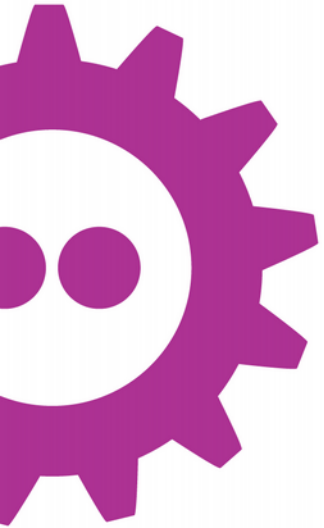




COLLABORA



GStreamer for Tiny Devices



Olivier Crête

olivier.crete@collabora.com

03/02/2018

FOSDEM¹⁸

What did I do ?

- GStreamer at Collabora since 2007
- Started with VVoIP: Telepathy & Farstream
- Helps our customers use GStreamer
 - Many embedded projects

A Tiny Device?

- Flash Storage
- Slow CPU
- Little RAM



Overview of GStreamer

gstreamer tools

gst-inspect
gst-launch
gst-editor

media player

VoIP & video conferencing

streaming server

video editor

(...)

multimedia applications



protocols

- file:
- http:
- rtsp:
- ...

sources

- alsa
- v4l2
- tcp/udp
- ...

formats

- avi
- mp4
- ogg
- ...

codecs

- mp3
- mpeg4
- vorbis
- ...

filters

- converters
- mixers
- effects
- ...

sinks

- alsa
- xvideo
- tcp/udp
- ...

