Waterford Institute *of* Technology

INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Module: Web App Development 2

Overview

Presenters:

Diarmuid O' Connor (doconnor@wit.ie)

Dr. Frank Walsh (fwwalsh@wit.ie)

Where:

https://tutors-design.netlify.com/course/wad2-2020-wit.netlify.com

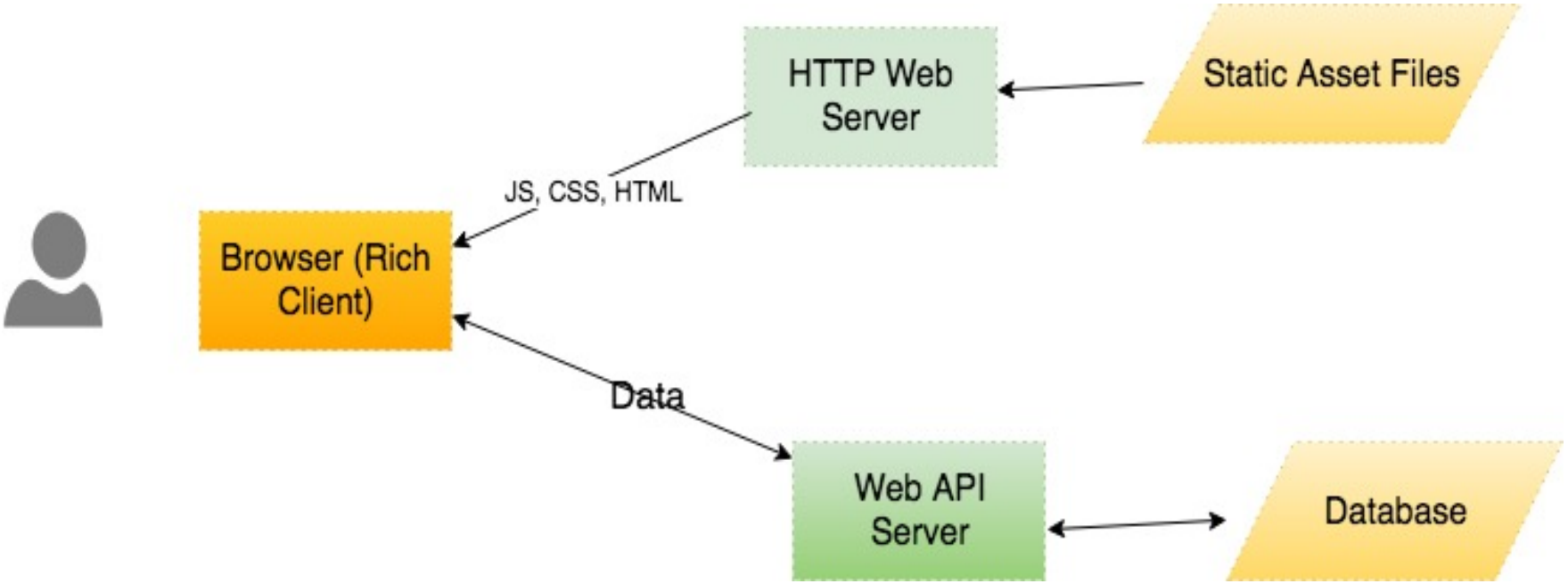
TL;DR

Design and develop modern, secure Web apps.

