2 November 2016

WebRTC

Björn Helgeson
Prototyp Stockholm AB

Björn Helgeson
bhelg@kth.se

Bingolabs

PROTOTYP

Today 09-17

1. Tech Introduction
2. Tasks 1-4
3. Do something new!

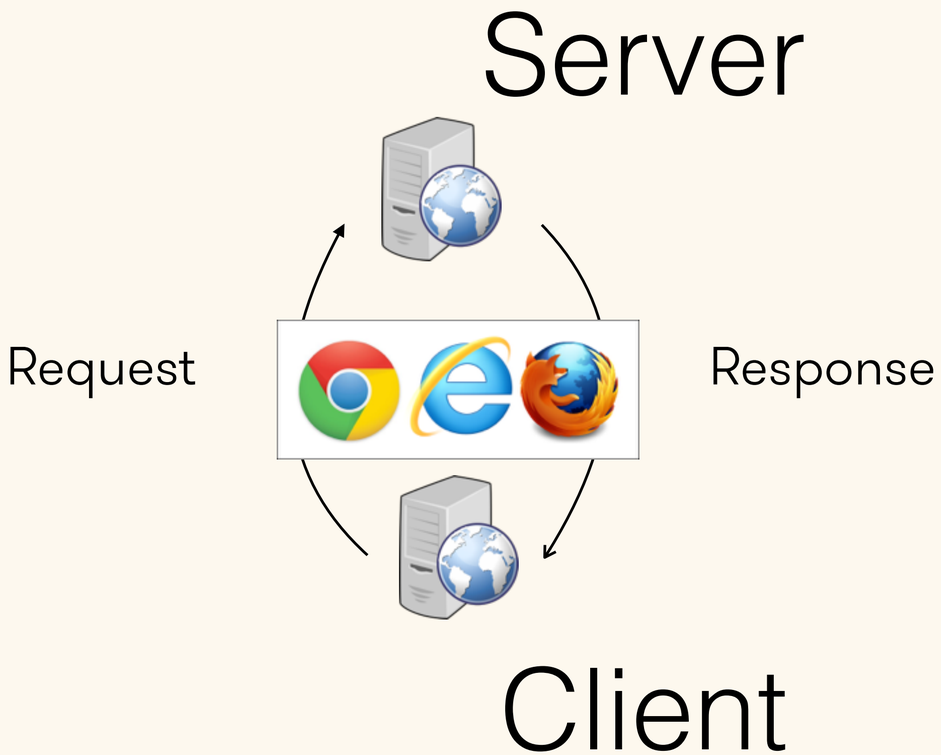
This workshop

```html
<!DOCTYPE html>
<html>
<head>
<title>My home page</title>
</head>

<body>
<h1>Welcome!</h1>
<p>This is my website</p>
</body>

</html>
```

What's this?

```
var fruitSalad = ["Banana", "Apple", "Strawberry", "Mango"];
var len = fruitSalad.length;
var text = "This fruit salad contains: ";

for (var i = 0; i < len; i++) {
    text = text + fruitSalad[i];
}
```

Web browsers



The Web & HTML5

- Traditional web technologies
  – HTML, Javascript & CSS
- New web technologies
  – HTML5, WebRTC

Enabling powerful browser applications
without the need to install additional software

HTML5 Features

- Local Storage
- Better video & audio support
- New elements like <video>, <audio> and <canvas>
- Support for Geolocation, Offline Web Applications, WebSockets
- WebRTC

