

archive.org

DECLARATION OF MINA CHING

1. I am a Records Request Processor at the Internet Archive. I make this declaration of my own personal knowledge.

2. The Internet Archive is a website that provides access to a digital library of Internet sites and other cultural artifacts in digital form. Like a paper library, we provide free access to researchers, historians, scholars, and the general public. The Internet Archive has partnered with and receives support from various institutions, including the Library of Congress.

3. The Internet Archive has created a service known as the Wayback Machine. The Wayback Machine makes it possible to browse more than 450 billion pages stored in the Internet Archive's web archive. Visitors to the Wayback Machine can search archives by URL (i.e., a website address). If archived records for a URL are available, the visitor will be presented with a display of available dates. The visitor may select one of those dates, and begin browsing an archived version of the Web. Links on archived files in the Wayback Machine point to other archived files (whether HTML pages or other file types), if any are found for the URL indicated by a given link. For instance, the Wayback Machine is designed such that when a visitor clicks on a hyperlink on an archived page that points to another URL, the visitor will be served the archived file found for the hyperlink's URL with the closest available date to the initial file containing the hyperlink.

4. The archived data made viewable and browseable by the Wayback Machine is obtained by use of web archiving software that automatically stores copies of files available via the Internet, each file preserved as it existed at a particular point in time.

5. The Internet Archive assigns a URL on its site to the archived files in the format http://web.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL] aka an "extended URL". Thus, the extended URL http://web.archive.org/web/19970126045828/http://www.archive.org/ would be the URL for the record of the Internet Archive home page HTML file (http://www.archive.org/) archived on January 26, 1997 at 4:58 a.m. and 28 seconds (1997/01/26 at 04:58:28). The date indicated by an extended URL applies to a preserved instance of a file for a given URL, but not necessarily to any other files linked therein. Thus, in the case of a page constituted by a primary HTML file and other separate files (e.g., files with images, audio, multimedia, design elements, or other embedded content) linked within that primary HTML file, the primary HTML file and the other files will each have their own respective extended URLs and may not have been archived on the same dates.

INTERNET ARCHIVE

archive.org

6. Attached hereto as Exhibit A are true and accurate copies of browser screenshots of the Internet Archive's records of the archived files for the URLs and the dates specified in the attached coversheet of each screenshot.

7. Attached hereto as Exhibit B are true and accurate copies of the Internet Archive's records of the archived files for the URLs and the dates specified in the attached coversheet of each file.

8. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

DATE: March 5, 2026

Mina Ching

EXHIBIT A

https://web.archive.org/web/20160106210441/http://gstreamer.freedesktop.org/data/events/gstreamer-conference/2015/

INTERNET ARCHIVE Wayback Machine http://gstreamer.freedesktop.org/data/events/gstreamer-conference/2015/ Go DEC JAN FEB 15 captures 7 Dec 2015 - 28 Aug 2025 2015 06 2016 2021 About this capture

Index of /data/events/gstreamer-conference/2015

Name	Last modified	Size	Description
Parent Directory		-	
Andres Colubri - Interactive video playback and capture in the Processing Language via GStreamer.pdf	13-Oct-2015 11:09	7.6M	
Balázs Kreith - Multi-path RTP in GStreamer.pdf	29-Oct-2015 05:49	2.1M	
Brion Vibber - ogvjs.pdf	07-Oct-2015 22:21	1.2M	
Dimitrios Katsaros - Hyperspectral Imagery in GStreamer.pdf	29-Oct-2015 05:49	1.0M	
Edward Hervey - Continuous Integration: An update.pdf	29-Oct-2015 05:49	104K	
Edward Hervey - decodebin3.pdf	09-Oct-2015 04:33	276K	
Florent Thiery- gst-gtklaunch-1.0.pdf	29-Oct-2015 05:49	783K	
GStreamer Conference 2015 - Brochure.pdf	05-Oct-2015 06:35	787K	
Håvard Graff - PEAQ performance: Audiocodecs, Jitterbuffers, and an old friend.pdf	29-Oct-2015 05:54	740K	
Hans Verkuil - Colorspaces and HDMI.pdf	12-Oct-2015 09:28	188K	
Jan Schmidt - Distributed Acoustic Triangulation.pdf	10-Oct-2015 01:52	1.3M	
Jan Schmidt - Stereoscopic 3D Redux.pdf	10-Oct-2015 01:51	1.5M	
Jussi Pakkanen - Done in 6.0 second - a new build sytem for GStreamer?.pdf	29-Oct-2015 05:49	232K	
Justin Kim - Another Android Camera Source for GStreamer.pdf	29-Oct-2015 05:49	758K	
Luis de Bethencourt - How to Contribute to GStreamer.pdf	12-Oct-2015 09:03	1.2M	
Marcin Kolny - Remote GStreamer debugger.pdf	29-Oct-2015 05:49	1.3M	
Marcin Kolny - gstreamermm c++ wrapper.pdf	29-Oct-2015 05:49	147K	
Marianna Buschle - Industrial Application Pipelines with GStreamer.pdf	29-Oct-2015 05:49	3.3M	
Matthew Waters - OpenGL for the GStreamer pipeline.pdf	14-Oct-2015 05:25	257K	
Melissa Montero - GST Daemon and Interpipes: A simpler way to get your applications done .pdf	29-Oct-2015 05:49	2.0M	
Nicolas Dufresne - Demistifying allocation query.pdf	04-Jan-2016 07:10	283K	
Nicolas Dufresne - V4L2 GStreamer elements.pdf	29-Oct-2015 05:49	50K	
Ramesh V - Handling interleaved and non-interleaved streams with GStreamer.pdf	29-Oct-2015 05:59	386K	
Romain Picard - The HeliosTv Distributed DVB stack.pdf	19-Oct-2015 01:48	3.8M	
Sebastian Dröge - GstPlayer Cross-platform Playback API.pdf	10-Oct-2015 01:50	494K	
Sebastian Dröge - Synchronised multi-room media playback and distributed live media processing and mixing with GStreamer.pdf	10-Oct-2015 01:51	1.8M	
Thiago Santos - Analyzing Caps Negotiation using GstTracer.pdf	29-Oct-2015 05:49	229K	
Thibault Saunier - Non Linear Engine: replacement for GnonLin.pdf	29-Oct-2015 05:49	149K	
Thibault Saunier - What is new in GstValidate 1.6.pdf	29-Oct-2015 05:49	59K	
Tim-Philipp Müller - GStreamer State of the Union.pdf	19-Oct-2015 15:29	77K	
Vanessa Chipirrás Navalón - OpenCV and GStreamer (Demo Video).mkv	29-Oct-2015 05:49	6.5M	
Vanessa Chipirrás Navalón - OpenCV and GStreamer.pdf	29-Oct-2015 05:49	307K	
Wim Taymans - Camera Sharing and Sandboxing with Pinos.pdf	13-Oct-2015 04:07	320K	
Wim Taymans - ORC experiments.pdf	29-Oct-2015 05:49	280K	

Apache/2.2.22 (Debian) Server at gstreamer.freedesktop.org Port 80

https://web.archive.org/web/20160109011525/http://gstreamer.freedesktop.org/conference/2015/index.html

INTERNET ARCHIVE Wayback Machine http://gstreamer.freedesktop.org/conference/2015/index.html Go DEC JAN FEB 09 2015 2016 2019 43 captures 1 Jul 2015 - 30 May 2024 About this capture

gstreamer

open source multimedia framework

GStreamer Conference 2015

8-9 October 2015 · The Convention Centre Dublin · Dublin, Ireland

The **6th GStreamer Conference** was again co-located with the *Embedded Linux Conference Europe (ELCE)* and *LinuxCon Europe*. The event was held on Thursday and Friday **8-9 October 2015** at the Convention Centre Dublin (CCD) in **Dublin, Ireland**.

Registration is now closed.

Post-conference follow-up

The **6th GStreamer Conference** is now over. Thanks to everyone who attended, we hope you enjoyed it as much as we did. Many thanks to our speakers who have again presented a fantastic range of topics, and last but not least of course a big thanks to our sponsors and partners without whom the conference would not have been possible in this form.

Talks have been recorded by Ubicast and will be put up on the GStreamer Conference videos web site once they're ready. We will announce on the mailing lists, Planet GStreamer and twitter when the videos are up.

Slides for some of the talks and the lightning talks can be found in the GStreamer Conference 2015 slides folder.

Linux Weekly News (LWN) have attended the conference and have been covering some of the talks (content might be subscriber only when initially published but is usually available freely for non-subscribers the week after):

- A simpler playback tool for GStreamer
- 3D video and device mediation with GStreamer
- Upcoming work in and around GStreamer
- Experimental applications of GStreamer

Details

The GStreamer Conference is a conference for developers, decision-makers, community members, industry partners, and anyone else interested in the GStreamer multimedia framework or Open Source and cross-platform multimedia.

Showcase or Lightning Talks

If you don't think you have enough material for a full talk, but perhaps just want to showcase something you're working on, or present a problem or challenge you are facing or have solved, and that you think might be interesting for others to hear about, why not submit a ca. 5-minute mini-talk for our *lightning talk/showcase* event?

Simply send a quick mail to gstreamer-conference@lists.freedesktop.org above to reserve your slot (subject to availability). There is no deadline for Lightning Talk submissions, slots will be allocated on a first-come first-serve basis.

Social Event

There will again be a social event for registered attendees on Thursday 8 October 2015 in the evening from 19.00-23.00h at **Toners Pub**, 139 Lower Baggot Street, Dublin 2, D02 N231, between St Stephen's Green and Merrion Square (map).

Food and Drink will be provided, including vegetarian options.

Please don't forget to bring your badge.

Schedule and speakers list

Click here for the full schedule

We have an exciting range of topics and speakers lined up for you again this year:

- Interactive video playback and capture in the Processing Language via GStreamer
  Andres Colubri
- Faster transcoding with GStreamer in the cloud
  Thiago Sousa Santos, Samsung
- Tiled Streaming of UHD video in real-time
  Arjen Veenhuizen, TNO
- GStreamer and WebKit
  Philippe Normand, Igalia
- Hardware accelerated multimedia on TI's Jacinto 6 SoC
  Pooja Prajod, Texas Instruments
- Demystifying the allocation query
  Nicolas Dufresne, Collabora
- Synchronised multi-room media playback and distributed live media processing and mixing with GStreamer
  Sebastian Dröge, Centricular
- Implementing a WebRTC endpoint in GStreamer: challenges, problems and perspectives
  Dr Luis López, Kurento
- OpenGL Desktop/ES for the GStreamer pipeline
  Matthew Waters, Centricular
- Robust lipsync error detection using gstreamer and QR Codes
  Florent Thiery, Ubicast
- GStreamer VAAPI: Hardware-accelerated decoding and encoding on Intel hardware
  Víctor M. Jáquez L., Igalia
- Colorspaces and HDMI
  Hans Verkuil, Cisco
- GStreamer State of the union

Sponsors

Silver Sponsors

Google

Centricular

COLLABORA

Bronze Sponsors

pexip

igalia

Special Media Partners

UBICAST
#digital #learning #video

Press Partners

LWN.net
News from the source

Interested in becoming a sponsor ?

