

A close-up photograph of a person's hand stirring a dark pot on a stove. The hand is holding a wooden spoon, and steam is rising from the pot. The background is blurred, showing a kitchen setting.

# **What's cooking in GStreamer**

FOSDEM, Brussels  
1 February 2014

Tim-Philipp Müller <tim@centricular.com>  
Sebastian Dröge <sebastian@centricular.com>



A close-up photograph of a person's hand stirring a dark pot on a gas stove. The hand is holding a wooden spoon, and steam is rising from the pot. The background is a blurred kitchen setting with a granite countertop.

# Introduction

- who are we ?
- what is GStreamer ?



A close-up photograph of a person's hand stirring a large, dark pot on a stovetop. The hand is holding a long, thin wooden spoon. The background is blurred, showing a kitchen setting with a window and some kitchenware. A semi-transparent dark rectangle is overlaid on the left side of the image, containing white text.

## **What is GStreamer ?**

- set of libraries
- pipeline-based: elements, components, pads, data flow
- plugins, abstract API
- often wrap other libraries (decoders, encoders, filters, etc.)

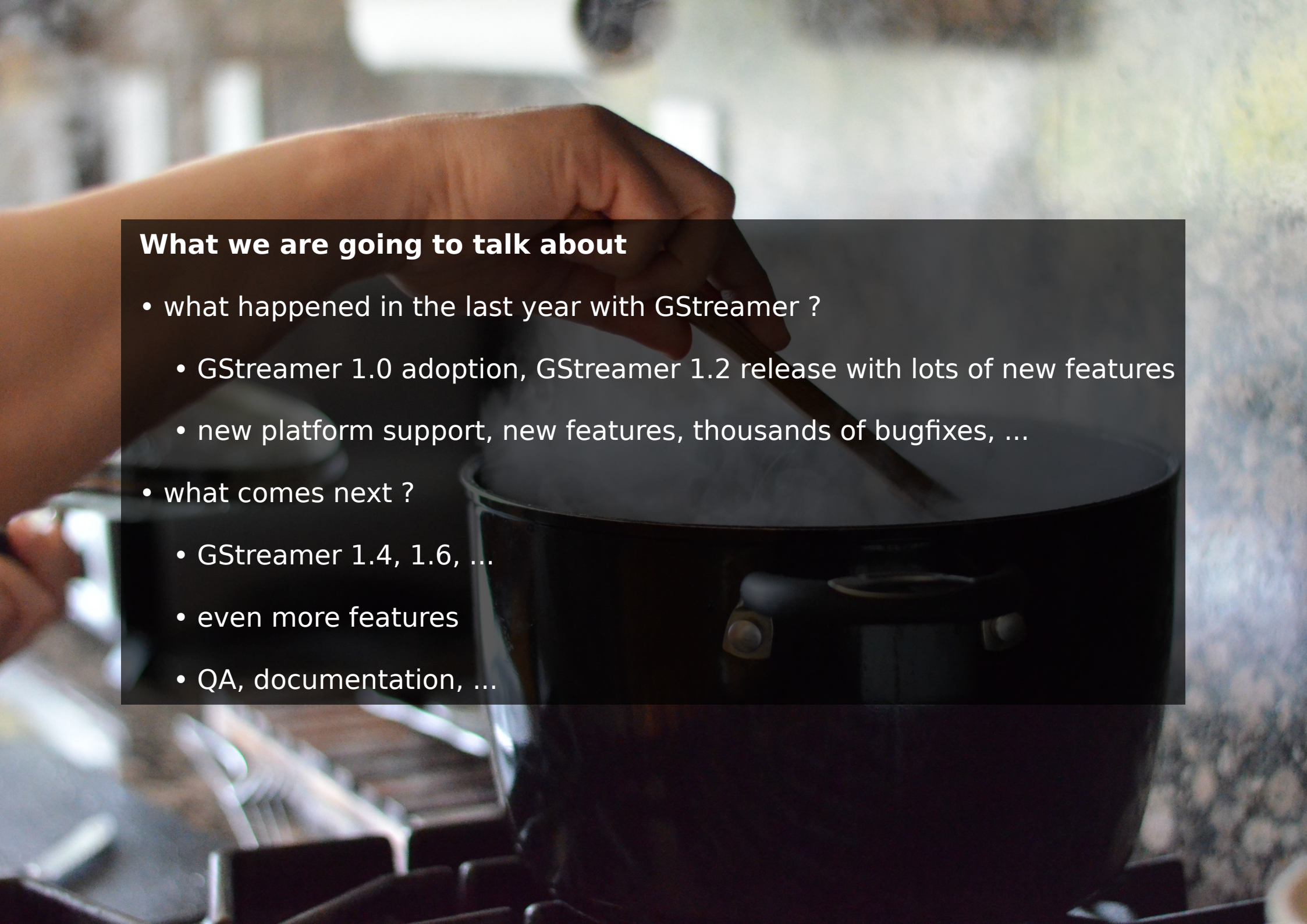


A background image showing a hand stirring a pot on a stove. The hand is holding a wooden spoon and is positioned over a dark pot. The background is slightly blurred, showing a kitchen setting with a window and some kitchen items.

## **What is GStreamer ? (cont'd)**

- low-level API and high-level API
- integration with other frameworks and projects
  - e.g. WebKit, Firefox, Clutter, Windows/OSX/Android/iOS
- goal is to adapt to platform/framework (inputs, outputs, decoders, DSPs/GPUs etc.)



A background image showing a person's hand stirring a large black pot on a gas stove. The pot is filled with a dark liquid, and steam is rising from it. The background is slightly blurred, showing a kitchen setting.

## **What we are going to talk about**

- what happened in the last year with GStreamer ?
  - GStreamer 1.0 adoption, GStreamer 1.2 release with lots of new features
  - new platform support, new features, thousands of bugfixes, ...
- what comes next ?
  - GStreamer 1.4, 1.6, ...
  - even more features
  - QA, documentation, ...



## GStreamer versioning

- 0.10 and before are dead, let's not talk about that
