

# Web Application Development

---

Produced  
by

David Drohan ([ddrohan@wit.ie](mailto:ddrohan@wit.ie))

Department of Computing & Mathematics  
Waterford Institute of Technology

<http://www.wit.ie>



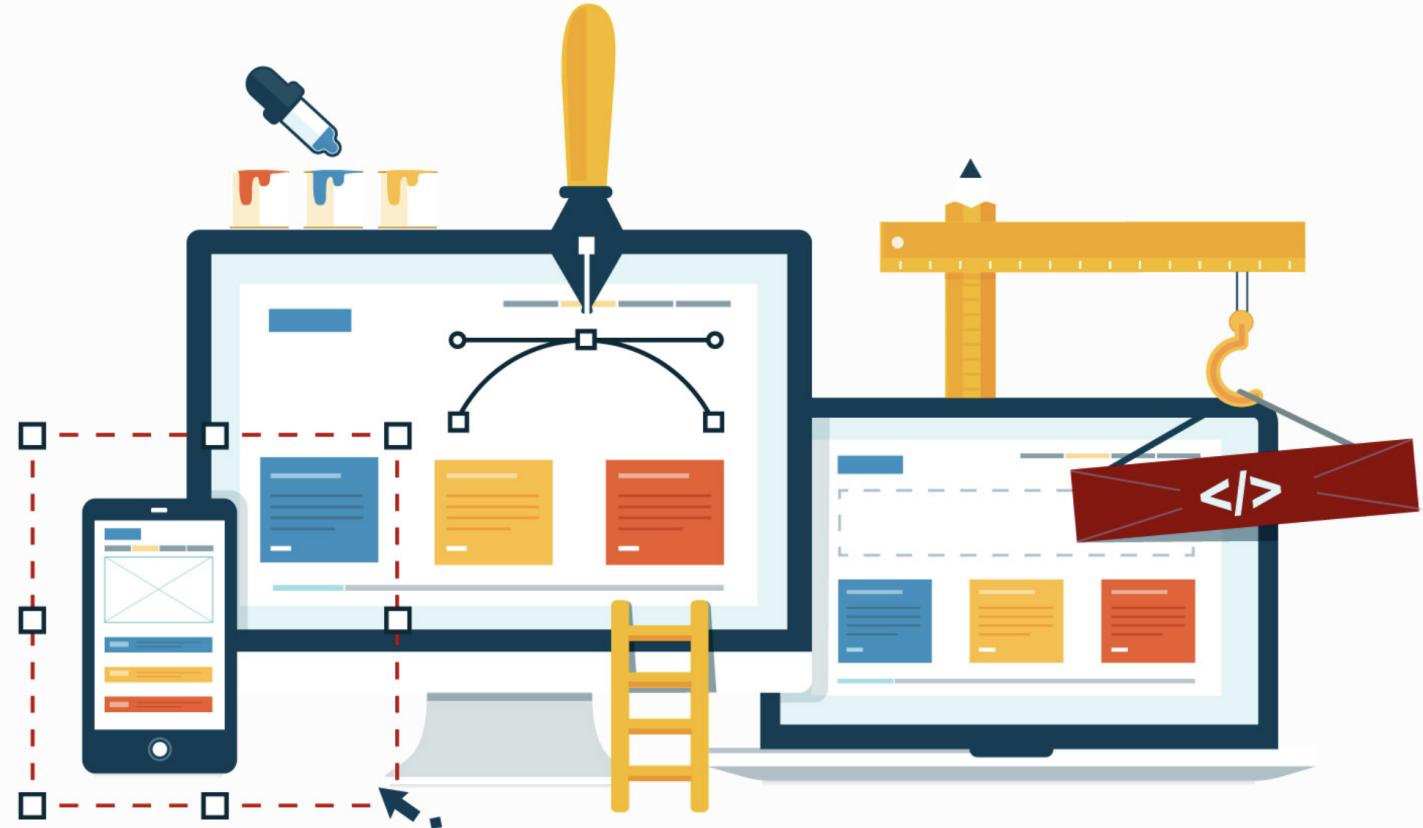
Waterford Institute of Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE



# Web Apps, Dev Tools & Frameworks

---

Data-centric applications in  
the Browser & How to Build them.

