

## CSCI 4140 – Tutorial 7

# Assignment 2 Overview (Server Side)

Matt YIU, Man Tung ([mtyiucse](mailto:mtyiucse@gmail.com))

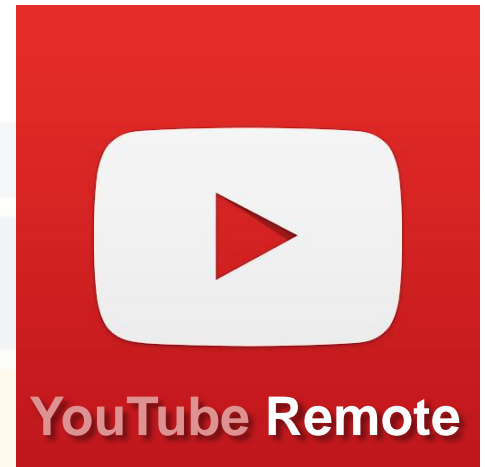
SHB 118

*Office Hour:* Tuesday, 3-5 pm

2015.03.05

# Outline

- Work flow
- Server side
  - Routing
  - Message forwarding
  - Retrieving video title

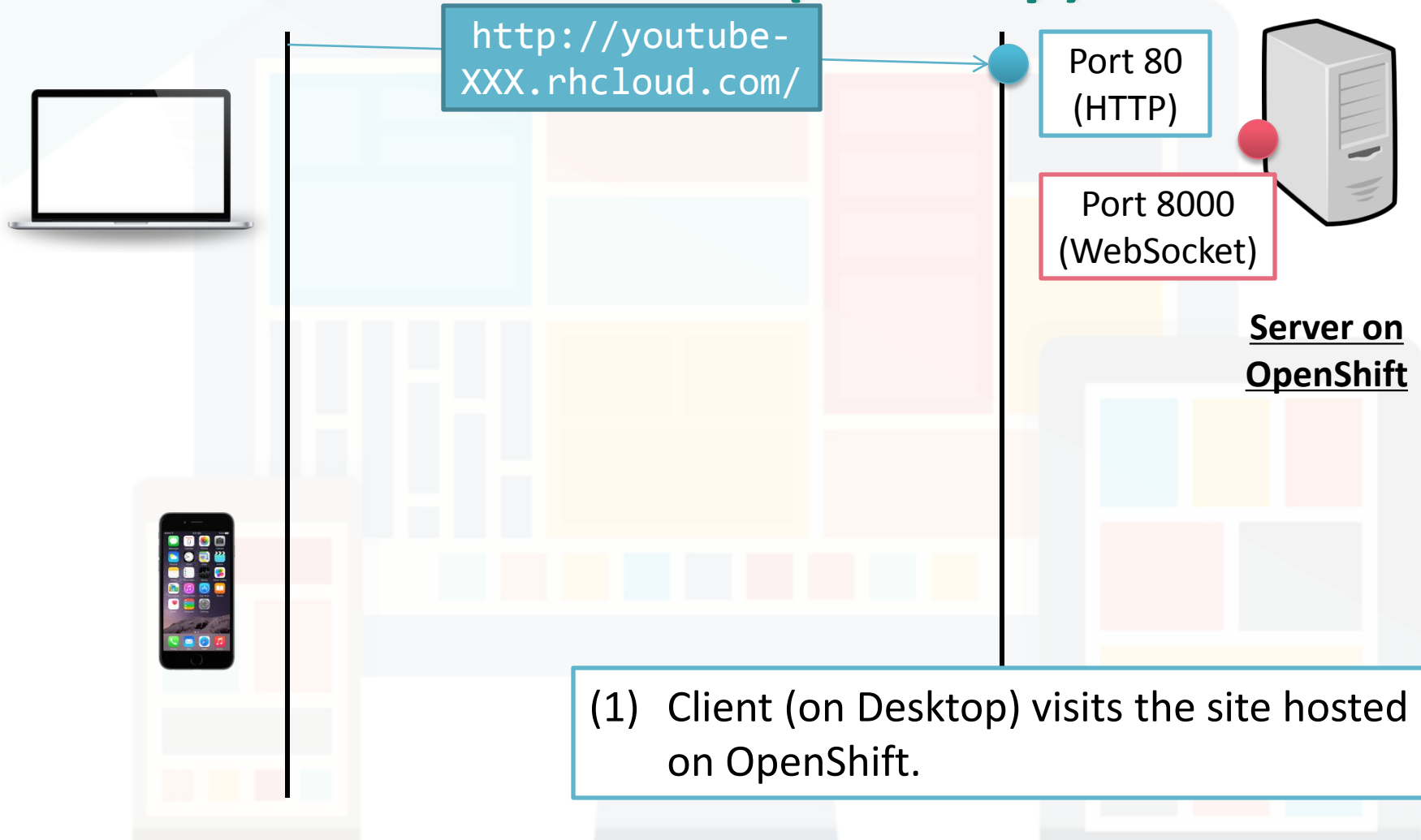




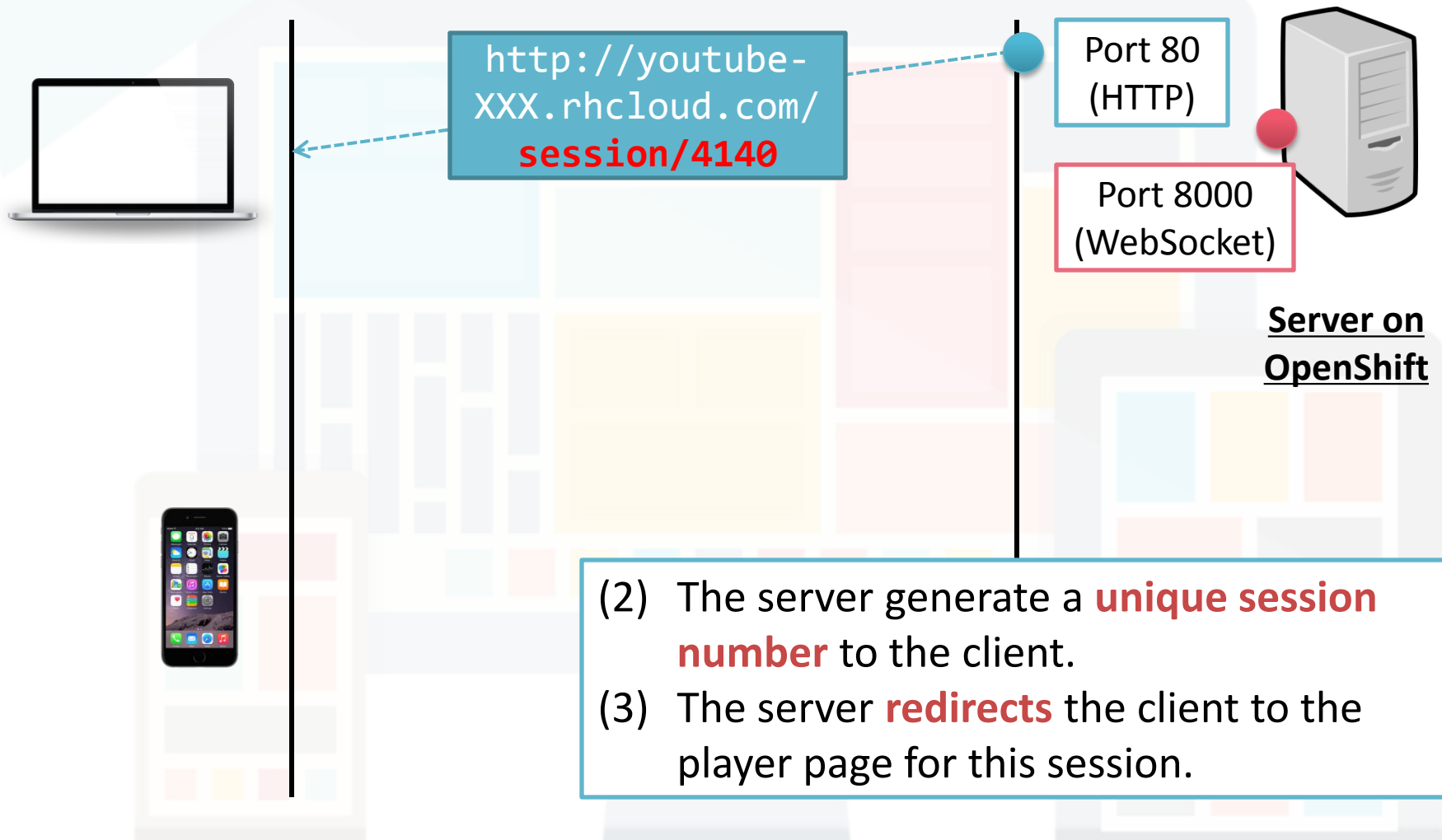
# Work flow

*Let's define what will happen between the clients and server...*

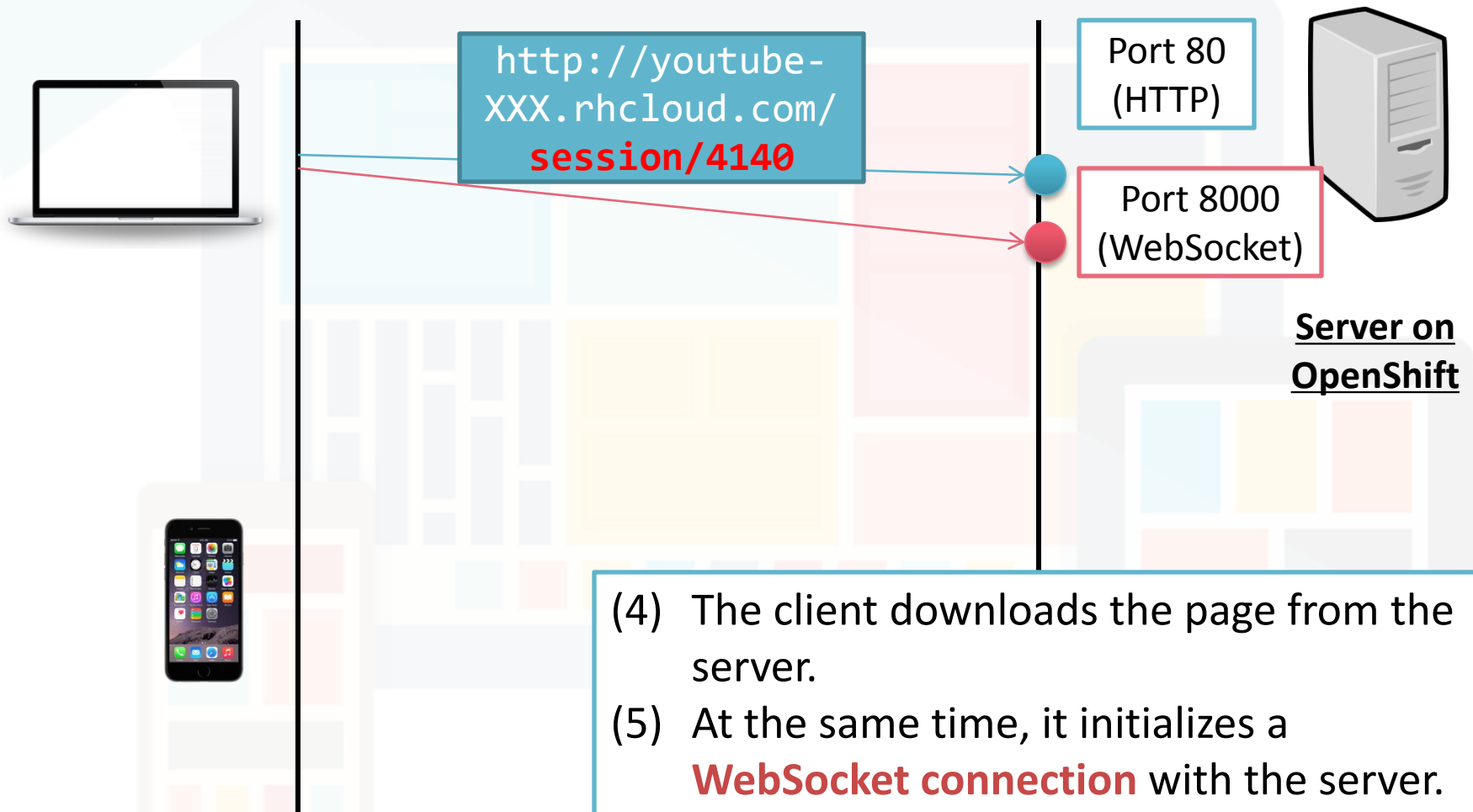
# Work flow: Initialization (Desktop)



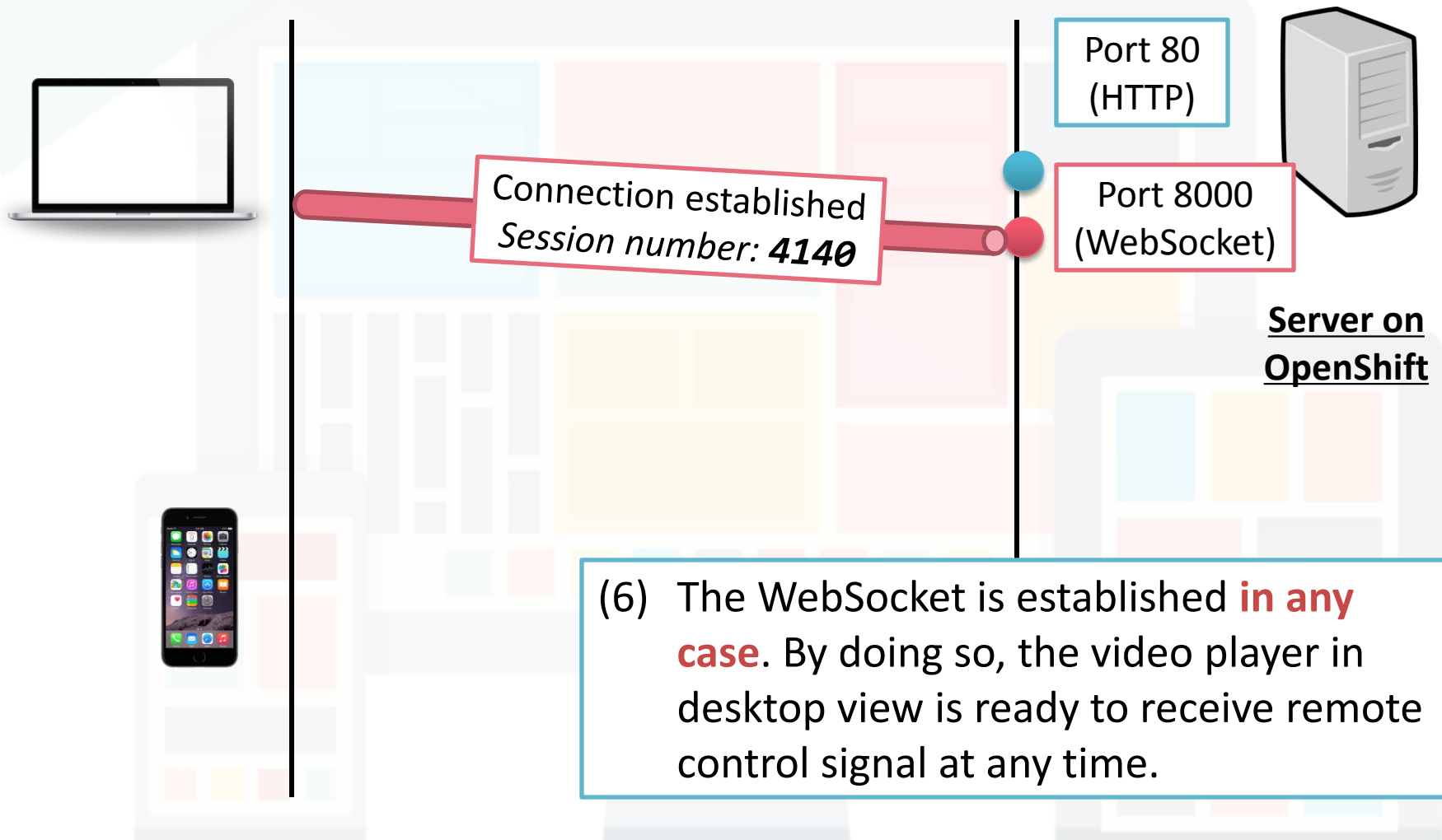
# Work flow: Initialization (Desktop)



