1. Someone must getUserMedia() - CLIENT1/Init/Caller/Offerer
2. CLIENT1 creates RTCPeerConnection
3. peerConnection needs STUN servers
    - we will need ICE candidates later
4. CLIENT1 add localstream tracks to peerConnection
    - we need to associate CLIENT1 feed with peerConnection
5. CLIENT1 creates an offer
    - needed peerConnection with tracks
    - offer = RTCSessionDescription
        1. SDP - codec/resolution information
        2. Type (offer)
6. CLIENT1 hands offer to (peer connection) pc.setLocalDescription
    - scripts.js
    ```js
        const offer = await peerConnection.createOffer();
        peerConnection.setLocalDescription(offer)
    ```

~7. ICE candidates can now start coming in (ASYNC)

SIGNALING (someone to help the browser find/talk to each)

8. CLIENT1 emits offer
    - socket.io server holds it for the other browser
    - associate with CLIENT1

~9. Once 7 happens, emit ICE c. up to signaling server
    - socket.io server holds it for the other browser
    - associate with CLIENT1

CLIENT1 and Signaling server wait.
- wait for an answerer/CLIENT2/reciever

10. CLIENT2 loads up the webpage with io.connect()
    - a new client is connected to signaling/socket.io server
11. socket.io emit out the RTCSessionDesc to the new client
    - an offer to be sent!
12. CLIENT2 runs getUserMedia()
13. CLIENT2 creates a peerConnection()
    - pass STUN servers
14. CLIENT2 adds localstream tracks to peerconnection
15. CLIENT2 creates an answer (createAnswer());
    - createAnswer = RTCSessionDescription (sdp/type)
16. CLIENT2 hands answer to pc.setLocalDescription
17. Because CLIENT2 has the offer, CLIENT2 can hand the offer to pc.setRemoteDescription

~18. when setLocalDescription, start collecting ICE candidates (ASYNC)

Signaling server has been waiting...

19. CLIENT2 emit answer (RTCSessionDesc - sdp/type) up to signaling server

~20. CLIENT2 will listen for tracks/ICE from remote.
- and is done.
- waiting on ICE candidates
- waiting on tracks

21. signaling server listens for answer, emits CLIENT1 answer (RTCSessionDesc - sdp/type)
22. CLIENT1 takes the answer and hands it to pc.setRemoteDesc

~23. CLIENT1 waits for ICE candidates and tracks

21 & 23 are waiting for ICE. Once ICE is exchanged, tracks will exchange