# LECTURE 1: INTRO TO WEB DEVELOPMENT

- **Objectives:**

  **Basic Web Application Model**
  **Web Development Frameworks/Languages**

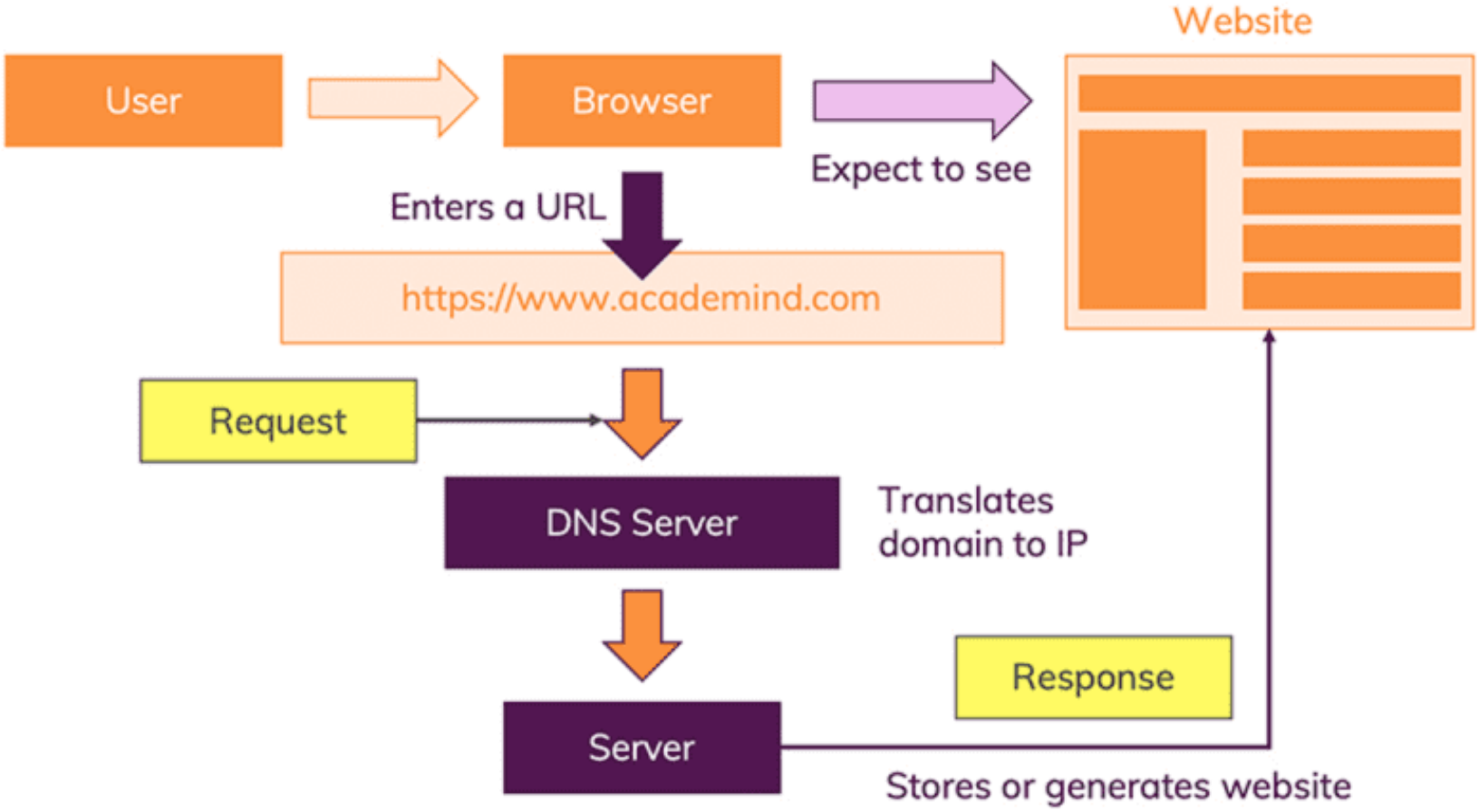
# HOW WEB WORKS

# HOW WEB WORKS

**UI (Front End (DOM, Framework))**

**Request Layer (Web API)**

**Back End (Database, Logic)**



# STEP 1: URL GET RESOLVED

**What is a server ?**

- The website code is obviously not stored on your machine and hence needs to be fetched from another computer where it is stored. This "other computer" is called a "server". Because it serves some purpose, in our case, it serves the website.
- Each server has its IP Address.
- Ipconfig( show your system IP Address)
- An IP address typically looks like this: **172.56.180.5**

# STEP 1: URL GET RESOLVED(CONTINUE)

- Protocol is an agreement between client and server for data transfer
- HTTP(Hypertext Transfer Protocol)
- HTTPS(Hypertext Transfer Protocol Secure)(encrypted data)
- FTP(File Transfer Protocol).

