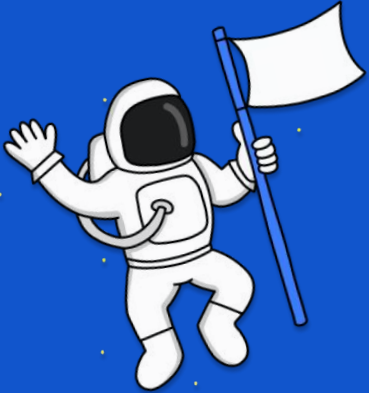


Introduction to Realtime Web Applications

Day 1 - 17 January 2021



Who are we



Low En
(@lowenhere)

SUTD ISTD Senior

Does software tings



Daryll Wong
(@daryllman)

SUTD ISTD Senior

Does tings

Logistics

Setup

For Day 1, make sure you have the following installed on your laptop:

- Git
- Node.js, along with a package manager (npm or yarn)
- A text editor / JavaScript IDE that you are comfortable with

For Day 2, you will need the following:

- A heroku account, and the heroku CLI installed
- Docker (optional)

Course Resources

Course Webpage (along with notes)

<https://realtime-apps-iap.github.io>

Github Repository

<https://github.com/realtime-apps-iap>

Other Resources / References

Here are a few resources that have been heavily referenced in the development of the workshop, mostly because they're really comprehensive and all around great:

Mozilla Developer Network (MDN)

<https://developer.mozilla.org/>

fireship.io YouTube channel

<https://www.youtube.com/channel/UCsBjURrPoezykLs9EqgamOA>

Agenda for Day 1

Introduction

JavaScript Primer / Refresher

Event-Driven Programming

Introduction to WebSockets

Tutorial: A Simple WebSocket Server (and Client)

Tutorial: A Group Chat Application

Introduction to WebRTC

Tutorial: A Video Calling Application

Realtime Web Application Development Challenge
Brief

Q&A

Agenda for Day 2

Review of Day 1

Tutorial: A Location-Sharing Application

A Brief Introduction to Docker

Deploying to Heroku

< Extra Bonus Content?>

Q&A

Introduction

Introduction to Realtime Web Applications

The real-time web is a network web using technologies and practices that enable users to receive information as soon as it is published by its authors, rather than requiring that they or their software check a source periodically for updates.

-Wikipedia, Real-time Web

JavaScript Primer / Refresher

JavaScript: An Introduction

The story so far:

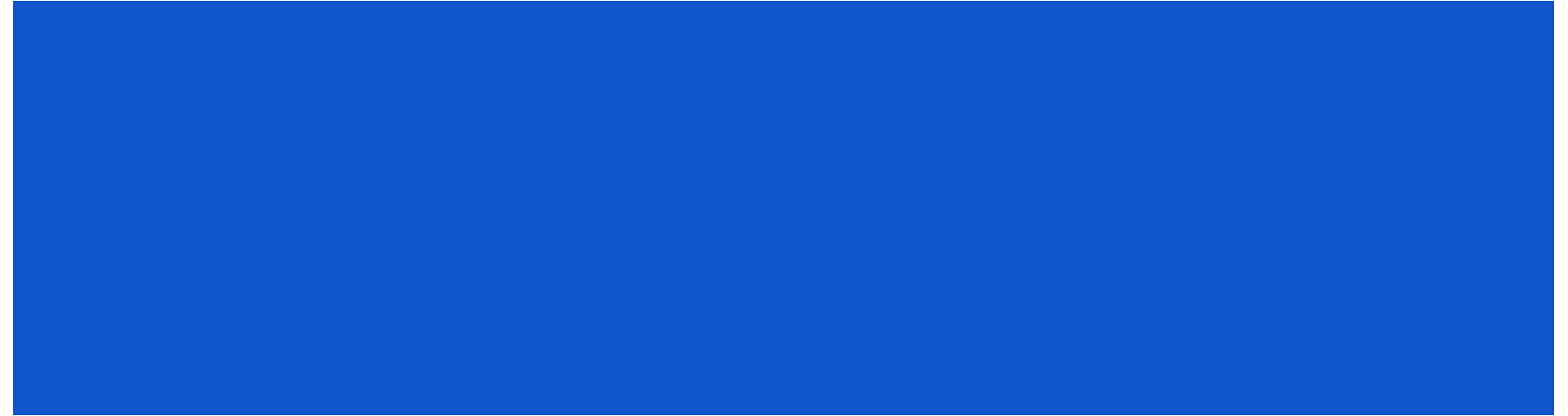
*In the beginning ~~the Universe~~ JavaScript was created.
This has made a lot of people very angry and been widely regarded as a bad move.*

