INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG2015/M37256
October 2015, Geneva, Switzerland

Source **Telecom ParisTech**
Status **For consideration at the 113th MPEG Meeting**
Title **Demonstration of MP4Box.js**
Author Cyril Concolato, Jean Le Feuvre

1 Introduction

MP4Box.js is an open-source library developed within the GPAC project to enable manipulations of ISOBMFF files in JavaScript. It is inspired from the original MP4Box (C version) but is written from scratch. MP4Box.js can be used within a browser as well as within REPL environments like Node.js.

This contribution presents some applications done with MP4Box.js, to demonstrate some of its latest developments and some of the interesting features of the ISOBMFF.

The source code for MP4Box.js is available here:
https://github.com/gpac/mp4box.js

Contributions to identify bugs, to fix them, to add features are welcome.

2 Online File Analyzer

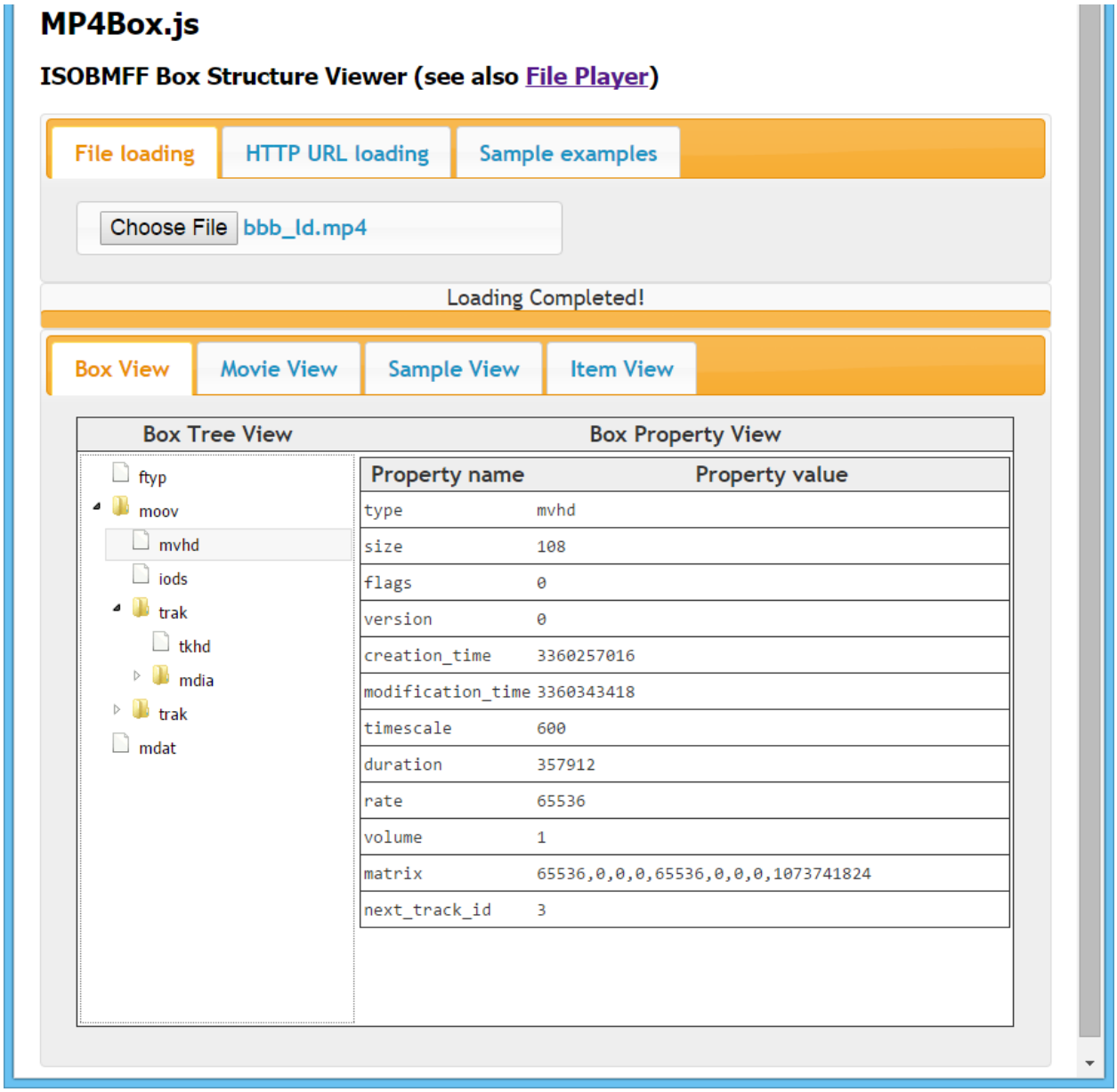
With MP4Box.js, it is possible to inspect the content of an MP4 file within a browser such as Firefox and Chrome. The files to inspect can be stored on the hard disk of the computer running the browser, or it may be located on the web, accessible via HTTP URLs. The elements that can be inspected are:

