Server communication

→ REST → JSON | XML | YAML → design Pattern

→ SOAP → Simple Object Access Protocol → Protocol

→ GraphQL → Graph Query language ***

→ gRPC → google Remote Procedure Call → protobuff ***

→ WebSocket → Uses Socket (socket.io)

React

GET http://mydomain.in/mypage.php

mypage.php → PHP interpreter → html/css + JS

GET - Request

↓

backend

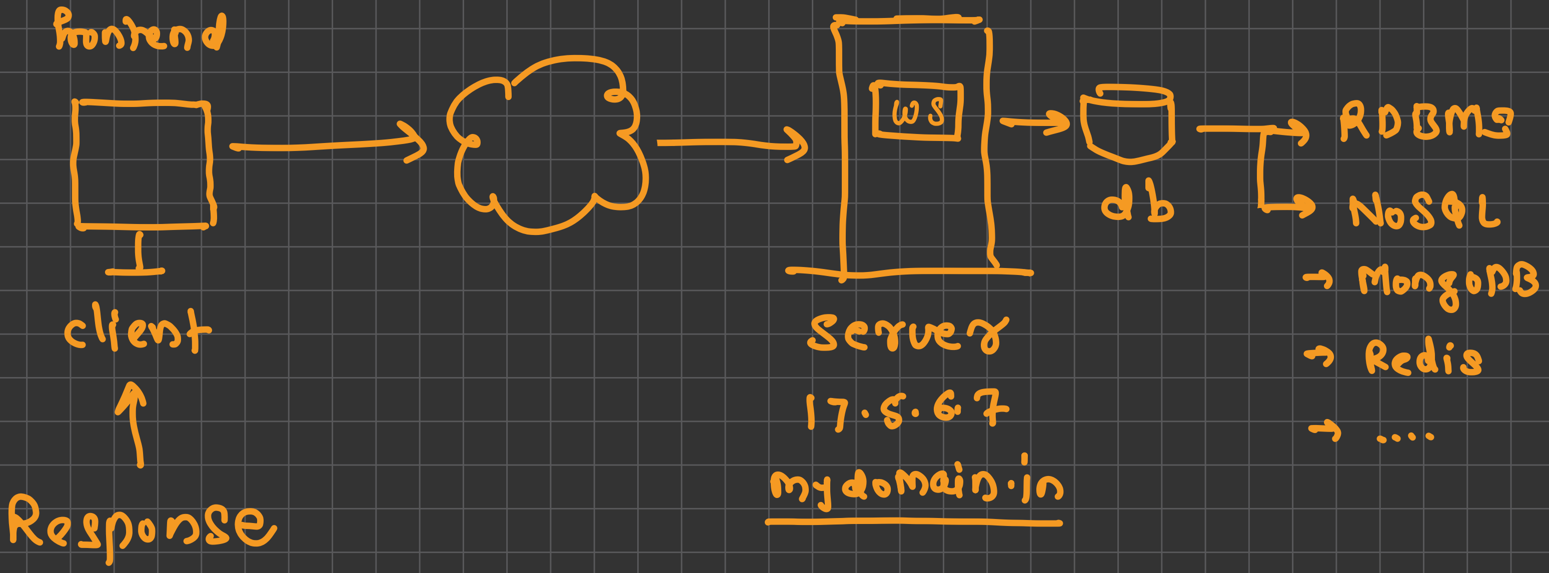
- JS → Express
- Java → Spring, SB
- PHP
- C# → .Net

designing ↑ | decoration ↑ | client side programming ↑

html + CSS + JS

- SPA
- MPA

- React → Meta
- Angular → Google
- VueJS → Alibaba



Server

→ always a software or application

→ types

1] web servers → used to serve http/https requests
   eg. apache, nginx, tomcat, express, MS IIS

2] DB server → used to persist the data
   eg. RDBMS (mysql, sql server, postgre, oracle)
   NoSQL (mongoDB, redis, cassandra, firebase)

3) DNS server → used to get IP address from a domain name
   eg. Bind

4] SMTP → used to send emails [Simple Mail Transfer Protocol]
   eg. postfix

5] POP → used to receive emails. [Post office Protocol]
   eg. Dovecot

6] File server → use to serve files
   eg. FTP → File Transfer Protocol → windows/linux/mac
   samba [SMB] - Server Message Block] → windows
   NFS → Network File System → Linux

Contents

About Instructor

- 16+ years of experience
- Associate Technical Director at Sunbeam
- Freelance Developer working in various domains using different technologies
- Developed 180+ mobile applications on iOS and Android platforms
- Developed various websites using PHP, MEAN and MERN stacks
- Languages I love and use in every programming: C, C++, Python, JavaScript, TypeScript, PHP