-Douglas Adams, The Restaurant at the End of the Universe

JavaScript: Fun (?) Facts

- Created in 10 Days in 1995 by Brendan Eich, who later went on to co-found the Mozilla Foundation, and now runs Brave Software (the web browser company)
- The most popular language for projects on GitHub since 2014
- Nothing to do with the Java Programming Language

JavaScript: Client-Side JavaScript vs. Node.js



JavaScript: Client-Side JavaScript vs. Node.js

Similarities:

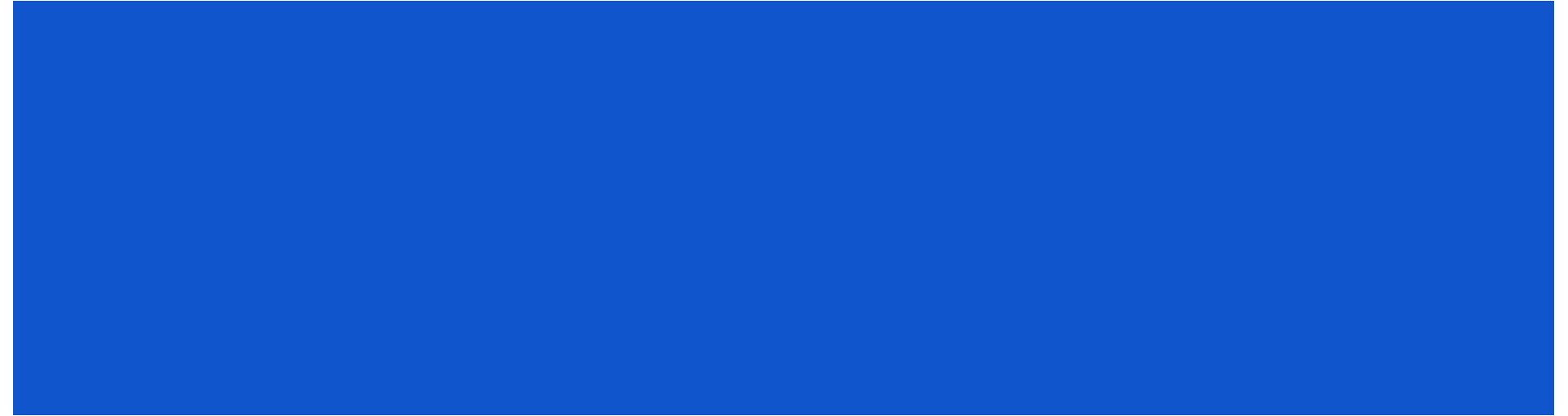
Both run JavaScript programs on a JavaScript Engine (like V8), which acts like an interpreter, translating JavaScript code into machine instructions (among other things)

Differences:

Global Object - On the client side, we have access to Web APIs via the window object

External Dependencies - Importing via `<script>` tags vs `require()` and `npm`

JavaScript: Language Features and some tricks



Short-Circuit Evaluation

Truthy and Falsy:

Falsy: `false`, `0`, `-0`, `0n`, `""`, `null`, `undefined`, and `NaN`

Truthy: all other values, e.g. `true`, `1`, `[]`, `{}`, `"hello"`

A Boolean OR statement returns the first value if it is truthy, else it returns the second value.

```
"Hello" || true //returns "Hello"  
"" || "there" //returns "there"
```

A Boolean AND statement returns the first value if it is falsy, else it returns the second value.

```
"Hello" && true //returns true  
"" && "there" //returns ""
```

var, let, const

var, let and const are three keywords that you can use to declare variables.