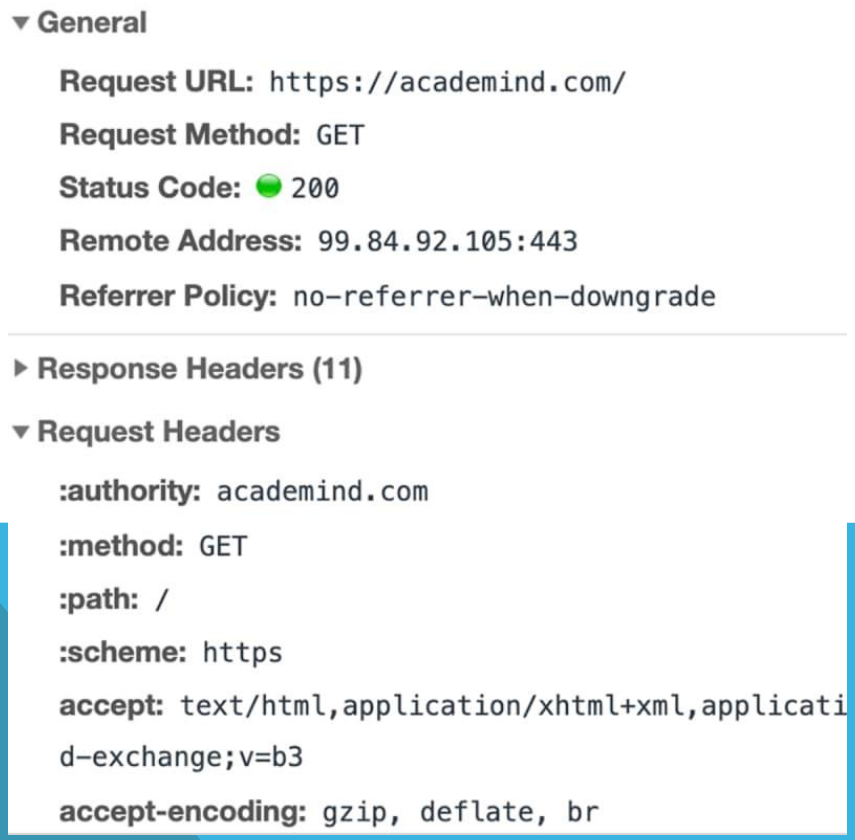
# STEP 1: URL GET RESOLVED(CONTINUE)

**How is the domain “invictussolutions.co" translated to its IP address?**

- There's a special type of server out there in the internet. A so called "name server" or "DNS server" (where DNS = "Domain Name System").
- The job of these DNS servers is to translate domains to IP addresses.
- huge dictionaries that store translation tables: **Domain => IP address**.

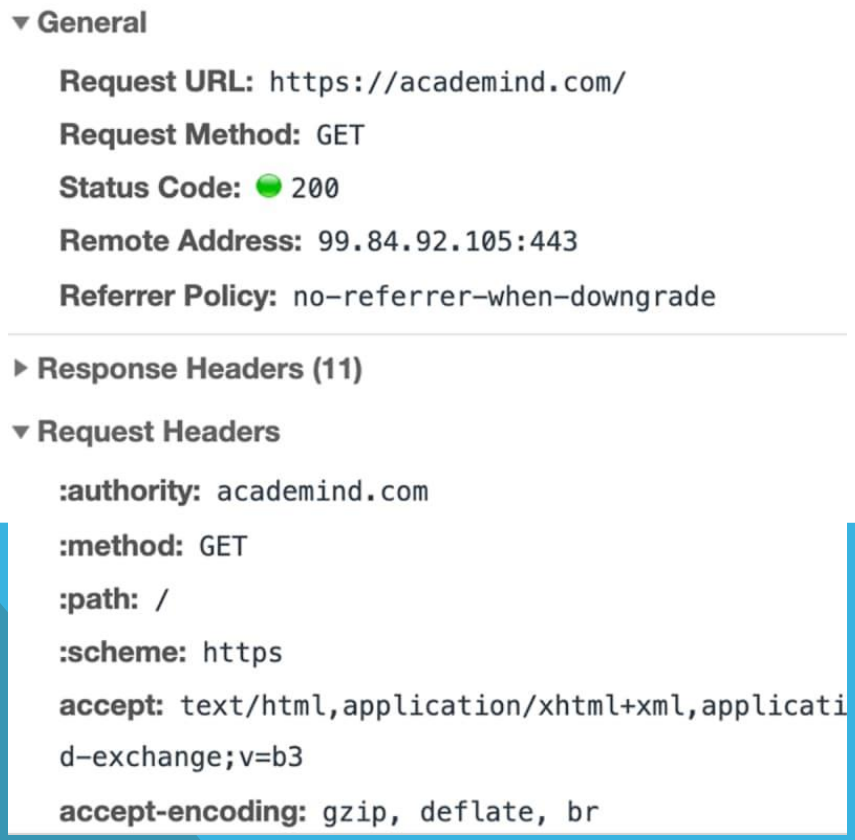
# STEP 2 - REQUEST IS SENT

- With the IP address resolved, the browser goes ahead and makes a request to the server with that IP address.
- A “”**request**" is not just a term. It really is a technical thing that happens behind the scenes.



