

# Web RTC

Real Time Communication

# Web RTC

Une API à l'état de brouillon au W3C

(Première version Octobre 2011)

# Web RTC

**Officiellement** développée pour:

Les appels de voix

Les appels vidéos

L'échange de fichier en P2P

# Web RTC

Fonctionne sans plug-ins

Directement dans le navigateur web

# Web RTC

Support encore limité (Aug 2016)

Chrome et Chrome Mobile

Android browser

Firefox

Opera

Edge (partiel)

# Web RTC

**Pas de support pour Safari  
et Safari IOS**

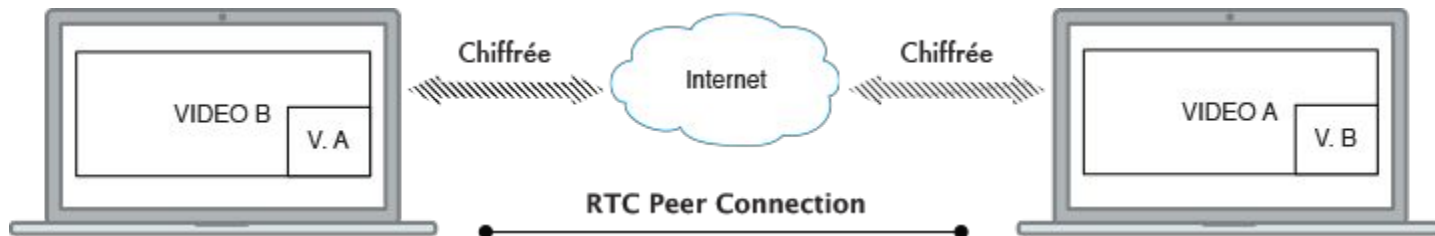
Devrait être supporté dans IOS10  
Peut être intégré dans une App

# Web RTC

¿ Comment ça marche ?

# Web RTC

## La base





# Web RTC

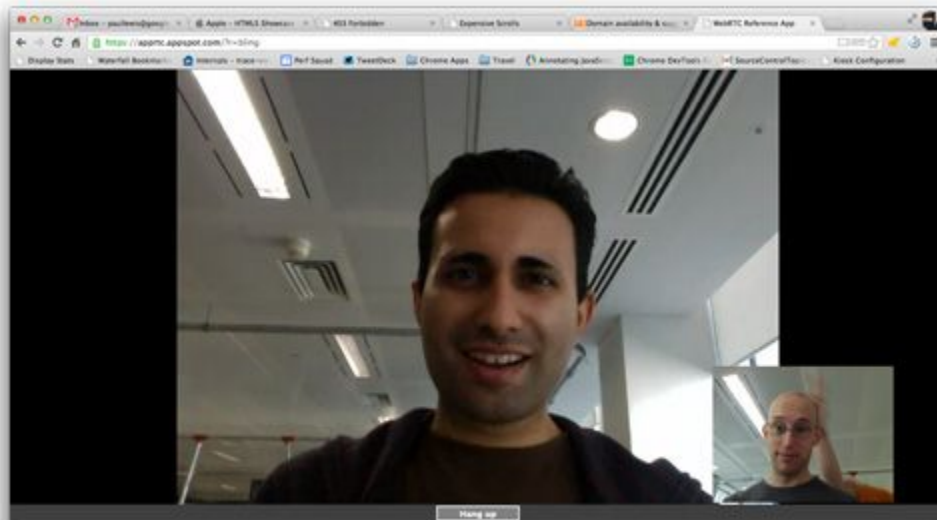


Image [HTML5rocks.com](https://html5rocks.com)

# Web RTC



# Web RTC

**Attention, ce n'est pas que de la vidéo**

Vidéo, Voix et Fichier

Fichier = Ce qu'on veut

# Web RTC

## **Les applications communes:**

Chat, vidéo conférence, screen casting  
Bref, de la messagerie

# Web RTC

**C'est le début, tout reste à inventer**

# Web RTC

**Ce qu'intègre le WebRTC**

# Web RTC

**La prise en charge des channels médias:**

Caméras, micros, écran, canevas

# Web RTC

**Le streaming des channels medias:**

H.264 / VP8



# Web RTC

## Les connections P2P

Communication directe sans serveur tiers.  
Point à point et multipoint

# Web RTC

## Des datas channels

Des tunnels de communication de données

# Web RTC

## La sécurité

La communication est basée sur un tunnel  
chiffré équivalent à un VPN

# Web RTC

## Déployer la technologie

Une API technique.

Demande une bonne connaissance du  
Javascript et des notions de réseau

# Web RTC

## Le workflow

- 1 Obtenir la source**  
Gérer le navigateur, demande d'accès (caméra, micro, ...)
- 2 Créer le stream**  
Créer un objet stream avec ses propriétés: résolution, codec, ...
- 3 Définir le tunnel**  
Créer le signaling pour que les autres peers puissent se connecter

