COLLABORA

The Art of Debugging GStreamer Software

@ndufresne
Nicolas Dufresne
nicolas@collabora.com

Open First

COLLABORA

Who am I ?

Open Source Linux Developer

- Develops **CODECs** drivers and API in **Linux Media**
- All around contributor to **GStreamer** (with a Linux focus)
- Helps with the GStreamer project **maintenance**
- Likes to help GStreamer users (discourse.gstreamer.org)
- Consultant at **Collabora** since **2009**

COLLABORA

Another journey as Embedded OSS Engineering Consultant

COLLABORA

This time we have a bug!

- The display goes solid color while seeking 4K videos
- Also, sometimes the system completely hangs instead
- A reboot is required to restore the display functionality

Fully OSS Setup

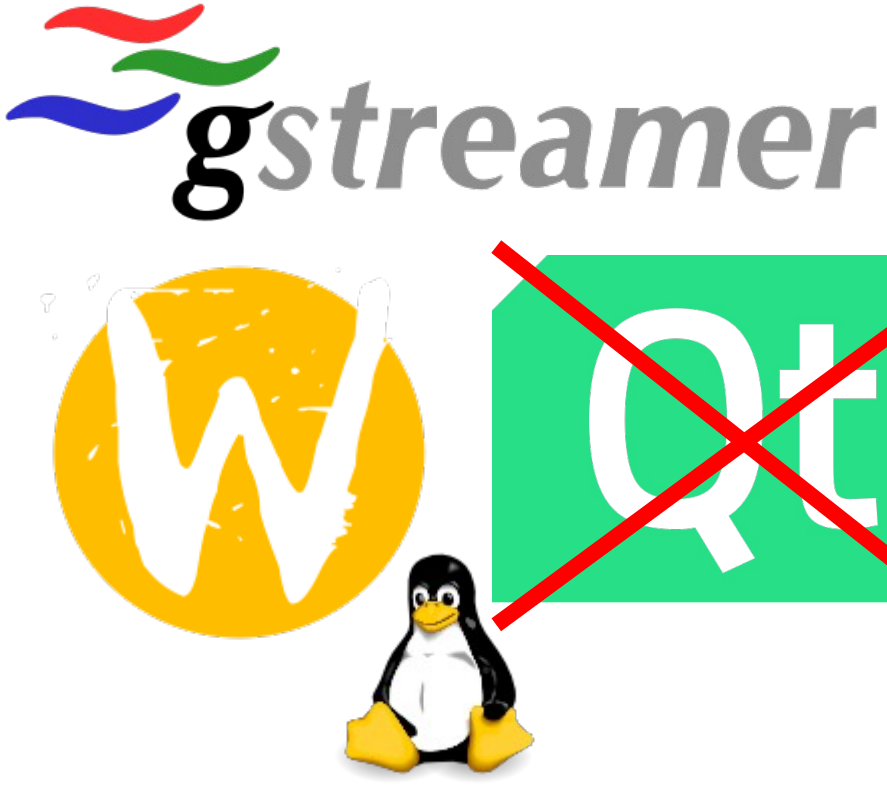
COLLABORA

Can you reproduce ?

COLLABORA

Reduce your scope

Can we remove Qt ?

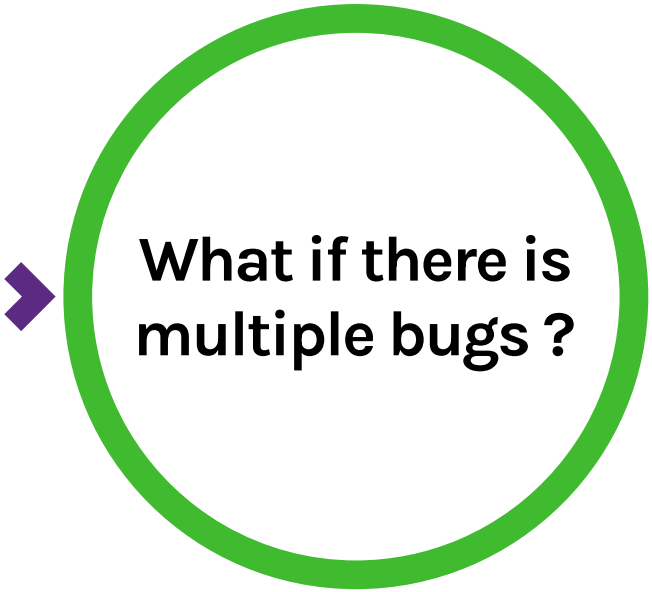



```
gst-play-1.0 –videosink=waylandsink some.mp4
```