# STEP 3 - RESPONSE IS PARSED

- With the IP address resolved, the browser goes ahead and makes a request to the server with that IP address.
- A “”**request**" is not just a term. It really is a technical thing that happens behind the scenes.

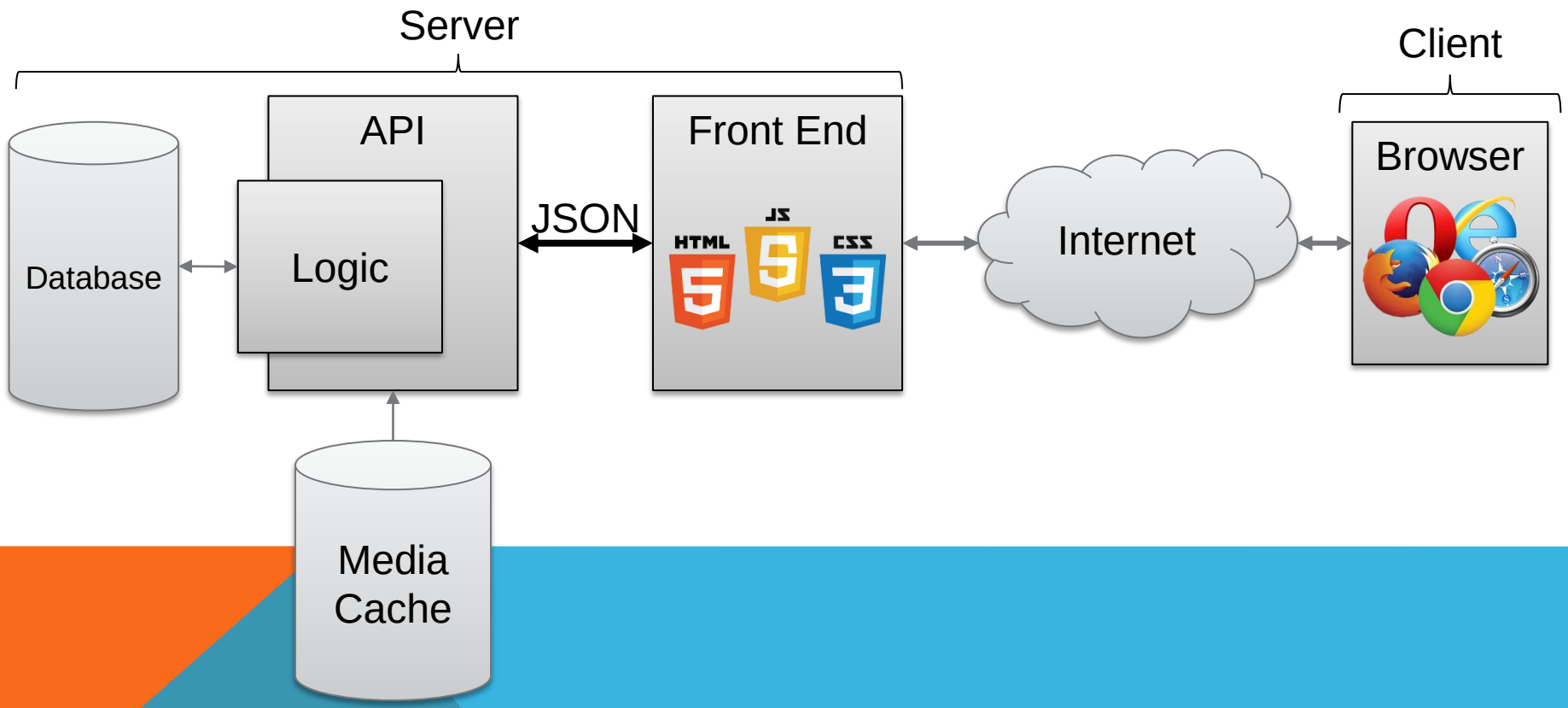


# CORE COMPONENTS OF WEB APPLICATIONS

**UI (Front End (DOM, Framework))**

**Request Layer (Web API)**

**Back End (Database, Logic)**



# SSR VS SPA

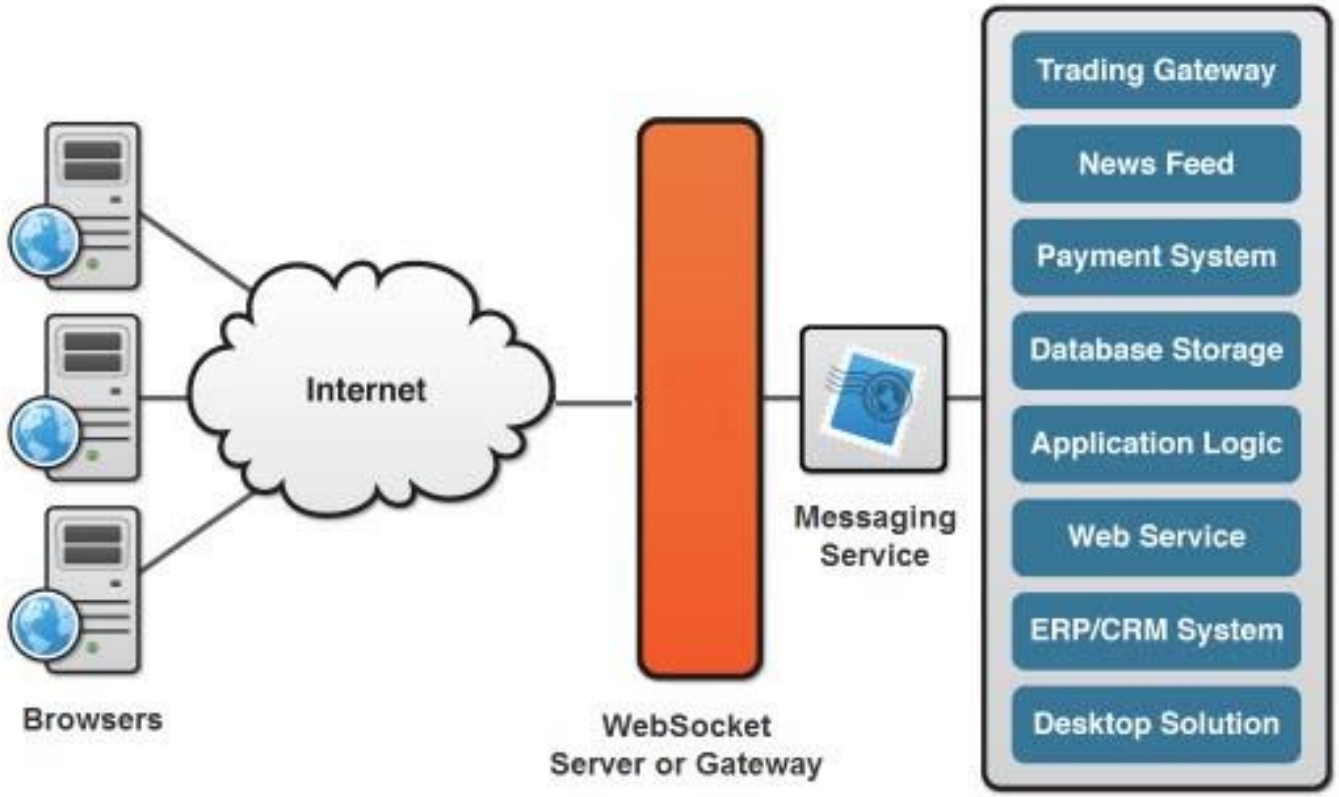
**SPA(Single Page Application)**

- Not Refresh After Reload
- Do not support SEO
- Do not show html in source code (Ctrl + U )