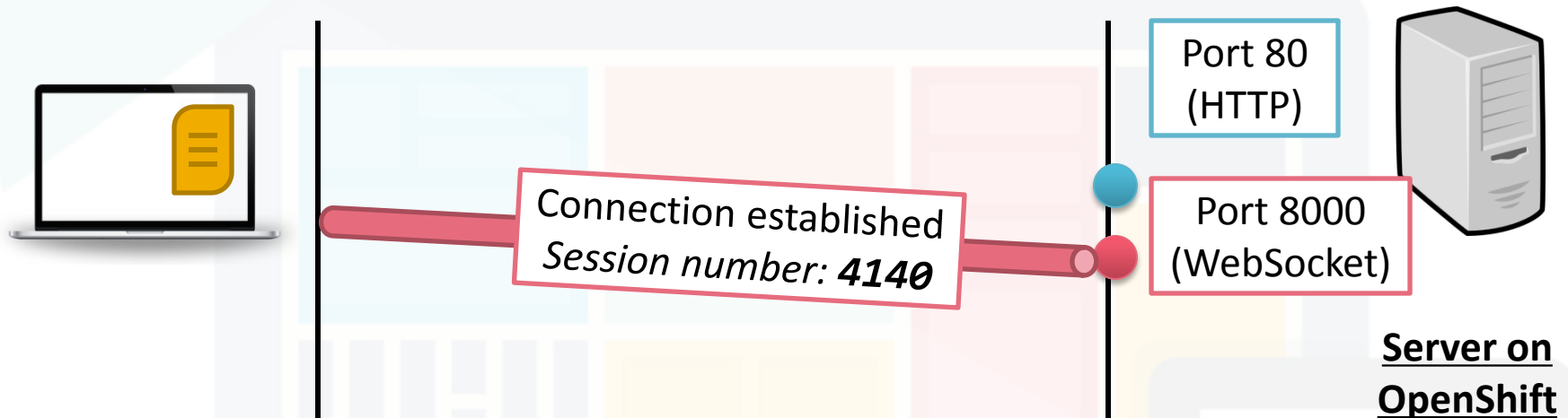
# Work flow: Initialization (Desktop)



# Work flow: Initialization (Desktop)



# Work flow: Initialization (Desktop)

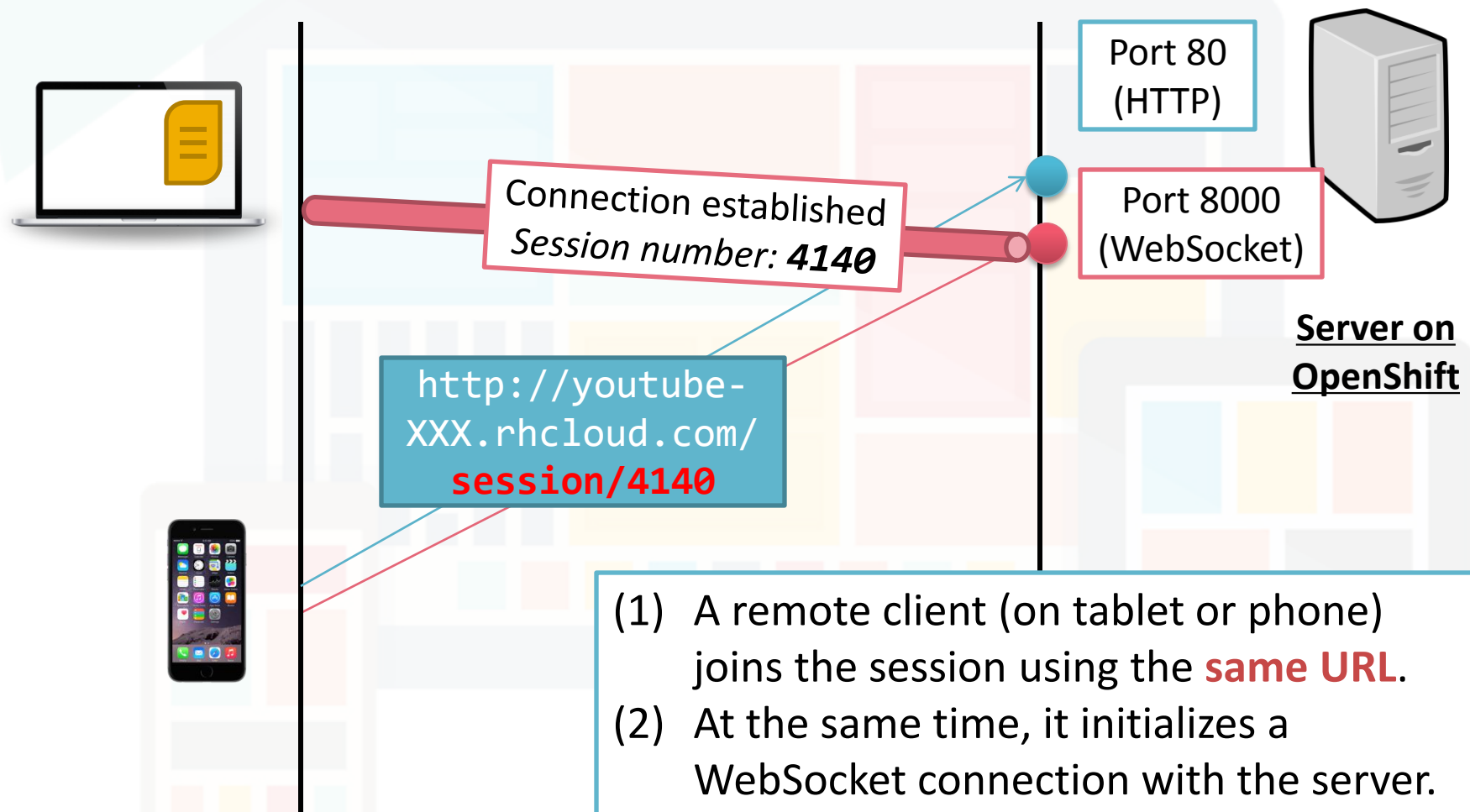


(7) The desktop client retrieves the playlist in the following order:

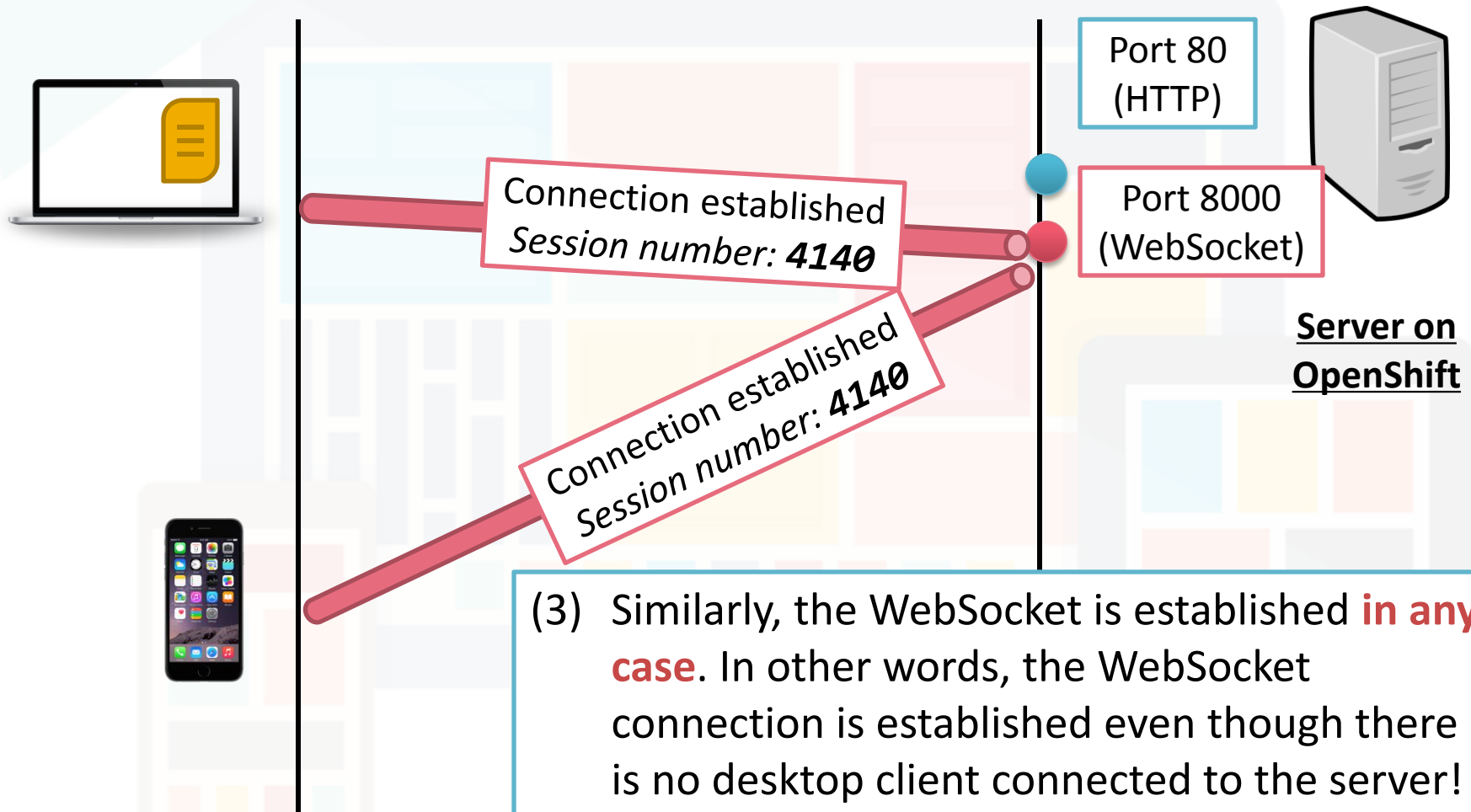
- **Synchronize** the playlist with the other client in the **same session**;
- If there are no existing clients, it reads the playlist stored in the **local storage**, provided that the desktop client has used the application before (not necessary with the same session number);
- If there are nothing stored in the local storage, generate an **empty playlist**.