- **Left/Right** arrows let you seek
- Perfect!

COLLABORA

Notice more!



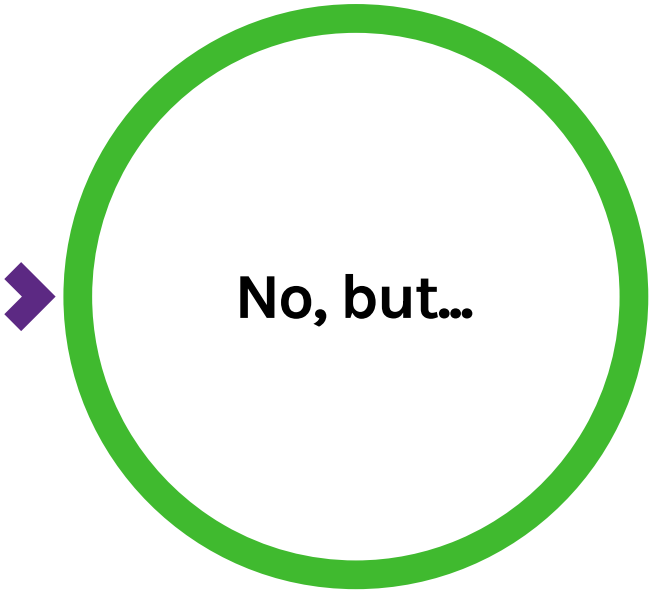
“Also, sometimes the system completely hangs instead”

Can we remove Weston ?


```
gst-play-1.0 --videosink=fakevideosink some.mp4
```

COLLABORA

Wait ! It’s not a GStreamer hanging ?



- My GStreamer **skills** were needed
- Now **easy** to reproduce
- **I can delegate to the kernel devs !**

COLLABORA

Damn it, its me the kernel dev!

Free kernel hang tips!

- **echo ‘w’** > /proc/sysrq-trigger

  [4139.676838] sysrq: Show Blocked State

  ...

- picocom: **ctrl+a ctrl+\ w**

COLLABORA

Improve error handling / reporting

- Trace all **error bits** in the ISR
  - Getting “bus errors” on failure
- Only **clear** the bits that **needs to**
- Do not assume the HW **have stopped**

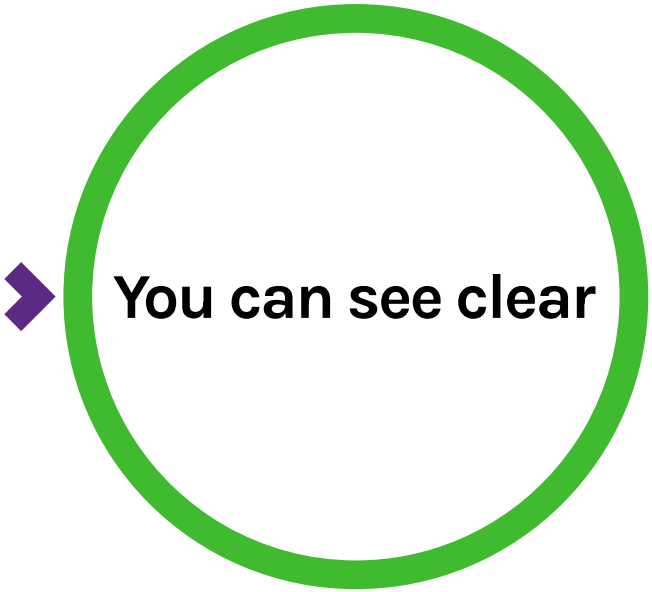
COLLABORA

Re-adding Weston

- The display still **freeze to solid** color
- The Hantro driver always **report errors**
- The number of **error IRQ** is massive (over **32K**)