- var is the oldest method of declaring a variable (and the only way to do so before the ES6 specification)
- let also lets you declare a variable, but has a few key differences from var (among others)
- let is block scoped (a block is a series of statements bounded by curly braces {})
- const lets you declare a variable, but you will not be able to re-assign the variable (but if the variable is an object, properties of the object can still be updated)

Arrow Functions

Arrow Functions are a compact alternative to a typical function expression.

Instead of:

```
const square = function(a) {  
    return a * a  
}
```

we can write :

```
const square = (a) => { return a*a }
```

JavaScript: Promises



JavaScript: Async / Await



Event-Driven Programming

On the Client Side

```
1 <button id="myButton"></div>
```

```
1 document.getElementById("button").addEventListener("click", function(event){  
2     alert("the button has been clicked!")  
3 })
```

On the Server Side

```
1 app.get('/', function (req, res) {  
2     res.send('Hello there!')  
3 })
```

Introduction to WebSockets

WebSockets: An Internet Protocol

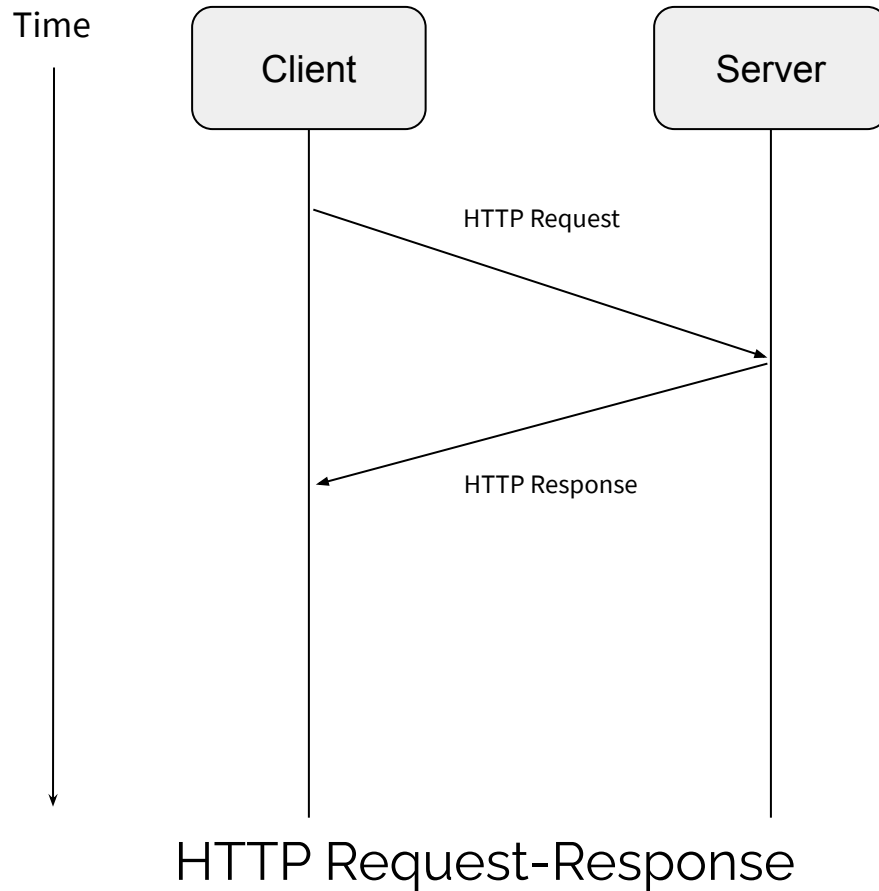
A protocol defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.