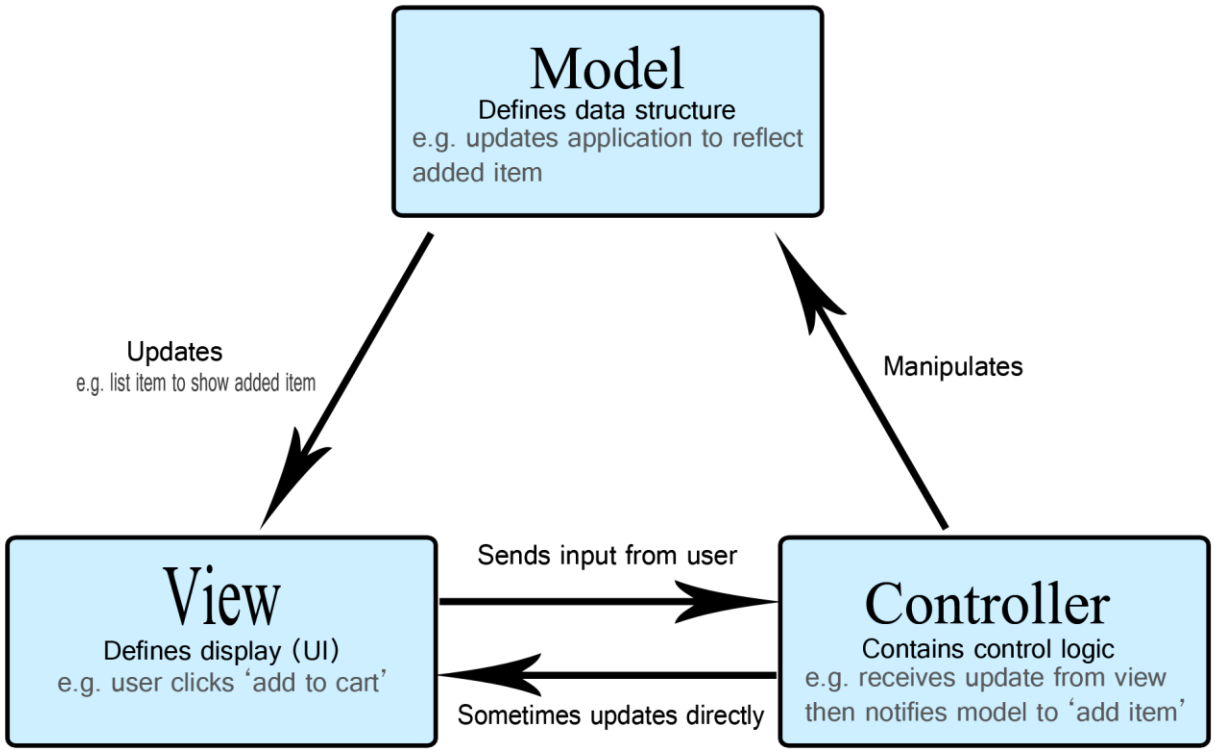
**SSR(Server Side Rendering)**

- Generate HTML on server
- Better For SEO
- Show HTML in source code(Ctrl + U)

# WEB SOCKETS


Trading Gateway
News Feed
Payment System
Database Storage
Application Logic
Web Service
ERP/CRM System
Desktop Solution
Internet
Messaging Service
Browsers
WebSocket Server or Gateway

# MVC


Model
Defines data structure
e.g. updates application to reflect added item
Updates
e.g. list item to show added item
Manipulates
View
Defines display (UI)
e.g. user clicks 'add to cart'
Sends input from user
Sometimes updates directly
Controller
Contains control logic
e.g. receives update from view then notifies model to 'add item'

