Web Apps (My Experience of Them)

Dr Steve Huckle

March 2, 2023

Lecture Overview

- Some web apps I’ve built
- The technology underlying those apps
- Q & A

Goals

At the end of this lecture, I hope you'll have a better understanding of Web Applications.

Maybe I can even persuade you to consider a career in Web Development!

Web Development

Web Development is the best job in the world:

1. Your platform of choice has nearly 5 billion daily active users
2. It’s at the forefront of the information age
3. It’s a platform that behaves like a super-intelligent brain:
    1. *It can cure diseases*
    2. *Eliminate poverty*
    3. *Advance science*

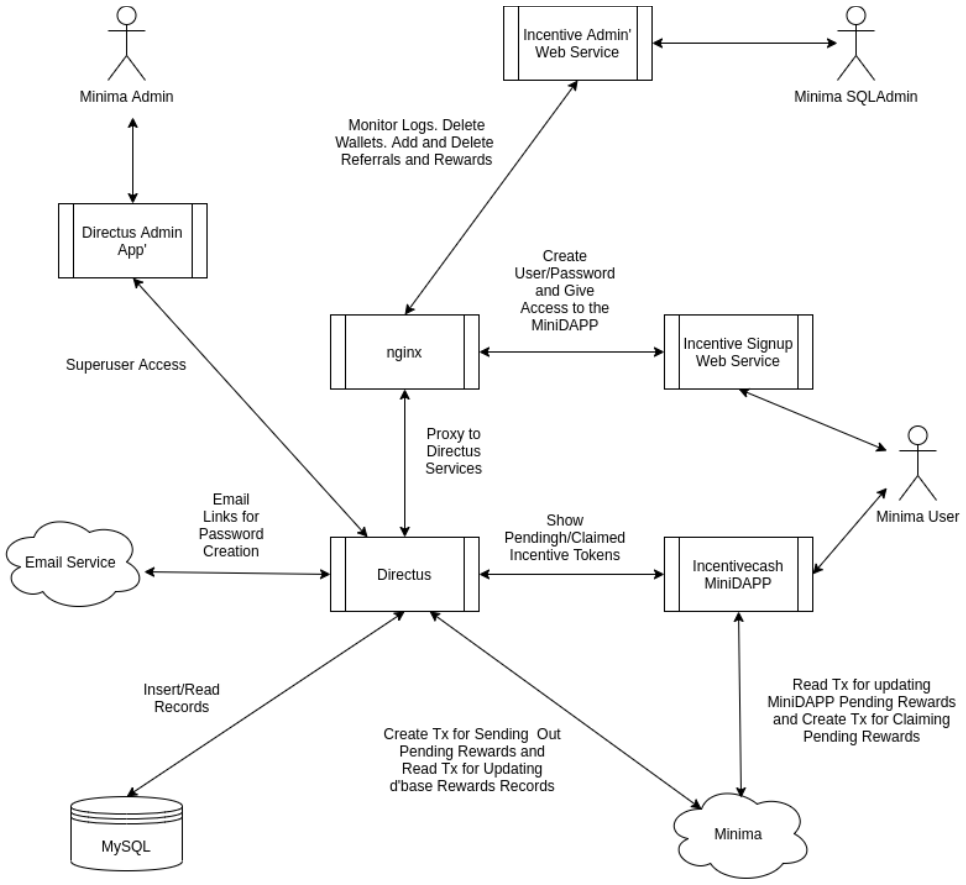
Web Development (cont'd)

"When you realize you're wrong in the middle of an argument"

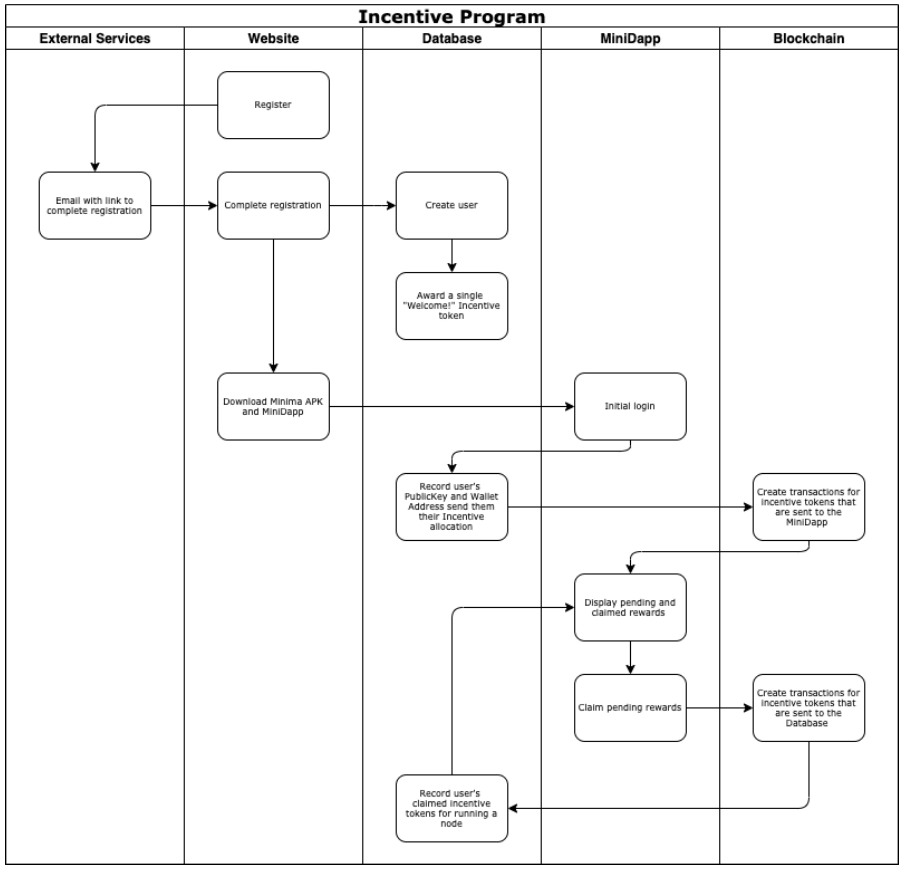
Minima

https://www.minima.global/

Incentivecash System



Incentivecash System (cont'd)