- 1.0 new stable API/ABI, 2.0 next incompatible new API/ABI
- Regular bugfix releases
- GNOME/GLib versioning scheme
- 1.1, 1.3, 1.5, etc. new development release series
- 1.2, 1.4, 1.6, etc. new stable release series
  - All backwards compatible with 1.x



## GStreamer 1.x

- 1.0 finally released in September 2012
- 1.2 released in September 2013
  - 9 + 2 bugfix releases since then
- experience so far: extremely positive, "better than 0.10"
- immediately adopted by GNOME with 3.6
- basically all applications ported by now
- **also:** used by Firefox and Enlightenment too now





## **GStreamer 1.x - what does that mean ?**

- API cleanup, concept generalization, simplification
- evolutionary changes, convergence. no revolution!
- fixes for conceptional problems in 0.10
- lessons learnt
- should provide us an API/ABI that is usable for a long time





## **GStreamer 1.x - what does that mean (cont'd) ?**

- better G-I compatibility (and thus bindings)
- basic concepts stayed the same
- complexity not increased and minimal API changes from app point of view
- new features that were not easily possible before





## Binary releases

- difficult to build manually on Windows, OS X, Android, iOS  
→ we're providing binaries for all stable 1.x releases
- including all plugins and dependencies
- integrating into the platform and IDEs





## **Development on GStreamer 1.4 started**

- started September 2013 and ...
  - ... expected to be released in March/April 2014
- lots of new features and bigger bugfixes





## **New features for hardware integration**

- sharing of hardware contexts in the pipeline
- new implementations and infrastructure for hardware specific memory
- proper negotiation of hardware features and capabilities between elements
- lots of cleanup and fixes for hardware related features
- ex: display server connections, dmabuf/EGLImage, OpenGL, OpenMAX, hardware video codecs, ...





## **New features for hardware integration (cont'd)**

- so what does that mean?
- gst-vaapi will be even faster and integrate more seamlessly and transparently (same for other APIs)
- support for more features of embedded systems
- less workarounds and more flexibility
- stuff just works out of the box!