- the box structures and content
- the movie, tracks and samples
- the metadata items

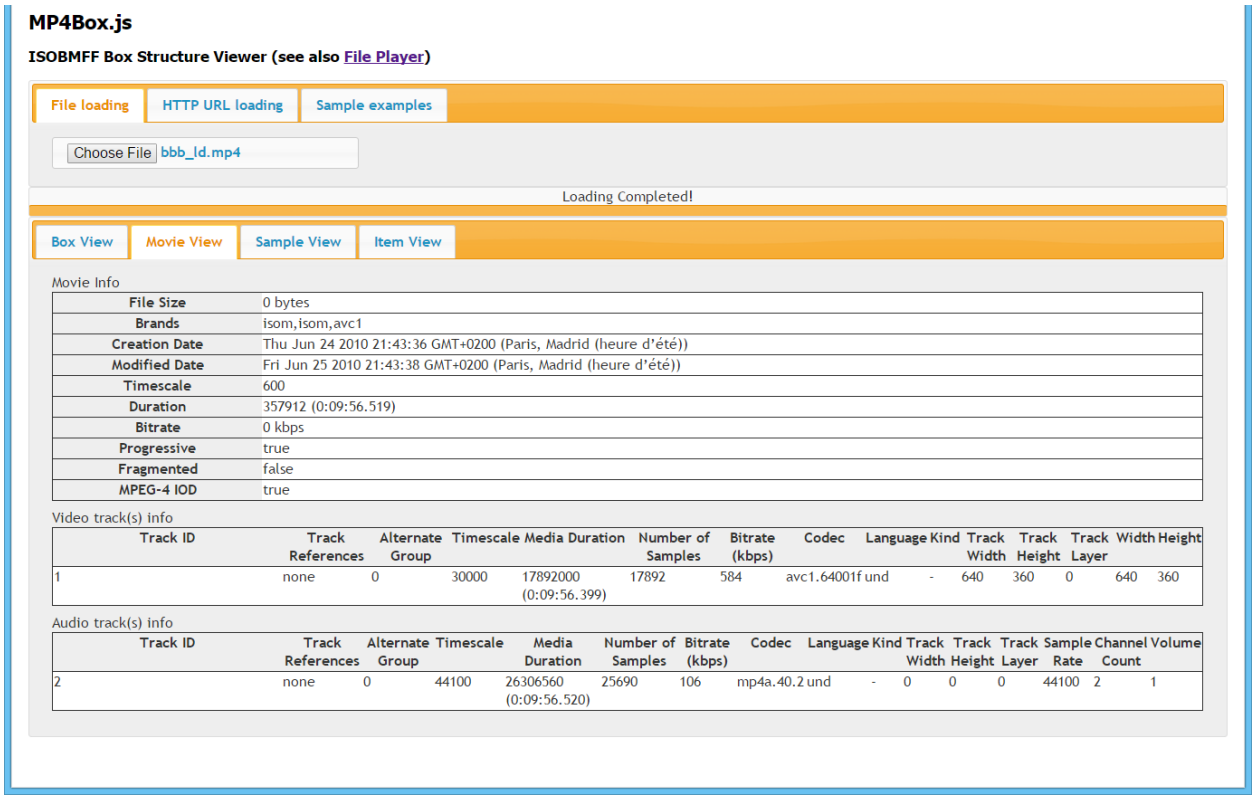
The analyzer is available at the following address:
http://download.tsi.telecom-paristech.fr/gpac/mp4box.js/filereader.html

It should be noted that this analyzer is capable of analyzing large files (> 4GB).