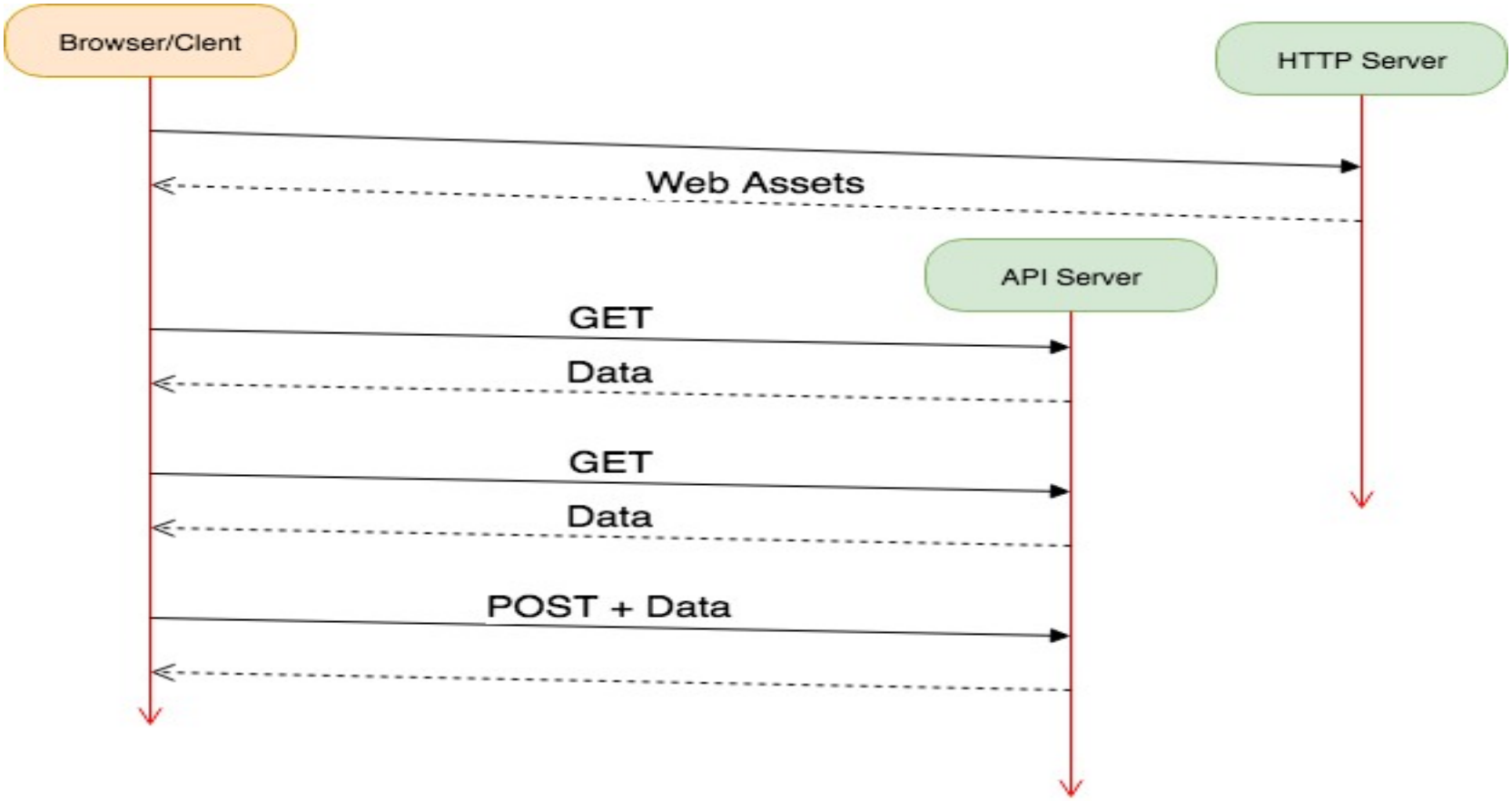
Agenda

- Context
- This module's focus
- Software installation requirements

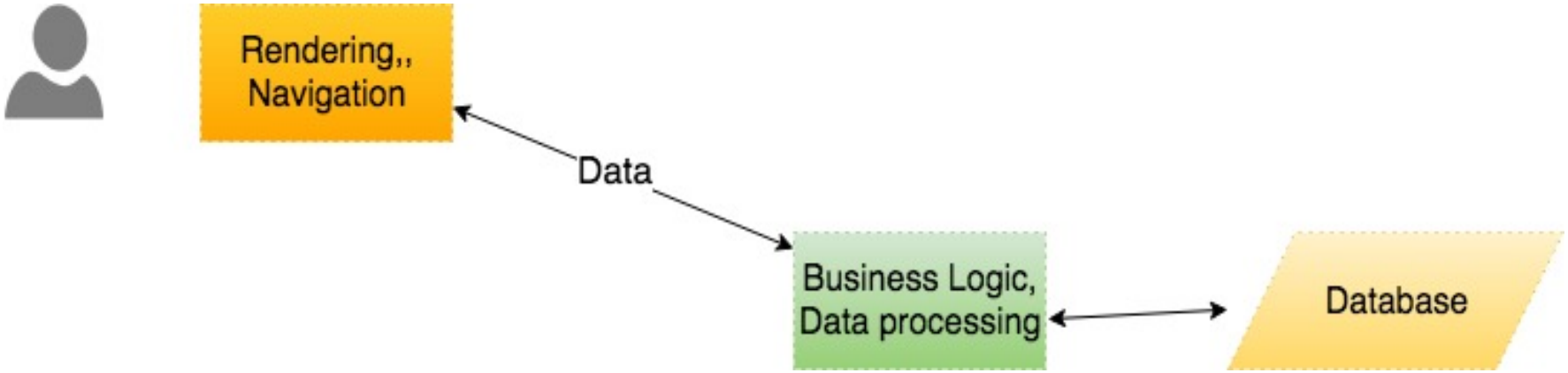
Modern Web Apps - Architecture



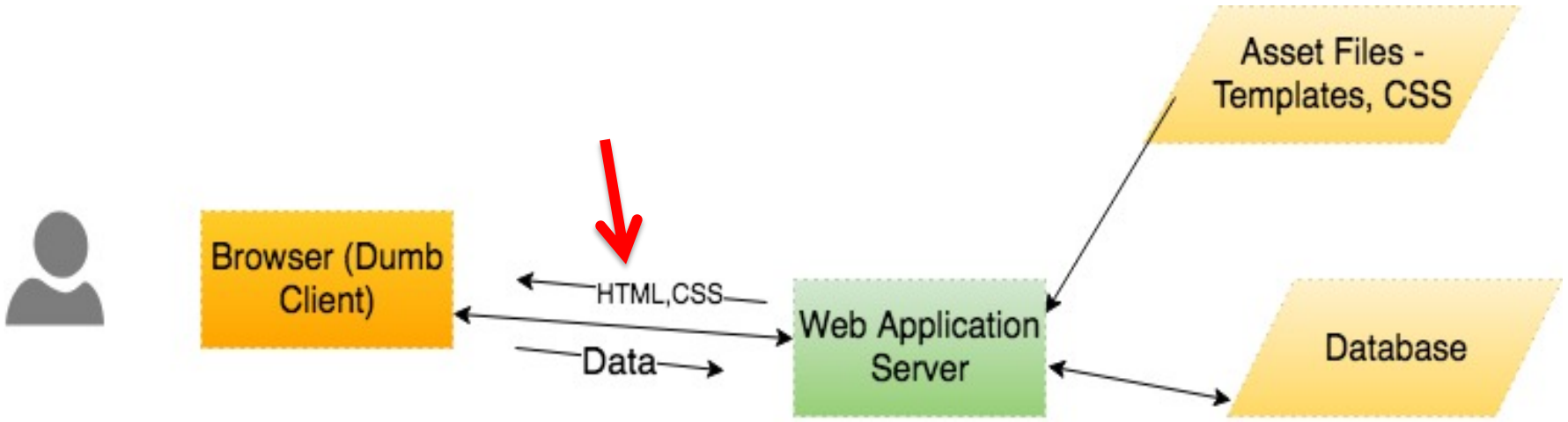
Modern Web Apps – HTTP Communications flow



