



APIs, Contextes, Réalisations autour du Web audio

Samuel Goldszmidt - Equipe APM (Analyse des Pratiques Musicales) / Ircam

27 Juin 2012 - GRAME

Conférence/Rencontre "Le web des fonctions"

Qui suis-je ?

Ingénieur informatique, web/multimédia à l'Ircam :

- APM (Analyse des Pratiques Musicales) : Ingénierie Multimédia (ex . [projet ANR MuTeC](#))
- CRI (Centre de Ressources Ircam) : base des données numériques ([brahms](#), sidney, repertoire ...)

Twitter : [@ouhouhsami](#)

Github : <https://github.com/ouhouhsami>

Technologies : Python/Django, JavaScript ...

Plan

1. WebRTC
2. Streaming
3. Formats
4. Stream multiformats
5. Timeline-js
6. Le projet WAVE
7. Web audio API

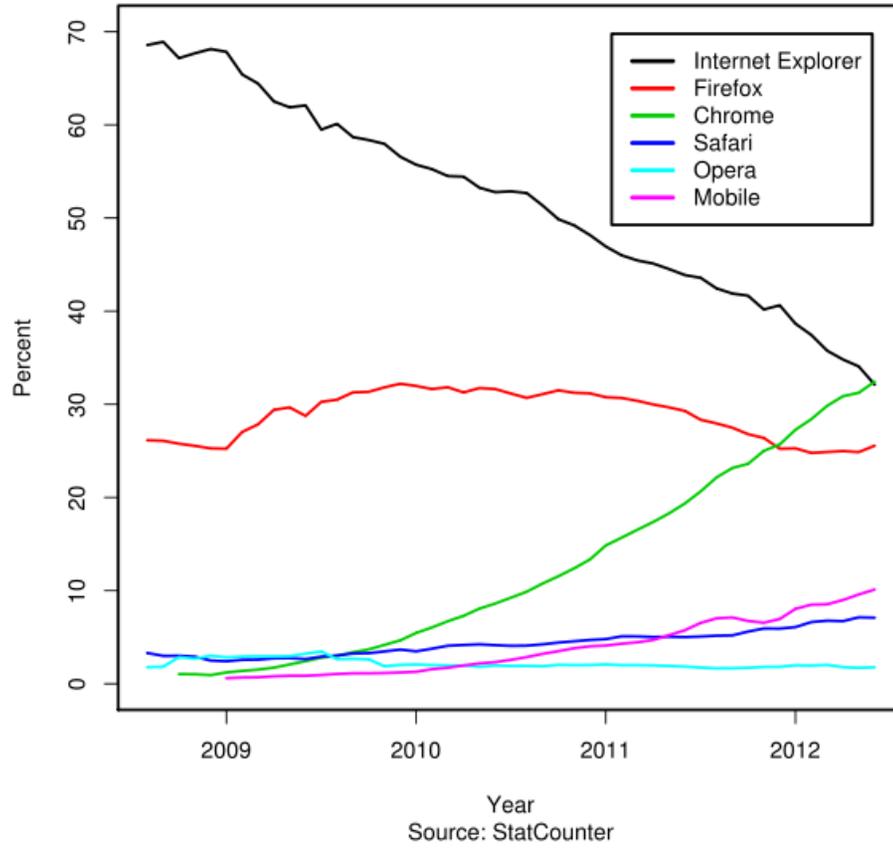


**Certains liens ne fonctionnent que sous Chrome Canary.
Certains thèmes ont certainement été abordés dans les présentations précédentes. N'hésitez pas à intervenir pour corriger/confirmer/infirmier/préciser ... certains points abordés dans cette présentation.**

Utilisation des navigateurs

Rappel avant de commencer

Usage share of web browsers



WebRTC

WebRTC

- Communications temps réels dans (et surtout entre) navigateurs
- Incluant notamment l'audio et la vidéo
- Composants utilisables via une API JavaScript
- WebRTC est (sera) intégré aux navigateurs (d'abord : Chrome, Firefox, Opera, même IE, mais Safari pas encore visiblement même si WebKit)

Demo

- [Installation de Chrome Canary](#)
- Activation du support PeerConnection dans les paramètres du navigateur (chrome://flags)
- apprtc.appspot.com

Avancement

- Manque le multi-pair (pour le moment communication 1:1)
- Fonctionne pour les données audio/vidéo, pour les autres type de données, cf. datachannel

Question WebRTC

- Est-ce que l'on stream à travers les balises media elements HTML5 (audio/video) ?
- Oui (et heureusement !)