- GStreamer State of the union
  Tim-Philipp Müller, Centricular
- Video Filters and their applications
  Sanjay Narasimha Murthy, Samsung
- Camera Sharing and Sandboxing with Pinos
  Wim Taymans, RedHat
- Stereoscopic (3D) Video in GStreamer Redux
  Jan Schmidt, Centricular
- Bin It! AKA, How to use bins and bin subclasses to keep state local and easily manage dynamic pipelines
  Vivia Nikolaidou, ToolsOnAir
- The HeliosTv Distributed DVB stack
  Romain Picard, SoftAtHome
- How to contribute to GStreamer
  Luis de Bethencourt, Samsung
- GstPlayer - A simple cross-platform API for all your media playback needs
  Sebastian Dröge, Centricular
- Improving GStreamer performance on large pipelines: from profiling to optimization
  Miguel París
- Kurento Media Server: experiences bringing GStreamer capabilities to WWW developers
  José Antonio Santos
- Going Live with ToolsOnAir's GStreamer-based broadcast mixing architecture
  Heinrich Fink, ToolsOnAir
- Distributed Acoustic Triangulation
  Jan Schmidt, Centricular
- Chromium: a new media Backend based on GStreamer
  Julien Isorce, Samsung
- ogv.js: bringing open codecs to Safari and IE with emscripten
  Brion Vibber, Wikimedia
- Bringing GStreamer to Radio Broadcasting
  Marcin Lewandowski
- Daala, NETVC, and the Alliance for Open Media: the plan for royalty free video
  Thomas Daede, Mozilla
- Simple techniques of pipeline performance measurements and time profiling of individual elements
  Kyrylo Polezhaiev
- Pointing cameras at TVs: when HDMI video-capture is not an option
  Will Manley, stb-tester
- Decodebin3, or dealing with modern playback use-cases
  Edward Hervey, Centricular

Lightning Talks:

- Industrial application pipelines · *Marianna Smidth Buschle, QTechnology*
- Hyperspectral imagery
  Dimitrios Katsaros, QTechnology
- gst-gtk-launch-1.0
  Florent Thiery, Ubicast
- liborc (JIT SIMD generator) experiments
  Wim Taymans, RedHat
- V4L2 GStreamer elements update
  Nicolas Dufresne, Collabora
- Analyzing caps negotiation with GstTracer
  Thiago Sousa Santos, Samsung
- Know your queues! queue, queue2, multiqueue, netbuffer and all that
  Tim-Philipp Müller
- Nle: A new design for the GStreamer Non Linear Engine
  Thibault Saunier
- What is new in GstValidate
  Thibault Saunier
- Continuous Integration update
  Edward Hervey
- Remote GStreamer Debugger
  Marcin Kolny
- gstreamermm C++ wrapper
  Marcin Kolny
- Multipath RTP (MPRTP) plugin in GStreamer
  Balázs Kreith
- OpenCV and GStreamer
  Vanessa Chipi
- Handling Interleaved and Non-Interleaved streams with GStreamer
  Ramesh Venkatachalapathy, LGE
- Done in 6.0 seconds: a new build system for GStreamer?
  Jussi Pakkanen
- Gst daemon and interpipes: A simpler way to get your application done
  Melissa Montero, RidgeRun
- Another android camera source for Gstreamer
  Justin Kim, Collabora
- ...
- *Submit your lightning talk now!*

Who should attend?

Anyone interested in open source cross-platform multimedia is encouraged to join us at the GStreamer Conference 2015. Whether you are a hobbyist or a student interested in the field of multimedia, or working for a company using or looking to

Pinterest Ex. 1010, Page 8 of 57

Whether you are a hobbyist or a student interested in the field of multimedia, or working for a company using or looking to use GStreamer in their products, this is the perfect event for you to network at and to learn more about the latest technological breakthroughs in and around GStreamer. The talks will target both developers and managers.

Registration

Registration is now open and will stay open until the conference.

Sign up for the GStreamer Conference here

There will be separate registration options for both professionals and for students and hobbyists:

Professional	275 USD incl. lunch (until 31 July)
Professional	385 USD incl. lunch (from 1 August)
Student/hobbyist	100 USD incl. lunch

The registration fee includes continental breakfast and lunch at the venue on both conference days.

For those uncertain about whether to register as a professional or hobbyist, the general rule is that if your company pays for you or sponsors you, you are a professional, and if you personally pay you are a student or hobbyist. If in doubt drop us a mail or just register as hobbyist. The important thing is that you'll make it to the conference!

Speakers who do a longer presentation will receive a discount code, so if you have submitted a talk or plan to submit a talk you may want to wait with registration until a decision has been made as to whether your proposal has been accepted.

Sponsors will receive a number of free registrations as part of their sponsorship package.

Call for Papers

The call for papers is now closed and talk proposals can no longer be submitted. Anyone who has submitted a talk proposal should have received an e-mail letting them know whether their proposal has been accepted.

Lightning talks may still be proposed.

Co-hosted conferences

There are a few other events taking place alongside or just before the GStreamer Conference on October 8-9 this year, so if you are making your way to Dublin you might want to consider combining the GStreamer Conference with some of the other events.

Embedded Linux Conference Europe (ELCE)	October 5-7, 2015
LinuxCon Europe	October 5-7, 2015
CloudOpen Europe	October 5-7, 2015

Sponsors

The GStreamer Conference 2015 is only possible thanks to the financial support from our sponsors. If your company would like to become a sponsor please refer to our sponsorship information package.

City

This year's conference will take place in Dublin, Ireland.

Getting there

Dublin airport (DUB) is Ireland's main airport with good regional and international connections.

Venue

The conference will take place in Liffey Meeting Room 2 (2a and 2b) at The Convention Centre Dublin, which is located in Dublin's Docklands district not too far from the city centre.

The venue is just around the corner of the LUAS (light rail) Red Line *Spencer Dock* stop. It is also within half an hour walking distance or 10-minute taxi ride of most city centre locations and served by various bus routes.

Hotels

If you have not booked your accommodation yet, it is advisable to do this as soon as possible.

The Linux Foundation had some hotel recommendations in the vicinity of the convention centre, all of which are sold out now, so you may have to look at places in other areas.

Most hotels in the city centre will be within walking distance (15-30mins) of the conference venue, or just a short taxi ride away. There is also public transport in form of busses and the LUAS light rail system. The convention centre is right next to the LUAS *Spencer Dock* stop which is on the Red Line.

Temple Bar is the name of the cultural/nightlife quarter which is popular with tourists and where also a lot of hotels are.

The following is just a small selection of hotels which may or may not be suitable. Dublin is an international business centre and tourist destination and as such has many hotels and hostels in all price categories. The below is merely a small selection, you can find more using the usual booking engines such as booking.com, lastminute.com or airbnb.com.

These hotels are located within half an hour of the convention center and had at least a couple of rooms left at the time of writing, but there are many other hotels in the city which may be even better located or provide better value:

- The Fleet Street Hotel (Temple Bar)
- Maldron Hotel Cardiff Lane
- Sandymount Hotel
- Ballsbridge Hotel
- Temple Bar Inn
- Georg Frederic Handel Hotel, Temple Bar
- For those on a budget airbnb.ie may have some suitable accommodation choices

If you find any other places that might be worth mentioning here, please send us a mail.

Contact

If you have any questions about the GStreamer Conference or are considering becoming a sponsor, please contact Tim-Philipp Müller.

Conference News and Updates

2015-September-28: the full conference schedule is now available as well

2015-September-28: The social event on Thursday 8 Oct in the evening is now confirmed to be at **Toners Pub** from 19.00-23.00h. Food and drink will be provided. Bring your badge.

2015-September-23: added some hotels which still have rooms available

2015-September-23: We're happy to to have **Linux Weekly News** again as Press Partner at this year's conference.

2015-September-22: The GStreamer project is thrilled to welcome back **Collabora** as a Silver sponsor this year.

2015-September-14: The GStreamer project is happy to welcome back **Igalia** as a Bronze sponsor this year.

2015-September-09: Talk abstracts and speaker bios published

2015-September-03: The GStreamer project is pleased to welcome **Pexip** as a Bronze sponsor.

2015-August-25: List of talks available

2015-August-05: The GStreamer project is delighted to welcome back **Google** as Silver sponsor

2015-August-01: **UbiCast** is welcomed back as special media partner who will again record talks and lightning talks

2015-July-31: Sponsorship brief available

2015-July-17: Registration is open

2015-July-17: Talk submission deadline is Sunday 9 August 2015

2015-July-02: Call for Papers

2015-June-12: Date and venue confirmed, talks may be submitted.

