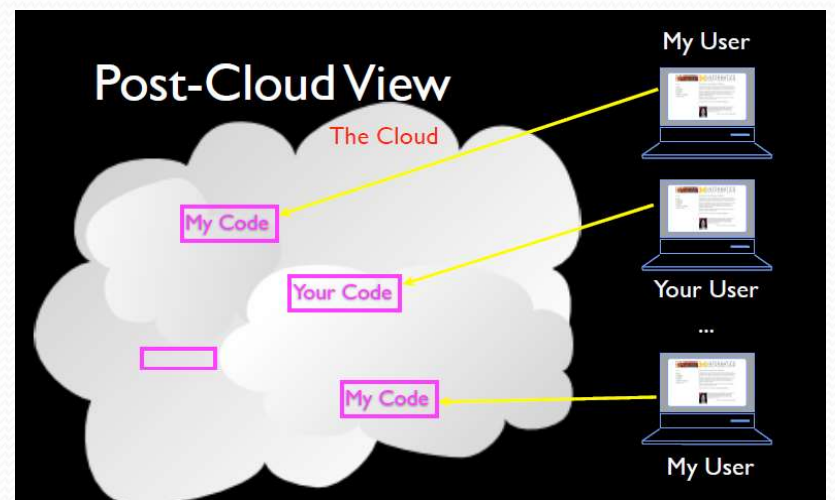
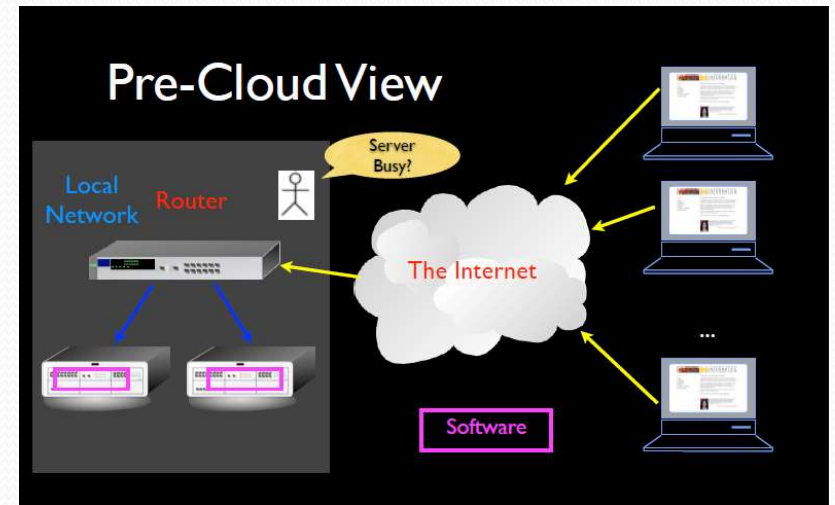
Node.js Architecture



- run JavaScript on the server
- JavaScript runtime built on Chrome's V8 JavaScript engine
- an asynchronous, event driven, light weight, Non-blocking I/O model