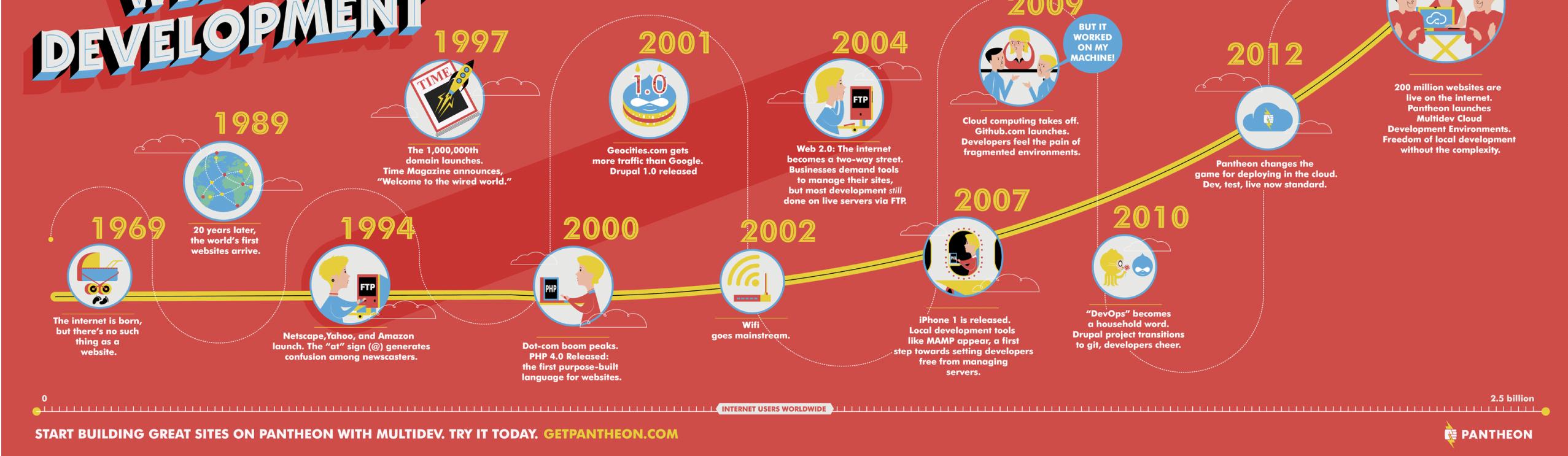




# Agenda

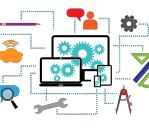
- ❑ Brief History of Web Site/App Evolution
  - ❑ Early Web Apps & AJAX
  - ❑ Stages of Web Development
  - ❑ Development Tools & Frameworks
  - ❑ Single Page Application Frameworks (SPAs)
  - ❑ Next Generation Development

# The history of **WEBSITE DEVELOPMENT**



# A brief history of Web Development

<https://pantheon.io/blog/history-website-development-infographic>



# Early Web Apps

---

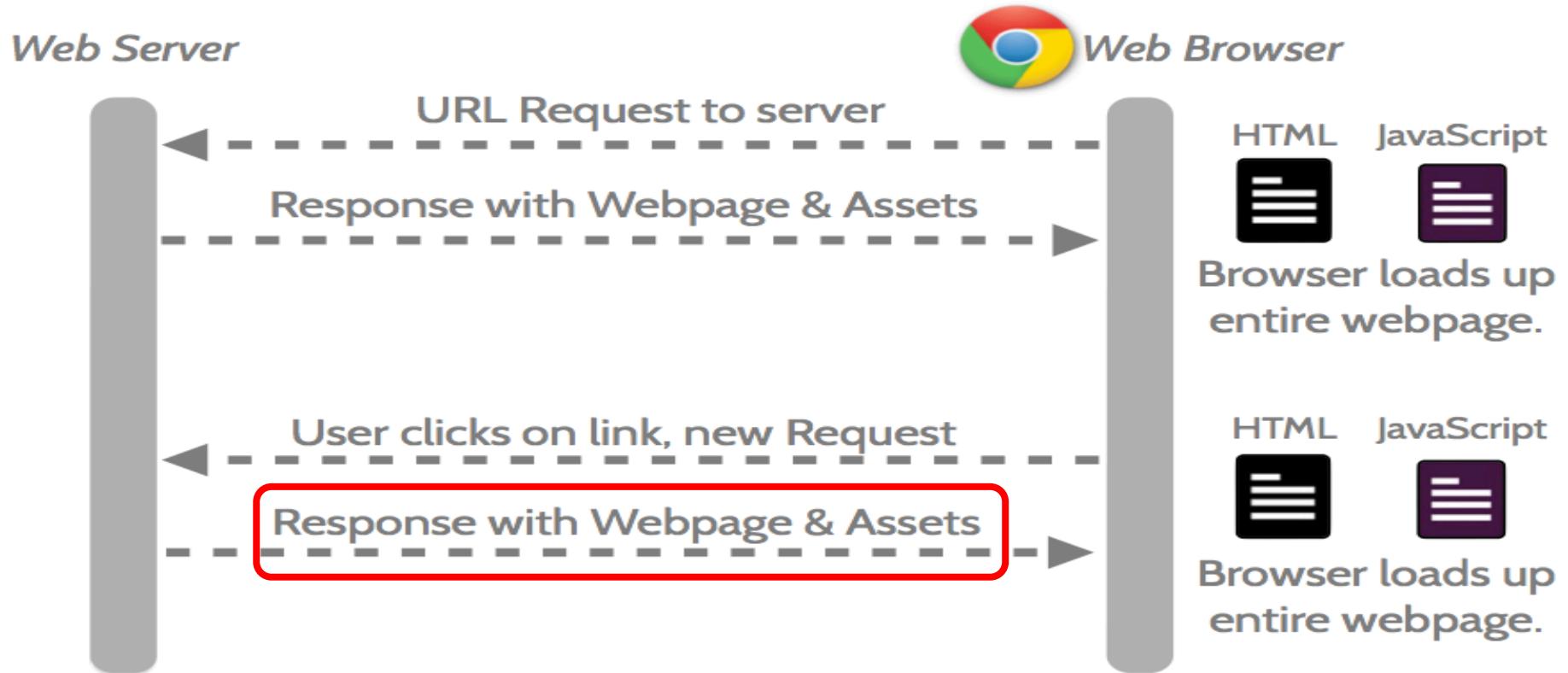
## ❑ Characteristics:

- Server creates pages / browser displays.
- Data input sent to, and processed by, the server
- Updated pages created on the server and resent to browser.

## ❑ For Example - PHP, JSP/Servlets, Struts, (and more recently) Ruby on Rails.



# Early Web Apps – Request/Response cycle



## ❑ Disadvantages:

- Bad UX (User eXperience) – reload of all UI parts
- Poor performance



# Web Evolution – AJAX (Asynchronous Javascript And Xml)





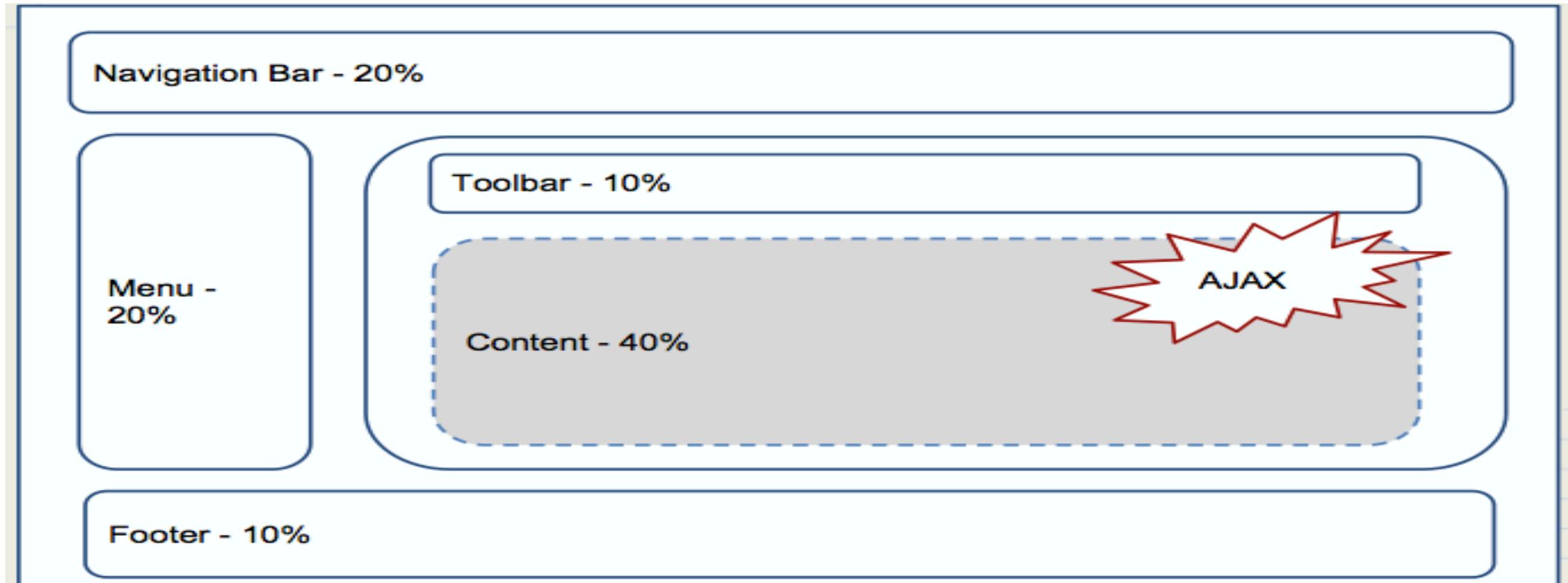
# What is AJAX?