The following figure shows the view of the box hierarchy and the properties of a selected box (e.g. mvhd).



The following view shows the movie view. It shows general information about the file such as brands, fragmentation, as well as track information (including audio, video, subtitles).



MP4Box.js

ISOBMFF Box Structure Viewer (see also File Player)

File loading | HTTP URL loading | Sample examples

Choose File bbb_ld.mp4

Loading Completed!

Box View | Movie View | Sample View | Item View

Movie Info

File Size	0 bytes
Brands	isom,isom,avc1
Creation Date	Thu Jun 24 2010 21:43:36 GMT+0200 (Paris, Madrid (heure d'été))
Modified Date	Fri Jun 25 2010 21:43:38 GMT+0200 (Paris, Madrid (heure d'été))
Timescale	600
Duration	357912 (0:09:56.519)
Bitrate	0 kbps
Progressive	true
Fragmented	false
MPEG-4 IOD	true

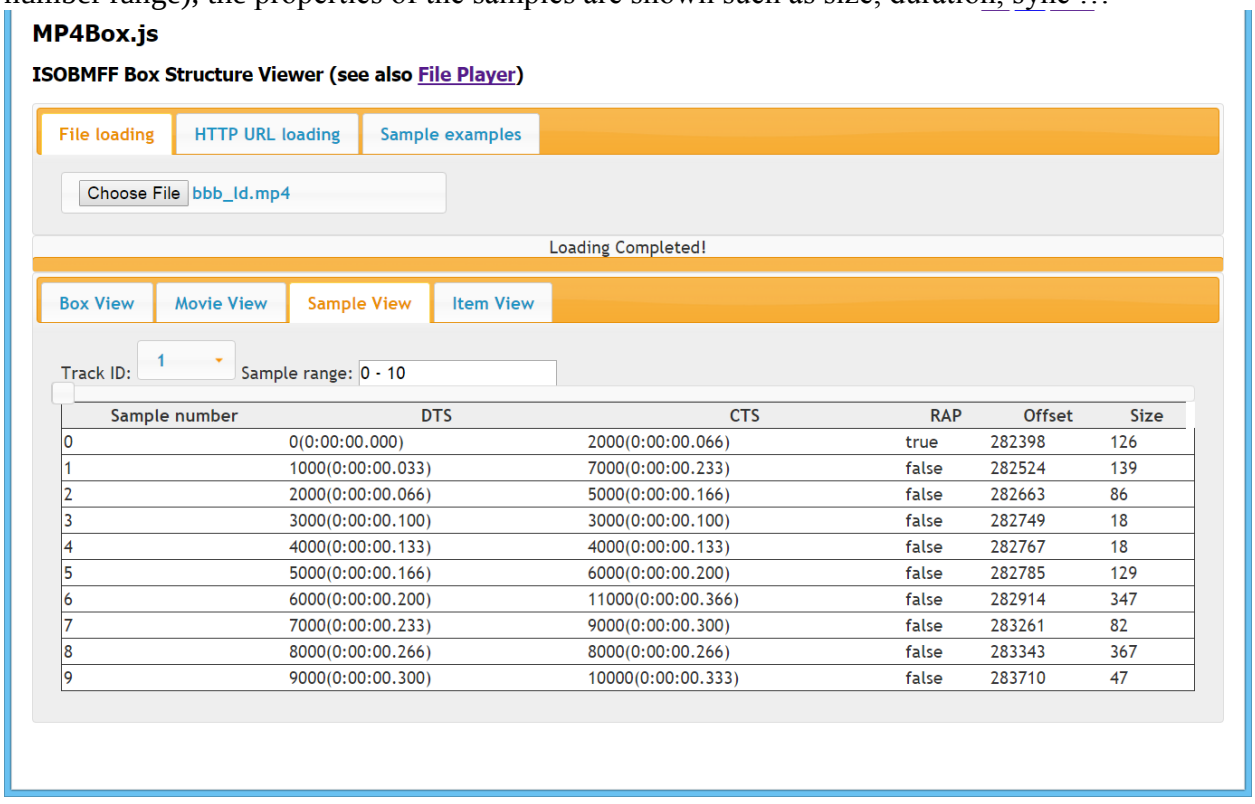
Video track(s) info