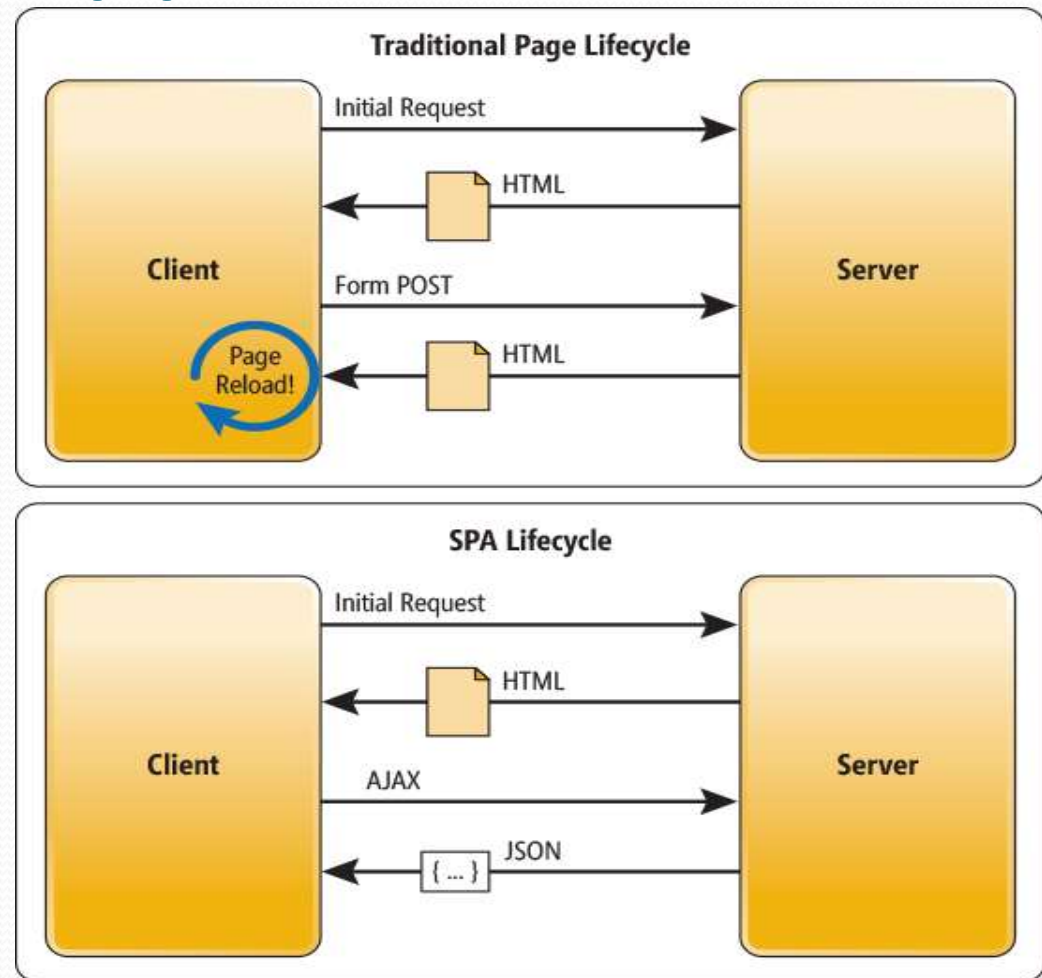
Google App Engine

- ✓ When you write a Google App Engine Application - you are running in the Google Cloud
- ✓ Just like you were a Google Developer
- ✓ You don't know where you are running or if one copy of a thousand copies of you are running
- ✓ Google hosts small applications for *free* - larger applications pay by usage

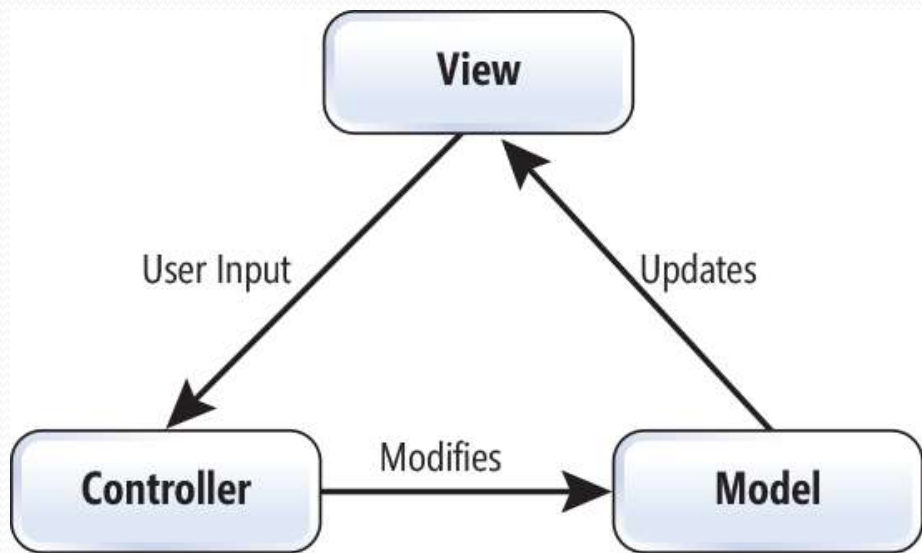


Traditional Vs. SPA (Single Page Apps)