3rd party plugins

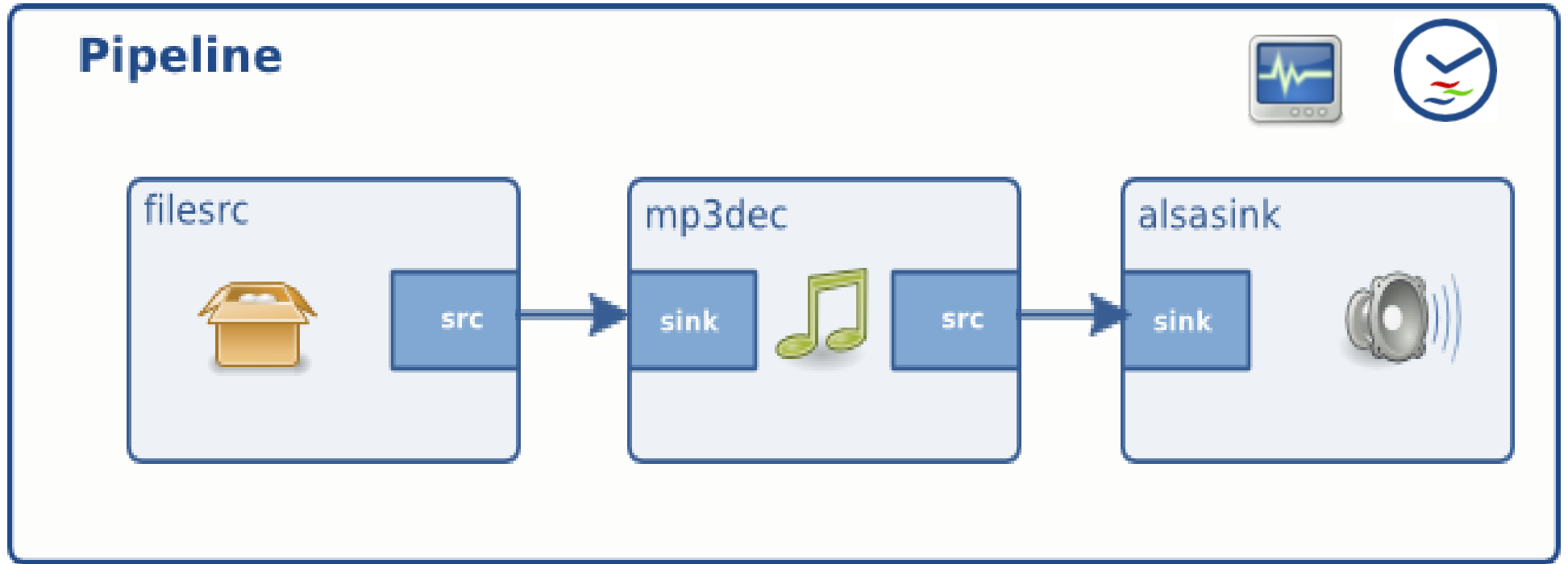
gstreamer plugins

gstreamer includes over 250 plugins

3rd party plugins

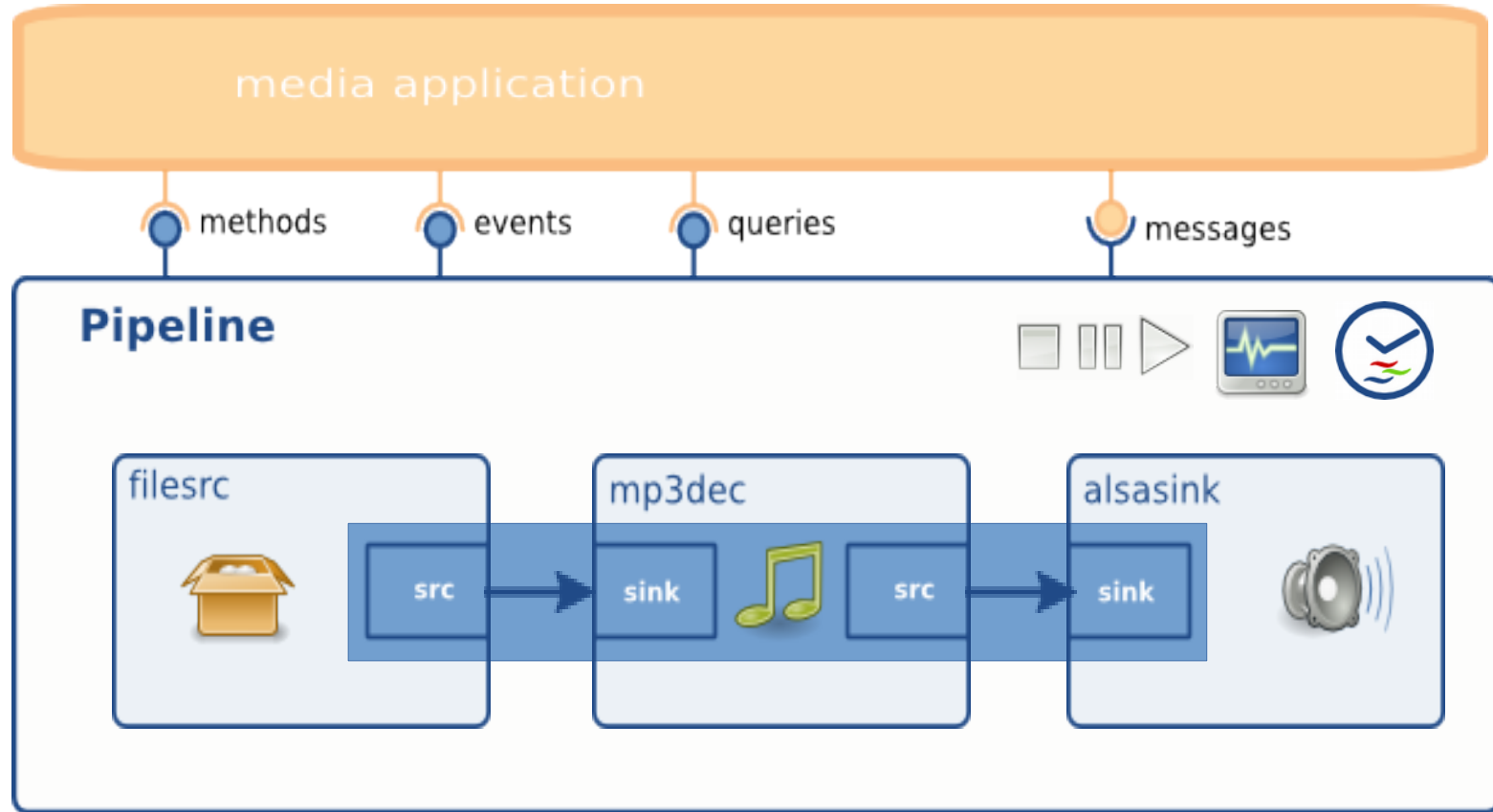


Pipelines!



