

Getting Started With WebRTC and Respoke



**What are we going
to cover?**

WebRTC

Respoke

Code Samples

</QA>

Who Is This Guy?

Kyle Tyacke

Flash / Web Dev

Developer Evangelist Respoke

Beer Geek

ktyacke@respoke.io

@geekgonenomad

Who Are You?

What the Heck Is WebRTC?



WebRTC

“is a free, open project that enables web browsers with Real-Time Communications (RTC) capabilities via simple JavaScript APIs.”

WebRTC

*“is a **free**, open project that enables web browsers with Real-Time Communications (RTC) capabilities via simple JavaScript APIs.”*

WebRTC

*“is a free, **open** project that enables web browsers with Real-Time Communications (RTC) capabilities via simple JavaScript APIs.”*

WebRTC

*“is a free, open project that enables web browsers with **Real-Time Communications (RTC)** capabilities via simple JavaScript APIs.”*

WebRTC

“is a free, open project that enables web browsers with Real-Time Communications (RTC) capabilities via simple JavaScript APIs.”

Contributors



WebRTC Examples

Amazon Mayday



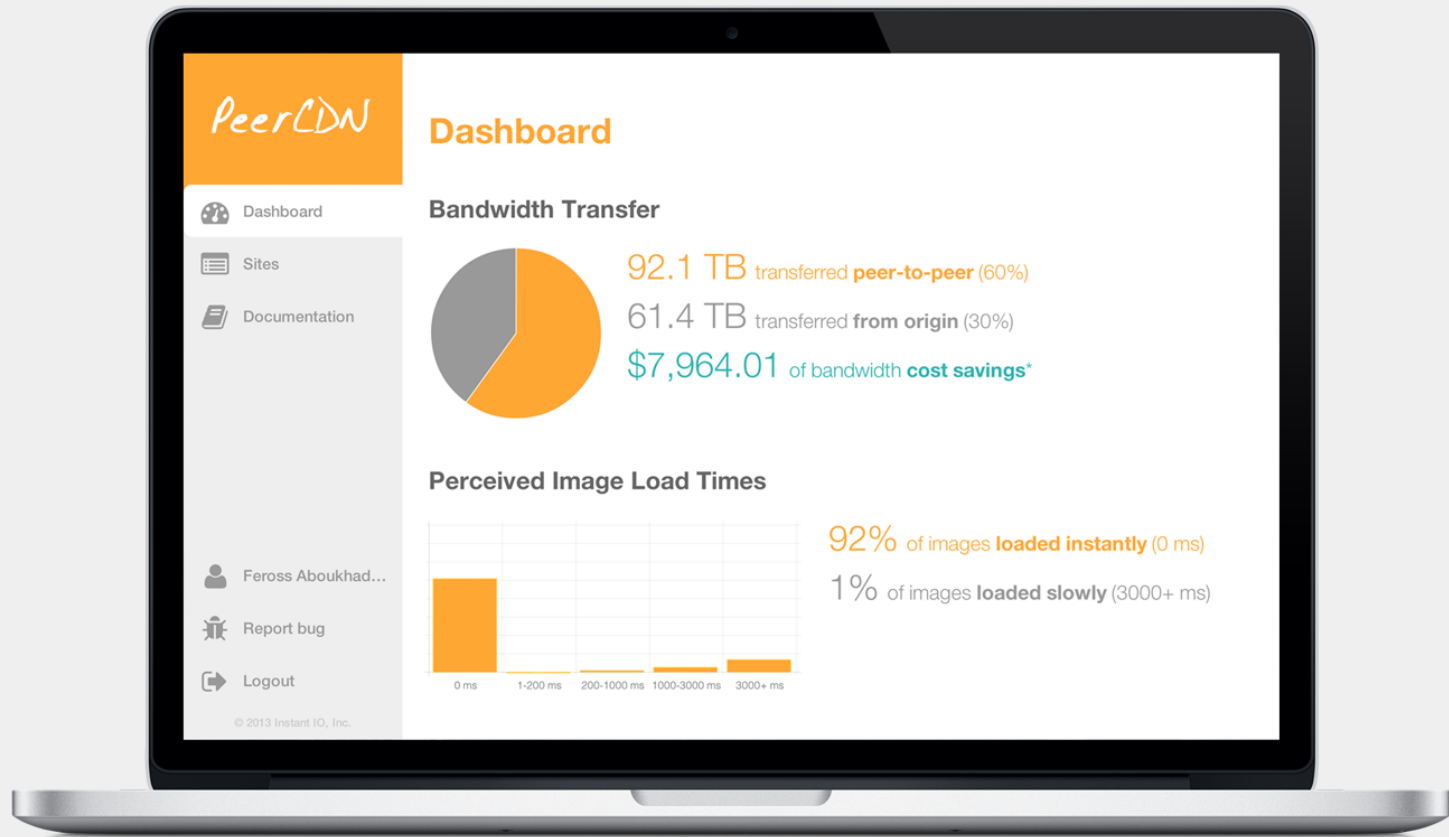
Google Hangouts

<https://plus.google.com/hangouts>



PeerCDN

<https://peercdn.com/>



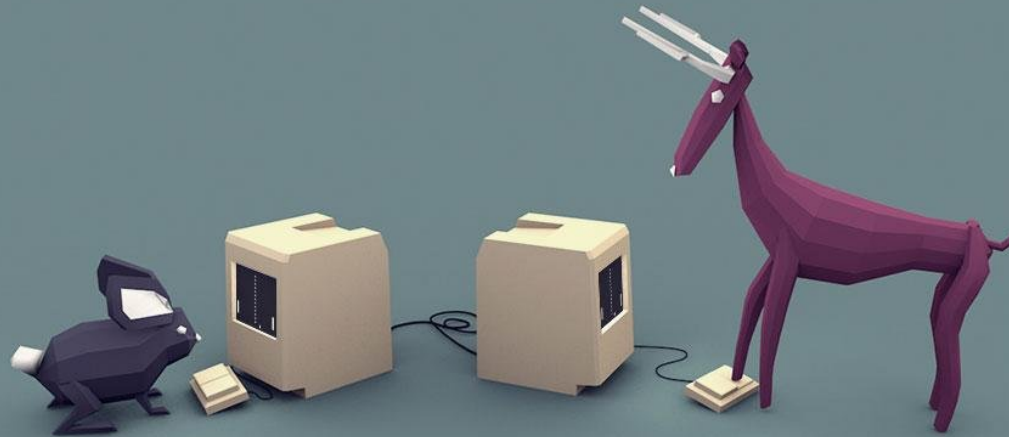
Sharefest

<https://sharefest.me>



CubeSlam

<https://www.cubeslam.com>



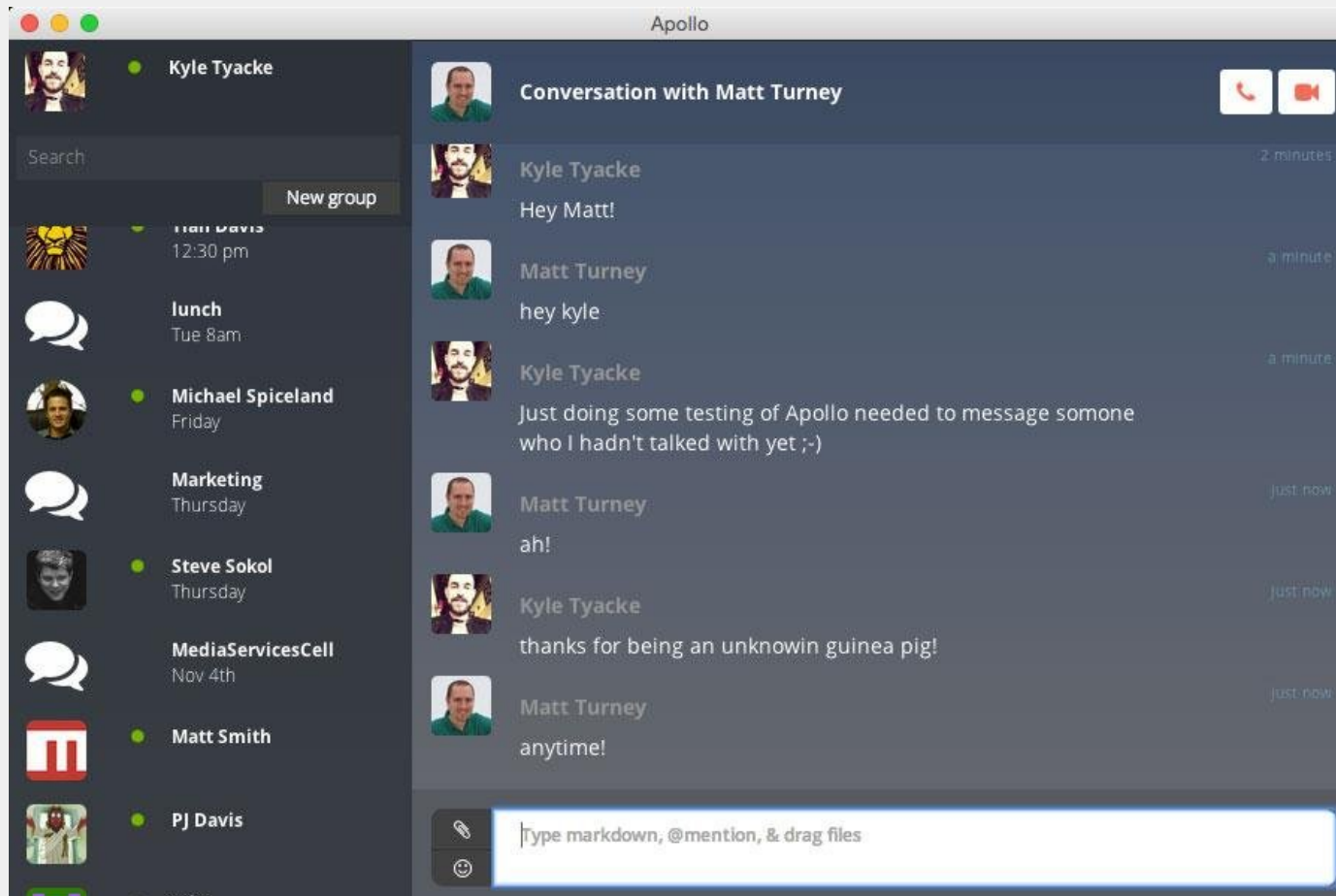
WebRTC



Cube Slam is built using WebRTC, an open web technology that lets you video chat with your friends right in the browser, no plug-ins necessary. The `getUserMedia` API provides access to your webcam and microphone, `RTCPeerConnection` sends the audio and video to your friends, and `RTCDataChannel` exchanges all the bits and pieces that keep the game in sync. When there are no firewalls in the way, game data can be sent directly peer-to-peer, greatly reducing server costs.