Extrait du code HTML de l'application précédente

```
1 <video width="100%" height="100%" id="localVideo" autoplay="autoplay"/>
```

Streaming

Stream audio / video

Installation du serveur de stream Xiph Icecast en local sur Mac OS X

Autres serveurs de stream : Gstreamer, Real Helix Streaming Server, Windows Media Services, Adobe Flash Media Server, Quicktime Streaming Server

```
1 brew install icecast //https://github.com/mxcl/homebrew/
```

Fichier de configuration

```
1 vim /usr/local/Cellar/icecast/2.3.2/etc/icecast.xml
```

Lancer le serveur de stream

```
1 cd /usr/local/Cellar/icecast/2.3.2/bin
2 icecast -c ../etc/icecast.xml
```

Interface admin <http://localhost:8000>

login:admin password:hackme

Par défaut les fichiers à streamer sont à déposer ici :

chemin local : /usr/local/Cellar/icecast/2.3.2/share/icecast/web/stream

exemple d'url : <http://localhost:8000/stream/chopin.ogg>

Stream à travers *media elements*

Balise audio

Ogg only

Balise video

Ogg only

Formats

Formats – Audio

HTML5 Compatibilité Audio

Dernières versions des navigateurs

Problèmes des formats non supportés, pour l'audio, OGG et MP3 suffisent pour être lisible sur les navigateurs supportant la balise audio, on peut ajouter AAC pour des rapports qualité/bande passante (non étudié ici).

Navigateur (dernière version)	Ogg	MP3	AAC
Internet Explorer	✗	✓	✓
Firefox	✓	✗	✗
Chrome	✓	✓	✓
Safari	✗	✓	✓
Opera	✓	✗	✗
Android	✓	✓	✓
iOS	✗	✓	✓

Formats – Vidéo

HTML5 Compatibilité Vidéo

Browser (last version)	Ogg	MP4	WebM
Internet Explorer	✗	✓	✗
Firefox	✓	✗	✓
Chrome	✓	✓	✓
Safari	✗	✓	✓
Opera	✓	✗	✓
Android	✓	✓	✓
iOS	✗	✓	✓

Informations non vérifiées pour Android et iOS

Projet [Boot2Gecko](#) de Mozilla, qui supportera les mêmes formats que Firefox.

Formats – Transcodage

ffmpeg

Encodage via ffmpeg : de l'audio .ogg en .aac et .mp3, de la vidéo en .ogg en .webm et .mp4

```
1 brew install ffmpeg
2 cd /usr/local/Cellar/icecast/2.3.2/share/icecast/web/stream
3 ffmpeg -i audio.ogg audio.mp3 audio.aac
4 ffmpeg -i video.ogg video.mp4 video.webm
```

Stream multiformats

Utilisation d'un player

Utilisation pour cette démo de <http://videojs.com/>

Dans le header

Ici en exemple via cdn "content delivery network", ce n'est pas le cas pour cette présentation

```
1 <link href="http://vjs.zencdn.net/c/video-js.css" rel="stylesheet">
2 <script src="http://vjs.zencdn.net/c/video.js"></script>
```

Dans le body

```
1 <video id="my_video_1" class="video-js vjs-default-skin" controls
2   preload="auto" width="640" height="264" poster="my_video_poster.png"
3   data-setup="{}">
4   <source src="http://localhost:8000/stream/sample.ogg" type='video/ogg'>
5   <source src="http://localhost:8000/stream/sample.webm" type='video/webm'>
6   <source src="http://localhost:8000/stream/sample.mp4" type='video/mp4'>
7 </video>
```

Stream dans HTML5 media element

... pour le moment pas de DRM, mais cela pourrait voir le jour (Google/Microsoft/Netflix).