Past GStreamer Conferences

- 16-17 October 2014 · Congress Centre Düsseldorf · Düsseldorf, Germany
- 22-23 October 2013 · Edinburgh International Conference Centre · Edinburgh, Scotland, UK
- 27-28 August 2012 · Sheraton San Diego Hotel & Marina · San Diego, USA
- 24-25 October 2011 · Clarion Congress Hotel · Prague, Czech Republic
- 26 October 2010 · University Arms Hotel · Cambridge, UK

Report a problem on this page.

https://web.archive.org/web/20161018073419/https://gstreamer.freedesktop.org/conference/2015/schedule.html

INTERNET ARCHIVE Wayback Machine https://gstreamer.freedesktop.org/conference/2015/schedule.html Go SEP OCT DEC 6 captures 18 Oct 2016 - 8 Sep 2024 2015 18 2016 2022 About this capture

gstreamer
open source multimedia framework

GStreamer Conference 2015 Schedule

Dublin, Ireland, 8-9 October 2015

Back to conference main page

Schedule for Thursday 8 October 2015

Schedule for Friday 9 October 2015

Thursday 8th of October 2015			
	Liffey Meeting Room 2a	Liffey Meeting Room 2b	
08.30	**Registration and Breakfast Buffet**		08.30
09.00-09.10	*Opening and welcome to the 2015 GStreamer Conference* Liffey Meeting Room 2		09.00-09.10
09.10-09.40	*Keynote* GStreamer State of the Union Tim-Philipp Müller, Centricular Liffey Meeting Room 2		09.10-09.40
09.40-10.00	*mini break*		09.40-10.00
10.00-10.30	Tiled Streaming of UHD video in real-time Arjen Veenhuizen, TNO	Kurento Media Server: experiences bringing GStreamer capabilities to WWW developers José Antonio Santos, Kurento	10.00-10.30
10.35-10.55	GStreamer-VAAPI: Hardware-accelerated encoding and decoding on Intel hardware Víctor M. Jáquez L., Igalia	ogv.js: bringing open codecs to Safari and IE with emscripten Brion Vibber, Wikimedia	10.35-10.55
10.55-11.20	**Coffee Break**		10.55-11.20
11.20-12.05	Decodebin3, or dealing with modern playback use-cases Edward Hervey (bilboed), Centricular	Chromium: a new media Backend based on GStreamer Julien Isorce, Samsung	11.20-12.05
12.10-12.30	GstPlayer - A simple cross-platform API for all your media playback needs Sebastian Dröge (slomo), Centricular	GStreamer and WebKit Philippe Normand (philn), Igalia	12.10-12.30
12.30-13.30	**Lunch Break: Buffet Lunch for GStreamer Conference attendees**		12.30-13.30
13.30-14.00	Going Live with ToolsOnAir's GStreamer-based broadcast mixing architecture Heinrich Fink, ToolsOnAir	The HeliosTv Distributed DVB stack Romain Picard, Soft@Home	13.30-14.00
14.05-14.25	How to Contribute to GStreamer Luis de Bethencourt (luisbg), Samsung	Stereoscopic 3D Video in GStreamer Redux Jan Schmidt (thaytan), Centricular	14.05-14.25
14.25-14.45	*mini break*		14.25-14.45
14.45-15.15	Faster transcoding with GStreamer in the cloud Thiago Sousa Santos, Samsung	Colorspaces and HDMI Hans Verkuil, Cisco	14.45-15.15
15.15-15.40	**Coffee Break**		15.15-15.40
15.40-16.10	Demystifying the allocation query Nicolas Dufresne (stormer), Collabora	Video Filters and their Applications Sanjay Narasimha Murthy, Samsung	15.40-16.10
16.15-16.45	Camera Sharing and Sandboxing with Pinos Wim Taymans (wtay), RedHat	Hardware-accelerated multimedia on TI's Jacinto 6 SoC Pooja Prajod, Texas Instruments	16.15-16.45
16.45-17.05	*mini break*		16.45-17.05
17.05-18.15	*Lightning Talks - part I* Liffey Meeting Room 2		17.05-18.15
18.15	**End of Day**		18.15
19.00-23.00	**Social Event: Drinks and Food** at the **Toners Pub**, 139 Lower Baggot Street, Dublin 2, D02 N231 *Between St Stephen's Green and Merrion Square* (map) Both drinks and food will be provided, incl. vegetarian options. Please don't forget to bring your badge.		19.00-23.00

Pinterest Ex. 1010, Page 12 of 57

Friday 9th of October 2015			
	Liffey Meeting Room 2a	Liffey Meeting Room 2b	
09.00-10.00	**Breakfast Buffet** (note that we start a bit later on Friday, at 10am)		09.00-10.00
10.00-10.30	OpenGL Desktop/ES for the GStreamer pipeline Matthew Waters (ystreet00), Centricular	Robust lipsync error detection using GStreamer and QR Codes Florent Thiery, Ubicast	**10.00**-10.30
10.35-11.05	Bin It! AKA, How to use bins and bin subclasses to keep state local and easily manage dynamic pipelines Vivia Nikolaidou, ToolsOnAir	Implementing a WebRTC endpoint in GStreamer: challenges, problems and perspectives Luis López Fernández	10.35-11.05
11.05-11.30	**Coffee Break**		11.05-11.30
11.30-12.15	Daala, NETVC, and the Alliance for Open Media: the plan for royalty free video Thomas Daede, Mozilla	Synchronised multi-room media playback and distributed live media processing and mixing with GStreamer Sebastian Dröge (slomo), Centricular	11.30-12.15
12.15-12.30	*mini break*		12.15-12.30
12.30-13.00	Bringing GStreamer to Radio Broadcasting Marcin Lewandowski, RadioKit	Improving GStreamer performance on large pipelines: from profiling to optimization Miguel París Díaz, Kurento	12.30-13.00
13.00-14.30	**Lunch Break: Buffet Lunch for GStreamer Conference attendees**		13.00-14.30
14.30-15.00	*Lightning Talks - part II* Liffey Meeting Room 2		14.30-15.00
15.00-15.30	**Coffee Break**		15.00-15.30
15.30-16.15	Interactive video playback and capture in the Processing Language via GStreamer Andres Colubri, processing.org	Distributed Acoustic Triangulation Jan Schmidt (thaytan), Centricular	15.30-16.15
16.15-16.30	*mini break*		16.15-16.30
16.30-17.00	Pointing cameras at TVs: when HDMI video-capture is not an option Will Manley, stb-tester.com	Simple techniques of pipeline performance measurements and time profiling of individual elements Kyrylo Polezhaiev	16.30-17.00
17.00	*Closing Session*		17.00
17.15	**End of Conference**		

Report a problem on this page.

https://web.archive.org/web/20160125231357/http://gstreamer.freedesktop.org/conference/2015/talks-and-speakers.html

INTERNET ARCHIVE Wayback Machine http://gstreamer.freedesktop.org/conference/2015/talks-and-speakers.html Go DEC JAN FEB 25 2015 2016 2023 About this capture
12 captures 2 Oct 2015 - 12 Sep 2024

Many applications need the ability to play back audio or video media in one way or another, and somehow integrate this media playback into their user interface. While GStreamer allows us to do that since 15 years now, it was never an easy task to do and especially much harder than on other platforms like Microsoft's MediaFoundation, Apple's AVFoundation and Android's MediaPlayer API. Adding cross-platform, GStreamer based playback to an application usually meant writing tens of hundreds of lines of code.

And this is where GstPlayer comes into play. It provides you with a simple media playback API that hides all the GStreamer details you don't want to worry about and allows you to write a simple "Hello World" playback application in less than 10 lines of code. On the other hand it also provides you with all API a full-features media playback application would need. This includes an API for all the basic playback operations (play, pause, seek, ...), media information, trick modes, audio visualization, subtitles and a simple video embedding API for embedding video into the application's user interface. And if all that is not enough, you still have access to the low-level GStreamer API and can customize everything.

In this talk I will introduce GstPlayer and give an overview of its API, the already existing features and the future plans, including integration into GStreamer. This will be followed by some time for questions and discussions.

If you write an application that needs to do media playback in some way, this talk is for you. Get an idea of GstPlayer and let's discuss if anything is missing to make GstPlayer the solution for your application.

The current status of the code can be found at https://github.com/sdroege/gst-player

Sebastian Dröge is a Free Software developer and one of the GStreamer maintainers and core developers. He has been involved with the project for almost 10 years now. He also contributes to various other Free Software projects, like Debian, GNOME and WebKit. While finishing his degree in computer sciences at the University of Paderborn in Germany, he started working as a contractor for GStreamer and related technologies.

Nowadays Sebastian is working at Centricular, a company providing consultancy services around GStreamer and Free Software in general.

Apart from multimedia related topics, Sebastian has an interest in digital signal processing, programming languages, machine learning, network protocols and distributed systems.

Synchronised multi-room media playback and distributed live media processing and mixing with GStreamer, Sebastian Dröge (slomo), Centricular

Many use cases today require synchronised multimedia handling between several, independent devices, possibly in different rooms. These requirements arise in consumer use cases, like multi-room playback of videos on TVs, mobile devices and other parts of a home entertainment system, to allow the user to switch between rooms without interrupting his multimedia experience. Similar requirements also arise in industrial and professional use cases, for example for building video walls as used for digital signage or control rooms, or for distributed live media processing and mixing in professional media production and editing scenarios.

In this talk we will discuss how the flexibility of the GStreamer multimedia framework allows to implement these use cases, and which features are already provided to make it very simple to develop such applications. We will briefly introduce how data flow handling and synchronisation works in GStreamer. After this we will discuss how various open

The project uses the emscripten cross-compiler to produce JavaScript code from the standard C codec libraries (libtheora, libvorbis, libopus, libvpx), which is then wrapped in a fairly lightweight JavaScript framework to run decoding and send output to a canvas element and Web Audio (or Flash on IE 10/11).

The JavaScript platform carriers a number of interesting challenges: slightly funky compilation, performance bottlenecks, limitations on threading, and a lack of synchronous i/o which some libraries expect.

Brion Vibber is currently the Lead Software Architect for the Wikimedia Foundation and a member of the Mobile Apps development team.

Chromium: a new media Backend based on GStreamer, Julien Isorce, Samsung

The GStreamer backend has to follow chromium's multiprocess and sandboxing architecture. Unlike in webkit based browsers, GStreamer pipelines cannot live in the tab-specific processes (Render processes). Chromium design and rules require a different approach than WebKit-based browsers. We will explain why it cannot live in any of the existing processes (Browser process, Renderer Process, GPU Process).

For the sake of the project this new backend must be as efficient as existing media backends in chromium (ffmpeg, chunk demuxers, HW decoder wrappers). One of the purpose of using GStreamer is also to re-use HW decoder elements it provides (gst-v4l2, gstreamer-vaapi, gst-omx, ...).

We will detail how the Media Process interacts with other processes using chromium IPC api and how GstPlayer is integrated. We will describe the sandbox policies we defined for the Media Process and how it blocks or filters system calls. We will explain how the GStreamer pipelines get data from the web through Browser Process to respect sandbox restrictions. Due to sandbox restrictions we will describe how GstGL can access the chromium GPU process to produces OpenGL textures.

Announcement: Announcing a new GStreamer Backend for Chromium

Julien Isorce works for Samsung R&D in London.

Bringing GStreamer to Radio Broadcasting, Marcin Lewandowski, RadioKit

A history of building a radio broadcasting software based on GStreamer.

From GStreamer 0.8 to 1.4.

From monolithic attempts to microservices.

From plain C to Ruby wrappers.

From no tests to test-driven development.

Short history of many tries and (unfortunately) even more failures.