Apollo

<https://github.com/respoke/apollo>



How Does WebRTC Work?

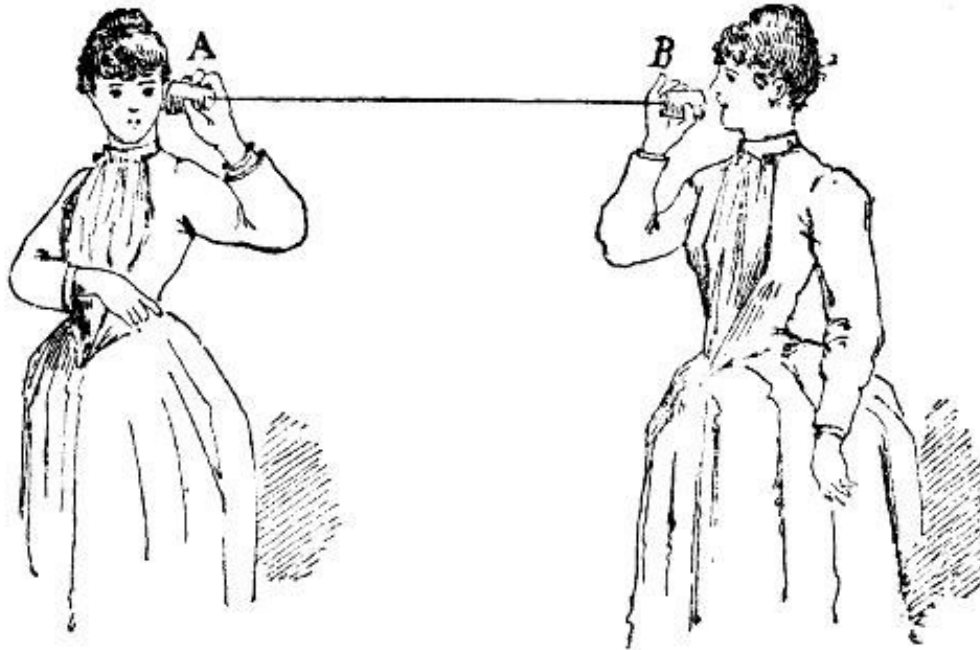
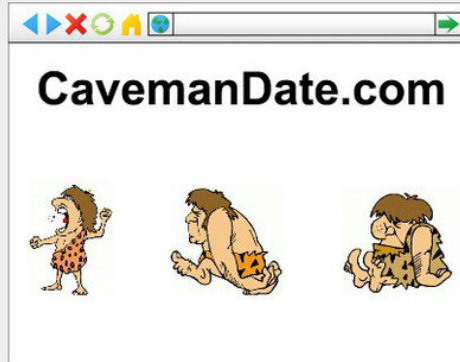


FIG. 76. Trådtelefon.