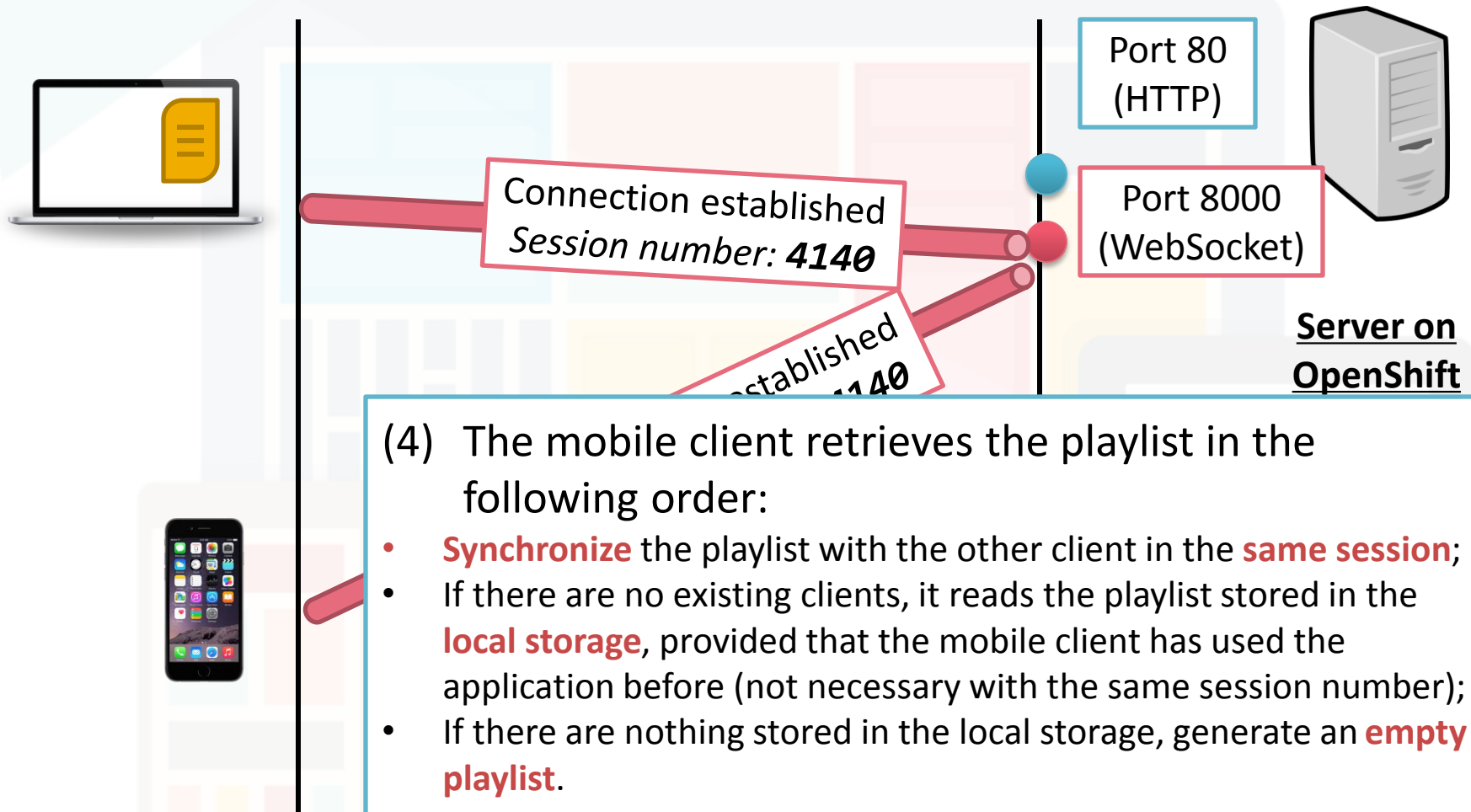
# Work flow: Initialization (Tablet or phone)



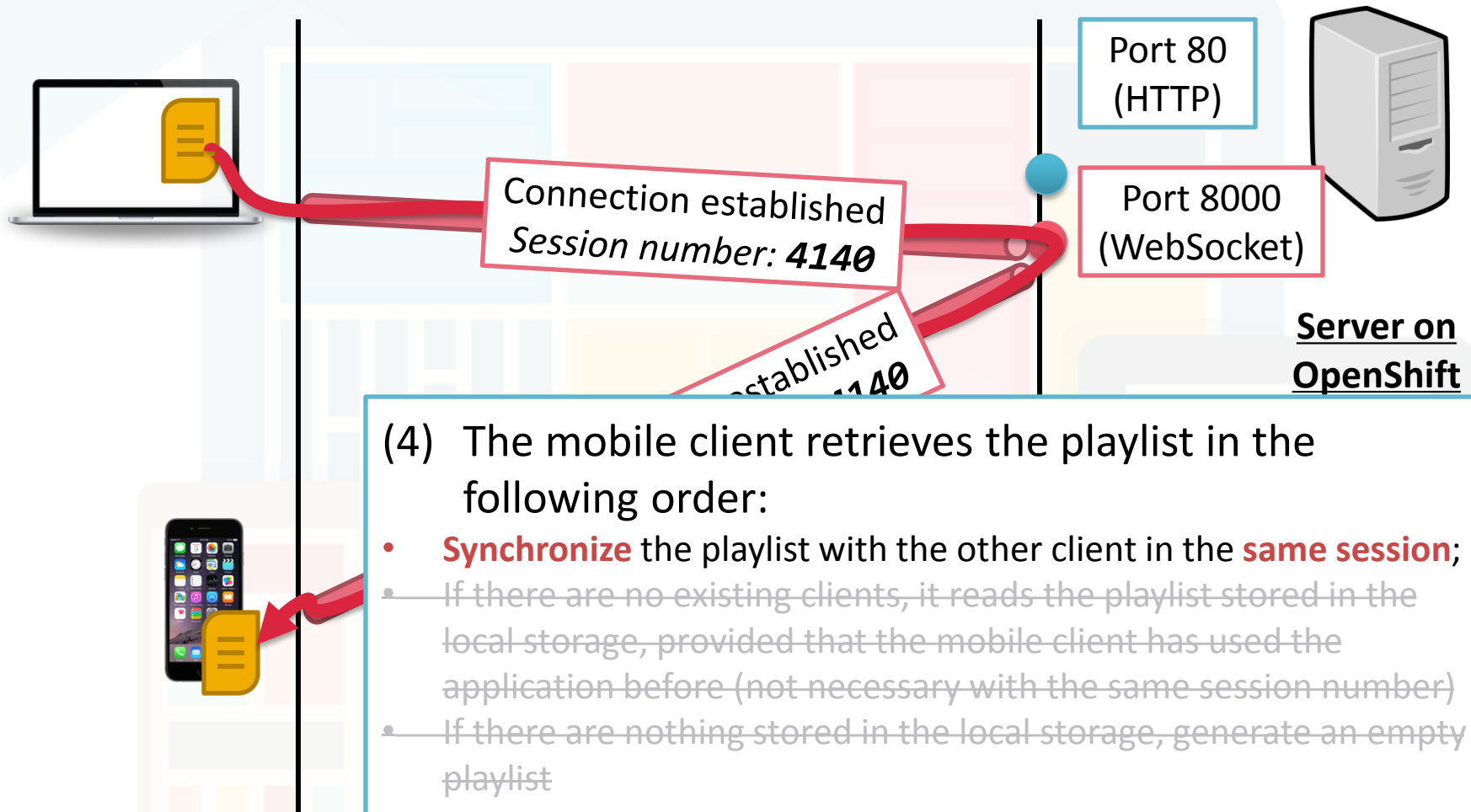
# Work flow: Initialization (Tablet or phone)