## Raspberry Pi support & OpenMAX IL

- usage of hardware encoders and decoders
  - gst-omx ported to 1.0 and finally released
- zerocopy decoding via GLESV2 and EGL
- successfully used for HD video display, multi-screen display walls, live streaming servers, ...  
... and everthing in a 25\$ mini computer!



The background of the slide is a close-up, slightly angled view of a stack of antique books. The spines of the books are bound in dark brown leather and feature intricate gold-tooled patterns, including floral motifs, geometric designs, and sunburst emblems. The leather shows signs of age and wear, with some fraying and discoloration visible. The books are stacked horizontally, creating a sense of depth and texture.

## **Other hardware integration**

- gst-omx, gst-vaapi, gst-vidpau
- V4L2 video decoder support
- others slowly coming along, hardware industry is slow



The background of the slide is a close-up, slightly angled view of a stack of antique books. The spines of the books are made of dark brown leather and are decorated with intricate gold-tooled patterns, including floral motifs, geometric designs, and sunburst-like emblems. The leather shows signs of age and wear, with some scuffing and discoloration. The books are stacked horizontally, with their edges visible on the right side of the frame. A semi-transparent dark grey rectangular box is overlaid on the left and center of the image, containing the title and a bulleted list.

## **gst-plugins-gl**

- to replace all the specialized GL hacks
- allows transparent usage of GL filters (shaders, etc) inside pipelines
- rendering to the screen or downloading from the GPU
- multi-threaded
- already runs on all platforms



# Other changes

- Bluez support merged
- HTTP adaptive streaming
- MPEG-TS and DVB
- H265/HEVC and VP9 support





## **Other changes (cont'd)**

- initial Daala support
- RTP/RTSP client/server
- NetClock improvements
- Wayland support
- GNonLin / gst-editing-services and PiTiVi




A close-up, slightly angled view of a stack of antique books. The spines are bound in dark brown leather, heavily decorated with intricate gold-tooled patterns, including floral motifs, geometric designs, and sunburst emblems. The leather shows signs of age and wear, with some scuffing and discoloration. The edges of the pages are visible, appearing aged and slightly yellowed. The lighting is warm, highlighting the texture of the leather and the metallic sheen of the gold tooling.

## Bugs, bugs, bgus

- bugzilla under control
- lots of bugfixing, cleanup and polishing
- many new tests for older features





## **The bright future - 1.4 and beyond**

- Before 1.4: device discovery API still missing
- 1.6 release hopefully 6 months after 1.4
- probably fewer new features, more cleanup, QA, finishing features and polishing
- improved documentation and tutorials



## The Web

- features needed for WebKit, i.e. WebAudio, MediaSource, <video>
- specifying of stream "kinds" (main, alternative, PiP, etc.)
- control over stream selection in playbin (+ allowing mixing, PiP)
- more buffering control
- WebVTT support
- WebRTC support



## 3D video

- done first for GSOC in 2009, now all infrastructure in place to merge it
- conversion  $2D \leftrightarrow 3D$ , red/green, ...
- signalling of different 3D methods (left-right, bottom-top, etc)
- some details to be figured out still



## **More hardware support**

- hopefully getting more native plugins using advanced 1.0 features for efficiency
- getting vendors on board and cooperate with them to provide non-broken plugins
- get things tested more widely, incorporate feedback
- lots of low-hanging fruits: OpenMAX, GL and V4L2 improvements



## Blu-ray

- should get this working finally
- just simple playback, no menus is simple
  - library available from VideoLAN
- menus have insane requirements
  - VM to run menu code
  - being able to decode and composite up to 5 HD streams



A photograph of a red concrete letter 'Q' and a red concrete cube placed on a paved surface with some grass. The 'Q' is on the left, and the cube is on the right. A black rectangular box with the word 'Questions?' in white text is overlaid on the center of the image.

**Questions?**



A close-up photograph of a person's hand holding a wooden spoon, stirring a dark liquid in a pot on a stove. The background is blurred, showing a kitchen setting. A semi-transparent dark rectangle is overlaid on the image, containing white text.

**Thank You!**

**Pictures**

*Cooking* by nicoleabalde

*Old Books* by skittledog

*Road Ahead* by Florian