When Bob Met Alice

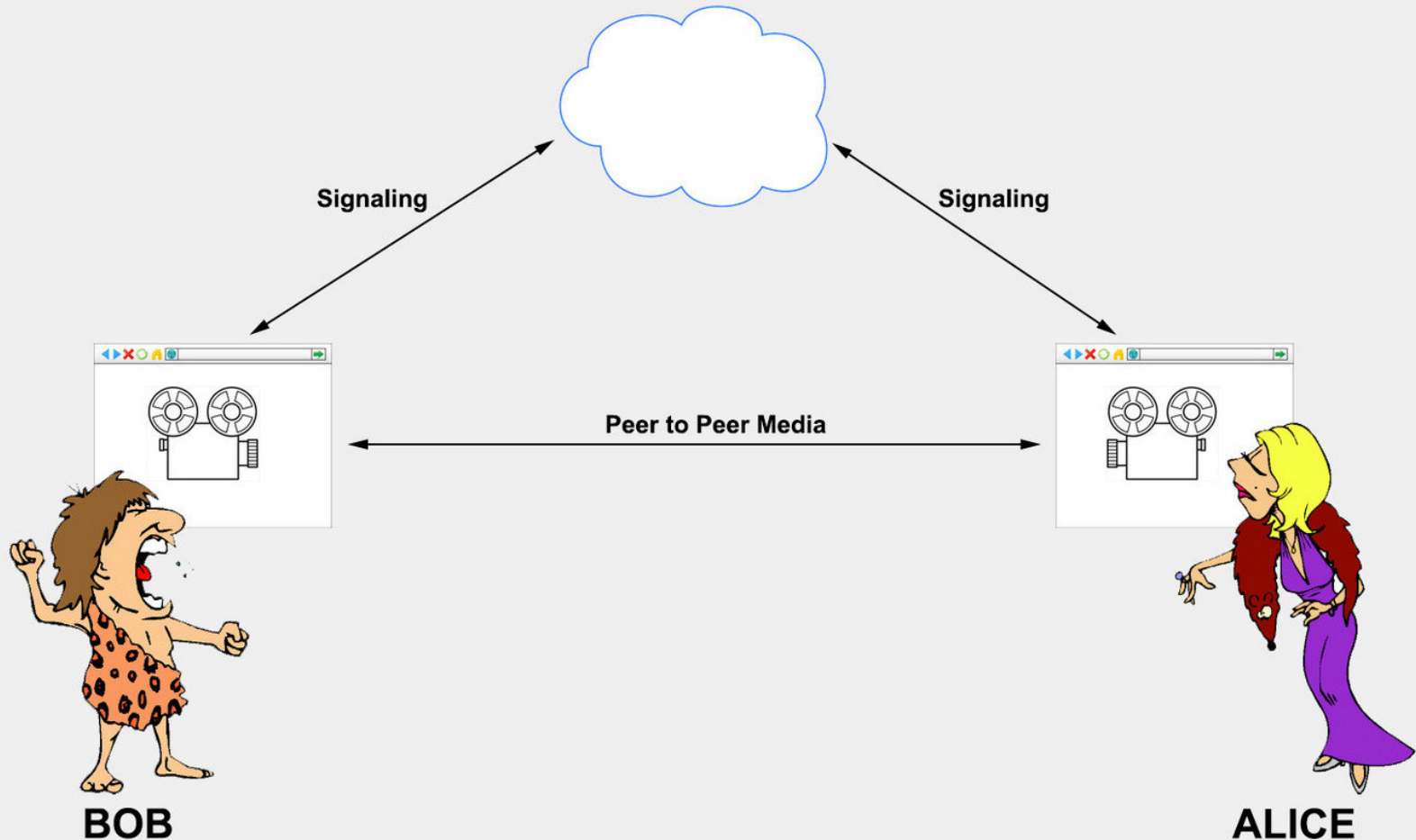


BOB

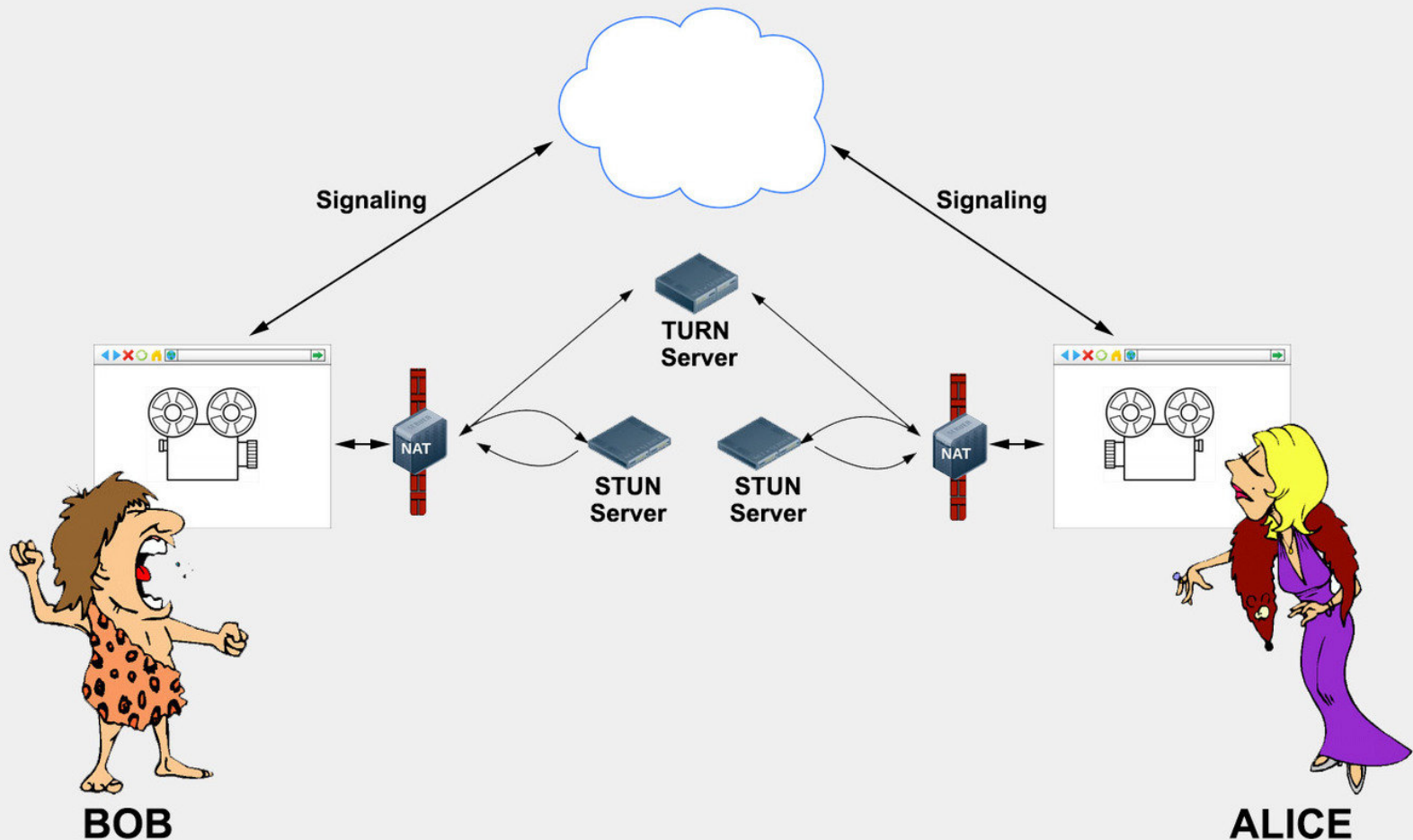


ALICE

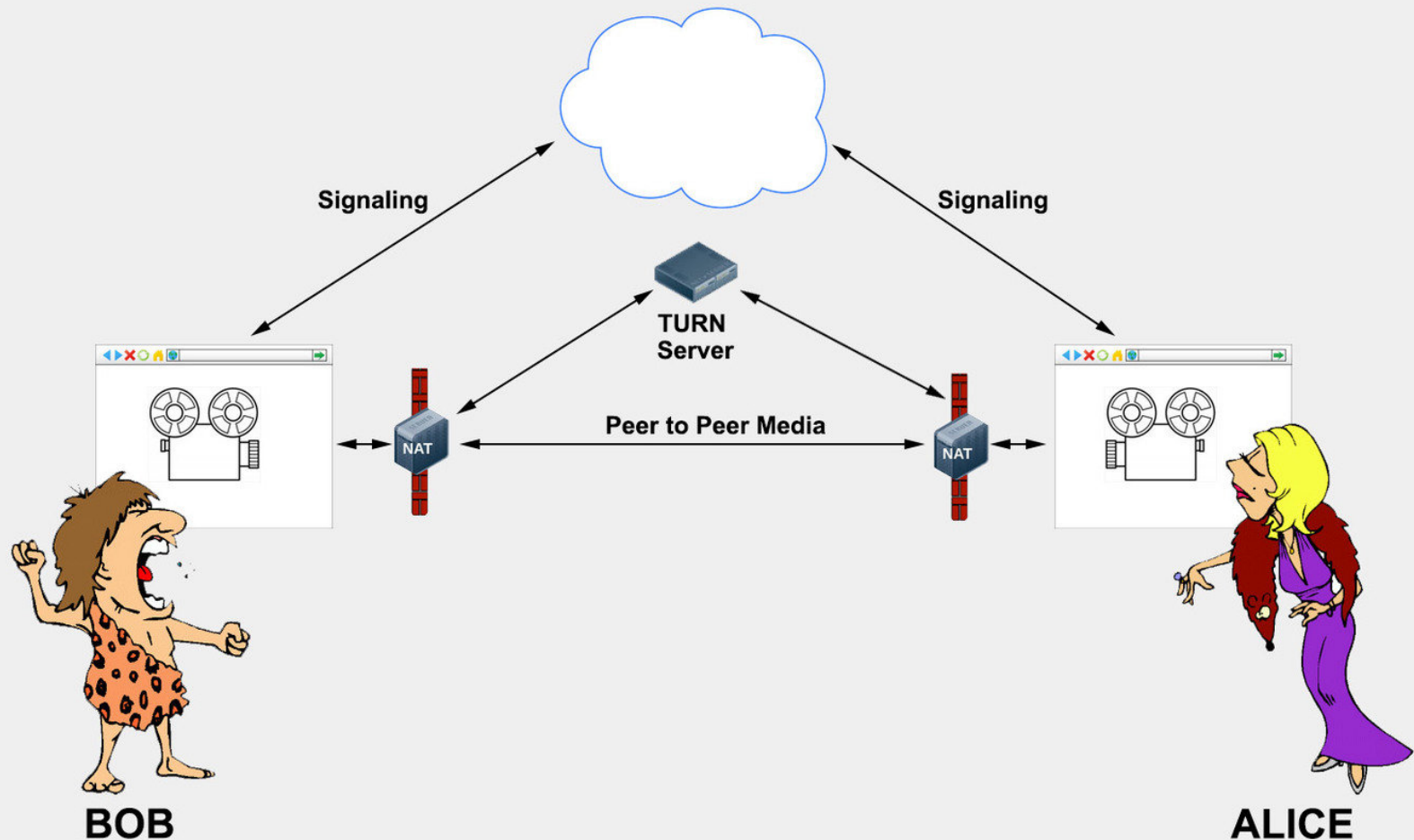
An Ideal World!



The Real World...



The Complete Connection



Simple Right?



respoke

Respoke?

A set of API's that gives developers the ability to easily add real-time communication to their applications.