COLLABORA

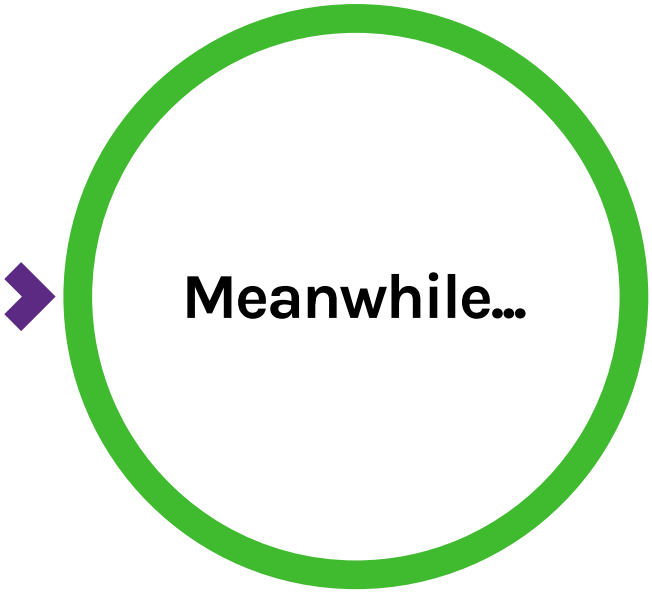
Give yourself some credit!



- There is **clear error** at the moment of failure
- You know there is **possible memory error**
- And that these cause massive **IRQ flood**

COLLABORA

The lucky information



- Some new reported feedback

  ”It feels like if I **seek to a specific position** it will **always trigger** the issue.”

COLLABORA

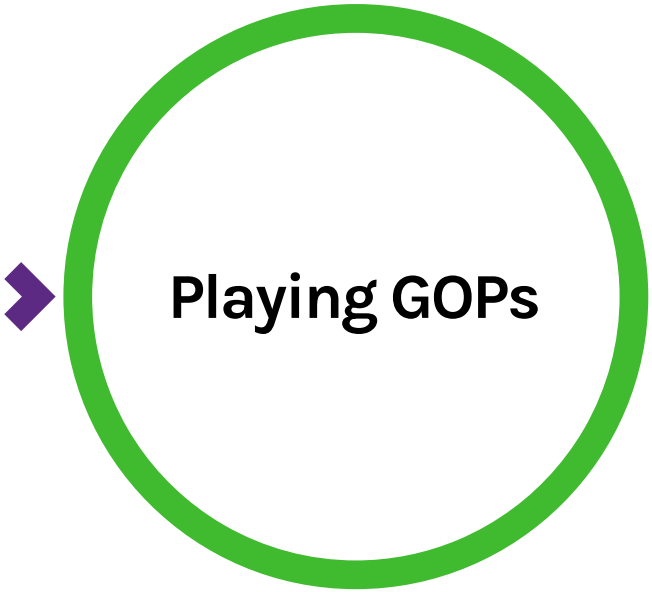
Skill required: Bitstream Cutting

Structure of an ISOMP4

- Has a index to all start of **GOP** in the stream
- Valid file are **guaranteed** to seek to a closed GOP
- A closed GOP is typically a set of pictures that can be decoded without any prior frames
- If you can split GOPs, you will be able to isolate and inspect that GOP

