Producing media content for the browsers using GPAC

Romain Bouqueau (GPAC Licensing)
Cyril Concolato (Telecom ParisTech)

Web & Media

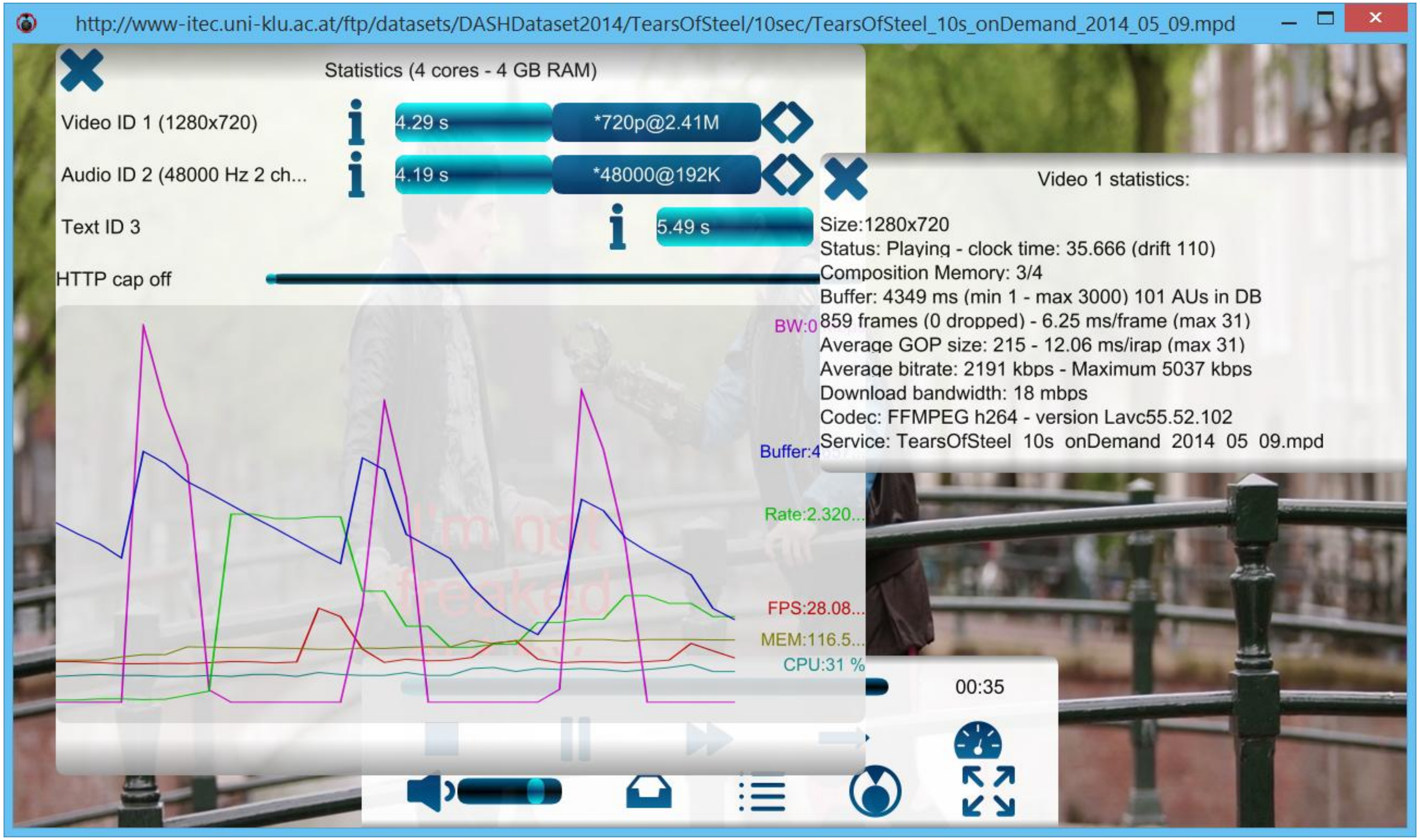
- Web Browsers are more and more capable of playing media data
  - Either simply with <audio>,<video>,<track>
  - Or via JavaScript
    - For improved UI and other features with dedicated JS Players
    - For Adaptive Streaming (MSE)
    - For Encrypted Media playback (EME)
    - For unsupported codecs (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
  - Started in 2000 NYC, now developed and maintained in Paris by Telecom ParisTech
  - Now officially hosted on GitHub
    - 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



