GSTREAMER 1.16 AND BEYOND

FOSDEM Open Media Devroom

02 February 2019, Brussels

Tim-Philipp Müller < tim@centricular.com >
WHO AM I?
WHAT IS GSTREAMER?

Most of you know this, so key points only.
Framework for multimedia processing.

Cross-platform, toolkit agnostic.

Any and all use cases.

Set of libraries and plugins.

Abstract API, very extensible.

We often wrap other libraries.
Low-level API and high-level API:

playbin, encodebin, RTSP server,
non-linear editing, WebRTC, VoIP etc.
Integration with other frameworks and projects
e.g. WebKit/Blink, OpenGL, Vulkan
Windows, Android, iOS, macOS.

Goal is to adapt to and integrate with other platforms and frameworks (inputs, outputs, decoders, DSPs/GPUs..)
SO, WHAT HAVE WE BEEN UP TO?
RELEASES!

- goal: 6-monthly(ish) release schedule
- 1.14: March 2018
- 1.16: February 2019
- 1.18: Summer/Autumn 2019 (hopefully)
WE'VE MOVED TO GITLAB!

bugzilla.gnome.org -> gitlab.freedesktop.org

Modern workflows:

• Merge Requests

• better patch review tools

• pre-merge continuous integration!
SOME THINGS THAT LANDED
IN 1.14 AND AFTER ...
WEBRTC

"How do I stream to my web browser?"

Low latency, works pretty much everywhere.

webrtcbin + gstwebrtc library

Leverage all of GStreamer:
transmuxing, hw-acceleration etc.
WEBRTC IMPROVEMENTS COMING UP:

• datachannel

• bundle support

• FEC (Forward Error Correction)

• RTX (Retransmission)

• lots of bug fixes and interoperability improvements
FORWARD ERROR CORRECTION (FEC)

• ULPFEC support (uneven level protection)
• latency/bandwidth trade-off vs. retransmission
• RTP / WebRTC
• RTSP server RECORD
• Todo: RTSP server PLAY
AV1 VIDEO CODEC SUPPORT

• Royalty-free next-generation video codec.

• support in Matroska and QuickTime/MP4 containers

• more configuration options and input formats for the AOMedia AV1 encoder
EMBEDDED SYSTEMS

Lots of improvements!

--> Olivier's talk later
SRT: SECURE RELIABLE TRANSPORT

• new video streaming protocol

• replacement for RTMP

• refactored source/sink elements
PLAYBIN3 GAPLESS PLAYBACK AND PRE-BUFFERING SUPPORT

Loading next URI as soon as loading of current track finishes.

Pre-buffering in the encoded domain, so no more decoder starvation.
NVDEC: HARDWARE-ACCELERATED VIDEO DECODING FOR NVIDIA

To complement already-existing nvenc encoder.

CUDA/SDK 9 support

New: VP8/VP9 decoding, H265 encoding

(Existing: H264, JPEG, MPEG-1/2/4)
SOME COOL THINGS THAT WILL LAND IN 1.16 ...
LOTS OF OPTIMISATIONS

• appsrc + appsink buffer list support
• fewer allocations
• non-interleaved audio
• latency improvements
LOTS OF OPTIMISATIONS (CONT'D)

• videodecoder / encoder: better parallelism

• udpsrc buffer pool

• OpenGL: dmabuf uploader + related improvements
CLOSED CAPTION SUPPORT

• SDI capture and output

• MOV

• MPEG-TS / ATSC / DVB

• SCC / MCC files
CLOSED CAPTION ELEMENTS

• ccextractor, cccombiner
• ccconverter
• line21decoder
• etc.
WEBKIT WPE-BASED WEB BROWSER SOURCE ELEMENT

Stream the output of a browser / web page!
RTSP SERVER

- perf improvement for TCP interleaved mode
- multicast handling fixes
- FEC support

Coming up:

- ONVIF trick modes + replay modes
INTEL MEDIA SDK PLUGIN (MSDK)

- Lots of improvements
- dmabuf support
- more codec support (VP9 etc.)
- video pre/post-processing (VPP)
MISSION: PLUGIN MOVES
AND MODULE CONSOLIDATION

Ongoing effort to move things out of -bad.

Latest:

• moved VideoAggregator + compositor to -base
• moved OpenGL video mixers to -base

Next (hopefully): GstPlayer
THE FUTURE ...
NEURAL NETWORKS

Hottest topic at last year's GStreamer Conference!

Talks here: https://gstconf.ubicast.tv
SCALEABLE STREAMS

SHVC etc.

Signalling, architecture, infrastructure.

Handle enhancement streams
in decodebin3 / playbin3
QUASI-INSTANTANEOUS TRICK MODE SPEED CHANGES
PERFORMANCE OPTIMISATIONS..

Everywhere, of course.

Lots of things in the pipeline to improve performance, latency and memory usage.
WHAT ELSE ?
MESON BUILD SYSTEM STATUS UPDATE:

Almost complete now!

• options for almost everything

• cerbero now builds GStreamer with Meson on all OS!

• MSVC build via gst-build or cerbero possible now

• Autotools will likely be phased out after 1.16
RUST

"Fast, safe and productive - pick three."
RUST

Perfect language for us technically.

Excellent C compatibility.

Fantastic community.

Superb ecosystem.

Lots of positive experience gained in the last year(s).
RUST (CONT'D)

No plans to rewrite everything.

No plans to make it a hard dependency.

We can do a lot without breaking backwards compat.

But let's play with it, experiment, gain more experience.

Need to make sure it works for all our users!

Something for the longer term.
GSTREAMER RUST BINDINGS

Bindings + plugins are official + upstream now!

Should be in really good shape.

Much more complete.

Many releases, many new users.

Subclassing/plugins possible for all important types.

Write more examples and plugins in Rust!

--> Zeeshan's talk later

--> Sebastian's talk tomorrow
WHAT ELSE IS ON OUR RADAR?

• SDI-over-IP standards (NDI source now on github)
• Better OpenCV integration, CUDA
• High dynamic range (HDR) video improvements
• VR
THAT'S ALL FOLKS
THANK YOU (AND THANKS TO THE ORGANISERS!)
QUESTIONS? COMMENTS?
PS:

- follow us on Twitter @GStreamer
- find us on IRC in #gstreamer on FreeNode
- GStreamer hackfest in spring
- GStreamer Conference + Hackfest in Lyon around the end of October
- [https://gstreamer.freedesktop.org/releases/1.16/](https://gstreamer.freedesktop.org/releases/1.16/)