Marcin Lewandowski is Spiritus Movens at RadioKit. Some call him a one men army, others prefer to call him a jack of all trades. Marcin has been juggling with IT and media for ages and it seems that in his life all roads lead to RadioKit instead of Rome. He's built several radio stations (both internet and FM) from scratch and always missed a tool that is web-based, simple as hell, easy to integrate and designed for age of social networks, not FM transmitters. So he made one.

How to Contribute to GStreamer, Luis de Bethencourt (luisbg), Samsung

Contributing to the GStreamer project is easier than you might think; the community is very welcoming.

The GStreamer community is spread over 6 continents and coming from a multitude of companies and backgrounds. How does this diverse group push together for the common goal of making GStreamer even better release after release? Luis will share who the community is, how it operates, the tools used and more behind-the-scenes information, with the aim to give you a running start to help you join and contribute.

Just like Soylent Green, GStreamer is people.

Luis de Bethencourt is a freedom-loving technocrat, who currently works for Samsung's Open Source Group in London. He has always enjoyed programming and playing around with video, so since he discovered GStreamer 5 years ago he's been hooked. Originally from the Canary Islands, computers felt like a door to the world. Luis saw open source software as the best way to enter the innovative technology community, see how it all works, and become a part of it. He enjoys being in front of the screen, behind the screen, Friday beers, Sunday ice-creams, walks in the park, and people who read bios to the end.

Implementing a WebRTC endpoint in GStreamer: challenges, problems and perspectives, Luis López Fernández

Pinterest Ex. 1010, Page 15 of 57

WebRTC is one of the main trends on the multimedia arena in the last few years. The ability of bringing real-time audio and video to WWW browsers opens new horizons for developers to create context aware customized applications for inter-human communications. However, for WebRTC technologies to work seamlessly in WWW applications, it's necessary to manage with a number of present and future complex challenges.

In this talk, we present the experience of the Kurento Media Server team in creating a WebRTC endpoint for GStreamer. We describe the main problems and limitations basing on current GStreamer status describing which parts of WebRTC standards can be currently implemented with GStreamer and which parts require further evolutions and efforts from the community. We will also describe the plans and drafts that are emerging at different standardization groups, including the WebRTC WG at W3C and the RTCWeb WG at IETF. Basing on this, we will try to forecast how WebRTC technologies in particular, but also how real-time multimedia communications in general, may be evolving in the next couple of years and the activities that the GStreamer community should be considering for adapting to these evolutions.

In particular, we shall introduce in detail topics such as the following:

- The evolution of ICE (Interactive Connectivity Establishment)
- Congestion control for RTC streams
- Implementing WebRTC security securely
- Implementing and optimizing the AVPF profile for RTP
- Benchmarking WebRTC: stats metrics
- Managing sensor data through DataChannels

Dr. Luis Lopez is associate professor at Universidad Rey Juan Carlos in Madrid, where he carries out different teaching and research activities in areas related to WWW infrastructures and services. His research interests are concentrated on the creation of advanced multimedia communication technologies and on the conception of Application Programming Interfaces on top of them. The aim of such technologies is to simplify the development of professional real-time communication services satisfying complex and heterogeneous requirements. Dr. Lopez research ideas have generated more than 60 scientific and technical publications and have been included into important research and industrial projects including FI-WARE (http://fi-ware.org) and NUBOMEDIA (http://www.nubomedia.eu). Currently, Dr. Lopez is leading the Kurento.org initiative: an open source software infrastructure providing server-side capabilities for WebRTC with features such as group communications, computer vision, augmented reality, transcoding, mixing and much more.

Improving GStreamer performance on large pipelines: from profiling to optimization, Miguel París Díaz, Kurento

When using GStreamer for creating media middleware and media infrastructures performance becomes critical for achieving the appropriate scalability without degrading end-user QoE. However, GStreamer does not provide off-the-shelf tools for that objective.

In this talk, we present efforts carried out for improving the performance of the Kurento Media Server during the last year. We present our main principle: "you cannot improve what you cannot measure". Developing on it, we introduce different techniques for benchmarking large GStreamer pipelines including callgrind, time profiling, gst-meta profiling, chain-profiling, etc.

We present results for different pipeline configurations and topologies. After that, we introduce some evolutions for GStreamer which could be helpful for optimizing performance such as the pervasive use of buffer-lists, the introduction of thread-pools or the appropriate management of queues.

To conclude, we present some preliminary work carried out in the GStreamer community for implementing such optimization and we discuss their advantages and drawbacks.

Miguel París is software engineer and has a Msc in Telematics Systems. He works as researcher in new multimedia systems and is the manager of real-time communication area in Kurento team, where is the responsible of the WebRtcEndpoint. He has participated in the design of Kurento architecture and APIs and in the development of Kurento GStreamer elements. In addition, he has contributed to GStreamer community with some patches and discussions about RTP stack and other stuff.

Kurento Media Server: experiences bringing GStreamer capabilities to WWW developers, José Antonio Santos, Naevatec/Kurento

GStreamer is a powerful framework, but creating applications using it requires a high degree of expertise, which is not available on the common WWW development community. However, the emergence of technologies such as WebRTC is increasing the interest of media technologies on that community, which opens a relevant opportunity of reinforcing the GStreamer community.

In this talk, we present the experience of Kurento Media Server for bringing GStreamer capabilities to WWW developers. We describe how we have created an abstraction layer on top of GStreamer for making available complex media capabilities through simple and seamless APIs enabling the creation and management of dynamic pipelines. We describe the limitations, problems and challenges we have found in doing so as well as the feedback gathered from WWW developers in relation to the simplicity and appropriateness of GStreamer for solving their needs. Through this feedback, we will analyze the evolutions further capabilities that WWW developers may be demanding in the near future.

To conclude, we present the roadmap of Kurento Media Server for answering to such demands using and extending GStreamer.

José Antonio Santos is a programmer and a software engineer working at Naevatec. He has been working in research, specially in the field of multimedia communications for the last 6 years. As part of this research, the Kurento project was born. Currently he is leading development of Kurento Media Server inside the Kurento project as well as working in integration with client APIs (java and JavaScript).

GStreamer-VAAPI: Hardware-accelerated encoding and decoding on Intel hardware, Víctor M. Jáquez L., Igalia

GStreamer-VAAPI is a set of GStreamer elements (vaapidecode, vaapipostroc, vaapisink, and several encoders) and libgstvapi, a library that wraps libva under a GObject/GStreamer semantics.

This talk will be about VAAPI and its integration with GStreamer. We will show a general overview of VAAPI architecture, the role of libgstvaapi, and finally, the design of GStreamer elements. Afterwards we will show what is ahead in the development of GStreamer-VAAPI, and the current problems and challenges.

Víctor started to work in GStreamer in 2006, on an initial implementation of GStreamer elements wrapping OMX components. Later on, he moved to other related projects such as the WebKit, Ekiga, etcetera. Last year, he returned to the GStreamer arena, helping with gstreamer-vaapi.

Hardware-accelerated multimedia on TI's Jacinto 6 SoC, Pooja Prajod, Texas Instruments

Performance is an inevitable factor for any product. In software world we quantize performance by time and efficient use of memory. To improve the time factor the best solution is to use hardware for computations. TI's J6 SoC has a dedicated IP called IVA-HD for hardware accelerated decode and encode. There is another IP called VPE which takes care of de-interlacing, scaling, colorspace conversion, cropping etc. Efficient use of memory is also interleaved with using hardware of accelerated multimedia. Gstreamer framework gives the flexibility to change the buffer allocation strategy for each plugin. The technique of dmabuf is used to share the buffer memory between hardware and userspace. To improve the performance the allocated buffers can be re-used provided the fd values associated with the buffer is stored as a metadata. The pipelines and buffer flow between plugins depends on the use-case.

Pooja Prajod has been working for Texas Instruments, India Pvt Ltd. since July 2014. Pooja is 24 years old and did her bachelors in Computer Science and Engineering from NIT Calicut, India. She handles GLSDK multimedia components with module ownership of J6 GStreamer deliverables, and before that handled the TI gstreamer plugin migration from 0.10 to 1.2. Pooja loves reading sci-fi books and travelling.

Colorspaces and HDMI, Hans Verkuil, Cisco

Colorspaces are a complex topic and this is even more true once theory meets practice. This talk will go into the practical details of implementing colorspace and colorspace conversion support for HDMI transmitter and receiver drivers. I will demonstrate the effect of the most common colorspace mistakes and look at some of the HDMI limitations with respect to colorspace handling.

Hans Verkuil started contributing patches to the MPEG encoder/decoder ivtv driver in early 2004 and it snowballed from there. Since 2013 he is the video4linux co-maintainer responsible for V4L2 bridge drivers and video receivers and transmitters. He lives in Oslo, Norway, working as a senior R&D software engineer at Cisco Systems Norway, developing - surprise! - video4linux drivers.

Pinterest Ex. 1010, Page 16 of 57

Pointing cameras at TVs: when HDMI video-capture is not an option, Will Manley, stb-tester.com

At stb-tester.com we use image processing to perform automated black-box testing of set-top boxes. This means that we capture video over HDMI and look for UI elements based on screenshots we'd previously captured. This works great for set-top boxes, but what about Smart TVs which don't have an HDMI output?

This talk will describe how we use GStreamer, OpenCV and OpenGL to go from an image captured with a camera back to a nice flat image as if it had been captured over HDMI.

William Manley is software developer and a co-founder of stb-tester.com, a company that provides automated testing equipment and services for set-top box manufacturers. In a previous life he worked on set-top box middle-ware. He has been working with GStreamer since 2012 with a particular interest in v4l and sending video over IPC.

GstPlayer - A simple cross-platform API for all your media playback needs, Sebastian Dröge (slomo), Centricular

Many applications need the ability to play back audio or video media in one way or another, and somehow integrate this media playback into their user interface. While GStreamer allows us to do that since 15 years now, it was never an easy task to do and especially much harder than on other platforms like Microsoft's MediaFoundation, Apple's AVFoundation and Android's MediaPlayer API. Adding cross-platform, GStreamer based playback to an application usually meant writing tens of hundreds of lines of code.

And this is where GstPlayer comes into play. It provides you with a simple media playback API that hides all the GStreamer details you don't want to worry about and allows you to write a simple "Hello World" playback application in less than 10 lines of code. On the other hand it also provides you with all API a full-features media playback application would need. This includes an API for all the basic playback operations (play, pause, seek, ...), media information, trick modes, audio visualization, subtitles and a simple video embedding API for embedding video into the application's user interface. And if all that is not enough, you still have access to the low-level GStreamer API and can customize everything.