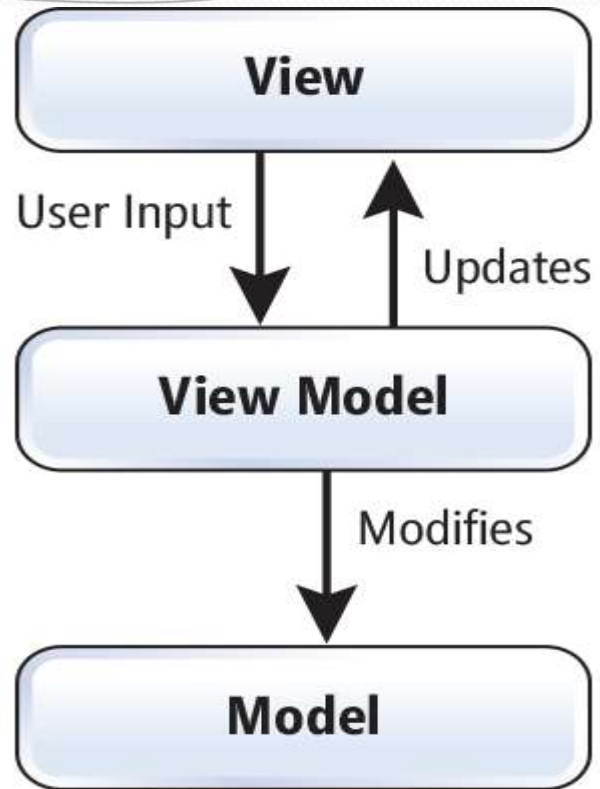
- Single-Page Applications (SPAs) are Web apps that load a single HTML page and dynamically update that page as the user interacts with the app.
- Best Example: **Gmail**
- JS Library: **AngularJS**



MVC vs. MVVM



MVC



MVVM

MVVM	Remarks
Model	Domain data
View Model	Abstract representation of view
View	Displays view model and sends user input to view model



MINDMAP

- [images/mindmap-1](#)



General Questions?

- What is the right combination of technology and platforms?
- Should we rely on cloud?
- Which is more secure technology?
- Should we start with front-end or backend DB.
- Should we write validation code on the client / on the server side?
- What is the expected load?
- How much of our code should be exposed?



High Level Requirements

- **Mobile First:** responsive mobile-friendly design that can be used on different devices - ideally designed with mobile as the foundation
- **Fast Loading:** quick loading times for users on all devices even with slow internet connections
- **Tracking Enabled:** web analytics tracking traffic as well as goals and conversions
- **SEO Ready:** all important on-page SEO tags and elements,
- **Powered by CMS:** a back-end Content Management System that enables publishing and editing content
- **Social Media:** integration with relevant social media platforms
- **Strong Security:** fundamental security and privacy protocols, such as basic security checks



Validation check list

- Validate HTML / XHTML
- Validate CSS
- Validate for (accessibility)
- Validate for WAI standards (accessibility)
- Validate Links (check for dead links)
- Check across different browsers
- Re-validate HTML and CSS
- Have others check your site



Testing

- Functionality
- Usability
- Interface
- Database
- Compatibility
- Performance
- Security
- Crowd