Track ID	Track References	Alternate Group	Timescale	Media Duration	Number of Samples	Bitrate (kbps)	Codec	Language	Kind	Track Width	Track Height	Track Layer	Width	Height
1	none	0	30000	17892000 (0:09:56.399)	17892	584	avc1.64001f	und	-	640	360	0	640	360

Audio track(s) info

Track ID	Track References	Alternate Group	Timescale	Media Duration	Number of Samples	Bitrate (kbps)	Codec	Language	Kind	Track Width	Track Height	Track Layer	Sample Rate	Channel Count	Volume
2	none	0	44100	26306560 (0:09:56.520)	25690	106	mp4a.40.2	und	-	0	0	0	44100	2	1

The following image shows the sample view. For a select set of samples (per track and sample number range), the properties of the samples are shown such as size, duration, sync …



MP4Box.js

ISOBMFF Box Structure Viewer (see also File Player)

File loading | HTTP URL loading | Sample examples

Choose File bbb_ld.mp4

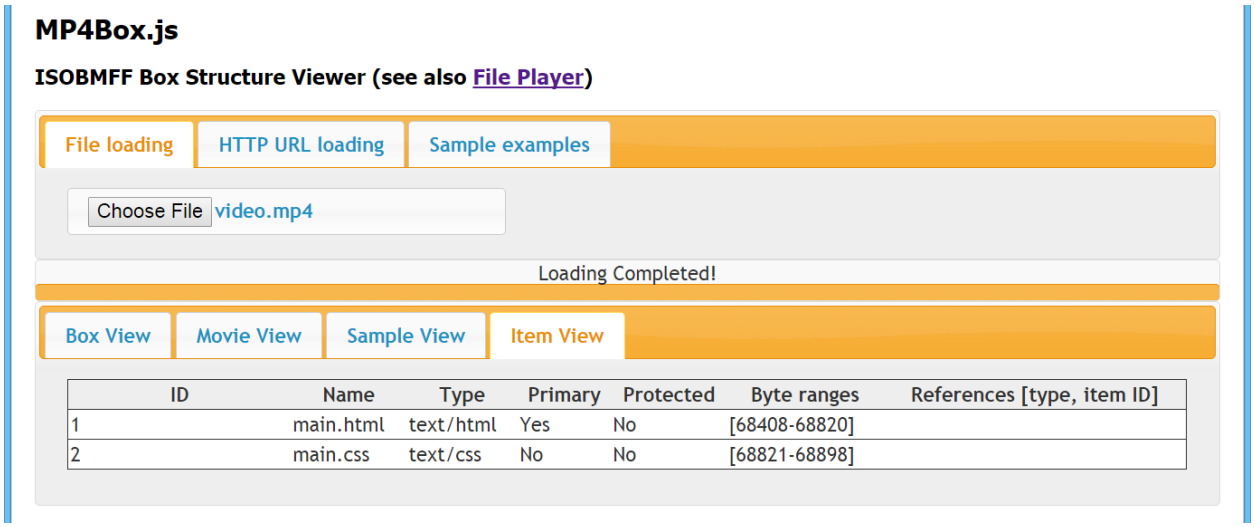
Loading Completed!