My Site

https://glowkeeper.github.io/

Rectangles

https://glowkeeper.github.io/rectangle-react/

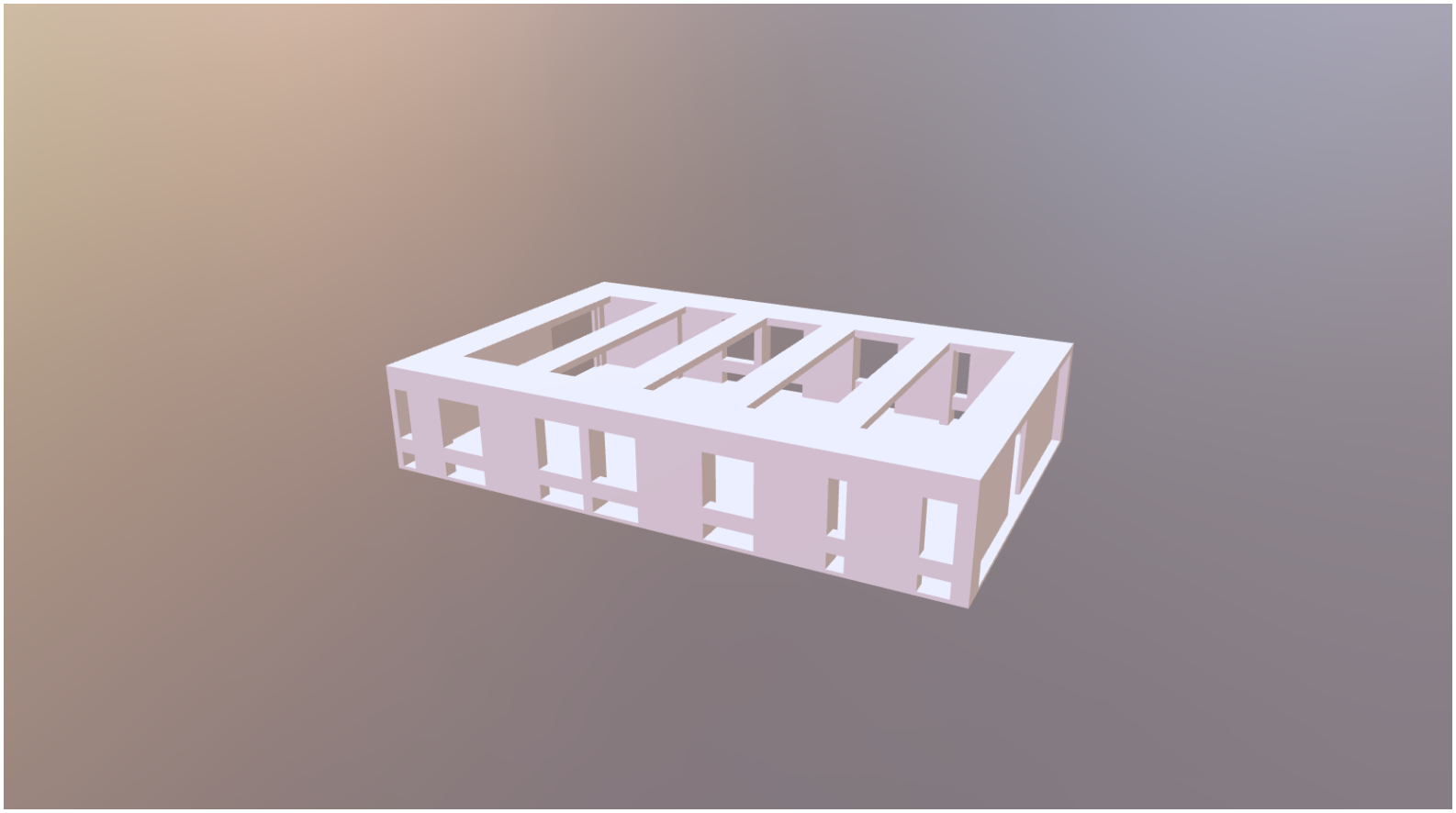
https://github.com/glowkeeper/rectangle-react