---

- ❑ AJAX = Asynchronous JavaScript and XML.
- ❑ Not a language but a technique.
- ❑ A technique for loading data (formatted as XML) in the background (asynchronous) and displaying it on the webpage, without reloading the whole page.
  - JSON formatting is now favoured over XML
- ❑ Examples: Gmail, Google Maps, Youtube.



# Web Evolution - AJAX



Static

Dynamic



# Web Apps - AJAX

---

## ❑ Interactive client-side web:

- Collect input from user.
- Update display.
- Communicate with server (AJAX)

## ❑ Client-side processing enabled by:

- JavaScript.
- DOM manipulation.
- HTTP server messaging.



# Web Evolution – SPA Frameworks

---

- SPA (Single Page Application) frameworks – Client-side JavaScript MVC frameworks. (MVC covered below)
  - Examples:
    - ◆ VueJS, AngularJS; Backbone; EmberJS; React, Knockout, .... many more ...
- Benefits:
  - Less boilerplate code.
  - Less effort on mundane programming tasks; allowing for better focus on what is ACTUALLY VALUABLE – THE LOGIC.
  - More efficiency in development.
  - Better client-side code architecture through clearer Separation of Concerns (SoC).
- All are built on core browser functionality – event-driven, asynchronous ; DOM manipulation; HTTP server communication



# Web Evolution – SPA frameworks

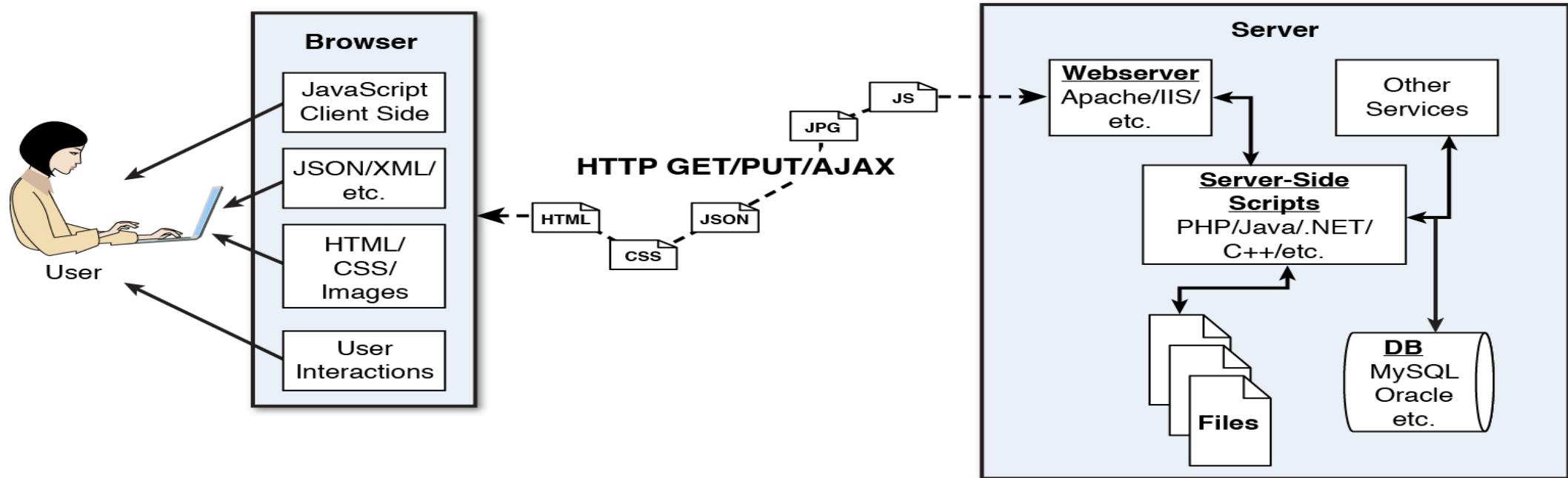
---

## □ Library Vs Framework

- Library (e.g. JQuery)
  - ◆ Passive functionality; Invoked by the application.
- Framework (e.g. VueJS)
  - ◆ Provides application architecture (SoC); Deals with common mundane requirements; Invokes application code.