Box View | Movie View | Sample View | Item View

Track ID: 1 Sample range: 0 - 10

Sample number	DTS	CTS	RAP	Offset	Size
0	0(0:00:00.000)	2000(0:00:00.066)	true	282398	126
1	1000(0:00:00.033)	7000(0:00:00.233)	false	282524	139
2	2000(0:00:00.066)	5000(0:00:00.166)	false	282663	86
3	3000(0:00:00.100)	3000(0:00:00.100)	false	282749	18
4	4000(0:00:00.133)	4000(0:00:00.133)	false	282767	18
5	5000(0:00:00.166)	6000(0:00:00.200)	false	282785	129
6	6000(0:00:00.200)	11000(0:00:00.366)	false	282914	347
7	7000(0:00:00.233)	9000(0:00:00.300)	false	283261	82
8	8000(0:00:00.266)	8000(0:00:00.266)	false	283343	367
9	9000(0:00:00.300)	10000(0:00:00.333)	false	283710	47

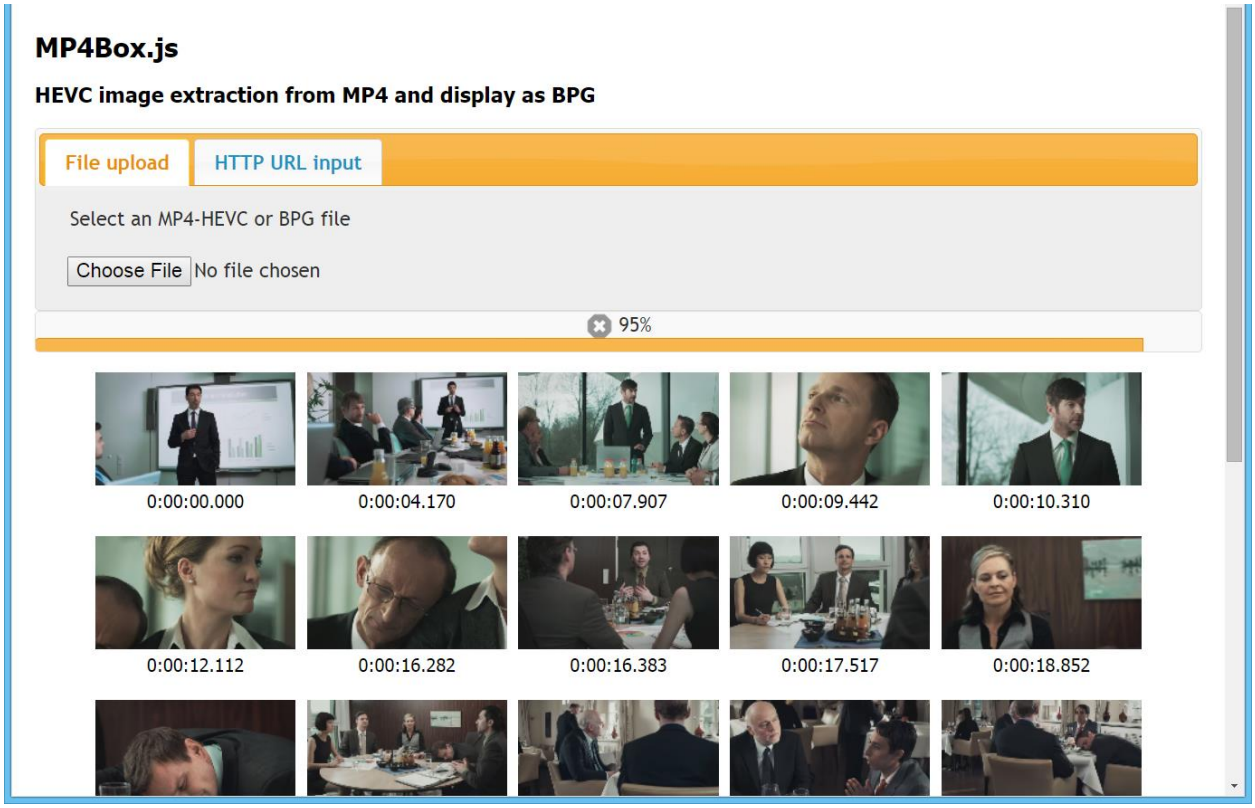
Finally, the item view shows for each item (currently only from the file level meta-box) the different properties of the item (id, name, …). For some item types (e.g. text, images), visualization is available.