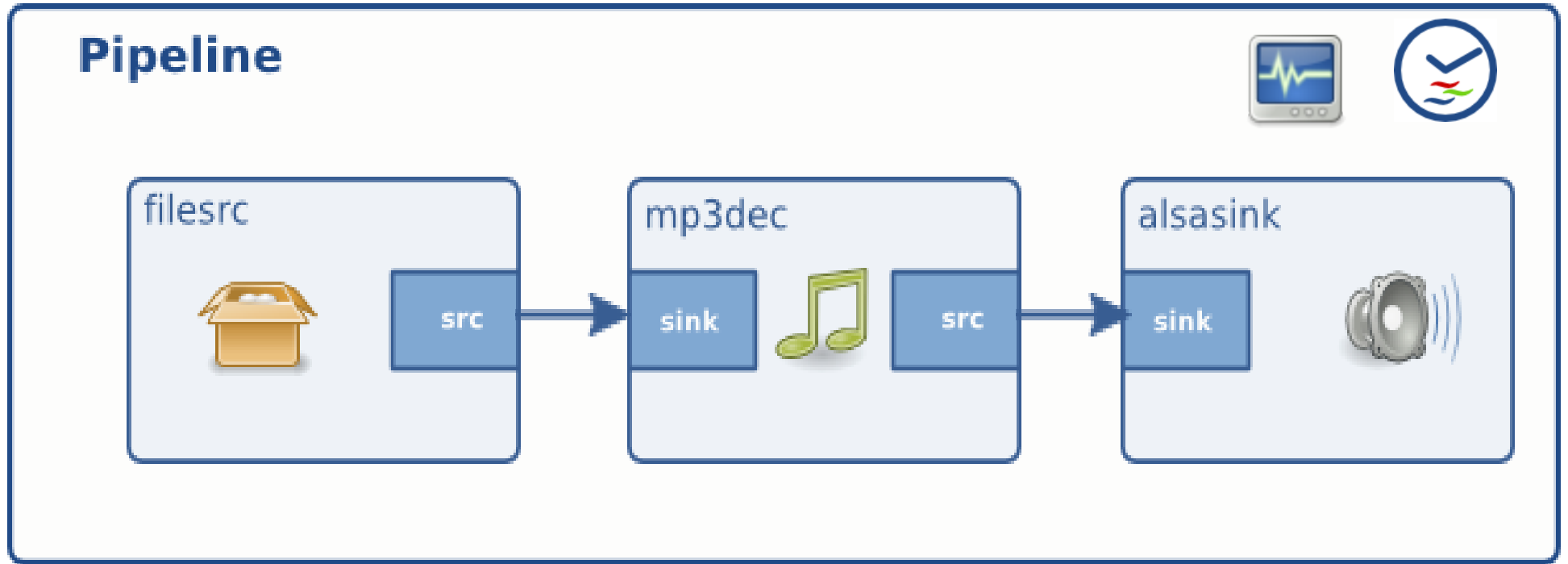


GStreamer in Applications





Pipelines!



GStreamer: Perfect for embedded!

- Complete Zero-Copy toolkit
 - Full negotiation
 - Buffer lifetime
 - Synchronization
- Loads and loads of hardware enabled plugins
- Very fast prototyping

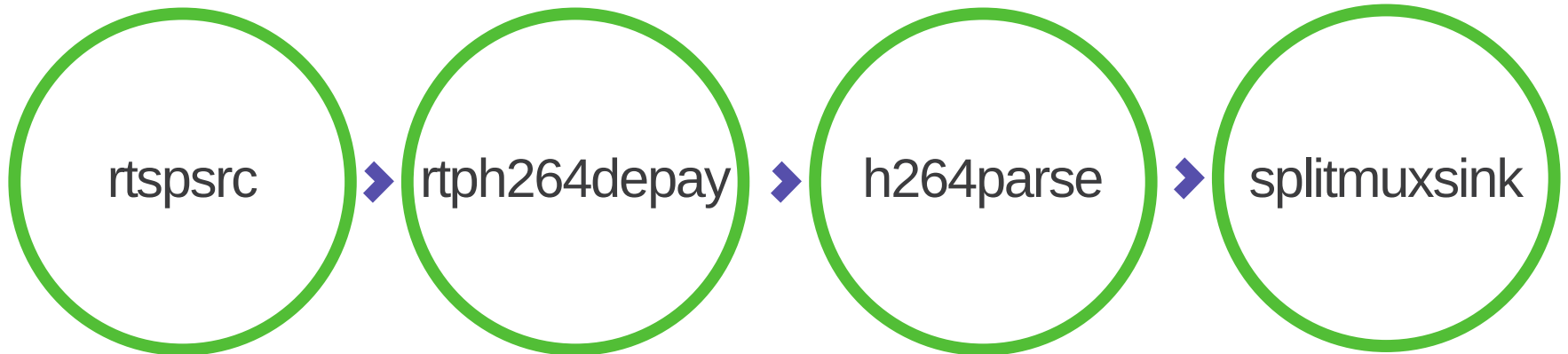
Example: A Security Camera

- 16 MB Flash
- 1 GB RAM
- RTSP Server
- ARMv7



Example: A RTSP to MP4 file segments

- Captures 2s clips from RTSP server





Example: A RTSP to MP4 file segments

- Prototype on PC:

```
gst-launch-1.0 rtspsrc location=rtsp://... !  
rtph264depay ! h264parse ! splitmuxsink  
location=file_template_%05d.mp4
```



Version 0: A shell script with `gst-launch-1.0`

- Complete GStreamer build
 - 287 MB for total build
 - 82 MB of dynamic libraries