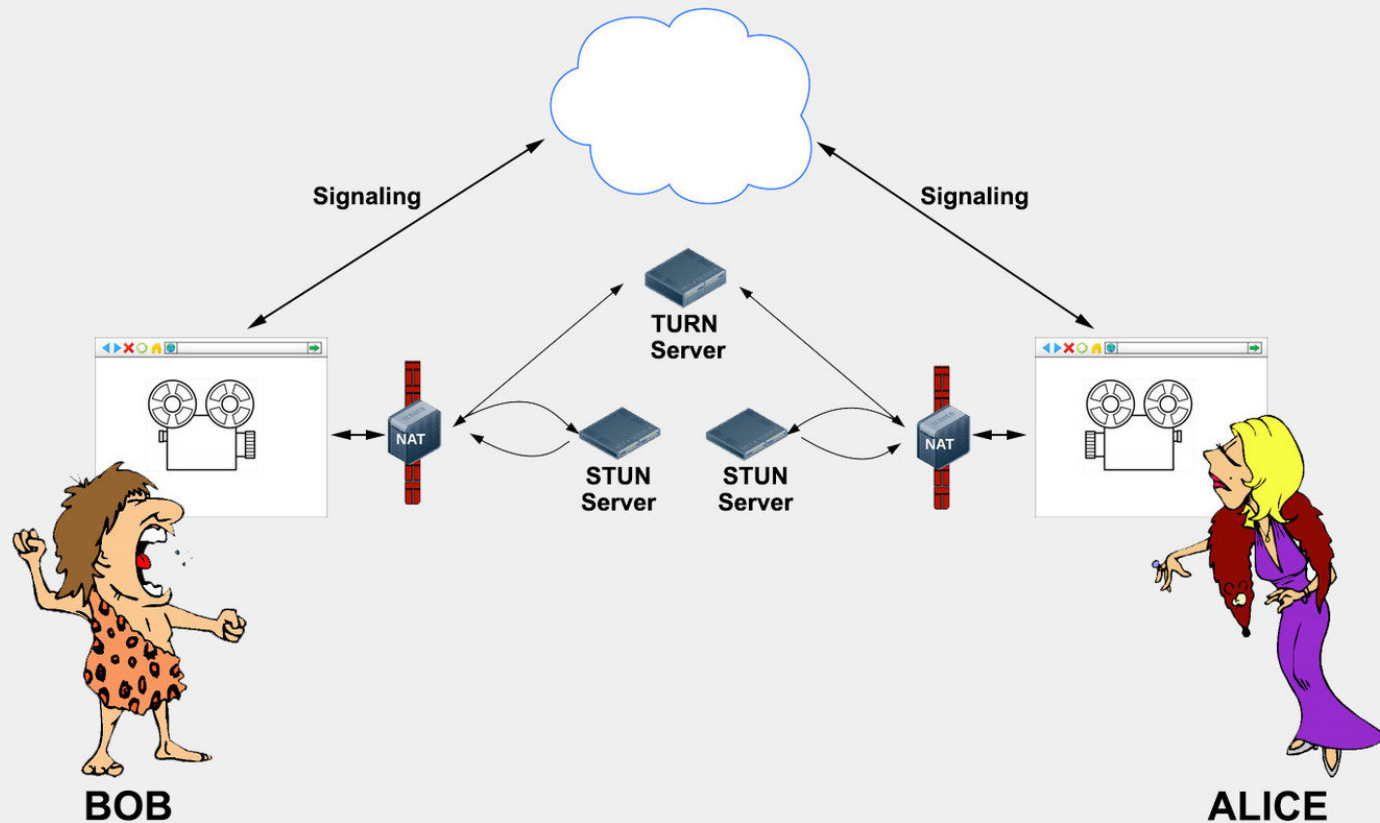
Like WebRTC?