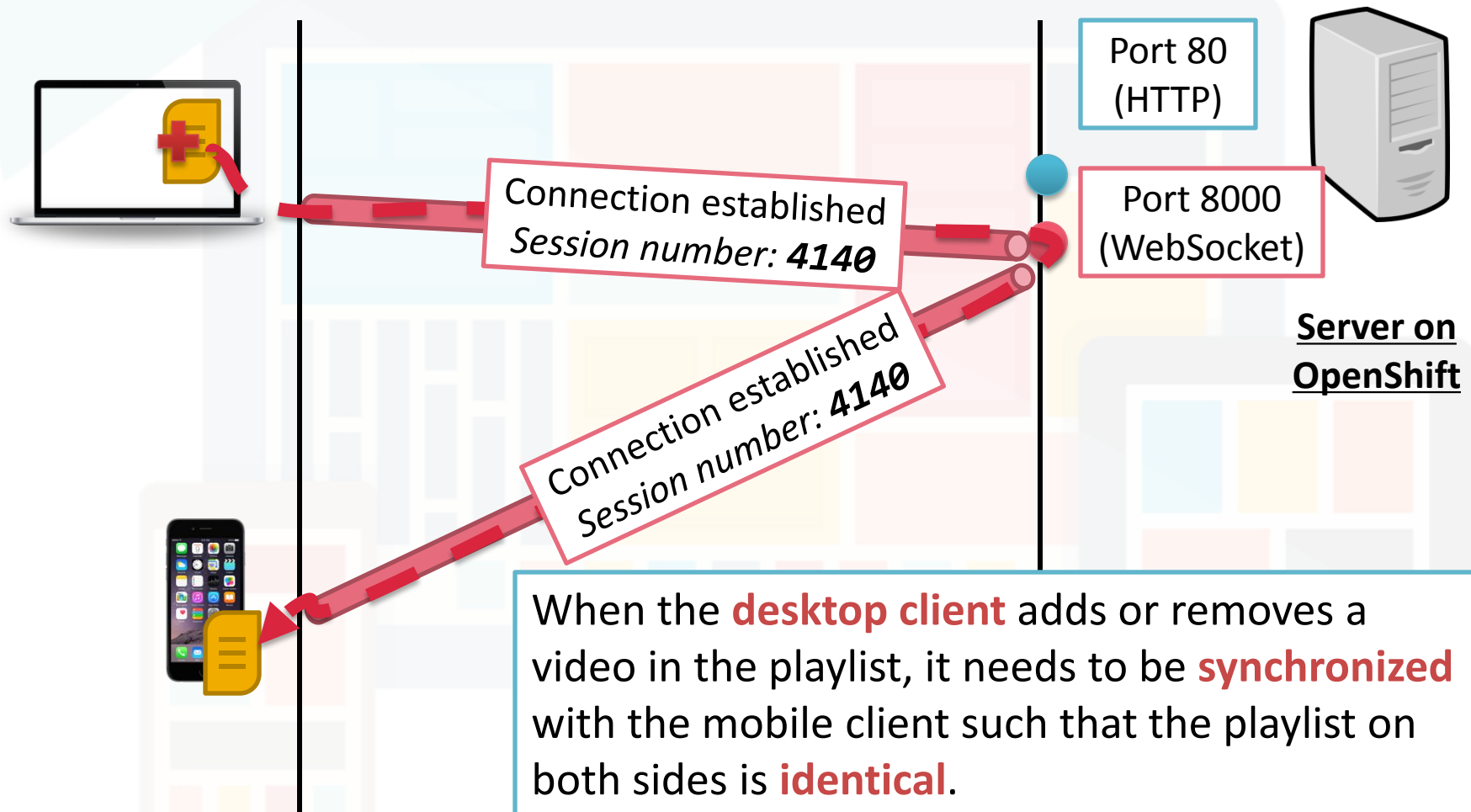
# Work flow: Initialization (Tablet or phone)



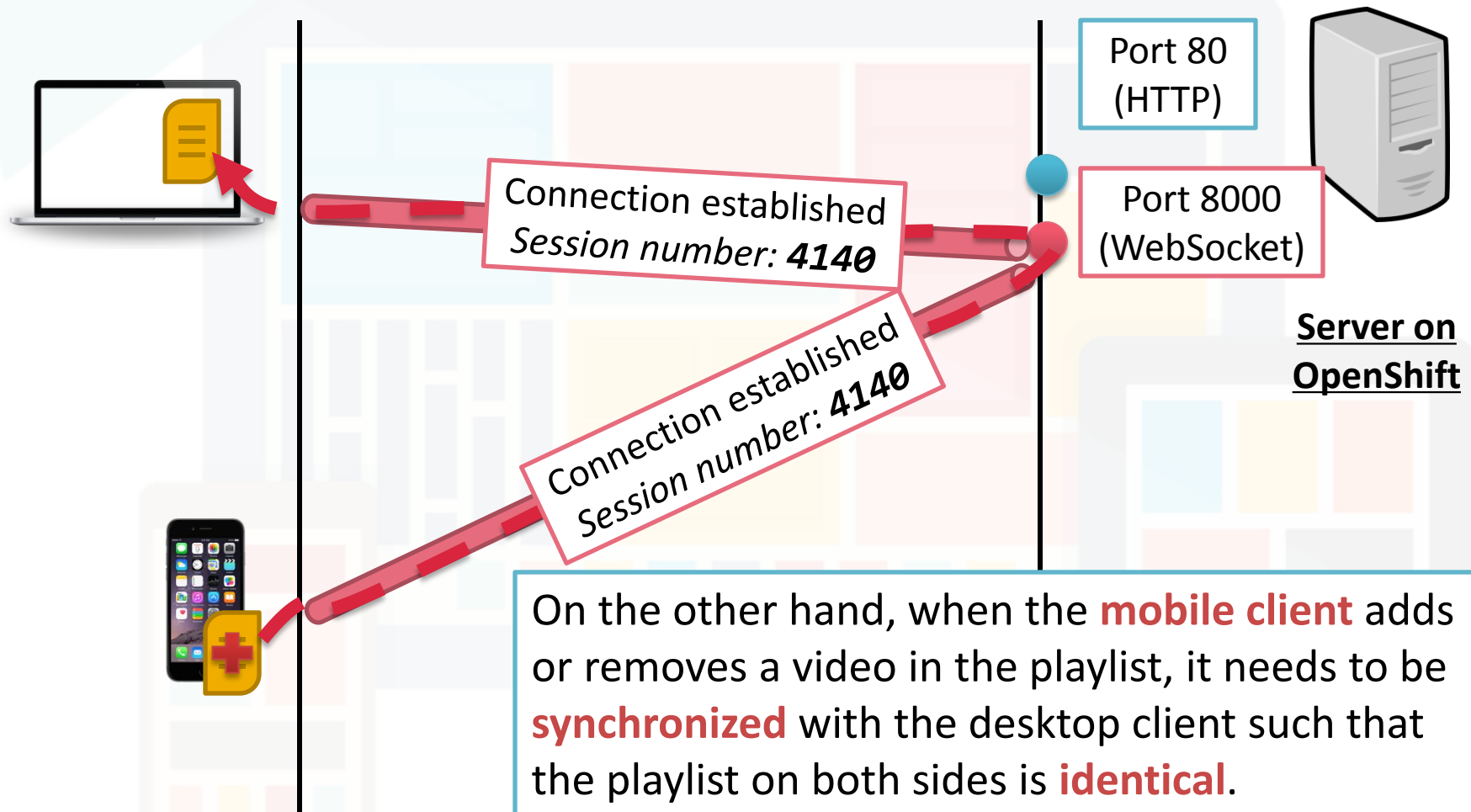
# Work flow: Initialization (Tablet or phone)



