



Full Stack Development For Startups: MERN



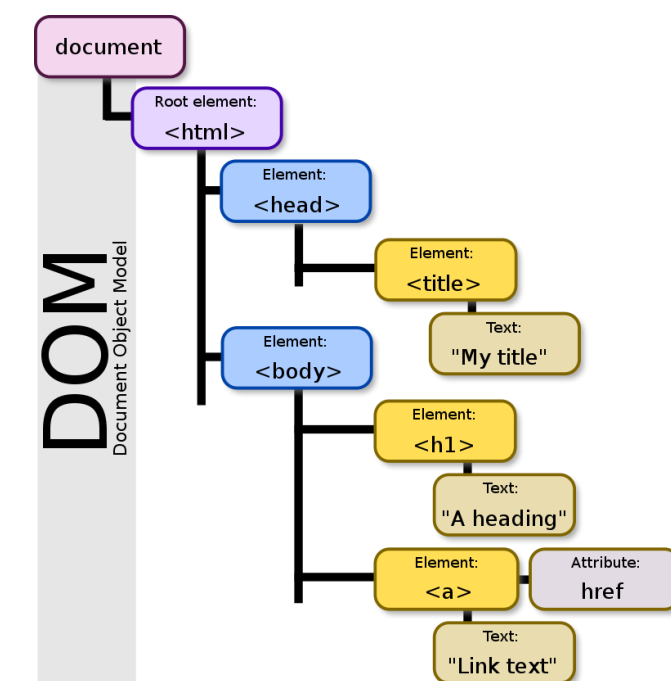
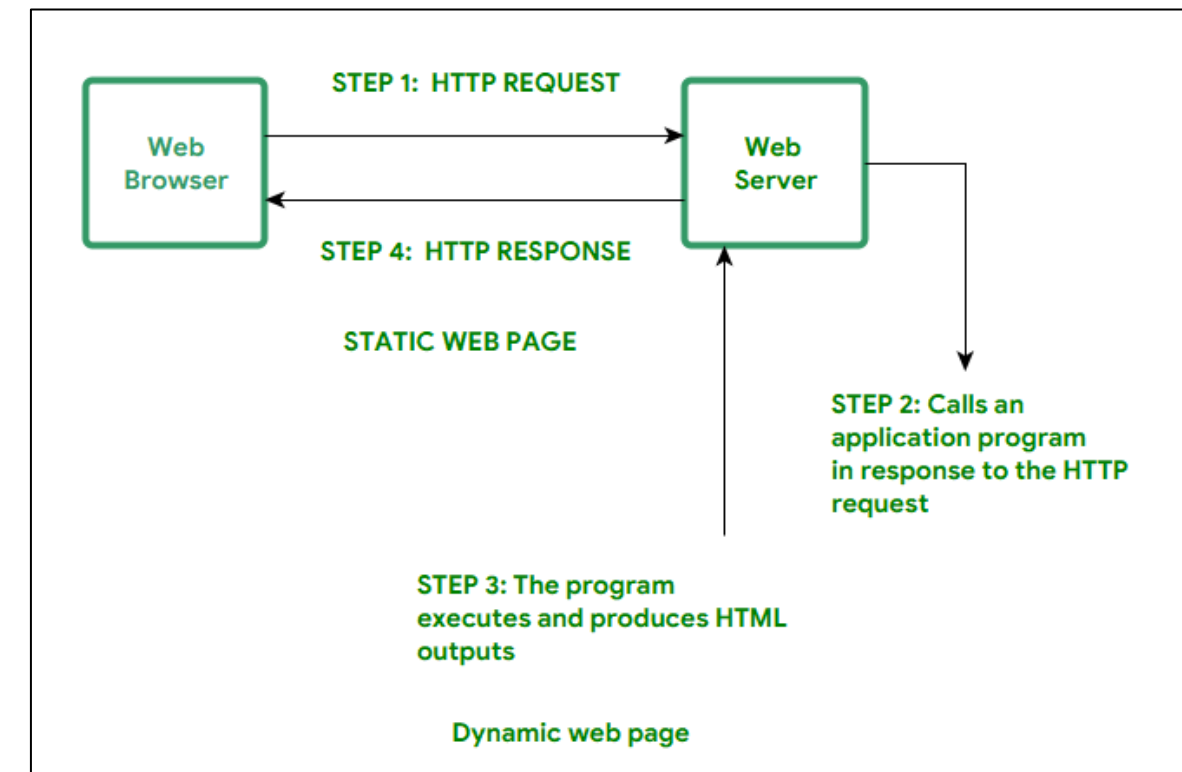
About The Assembly