Breaking GOPs with multifilesink

```
gst-launch-1.0 filesrc location=h265.mp4 \
  ! qtdemux \
  ! h265parse config-interval=-1 \
  ! video/x-h265,stream-format=byte-stream \
  ! multifilesink location="gop-%03d.h265" \
                  next-file=key-frame
```



gst-play-1.0 gop-003.h265

COLLABORA

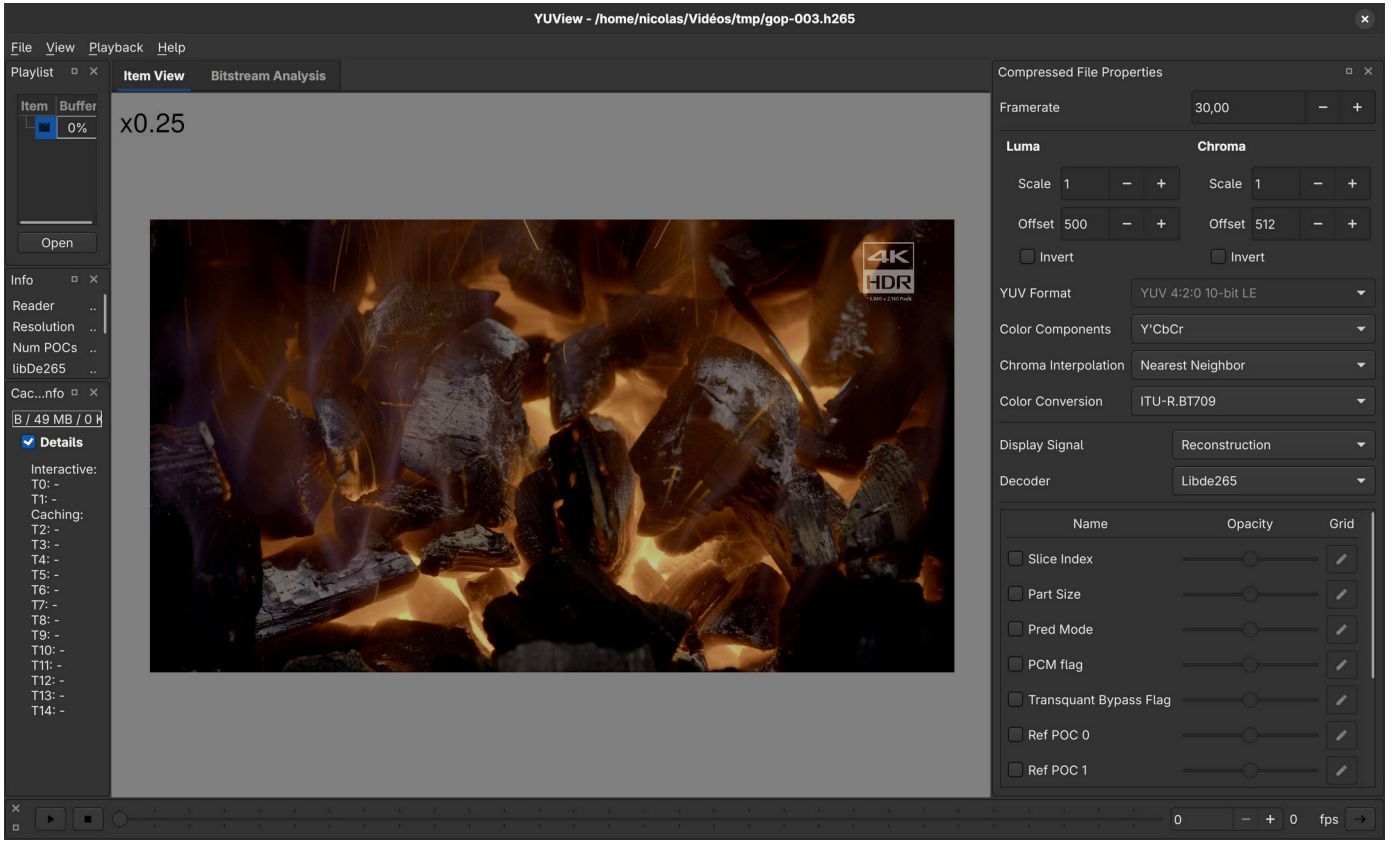
Bitstream analyses

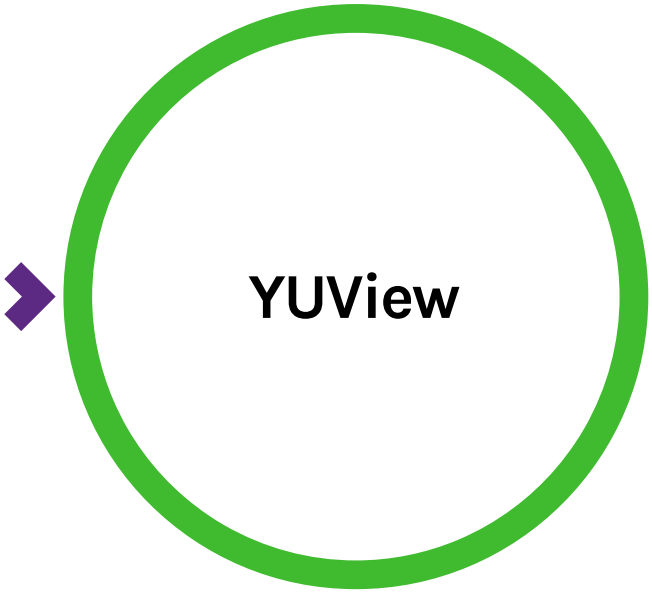
- Most bugs are investigated through tracing
- Tracing H.265 is possible, but tedious.
-
```
GST_DEBUG="codecparsers*:7" \
gst-launch-1.0 filesrc location=some.mp4  \
                  ! parsebin ! fakesink
```

Logging the parsers

```
gsth265parser.c:1868:gst_h265_parse_vps: parsing VPS
gsth265parser.c:319:gst_h265_parse_profile_tier_level: parsing "ProfileTierLevel parameters"
gsth265parser.c:1844:gst_h265_parser_parse_vps: adding video parameter set with id: 0 to array
gsth265parser.c:2045:gst_h265_parse_sps: parsing SPS
gsth265parser.c:319:gst_h265_parse_profile_tier_level: parsing "ProfileTierLevel parameters"
gsth265parser.c:499:gst_h265_parse_vui_parameters: parsing "VUI Parameters"
gsth265parser.c:2016:gst_h265_parser_parse_sps: adding sequence parameter set with id: 0 to array