Golang, Rust,

Introduction

↪ less memory footprint → requires less memory

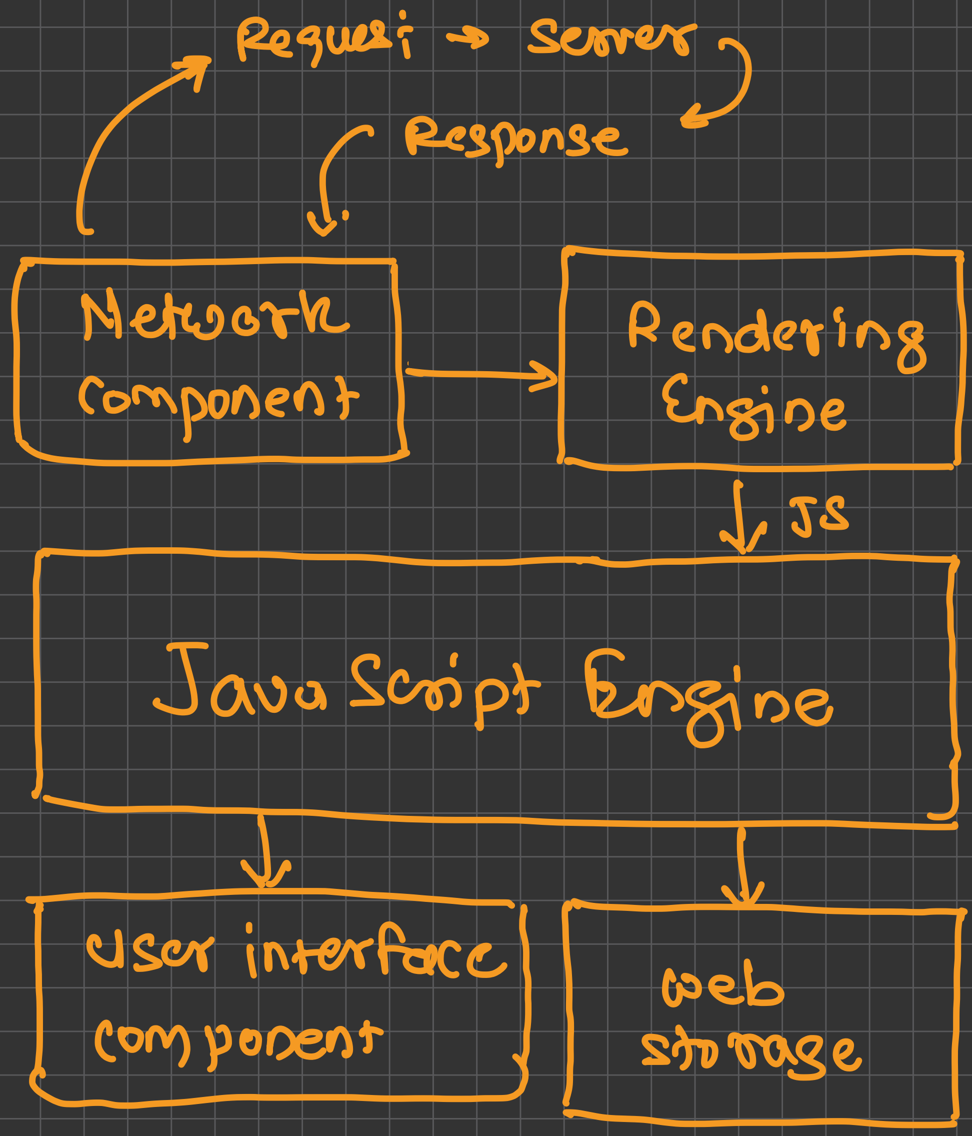
- React, also known as ReactJS, is a popular and powerful JavaScript library used for building dynamic and interactive user interfaces, primarily for single-page applications (SPAs)

↪ faster

- It was developed and maintained by Facebook and has gained significant popularity due to its efficient rendering techniques, reusable components, and active community support
- React is a declarative, component based library that allows developers to build reusable UI components and It follows the Virtual DOM (Document Object Model) approach, which optimizes rendering performance by minimizing DOM updates. React is fast and works well with other tools and libraries
- Prerequisite of React
  - For learning React first you have a clear understanding of HTML, CSS and JavaScript ✓
  - As React is a JavaScript library and uses most of its concept so you really have to understands the major concepts of it
    - → functions
    - → promises
    - → function References
    - → functional programming
    - → Rest operator
    - → destructuring
  - HTML and CSS ✓
  - JavaScript and ES6 ✓
  - JSX (JavaScript XML)
  - Node + NPM ✓
  - Git ✓

DOM → tree like structure of objects of each HTML element in a page
→ JS → document object
→ created by Rendering Engine

Browser Architecture → based on JS Engine



* Network → used to communicate with server (sending Request, and receiving response)

* Rendering Engine → used to convert HTML/CSS to JS objects, it builds the DOM of the website. Also known as layout engine.