Storymaker

https://glowkeeper.github.io/storymaker/

https://github.com/glowkeeper/storymaker

Web XR



Story Builders

http://www.story-builders.co.uk/

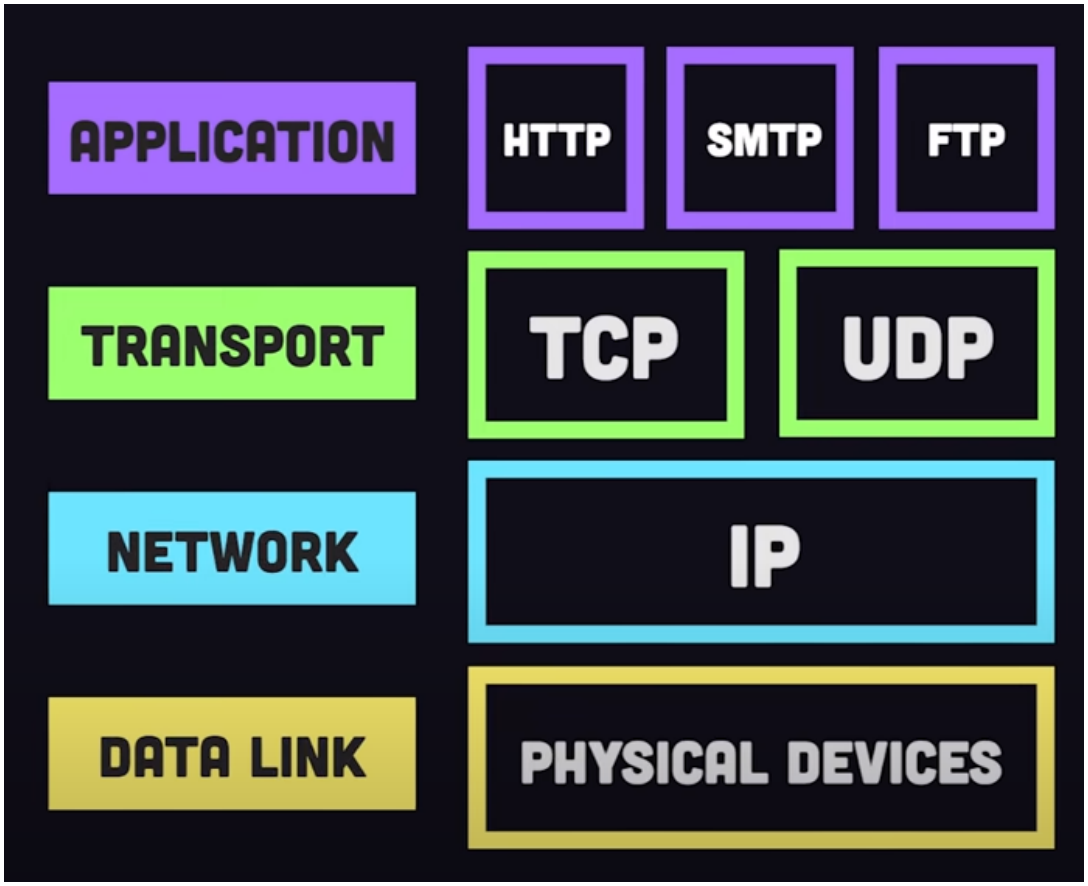
Web Development

To take advantage of that super-intelligent brain, you’re going to want to know about the tech’ that constitutes that brain

Internet

The Internet is a network of billions (trillions?) of connected machines, first established in the early 1980s. You can think of it as the hardware underlying all the software built on top of it

Protocol Suite

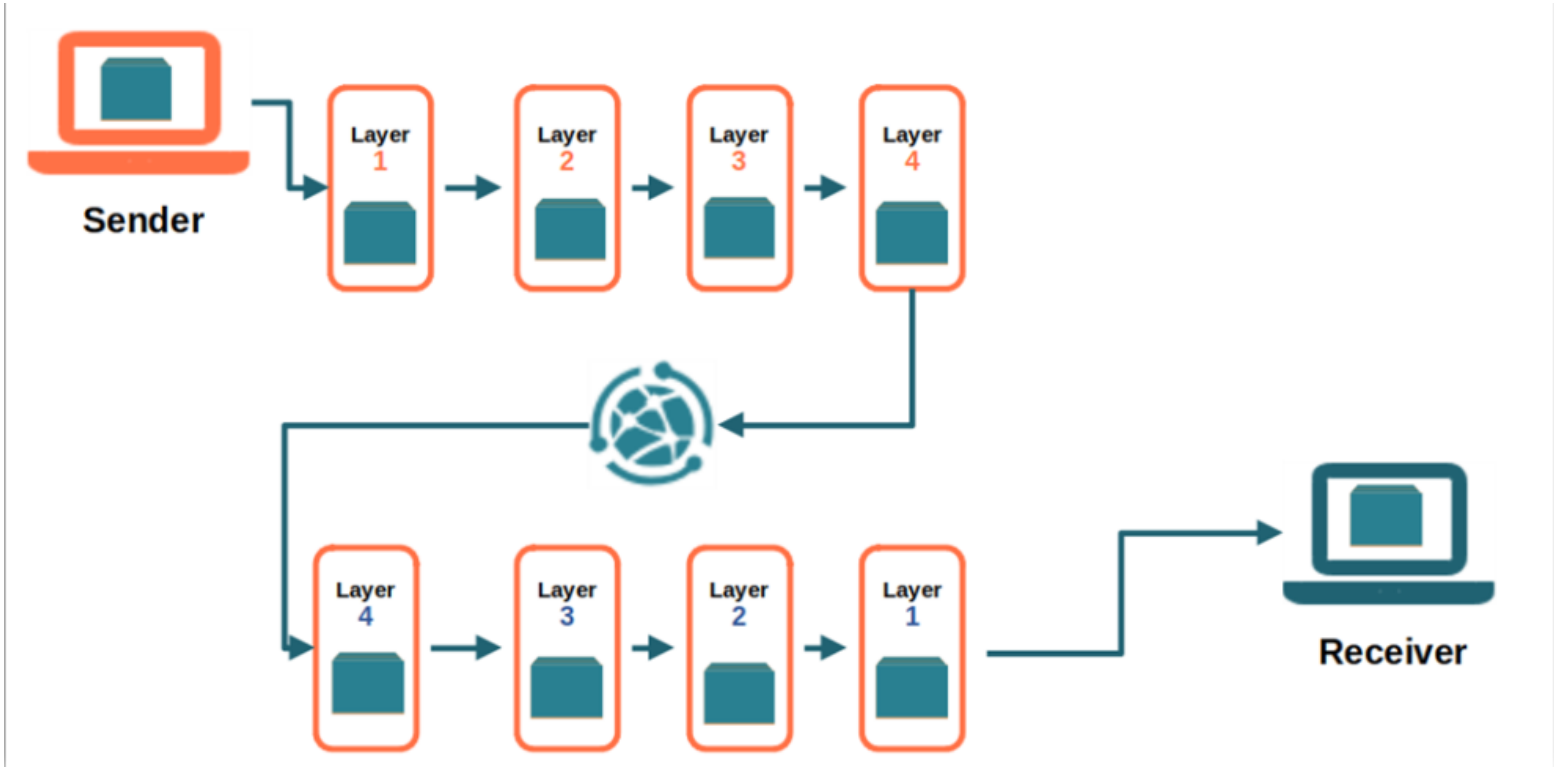


Internet Protocol

Network IP is used to identify machines on the Internet by assigning them a unique IP address, e.g.

