Autor: Marco Rodrigues nº4652

Mestrado em Engenharia de Software

Disciplina: Programação Web & Cloud

**Web RTC – File Transfer**

**O que é?**

Web Real-Time Communication é um conjunto de protocolos de comunicação e APIs que permitem aos browsers executar trocas de dados arbitrários, áudio ou vídeo sem recorrer a plugins, extensões ou recursos de servidores, fazendo-o dentro de uma arquitetura peer-to-peer.

A visão do WebRTC é possibilitar a vários dispositivos como telefones, tv, computadores, etc comunicarem numa plataforma comum e permitir que adição de chat de vídeo e partilha de dados seja feita de forma fácil e rápida numa aplicação web. A partilha de RTC deve ser tão natural quanto inserir texto num campo de input.

**Atualmente**

É suportado por vários browsers modernos (exceto o Safari), no entanto já é um conceito bem maturado para vários casos de uso. Inclusivamente está a ser standardizado pelo W3C e pelo IEFT.

**Necessidade**

Há muitos serviços que já usam o conceito de RTC, mas é necessário efetuar download e instalar algum plugin ou alguma app e integrá-la. O processo de download e de instalação pode ser complexo, demorado, propenso a erros e do ponto de vista de quem desenvolve fazer debug ou testes e manutenção a um plugin ou integrá-lo com outras tecnologias pode-se tornar complexo e difícil. Assim está identificada a necessidade tanto para utilizadores da web como para developers de usufruir de um serviço de RTC sobre a web que seja de fácil e natural utilização assim como interfaces de programação que tornem o seu acesso e o desenvolvimento e integração também fáceis.

**Vantagens**

Assim, o serviço de WebRTC apresenta as seguintes vantagens:

* Open-source e de livre utilização
* Grátis
* Standardizado e com padrões bem identificados
* Embutido nos browsers (sem necessidade de instalações extra ou de integração de plugins)
* Mais eficiente
* Menos exposto a complexidades resultantes de instalações/integrações
* Menos propenso a erros

**Três tarefas principais:**

* Adquirir vídeo e áudio
* Comunicar vídeo e áudio
* Comunicar dados arbitrários

Para isso o WebRTC implementa três APIs principais:

* MediaStream (ou getUserMedia)
* RTCPeerConnection
* RTCDataChannel

**MediaStream (ou getUserMedia)**

Representa uma stream (corrente) de conteúdo mídia (áudio ou vídeo), no entanto o foco deste trabalho incide nas outras duas APIs.

**RTCPeerConnection**

Esta API representa uma conexão entre o computador local e o peer remoto e fornece métodos e propriedades para gerir e consultar o estado de uma conexão (conectar, monitorizar, fechar conexão, etc), permite obter informação acerca do peer remoto e também permite obter informação sobre o estado do sinal da conexão.

**RTCDataChannel**

Esta API representa o canal de rede que pode ser utilizado para transferência de dados na conexão peer-to-peer de uma forma bidirecional. Cada canal está associado a uma RTCPeerConnection.

Para criar um canal e pedir que um peer se junte ao mesmo deve ser utilizado o método *createDataChannel()* da API RTCPeerConnection. O peer ao ser convidado recebe um evento do tipo RTCDataChannel.

Casos de uso para as APIs RTCPeerConnection e RTCDataChannel são jogos, aplicações desktop remotas, chats de texto em tempo real, transferência de ficheiros, etc.

**File Transfer**

A transferência de ficheiros via WebRTC é feita seguindo os seguintes passos:

* São utilizados canais de informação (Data Channels)
* O ficheiro é dividido em pedaços que são enviados pelo canal
* O envio e o progresso podem ser monitorizados através de elementos de progresso HTML5
* No destino o ficheiro é montado através da Blob API e fica assim disponível para download

**Adapter.js**

O adapter.js é um adaptador javascript que permite usar o mesmo código WebRTC em vários browsers permitindo a abstração das diferenças entre eles, nomeadamente os prefixos que nesta fase experimental da tecnologia os mesmos ainda possuem.

**Frameworks WebRTC**

* PeerJS (<http://peerjs.com/>)
* EasyRTC (<https://easyrtc.com>)
* SimpleWebRTC (<https://simplewebrtc.com/>)

Na demonstração prática utilizei a framework PeerJS.

**PeerJS**

A livraria PeerJS cobre a implementação WebRTC do browser e fornece uma API de conexão peer-to-peer completa, configurável e fácil de usar.

É necessária uma chave para a API e um ID para o peer, de seguida pode utilizar a API para criar um stream de dados ou media para um peer remoto.

Exemplos de utilização da framework: http://peerjs.com/examples