In this talk I will introduce GstPlayer and give an overview of its API, the already existing features and the future plans, including integration into GStreamer. This will be followed by some time for questions and discussions.

If you write an application that needs to do media playback in some way, this talk is for you. Get an idea of GstPlayer and let's discuss if anything is missing to make GstPlayer the solution for your application.

The current status of the code can be found at https://github.com/sdroege/gst-player

Sebastian Dröge is a Free Software developer and one of the GStreamer maintainers and core developers. He has been involved with the project for almost 10 years now. He also contributes to various other Free Software projects, like Debian, GNOME and WebKit. While finishing his degree in computer sciences at the University of Paderborn in Germany, he started working as a contractor for GStreamer and related technologies.

Nowadays Sebastian is working at Centricular, a company providing consultancy services around GStreamer and Free Software in general.

Apart from multimedia related topics, Sebastian has an interest in digital signal processing, programming languages, machine learning, network protocols and distributed systems.

Synchronised multi-room media playback and distributed live media processing and mixing with GStreamer, Sebastian Dröge (slomo), Centricular

Many use cases today require synchronised multimedia handling between several, independent devices, possibly in different rooms. These requirements arise in consumer use cases, like multi-room playback of videos on TVs, mobile devices and other parts of a home entertainment system, to allow the user to switch between rooms without interrupting his multimedia experience. Similar requirements also arise in industrial and professional use cases, for example for building video walls as used for digital signage or control rooms, or for distributed live media processing and mixing in professional media production and editing scenarios.

In this talk we will discuss how the flexibility of the GStreamer multimedia framework allows to implement these use cases, and which features are already provided to make it very simple to develop such applications. We will briefly introduce how data flow handling and synchronisation works in GStreamer. After this we will discuss how various open standards like RTP/RTSP and PTP/NTP can be leveraged to implement these use cases, while providing interoperability with other solutions. We will discuss how these are integrated into GStreamer and which challenges exist.

Sebastian Dröge is a Free Software developer and one of the GStreamer maintainers and core developers. He has been involved with the project for almost 10 years now. He also contributes to various other Free Software projects, like Debian, GNOME and WebKit. While finishing his degree in computer sciences at the University of Paderborn in Germany, he started working as a contractor for GStreamer and related technologies.

Nowadays Sebastian is working at Centricular, a company providing consultancy services around GStreamer and Free Software in general.

Apart from multimedia related topics, Sebastian has an interest in digital signal processing, programming languages, machine learning, network protocols and distributed systems.

OpenGL Desktop/ES for the GStreamer pipeline, Matthew Waters (ystreet00), Centricular

OpenGL is a powerful API usually accompanied by dedicated hardware. Equipped with GLSL, one can envisage complex (or simple) filters, mixers, sources and sinks that transform, produce or consume the typical video stream in extraordinary ways.

This talk will provide for an overview on the current integration state of GStreamer + OpenGL and a look into the future of GStreamer with OpenGL.

Matthew Waters has only just started his hopefully long and rewarding FOSS career after using Linux for the past couple of years. When he isn't hacking on GStreamer's OpenGL support, he is attending University and playing around with waveforms.

Robust lipsync error detection using GStreamer and QR Codes, Florent Thiery, Ubicast

Audio to video synchronization (also called lipsync) is a critical aspect of multimedia processing that may be affected anywhere along the media workflow, from capture to delivery. This talk will describe a blackbox-style approach and analysis software based on gstreamer and QR Codes designed to help solve this recurrent problem regardless of the nature of the system.

Florent Thiéry is the C.T.O. and one of the co-founders of UbiCast, a 2007-founded french startup company that builds gstreamer-based solutions designed capture and webcast interactive videos, like the Gstreamer Conference video archive.

Bin It! AKA, How to use bins and bin subclasses to keep state local and easily manage dynamic pipelines, Vivia Nikolaidou, ToolsOnAir

At ToolsOnAir we have dynamic pipelines, where inputs and outputs of varying complexity are added when the main pipeline is PLAYING. In this scenario, state changes and callbacks can be very tricky and can lead to bug-prone spaghetti code. An easy way to keep our code clean, manageable and bug-free™ is by using bins and bin subclasses. These help manage groups of elements that usually go together and perform all the complicated state change black magic for us. As an added bonus, error handling is made much easier. At the end of the day, these advantages greatly outbalance the initial overhead required to write a bin subclass.

Paraskevi Nikolaidou (also known as Vivia) is currently working as a GStreamer developer. She has been active in the Open Source community and has participated in various Free and Open Source projects since 2004 when she joined the Agent Academy project. Vivia has obtained her PhD in Electrical and Computer Engineering from the Aristotle University of Thessaloniki in 20211, where she worked on multi-agent systems as well as data mining methods in supply chain management. Her open source contributions range from SCCORI Agent which was part of her PhD studies, to her contributions to the GStreamer multimedia framework, passing by her involvement with the aMSN project during her spare time. She lives in Thessaloniki, Greece, where she is currently working remotely for ToolsOnAir, a company based in Austria that works with broadcast production software, working on their GStreamer-based platform. She likes ducks, green tea, learning foreign languages and playing the flute.

The HeliosTv Distributed DVB stack, Romain Picard, Soft@Home

HeliosTv is a project whose aim is to allow the implementation of a distributed DVB stack. Distributed in this context means that the different components of a DVB stack can be implemented in different processes and in different devices. The project is composed of a deamon and a client library. The daemon manages the reception of DVB services, and

forwards the TS streams or DVB sections to clients via an IPC or network link.

The client library is used by client applications to communicate with the server. A GStreamer source element based on this library was developed.

This project allows to implement DVB components that are independent from each other and that do not have to manage the frontends : EPG database, live player, live recorder, DSMCC downloader, AIT monitoring can all be implemented independently, and can be executed on different devices in the LAN.

HeliosTv is inspired from the TvHeadend and Sat2Ip projects, and is based on GStreamer, Boost, and MsgPack.

The project is at an early stage where only services are streamed by the daemon. The next step is to add DVB section support.

Romain Picard is software architect at SoftAtHome where he designs and develops software for set-top-box devices. During the last years he was especially involved in GStreamer based software and adaptive streaming technologies.

Video Filters and their Applications, Sanjay Narasimha Murthy, Samsung

Video filtering is not only for noise removal and enhancement but it also has a huge number of applications. Application can range from simple effects which work on player or camera like sketch, paper camera to advanced uses of AR applications, high speed filtering for sports mode, camera effects.

High speed filtering has become very important too these days due to the introduction of High frame rate cameras on mobile phones.

This talk will cover some of the most popular filtering techniques including convolution and some of the practical use cases which are already in use. This talk will also cover the challenges faced in terms of memory requirements and CPU usage and quality of output. Finally we will see the near term possibilities of these filters.

Sanjay currently works for Samsung Research India in Bangalore. He has a masters degree in computer science. He has around 14 years of experience in the field of Multimedia. He has worked on various Samsung Mobile phones which range from SHP, Bada to Android and has implemented several multimedia features and applications for Samsung Mobiles, is currently working on Tizen Multimedia and contributing to GStreamer Open Source. Tizen Multimedia is mostly based on GStreamer.

Tiled Streaming of UHD video in real-time, Arjen Veenhuizen, TNO

Since 2011, Dutch independent contract research organization TNO has been developing a technology for interactive delivery of ultra-high resolution video. This technology, referred to as tiled streaming, enables (mobile) devices to display and freely navigate (using pan-tilt-zoom) ultra-high resolution (4K, 8K, 16K and up) video recordings, while keeping bandwidth requirements low (e.g. 4-6 Mbps). Key asset is that this streaming technology is based on standard protocols and codecs (MPEG DASH/HLS and H.264) and that it potentially scales to millions of simultaneous users in real time.

With tiled streaming, the high resolution panorama is spatially divided into in a number of video regions (“tiles”) at various resolutions and the end-user device only retrieves those tiles that are required to render a specific viewpoint. Display and interaction takes place at the front-end (e.g. a tablet) while the tiles are actually rendered at the back-end.

Processing such ultra-high resolution video streams in high quality (e.g. bandwidths of 1Gbps+ or higher) poses great challenges, especially when the processing must be performed in real-time.

GStreamer is extensively used in the back-end of the tiled streaming system and a (distributed) tiled streaming platform is currently being developed, optimized and tested within the Amsterdam ArenA football stadium in the Netherlands. Our talk will focus on this distributed back-end, its design, challenges, lessors learned and future work.

Arjen Veenhuizen (1985) is a Research Scientist at the Netherlands Organization for Applied Scientific Research (TNO) with a passion for hands-on development of cutting-edge technology in the area of media networking. He excels in rapid prototyping and achieving his goals in a lean and mean fashion and is profound in coordinating and developing larger scale platforms. Furthermore, he values the importance of applying new technologies to the present day. He is lead architect for TNO tiled streaming technology, allowing for real time efficient and scalable delivery of ultra-high resolution video to mobile devices. He has been pioneering multi-source and multi-device synchronisation technology in the European HBB-Next project which has party been standardized in HbbTV 2.0.

Camera Sharing and Sandboxing with Pinos, Wim Taymans (wtay), RedHat

Pinos is a deamon that aims to make it easier to access multimedia content in applications. It is built on top of DBus and GStreamer and uses fd-passing to efficiently pass multimedia around. It's first goal is to provide access to v4l2 cameras and provide the access control checks needed for sandboxing. We will go over the design, show some demos and talk about the future.

Wim Taymans has a computer science degree from the Katholieke Universiteit Leuven, Belgium and decades of software development experience. He co-founded the GStreamer multimedia framework in 1999 and is the person behind much of the current design. Wim Taymans is a Principal Software Engineer at Red Hat, responsible for various multimedia packages and pulseaudio.

Going Live with ToolsOnAir’s GStreamer-based broadcast mixing architecture, Heinrich Fink, ToolsOnAir

Within the last year, we have used GStreamer to develop the architecture of a new live broadcast mixing engine. A single large live pipeline is constantly running. Inputs, outputs and new channel busses can be created and added to the running pipeline at any time. The mixer output and its inputs can be manipulated and monitored remotely by an accompanying OSX mixing application. This application also uses GStreamer to play back streams coming from the engine. Application and engine are both synchronised to a common global PTP clock, using GStreamer's own synchronisation methods.

