Producing media content for the browsers using GPAC

Romain Bouqueau (GPAC Licensing)
Cyril Concolato (Telecom ParisTech)
Web & Media

- Web Browsers are more and more capable of playing media data
  - Either simply with `<audio>,<video>,<track>`
  - Or via JavaScript
    - For improved UI and other features with dedicated JS Players
    - For Adaptive Streaming ([MSE](#))
    - For Encrypted Media playback ([EME](#))
    - For unsupported codecs (HEVC) or formats (MPEG-2 TS)
    - For audio manipulations ([WebAudio](#))
- GPAC is an OSS that can help you produce content for these Browsers
What is GPAC?

- **Facts**
  - Open Source Software (LGPL), also commercially licensed by [GPAC-Licensing](http://example.com)
  - Started in 2000 NYC, now developed and maintained in Paris by [Telecom ParisTech](http://example.com)
  - Now officially hosted on [GitHub](http://example.com)
    - Pull-requests are welcome!

- **Tools**
  - Cross-Platforms (Linux, Mac, Win, Android, iOS)
    - 700 kloc, mainly C
  - MP4Box
    - Manipulations of MP4 files (and other files: MPEG-2 TS, AVI, SRT, …)
    - Real-time streaming server (RTP, DASH)
    - Now partly available in the browser: MP4Box.js
  - MP4Client
    - In-between a media player (VLC) and a browser
    - Support for many media formats and protocols (extensible)
    - Support for 2D/3D graphics rendering (VRML, X3D, SVG, MPEG-4 BIFS …)
    - Support for interactivity (JavaScript)
  - DashCast
    - Media Encoder based on Ffmpeg and MP4Box
MP4Client

DASH Player with Configurable SVG/JS interface
Basic A/V on the Web

- Initial market fragmentation
  - Many file formats (MP4, WebM, MKV, OGG, MPEG-2 TS…)
  - Many codecs
    - Video (H.264|AVC, HEVC, VP8, VP9, Theora, Daala)
    - Audio (MP3, AAC, USAC, Vorbis, Opus, …)

- Now getting stable
  - MP4 (ISO 14496-12, a.k.a. ISOBMFF and derivatives) with H.264 and AAC getting broad support
    - [http://www.openh264.org](http://www.openh264.org)
Advanced A/V on the Web

- Initial browser support through `<audio>`, `<video>`
  - Requires A/V content to be in the same file
  - Allows subtitles in separate files
  - But every thing is a file (no streaming!)

- Media Source Extension
  - Separate network pipeline from decoding pipeline
    - Deals with encoded data buffers (not files)
  - Ability to stream content, possibly live and adaptively
  - Supported in Chrome, IE11, FF (partial)
  - Some limitations
    - Requires "fragmented" MP4
    - MPEG-DASH implemented in JS on top of MSE
MP4Box for the Web

- “Simple” MP4 files muxed are supported by browsers (if not, file bugs!)

  ```bash
  MP4Box -add file.264 -add file.aac output.mp4
  ```

- For MSE, fragmentation can be done
  - Simply
    ```bash
    MP4Box -frag 1000 file.mp4
    ```
  - Within the DASH segmentation process (see next slide)

- Some oddities
  - Be careful with Chrome and frame-based sync (bug)
MP4Box for DASH on the Web

- Generation of DASH MPD and segments from input MP4 files
  - Continuous generation (-dash-live) or one-shot (-dash)

- Supported profiles
  - MPEG profiles: "onDemand", "live", "main", "simple", "full"
  - DASH-IF profiles: "dashavc264:live", "dashavc264:onDemand" and HEVC profiles (when finalized)

  \[\text{MP4Box } \text{-dash} \ 1000 \ \text{-profile dashavc264:live file.mp4}\]

- Some oddities
  - Playback in IE requires specific segmentation options
The Web is faced with 2 (!) subtitling formats:
- WebVTT
- TTML

ISO/IEC 14496-30 defines storage for both formats in MP4 files
- For multiplexing audio/video/subtitles in the same file
- For DASH usages

MP4Box supports both formats ([blog](#))

MP4Client partially supports rendering of WebVTT (no TTML yet)
GPAC and Encryption

- Common Encryption (CENC) with EME:
  - www.gpac.io/mp4box/encryption/common-encryption/

