



COLLABORA

Shifting Media app development into high gear

Using virtual drivers to speed up development

Open First

Helen Koike
helen.koike@collabora.com



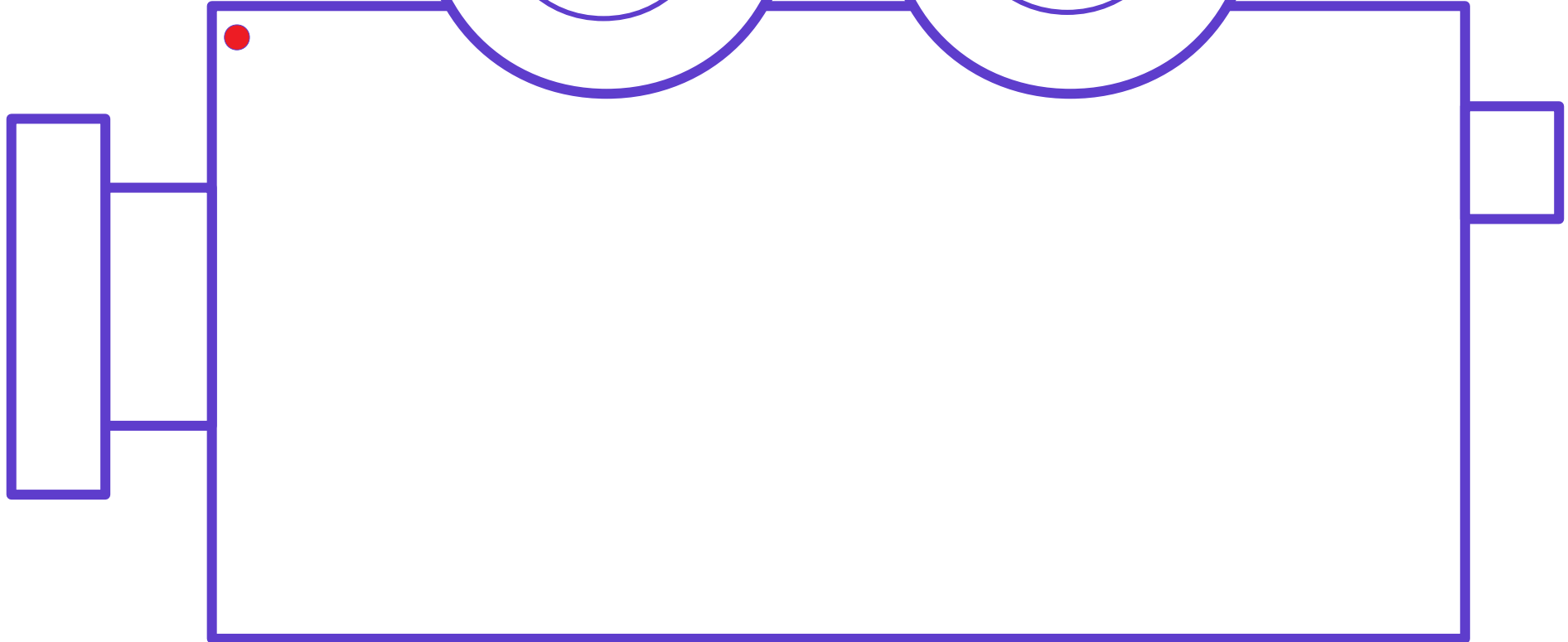
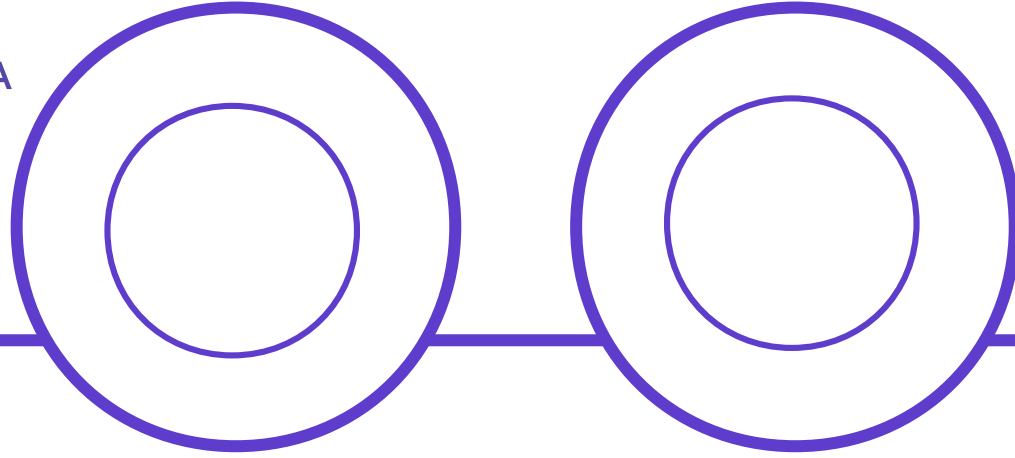
Summary