# WEB VERSIONS

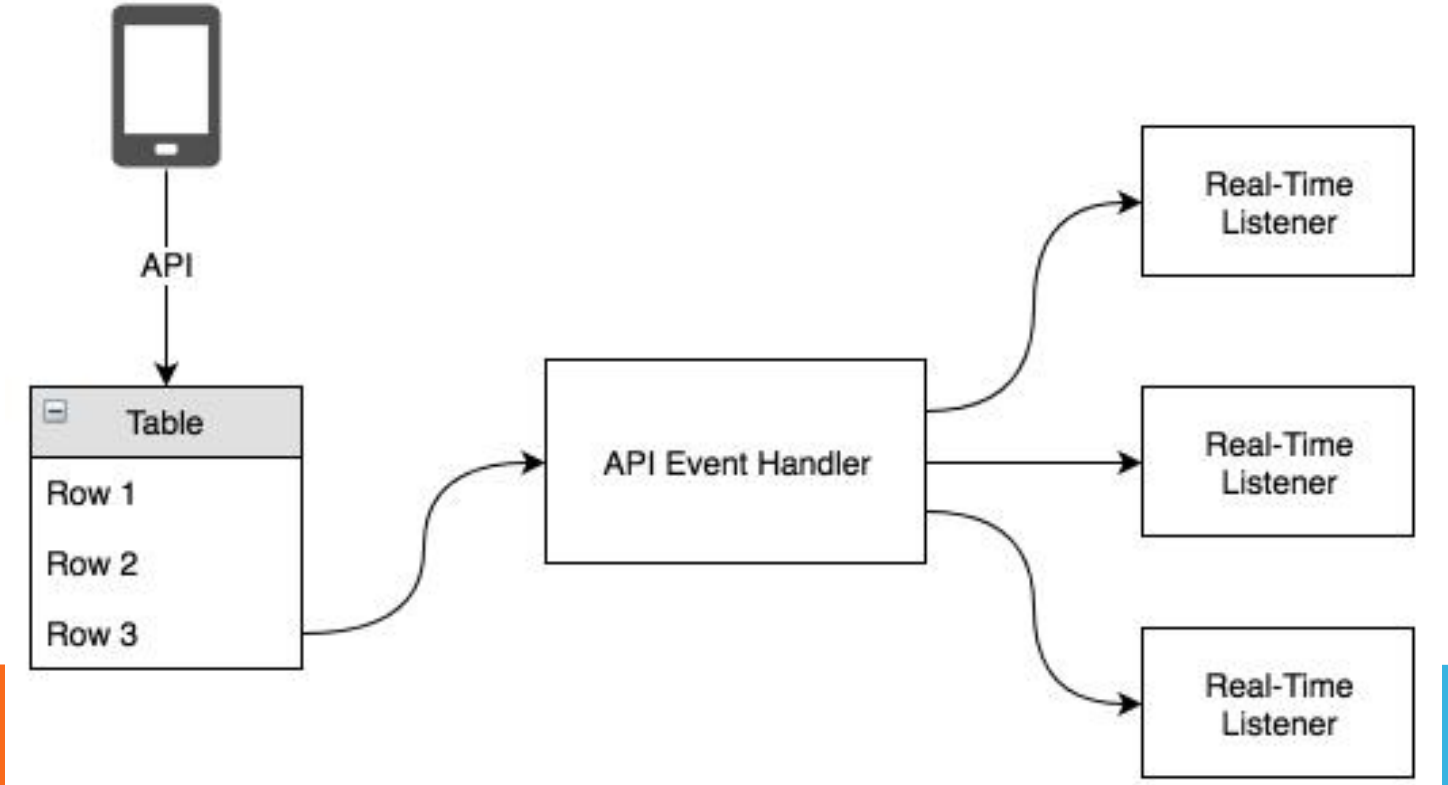
**Web 1.0:** Read

**Web 2.0:** Read + Write

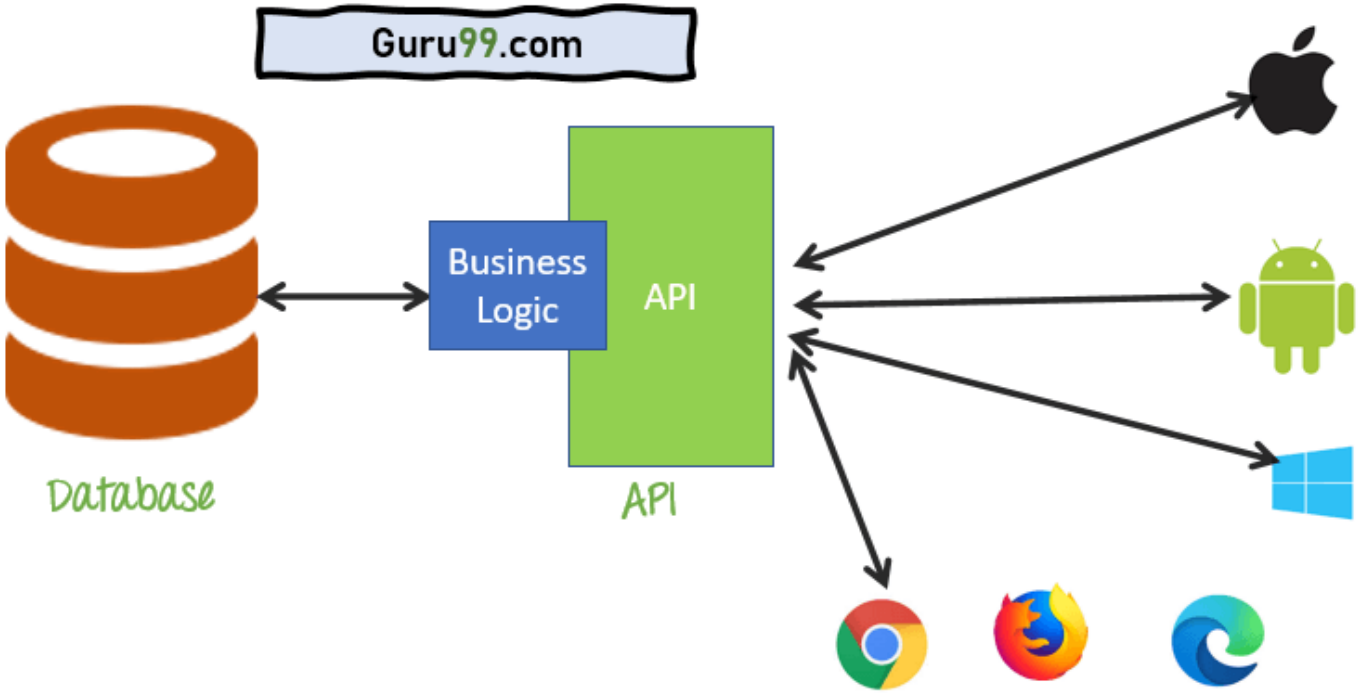
**Web 3.0:** Read + Write + Ownership

# EVENT LISTENERS

**Example:** User Signup Event activate send welcome email listener



# APIS