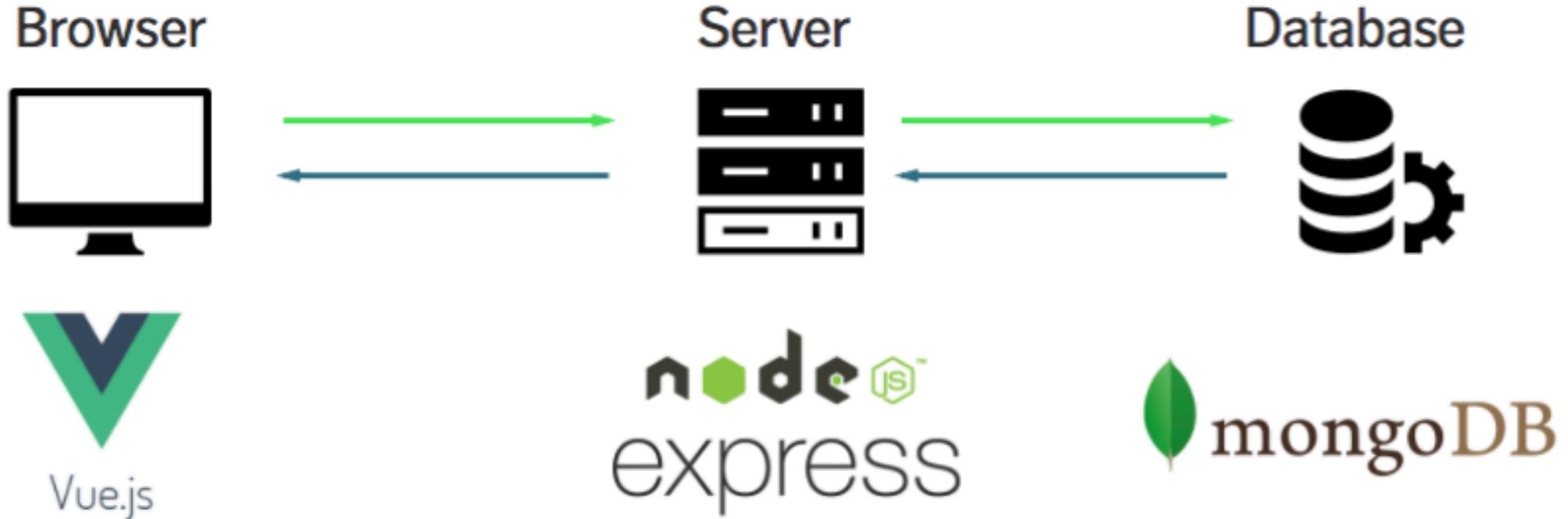


## Components of a **Classic / Basic** Web App (Pre SPAs)





# Components of a **MEVN** Web App (SPA) \*



All Javascript !



# Web Development Stages

## 1 – Concept Development

Write down the following:

- Who is this website/app for?  
(the types of users)
- Why am I making it?  
(what is the problem?)
- What is the **most important** thing visitors  
should do on your website/app.
- Write, read. Write, read. Repeat.
- Sketch out (draw) your website/app 'look'.





# Web Development Stages

## 2 – Prototype Development

❑ A “broken” version of your website:

- Links work, but don’t lead to any content.
- “User Journey” is complete:
  - “As a ... I want to ... ”
- Quick to change and move things around if and when needed.

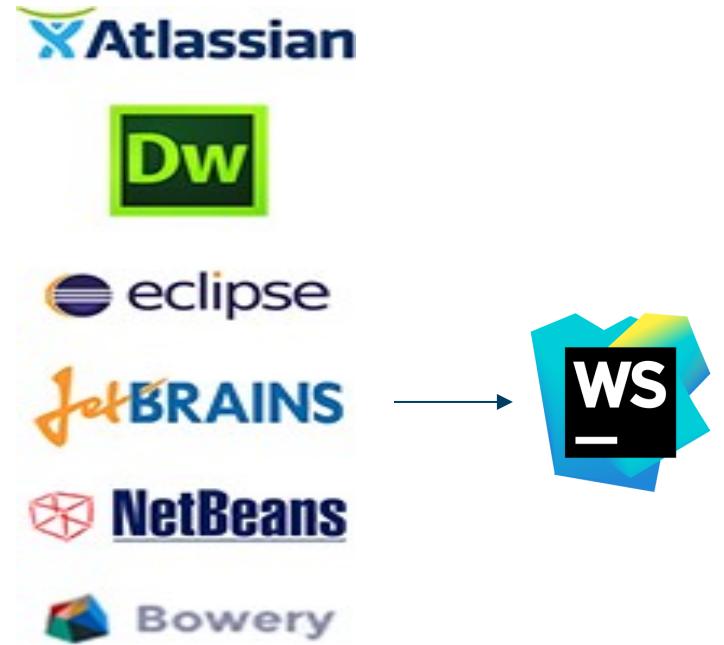
Presentation software	Mobile apps	Mobile prototyping applications		
gliffy	AppCooker	Hotspot apps	Wireframe / Mockup tools	Prototyping tools
Keynote	iMockups	APP IN SECONDS	balsamiq	AppSketcher
THE OMNI GROUP	Interface 2	FIELDTEST	mockups	axure
Visio	mockapp	invision	pidoco*	Indigo STUDIO
		POP	SOFTANDGUI	proto.io



# Web Development Stages

## 3 – Actual Development

- Ideally, your requirements are complete
- Work at an iterative pace:
  - Develop at the lowest possible work unit
- Use the tool that suits you, and makes you the most productive:
  - Text Editor
  - IDE
  - Terminal
  - GUI





# Web Development Stages

## 4 – Testing

- If you don't test; prepare to fail.
- Types of testing:
  - Functional Test (does the feature work?)
  - User acceptance test (is it what the user wants?)
  - Unit test (is the code stable?)
  - Coverage tests (are all code paths executing?)