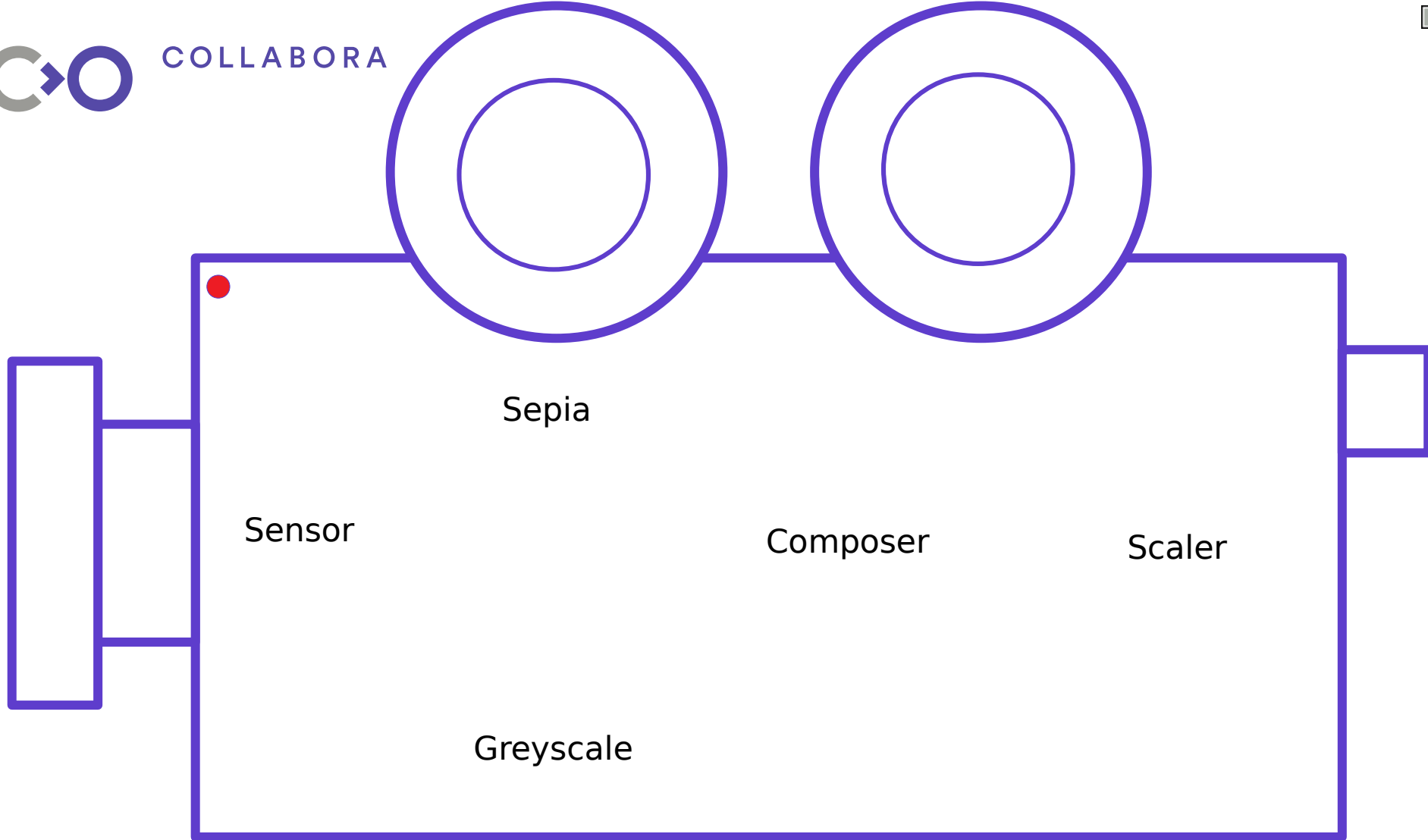
- Classic V4L2 API
 - Vivid Driver
 - Vimc Driver
 - Vicodec Driver
- Media API
- Codecs
- Vimc:
 - Submodules
 - Configfs API
 - Future work