* JS Engine → used to execute JS code and produce the o/p

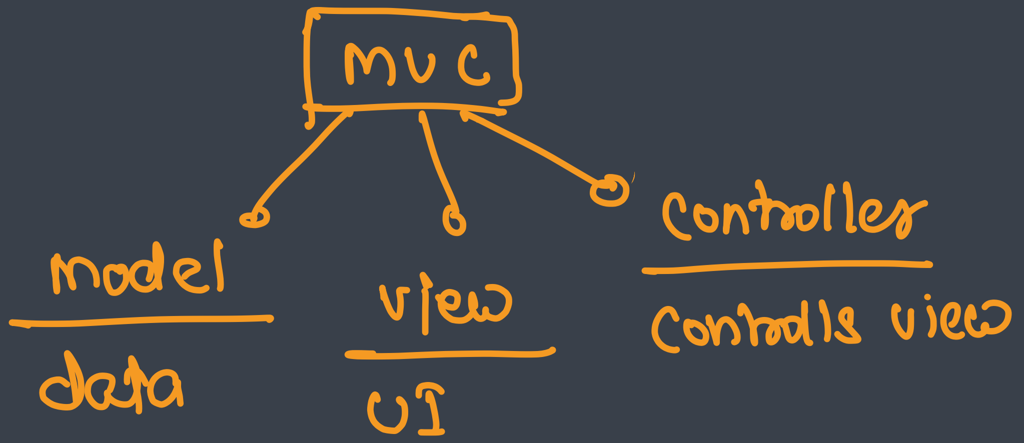
* UI component → used to load the user interface (output of ws)

* web storage component → used to store the data on client. eg. history, session storage, local storage

	Rendering Engine	Javascript Engine
Edge	EdgeHTML	chakra
Safari	WebKit	Nitro JS
Firefox	Gecko	SpiderMonkey
Chrome	Blink	$\boxed{\text{V8}} \rightarrow$ C++

History

- It was created by **Jordan Walke,** who was a software engineer at **Facebook**
- It was initially developed and maintained by Facebook and was later used in its products like **WhatsApp** & **Instagram**
- Facebook developed ReactJS in **2011** in its newsfeed section, but it was released to the public in the month of **May 2013**
- Today, most of the websites are built using MVC (model view controller) architecture
- In MVC architecture, React is the 'V' which stands for view, whereas the architecture is provided by the Redux or Flux



Features

- **Declarative UI**
  - React allows developers to design user interfaces in a declarative way
  - Developers describe what the UI should look like for any given state, and React updates the DOM to match that state automatically
- **Component-Based Architecture**
  - Applications in React are built as a collection of small, reusable components
  - Encourages modularity
  - Makes code reusable and easier to test and maintain
- **JSX (JavaScript XML)**
  - React uses **JSX**, a syntax extension that lets you write HTML-like code inside JavaScript
  - Improves readability and allows developers to embed JavaScript expressions directly within the markup
- **Virtual DOM**
  - React uses a **virtual DOM**, a lightweight copy of the actual DOM
  - Efficiently updates the real DOM by minimizing changes
  - Enhances performance, especially in dynamic applications
- **Unidirectional Data Flow**
  - React enforces a **unidirectional data flow**, meaning data flows in a single direction (from parent to child)
  - Makes debugging easier and improves control over how data is passed and managed

Features

- React Hooks
  - Hooks are functions like useState and useEffect that allow function components to use React features such as state and lifecycle methods
  - Simplifies code by reducing the need for class components and makes managing state and side effects straightforward
- React Router
  - React Router is used for implementing dynamic routing in React applications
  - Allows the creation of single-page applications (SPAs) with seamless navigation
- Context API
  - The Context API enables global state management without needing third-party libraries like Redux
  - For managing themes, authentication, or other data shared across multiple components
- Code Splitting and Lazy Loading
  - React supports code splitting through React.lazy() and dynamic imports
  - Loads only the required code for a specific page or feature
  - Reduces initial load time and improves performance
- Server-Side Rendering (SSR)
  - With frameworks like Next.js, React supports server-side rendering
  - Improves SEO and reduces time-to-interactive for end users
- Performance Optimization
  - React offers built-in tools and techniques for optimization
  - Concurrent Rendering: React 18 introduced concurrent rendering to handle complex UIs efficiently
- Cross-Platform Development
  - With React Native, developers can build mobile applications using React
  - Share code between web and mobile platforms

Advantages

- Easy to learn and use
- Creating dynamic web application becomes simpler
- Reusable components
- Performance enhancements
- Support of handy tools and libraries
- Benefits of being a JS library
- Easy to unit test the application

Disadvantages

- The high pace of development
- Poor documentation
- Its only about the View
- Learning curve for JSX