-Page 37, Computer Networking - A Top-Down Approach (7th Ed), James Kurose & Keith Ross

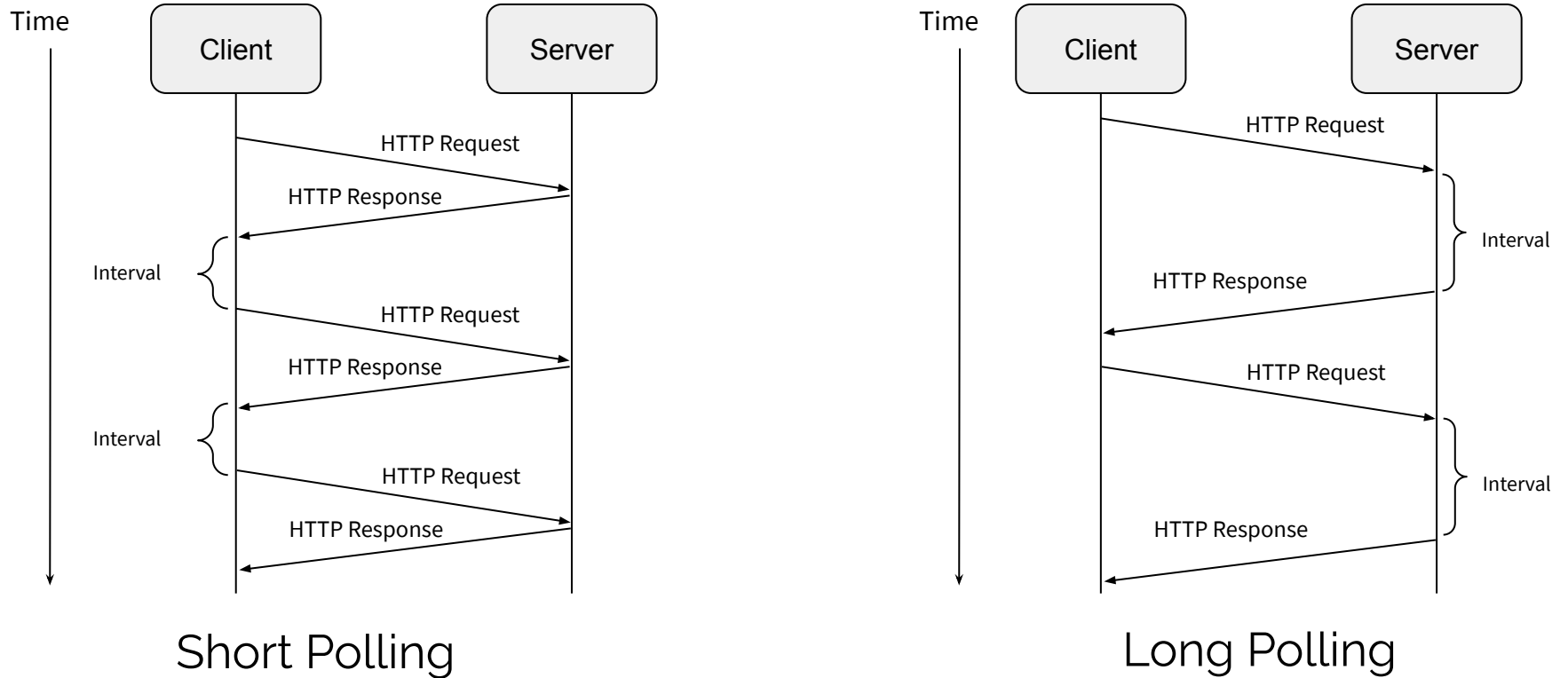
WebSockets and HTTP

Both WebSockets and HTTP are client-server protocols.

HTTP is a request-response protocol - the client sends a request message to the server, the server returns a response message