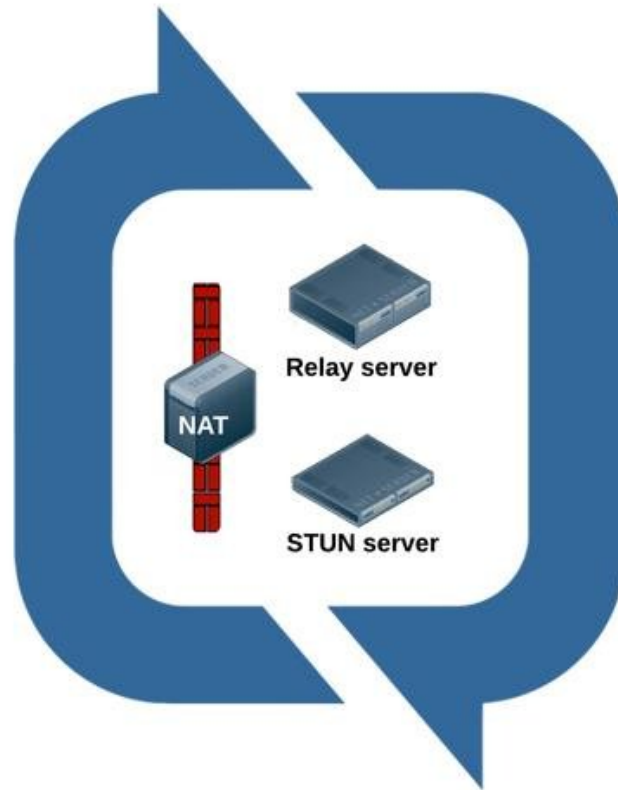


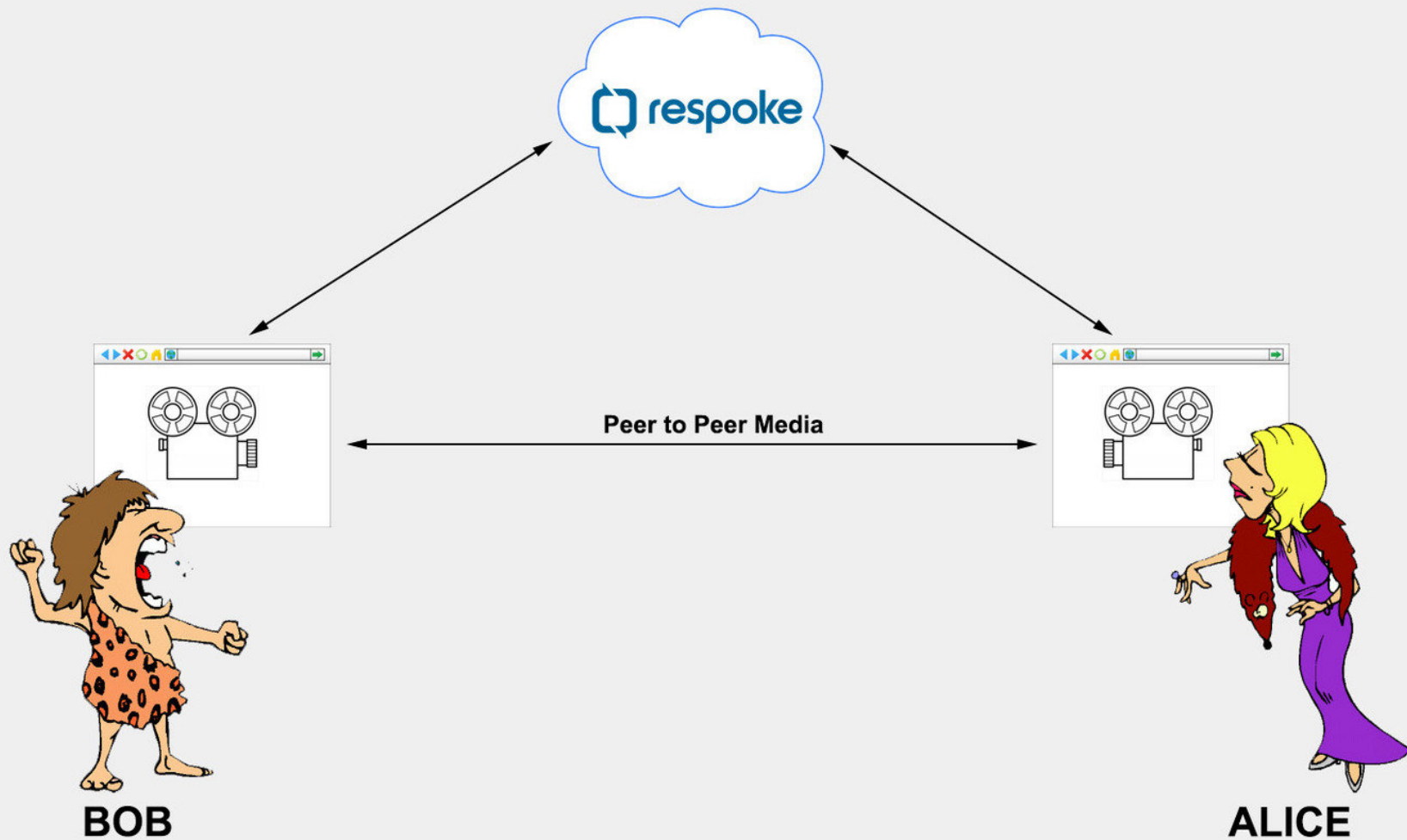
Sort of...



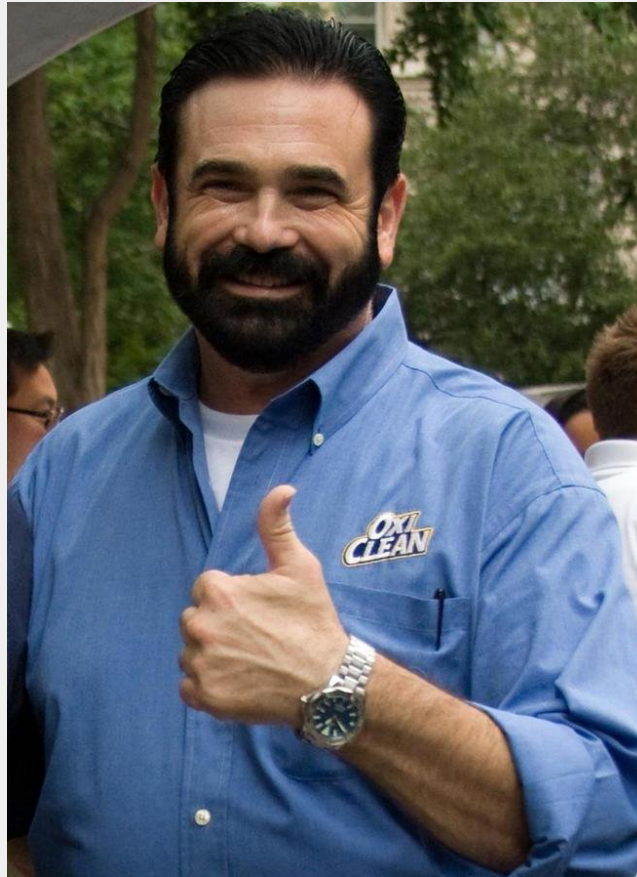
Remember This?







But Wait! There's More...