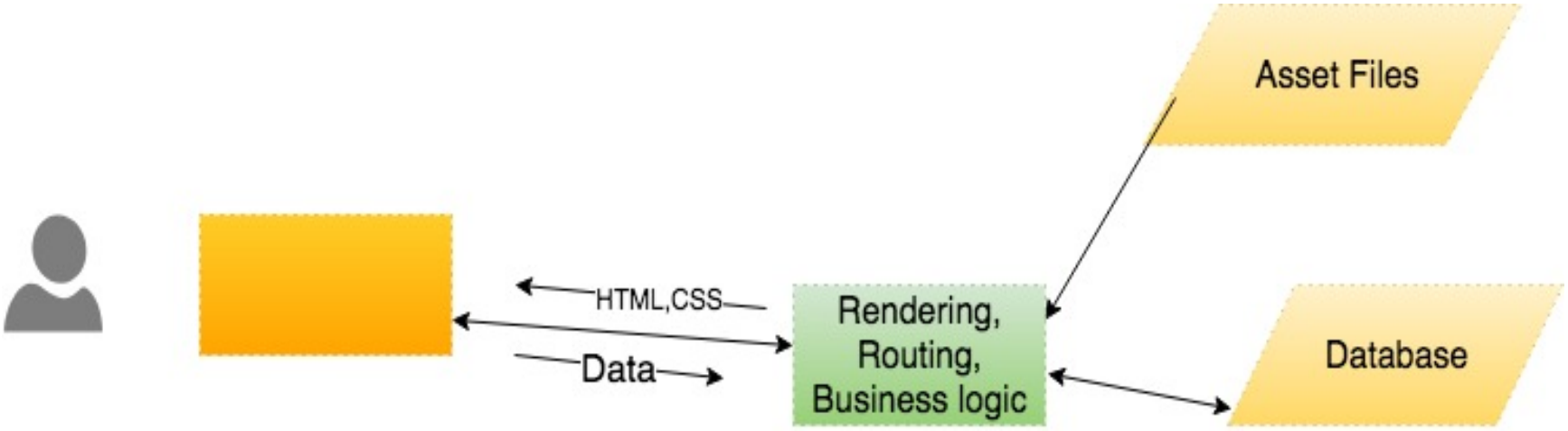
Modern Web Apps – Sub-system roles



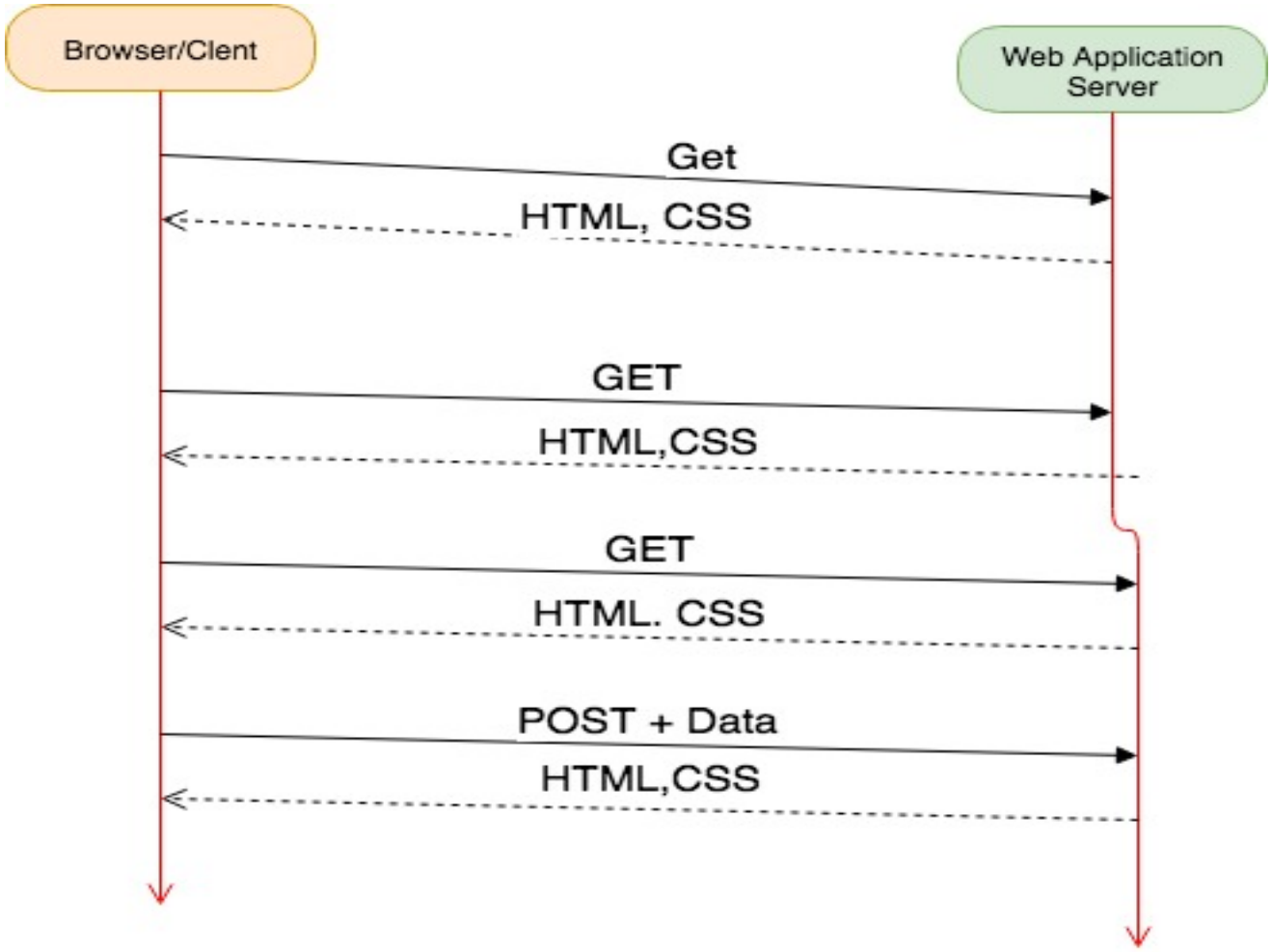
Traditional Web Apps – Architecture



Traditional Web Apps – Sub-system roles.



Traditional Web Apps – HTTP Communication flow.



Modern Web Apps.

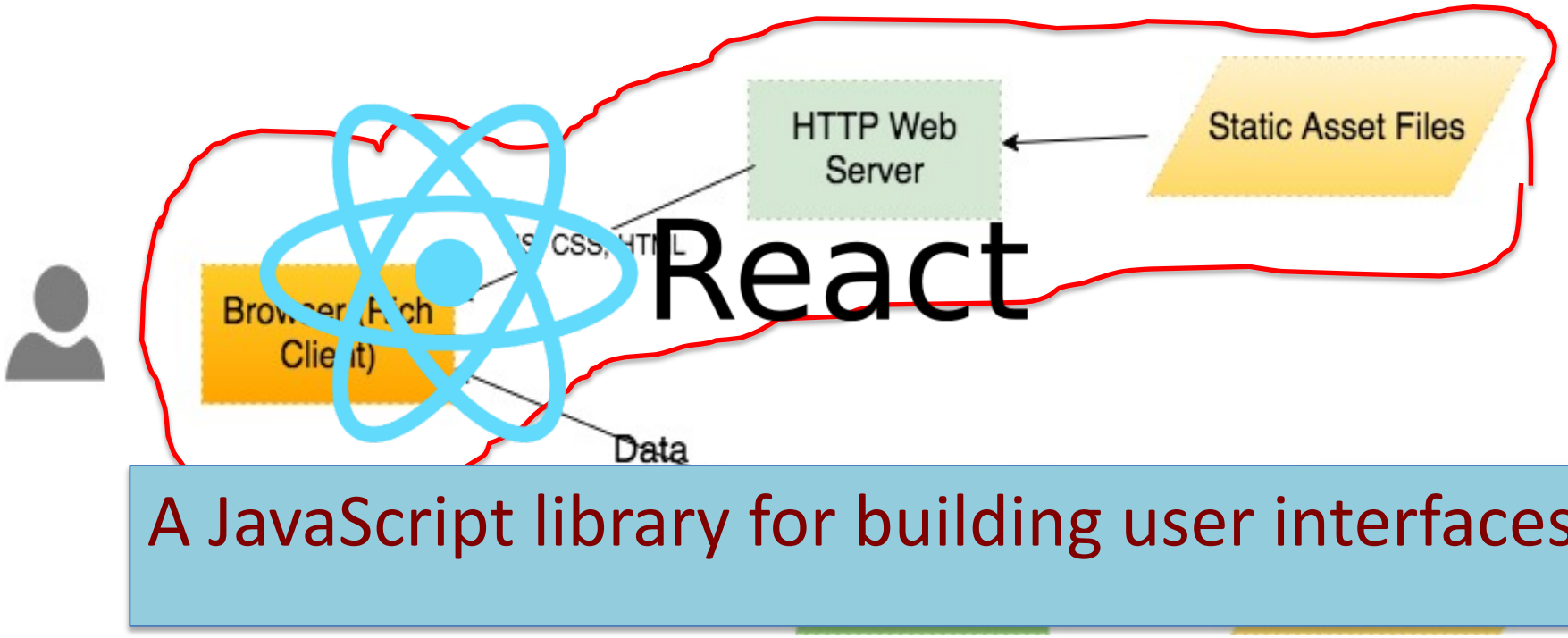
1. **Early Examples: Gmail (2004), YouTube (1st generation)**
   - **Disadvantage:**
     - **Poor developer experience (DX) – code bloat; poor maintainability; browser inconsistencies.**
     - **Lacked addressability (Key principle of the web).**

2. **Jquery – A** cross-platform **JavaScript library to** simplify **the client-side scripting of HTML pages. (2006)**
   - **Adv.: Better developer productivity;**
     - **Reduced code bloat**
     - **Cross-browser**
   - **Disadvantages –**
     - **Lacked Addressability.**
     - **Poor code maintainability (spaghetti code).**
3. **Single Page App (SPA)** *frameworks*. **(2010 ish)**
   - **Support addressability; Improved DX; Improved maintainability.**