# FRONTEND DEVELOPMENT

# FRONT END LANGUAGES

**HTML/CSS**

**Javascript**

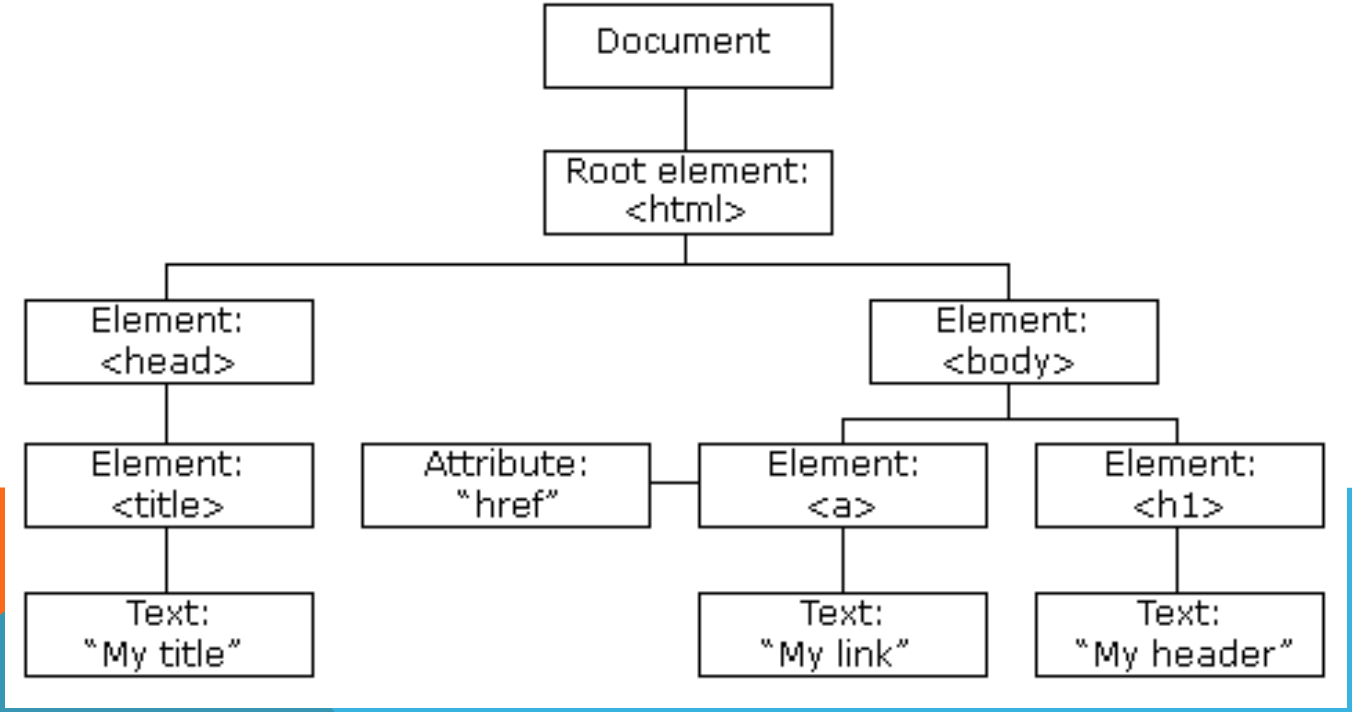
**What is the most popular?**

**Answer: Javascript/HTML/CSS is the only real option for front-end native languages and is basically the standard. But there are many variations on JavaScript that are used.