SECURITY TOOLS / SOFTWARE

1. OWASP (Open Web App Security Project)

<https://www.owasp.org>

2. Burp suite (Burp Suite is a Java based Web Penetration Testing framework)

3. Open Source Web Application Vulnerability Scanners

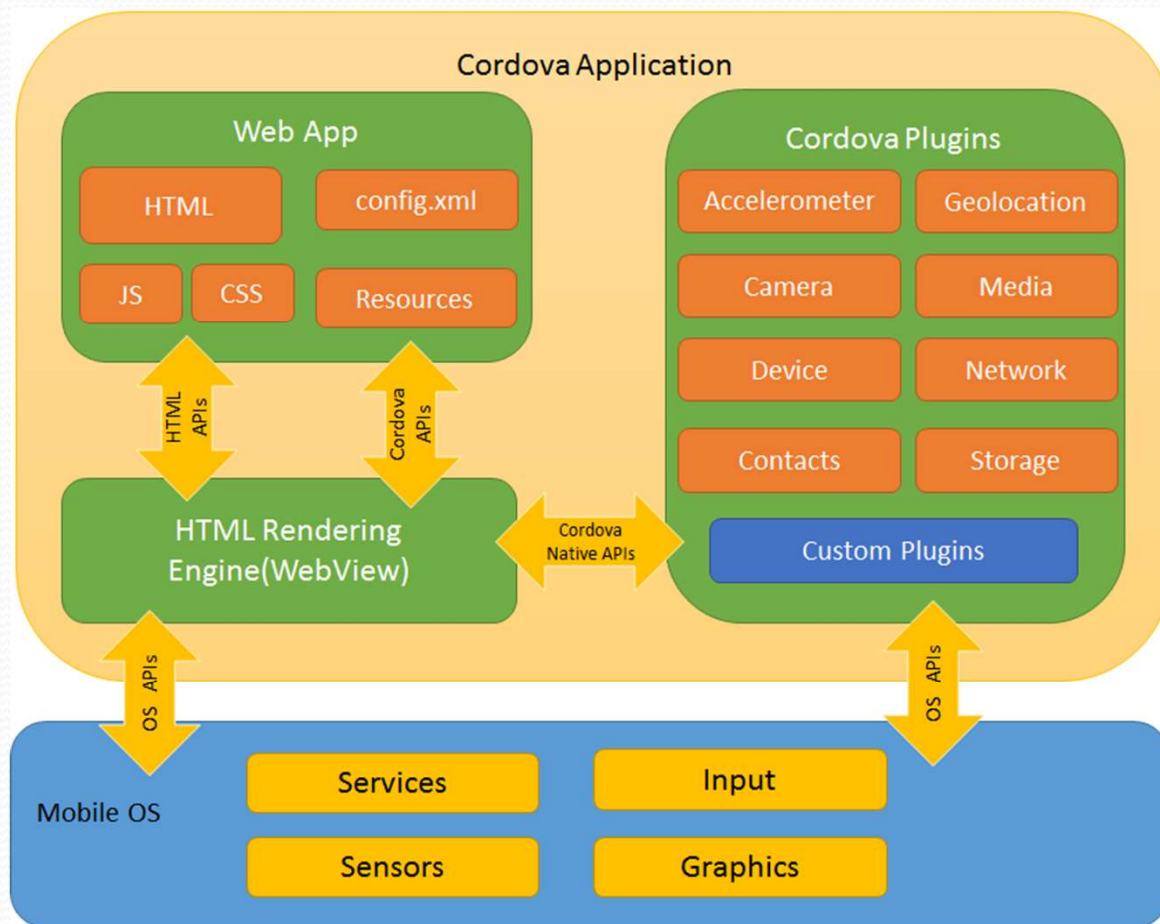
4. Web Application Firewalls





Other utility Software

- Convert Web App to Mobile App with **Apache Cordova**
- Apache Cordova is an open-source mobile development framework.
- It allows you to use standard web technologies - HTML5, CSS3, and JavaScript for cross-platform development.
- Applications execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's capabilities such as sensors, data, network status, etc.



irctc.co.in

Component	Technology
Server Side	Java
Client Side	JavaScript
JavaScript Library	jQuery
Markup Language	XHTML 1.0
Character encoding	UTF-8
Image File formats	PNG, GIF
Site Elements	CSS -External, Embedded, Inline, Cookies – session, HttpOnly, https, HTTPSTS
SSL	Geo Trust Inc.
Traffic Analysis	Google Analytics



Web app validation software

- HTML checker
- Link checker
- Internationalization checker
- Web Platform Tests (interoperability of the web)
- RDF validator

Rendering web pages by browsers

- The two modes
 - Quirks
 - Standards (Almost Standard, Full Standard)

Quirks		Almost Standard	Full Standard
For backward compatibility and emulation of old browsers	For backward compatibility and emulation of old browsers	Less of Quirks and more of Standards	No Quirks and Rendering is based on HTML and CSS spec.
Images are aligned to the bottom	Images are aligned to the bottom	Images are aligned to the baseline	Images are aligned to the baseline