Summary

- Classic V4L2 API
 - Vivid Driver
 - Vimc Driver
 - Vicodec Driver
- Media API
- Codecs
- Vimc:
 - Submodules
 - Configfs API
 - Future work

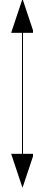






User space

APP



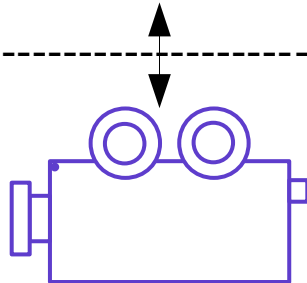
----- /dev/video* -----

Kernel space

Driver



Physical device





User space

APP

Read/Write/MMAP

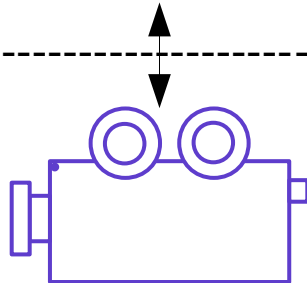
/dev/video*

Kernel space

Driver

Video stream

Physical device





User space

APP

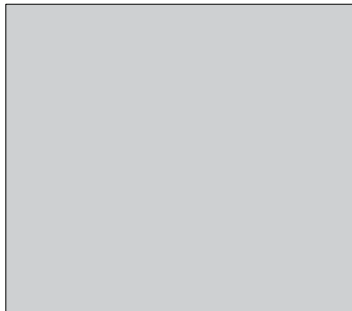
Read/Write/MMAP

IOCTLs

----- /dev/video* -----

Driver

Video stream



General configs

- * Img fmt
- * Buffers
- * Video std
- * Frame rate
- .
- .

Standard Controls

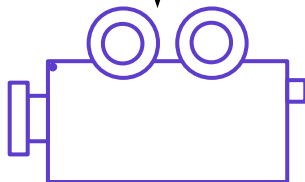
- * Contrast
- * Brightness
- * Gamma
- .
- .
- .

Custom Controls

- * DRV DEF 1
- * DRV DEF 2
- * DRV DEF 3
- .
- .
- .

Kernel space

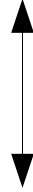
Physical device





User space

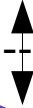
APP



----- /dev/video* -----

Kernel space

Driver



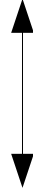
Physical device





User space

APP



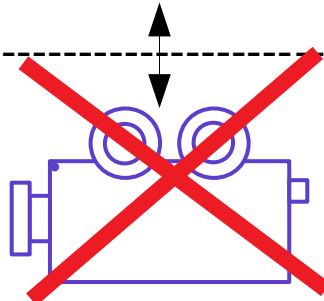
/dev/video*

Kernel space



Driver

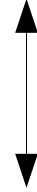
Physical device





User space

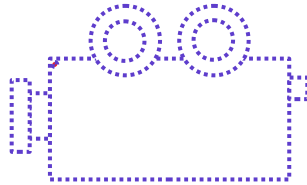
APP



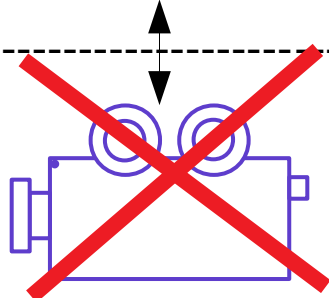
----- /dev/video* -----

Kernel space

The Virtual Video Test Driver (VIVID)