# Web RTC

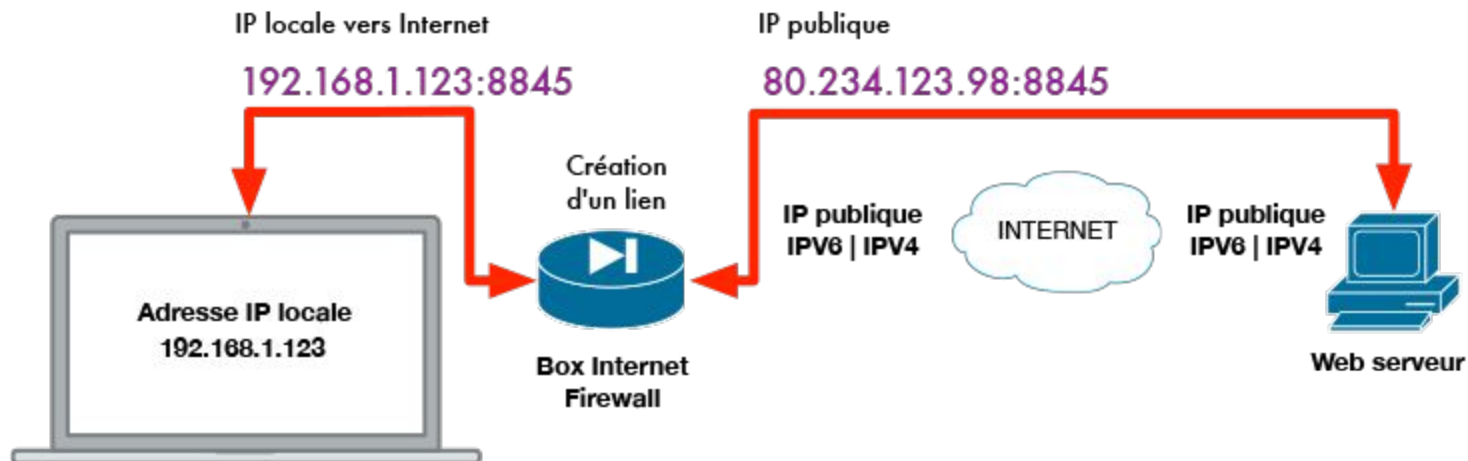
## Le signaling

### ICE / TURN / STUN

Pour mettre les clients (peers) en communication

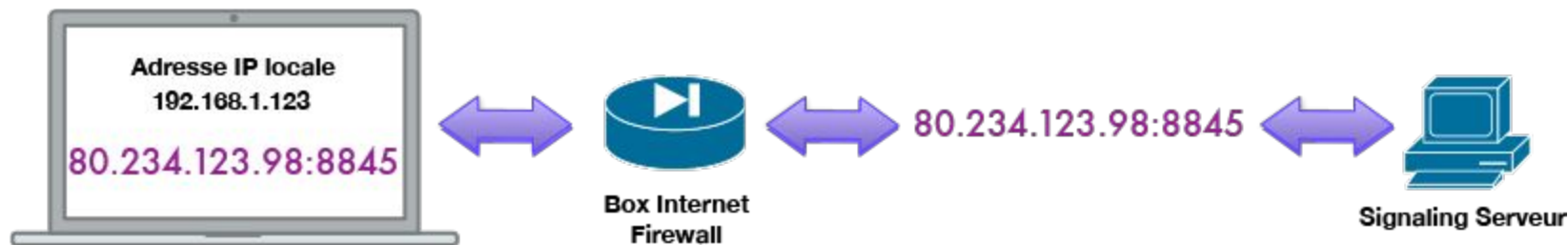
# Web RTC

## UPNP / TCP / UDP



# Web RTC

## ICE / TURN / STUN

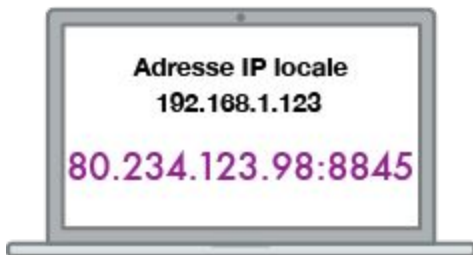




# Web RTC

## Peer Connection

```
{  
  'iceServers': [  
    {  
      'url': 'stun:stun.l.google.com:19302'  
    },  
    {  
      'url': 'turn:80.234.123.98:8845?transport=udp',  
      'credential': 'JZE0Et2V3Qb0y27GRntt2u2PAYA=',  
      'username': '28224511:1379330808'  
    },  
    {  
      'url': 'turn:212.145.23.19:9976?transport=tcp',  
      'credential': 'JZE0Et2V3Qb0y27GRntt2u2PAYA=',  
      'username': '28224511:1379330808'  
    }  
  ]  
}
```



# Web RTC

**Quelques exemples**

# Web RTC

## Canvas to Stream

<https://webrtc.github.io/samples/src/content/capture/canvas-video/>

# Web RTC

## Multiple Peer Connections

<https://webrtc.github.io/samples/src/content/peerconnection/multiple/>

# Web RTC

## Transfer Text

<https://webrtc.github.io/samples/src/content/datachannel/basic/>

# Web RTC

## Transfer Data

<https://webrtc.github.io/samples/src/content/datachannel/datatransfer/>

# Web RTC

## Video Chat

<https://apprtc.appspot.com/>

# Web RTC

## Quelques liens

<http://www.html5rocks.com/en/tutorials/webrtc/basics/>

<https://github.com/diafygi/webrtc-ips>

<https://webrtc.org/>

<https://codelabs.developers.google.com/codelabs/webrtc-web/>

<https://webrtc.github.io/samples/>