- A **smart lab & makerspace** based out of **in5** since 2014
- Over **300** free workshops done
- **Assembly : HACK** - Embedded systems, IoT, hardware
- **Assembly : CODE** - Software - APIs, frameworks, apps
- **Assembly: Data Science** - Advanced topics in AI/ML
- Audience – **Students | Professionals | Entrepreneurs**
- Focus on Smart Technology & Practical Applications
- Social Media: **@makesmartthings** | members.theassembly.ae
- www.theassembly.ae -> Online Workshops (for past videos)



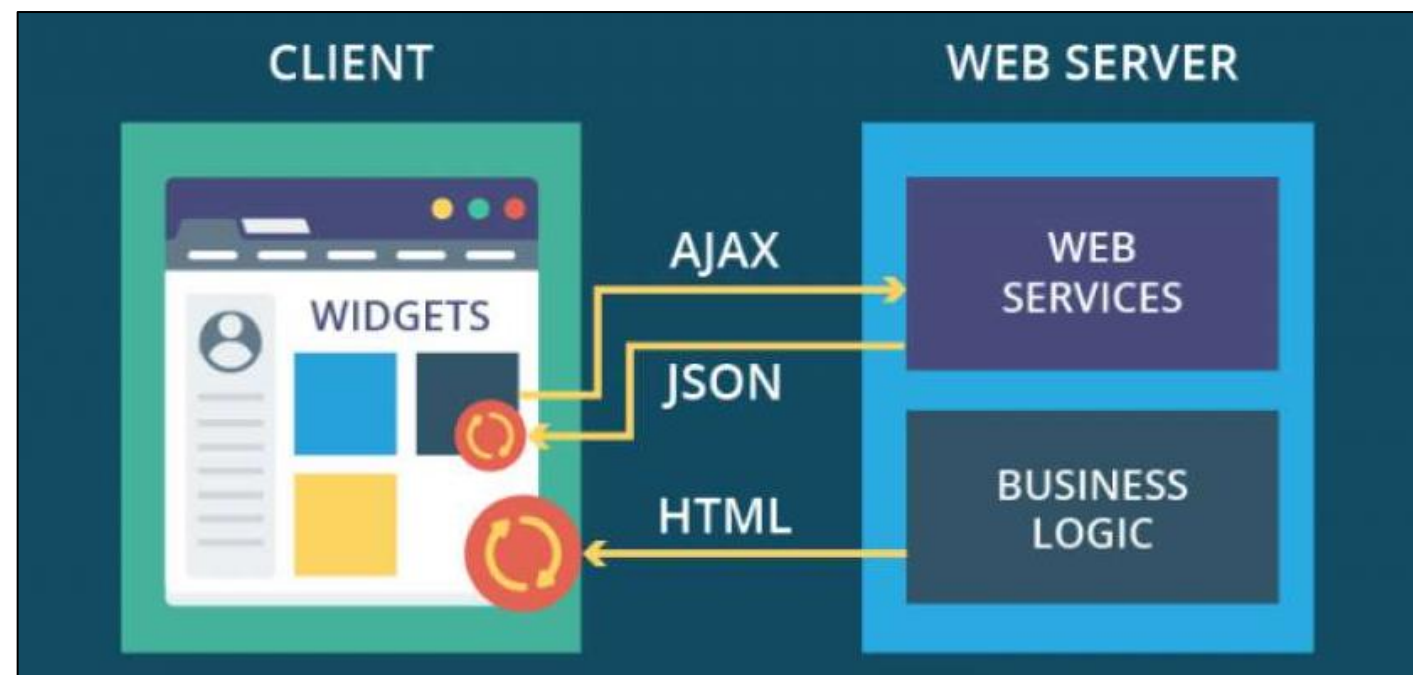
The Evolution Of Web Development

- Old way: Server does all the work
- Client is a dummy terminal – make HTTP requests, display HTML
- Server-side languages like **ASP** & **PHP** add dynamic application logic
- **1995: JavaScript** created in 10 days
- Created by Netscape – to make HTML more interactive – can manipulate DOM
- Named for similarity to Java