Agenda

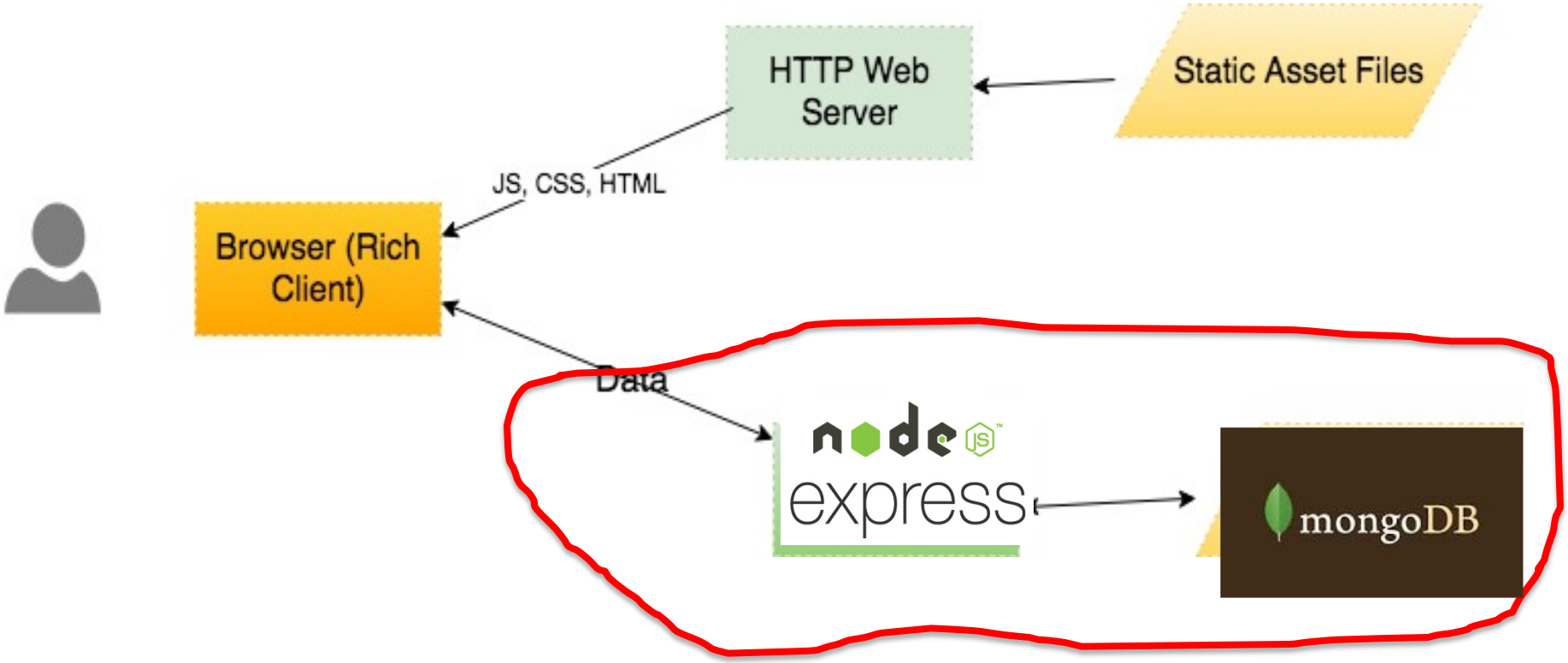
- Context. ☑
- This module's focus

Modern Web Apps

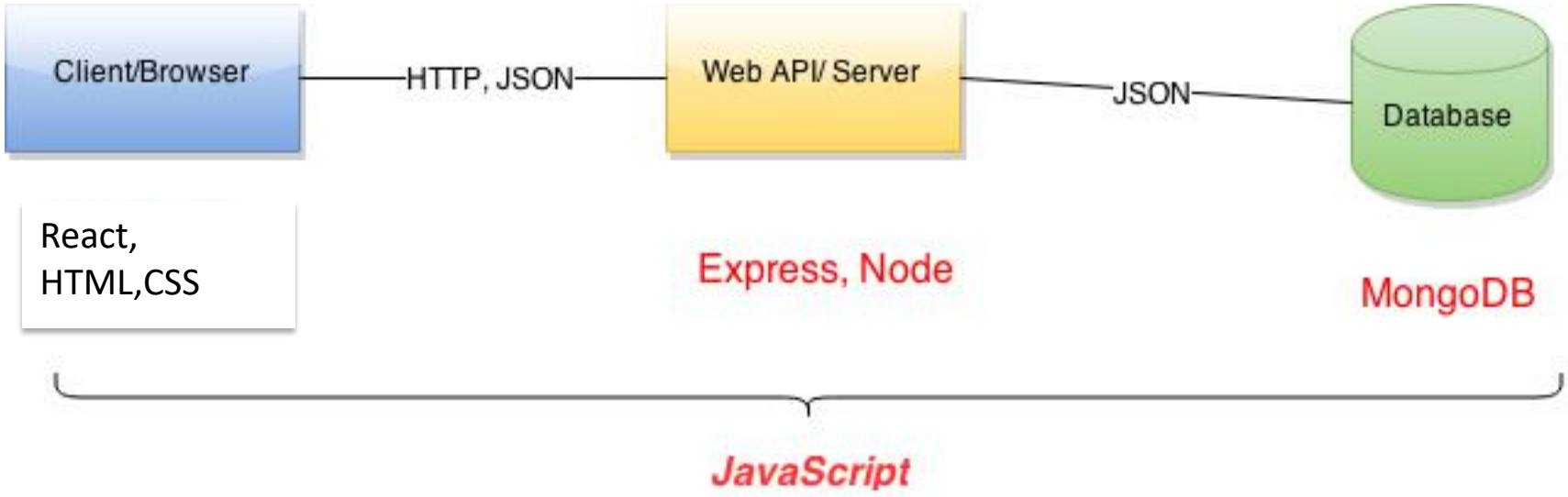


A JavaScript library for building user interfaces

Modern Web Apps



Modern Web App architecture.