Timeline-js

Timeline-js

timeline-js, très petite librairie JavaScript pour la représentation temporelle de flux audio/vidéo, basé sur HTML5 (présentée pendant les dev track du WWW2012) :

- Media Elements via `<audio />` and `<video />`
- Media Events via `timeupdate` events pendant la lecture du media
- Canvas : interface graphique de la timeline et évènements : `click`, `mouseover`, `mouseout`, `mousedown`, `mouseup`

Etat de l'art (partiel) – Player

Liens

- mediaelementjs.com
- jplayer.org
- videojs.com
- sublimevideo.net

Détails

- UI
- fallback flash

Etat de l'art – Media synchronization

(partiel)

- [popcorn.js](#)
- [mugeda.com](#)

Détails

- Multiflux, pas de flux maître
- timeline

Etat de l'art – Intra-navigation media

(partiel)

- [SoundCloud](#)
- [Freesound](#)

Détails

- représentation (forme d'onde)
- annotation du flux (cf. Semantic-Hifi European project IST-FP7 2005, flash intra sound navigation)

Etat de l'art – Standards

(partiel)

- [timesheets.js](#) (cf. présentation de Cécile Roisin)

timeline-js – Démonstration

- configuration de base
 - curseur, échelle de temps, piste vide
 - pas de boutons de contrôle (play, pause, stop, zoom, scroll, laissé "à discrétion" de l'utilisateur)
- annotations
 - périodes (time_in et time_out)
 - marqueurs (time)
 - [(1..n) périodes, (1..) marqueurs]
- CRUD (Create, Read, Update, Delete modes)
- Events (createperiod, deleteperiod ...)
- Divers
 - options en *live*
 - représentation alternative

timeline-js – Futur

- HTML5
- Suivi du curseur
- (SVG à la place de canvas)
- Visualisation depuis la Web audio API
- Media fragments

timeline-js site web – demo en ligne : <http://ouhouhsami.github.com/timeline-js/>

"Forkme" on github <https://github.com/ouhouhsami/timeline-js>

Le projet WAVE

Le projet WAVE

Programme ANR CONTINT, soumission 2011, accepté 06/2012 – 30 mois

WAVE : Web Audio Visualisation/Edition

Partenaires :

- Ircam
 - APM – Samuel Goldszmidt, Nicolas Donin
 - IMTR – Norbert Schnell, Frédéric Bevilacqua
- Universal Music Publishing France – Eric Denuit
- Eurecom – Raphaël Troncy
- Vi-live – Emmanuel Fréard

Objectif résumé

Formaliser et réaliser les composants logiciels pour la navigation, la visualisation et l'édition de l'audio temps-réel en ligne.

Exemples de livrables

- Guides d'écoutes hypermédia
 - Blocs gigones
 - Parcours interactifs dans les Variations d'Anton Webern (cf. diapo suivante)
 - 3^e mouvement de Voi(rex) (cf. diapo suivante)
- Démonstrateur artistique

Exemples annexes

- Songle (flash)
- Escucha / Rosario Etcheverry Doctorante Université Lille 3

Variations d'Anton Webern – "Niveau 1"

Variationen für Klavier



6

II

Sehr schnell $\text{♩} = 160$

23

7

III

Ruhig fließend $\text{♩} = 60$

13

Mouvement binaire : il est constitué d'une succession de cellules de deux croches, et s'articule en deux parties répétées.
Exacerbation du rapport main droite main gauche à travers la variation permanente des paramètres musicaux (intensité, densité, attaques, hauteurs, silences, espace...).