http://www-itec.uni-klu.ac.at/ftp/datasets/DASHDataset2014/TearsOfSteel/10sec/TearsOfSteel_10s_onDemand_2014_05_09.mpd

Statistics (4 cores - 4 GB RAM)

Video ID 1 (1280x720) 4.29 s *720p@2.41M

Audio ID 2 (48000 Hz 2 ch... 4.19 s *48000@192K

Text ID 3 5.49 s

HTTP cap off

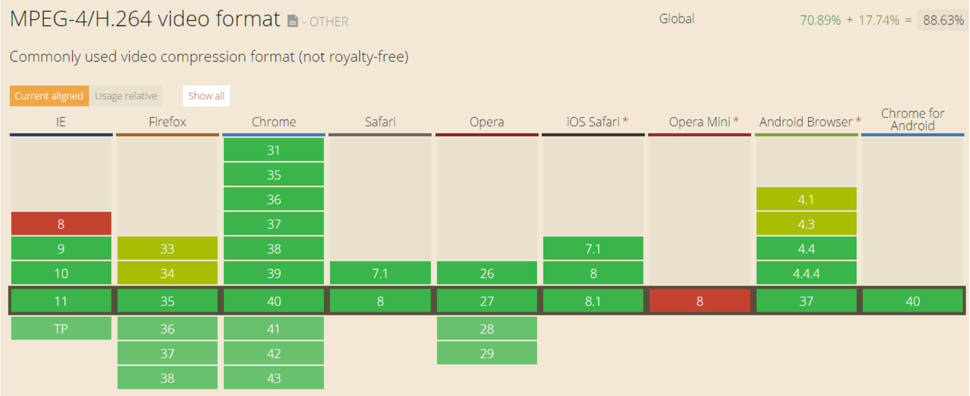
Video 1 statistics:

```
Size:1280x720
Status: Playing - clock time: 35.666 (drift 110)
Composition Memory: 3/4
Buffer: 4349 ms (min 1 - max 3000) 101 AUs in DB
859 frames (0 dropped) - 6.25 ms/frame (max 31)
Average GOP size: 215 - 12.06 ms/irap (max 31)
Average bitrate: 2191 kbps - Maximum 5037 kbps
Download bandwidth: 18 mbps
Codec: FFMPEG h264 - version Lavc55.52.102
Service: TearsOfSteel  10s  onDemand  2014  05  09.mpd
```

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/



MPEG-4/H.264 video format - OTHER

Global 70.89% + 17.74% = 88.63%

Commonly used video compression format (not royalty-free)

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		31						
		35						
		36					4.1	
8		37					4.3	
9	33	38			7.1		4.4	
10	34	39	7.1	26	8		4.4.4	
11	35	40	8	27	8.1	8	37	40
TP	36	41		28				
	37	42		29				
	38	43						