gsth265parser.c:2306:gst_h265_parse_pps: parsing PPS
```

YUView


YUView - /home/nicolas/Vidéos/tmp/gop-003.h265
File View Playback Help
Playlist
Item Buffer
0%
Open
Info
Reader
Resolution
Num POCs
libDe265
Cac...nfo
B / 49 MB / 0 K
Details
Interactive:
T0: -
T1: -
Caching:
T2: -
T3: -
T4: -
T5: -
T6: -
T7: -
T8: -
T9: -
T10: -
T11: -
T12: -
T13: -
T14: -
Item View
Bitstream Analysis
x0.25
4K HDR
Compressed File Properties
Framerate 30,00
Luma
Chroma
Scale 1
Offset 500
Invert
Scale 1
Offset 512
Invert
YUV Format YUV 4:2:0 10-bit LE
Color Components Y'CbCr
Chroma Interpolation Nearest Neighbor
Color Conversion ITU-R.BT709
Display Signal Reconstruction
Decoder Libde265
Name Opacity Grid
Slice Index
Part Size
Pred Mode
PCM flag
Transquant Bypass Flag
Ref POC 0
Ref POC 1
0
0
fps



GOP 0

Item View | Bitstream Analysis

Stream Info | Packet Analysis | B

Show stream | Color Code Strear

Name

- NAL 0: 32 VPS_NUT ID 0
- NAL 1: 33 SPS_NUT ID 0
- NAL 2: 34 PPS_NUT ID 0
- NAL 3: 20 IDR_N_LP (POC 0)
- NAL 4: 1 TRAIL_R (POC 5)
- NAL 5: 1 TRAIL_R (POC 3)
- NAL 6: 0 TRAIL_N (POC 1)
- NAL 7: 0 TRAIL_N (POC 2)
- NAL 8: 0 TRAIL_N (POC 4)