**Sinon.JS**

Standalone test spies, stubs and mocks for JavaScript.  
No dependencies, works with any unit testing framework.





# Web Development Stages

## 5 – Deployment

### □ Deployment Process Should Be:

- Documented
- Repeatable
- Predictable
- Rehearsed
- Tools help!





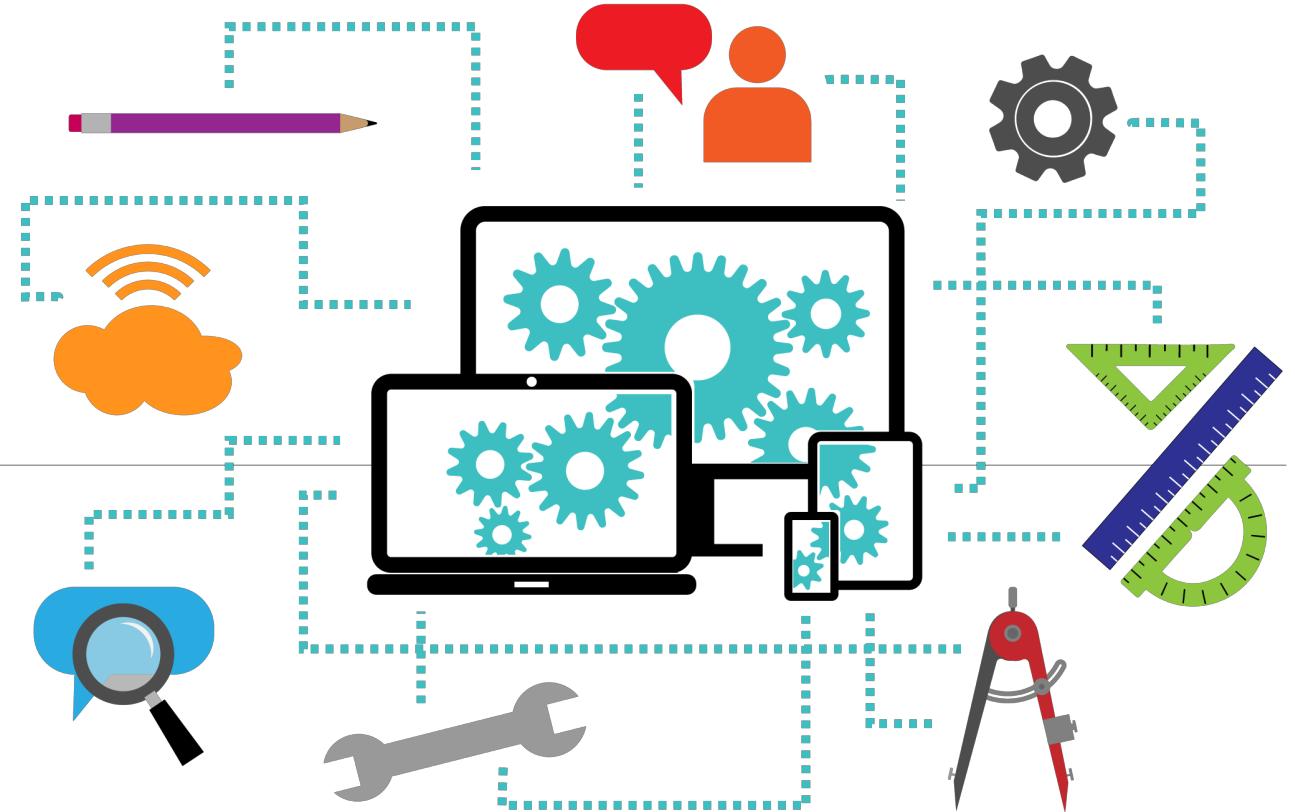
# Web Development Stages

## 6 – Maintenance

- ❑ Small effort, big rewards
- ❑ Enable maintenance by tracking:
  - Successful requests
  - Failed requested
  - Repeated failed requests
  - Key function execution
  - Infrastructure



# Development Tools & Frameworks









# Categories of Tools & Frameworks:

# *Server Side Development*

# django





# Categories of Tools & Frameworks: \*

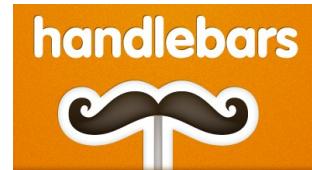
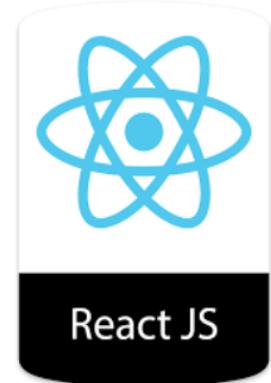
## *Server Side Development*





# Categories of Tools & Frameworks: \*

## *Frontend Development*





# Categories of Tools & Frameworks: \*

## *Frontend Development*





# Categories of Tools & Frameworks:

## Platforms



SQUARESPACE



OPENSIFT



tumblr.