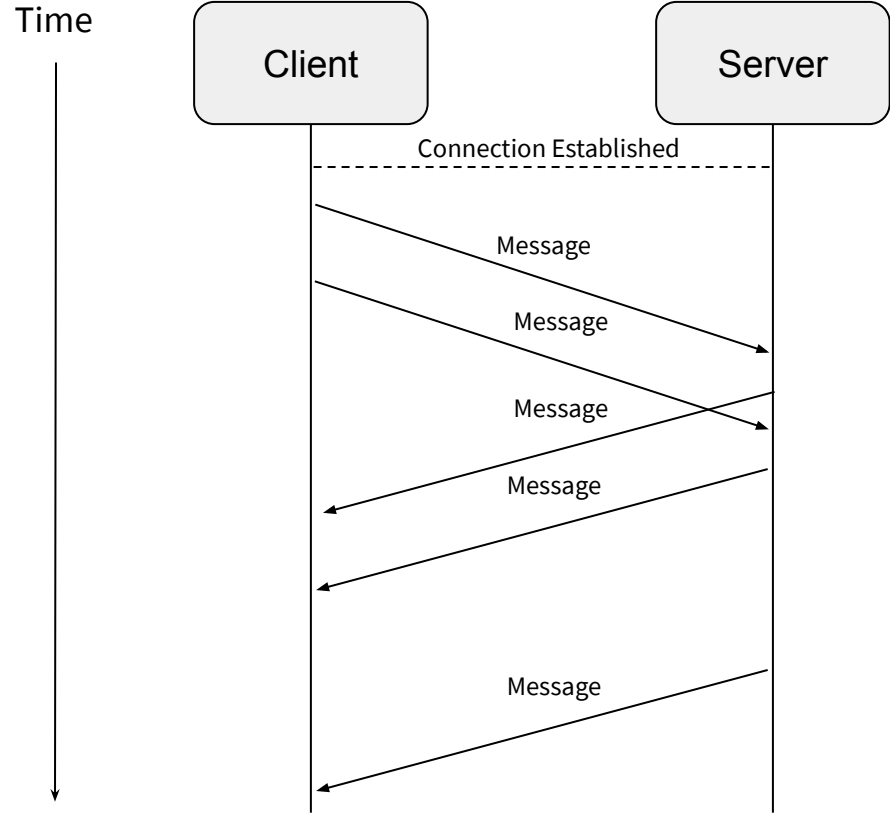
Some hacks: Short and Long Polling



WebSockets

WebSocket Connections:

- Full-duplex connection, meaning that the client and server can both send data along the connection at the same time.
- Data is sent as discrete messages. Message data has to either be in a string (USVString), or some kind of binary data type (ArrayBuffer, Blob, ArrayBufferView).



WebSocket Connection

Tutorial: Simple WebSocket Server (& Client)

<https://github.com/realtime-apps-iap/simple-websocket-server>

Tutorial: Group Chat Application

<https://github.com/realtime-apps-iap/chat-app-starter>

Introduction to WebRTC

WebRTC

Web Real Time Communications (WebRTC) is an open-source project that provides web browsers with with real-time communication via simple APIs.

It is a HTML5 specification you can use to add real time media communications directly between browsers

*----a robust set of APIs with versatile usage - mainly for video/audio call related applications**

What can you do with WebRTC

- *Realtime Video Call applications - GoogleMeet, GoogleHangouts, Discord, etc.*
- *Multiplayer games - Simple FPS, WebVR, etc.*

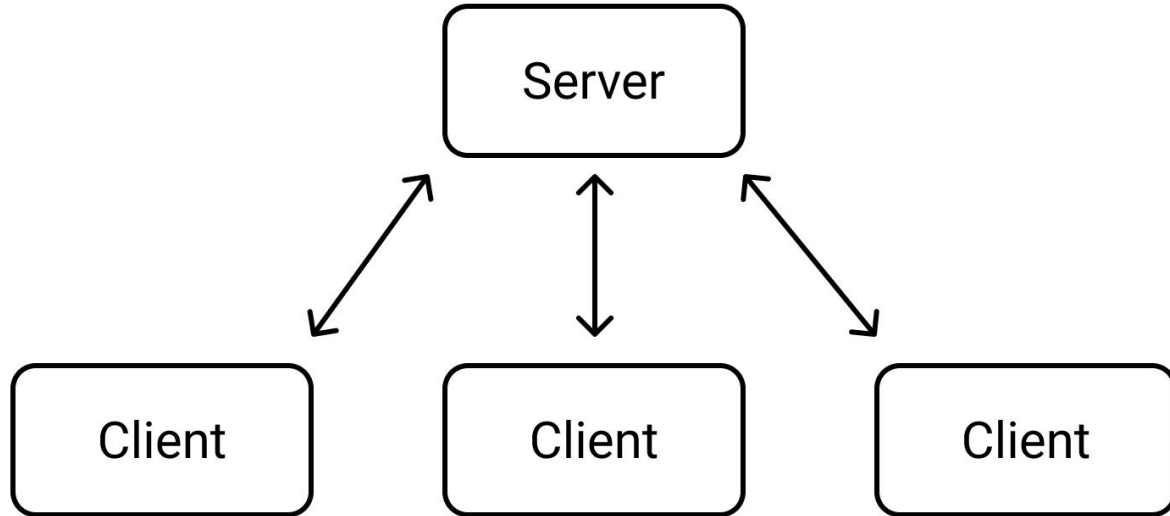
-> *If you realise...all these are mainly **audio or video applications***