GOP 3

Item View | Bitstream Analysis

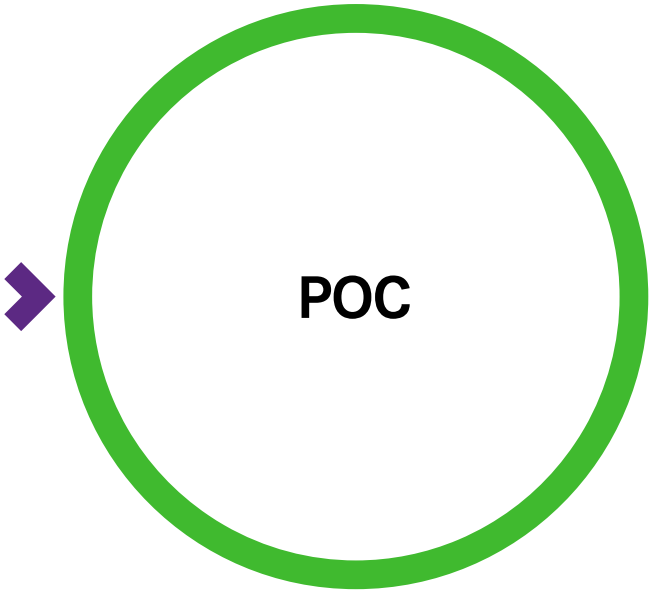
Stream Info | Packet Analysis | B

Show stream | Color Code Strear

Name

- NAL 0: 32 VPS_NUT ID 0
- NAL 1: 33 SPS_NUT ID 0
- NAL 2: 34 PPS_NUT ID 0
- NAL 3: 21 CRA_NUT (POC 135)
- NAL 4: 9 RASL_R (POC 133)
- NAL 5: 8 RASL_N (POC 132)
- NAL 6: 8 RASL_N (POC 134)
- NAL 7: 1 TRAIL_R (POC 137)
- NAL 8: 0 TRAIL_N (POC 136)

- **IDR:** Instantaneous Decoding Refresh
- **CRA:** Clean Random Access
- **RASL:** Random Access Skipped Leading (Picture)



- **CRA** has POC 135
- **RASL:** have POC 132-134
- The RASL frame are before our seek point
- Worse, these RASL uses past prediction
- **Problem:** GStreamer isn’t **skipping** them

COLLABORA

But … Linux shouldn’t break on missing references !

v4l2codecs

```
static guint8
lookup_dpb_index (struct v4l2_hevc_dpb_entry dpb[16], GstH265Picture * ref_pic)
{
  guint64 ref_ts;
  gint i;

  /* Reference list may have wholes in case a ref is missing, we should mark
   * the whole and avoid moving items in the list */
  if (!ref_pic)
    return 0xff;

  ref_ts = GST_CODEC_PICTURE_TS_NS (ref_pic);
  for (i = 0; i < 16; i++) {
    if (dpb[i].timestamp == ref_ts)
      return i;
  }

  return 0xff;
}
```

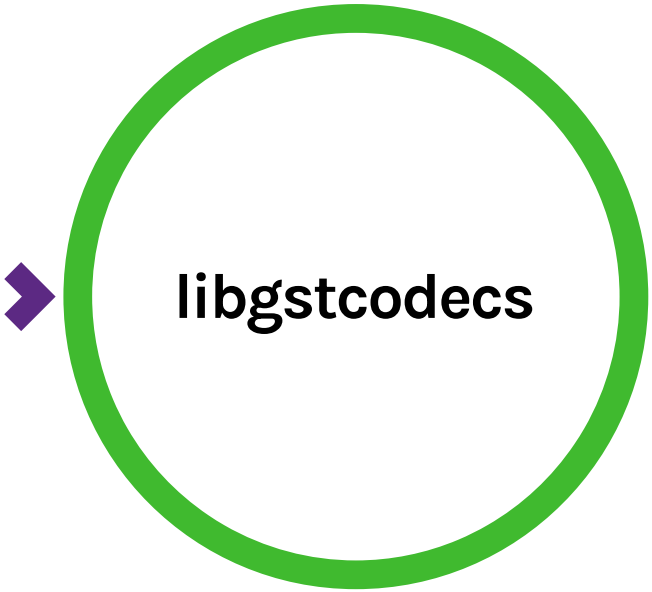
Hantro Decoder

```
commit 47825b1646a6a9eca0f90baa3d4f98947c2add96
Author: Nicolas Dufresne <nicolas.dufresne@collabora.com>
Date:   Mon Sep 22 14:43:39 2025 -0400

    media: verisilicon: Protect G2 HEVC decoder against invalid DPB index
```