# Categories of Tools & Frameworks:

## *Platforms*





# Categories of Tools & Frameworks: \*

## *Development Helpers*



Bower



GRUNT



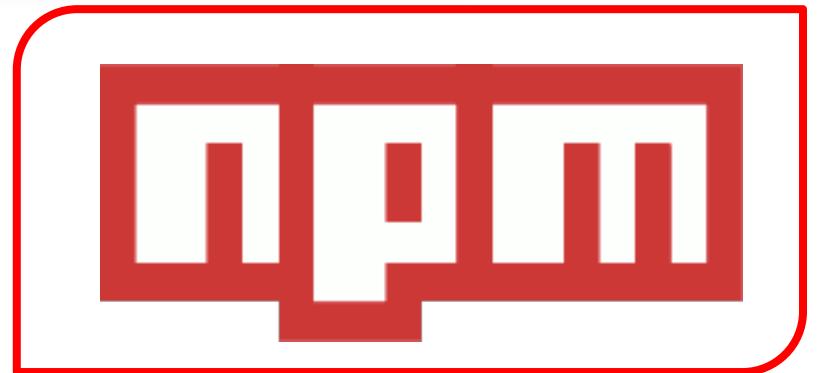
YEOMAN





# Categories of Tools & Frameworks: \*

## *Development Helpers*



# THE FRONT-END SPECTRUM





# RECAP : How Does it All Work?

---

THE DETAILS

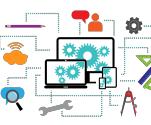


---

# The Internet

---

A LARGE NETWORK OF COMPUTERS, CONNECTED THROUGH A COMMON NETWORK SPREAD ACROSS A LARGE AREA.



---

# World Wide Web

---

AN INFORMATION NETWORK  
BUILT ON TOP OF THE INTERNET USING HYPERLINKED DOCUMENTS.



---

# HTML & HTML5

---

HYPertext Markup Language. A language consisting of a series of tags used to annotate documents that are to be linked together on the web. The latest version is 5



---

# CLIENT SIDE

---

A TERM USED TO DESCRIBE TECHNOLOGY THAT RUNS ON THE DEVICE CONNECTED TO THE INTERNET AND REQUESTING DOCUMENTS FROM A SERVER.

EXAMPLES: JAVASCRIPT, FLASH, VBSCRIPT



---

# SERVER SIDE

---

A TERM USED TO DESCRIBE TECHNOLOGY THAT RUNS ON THE SERVERS CONNECTED TO THE INTERNET PROVIDING ONE OR MORE SERVICES.



---

# USER AGENT

---

THE TECHNICAL TERM FOR THE PIECE OF SOFTWARE THAT IS USED TO DISPLAY DOCUMENTS ON THE WORLD WIDE WEB (WWW).

EXAMPLE : GOOGLE CHROME, MICROSOFT EDGE



---

# DOCUMENT OBJECT MODEL

---

A CONVENTION OF DESCRIBING AND INTERACTING WITH HTML DOCUMENTS AS A SERIES OF NODES ON A TREE.



---

# TELNET

---

A PLAIN TEXT BI-DIRECTIONAL PROTOCOL USED TO COMMUNICATE WITH SERVERS. *IT IS ALSO THE NAME OF A PROGRAM THAT IS USED TO CONNECT TO SERVERS.*



---

# TCP/IP IPv4 IPv6

---

PROTOCOLS USED TO COMMUNICATE WITH SERVERS ON THE INTERNET. EACH DEVICE THAT CONNECTS TO INTERNET NEEDS AN IP ADDRESS.



---

# DOMAIN NAME SYSTEM

---

A DISTRIBUTED SYSTEM DESIGNED TO ASSIGN NAMES TO IP ADDRESSES, IN ORDER TO IDENTIFY RESOURCES ON A NETWORK.



# WEB SERVER

A PIECE OF SOFTWARE CODE THAT LISTENS TO CONNECTIONS AND RESPONDS WITH DOCUMENTS OR OTHER RESOURCES ON THE WORLD WIDE WEB.