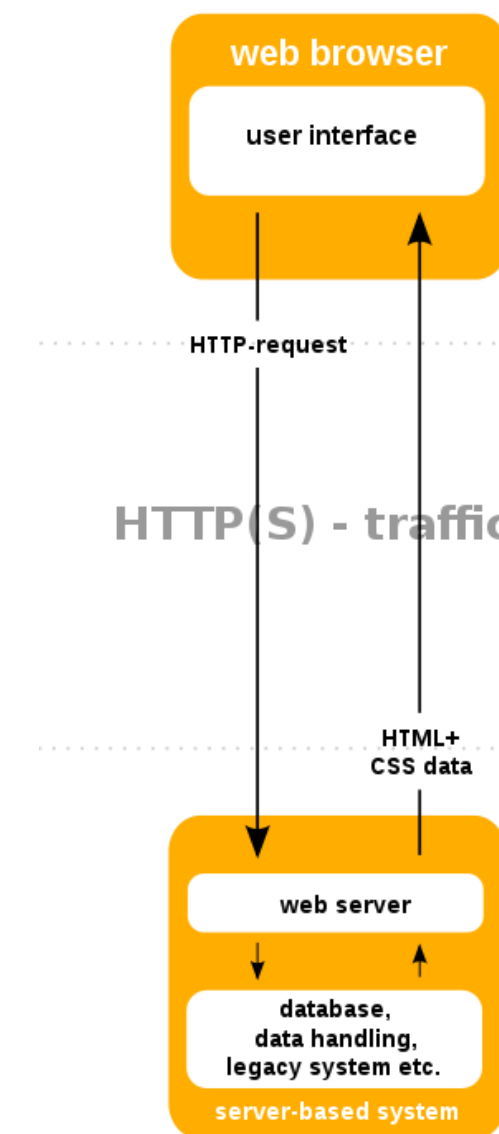


The Evolution Of Web Development

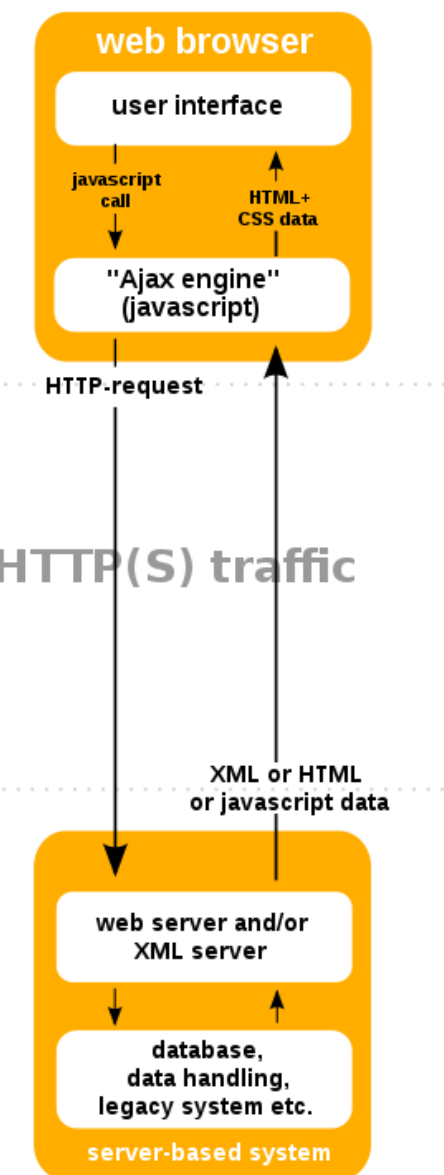
- **1999: AJAX**
Asynchronous JavaScript And XML
- Set of concepts & techniques – enable client-side sophistication
- Plugs in well with SAAS



