Modern web.

Chris Bracewell

The problem for us

- Uninspiring projects.
- Generally displaying content from a DB to the user.
- Collecting data from the user and sending it to the backend.
- Communicating with APIs.
- Doing both in the front end with some state management solutions
- Discussing which is better, Angular, React or View
- Trying to understand Webpack
- Just simple IO
- Not the most interesting conversation to friends & family

Why?

- In consultancy we’re charged out to do these things because we’re good at it
- We build all kinds of solutions for e-commerce, insurance and banking.
- all using the same skills.
- The skills we learn and refine from doing this allow are typically HTML, CSS and JS
- With these core skills we can create much more complex & interesting things
- These projects enable and teach us to build way cooler things

Reasons to be excited with JS / Frontend

- Mobile development - React Native, Ionic
- Browser support for native features, Vibrate API, GYRO
- Animation libraries, WebGL, Canvas, SVG, ThreeJS
- Virtual reality - AFrame, WebVR
- Chat systems - Twitch
- Desktop applications, Electron, VSCode, Slack etc
- Off topic:
    - 5G and network performance opening doors, IOT
    - Gigabit wifi
- Much more.

My recent interests..

Websockets - Socket.io

Canvas - ThreeJS

Websockets.

How do they work?

- Client and server handshake
- Agree to a web socket connection and upgrade to websocket
- **Bi-directional**: Protocol where there are no predefined message patterns such as request/response. Either client or server can send a message to the other party.
- **Full-duplex**: In HTTP at a given time, either client is talking to server or server is talking to client. WebSocket allows client and server to talk independent of each other.
- **Single TCP Connection**:
    - HTTP: TCP opens on request, closes on response. For WebSocket, the HTTP connection is upgraded using standard HTTP Upgrade mechanism and client and server communicate over that same TCP connection for the lifecycle of WebSocket connection.
    - What is TCP: TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent.

Canvas.

Why?

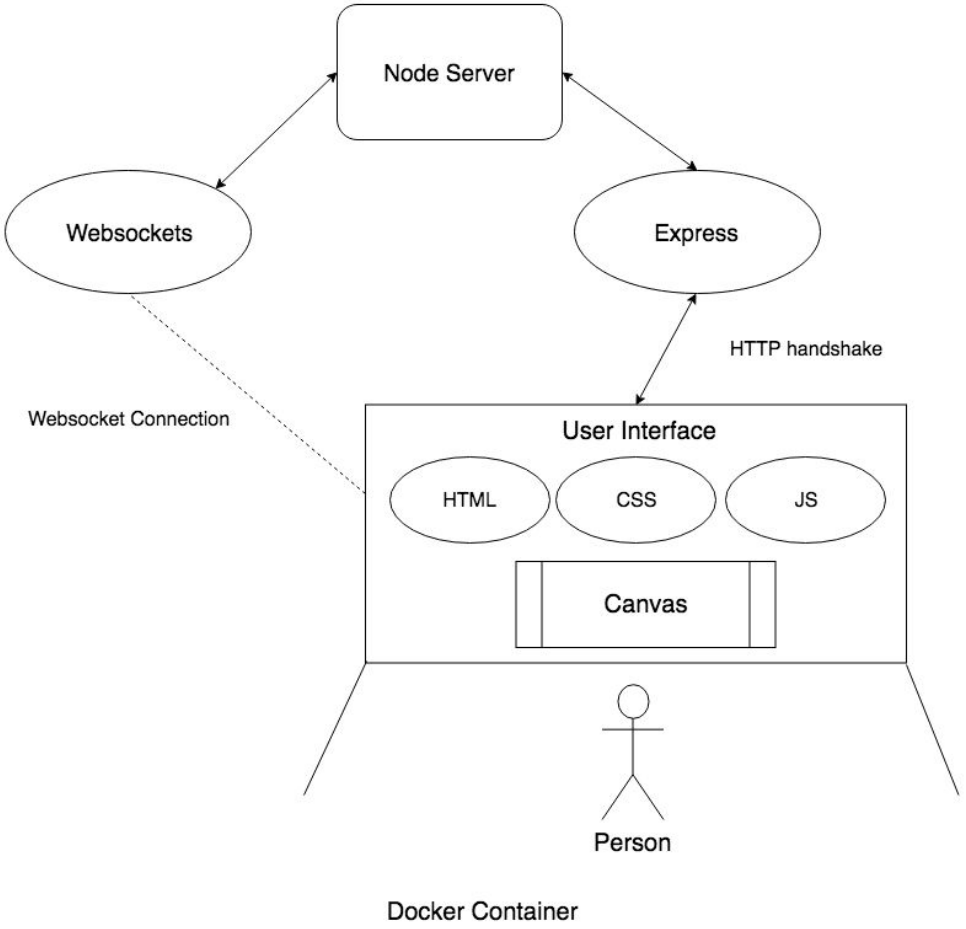
- Performant
- High quality graphics
- Hardware accelerated - Uses GPU or dedicated
- https://aframe.io/examples/showcase/helloworld/

Drawbacks

- Steep learning curve (for me)

Combining.

- Node server
- Express
- Websockets
- HTML
- CSS
- JS
- Docker



https://pure-ridge-30816.herokuapp.com