3 HEVC Image Viewer

This section presents a demonstrated that has been already shown during the 112th MPEG meeting. With MP4Box.js, it is possible to extract sample media data from an MP4 and to process the data with additional libraries. In the HEVC Image viewer application, HEVC images stored in MP4 files, in a video track, are extracted, decoded using the BPG library and rendered on screen. Single images can also be viewed at full resolution and downloaded as BPG images. Support for the Image File Format is not fully available, but should be in the future weeks.

The application is available at:
http://download.tsi.telecom-paristech.fr/gpac/mp4box.js/hevcimageviewer



4 MP4 File Player

Current browsers support playback of MP4 files using two different approaches:

a) by setting the src attribute of a <video> or <audio> element with a URL to an MP4 file;
b) by appending MP4 byte streams to a MediaSource SourceBuffer.

There are several limitations to these modes:

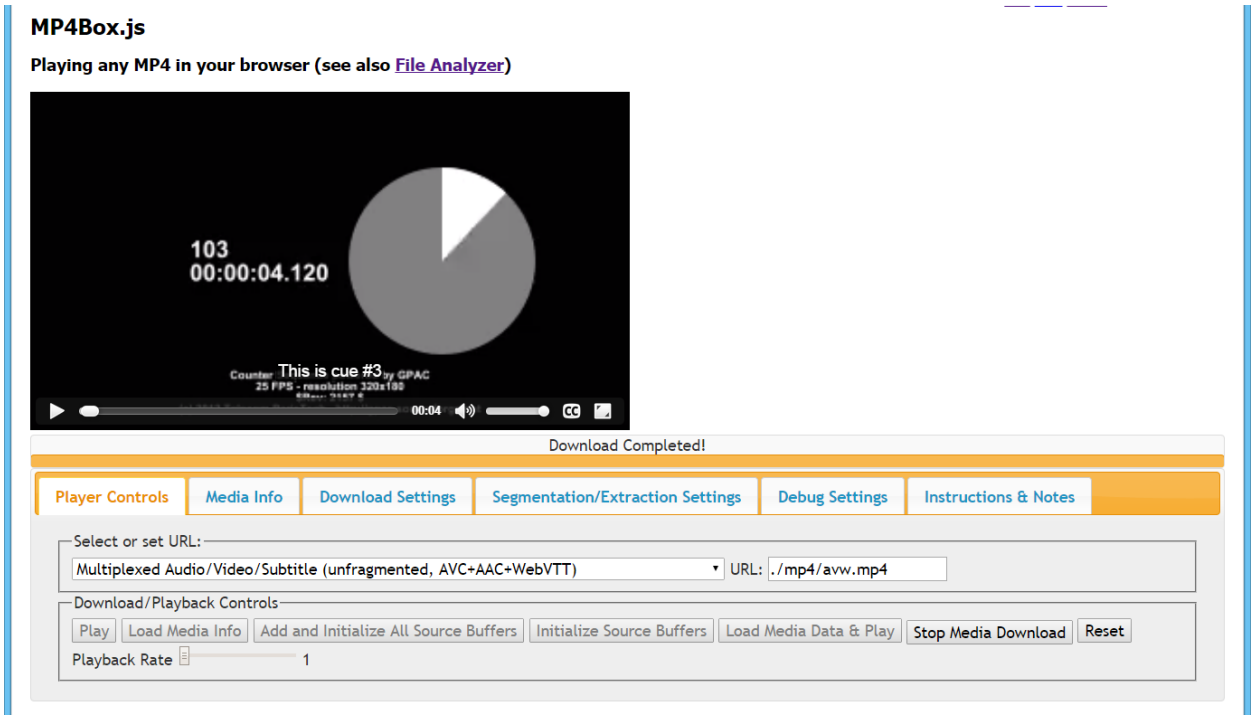
- Using option a), it is not possible to process, in the web application, the media data located in tracks that are not natively supported by the browser. This could be for instance WebVTT tracks, TTML tracks, BIFS tracks, SVG tracks, MPEG-7 tracks, JPEG tracks …
- Using option a), the playback of files that have the 'moov' box at the end of the file is typically not progressive. The file has to be downloaded entirely before it starts to play.
- Using option a), it is not possible/easy/practical to play large files as the file is in the end, entirely downloaded.
- Using option b), it is not possible to process non-fragmented files

With MP4Box.js, the following scenarios are possible:

1) progressively play non-fragmented files delivered over protocols other than HTTP, such as WebSockets, WebRTC Data Channel, …
2) progressively play large files without having to store them on the disk
3) progressively play files with the 'moov' box at the end
4) progressively play tracks natively supported by the browser and forward media data for tracks not supported natively to the web applications.