168.172.25.1

TCP


Sender
Layer 1
Layer 2
Layer 3
Layer 4
Layer 4
Layer 3
Layer 2
Layer 1
Receiver

Packets

Data is sent as a collection of small packets across the data link, and resequenced by the receiving device

World Wide Web

It’s like some software that sits on top of the hardware (the Internet)

Http

The Hypertext Transfer Protocol (Http) allows people to access web applications

Http Methods

- GET
- POST
- PUT
- PATCH
- DELETE

DNS

Every website has a unique domain name that maps to a specific IP Address via DNS:

Name: www.google.com

Address: 142.250.178.4

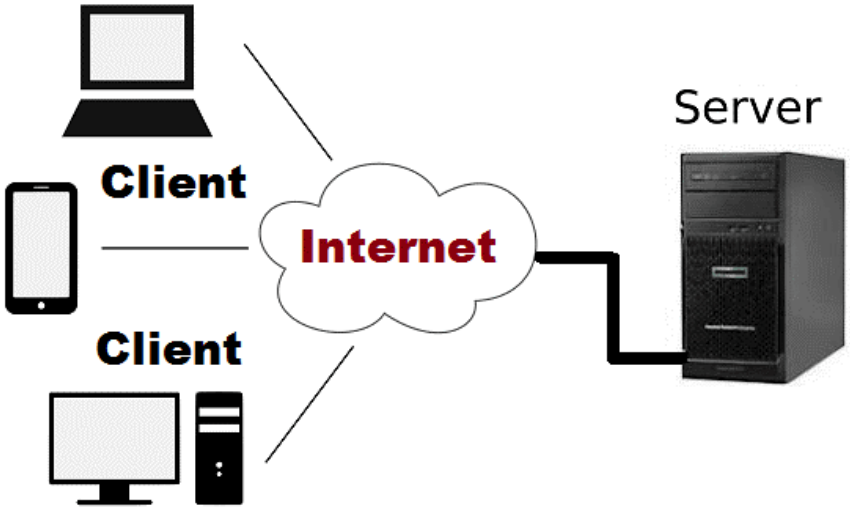
Registar

URL

Http gives resources on the web a uniform resource locator (URL) - some unique address under each domain:

http://www.google.com/gmail

Client Server



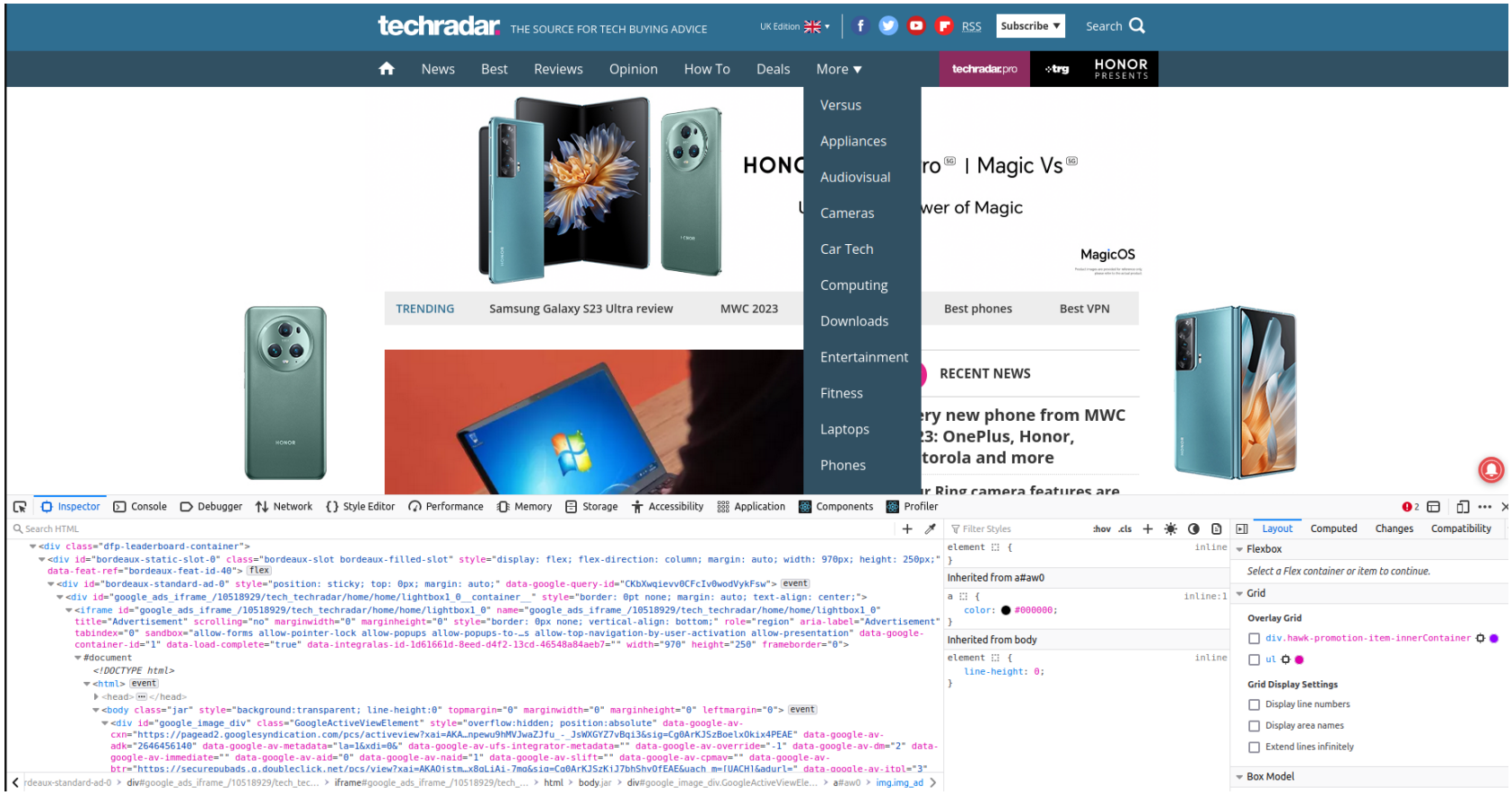
Browser

Browsers allow us to access the web by rendering content from URLs served by those servers

HTML

The content rendered by a browser is represented by Hypertext Markup Language (HTML)

Developer Tools



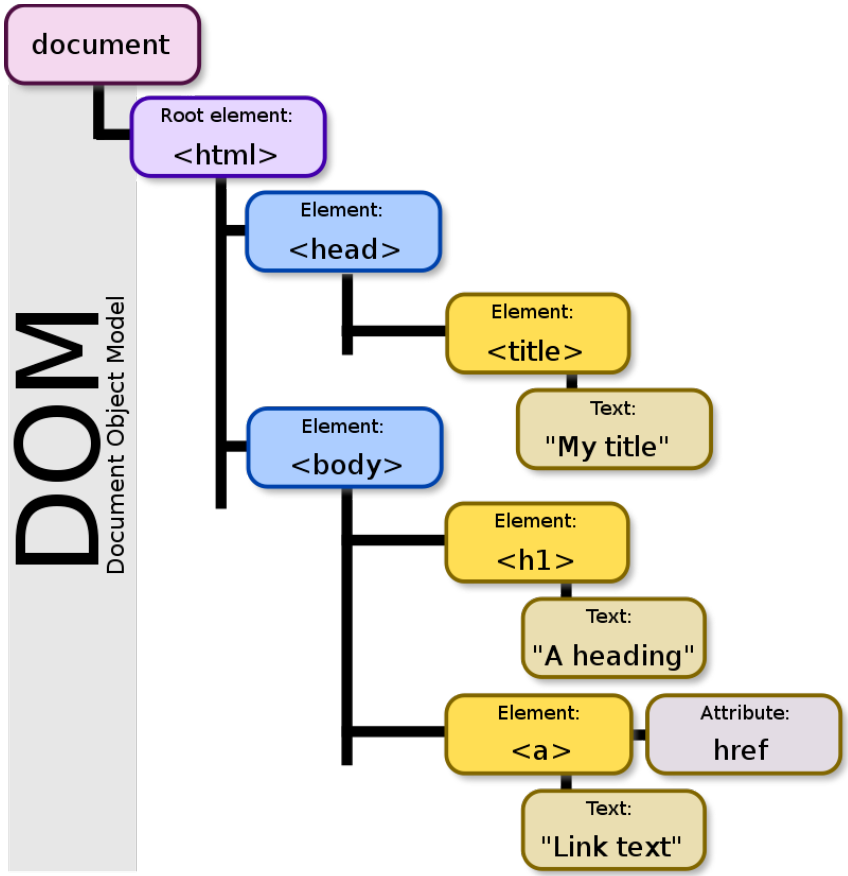
Semantic HTML Elements

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="style.css">
        <script defer src="index.js"></script>
        <title>Web Development</title>
    </head>
    <body>
        <main>
            <article>
                <h1>Web Development ROX</h1>
                <p>...you can reach 5 billion people - huzzah!</p>
            </article>
        </main>
    </body>
</html>
```

Anchors

```
<a href="http://example.com">example.com</a>
```

DOM

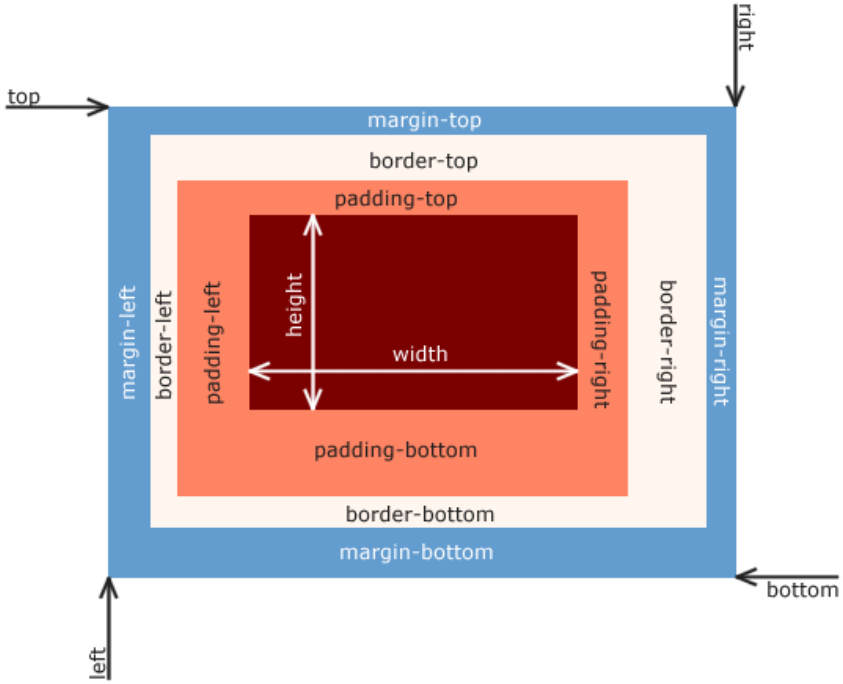

DOM
Document Object Model
document
Root element:
<html>
Element:
<head>
Element:
<title>
Text:
"My title"
Element:
<body>
Element:
<h1>
Text:
"A heading"
Element:
<a>
Attribute:
href
Text:
"Link text"

CSS