- Encryption:
  - $ MP4Box -cryption drm_file.xml movie.mp4 -out movie_encrypted.mp4
  - Microsoft PlayReady, Adobe Access, Google Widevine
    - Possibly all schemes: descriptive format for PSSH

```
<DRMInfo type="pssh" version="1" cypherOffset="9" cypherKey="0x677061636e56e6364726d746f66c31" cypherIV="0x677061636e56e6364726d746f66c31"> 
  <BS ID128="677061636e56e6364726d746f66c31"/>
  <BS value="2" bits="32"/>
  <BS ID128="0x279926496a7f5d25da69f2b3b2799a7f"/>
  <BS ID128="0x676cb88f302d1022799264985984045"/>
  <BS bits="8" string="CID-Toto"/>
  <BS ID128="0xccc0f2b3b279926496a7f5d25da692f6"/>
  <BS ID128="0xccc0f2b3b279926496a7f5d25da692d6"/>
</DRMInfo>
```

- Playback:
  - GPAC Player limited to GPAC SystemID (with unencrypted keys in PSSH)
    - $ MP4Client movie_encrypted.mp4
  - dash.js with CableLabs
    - Microsoft Playready, Google Widevine
  - https://groups.google.com/forum/#!topic/dashjs/4kQKcDhs99WA

GPAC - FOSDEM 2015
MP4Box.js

- New project
  - Client-side manipulation of MP4 file
  - JavaScript MP4Box
  - Hosted on GitHub

- Current Features:
  - Ability to parse (f)MP4
    - Non-linear parsing
  - On-the-fly generation of segments/fragments for playback through MSE
    - Including seek operations
  - Extraction of data for the Web app
    - metadata, unsupported formats ...

GPAC - FOSDEM 2015
Streaming of SVG Graphics

Goals
- Deliver graphics in streaming mode
  - Synchronously with media (if any)
  - Packaged or not with media data in media containers
  - Adaptively or not
- Use cases
  - Cartoons
  - Synchronized graphically-rich lyrics
  - Synchronized graphically-rich overlays

Current work at W3C
- SVG Streaming Community Group
- Draft spec
  - Definition of SVG stream
  - Storage in MP4 files

Work at GPAC
- Conversion of simple Flash animations to SVG streams
- Packaging/streaming of SVG streams in MP4 files
zenbuild

- **A component-level build system:**
  - Easing the build of free software projects
  - Seamless cross-compiling
  - Rescue console when things go wrong

- **GPAC, FFmpeg/libav, VLC, Mplayer, uTox**

- **Supported deps:**
  - fribidi libmad libvorbis tre gmplib libnettle libxau utox gnutls libogg libxcb vlc gpac libopenjpeglibxvidcore x264 jack libpng opencore-amr x265 alsa-lib liba52 libpthread openhevc xcb-proto libass librtp opus xproto faad2 libav libsamplelib portaudio zlib ffmpeg libfdk-aac libssl pthreads fontconfig libgCRYPT libsndfile sodium freetype2 libjpeg libtheora toxcore

- **Can emit subscripts:**
  - `$ ./make-extra.sh gpac > build_gpac.sh`
  - `$ ./build_gpac.sh <targetArchitecture>`

- [https://github.com/gpac/zenbuild](https://github.com/gpac/zenbuild)
Signals

- A modular system to ease multimedia communications
  - Initially: improve GPAC architecture at a low level
  - Allow more code reuse at a component higher level (in/out for ISOBMF, M2TS, DASH, etc.)

- Designed:
  - Layered:
    - Application (pipeline of libmm components)
    - libmm (multimedia specific)
    - libmodules (generic interface)
    - libsignals (generic signal/slots)

- C++11: strong types, exceptions, code reuse through patterns, etc.
- Interface to anything (like a pipe)
- Anywhere (single or multiple processes, cloud, …)

- Not free software yet (expected 2015):
  - R&D financed by industry, source code given with a restrictive license
  - Real-time DASHer, origin server (RTMP to HLS/MSS/HDS/DASH), Player in TVs, Cloud transcoder
  - Ask for an access to the codebase (not mature)
Hot Topics in GPAC (present or future)

- **ISOBMFF**
  - Storage of Tiled or Layered HEVC
  - Storage of Opus
  - Storage of HEVC Images (MPEG-style or BPG)

- **MPEG-DASH**
  - New DASH-IF Profiles
  - HBBTV2
  - Subtitling

- **New delivery**
  - Hybrid media delivery
  - WebRTC
    - Use GPAC as a WebRTC Peer (on-going work)

- **Platform tools**
  - [Zenbuild](#)
  - Signals
Questions