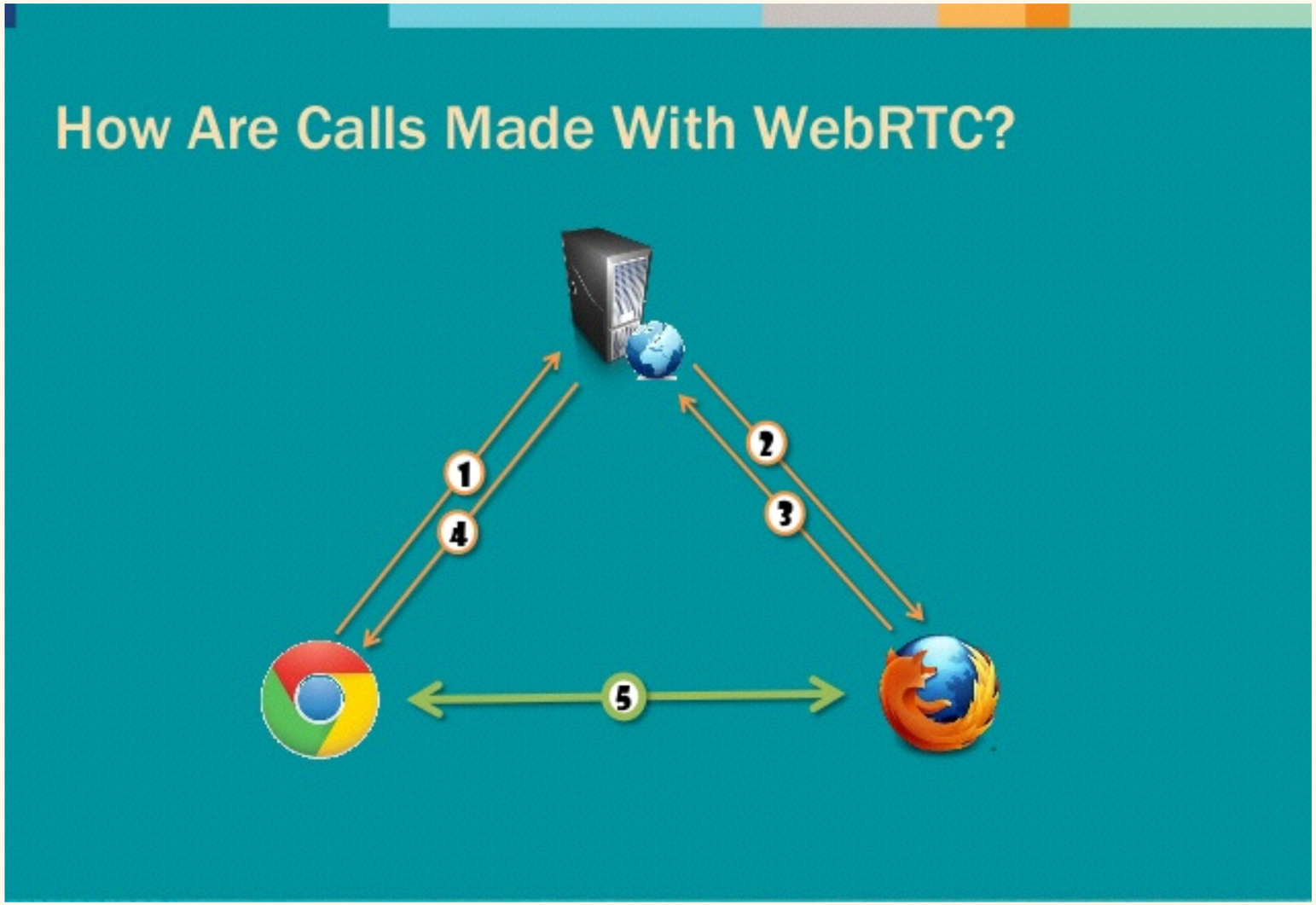
What is WebRTC?

- Real Time Communication
  - Voice Calling
  - Video Chat
  - P2P file sharing

- It’s not a service, it’s a technology

Is webrtc ready yet?

WebRTC - What’s the thing?



