How to make professional media users about FOSS

Kieran Kunhya
Structure

• Introduction to (live) broadcasting
• Technical issues
• Non-technical issues
Highly simplified Broadcast Chain

- Live sources
- Recorded Sources
- Automation
- Home
- Archive
Why bother?

• Get FOSS used in mission-critical roles delivering to millions
• Make FOSS software reach a professional quality
  – And then better than proprietary alternatives
• Increased flexibility (reconfigurability)
• People interested in using FOSS
Why would a broadcaster look at FOSS?

• They have no money
  – Usually niche channels (e.g religious, ethnic, independent media)

• They have money but can’t find anyone to give it to
  – Nothing available for their needs
Broadcast chain in the eyes of some

- Often using consumer grade interface (HDMI)
- Shaky design that might look good in their eyes
- Avoid these people setting agenda for project
Focus on the right audience

• In an ideal world broadcast FOSS will suit needs of everyone
  – Reality is need to focus on mainstream broadcast FOSS

• Make things that can be easily inserted into the broadcast chain
FOSS has the ingredients, but few recipes?

- FFmpeg has fast decoders (inc professional profiles), filtering. (de)muxing less good.
- x264 top class H.264 encoder (Blu-ray, broadcast etc)
- Most low-mid range products use FFmpeg
  – Often without correct licensing (let alone attribution)
Often so simple to provide recipes
TECHNICAL ISSUES
Timestamps

• Broadcasting is constant framerate
  – One case of variable framerate is special cased

• Most (all?) FOSS tools are variable framerate

• VFR has a big problem
  – **What is the duration of the last frame?**
    • Splicing problems, adaptive streaming problems etc

• Loss of precision in timestamps
  – e.g NTSC 33.33... period in a millisecond timebase
  – 0, 33, 67, 100, 133, 167
In MPEG-TS timestamps are special
  – DTS = CPB Removal Time, PTS = DPB Removal Time
  – Few OSS programs implement this correctly
    • They assume arbitrary remuxing anything into MPEG-TS

Timestamps can be negative
  – e.g. PTS of zero with b-frames means negative DTS
  – uint64_t pts = wrong!

Should really be using PTS and duration
Analogue Legacies

• Analogue clocks derived from constant framerates
  • Can go black-and-white otherwise

• (Whether you like it or not) most broadcasting is interlaced.

• Aspect ratio legacy
  – Aspect ratios apply to the analogue samples not the digital data (whether you like it or not)
Wrong Interlaced Chroma Upsampling

4:2:2 – BT.601

4:2:0 – MPEG-2
NON-TECHNICAL ISSUES
Standards Bodies

• Broadcast is heavily standards based
• Standards can cost a lot of money
  – Require you to buy dozens
  – Corporate licences available but meaningless for OSS
• Lack of return path for reporting issues/ambiguity
  – MPEG has good return path (jvt-experts, mp4-tech mailing lists)
  – SMPTE has no way of reporting
    • Leads to major interoperability problems
    • No place to discuss edge cases
Patents

• Many processes may or may not be patented (IANAL)
  – Broadcasters assume worst and expect equipment to have royalties paid
  – Lots of FUD – sadly some spread by OSS orgs

• We know where we stand
  – Source code not patentable
Support

• Broadcasters need commercial support (and someone to blame when it goes wrong!)
The future

• FOSS broadcast lacks a “LAMP Stack”
  – Low level enough to have precise control
  – Simple enough that detailed knowledge
    • e.g zero understanding of HTTP to use LAMP
  – Reliable

• This is HARD