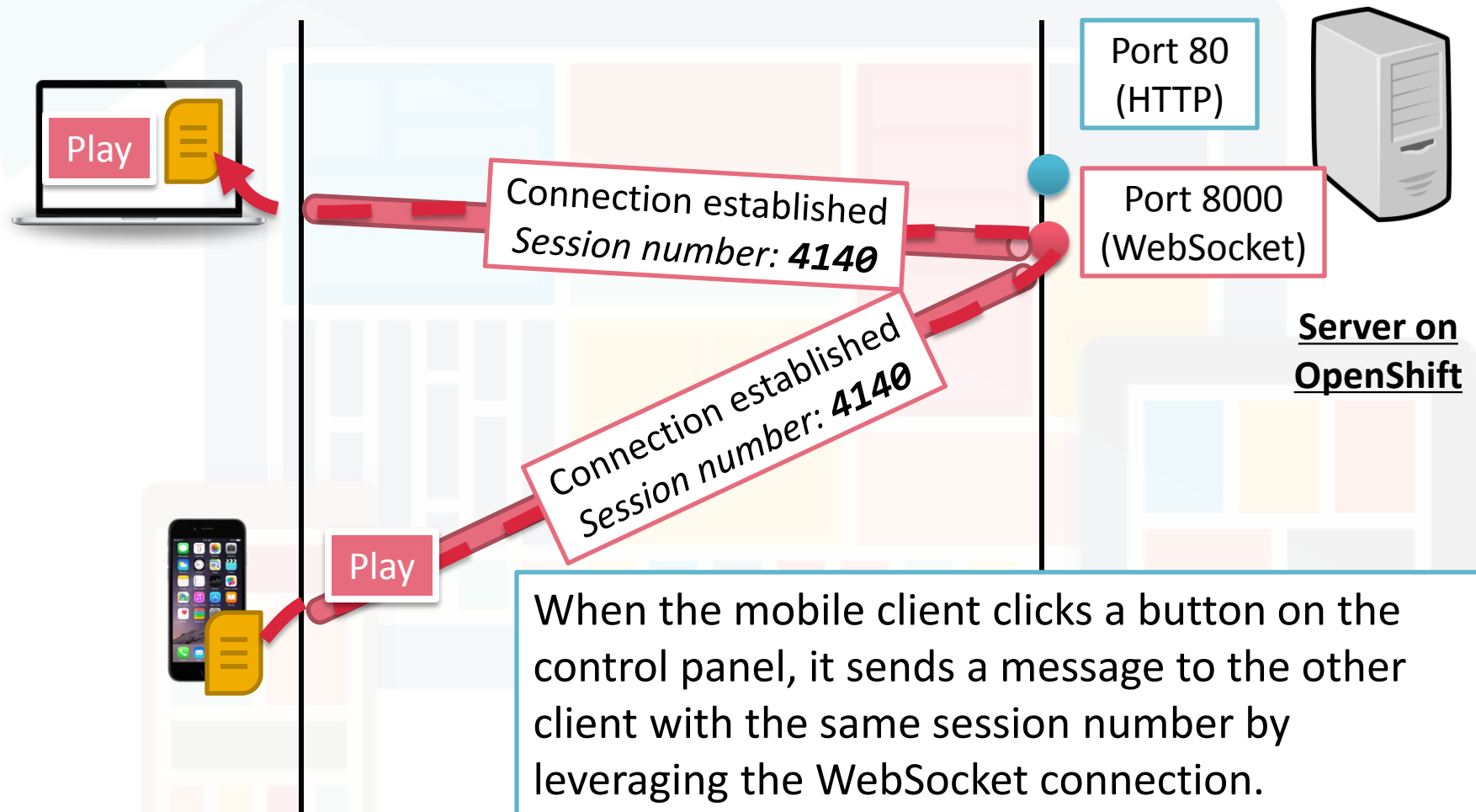
# Work flow: Playlist synchronization



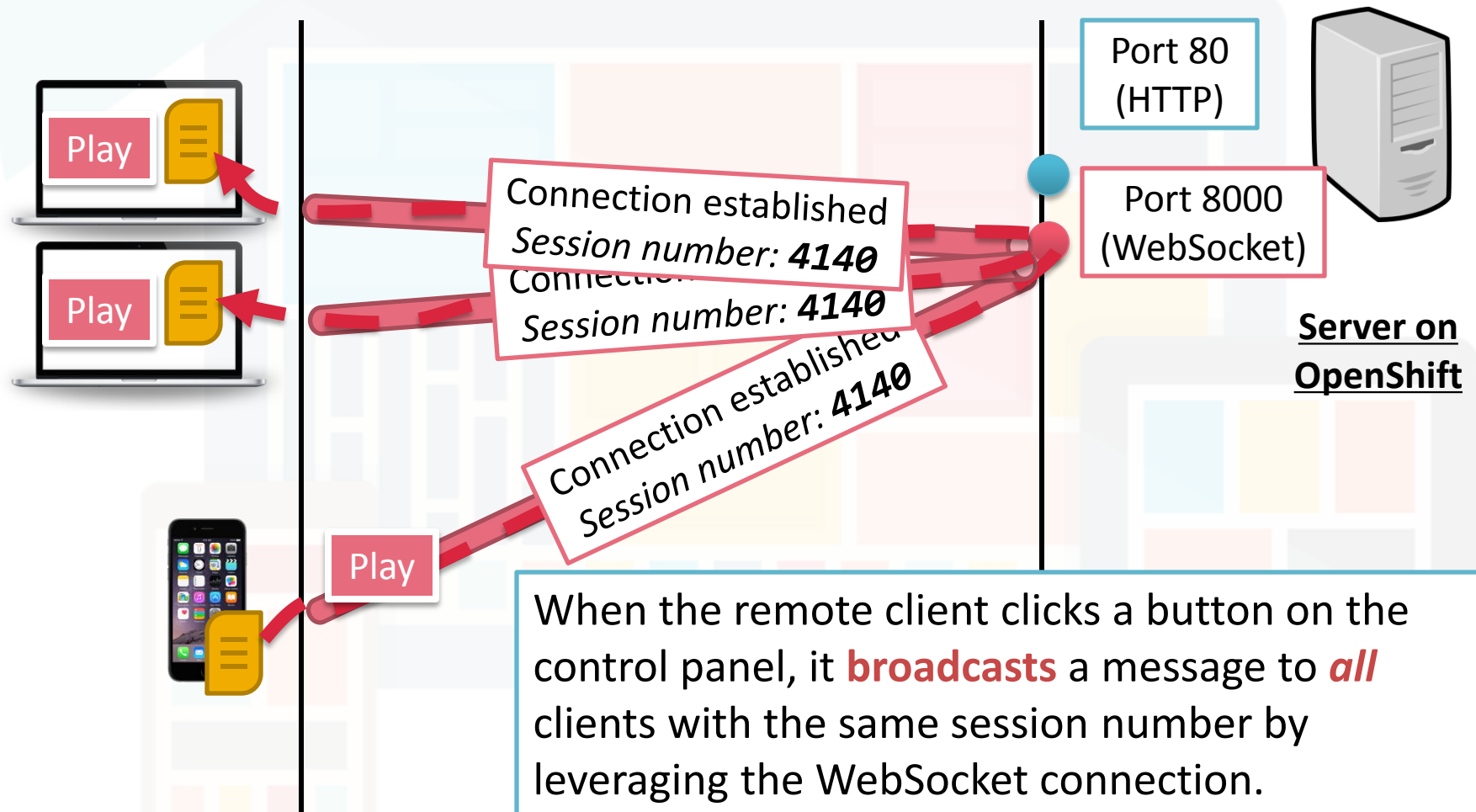
# Work flow: Playlist synchronization



# Work flow: Remote control

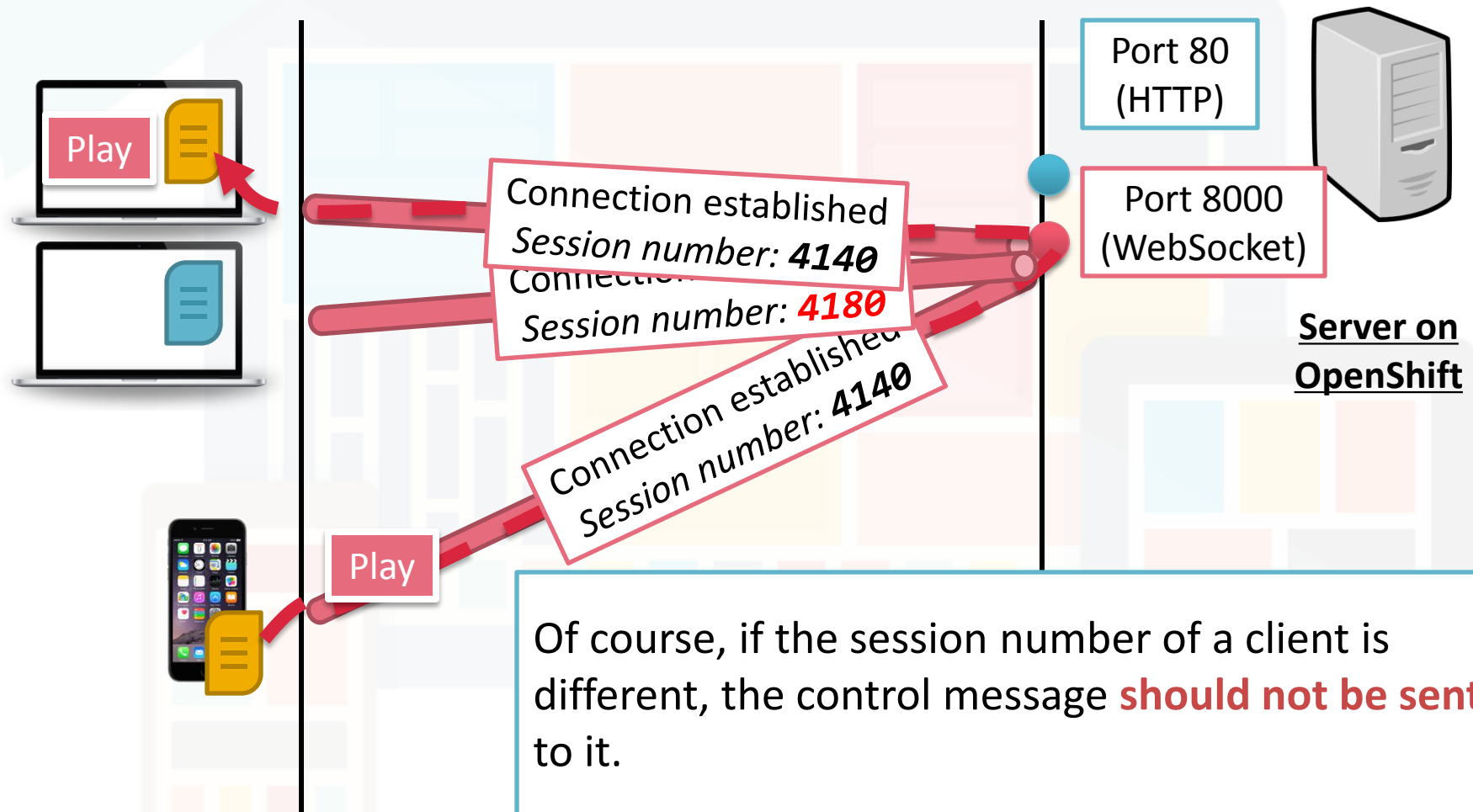


# Work flow: Remote control

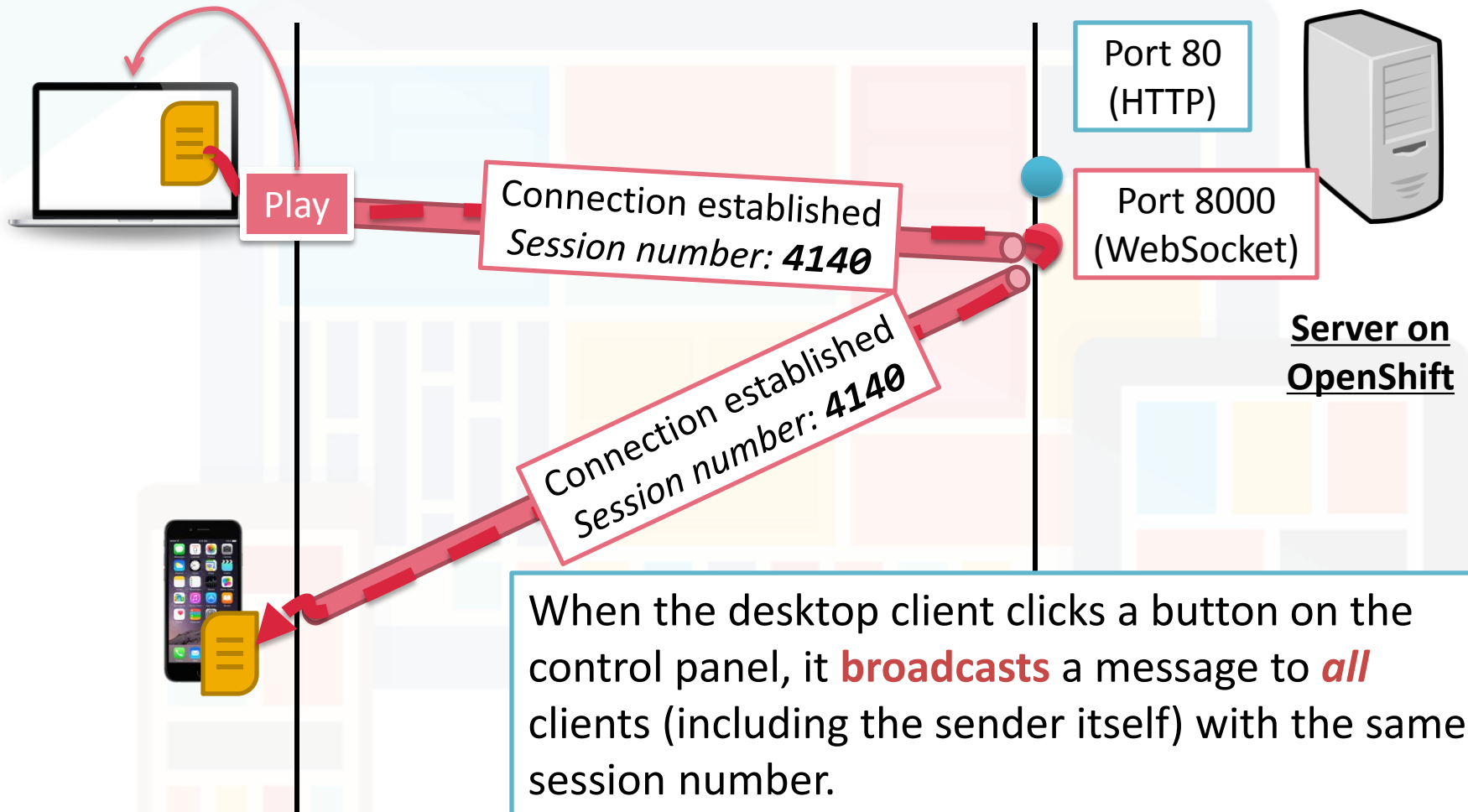




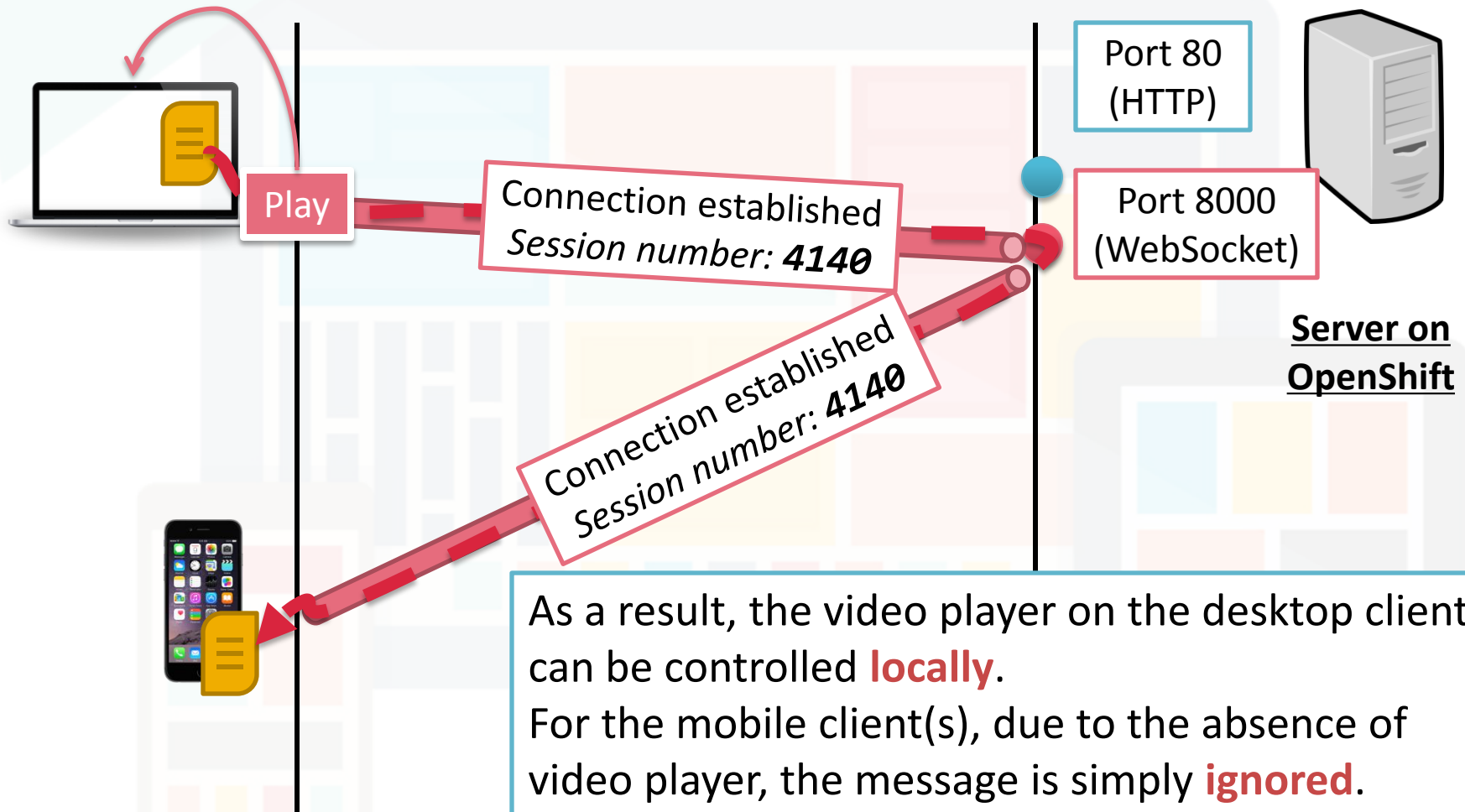
# Work flow: Remote control



# Work flow: Remote control



# Work flow: Remote control





# Server side

*Routing, message forwarding & retrieving video title*

# Server side: Routing

Client enters the page without session number  
(e.g., <http://youtube-XXX.rhcloud.com/>)

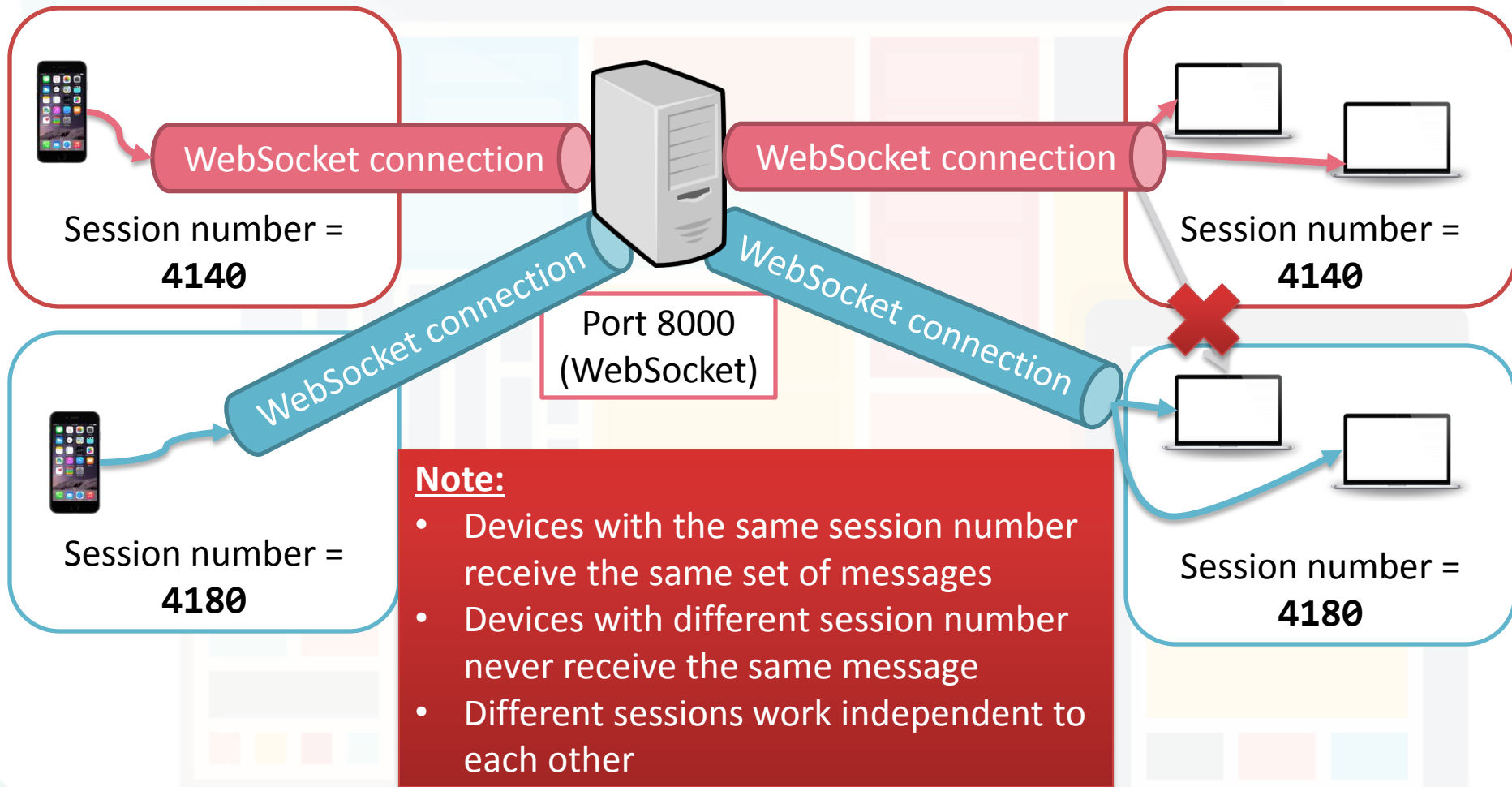


Server generates a unique session number