The **MERN** stack. (Mongo, Express, React, Node)

Agenda

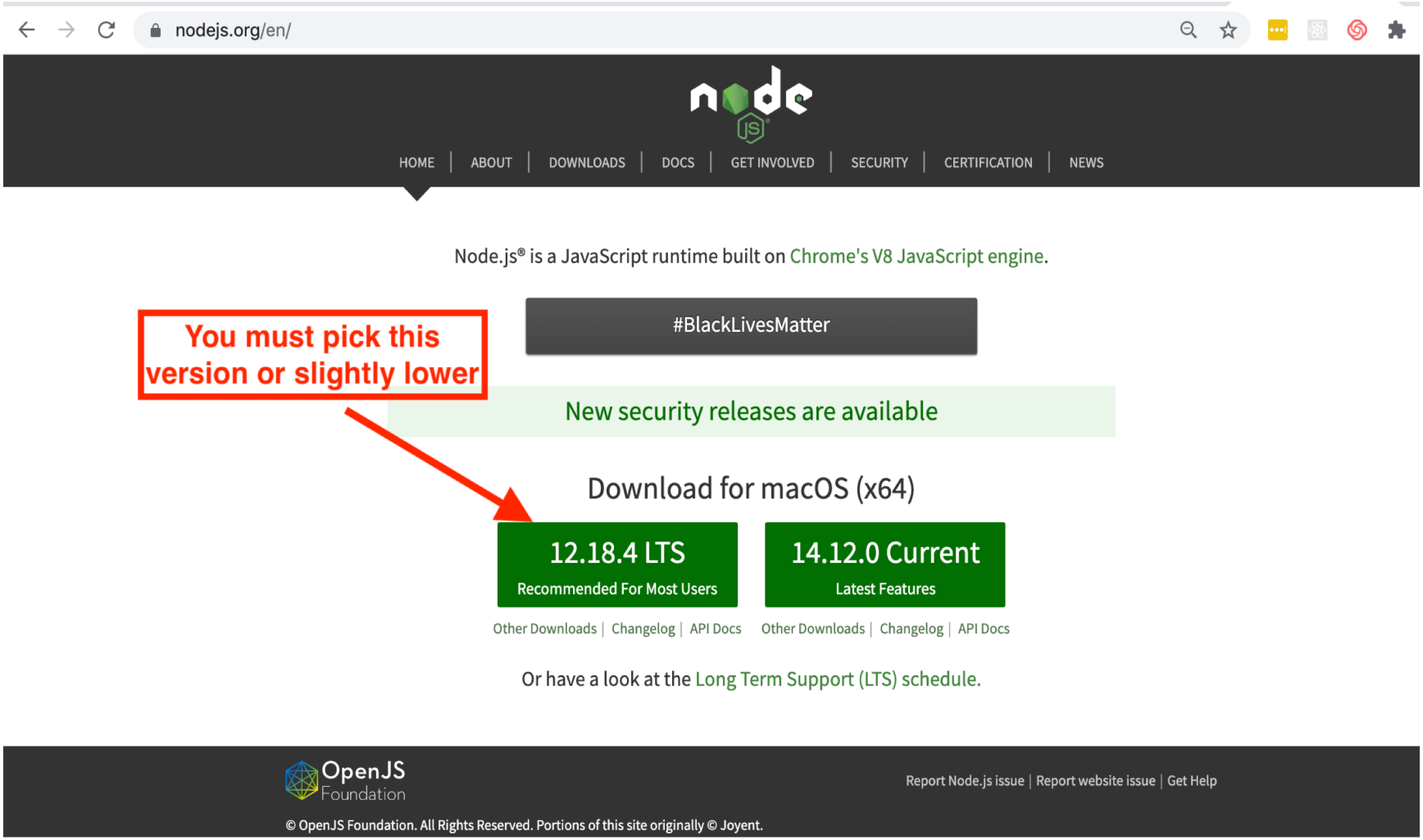
- Context. ☑

- This module's focus
  - Single Page Apps, using React ☑
  - Web APIs, using Node and MongoDB ☑

