

Virtual Hardware

Jan Schmidt

jan@centricular.com

<https://aus.social/@thaytan>

London October 2025



The Hardware Dependency Hurdle

Hardware-accelerated decoding is great... when you have the hardware!



Challenges for early development & CI/CD

Target hardware is expensive and/or unavailable
Debugging requires specific setups
Portability is hindered

Challenges for early development & CI/CD:

Need a consistent, reproducible development environment

but not every developer needs the actual hardware



Software Emulation

A Virtual V4L2 Decoder

A software decoder presented via video4linux2
stateful decoder API



Software Emulation

A Virtual V4L2 Decoder

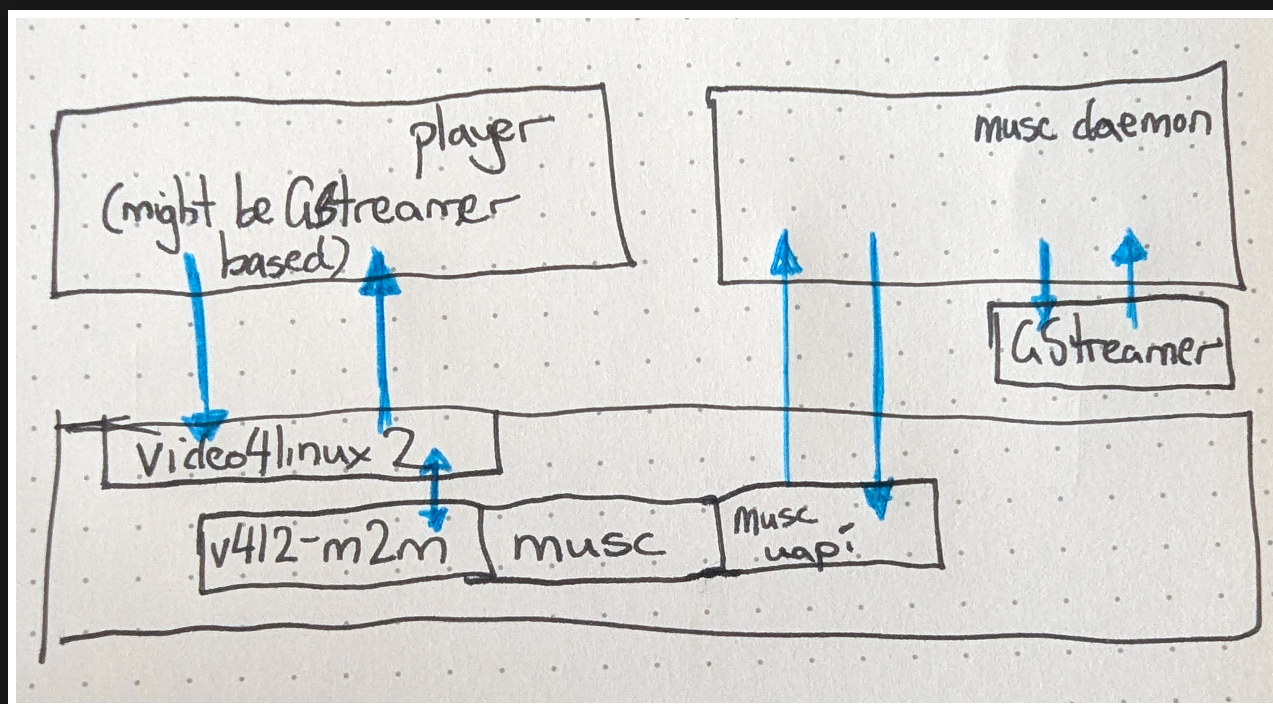
Develop and test pipelines on any Linux system.

Faster iteration and improved code quality.

Simplified CI/CD for the v4l2 plugin.

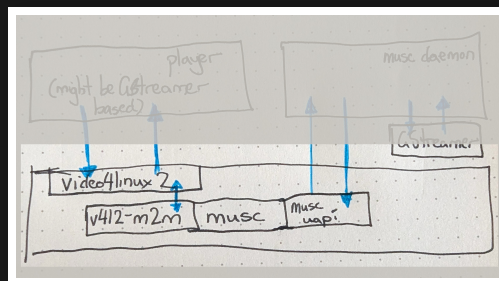


musc (multimedia userspace codec)



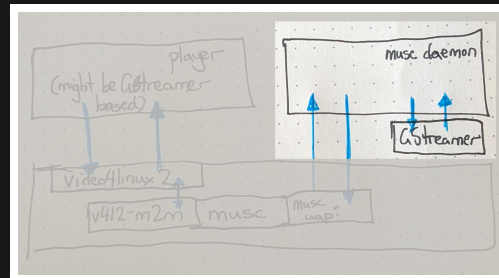
Architecture: V4L2 Kernel Module

- Implements V4L2 ioctls for device control and data transfer
- Receives requests from the player, routes to software backend
- Handles buffer management and synchronization between queues



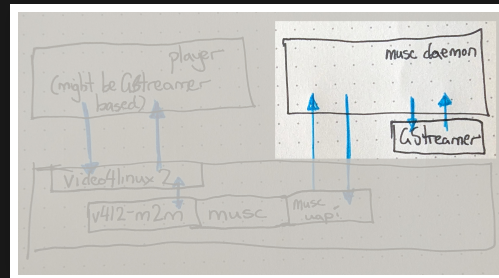
Architecture: Proxy Daemon

- The proxy daemon performs the actual video decoding
- Communicates with the kernel module via a simple device interface
- Declares which codecs are available, and frame formats



Architecture: Proxy Daemon

- Handles frame processing and buffer delivery
- Leverages GStreamer software decoding (but could be anything)



Challenges

dmabuf support
clean recovery in all cases
kernel version changes

Challenges

more color formats

HDR handling

stateless decoding?

Thank You!

<https://github.com/centricular/musc-decoder.git>

Questions?