In this talk we are going to present the overall architecture of this system. We show how bleeding-edge features of GStreamer 1.6 enable a very powerful and extensible mixing pipeline for IP-based broadcast video. Among those are PTP clocking support, RTP/RTSP enhancements, improved use of GPUs, improved OSX support, better decklink elements etc. We'll also show the successful integration of BBC's IP studio system with our GStreamer-based pipeline. This is the result of a recent field-trial of the ICoSOLE EU project that ToolsOnAir and the BBC are part of.

Heinrich Fink is a software engineer at ToolsOnAir. He has a MSc in Visual Computing and is currently working in the R&D team at ToolsOnAir. He has been working as a teaching assistant for several years together with Professor Michael Wimmer at TU Vienna. His work “Teaching a modern graphics pipeline using a shader-based software renderer” was published by the Computers & Graphics Journal. During his master thesis “GPU-based Video Processing in the Context of TV Broadcasting” he implemented the open-source OpenGL benchmarking tool “gl-frame-bender”. At ToolsOnAir, he is leading the development of a new GStreamer-based media engine which is used by future and current products.

Faster transcoding with GStreamer in the cloud, Thiago Sousa Santos, Samsung

Transcoding media files is a common activity when dealing with online content. Users will submit files in different formats which need to be transcoded before distribution as clients on different target devices will have different requirements. Transcoding can be a resource and time consuming task and a possible way to reduce the time needed is to distribute the processing to multiple machines.

This talk will show how GStreamer pipelines were used in a distributed transcoding solution and present some of the results obtained in its experiments when compared with a single computer transcoding.

Thiago started writing multimedia applications from the early days of his Computer Science course. He first used OpenCV and, a bit later, got interested in developing with and in GStreamer itself, and he's been part of the community ever since. Thiago has chosen Open Source to be his career and is now one the founder members of the Samsung OSG Multimedia team. Day to day work includes maintaining GStreamer and bringing internal projects closer to upstream.

Distributed Acoustic Triangulation, Jan Schmidt (thaytan), Centricular

There are already a lot of microphones connected to computers in most houses.

This talk is about using them as an array to do clever things like narrowing down which light a person needs when they say 'turn on the lights'.

Jan Schmidt has been a GStreamer developer and maintainer since 2002. He is responsible for GStreamer's DVD support, and primary author of the Aurena home-area media player. He lives in Albury, Australia and keeps sheep, chickens, ducks and more fruit trees than is sensible. In 2013 he co-founded Centricular - a consulting company for Open Source multimedia and graphics development.

Stereoscopic 3D Video in GStreamer Redux, Jan Schmidt (thaytan), Centricular

Last year, I talked about a plan to implement handling of stereoscopic video in GStreamer. This talk is about how that plan played out.

GStreamer doesn't currently provide any explicit support for stereoscopic content. This talk is about ongoing work to integrate strong 3D and multiview support for GStreamer 1.x - to support technologies like 3D TV that are already widely available, as well as an extensible framework for future systems.

Jan Schmidt has been a GStreamer developer and maintainer since 2002. He is responsible for GStreamer's DVD support, and primary author of the Aurena home-area media player. He lives in Albury, Australia and keeps sheep, chickens, ducks and more fruit trees than is sensible. In 2013 he co-founded Centricular - a new company for Open Source multimedia and graphics development.

Interactive video playback and capture in the Processing Language via GStreamer, Andres Colubri, processing.org

The Processing project is a Java-based open-source programming language and development environment that aims to bridge art, science, and design by turning code into a creative medium for artists and designers, and by promoting visual thinking among scientists and engineers.

These goals often require the use of video in different applications: interactive installations, generative artworks, and data visualizations, to name a few. The GStreamer framework provided the underlying foundation for real-time video playback and capture in Processing for almost a decade since the 0.10 release, and by doing so has allowed creators all around the world to use high-quality, interactive video in their work (e.g. this or this).

In this paper, we would like to show how the development of the java bindings for GStreamer accompanied this and were refined over the years to allow innovative uses of video.

After the 1.0 release of GStreamer, new strategies needed to allow Processing and Java applications in general to continue to GStreamer, and we will describe the ongoing work in order to update the bindings and video library in Processing to use GStreamer 1.x.

Andres is a researcher working on data visualization and interactive graphics. He is an active contributor to the Processing project, a language and programming environment for computational arts and design, and the Java bindings for the GStreamer multimedia framework. He is the main developer of the OpenGL renderer and the GStreamer-based video library in Processing 2 and 3. He originally studied Physics and Mathematics in Argentina and later on did an MFA at the UCLA Design Media Arts program. He uses Processing as the main tool to bridge his interests in computer graphics, visualization, and statistical modeling.

GStreamer State of the Union, Tim-Philipp Müller (__tim), Centricular

This talk will take a bird's eye look at what's been happening in and around GStreamer in the last twelve months and look forward at what's next in the pipeline.

Tim Müller is a GStreamer core developer and maintainer, and backseat release manager. He works for Centricular Ltd, an Open Source consultancy with a focus on GStreamer, cross-platform multimedia and graphics, and embedded systems. Tim lives in Bristol, UK.

Demystifying the allocation query, Nicolas Dufresne (stormer), Collabora

The allocation query is a powerful and yet to be fully understood mechanism of the GStreamer 1.0 framework. Many contributors go through a long series of uncertainties when comes the time to implement such a mechanism for their elements. In this talk, I'd like to better explain how this mechanism works, where are the pitfalls and what solutions can be employed to get around them.

Nicolas is Senior Multimedia Engineer at Collabora, based in Montréal, he was initially a generalist developer, with background in set-top-box development. Nicolas started in 2011 contributing to GStreamer Multimedia Framework, adding infrastructure and primitive to support accelerated upload of buffers to GL textures. His work toward fully Open Source general purpose use of accelerator in GStreamer continues today at Collabora with the recent addition of Video4Linux accelerated decoders and converters support, enabling playback of today's content on Cotton Candy and the HardKernel Odroid U2.

Decodebin3, or dealing with modern playback use-cases, Edward Hervey (bilboed), Centricular

decodebin2 (decodebin in 1.x) is 10 years old. While it did serve its purpose, there are a number of limitations in regards to handling modern use-cases, features that are in playbin that should be present in decodebin, non-optimal memory usage, and so forth. This talk might include apologies from the original decodebin2 author *cough*.

After looking at the current design and limitation of stream handling in general, and decodebin2 in particular, we will look at at proposal for a new way of handling stream listing, stream selection, how this help with having predictable behaviour and drop all 0.10-ism that still remain in decodebin2.

The new design will allow reducing cpu/memory usage by only using the decoders actually needed, re-use decoders when switching streams (where possible), and better deal with adaptive streaming and dynamic MPEG-TS use-cases in general, amongst other things.

Edward Hervey has been contributing for over 12 years to GStreamer, ending up there after starting the PiTiVi video editor and then maintaining various components over the years. After having started Collabora Multimedia in 2007, attempting to go on sabbatical, and doing various freelancing, Edward Hervey is currently a part-time consultant for Centricular.

Daala, NETVC, and the Alliance for Open Media: the plan for royalty free video, Thomas Daede, Mozilla

NETVC is a new working group at the IETF to create a high quality and royalty-free video codec for the Internet. It follows the success of Opus, the audio codec standardized by the IETF as RFC 6716. The working group is receiving contributions from Xiph and Mozilla's Daala codec, as well as the new Thor codec contributed by Cisco.

The Alliance for Open Media is a brand new group formed to do the patent analysis necessary to achieve a royalty free codec. Its founding members represent some of the biggest names in online video, such as Netflix, Amazon, and YouTube, multiple browser vendors including Mozilla, Microsoft, and Google, and key technology providers like Cisco and Intel.

This talk will cover progress made in the last year in the development of Daala, as well as an overview of Thor and other contributions to the NETVC working group. It will also discuss the role of the Alliance and the importance of high quality video that anyone can use without complex and costly licensing.

Thomas Daede is a Video Codec Engineer at Mozilla.

Simple techniques of pipeline performance measurements and time profiling of individual elements, Kyrylo Polezhaiev

In this talk I will present few simple ideas (and tools) for time profiling and data flow inspection between particular elements in GStreamer pipelines. I will underline few aspects that makes common profiling tools usage less effective for GStreamer-based applications and propose few tweaks for GStreamer to make tracing easier.

Kyrylo Polezhaiev is a software engineer and former video game developer, and has an MSc in Systems Engineering. He assembled radio receiver at age eleven, and created a video game (using GStreamer) with 200k+ downloads when he was 21. Kyrylo lives in Kharkiv, Ukraine.

Lightning Talks

Lightning talks are short talks by different speakers about a range of different issues. We have the following talks scheduled so far (in no particular order):

- Industrial application pipelines - *Marianna Smith, Pexip*
- OpenGL

Pinterest Ex. 1010, Page 19 of 57

- Industrial application pipelines · *Marianna Smith Buschle, QTechnology*
- Hyperspectral imagery · *Dimitrios Katsaros, QTechnology*
- gst-gtk-launch-1.0 · *Florent Thiery, Ubicast*
- liborc (JIT SIMD generator) experiments · *Wim Taymans, RedHat*
- V4L2 GStreamer elements update · *Nicolas Dufresne, Collabora*
- Analyzing caps negotiation with GstTracer · *Thiago Sousa Santos, Samsung*
- Know your queues! queue, queue2, multiqueue, netbuffer and all that · *Tim-Philipp Müller*
- Nle: A new design for the GStreamer Non Linear Engine · *Thibault Saunier*
- What is new in GstValidate · *Thibault Saunier*
- Continuous Integration update · *Edward Hervey*
- Remote GStreamer Debugger · *Marcin Kolny*
- gstreamermm C++ wrapper · *Marcin Kolny*
- Multipath RTP (MPRTP) plugin in GStreamer · *Balázs Kreith*
- OpenCV and GStreamer · *Vanessa Chipi*
- Handling Interleaved and Non-Interleaved streams with GStreamer · *Ramesh Venkatachalapathy, LGE*
- Done in 6.0 seconds: a new build system for GStreamer? · *Jussi Pakkanen*
- Gst daemon and interpipes: A simpler way to get your application done
  Melissa Montero, RidgeRun
- Another android camera source for Gstreamer
  Justin Kim, Collabora
- ...
- *Your talk here?*

Lightning talk speakers, please export your slides into a PDF file and either send it to Tim by e-mail (you will receive an e-mail from him about your lightning talk before the event) or have it ready on a usb stick before the start of the lightning talks on Thursday. The idea is that everyone uses the same laptop so that we don't waste time hooking up laptops to the projector and configuring them. There is no particular order or schedule for the talks. When a speaker is called up, we will also mention who is up next. Every speaker has up to ca. 5 minutes for their talk. There will be a countdown timer running, and there will be some music playing towards the end so the speaker knows they have to wrap up. If you don't want to use up the full 5 minutes, that's fine as well. It's not possible to go over time, you'll have to finish up so that everyone has an opportunity to present their talk. We will try to squeeze as many talks into the Thursday slot as possible, and then finish with the rest on Friday.