- Software Installation Requirements

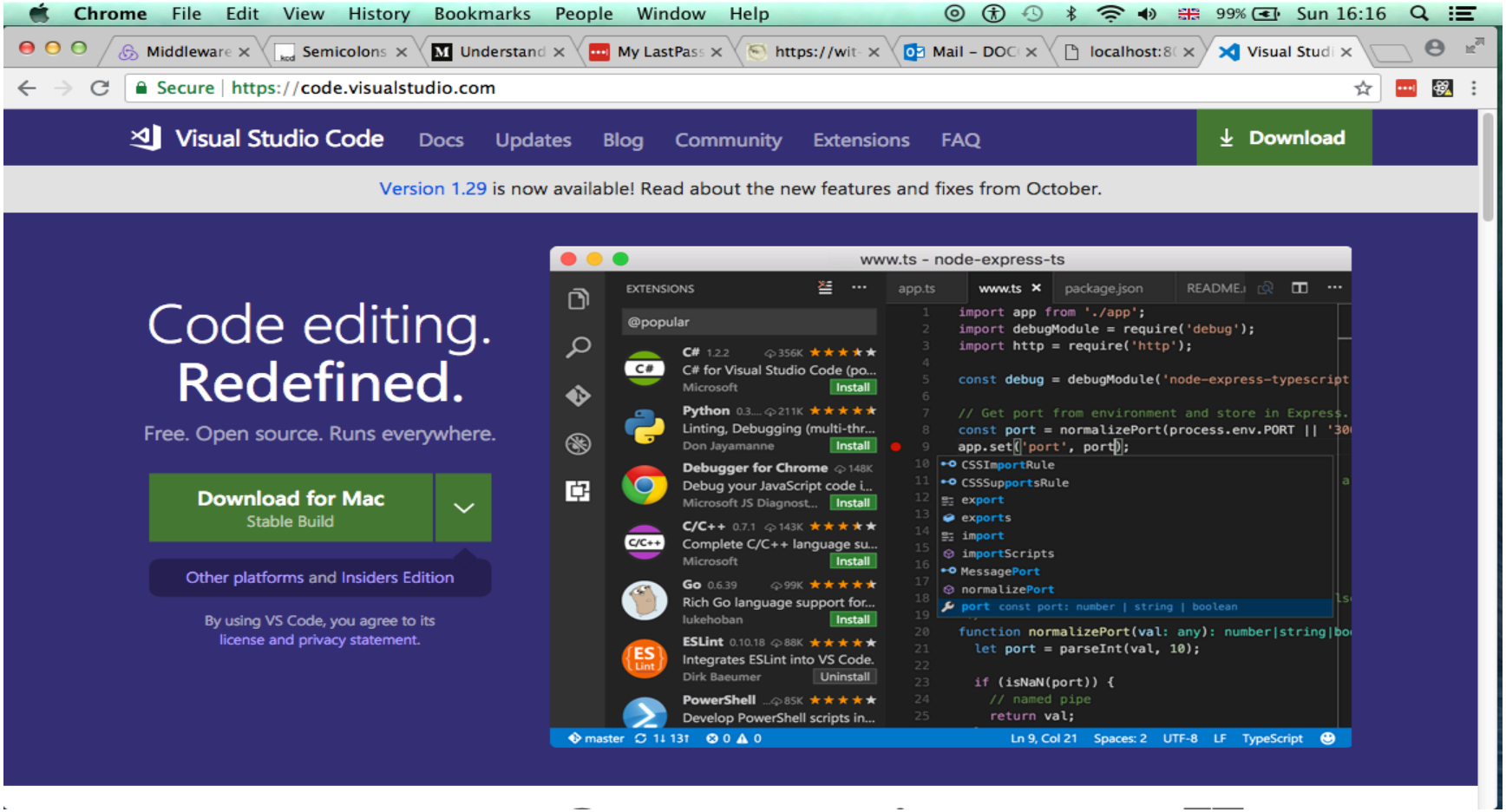
Node.js

(A JavaScript runtime platform)