Rendering web pages by browsers

HTML VERSIONS	Declaration	Remarks
HTML5.0	<code><!DOCTYPE html ></code>	Standard
XHTML 1.0	<code><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"></code>	With System Identifier and XML declaration (Standard)
XHTML1.1	<code><?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd" ></code>	With System Identifier and XML declaration (Standard)

Top websites of Tamil Nadu

1-10	11-20
thehindu.com	siliconhouse.net
behindwoods.com	squarebrothers.com
dinakaran.com	vit.ac.in
agriya.com	dotcominfoway.com
tn.gov.in	hosuronline.com
tamilstar.com	pothys.com
ebharatgas.com	tnau.ac.in
india.ford.com	tnvelaivaaiippu.gov.in
kvb.co.in	gati.com
chennaionline.com	thechennaisilks.com

Web Analytics

- Web analytics is the measurement, collection, analysis and reporting of web data for purposes of understanding and optimizing web usage. (Wikipedia)
- Process (Key Process Indicators)

Tools:

- Google Analytics
- Spring Metrics
- Clicky

(Wikipedia, Tutorials Point)





Mobile friendly web apps

Responsive Web Design ([Bootstrap](#) framework)

- Purely front-end
- The goal of **responsive design** is to build web pages that detect the visitor's screen size and orientation and change the layout accordingly.
- Responsive web design makes your web page look good on all devices (desktops, tablets, and phones).
- Responsive web design uses only HTML and CSS.
- *<meta name="viewport" content="width=device-width, initial-scale=1.0">*