▶ **287 MB**



Version 0.1: A shell script with `gst-launch-1.0`

- Strip it
 - 17 MB of dynamic libraries

➤ **17 MB**



Version 1: A C program

```
#include <gst/gst.h>
#include <unistd.h>

int main (int argc, char **argv)
{
    GstElement *pipeline;
    GError *error = NULL;

    gst_init (&argc, &argv);

    pipeline = gst_parse_launch ("rtspsrc location=rtsp://127.0.0.1:8554/test ! rtpH264Depay !
h264parse ! splitmuxsink max-size-time=2 location=video%05d.mp4", &error);
    if (!pipeline)
        g_error ("Error: %s", error->message);

    gst_element_set_state(pipeline, GST_STATE_PLAYING);
    pause();
    gst_element_set_state(pipeline, GST_STATE_NULL);

    return 0;
}
```



Version 1: A C program (Makefile)

```
SYROOT=/home/ocrete/collabora/cerberero/toolchain/sysroot-glibc-linaro-2.25-2017.08-arm-linux-gnueabihf
```

```
CC=/home/ocrete/collabora/cerberero/toolchain/gcc-linaro-7.1.1-2017.08-i686_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-gcc
```

```
PKG_CONFIG_PATH=/home/ocrete/collabora/cerberero/build/dist/linux_armv7/lib/pkgconfig
```

```
program: program.c Makefile
```

```
libtool link --tag=CC $(CC) `PKG_CONFIG_PATH=$(PKG_CONFIG_PATH) pkg-config --libs --cflags gstreamer-1.0` program.c -o program
```

Version 1: A C program

- Binary: 13K
 - Stripped: 5.5K
- Plus 17 MB in libraries

- Total: 17 MB



17 MB

Static Build?

- Using libtool
 - `libtool -static-libtool-libs`
 - `pkg-config --static`
- 7.5 MB binary
- Stripped: 1.5 MB static binary



1.5 MB

Try on device

- Put on Flash, run it:

```
  ** (program:14806): ERROR **: Error: no element  
  "rtspsrc"
```

- Missing all the plugins!

Copy plugins

- All ? 17MB
- Only relevant?!
 - `GST_DEBUG=GST_PLUGIN_LOADING:4`
- Total 1.7 MB in plugins once stripped

➤ **1.7 MB**



Try on device again

```
(program:23211): GStreamer-WARNING **: Failed to load plugin '/lib/gstreamer-1.0/libgstudp.so': libgstnet-1.0.so.0: cannot open shared object file: No such file or directory
```

- Plugins are not static!
 - Need to re-add libraries
 - Back to 17 MB



Statically build plugins

- Declare functions like

```
GST_PLUGIN_STATIC_DECLARE (coreelements);
```
- Register plugins
 - After `gst_init()`
 - `GST_PLUGIN_STATIC_REGISTER (coreelements);`

Statically build plugins

- Link with

```
-L/home/ocrete/collabora/cerbero/build/dist/linux_armv7/lib/gstreamer-1.0 -lgstcoreelements  
-lgstisomp4 -lgstvvideoparsersbad -lgstrtp  
-lgstrtsp -lgstrtpmanager -lgstudp
```

Statically build plugins

- Real static binary is 28 MB
- Stripped to 4.7 MB

- Really working on device



4.7 MB

Ask the compiler for HEEEELLLLPP !

- Compile with -Os
- Disappointing, it hasn't improved!
 - Stripped Binary: 4.7 MB



4.7 MB



Strip functions that are not used

- Build with:
 - ffunction-sections -fdata-sections
- Link with:
 - Wl,--gc-sections
- Stripped Binary : 4 MB



Dig into!

- Find which .o files contribute to the sink
- Used
 - objdump
 - Python script
- Tool: Bloaty McBloatface
 - Wish I had known it

Butcher GLib

- GLib has internal plugins
- Always registered
- Not used, but not garbage collected
- Butcher them out

Butcher GLib

- Removed
 - GSettings
 - GDBus
 - AppInfo
 - Gapplication
 - Notifications

<https://people.collabora.com/~tester/Butcher-glib-remove-gdbus-gapplication-desktopap.patch>

Butcher GLib

- Stripped binary down to 3.8 MB



3.8 MB

Cheating time: Compressed binary

- Using upx

```
upx --best program
```

- Compress down to 2 MB



2 MB

More steps

- Remove UTF-8 tables in GLib
- Dig into exact linked code

Conclusion

- GStreamer is not just for the desktop!
- Not just for larger embedded devices
- Even for your smaller project!



COLLABORA

FOSDEM¹⁸

Any questions?

We're hiring
<http://col.la/careers/>

