

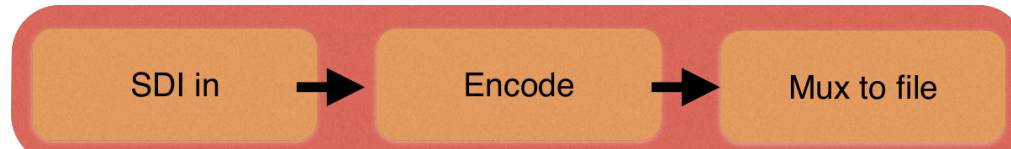
# SMPTE timecodes in GStreamer

Vivia Nikolaidou  
ToolsOnAir Broadcasting GmbH  
2016-10-11

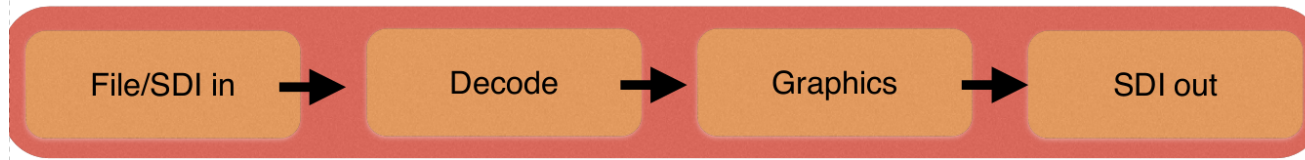
# ToolsOnAir

- Based in Vienna, Austria
- Existing products (OSX-only)

**just:in**



**just:out**



# Broadcast production



CC-BY-2.0, Nayu Kim

# SMPTE timecodes

Frame labeling

**hh:mm:ss:ff**  
**(hours : minutes : seconds : frames)**

*(PS: there are better solutions for synchronization)*

# SMPTE timecodes and strange frame rates

- Common in standards like NTSC
  - 29.97 fps ( $30000/1001$ ), 59.94 fps ( $60000/1001$ )
- Drop-frame flag
  - The first 2 and 4 (respectively) timecodes of each minute are skipped except every tenth minute
  - This way we lose less than one frame's duration during the whole day
- ... but we still lose a bit
  - Daily jam mandatory, once per day we reset to 00:00:00;00

# SMPTTE timecodes implementation

«enum»

## GstVideoTimeCodeFlags

GST\_VIDEO\_TIME\_CODE\_FLAGS\_DROP\_FRAME  
GST\_VIDEO\_TIME\_CODE\_FLAGS\_INTERLACED

## GstVideoTimeCodeConfig

- fps\_n : guint
- fps\_d : guint
- latest\_daily\_jam : GDateTime
- flags : GstVideoTimeCodeFlags

## GstVideoTimeCode

- hours : guint
- minutes : guint
- seconds : guint
- frames : guint
- field\_count : guint
- config : GstVideoTimeCodeConfig

# SMPTTE timecodes implementation

- Add / subtract frames, increment
- Number of frames since midnight (or daily jam)
- new, new\_empty, init, clear
- is\_valid
- compare (Will take daily jam into account if both have it)
- to\_string, to\_date\_time
- nsec\_since\_daily\_jam, frames\_since\_daily\_jam

# Retrieving a timecode

- From a decklink device (decklinkvideosrc)
  - If found, it will automatically be attached to the frames as meta



# Adding a timecode (for debugging)

- timecodestamper
  - Automatically detect frame rate
  - Start from 00:00:00
  - Flags:
    - Override existing time code
    - Drop frame
    - Daily jam information
    - Source clock (for first timecode)

# Displaying a timecode (for debugging)

- `timeoverlay time-mode=time-code`

# Waiting for a timecode

- `timecodewait`
  - Will drop all buffers until desired timecode is reached
  - Synchronized audio-video waiting
    - This means that both audio and video pass through it
  - Target timecode can be configured as string or object

# Outputting a timecode

- File output
  - qtmux will automatically write a timecode track if detected
- Decklink output
  - decklinkvideosink will automatically output timecode information if detected

# Summary of changes

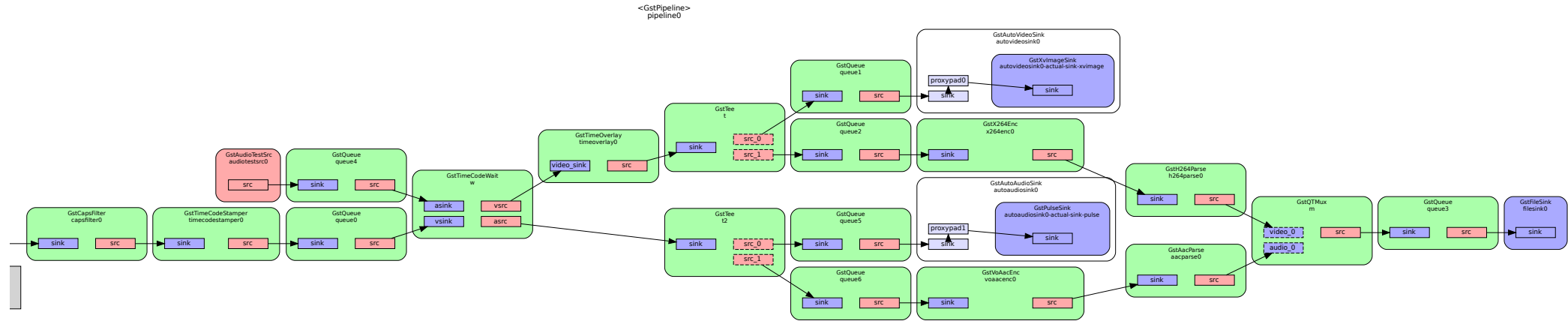
- `gst-plugins-base`
  - `GstVideoTimeCode`, `GstVideoTimeCodeMeta`
  - `timeoverlay`
- `gst-plugins-good`
  - `qtmux`
- `gst-plugins-bad`
  - `decklinkvideosrc`, `decklinkvideosink`
  - `timecodestamper`
  - `timecodewait`

# Sample image

- videotestsrc !  
timecodestamper !  
timeoverlay



# Bigger demo pipeline



# Demo time!

```
gst-launch-1.0 -e videotestsrc num-buffers=250 !  
"video/x-raw,width=1280,height=720" ! timecodestamper ! queue ! timecodewait  
target-timecode-string="00:00:03:00" name=w ! timeoverlay time-mode=time-code  
! tee name=t t. ! queue ! autovideosink t. ! queue ! x264enc tune=zerolatency  
! h264parse ! qtmux name=m ! queue ! filesink location=output.mov audiotestsrc  
num-buffers=500 ! queue ! w.asink w.asrc ! tee name=t2 ! queue ! autoaudiosink t2. ! queue !  
voaacenc ! aacparse ! m.
```



This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 610370.



00:00:03:14

# Thank You!

vivia@toolsonair.com

TOOLS  air