Conventional model of a web application



Ajax model of a web application

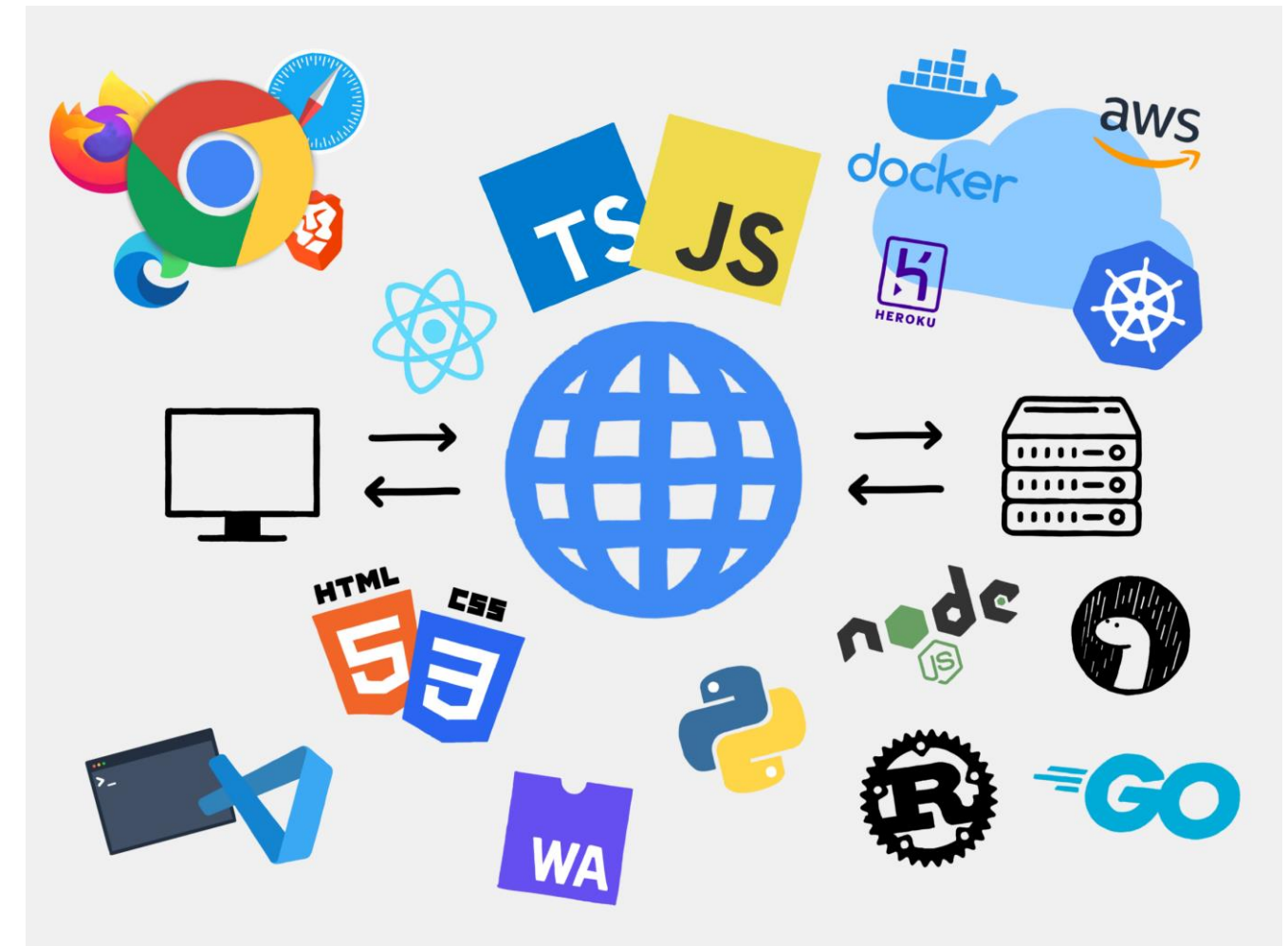


- JavaScript frameworks like **JQuery** standardize across browsers

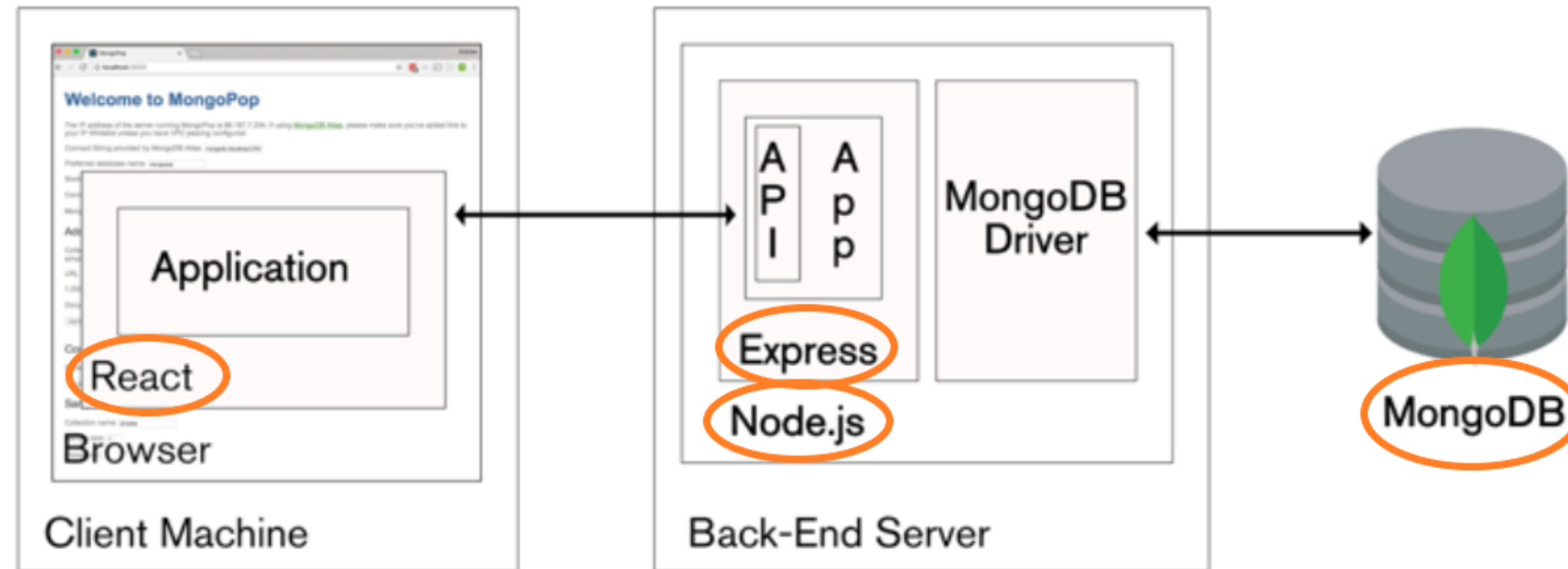


Full Stack Development

- Ability to build software at any tier - client, server, db, mobile, cloud
- With **JavaScript** now available server-side through **Node.js**, natural choice for unifying language
- Different stacks: **LAMP** (JS, Linux, Apache, MySQL, PHP), **MEAN** (JS, MongoDB, Express, Angular, Node), **Django** (JS, Python, Django, MySQL), **Ruby On Rails** (JS, Ruby, SQLite, Rails)



The MERN Stack



MongoDB Database on back-end Stores data in JSON; easily imported into JavaScript <i>(used by Uber, Lyft, DeliveryHero etc)</i>	Express Back-end app framework Runs on top of Node.js <i>(used by Twitter, Stack, Accenture etc)</i>
React JS Handles front-end Creates dynamic UI by running JavaScript in browser <i>(used by Airbnb, Netflix, Amazon, Pinterest, Facebook etc)</i>	Node.js JavaScript runtime environment for implementing back-end <i>(used by eBay, Trello, PayPal, NASA, LinkedIn etc)</i>



MERN for Startups

- Easy to develop MVPs around a single language (JavaScript)
- Ideal for agile development with quick iteration and small teams
- Open source with incredible community support
- Short learning curve – huge talent pool
- Source control and testing easy to set up
- MVC style great for adapting real world workflows
- Easy to scale up when time comes



MongoDB

- Open-source document-based NoSQL database
- Data stored in **BSON** (Binary JSON) – readymade for use by JavaScript developers
- **SQL v MongoDB :**
Table=>Collection,
Row=>Document, Join=>\$lookup,
Foreign Key=>reference etc
- **Mongoose** plugin in Node allows for easy connection

```
{  
  name: "John Doe",  