Report a problem on this page.

EXHIBIT B

https://web.archive.org/web/20170226222859/https://gstreamer.freedesktop.org/data/events/gstreamer-conference/2015/Edward%20Hervey%20-%20decodebin3.pdf

“decodebin3”
Modern playback use-cases

GStreamer Conference 2015, Dublin
October 8th 2015

Edward Hervey a.k.a bilboed

edward@centricular.com





- Playback ?
- Current implementation
- Pitfalls and limitations
- Some solution (no spoilers)

gstreamer

Centricular

What is playback ?

As a user
I want to play/pause/seek a media
Switch streams if available
....

i.e. the gst-player API
(go see slomo's talk just after)





Playback in GStreamer

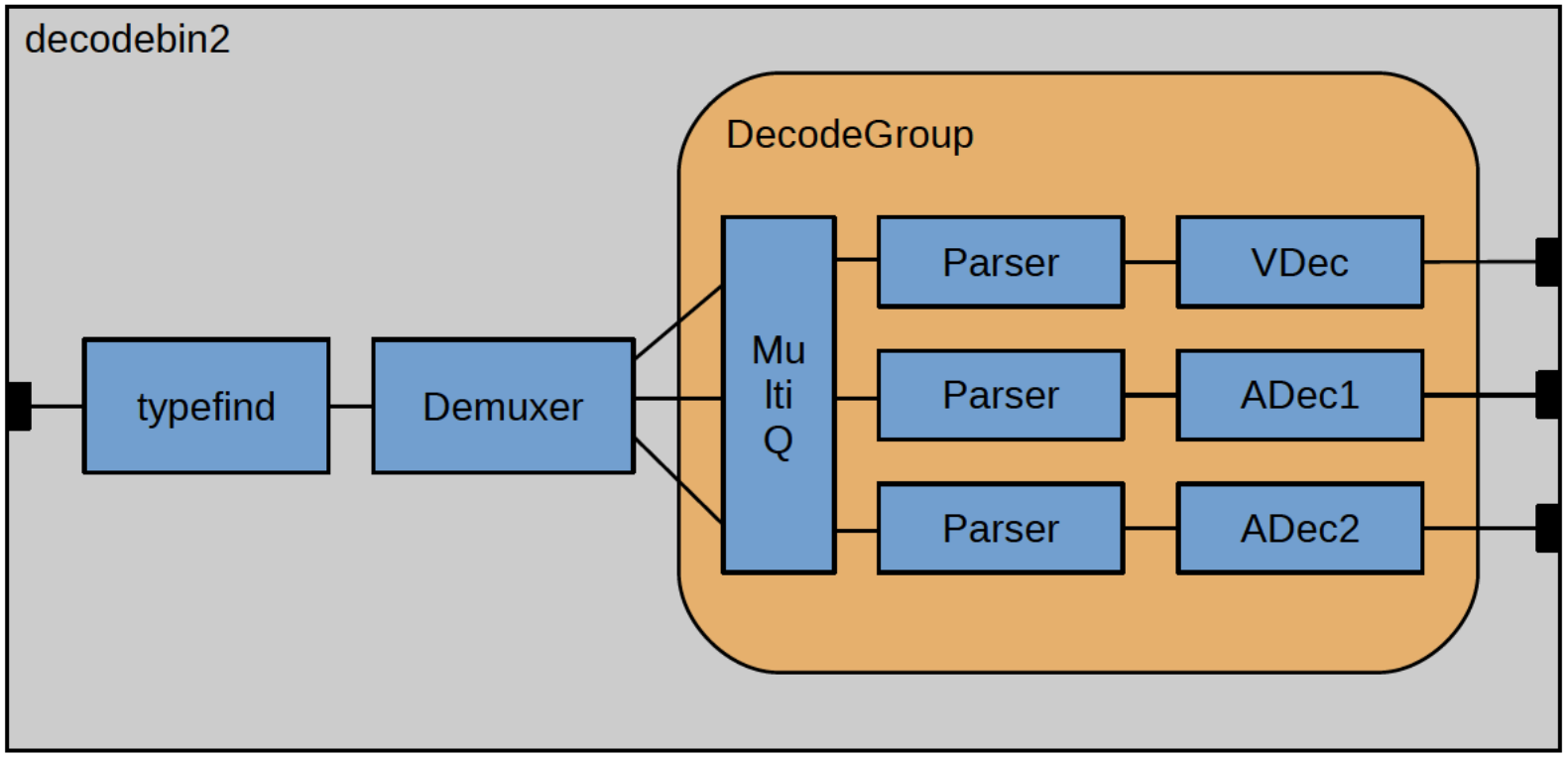
- `playbin` (convenience pipeline)
  - `uridecodebin`
    - `decodebin`
  - `playsink`
  - And some non-reusable code

gstreamer

Centricular

decodebin

- Goal : “Take this input stream, figure out what elements are needed to decode it”
- Actually decodebin2 (2006, 0.10)
- Recursively figure out elements needed
- Support for hardware outputs (assisted auto-plugging)
- Stream switching without data loss
- “chained” files (ex: ogg)



gstreamer

Centricular

Decodebin2: Stream Switching

We might switch streams later on …

We don't want the other stream to be drained completely

    Else we will have a gap when switching

`multiqueue` throttles unlinked streams

But … we're decoding everything :(

    Useless CPU/MEM/IO usage





Decodebin2: Chained File Support

- i.e. dealing with streaming formats (topology can change at any point in time)
- … but only with OGG in mind
- … and brings a bag of issues

gstreamer

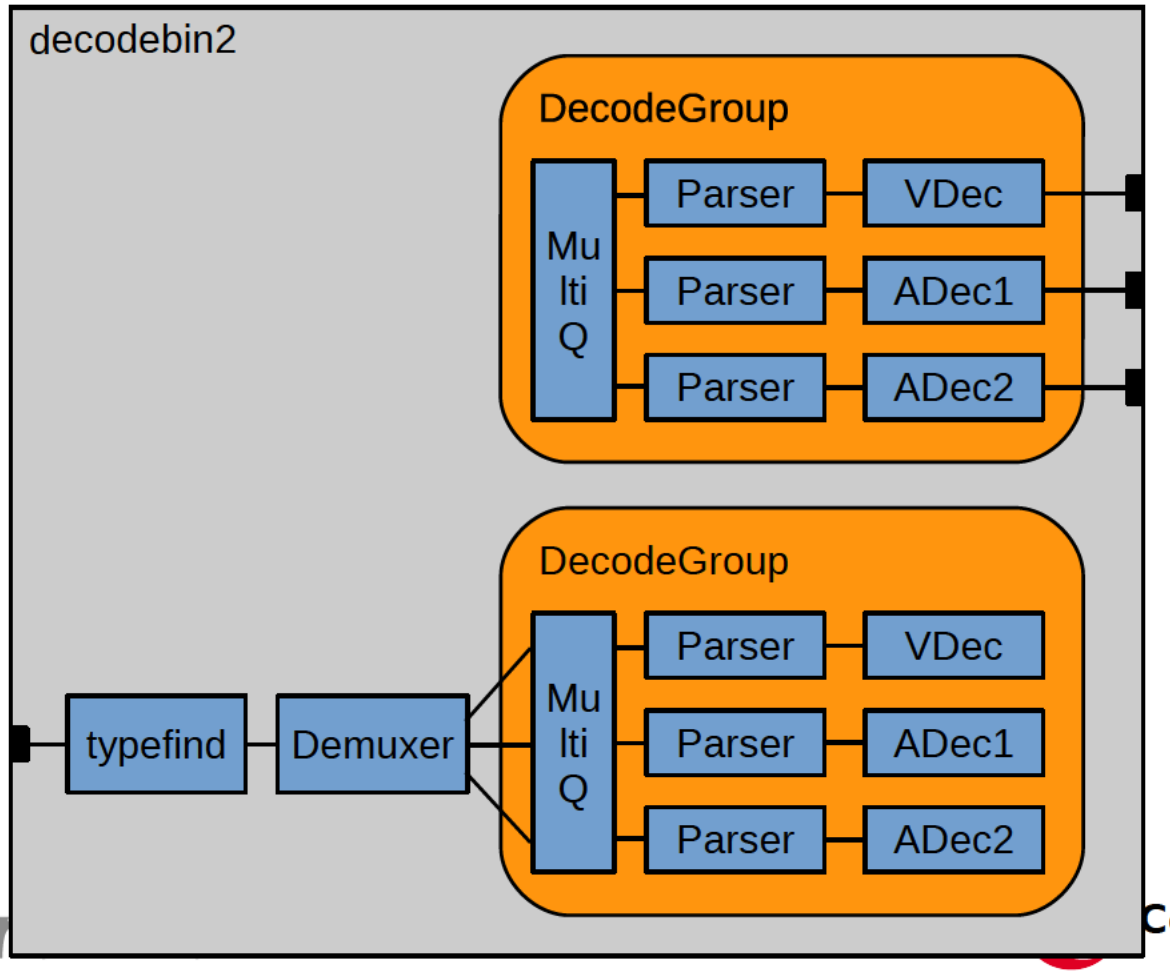
Centricular

Decodebin2: Demuxer expectations

- To switch
  - Emit 'no-more-pads'
  - Add new pads
  - Send EOS on old pads
  - Remove old pads
- Decodebin2 will create a new DecodeGroup
  - Blocks new DecodeGroup
  - Waits until old DecodeGroup is drained (EOS reaching the end of that group)
  - Switches DecodeGroup over (ghostpads point to new group)






decodebin2
DecodeGroup
Mu
lti
Q
Parser
VDec
Parser
ADec1
Parser
ADec2
DecodeGroup
typefind
Demuxer
Mu
lti
Q
Parser
VDec
Parser
ADec1
Parser
ADec2

Chain pitfalls

- New “pending” DecodeGroup
  - Increased memory usage (multiqueue)
  - Increased CPU usage (duplicated elements)
- Input and output of decodebin is no longer fully linked
  - Ex : seek event ending nowhere :(
- Want to just add/remove a stream ?
  - Still need to re-create a new bag of source pads
  - Breaks playback (switch video decoder in GOP)
  - Note: wasn't possible in 0.10 due to SEGMENT limitations





Decodebin2 : more issues :(

- ~~Calculating~~ Guessing the ideal multiqueue size
  - Not too much (keep memory usage down)
  - But enough to cope with interleave
    - Distance in time between co-located buffers
- Hard to do because we're not dealing with timed input in multiqueue
  - Parsers (i.e. guaranteeing timed/chunked data) is after multiqueue

gstreamer

Centricular

uridecodebin

- Gets the proper source element to use for the URI
- If needed, add buffering/queueing (network streams)
  - Ideally this is where “network” buffering/queuing should happen
  - Not always the case (adaptive demuxers in decodebin)
- Plug result into decodebin
  - … or more than one decodebin (ex: RTSP)

PlayBin

- Uses uridecodebin
  - Output of uridecodebin to inputselector(s) (to switch streams)
  - Into playsink
- Allows stream listing/selection
- Extra-uri (for subtitle in separate URI)
  - Creates a new parallel uridecodebin
- Gapless support (about-to-finish / next-uri)
  - Prerolls another uridecodebin and switches

Pitfalls of Stream Selection

- Not a generic API
  - Need to duplicate that code/logic in your custom pipeline
- Expects all streams to switch from to be decoded (and switch happens in inputselector)
- How do we list streams ?
  - Entirely based on source pads from uridecodebin
  - What if there are no pads ? (ex: Alternate HLS/DASH)

A solution to all these issues ?