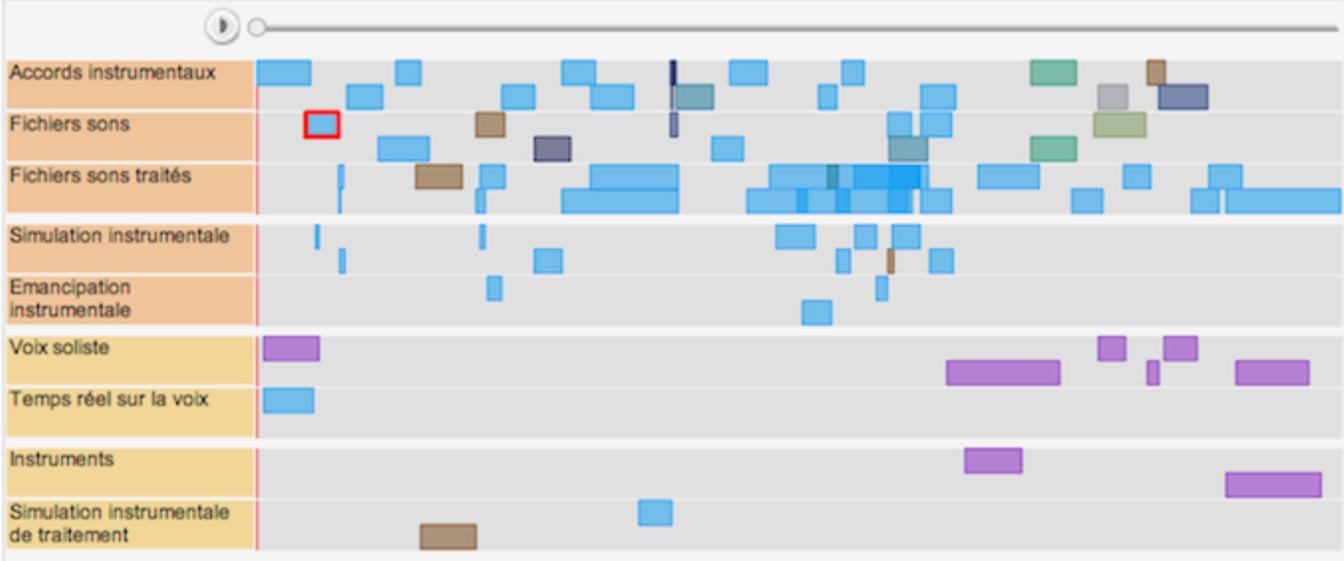
Variations d'Anton Webern – "Niveau 2"

Leroux Séquenceur (général)

Introduction à ce mouvement

Les manipulations du son dans le mouvement III *De part [...] En part*

Déplacer et altérer les échantillons sonores pour recomposer le mouvement Désactiver les bulles d'aides Réinitialiser



Légende	Informations contextuelles	Espace de manipulation et/ou visualisation
<ul style="list-style-type: none">Accord n° 1Accord n° 2Accord n° 5Accord n° 6Accord n° 8Accord n° 10Accord n° 15Accord n° 16AutresPrise de son (ressac)	<p>Nom du fichier: fichierson_mes174.wav</p> <p>Début: 14.566 secondes</p> <p>Fin: 24.377 secondes</p> <p>N° de mesure (initial): 174</p>	<p>3128 Hz 0.82 dB 5 Q</p> <p>Appliquer l'effet de filtrage</p> <p>1000 Hz Appliquer l'effet frequency shifter</p> <p>1 Vitesse Appliquer l'effet doppler</p> <p>0.5 Etirer le son temporellement</p> <p>12 Demi-ton Appliquer l'effet pitch shifter</p> <p>Appliquer l'effet freeze</p>

Leroux Séquenceur (détail partition)

The screenshot displays the interface of the Leroux Sequencer. At the top, there are four horizontal tracks with purple rectangular notes. A red square highlights a specific area on the third track from the top. Below these tracks is a grey bar with a green double-headed arrow icon on the left and the text "Espace de manipulation et/ou visualisation" in the center. Underneath this bar is a large yellow area divided into two parts. The left part shows a musical score with multiple staves, including handwritten notes and a ".wav" label on the left side. The right part is a solid yellow rectangle.

Leroux Séquenceur (traitement sonore)

The screenshot displays the interface of the Leroux Sequencer software, which is used for audio processing. The interface is divided into two main sections: "Informations contextuelles" (Contextual Information) and "Espace de manipulation et/ou visualisation" (Manipulation and/or visualization space).

Informations contextuelles (Contextual Information):

- Nom du fichier: acinstr201.wav
- Début: 73.293 secondes
- Fin: 83.168 secondes
- N° de mesure: 201 (initial)

Espace de manipulation et/ou visualisation (Manipulation and/or visualization space):

- Sélectionner un extrait du son (Select an audio excerpt)
- Lire le fichier son à l'envers (Play the audio file backwards)
- Appliquer une courbe de volume (Apply a volume curve)

The visualization area shows a purple rectangular selection box over a waveform. A volume curve is applied to the selected segment, with a peak labeled "0dB".