**

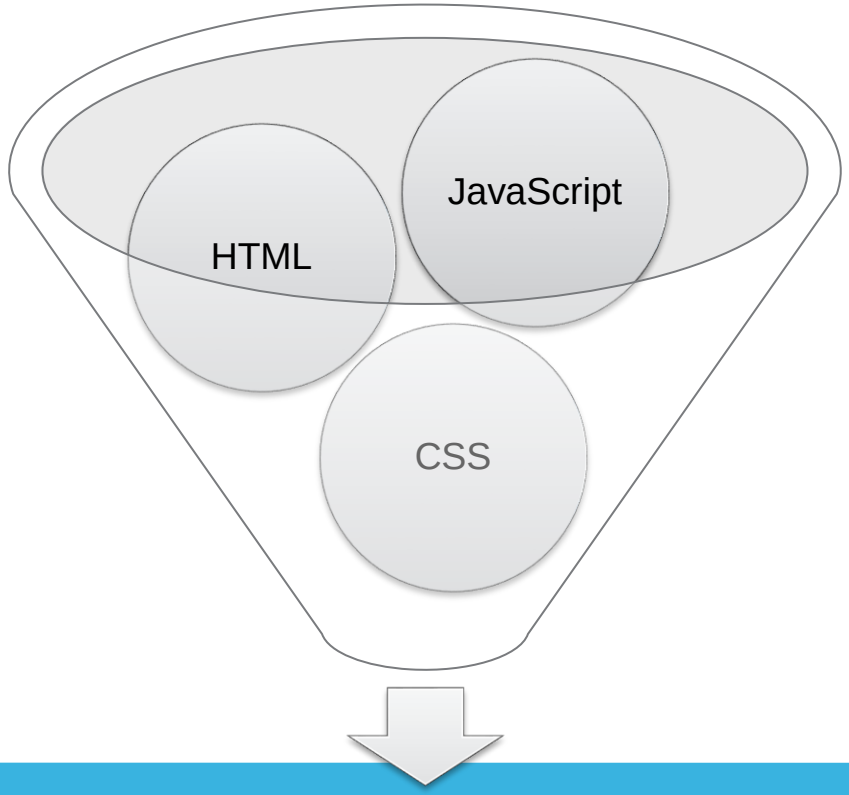
# DOM (DOCUMENT OBJECT MODEL)

**Document Object Model makes every addressable item in a web application an Object that can be manipulated for color, transparency, position, sound and behaviors.**