Open Source

Cross-browser Support

Endpoint Discovery

Endpoint Messaging

Groups

Presence

Phone System Support

<CODE/>

Create a Respoke Account

<http://www.respoke.io>

Connecting to Respoke

<http://bit.ly/1IgDTJ1>

```
// App ID value from the dev portal. You can play
// around with the supplied ID or replace it with
// your own.
var appid = "b4931d40-ff2b-4c46-8487-bf955a75501d";
var endpointId;

// Create the client object using the App ID
var client = respoke.createClient({
  appId: appid,
  developmentMode: true
});

// "connect" event fired after successful connection to Respoke
client.listen('connect', function() {
  $("#status").html("Connected to Respoke as \"" + endpointId + "\"");
});

// Connect to Respoke when the user clicks "connect"
$("#doLogin").click(function() {
  // Update the status message
  $("#status").html("Connecting...");

  // Grab our username
  endpointId = $("#endpoint").val();

  // Connect to Respoke
  client.connect({
    endpointId: endpointId
  });
});
```

Messaging

<http://bit.ly/1HzXGVo>


```
// Listen for incoming messages
client.listen('message', function(evt) {
    $("#messages").append(
        "<li>" + evt.message.message + "</li>"
    );
});

// Send message
$("#sendMessage").click(function() {
    // Get the recipients name
    var remote = $("#remoteId").val();

    // Make an endpoint for that recipient
    var endpoint = client.getEndpoint({
        id: remote
    });

    // Grab the text to send
    var messageText = $("#textToSend").val();

    // Send it
    endpoint.sendMessage({
        message: messageText
    });

    // Show yourself the message
    $("#messages").append(
        "<li>" + messageText + "</li>"
    );

    // Clear the text you just sent
    $("#textToSend").val(' ');
});
```

Video Calling

<http://bit.ly/1wcJf2j>

```
// The options for our video call including constraints and callbacks
var callOptions = {
  constraints: {
    audio: true,
    video: true
  },

  // Your video
  onLocalMedia: function(evt) {
    setVideo('localVideoSource', evt.element)
  },

  // Their video
  onConnect: function(evt) {
    setVideo('remoteVideoSource', evt.element)
  }
};

// Listen for incoming calls
client.listen('call', function(evt) {
  activeCall = evt.call;

  // We only want to answer if we didn't initiate the call
  if (activeCall.caller !== true) {
    activeCall.answer(callOptions);

    // The hangup event indicates the call is over
    activeCall.listen('hangup', function() {
      hangUp();
    });
  }
});
```

Docs and Additional Examples

<https://docs.respoke.io/>

What's It All Mean?

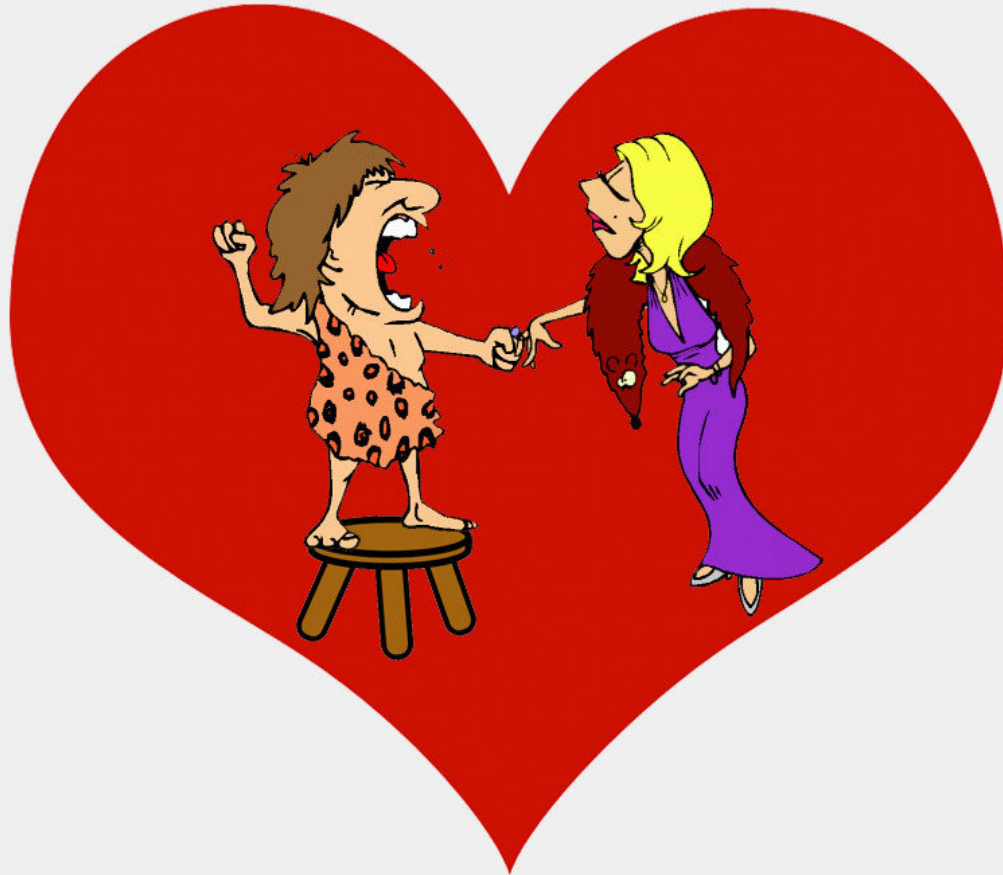
Faster Development

Better User Experience

Happy Bosses



Happy ~~Bosses~~ Users!



Communication
should be a feature,
not a product...

Kyle Tyacke

www.respoke.io

ktyacke@respoke.io

[@geekgonenomad](#)

Examples and Source

<https://github.com/ktyacke/respoke-webrtc-preson>