```
p {
  margin-bottom: 1rem;
  line-height: 1.3rem;
  color: #0000cc;
}
```

Layout and Positioning

(CSS is tricky - it takes time, practice and patience)



Responsive Layouts

CSS provides tools for making your web apps look good across the range of devices upon which they might be viewed

```
@media only screen and (max-width: 600px) {

    p {
        margin-bottom: 0.5rem;
        line-height: 1rem;
        color: #00cccc;
      }

}
```

Responsive Layouts (cont’d)

Grids and flexboxes

```
.grid-container {
    max-width: 600px;
    padding: 8px;

    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
    grid-auto-rows: minmax(200px, auto);
    gap: 8px;

    border: 2px dotted lightcoral;
}
```

Javascript

Adds programmability and user interaction to the content

```
<script>
    const hello = "Hello World!";
    alert(hello);
</script>
```

Events

Events Listeners

```
const button = document.querySelector('#submitButton')
button.onClick = () => {
    alert('submitted')
}
```

ECMAScript

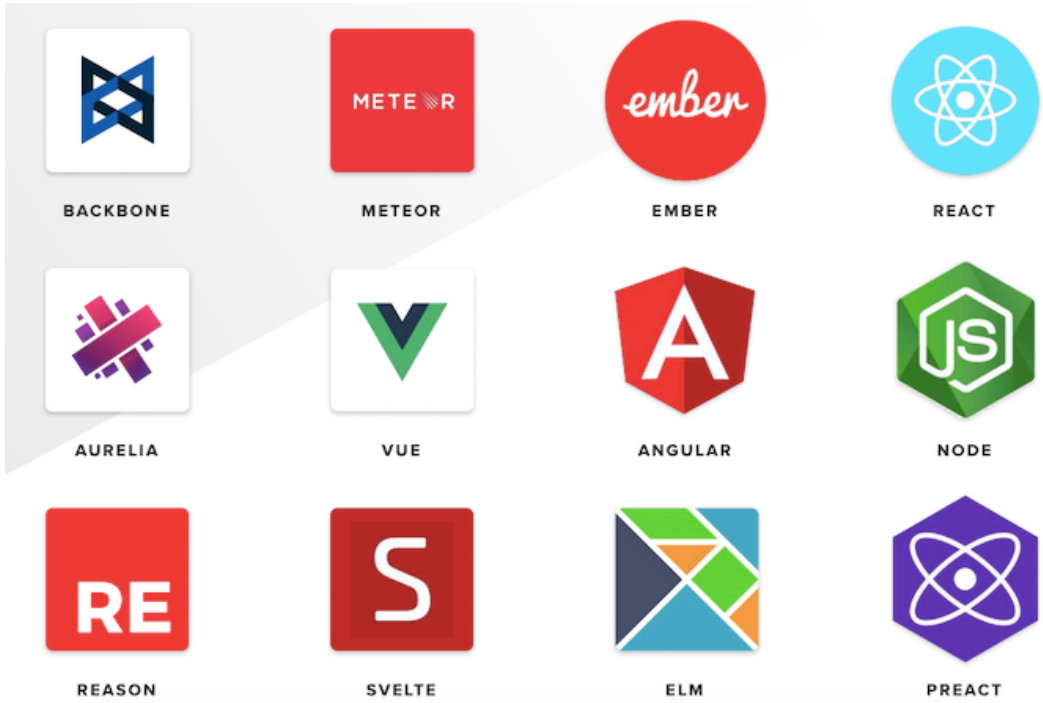
Javascript is standardised across all major browsers

Typescript

Javascript is dynamically typed. Typescript is a language that is a superset of Javascript that adds syntax for types

```
let hello: string = `hello ${name}`;
```

Frontend Frameworks



Component-based

```
<About />
```

Node Package Manager

Export

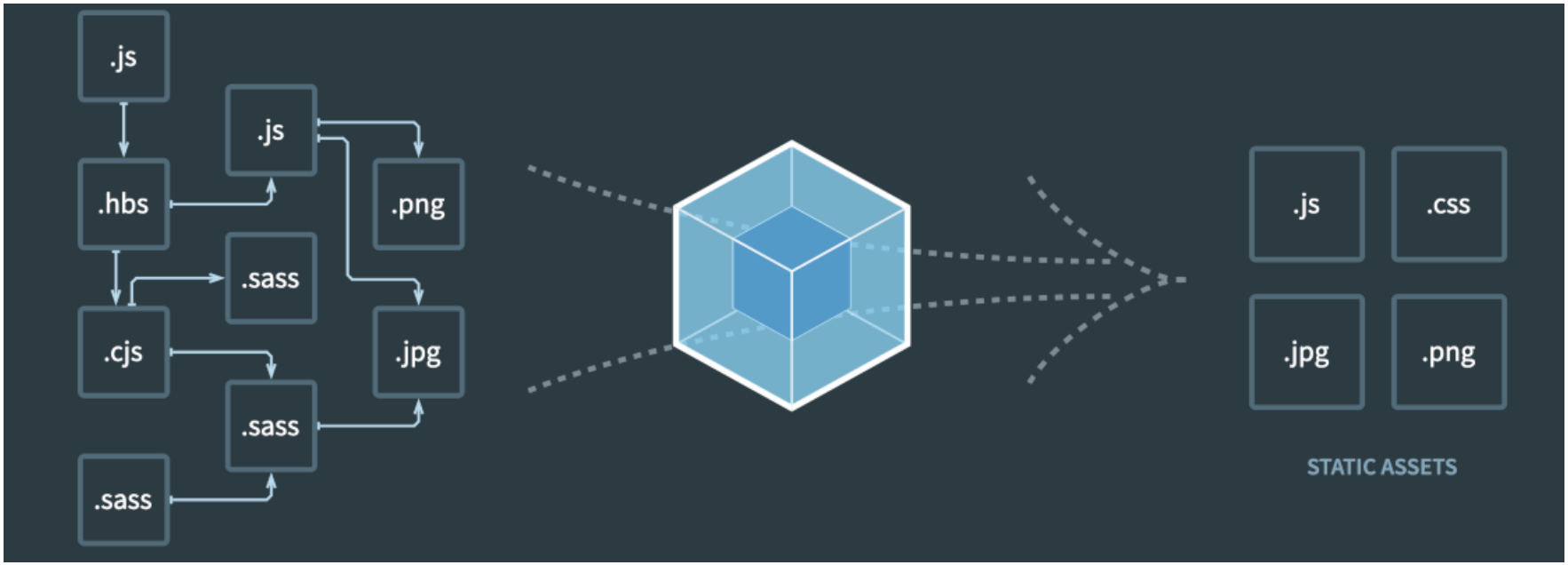
```
export const fetchData = async (props) => {
    const { url, options, cb } = props

