What’s in GPAC?

- Started in 2000. Mostly written in C.
- 40+ standards from MPEG, IETF, 3GPP, DVB.
- Includes: packager, streamers, players.
- GPAC’s player integrates a scene graph for media composition with JavaScript support.
- Windows/Linux/OSX, iOS, Android nightly installers available.

Adding more stuff year after year

In a Nutshell, gpac...

... has had 9,599 commits made by 58 contributors
representing 753,938 lines of code

... is mostly written in C
with an average number of source code comments

... has a well established, mature codebase
maintained by a large development team
with increasing Y-O-Y commits

... took an estimated 206 years of effort (COCOMO model)
starting with its first commit in July, 2005
ending with its most recent commit 3 days ago

Source: https://www.openhub.net/p/gpac/

Maintenance costs getting higher

Challenges

Keep API unified and secure.

Make build, distribution, configuration, and execution modular.

Offer deterministic modes to ease bug reporting and debugging.

GPAC suffered from a 15-years old design not modular enough.

Go modular!
Filters

What is Filters

Filters is a major re-architecture of GPAC's streaming tools

- Improve modularity and reusability of media processing blocks
- Easily support more workflows than the original MP4Box & MP4Client
- Binary compatibility of outputs with previous version
- More test coverage (from really low to 90% FUNC / 75% LOC)
- Unmodified usage of MP4Box & MP4Client, libgpac API 99% backward compatible
- Auto-generated documentation for all GPAC applications and for all filters (see https://github.com/gpac/gpac/wiki)

How does that work?

- Filters represents processing as a graph
- Inputs, outputs, muxers, demuxers, decoders, de/en-crypters, rescalers, encoders, display are nodes of the graph

What does it bring you?

- Serialized IO over sockets/pipes/files (AES-128 protection)
- Encoding/Transcoding in your classical MP4Box DASH setup
- On the fly CENC encryption/decryption (no local MP4 needed)
- Media processing from live sessions (RTP/RTSP, MPEG2 TS, DASH/HLS) and grabbers (FFMPEG)
- Raw audio/video input/output
- Dual-packaging DASH + HLS (CMAF)
- Graphics overlay
- HEVC Tile splitting & merging (VR) (2016 EBU meetup)
- And much more through custom filter chains !

A few examples

- gpac -i source.avc -o test.mp4
- gpac -i source1.mp4 -i source2.mp4 -o test.mpd:dur=2.5
- gpac -i source.mp4 cecrypt:cfile=DRM.xml @ -o protected.mp4
- gpac -i source.avc:FID=1 ffsws:osize=512x512:SID=1 @ ffenc:c=avc:fintra=1:FID=EV1 ffsws:osize=256x256:SID=1 @ ffenc:c=avc:fintra=1:FID=EV2 -o file.mpd:profile=live:SID=EV1,EV2

Roadmap to 1.0

- Freeze the filter API
- Replace javascript interpreter and make filters scriptable
- Add support for FFMPEG filters
- Improve security (fuzzbot)

More about GPAC (bonus)

History of GPAC versions

- 2003-2012: pre 0.5: BIFS/VRML/SVG/LASeR, ISOBMFF, MPEG-2 TS, RTP/RTSP, ISMA E&A,...
- 2012/05: 0.5: DASH and HLS
- 2015/01: 0.5.2: HEVC/SHVC, MP4Box.js, zenbuild, Signals
- 2016/02: 0.6: HbbTV, Subtitles, TEMI, Scalable HEVC
- 2017/04: 0.7: VR, Image File Formats, mobile (iOS, Android)
- 2019/06: 0.8: AV1/Opus, ATSC3, ProRes/QTFF, HDR, full CENC, HEIF
- 2019/12: 0.9: Filters