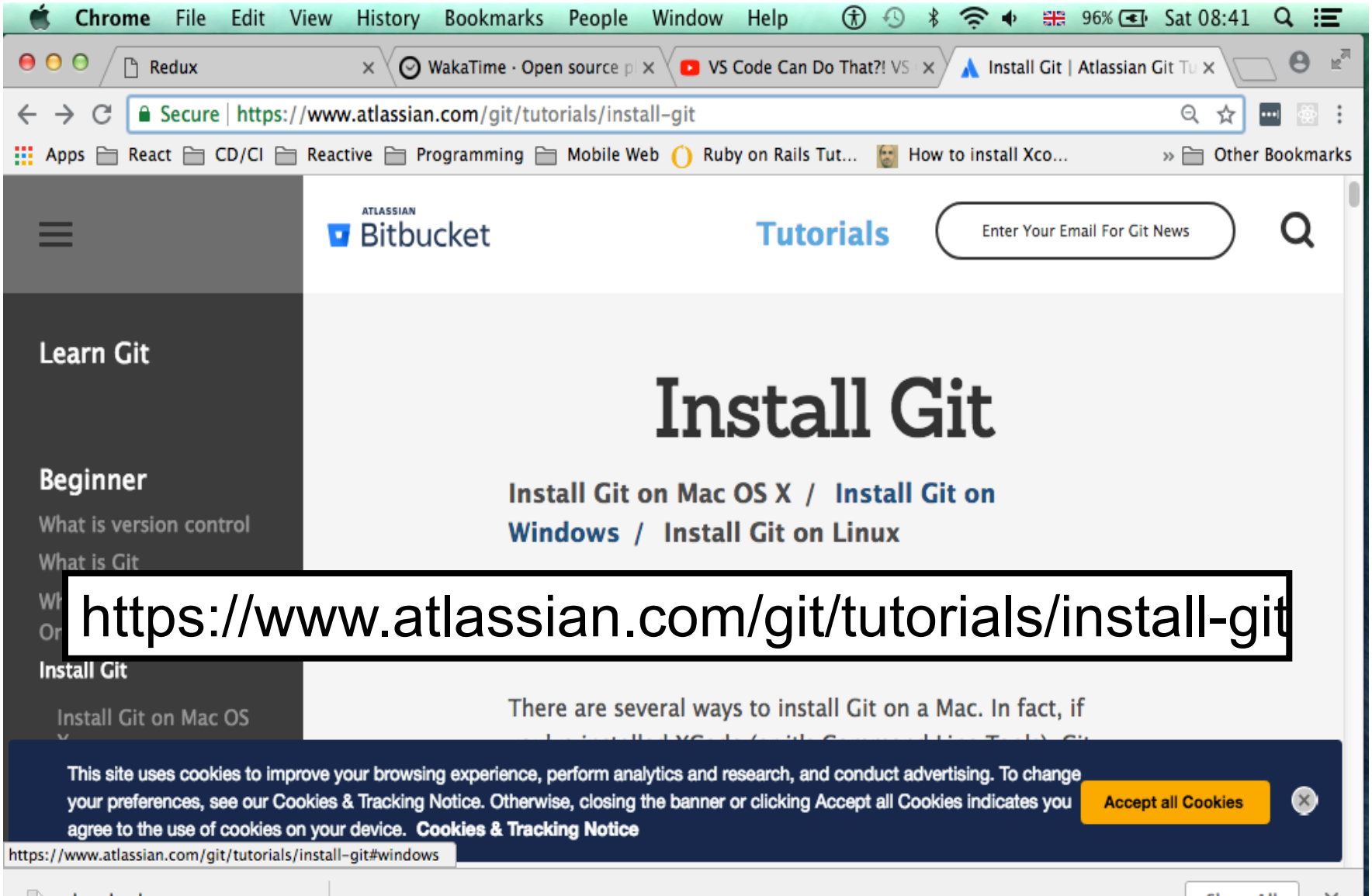


Text Editor

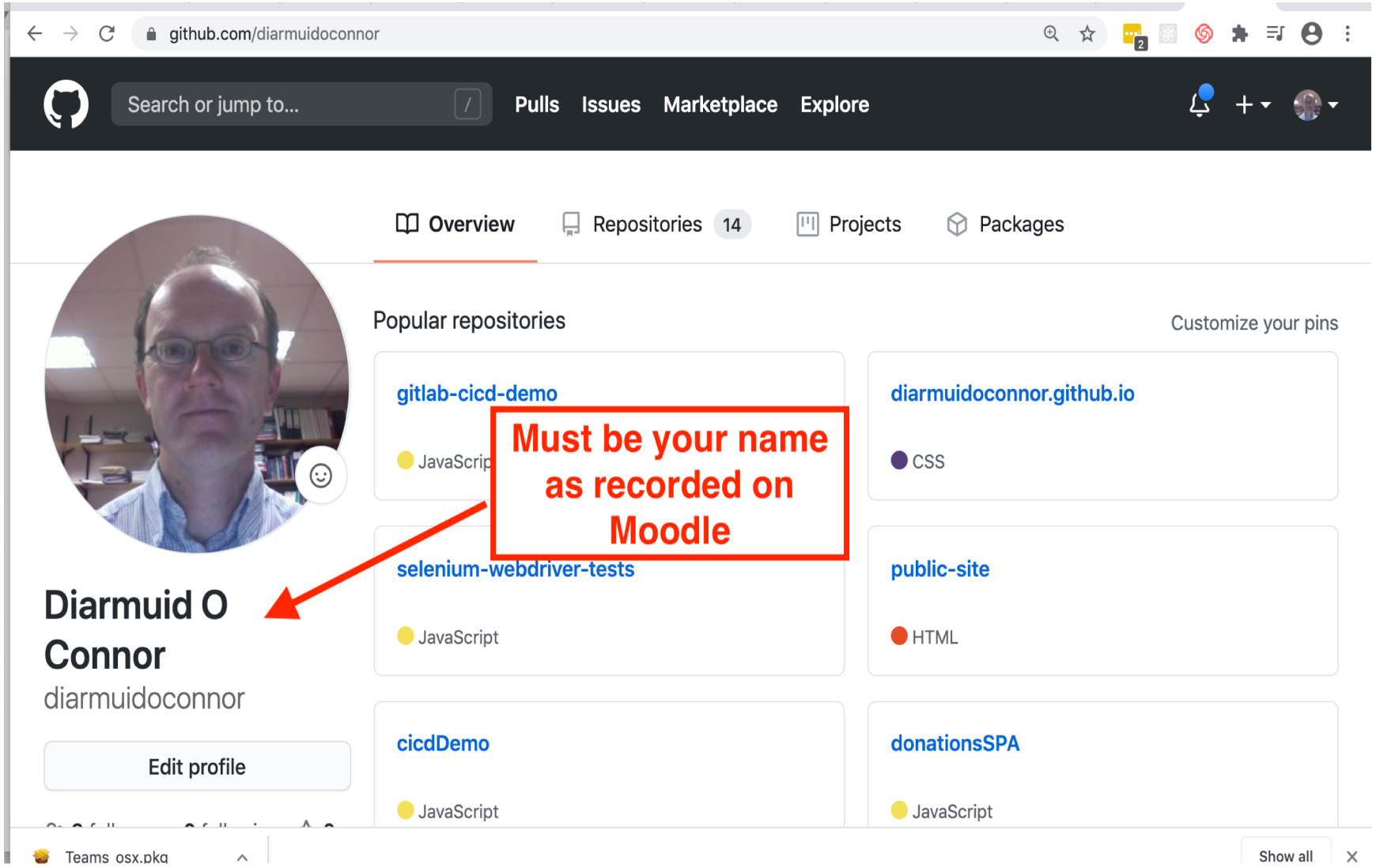
(Not a heavyweight IDE, thankfully)



Source Control (Git)



Create a GitHub account



Summary.

- **Two models of web apps:**
  - **Traditional – Server Side Rendering (SSR)**
  - **Modern – Client Side Rendering (CSR)**

- **CSR model has evolved:**
  - **Unmaintainable JS codebase to SPA framework based codebase.**
  - **No addressability to addressable.**

- **Server-side:**
  - **From SSR web applications to Web APIs.**
  - **REST style interface.**

- **NoSQL databases.**