---

# APACHE / IIS

# NGINX / Node

---

POPULAR WEB SERVER USED ON THE INTERNET. APACHE STILL CONTROLS THE MAJORITY OF THE MARKET.



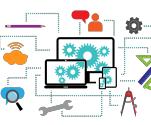
---

# HTTP

# HTTPS

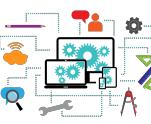
---

HYPertext Transfer Protocol. A text protocol consisting of a series of commands used to send data and request documents over the web.



# REQUEST / RESPONSE

THE NAME GIVEN TO THE COMPLETE CYCLE TO FETCH DOCUMENTS AND RESOURCES USING HTTP. THE CLIENT INITIATES THE REQUEST, AND THE SERVER SENDS THE RESPONSE.

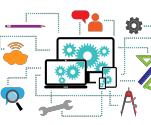


---

# FTP / SFTP

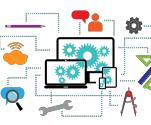
---

FILE TRANSFER PROTOCOL. A METHOD OF TRANSFERRING DATA  
BETWEEN SERVERS. SFTP IS SECURE-FTP



# SECURE SHELL (SSH)

# AN ENCRYPTED METHOD OF CONNECTING TO REMOTE COMPUTERS AND EXECUTING COMMANDS.

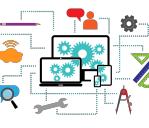


# **GIT /**

# **MERCURIAL /**

# **SUBVERSION**

POPULAR REVISION CONTROL SYSTEMS. THESE SYSTEMS TRACK CHANGES IN FILES ALLOWING EASY RECOVERY OF PREVIOUS VERSIONS.



# Classic Web Application Architecture

BASIC BUILDING BLOCKS FOR MOST WEB APPLICATIONS



---

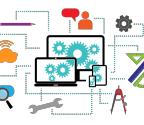
**DATABASE**  
(the models)

Server Code  
(controllers)

View  
(code to render HTML)

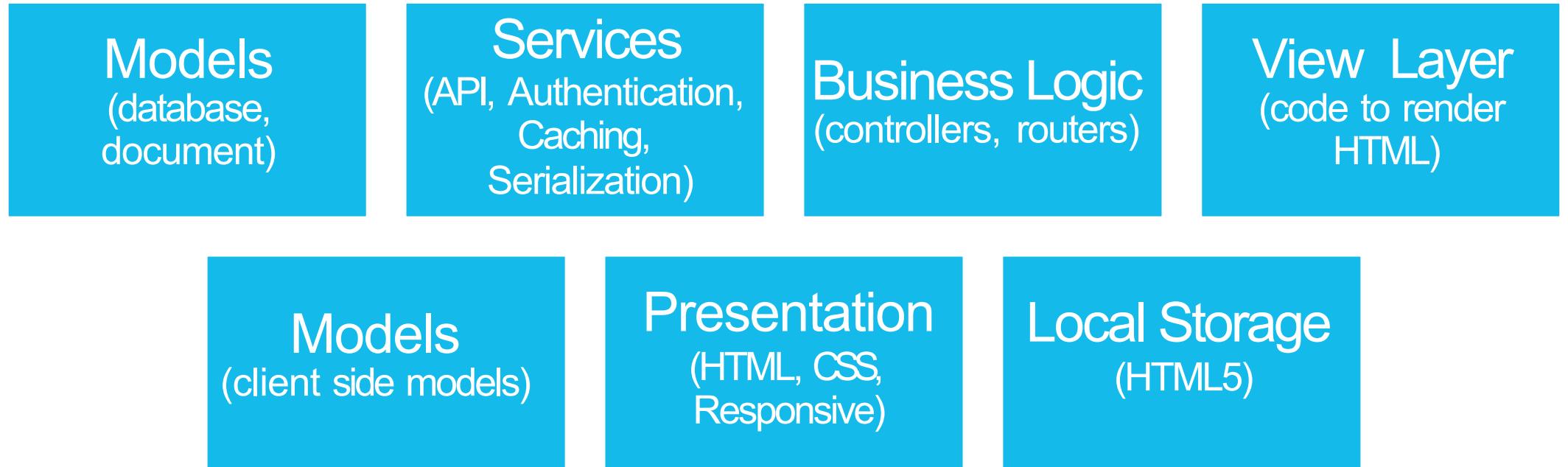
Client Side  
JavaScript  
(jquery components)

HTML Templates;  
CSS; layouts



# Modern Web Application Architecture

## ADVANCED APPLICATIONS – FOR ADVANCED DEVICES





Designed by DevNetwork.com - May 2014



# References

---

- [https://www.slideshare.net/burhankhalid/gust-webdevelopment2015apr?from\\_action=save](https://www.slideshare.net/burhankhalid/gust-webdevelopment2015apr?from_action=save)



---

# Questions?