    try {

        const response = options ? await fetch(url, options) : await fetch(url)
        const data = await response.json()
        if (cb) cb(data)

    } catch (error) {
        console.error('fetchData', error)
    }
}
```

Import

```
import { fetchData } from "./utils";
...
const fetchParams = {
    url: process.env.REACT_APP_DBASE + ":" +  process.env.REACT_APP_DBASE_PORT + Remote.website,
    cb: fetchCallback
}
fetchData(fetchParams)
```

Single-page Applications (SPA)

Bundlers

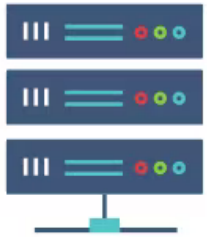

.js
.js
.hbs
.png
.sass
.cjs
.jpg
.sass
.sass
.js
.css
.jpg
.png
STATIC ASSETS

Frontend Tooling

Static-site Generation

Server-side Rendering

User requests a website



Server sends HTML file with resources

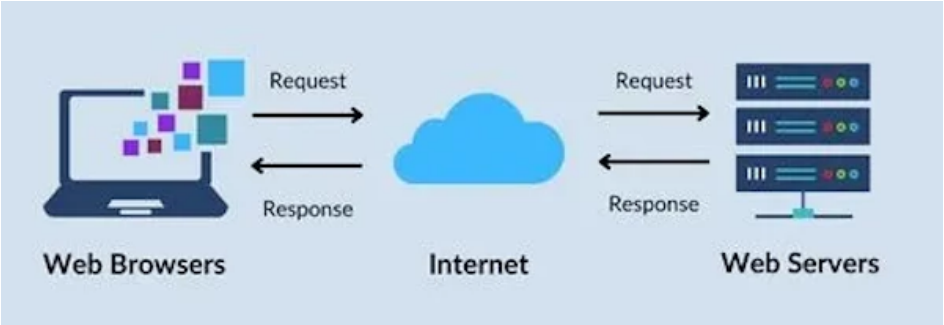
Browser downloads HTML

Browser downloads JS and CSS

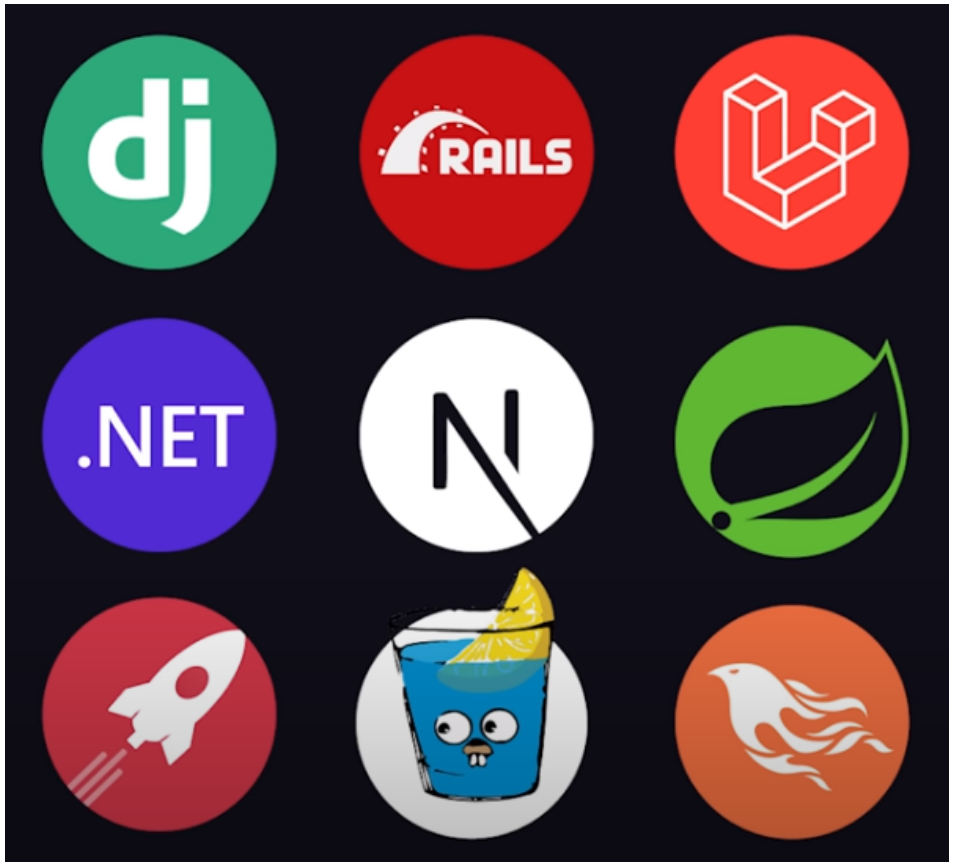
Browser renders the website

Servers

Web Servers


Request
Response
Request
Response
Web Browsers
Internet
Web Servers

Backend (Server) Systems



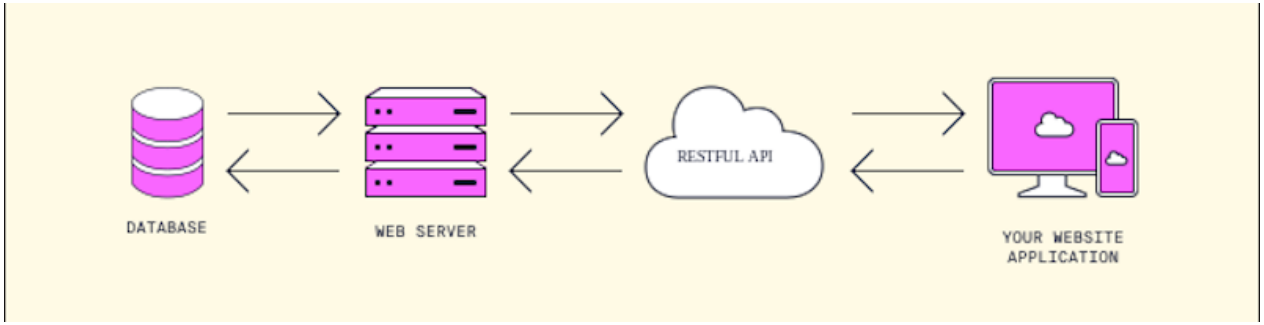
Node.js

Express


Express
4.18.1
Fast, unopinionated,
minimalist web framework for
Node.js

Content Management Systems

REST API



Databases


MySQL
amazon REDSHIFT
IBM
mongoDB
SQL
Google BigQuery
cassandra
druid
ORACLE
snowflake

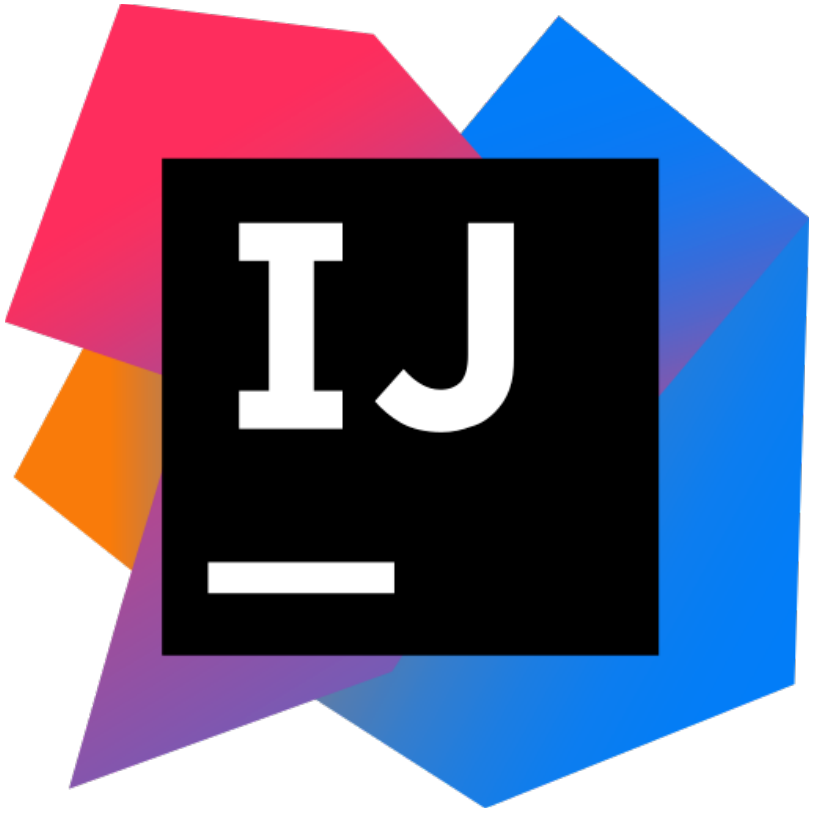
Object Resource Managers

Data Exchange (JSON)