COLLABORA

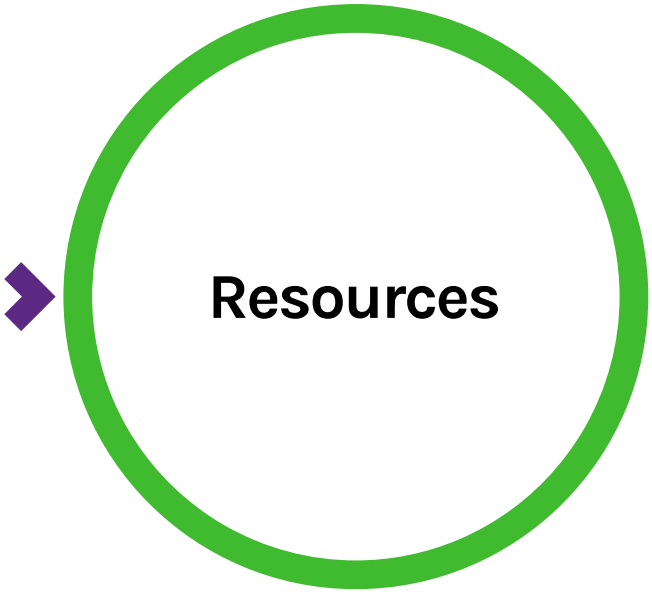
And the problem was gone!



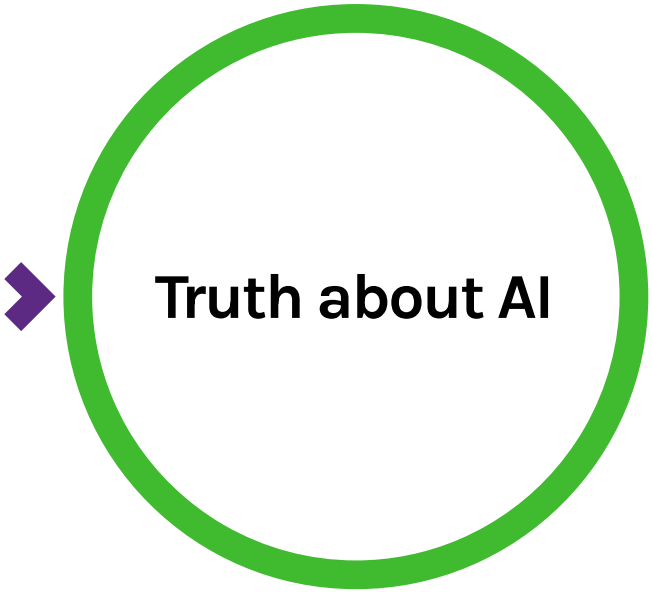
- On my **ToDo**
- The kernel fix was **enough** for now
- I had to move on, but should be fixed
- Image if you have that issue with a **closed source** decoder driver?

COLLABORA

What if I can’t find what to do?



- Search or ask on **discourse.gstreamer.org**
- File an **issue** in Gitlab
- Ask an **AI** assistant, but be **careful**



- Its wants to **please** you
- Most of the time can’t give **references**
- Answers from AI have to be **fact checked**
  - Check against **specs**
  - Check against **code**
  - Ask **someone else**, or another AI

- Debugging is **tedious** and **unpredictable**
- Since GStreamer **aggregates** everything, its may not even be a GStreamer bug
- Yes, your GStreamer **skills** are required
- With time, you learn **some patterns**, and that is what will **speedup** the process

Question ?

Thank you!