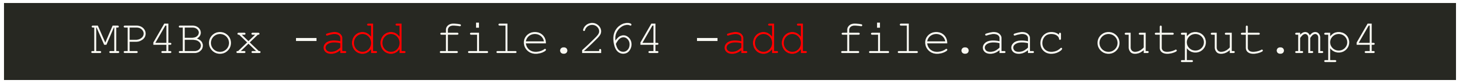
Advanced A/V on the Web

- 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

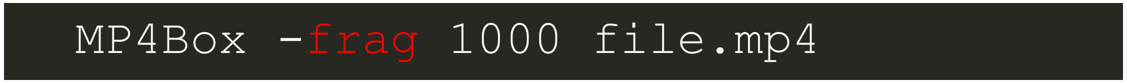
MP4Box for the Web

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



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

- For MSE, fragmentation can be done
  - Simply



```
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)

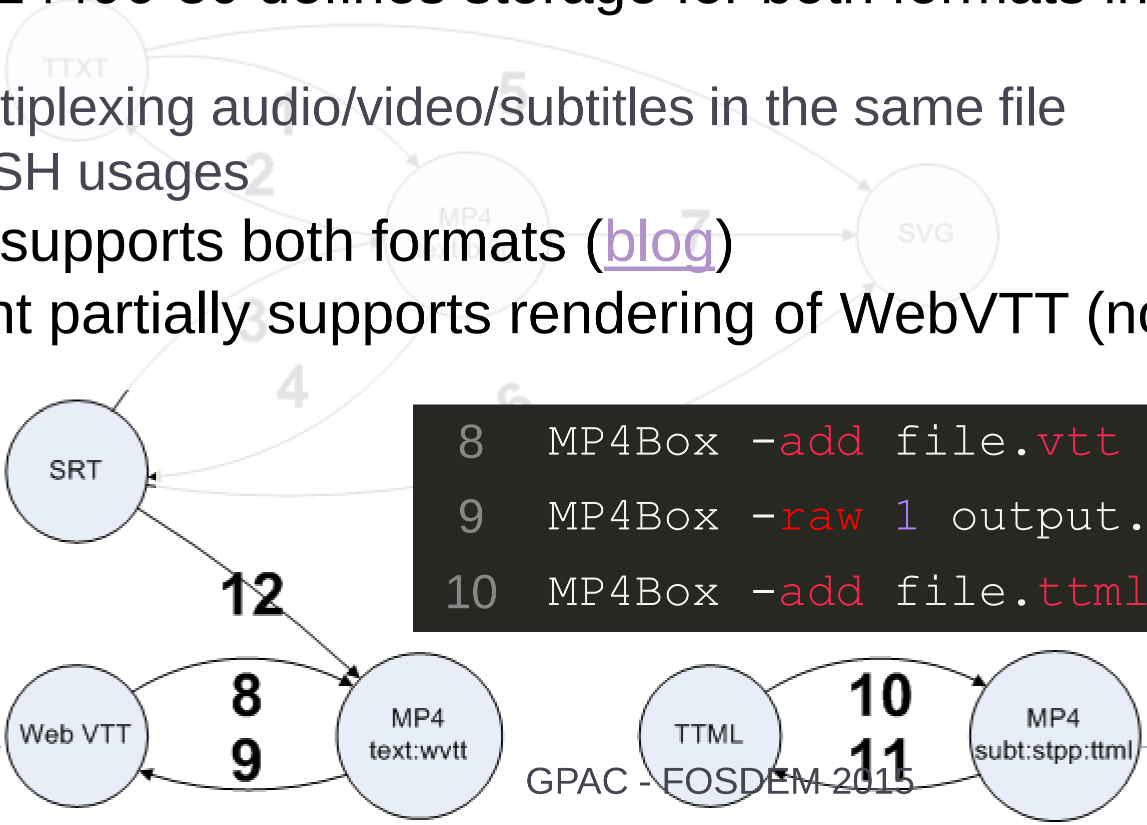
```
MP4Box -dash 1000 -profile dashavc264:live file.mp4
```

- Some oddities
  - Playback in IE requires specific segmentation options

GPAC and Subtitling for the Web

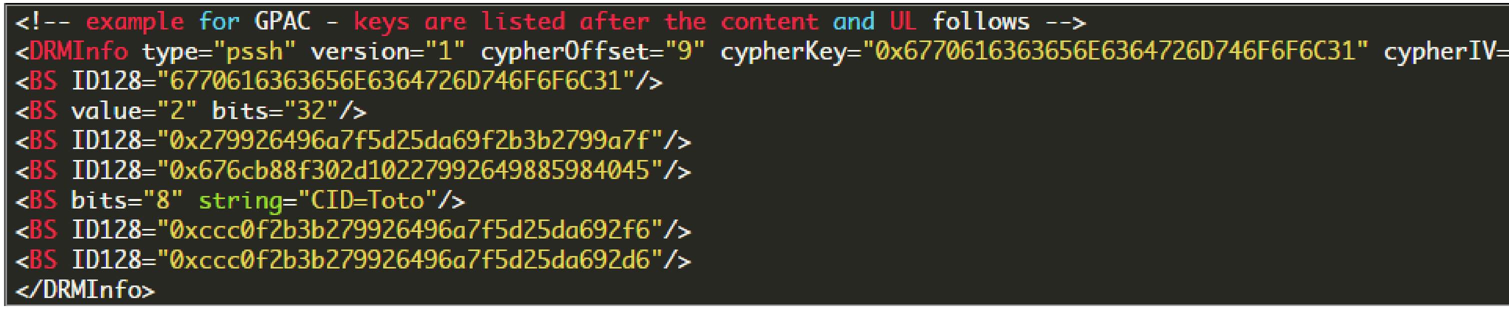
- The Web is faced with 2 (!) subtitling formats:
  - WebVTT
  - TTML
- ISO/IEC 14496-30 defines storage for both formats in MP4 files
  - For multiplexing audio/video/subtitles in the same file
  - For DASH usages
- MP4Box supports both formats (blog)
- MP4Client partially supports rendering of WebVTT (no TTML yet)