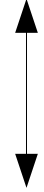
Physical device





User space

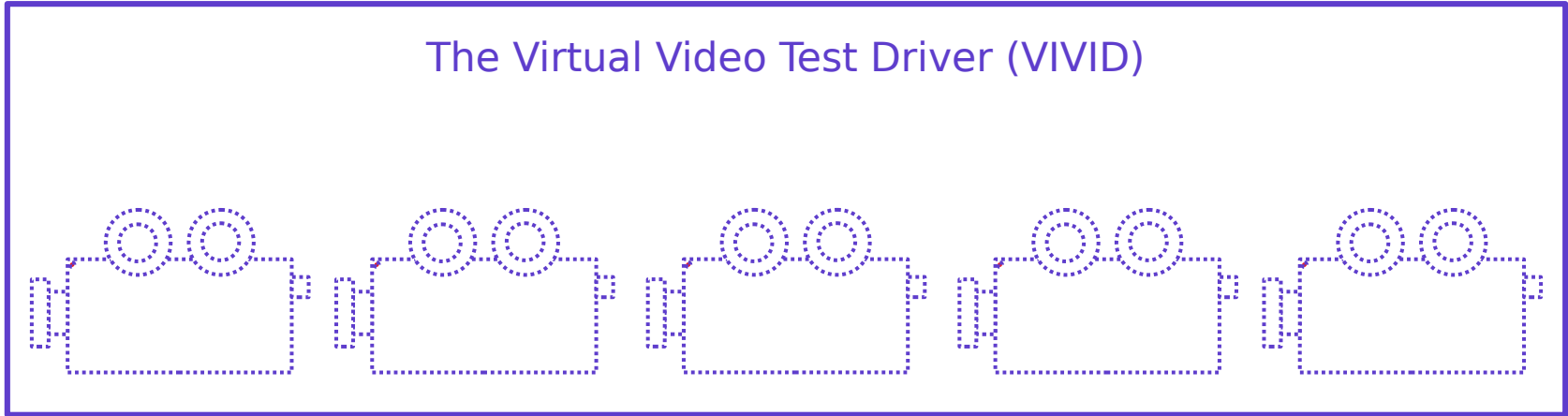
APP



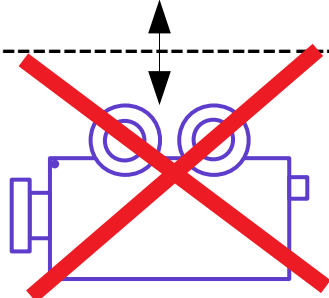
/dev/video*

Kernel space

The Virtual Video Test Driver (VIVID)



Physical device



Vivid driver

V4L2 Test Bench

File Capture Help

General Settings User Controls Digital Video Controls Vivid Controls

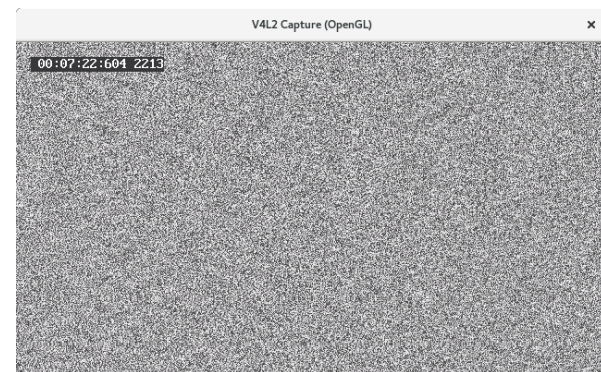
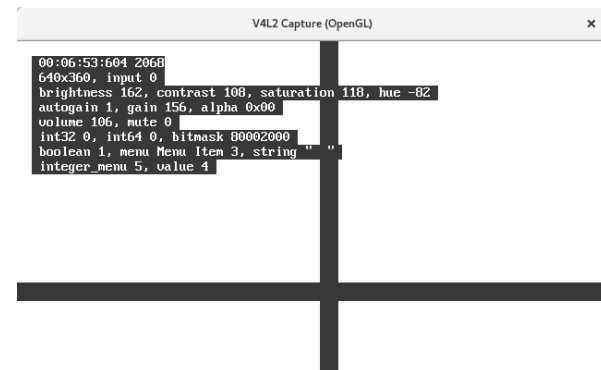
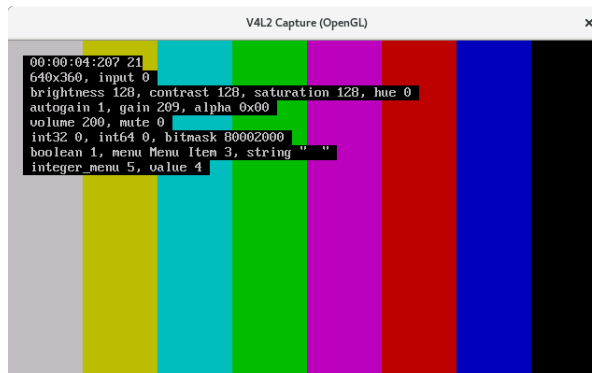
Vivid Controls