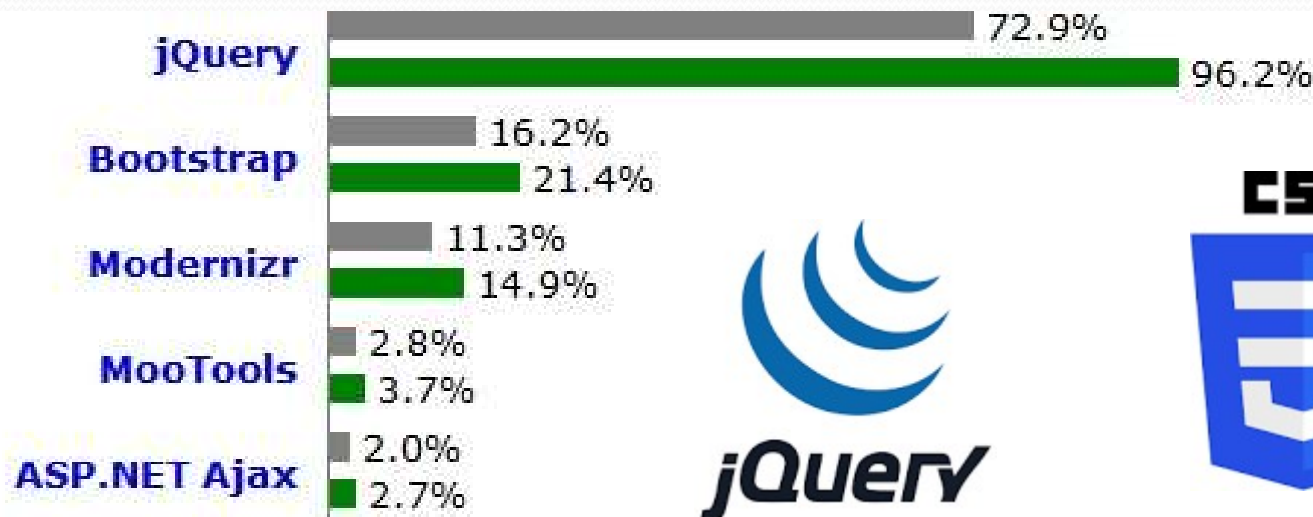
Comparison of Client-Side scripts

As per w3techs.com



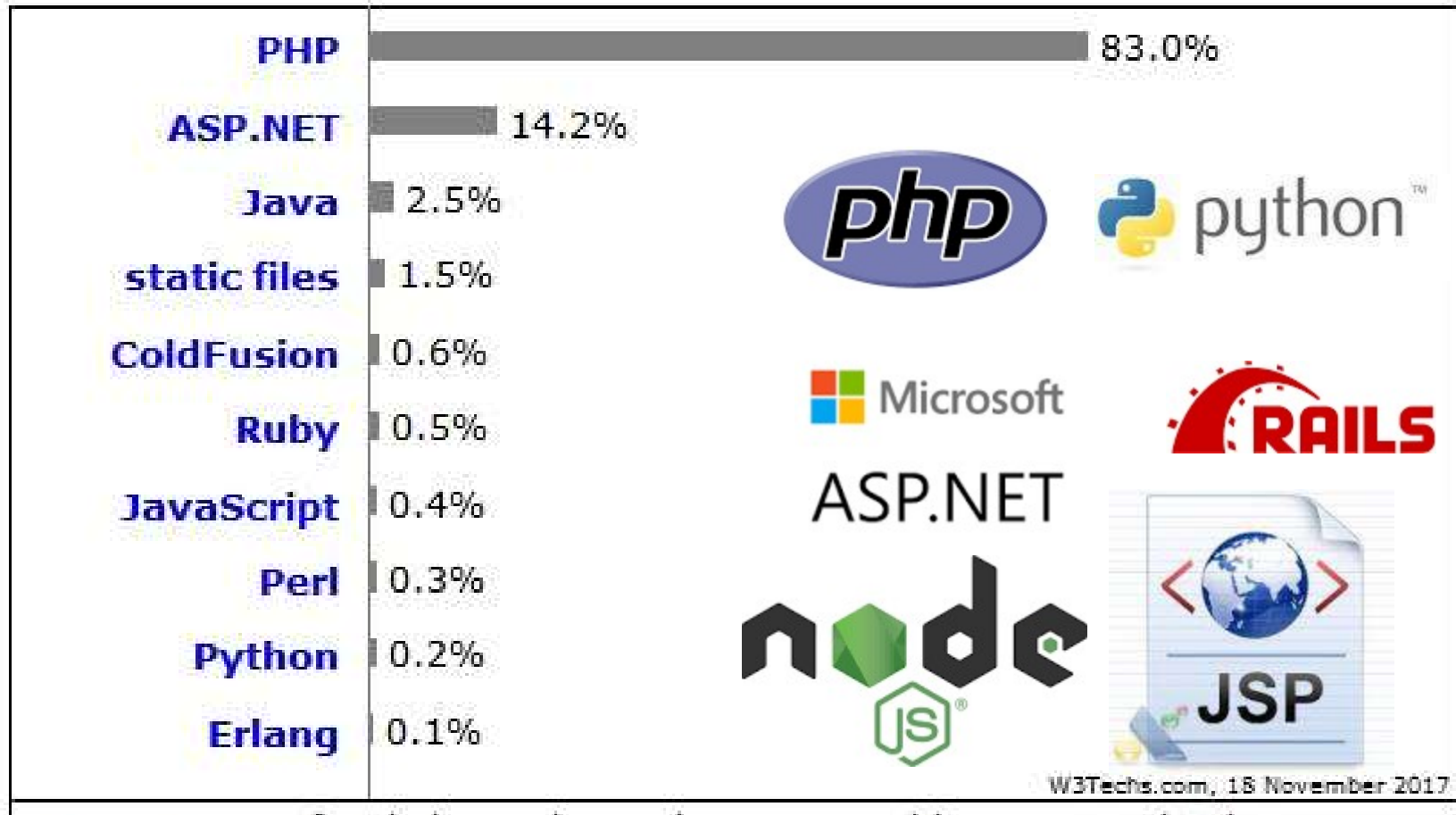
W3Techs.com, 18 November 2017

JavaScript Libraries

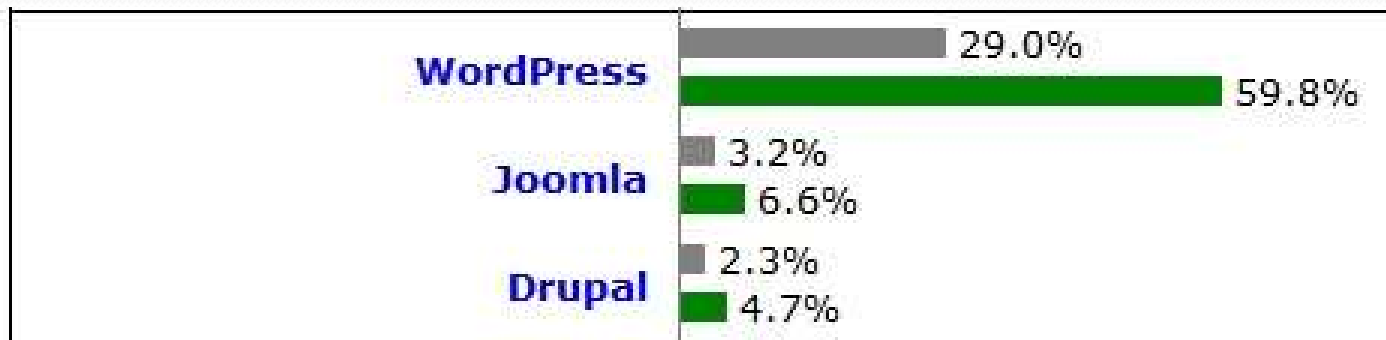


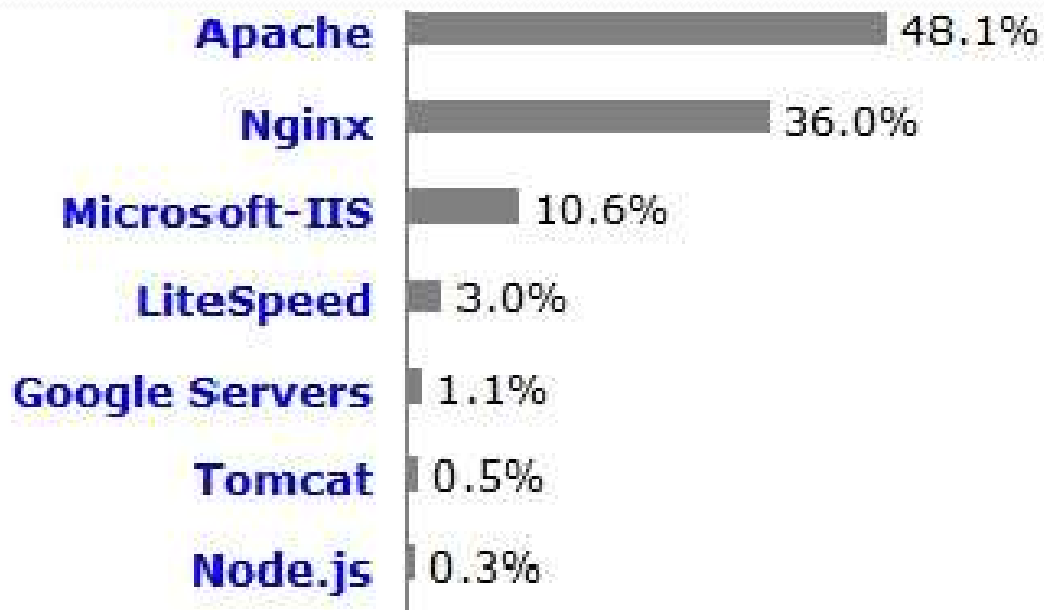
Comparison of Server-Side scripts

[As per w3techs.com]



Comparison (CMS)





Apache

NGINX



Tools for Website development



Search Engine Optimization (SEO)

- Search Engines – Crawl, Index, Process, Calculate relevancy, Retrieve Results
- designing and developing a website to rank well in search engine results.
- improving the volume and quality of traffic to a website from search engines.
- marketing by understanding how search algorithms work, and what human visitors might search.



SEO

- Keep the web page filename short, simple, descriptive, and relevant to the page content.
- Try to use a maximum of 3-4 keywords in your filename, and these keywords should appear on your web page title as well.
- Separate all keywords with hyphen rather than with underscore.
- Keep your sub-directories name as short as possible.
- Restrict the file size to less than 101K because Google chops almost everything above that.

SEO

- You should have more text content than HTML elements.
- No frames.
- No JavaScript. If you need JavaScript, call it from an external file rather than dumping the code in the HTML file.
- Keywords in the `<title>` tag(s), `<meta name="description">`, `<meta name="keyword">`, `<h1>`,
- `keywords` link tags, body copy, alt tags, `<!-- insert comments here>`, comments tags. URL or website address.