```
8   MP4Box -add file.vtt output.mp4
9   MP4Box -raw 1 output.mp4
10  MP4Box -add file.ttml output.mp4
```



GPAC and Encryption

- Common Encryption (CENC) with EME:
  - www.gpac.io/mp4box/encryption/common-encryption/
- Encryption:
  - `$ MP4Box -crypt drm_file.xml movie.mp4 -out movie_encrypted.mp4`
  - Microsoft PlayReady, Adobe Access, Google Widevine
    - Possibly all schemes: descriptive format for PSSH

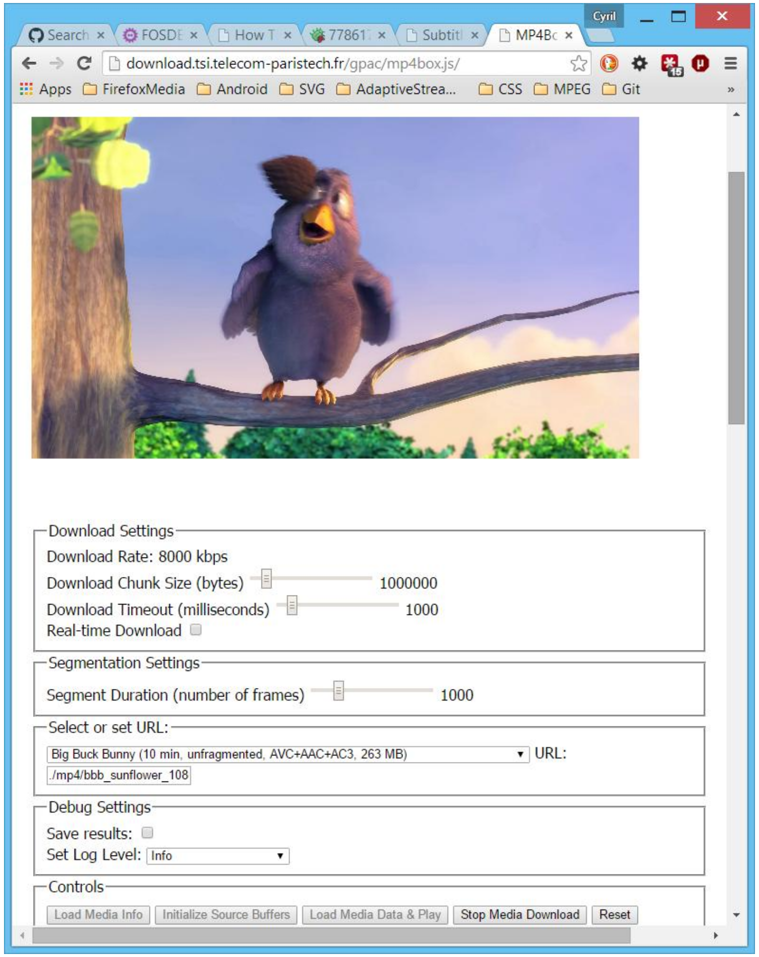