```
{
  "squadName": "Super hero squad",
  "homeTown": "Metro City",
  "formed": 2016,
  "secretBase": "Super tower",
  "active": true,
  "members": [
    {
      "name": "Molecule Man",
      "age": 29,
      "secretIdentity": "Dan Jukes",
      "powers": ["Radiation resistance", "Turning tiny", "Radiation blast"]
    },
    {
      "name": "Madame Uppercut",
      "age": 39,
      "secretIdentity": "Jane Wilson",
      "powers": [
        "Million tonne punch",
        "Damage resistance",
        "Superhuman reflexes"
      ]
    },
    {
      "name": "Eternal Flame",
      "age": 1000000,
      "secretIdentity": "Unknown",
      "powers": [
        "Immortality",
        "Heat Immunity",
        "Inferno",
        "Teleportation",
        "Interdimensional travel"
      ]
    }
  ]
```

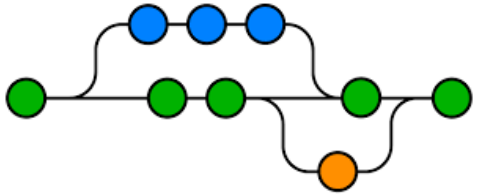
```
}
```

IDE



Linters

Version Control



Slides

https://github.com/glowkeeper/wepp-apps-presentation/blob/main/presentation/webapps.md

Don’t Panic!

…there’s always stackoverflow

Thank-you

Dr Steve Huckle

s.huckle@sussex.ac.uk