Examples of Web Applications

- Examples of Web applications include:
 - Reservation systems
 - Weblogs
 - Massively-Multiplayer Online Role-Playing Game (MMORPG)
 - Online shopping
 - Online auction
 - Games
 - Multimedia applications
 - Calendars
 - Maps
 - Chat applications
 - Clocks
 - Interactive design applications
 - Stock tickers
 - Currency converters
 - Data entry/display systems



Nature of Web Applications

- Web applications:
 - have features and benefits of desktop applications
 - have some form of programmatic control either on the client side, or on the server, or both
 - emphasize on real data separation as opposed to markup/style separation
 - are usually smaller in file size than desktop applications
 - can have rich graphical-user interfaces (GUI)
 - have reduced client-requirements
 - have portable data



Building Web Applications

- Two major components needed to build web applications include:
 - **Hardware platforms** – could be a single shared server running on a web server and a database
 - **Software platforms**
 - **Schema** – for data storage
 - **Business rule (logic)** – for accessing and modifying data
 - **Interactive logic** – for presenting data to users

References-1

- <https://www.w3.org/>
- <https://cloud.google.com/appengine/>
- <https://w3techs.com/sites/info/irctc.co.in>
- https://www.owasp.org/index.php/Main_Page
- <http://ietf.org/>
- https://en.wikipedia.org/wiki/Quirks_mode
- <http://www.bestwebsiteinindia.com/state/tamil-nadu-sites.html>
- <https://www.agriya.com/>
- <https://W3schools.com>

References-2

- <http://php.net/manual/en/session.examples.basic.php>
- <https://www.tutorialspoint.com/seo/>
- <https://www.researchgate.net/>
- <https://docs.oracle.com/javaee/5/tutorial/doc/geysj.html>
- <https://msdn.microsoft.com/en-us/library/ee658099.aspx>
- <https://msdn.microsoft.com/en-us/magazine/dn463786.aspx>
- <https://nodejs.org/en/>
- <https://www.owasp.org>

References - 3

- <https://cordova.apache.org/docs/en/latest/guide/cli/index.html>
- <https://auth0.com/blog/converting-your-web-app-to-mobile/>
- <https://cmf.gov.in/>
- <https://www.w3.org/>
- <https://whatwg.org/>



COVERAGE

- HTML
- CSS
- JAVASCRIPT
- PHP
- QUERYING A DATABASE
- CLIENT SERVER COMMUNICATION
- WEB APP DEVELOPMENT

LAB requirements

1. XAMPP on Windows / LAMP on Ubuntu
2. Visual Studio Code / Notepad++ / Bluefish

Evaluation Pattern: C-L-T-P-EL-C pattern: 5 -1 -0- 4 -3- 4

Sl. no	Category of Courses	Continuous Assessment	Mid – Semester Assessment	End Semester
1.	Practical Integrated with Theory	40(P)	20(T)	40(P)

CA: Every lab has a spot question with exercises: 25 Marks
Observation-exercises-spot (5-15-5)



FIRST STEPS

- Familiarize with HTML tags, attributes, elements
- Practice Block level elements
- Creating Hyperlinks
- Embedding Content
- Creating TABLES
- Using DOM
- HTML events
- FORMS

HTML template

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>template-1</title>
<meta charset="UTF-8">
<link rel="stylesheet" href="style1.css" title="big text">
<script src="support.js"></script>
</head>
<body><h5>document.write()</h5><ul><li>Familiarize with HTML tags, attributes, elements
<li>Creating Hyperlinks</li>
<li>Embedding Content</li>
<li>Creating TABLES</li>
<li>Using DOM</li>
<li>HTML events</li></ul>
<p id="one" style="font-family:verdana"><a href="www.google.com" target="_blank">www.google.com</a></p>

<button onclick='alert("check template")'>button click</button>
<script>document.getElementById("one").style.font-family="arial";
</script>

<table border="2">
<tr>
<td align="center">HEAD section</td>
<td align="center">BODY section</td>
</tr>
<tr>
<td align="center">meta, scripts, styles</td>
<td align="center">rendered content</td></tr>
</table> </body>
</body> </html>
```


HTML DOM