**Every HTML Tag is a DOM object**



# DOM (DOCUMENT OBJECT MODEL)




DOM

# WHAT IS A FRAMEWORK?

**Software Framework designed to reduce overhead in web development**

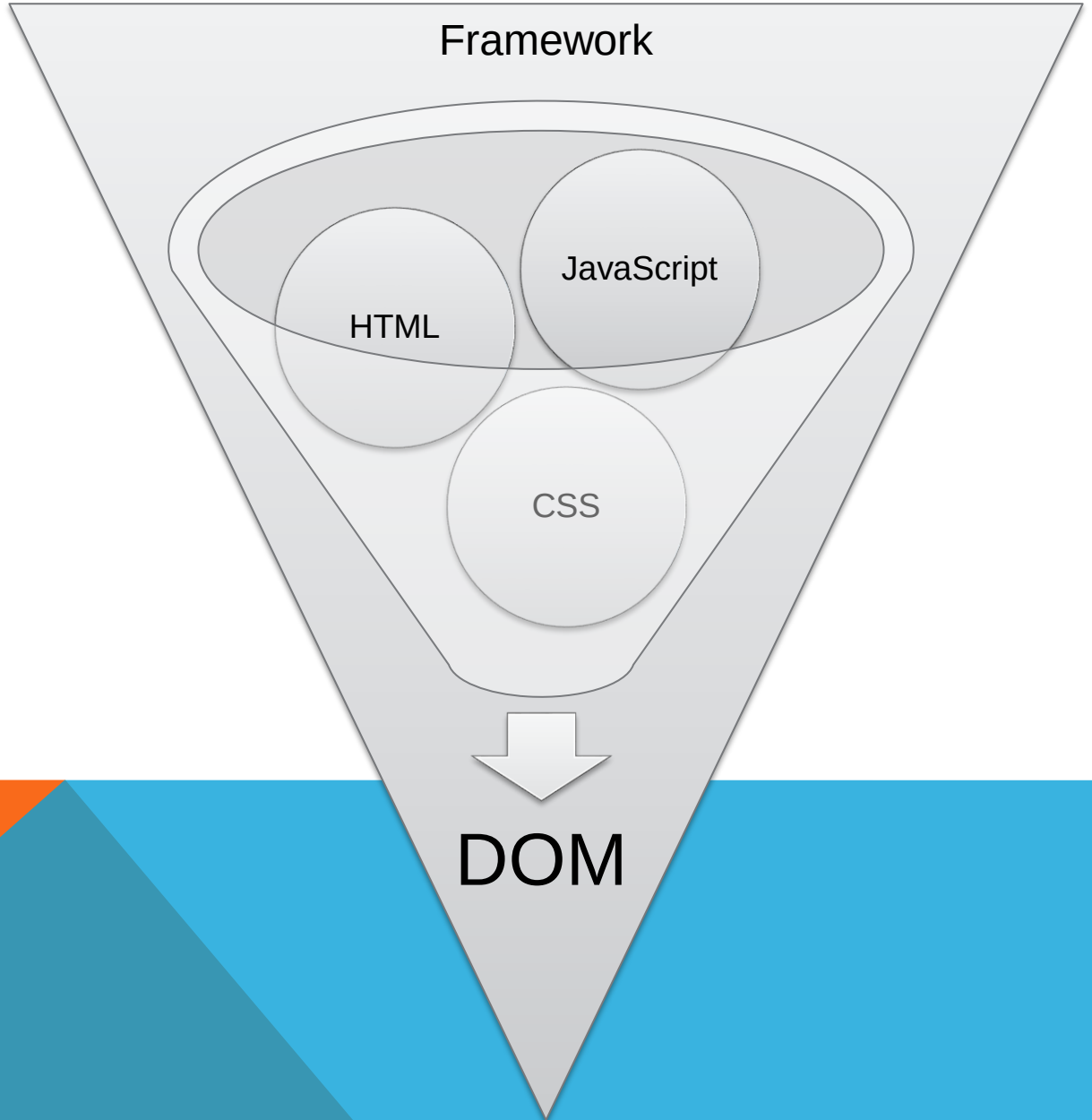
**Types of Framework Architectures**

- Model-View-Controller (MVC)
- Push vs Pull Based
  - Most MVC Frameworks user push-based architecture “action based” (Django, Laravel, Symfony)
  - Pull-based or “component based” (Lift, Angular2, React)
- Three Tier Organization
  - Client (Usually the browser running HTML/Javascipt/CSS)
  - Application (Running the Business Logic)
  - Database (Data Storage)

**Types of Frameworks**

- Server Side: Django, Laravel
- Client Side: Angular, React, Vue

# FRAMEWORK



http://hotframeworks.com

# JAVASCRIPT FRAMEWORKS

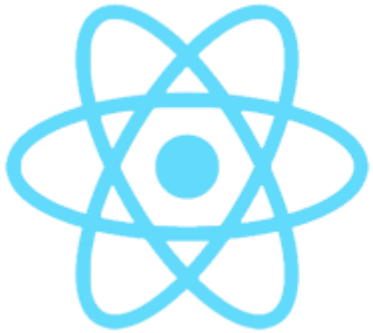
**Angular**

**React**

**Polymer 1.0**

**Ember.js**

**Vue.js**



React

ember

2.0 NOW IN BETA!