

# Survey of WebRTC Architecture & Source Code

WebRTC - Western Illinois University Graduate Thesis

Dennis McMeekan: November 2020

## Abstract

This is a survey intended to provide a detailed analysis of a simple WebRTC application's architecture and source code. Through the breakdown of a WebRTC application, this will allow for a needed detailed explanation on how to advance with security protocol explanation and project implementation.

## WebRTC – Base Level

### Key.pem & Cert.Pem

These are keys that are pre-generated and are called inside generate\_cert.sh, a script that is able to generate a “secure” key when creating a peer connection.

### Generate\_cert.sh

This is the actual script that is able to generate SSL certification for authentication purposes.

### .gitignore

This is a file that is used when using Source Control resource, Git, to allow for intentionally untracking certain files.

### README.md

This is a file that allows for a project breakdown to occur and explain what the main course of action that was taken for each case implementation.

## License

The MIT License, along with Google's WebRTC copyright to ensure that proper credit is given when needed, and that the code is open source along as it is not used for monetary gains or promotion.

## JSON Packages

Both are intended to call and package appropriate files upon run time. One file being intended to gather each module that is needed, which for this project implementation those are not currently used. While another file is able to run certain commands and actually call server.js to begin providing a peer-to-peer connection.

## Server

### Server.js

This creates a very simple WebSocket server, also using https to ensure that a peer-to-peer connection can be established locally. This provides a basis on being able to judge statistics and measurements among implementation of our security protocols.

## Client

### Index.html

This is the html page that is displayed to each client, when going to <https://localhost:8443/>. This webpage allows for the ability to display certain information, statistics, along with video and canvas elements which will allow for our security protocols to take effect. This is done by then calling *WebRTC.js* to allow for the JavaScript behind the scenes to take place.

## WebRTC.js

This is the code that is being called to create a “WebRTC Instance.” This will allow for WebRTC Architecture, security, and project manipulation, and has many methods or functions involved in doing so.

## Modules

Although these are not particularly used throughout the security implementation and project work, these modules are made available for use. Essentially, these act as add-ons to allow for more capabilities and features. WS is the only one that is highly used for our project work.

### Async-Limiter

Module should the absolute minimum work necessary to queue up functions.

### Safe-Buffer

Drop-In replacement for Buffer.

### Ultron

High-Intelligence robot to help remove event emitters.

### WS

WebSocket client and server implementation, at a global level.

## Resources

### Simple Open Source Introduction Implementation

S. Tulley, "A Dead Simple WebRTC Example," *GitHub*. Accessed on: Oct. 1, 2020. [Online]. Available:

<https://shanetully.com/2014/09/a-dead-simple-webrtc-example/>

### Understanding API and Source Code

S. Dutton, "Get Started with WebRTC," *HTML5Rocks*. Accessed on: Oct 10, 2020. [Online] Available:

<https://www.html5rocks.com/en/tutorials/webrtc/basics/>