WebSockets vs WebRTC

	WebSockets	WebRTC
Definition	WebSocket is a computer communications protocol, which provides communication channels over a single TCP connection	WebRTC is a open-source project that provides browsers and mobile apps with Real-Time communications capabilities via simple APIs
Latency	Low Latency	Near Real-time
Data Transfer	TCP	UDP
Uses	- Chat rooms, Location-based Apps, File Transfer, etc	- Streaming, Video calls, Multiplayer games, etc

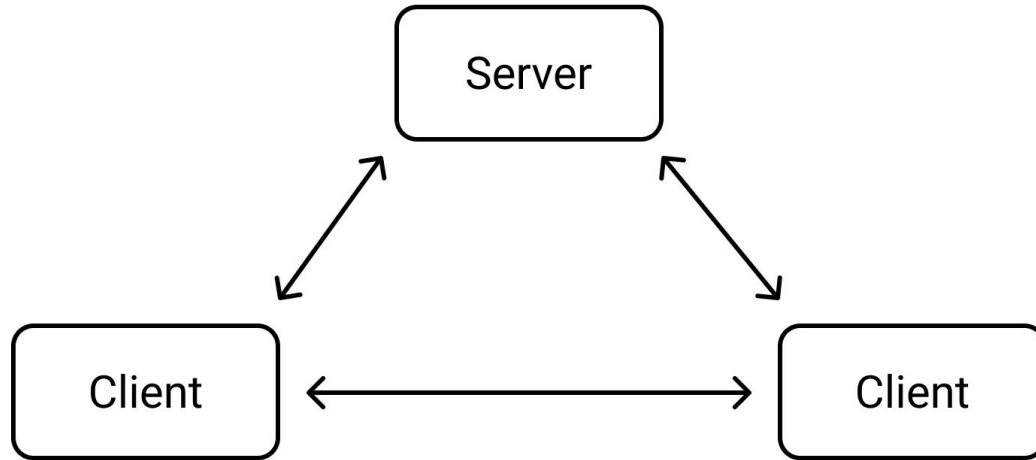
WebSockets vs WebRTC

WebSockets



WebSockets vs WebRTC

WebRTC



Other components in WebRTC...

- Signal processing to remove ambient noise from audio/video feed
- Codec handling to compress & decompress data
- Routing from one peer to another through firewalls, NATs, Relays to create an Interactive Connectivity Establishment (ICE)
- Encrypting user data before transmitting across connections
- Managing Bandwidth to user what each peer has to give
- Etc....

Challenges in WebRTC

- WebRTC uses UDP
 - Not a Reliable Data Transfer protocol, if this is important
- No standard signaling protocol
- Not fully compatible with all browsers
 - Modern Chrome, Firefox is good
 - May have issues with Edge, Safari - Extra plugins required

Why WebRTC

- Great for services that requires it to be real-time & fast
- No need for extra 3rd party apps
 - No need to install software
- Embedded in web technologies
 - Client, server can be run directly on the web browser
- Secure
 - Runs inside browser without creating new processes
 - Less chance for spyware, malware, etc
 - E.g. if you need camera, rights must be granted (because you are using browser technologies)

Tutorial: Video Call Application

<https://github.com/realtime-apps-iap/video-call-app>

Competition :)

<https://realtime-apps-iap.github.io/competition>

Main Course Website

<https://realtime-apps-iap.github.io>