  job: "Developer",  
  address: {  
    office: "1B"  
    building: "in5"  
    city: "Dubai"  
  },  
  skills: ["MongoDB", "Node", "JS"],  
  ID: 123  
}
```



Node.js & Express

- Built on **Chrome V8 JS Engine** – open sourced by Google in 2008
- Allows for execution of JS outside of browser
- **npm** (Node package manager) – powerful tool to install third party dependencies over web
- Event-driven architecture capable of asynchronous I/O
- **Express**: minimalist framework running on top of Node.js – sets up server to handle HTTP requests



```
// Import the Express library.
const express = require('express');

// Initializing the app.
const app = express();

// Getting the path request and sending the
// response with text
app.get('/', (req, res) => { res.send('Hi, your
request has been received'); })

// Listen on port 2000
app.listen(2000, () => { console.log('listening at
http://localhost:2000'); })
```



React

- Component based declarative UI development JS library created and maintained by **Facebook** (Meta)
- Uses **JSX** – resembles HTML but is still JS
- Components have states and properties to handle data
- Rendered on to virtual DOM first
- **Create-React-App** – toolchain to create single page app
- <https://reactjs.org/docs/hello-world.html>

```
ReactDOM.render(  
  <h1>Hello, world!</h1>,  
  document.getElementById('root')  
) ;
```

```
class ButtonCounter extends React.Component {  
  constructor() {  
    super()  
    this.state = { clicks: 0 }  
    this.handleClick = this.handleClick.bind(this)  
  }  
  
  handleClick() {  
    this.setState({ clicks: this.state.clicks + 1 })  
  }  
  
  render() {  
    return (  
      <Button  
        onClick={this.handleClick}  
        text={`You've clicked me ${this.state.clicks} times!`} />  
    )  
  }  
}
```





THANK YOU!