Test Pattern: 75% Colorbar	Enable Capture Composing: <input checked="" type="checkbox"/>
OSD Text Mode: All	Enable Capture Scaler: <input checked="" type="checkbox"/>
Horizontal Movement: No Movement	Loop Video: <input type="checkbox"/>
Vertical Movement: No Movement	Wrap Sequence Number: <input type="checkbox"/>
Show Border: <input type="checkbox"/>	Wrap Timestamp: <input type="checkbox"/>
Show Square: <input type="checkbox"/>	Maximum EDID Blocks: 2
Insert SAV Code in Image: <input type="checkbox"/>	Fill Percentage of Frame: 100
Insert EAV Code in Image: <input type="checkbox"/>	Reduced Framerate: <input type="checkbox"/>
Sensor Flipped Horizontally: <input type="checkbox"/>	HSV Encoding: Hue 0-179
Sensor Flipped Vertically: <input type="checkbox"/>	Standard Signal Mode: Current Standard
Standard Aspect Ratio: 4x3	Standard: NTSC-M
DV Timings Aspect Ratio: Source Width x Height	DV Timings Signal Mode: Current DV Timings
Timestamp Source: End of Frame	DV Timings: 640x480p59
Colorspace: sRGB	Percentage of Dropped Buffers: 0
Transfer Function: Default	Disconnect: <input type="button" value="↔"/>
YCbCr Encoding: Default	Inject V4L2_BUF_FLAG_ERROR: <input type="button" value="↔"/>
Quantization: Default	Inject VIDIOC_REQBUFS Error: <input type="button" value="↔"/>
Limited RGB Range (16-235): <input type="checkbox"/>	Inject VIDIOC_QBUF Error: <input type="button" value="↔"/>
Apply Alpha To Red Only: <input type="checkbox"/>	Inject VIDIOC_STREAMON Error: <input type="button" value="↔"/>
Enable Capture Cropping: <input checked="" type="checkbox"/>	Inject Fatal Streaming Error: <input type="button" value="↔"/>

Update on change

Set Defaults Refresh Update

Frame: 41 Fps: 5.00 Scale Factors: 1x1 SeqNr: 40





Vivid driver: current state

- Merged in 3.17 by Hans Verkuil
- Good coverage of the API
- Error injection
- Up to 4K resolution
- ...

More info:

<https://linuxtv.org/downloads/v4l-dvb-apis/v4l-drivers/vivid.html>



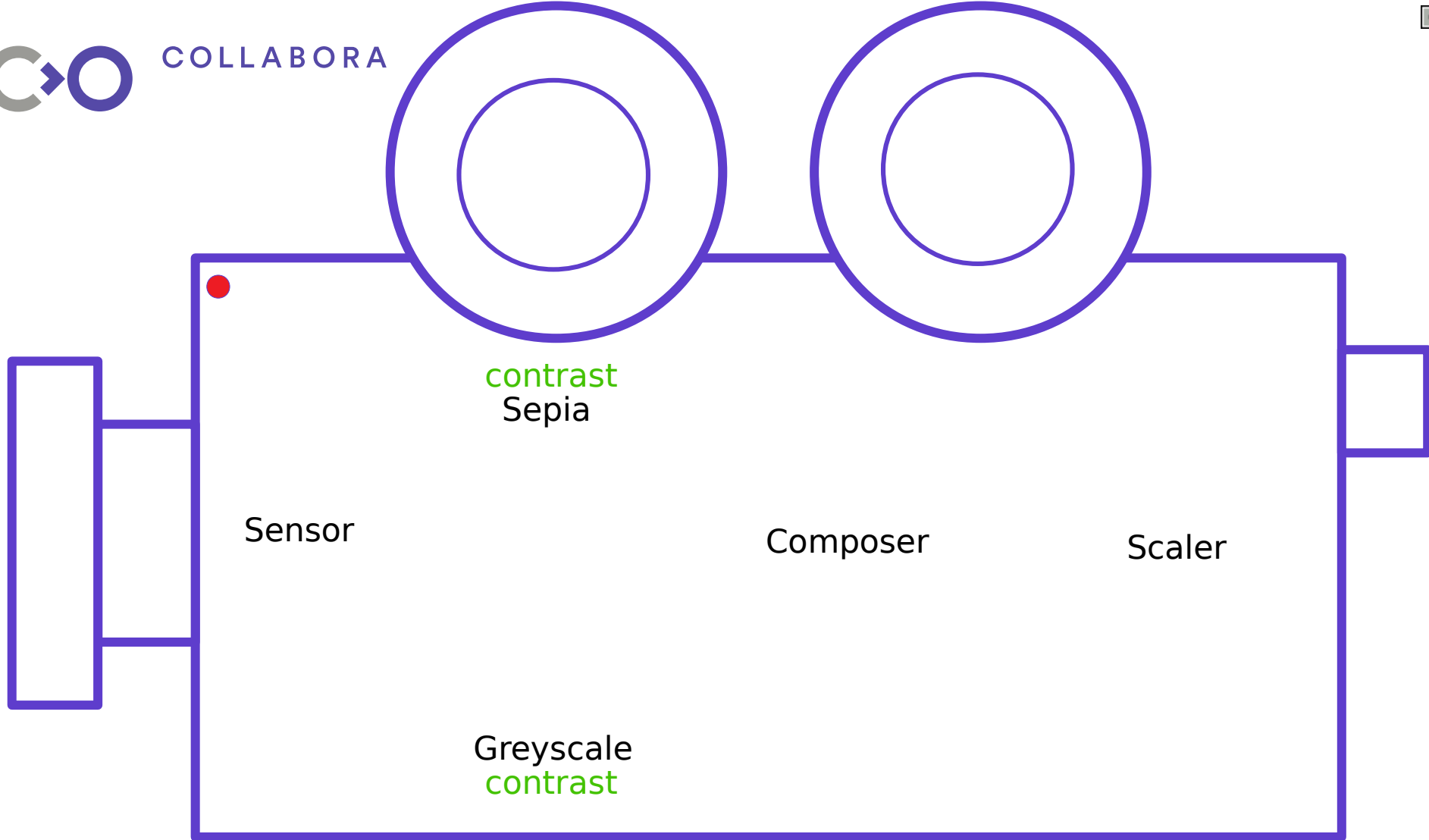
Summary

- Classic V4L2 API
 - Vivid Driver
 - Vimc Driver
 - Vicodec Driver
- Media API
- Codecs
- Vimc:
 - Submodules
 - Configfs API
 - Future work



Media API - Motivation

- Embedded systems: great variety of devices
- Hardware complexity
- Similar configuration in different components of the device





User space

APP

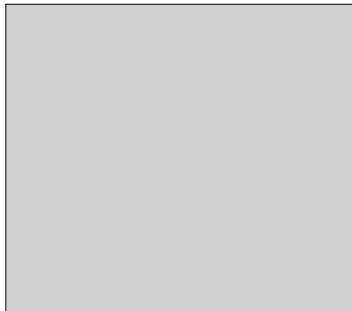
Read/Write/MMAP

IOCTLs

----- /dev/video* -----

Driver

Video stream



General configs

- * Img fmt
- * Buffers
- * Video std
- * Frame rate
- .
- .