Outil auteur annotation plurimedia

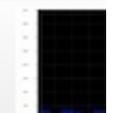
Projet ANR MuTeC 2009/2012

Outil auteur PACOME annotation plurimedia

SÉLECTION DES MÉDIAS À CONSULTER ET ANNOTER

Cliquer sur les médias à consulter et annoter

Ressources simples

 entretien-k7-1ffogg	 stefano12janvier-1ogg	 wer-60127-50_track_016ogg	 1jpg_	 br-2jpg	 periodesogg
 1jpg	 2jpg_	 3-01-06jpg	 3jpg_	 4jpg_	 5jpg_
 6jpg_	 7jpg_	 8-03-05-Objets-inclus-	 8-03-05-afairedansom3jpg	 test2png	 africajpg

Collections

 collection, 11 éléments	 canti-capri, 11 éléments	 partition, 15 éléments	 carnet, 2 éléments
--	---	---	--

Outil auteur annotation plurimedia

† canti-capri
† stefano12janvier-1ogg
† wer-60127-50_track_016ogg
† 2jpg_

canti-capri

- 100% + 60.000000000000001

Toolbar

C U D G L T R

Rect. Circ. Ell. Pth

fill 0.2

stroke Black 1.0 3

Molto lentamente (♩ = 50)

f *mf* *mf* *mf*

f *no dolce*

ritardando *molto* *mf*

Web Audio API et +

Pour rappel, toujours du point de vue d'un développeur web

Les API

- Avant : *Web Audio API* et *MediaStream Processing API*
- Maintenant : *Web Audio API* et *MediaStream Processing API*

Toujours la même dualité ? ... non, et tout se tient !

- Web Audio API, dernier brouillon le 15 mars 2012
- Media Stream Processing (dernier brouillon fin mai), qui dépend de WebRTC (WebRTC, dernier brouillon le 30 mai 2012)

Pour simplifier à l'extrême, ce que je retiens/résume :

WebRTC/Media Capture and Streams (les 2 faisant parti du Web Real-Time Communications Working Group) définit 'getUserMedia' permettant à 'Media Stream Processing' de manipuler les stream entrées/sorties audio/vidéo. La Web Audio API, elle, propose les traitements du flux audio." (Media Stream Processing, et Web Audio API font partie du Web Audio Working Group)

Breaking news : getUserMedia doit être dans la nightly de Firefox (permettant donc le "media capture")

L'API Web Audio dans la pratique

2 alternatives :

1. Un unique JavaScriptAudioNode
2. Un graphe utilisant les autres Node disponibles dans l'API

Des interrogations à lever sur :

- Performances comparées entre 1. et 2.
- Point de vue pédagogique

Web Audio API buzz – points d'intérêt

- [AngryBird utilise la web audio API sur Chrome](#)
- [Doodle Moog utilise la Web audio API sur Chrome](#)
- [Détection de camera pour jouer du son](#)
- [Mp3, FLAC JavaScript decoder !](#)
- [Javascript Library for Objective Sound Programming](#)

Dernières nouvelles

A propos de la Web Audio API, Ex. d'un message issu de la liste de discussion qui montre la rencontre de 2 univers : le web et le DSP.

Craig Handson, Lead Software Engineer à iZotope (xbox, harmonix, plug'in, jeux ...). D'après lui :

1. Le DSP doit pouvoir faire des traitements continuellement
2. L'analyse du CPU doit se faire sur les piques, non sur des valeurs moyennes
3. Qu'est ce que l'on entend par latence ? : l'entrée / la sortie (A/D – D/A), ou l'algorithme, à préciser dans l'API pour informer le développeur
4. Existence de paramètres de type Booléens et Entiers à prendre en compte, en plus des Flottants (ex. interface VST).
5. Nécessité de pouvoir prédire le CPU disponible
6. Avoir un processus dédié au DSP
7. Problème d'overhead du JavaScript
8. Problème du garbage collector de Javascript qui rend imprévisible le CPU
9. LLVM solution => bytecode (... se rapproche de solutions propriétaires pour les algorithmes)

Le MIDI

Un brouillon d'API récent

- Démo !
- Midi