WebRTC consists of

1. An implementation of SRTP with an SDP control mechanism on top
2. A media engine with G.711, Opus and VP8 Codecs
3. A VoIP implementation using STUN, TURN and ICE for NAT traversal

WebRTC - Examples

- http://webcamtoy.com/
- https://appear.in/
- https://live.pics.io/
- https://www.sharefest.me/
- http://seriouslyjs.org/
- Our Demo

WebRTC - Major Components

1. `getUserMedia` – grabs your computers camera and/or microphone after being granted permission, and captures the streams.
2. `PeerConnection` – sets up audio/video calls between two parts
3. `DataChannels` - allow browsers to share files or other data directly via peer-to-peer

WebRTC - getUsermedia

```
navigator.getUserMedia(
  constraints, success, error
)
```

See task1.html

WebRTC - Important components

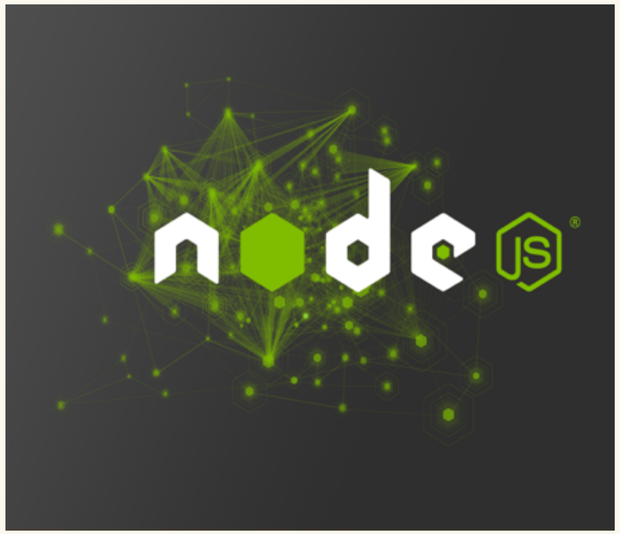
- Session control messages - to initialize or close the communication
- ICE Candidat - tells you who you are to the outside world (your computer's IP address and port)
- Session description - tells you what codecs and resolutions can be handled by your browser and the browser it wants to communicate with

Other Tools

Node.js

Platform to easily build fast network applications

Host a server easily on your own computer



Other Tools

Seriously.js

A real-time, node-based video effects compositor for the web built with HTML5 & Javascript

Preparation

1. Go to http://nodejs.org/
2. Download node.js for your system
3. Make sure you have a text editor or download for instance Sublime Text 3

First Task

Programmers:

1. Download the material from the course web
2. Run the server. In a terminal:
   a. cd PATH/TO/FOLDER
   b. node channel_server.js
3. Test at http://localhost:8080/task1.html
   If everything is working your browser should ask you for your camera and after accepting you should see yourself.
4. Enable the audio so you can hear yourself (beware of acoustic feedback)

Everyone else:

Warm-up exercise!

Second Task

Programmers:

1. Try out WebRTC video chat:
   http://localhost:8080/task2.html
2. See if you can connect two computers.
   Use ifconfig (Mac) or ipconfig (Windows) to find the IP address of your computer

The rest:

Brainstorm ideas for a WebRTC application that might be useful in your project

Third Task

Programmers:

1. Go to: http://localhost:8080/task3.html
2. Play with the filters!
3. Try and replace the image with webcam video
4. Apply CSS filters to your video chat

Non-programmers:

Pick the best idea you've come up with and begin sketching on a mockup version

Fourth Task

Programmers:

1. More advanced filters: http://seriouslyjs.org/
2. Take a look at http://localhost:8080/seriouslyexample.html
3. Try and use Seriously.js filters instead of CSS in your video chat
a. webrtc-workshop/client/Seriously.js/seriously.full.min.js
4. Can you use the Chroma Key filter to replace the background in your video chat?
    a. tip 1: use the 'chroma' effect on the video, then 'blend' with an image
    b. tip 2: http://localhost:8080/task4.html show how to apply effects to video

The rest:

Get the programmers to implement your mockup! :)

Links

W3C

WebRTC Working Group

http://www.w3.org/2011/04/webrtc/

Specifications

http://dev.w3.org/2011/webrtc/editor/getusermedia.html http://dev.w3.org/2011/webrtc/editor/webrtc.html

IETF

http://tools.ietf.org/wg/rtcweb/


PROTOTYP