For all the above scenarios, traditional controls such as pause/resume, seek, playback rate changes, track disabling/enabling are available.

A demonstration player is available here:
http://download.tsi.telecom-paristech.fr/gpac/mp4box.js/



The current player allows playing any browser-support track types plus the following track types not supported natively:

- WebVTT in MP4 tracks
- SVG in MP4 tracks
- X3D in MP4 tracks

The current player also allows displaying a cover image for audio-only MP4, extracted from the MP4. The extracted cover image is the primary item of the top-level meta box, when the MP4 is also compatible with the Image File Format. Note that in this demonstration, the cover image is a JPEG image.

5 MP4 Packaging for HTML/CSS/JS files

The ISOBMFF specification allows through the use of the 'meta' box to use an MP4 file as a packaging format, with a so-called shadowing mechanism.

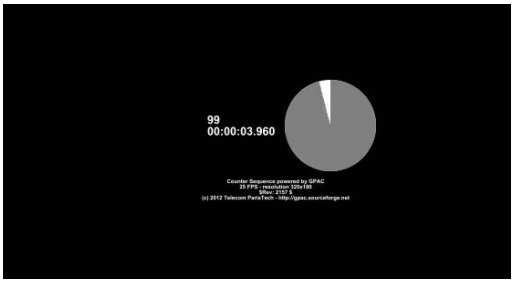
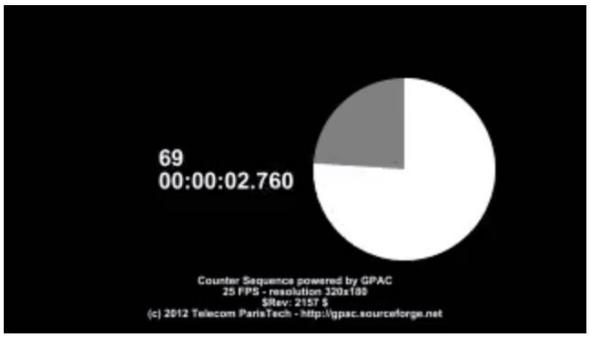
The Service worker API is a new API in browsers allowing to program client-side proxys that run in the browser.

We developed a Service Worker based on MP4Box.js (http://github.com/gpac/mp4box-sw/) which works in Firefox and Chrome.

This Service Worker intercepts requests to resources made by Web Applications in its scope. It checks if the response is an MP4 file. If so, it parses it and determines if there is HTML content in the MP4 file (stored as a primary item in a 'meta' box whose handler is 'html'). If so, the Service Worker forwards first the HTML content to the browser. All further requests made by the browser (because referenced from the HTML page) will be checked by the Service Worker to see if the resource is in the MP4 file. The MP4 file acts as a package for resources associated to the MP4 file.

Two demonstrations are available here: https://gpac.github.io/mp4box-sw/

- a basic example where the MP4 only stores an HTML file and a CSS file. The HTML contains a video track

MP4 Played without MP4Box Service Worker	MP4 Played in a Browser with MP4Box Service Worker
	This is an HTML page embeded in the MP4 file 

- a more complex example where the MP4 contains many assets (JS, PNG …) to be processed together with the video to provide an advanced viewing

MP4 Played without MP4Box Service Worker	MP4 Played in a Browser with MP4Box Service Worker
	  Next play: white Previous move: 3. Nc3 ... d5 Outcome odds: W:33.9% D:41.5% B:24.6% This board also appeared in: Kramnik vs Kasparov, 2000 (1-0)

6 Conclusion

We recommend MPEG to update the exploration document on interactivity support in MP4 based on the use cases demonstrated above.