```
<!-- example for GPAC - keys are listed after the content and UL follows -->
<DRMInfo type="pssh" version="1" cypherOffset="9" cypherKey="0x6770616363656E6364726D746F6F6C31" cypherIV=
<BS ID128="6770616363656E6364726D746F6F6C31"/>
<BS value="2" bits="32"/>
<BS ID128="0x279926496a7f5d25da69f2b3b2799a7f"/>
<BS ID128="0x676cb88f302d10227992649885984045"/>
<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/4kQkDhs9BWA

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 …



Download Settings

Download Rate: 8000 kbps

Download Chunk Size (bytes) 1000000

Download Timeout (milliseconds) 1000

Real-time Download □

Segmentation Settings

Segment Duration (number of frames) 1000

Select or set URL:

Big Buck Bunny (10 min, unfragmented, AVC+AAC+AC3, 263 MB) URL: ./mp4/bbb_sunflower_108

Debug Settings

Save results: □

Set Log Level: Info

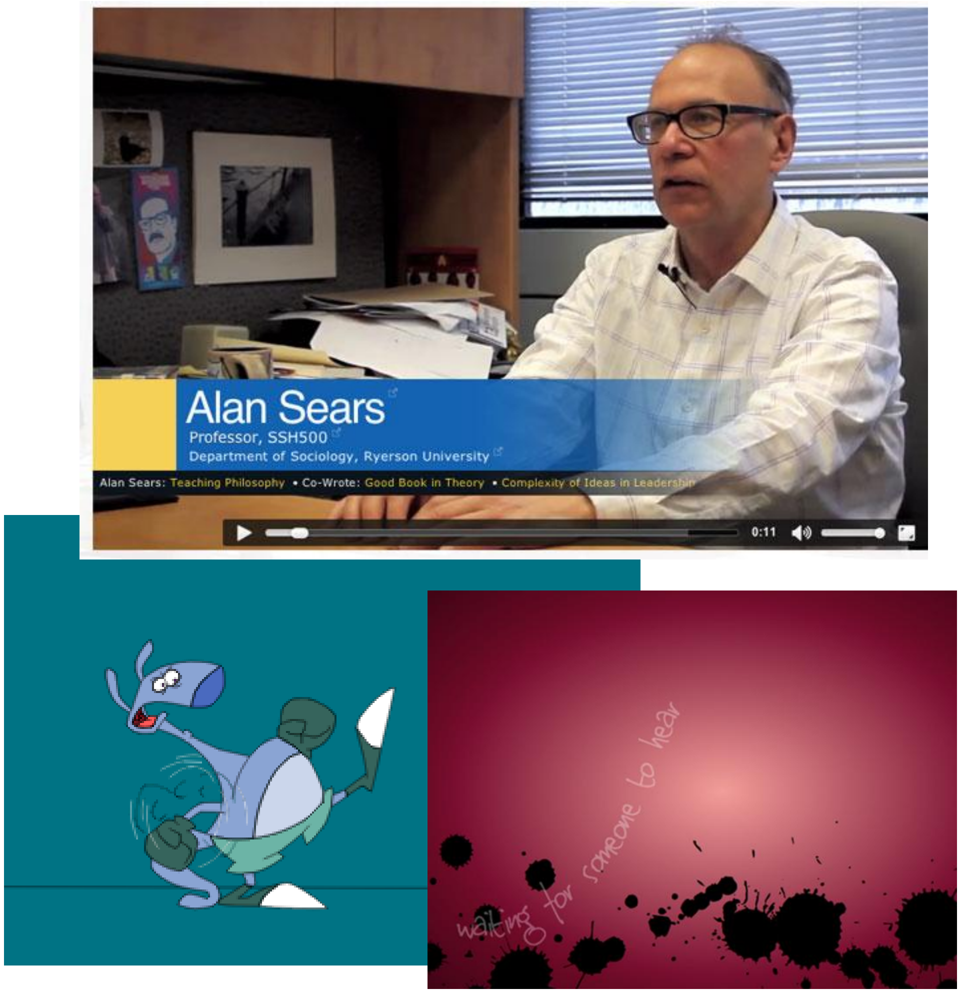
Controls

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


Alan Sears
Professor, SSH500
Department of Sociology, Ryerson University