- A generic API for listing streams...
  - Not tied to pads, Elements can list “hidden” streams
- … and to select streams :
  - Elements need to **know** what downstream wants
  - No longer relying on GST_FLOW_NOT_LINKED
- Reduce CPU, Memory and I/O usage to the minimum needed
- And let's use 1.0 improvements
  - Segment base (adding streams at any time)
  - Better/Saner renegotiation/reconfiguration

gstreamer

Centricular

Generic Stream API

- Listing Streams
  - Could just use stream-id … but not that useful for user
  - Also need type of stream, caps, tags, …
- We first need a more convenient way of dealing with “streams”
  - New GstStream high-level object

GstStream object

- GstObject subclass (refcounted)
  - Id (same as STREAM_START stream-id)
  - GstStreamType
    - GST_STREAM_TYPE_{AUDIO, VIDEO, TEXT, ...}
  - GstCaps
  - GstTagList
- It is just collecting information stored in various places
- In GST_EVENT_STREAM_START
  - Get all the information from one place





```
GstStream *gst_stream_new            (const gchar *stream_id,
                      GstCaps *caps,
                      GstStreamType type,
                      GstStreamFlags flags);

const gchar *gst_stream_get_stream_id (GstStream *stream);

void           gst_stream_set_stream_flags (GstStream *stream, GstStreamFlags flags);
GstStreamFlags gst_stream_get_stream_flags (GstStream *stream);

void          gst_stream_set_stream_type (GstStream *stream, GstStreamType
stream_type);
GstStreamType gst_stream_get_stream_type (GstStream *stream);

void        gst_stream_set_tags (GstStream *stream, GstTagList *tags);
GstTagList *gst_stream_get_tags (GstStream *stream);

void     gst_stream_set_caps (GstStream *stream, GstCaps *caps);
GstCaps *gst_stream_get_caps (GstStream *stream);
```





GstStreamCollection

- A immutable collection of GstStream
  - Usually posted by demuxers or other elements that can offer a “collection” of streams
  - Does not need to have a GstPad associated
- GST_MESSAGE_STREAM_COLLECTION
  - User, application, bin, can be informed of available streams
- Not tied to playbin
  - Got a custom pipeline ? Win

GstStreamCollection

```
GstStreamCollection *gst_stream_collection_new (const gchar *upstream_id);

const gchar *gst_stream_collection_get_upstream_id (GstStreamCollection
*collection);

guint gst_stream_collection_get_size (GstStreamCollection *collection);
GstStream *gst_stream_collection_get_stream (GstStreamCollection *collection,
guint index);

gboolean gst_stream_collection_add_stream (GstStreamCollection *collection,
                        GstStream *stream);
```

Selecting Streams

- GST_EVENT_SELECT_STREAMS
  - List of stream-id to be selected
- Elements can now reliably know **which** streams will be needed downstream
  - Avoid processing (decoding anyone ?)
  - Hidden streams to activate (Alternate HLS/DASH, switching DVB channel, ...)
- Not tied to playbin
  - Custom pipeline ? Win again



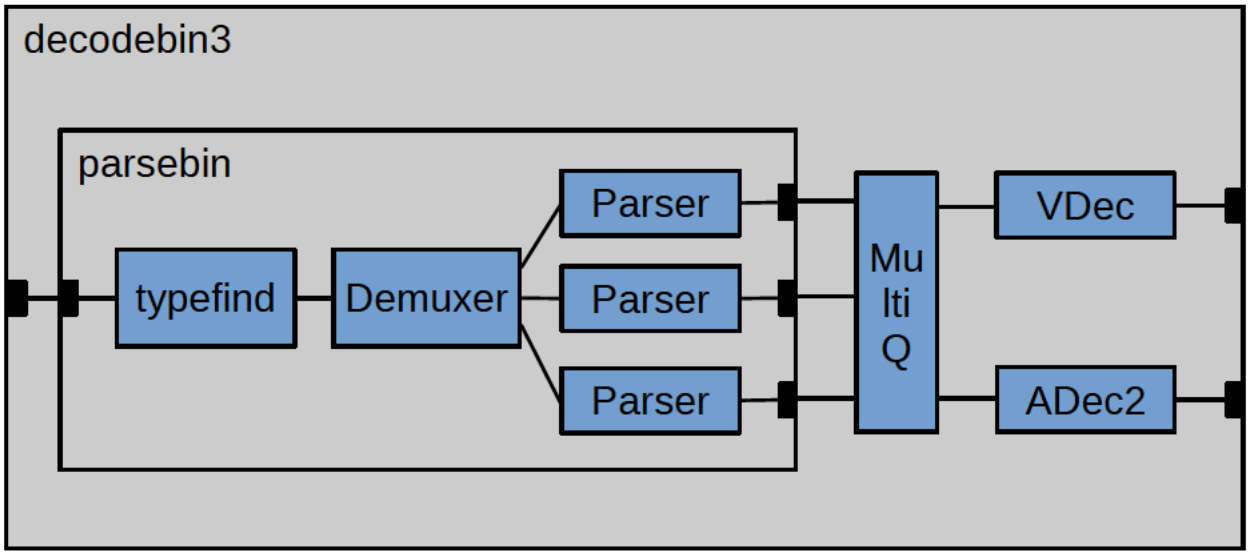


decodebin3

- Because it's been 9 years since I committed decodebin2
- More seriously
  - Use the new stream API to reduce processing as much as possible
  - Re-use as many elements as possible
  - Reduce buffering
  - … and more

gstreamer

Centricular



gstreamer

Centricular

GstParseBin

- One single input sink pad
- Recursively figures out “decodable” elements needed
  - Demuxers, depayloaders, parsers … but not decoders
- No queueing
- Creates/Posts GstStreamCollection/GstStream if the element didn't create it
  - All pipelines can get new API support \o/
- Longer term : Reconfigurable
  - On input changes, re-use elements if possible, else switch





Centricular

decodebin3

- Output of GstParseBin(s) fed to multiqueue
  - Parsed Elementary Streams
  - Multiqueue “slots” (sink + src pad) are typed (audio, video,…)
  - If available slot of proper type, re-use, else create new one
- Selection is done post-multiqueue
  - Only thing that (might) need to be plugged is a decoder
  - By default only expose/decode one stream of each type
  - But you can also expose everything...
  - Or just what you selected via GST_EVENT_SELECT_STREAMS

Selecting Streams (from upstream)

- Startup example
  - 3 streams from GstParseBin : video1, audio1, audio2
  - Do a pre-emptive decision of which streams to use
    - Pending selection : video1, audio1
  - Feed **all** streams through multiqueue
  - On multiqueue output, check STREAM_START / CAPS
    - If in Pending selection : create decoder and expose
    - Else leave unlinked

Switching Streams

- GST_EVENT_SELECT_STREAMS
  - Ex : “video1”, “audio2”
  - Compare available, active, pending and requested streams.
  - Need to switch audio1 to audio2
  - Put audio2 in pending selection
  - Set idle probe on multiqueue output of audio1
    - Unlinks from decoder
    - Send GST_EVENT_RECONFIGURE to audio2 multiqueue source pad
    - Audio2 multiqueue source pad will use same logic as before

Re-using decoders

- STREAM_START / CAPS on multiqueue source pad
  - Checks if in requested / pending selection
  - Check if there is an output available to be re-used
  - Check if decoder can accept CAPS
    - If so, just link
    - If not, unlink, insert new decoder, link
- No pad removed/added on decodebin3 output
  - Just a stream that reconfigures itself
  - Simpler usage in any pipeline





Demuxer handling

- Decodebin3 backwards compatible with current behaviour
- Demuxers (ex: tsdemux) can be smarter
  - Ex: Video stream remains the same, but other streams change
  - No need to remove/add pad for the video stream
    - No breakage in data stream
- Any element can add/remove streams at any time
  - Ex: [CC] parsing in video parsers

Decodebin3 multiple input

- A “media” might be composed of several different input streams
  - RTSP, Pro-cinema (separate files for audio/video), extra subtitle files, …
  - Instead of creating multiple decodebin, just use one
- Decodebin3 has GST_PAD_REQUEST sink pads
  - Feel all streams corresponding to one media
  - Commmon interleave (one multiqueue)
  - One GstStreamCollection (to rule them all)
- No longer dependent on playbin





Gapless playback in decodebin3

- Can we re-use more elements ?
- Just re-use elements for gapless playback
- When EOS reaches all multiqueue slot input, emit 'about-to-finish'
  - Users have a chance to change the input of decodebin3
  - Earlier than with legacy playbin (EOS on output of decodebin)
- GstParseBin reconfigurable
  - Handles the reconfiguration
  - And we just get new streams… as if it was a demuxer update





Playbin and uridecodebin

- Shifted most of the logic to decodebin3 (stream selection, gapless, multi-input support)
- Should not require new elements… but we'll see

Summary

- Avoid as much unneeded processing as possible (CPU, MEM, I/O, …)
- Re-use as many elements as possible
- Still WIP

gstreamer

Centricular

Questions ?