Server redirects the client to the page representing this session  
(e.g., <http://youtube-XXX.rhcloud.com/session/4140>)

# Server side: Message forwarding



## Server side: Retrieving video title

- You need to display the **video title** for each video in the playlist
- How to retrieve video title from an ID?

```
http://www.youtube.com/oembed?  
url=http://www.youtube.com/watch?v=sQANr6r4Kms
```

Put the video ID here

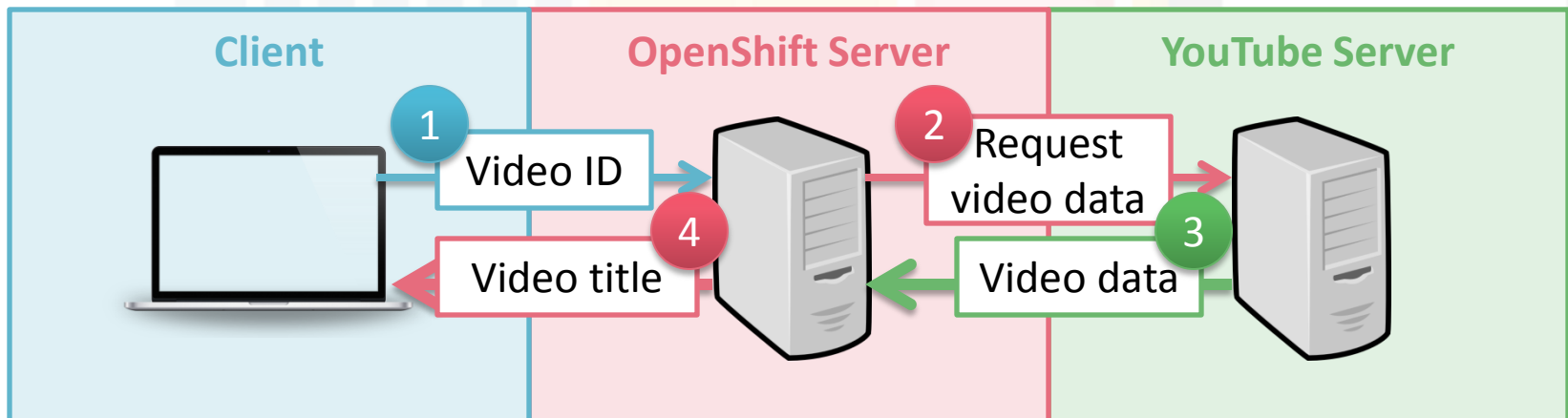
- The server returns a **JSON object**
  - The video title is returned inside the “**title**” field

```
{"title": "CSCI4140 Ruby on Rails for Dummies",  
 "height": 344, "width": 459, ...}
```

- Can we use AJAX to get this JSON object directly from YouTube?
  - No because of the **same-origin policy (SOP)**
  - How to bypass this?

## Server side: Retrieving video title

1. **Client** send a request with the video ID to **OpenShift server**
2. **OpenShift server** forwards the request to **YouTube server**
3. **YouTube server** responds with the video data
4. **OpenShift server** replies the client with the video title





# Reminder

- You should finish the **UI design, player** and **control logic**
- You should start implementing the **server side**
  - We will cover **Node.js** in this tutorial
  - We will cover **Socket.IO** in next tutorial
  - In other words, you can complete Assignment 2 **after next tutorial**
- Try to deploy your Node.js application on **OpenShift**

– End –