Alan Sears: Teaching Philosophy • Co-Wrote: Good Book in Theory • Complexity of Ideas in Leadership
0:11
walking for someone to hear

zenbuild

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

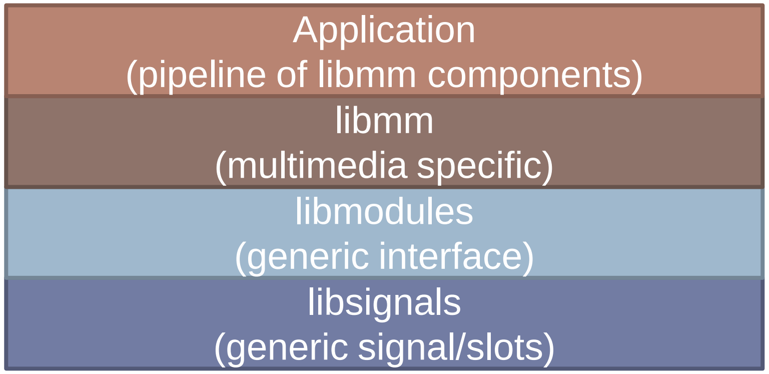
- GPAC, FFmpeg/libav, VLC, Mplayer, uTox
  - Supported deps:
    - fribidi libmad libvorbis tre gmplib libnettle libxau utox gnutls libogg libxcb vlc gpac libopenjpeglibxvidcore x264 jack libpng opencore-amr x265 alsa-lib liba52 libpthread openhevc xcb-proto libass librtmp opus xproto faad2 libav libsamplerate portaudio zlib ffmpeg libfdk-aac libsdl 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

Signals

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

- Designed:
  - Layered:

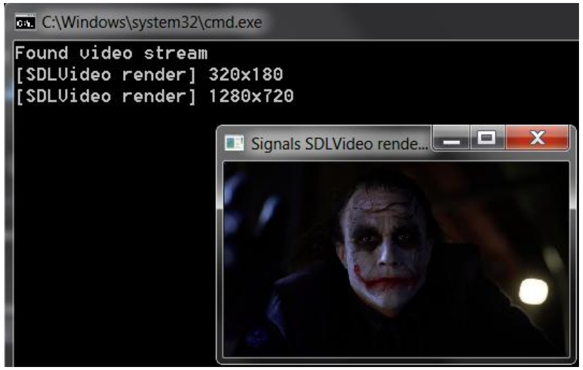


Application
(pipeline of libmm components)

libmm
(multimedia specific)

libmodules
(generic interface)

libsignals
(generic signal/slots)



```
Found video stream
[SDLVideo render] 320x180
[SDLVideo render] 1280x720
```

  - 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