Standard Controls

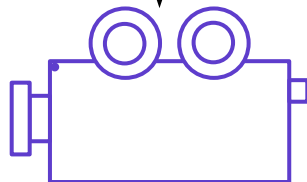
- * **Contrast**
- * Brightness
- * Gamma
- .
- .
- .

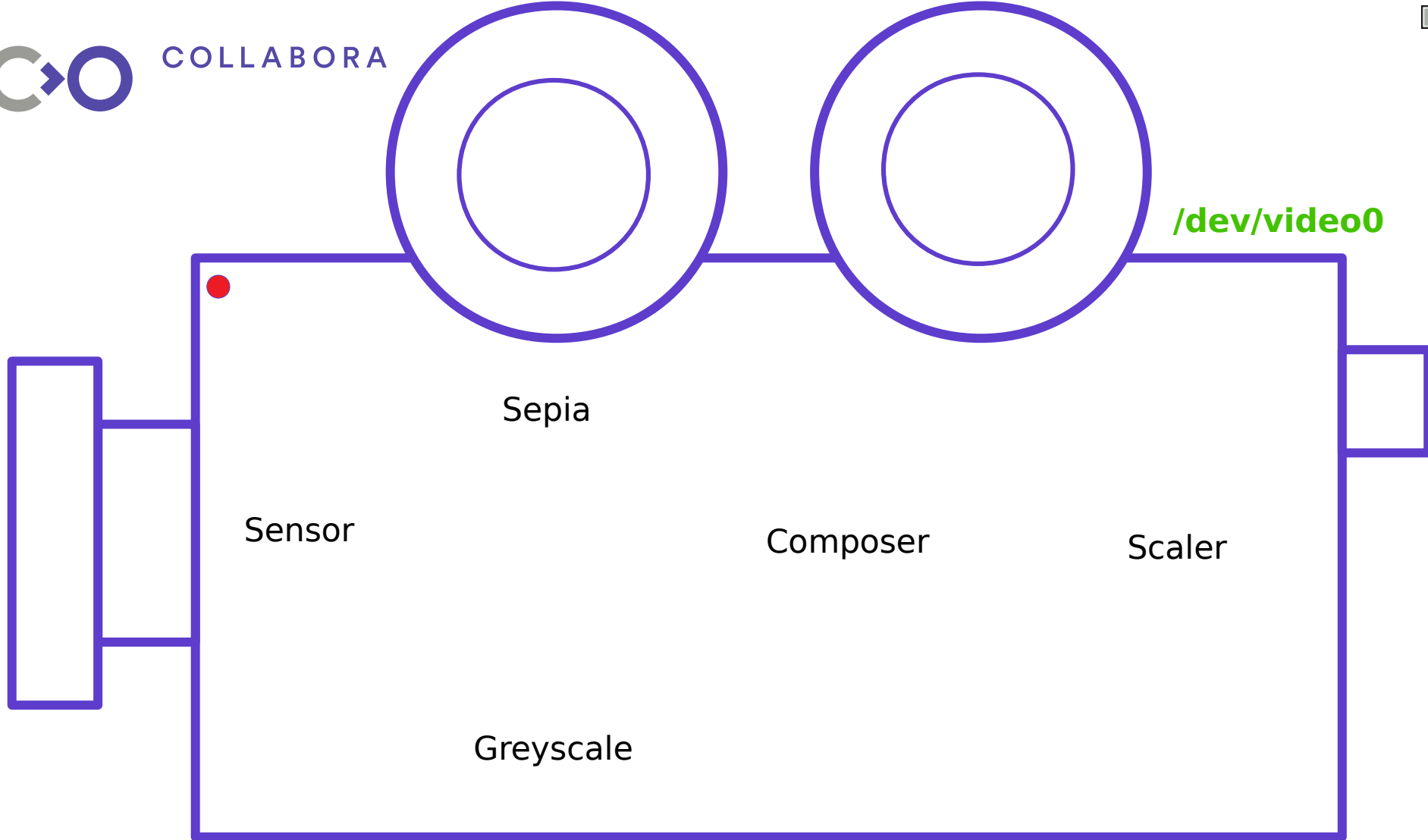
Custom Controls

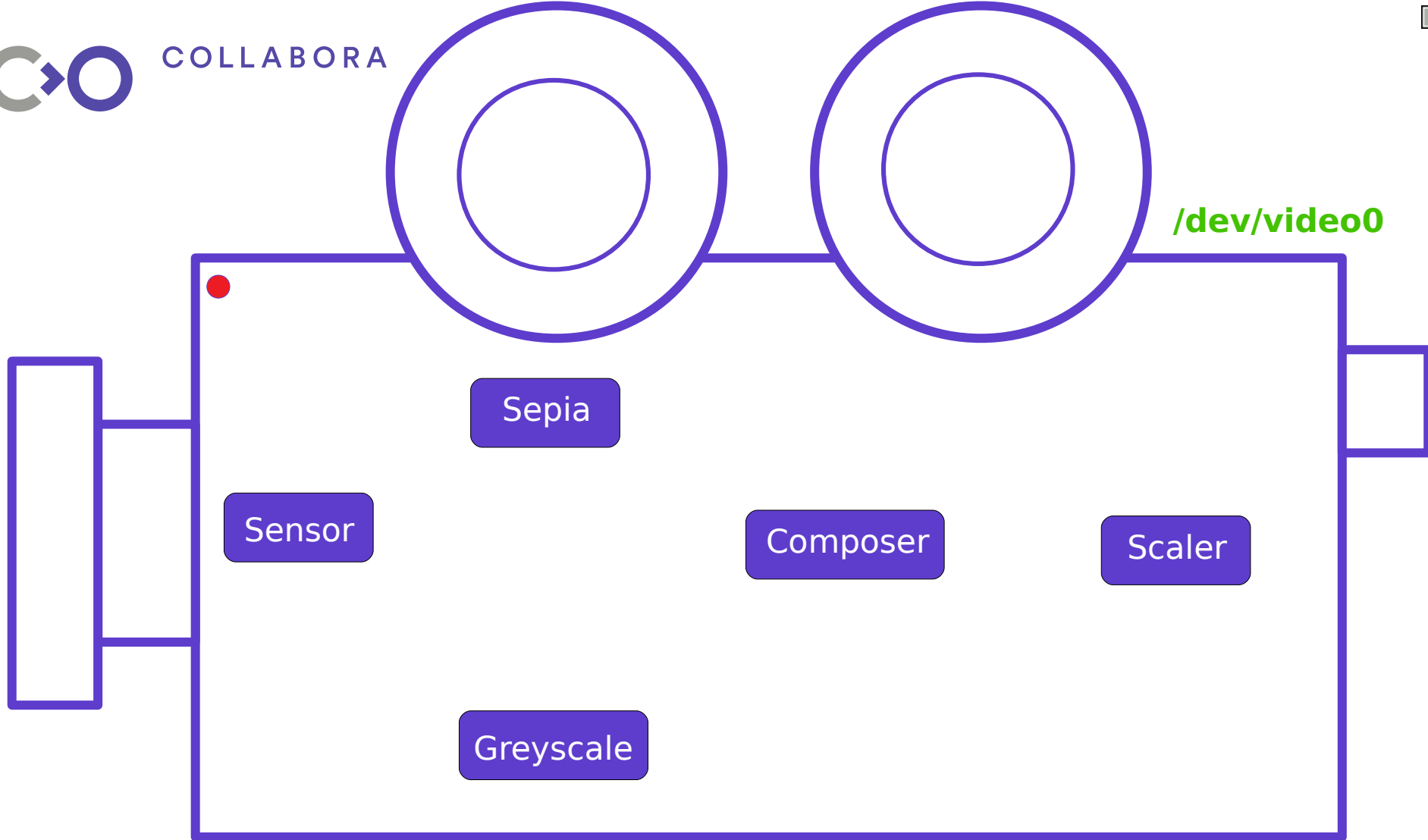
- * DRV DEF 1
- * DRV DEF 2
- * DRV DEF 3
- .
- .
- .

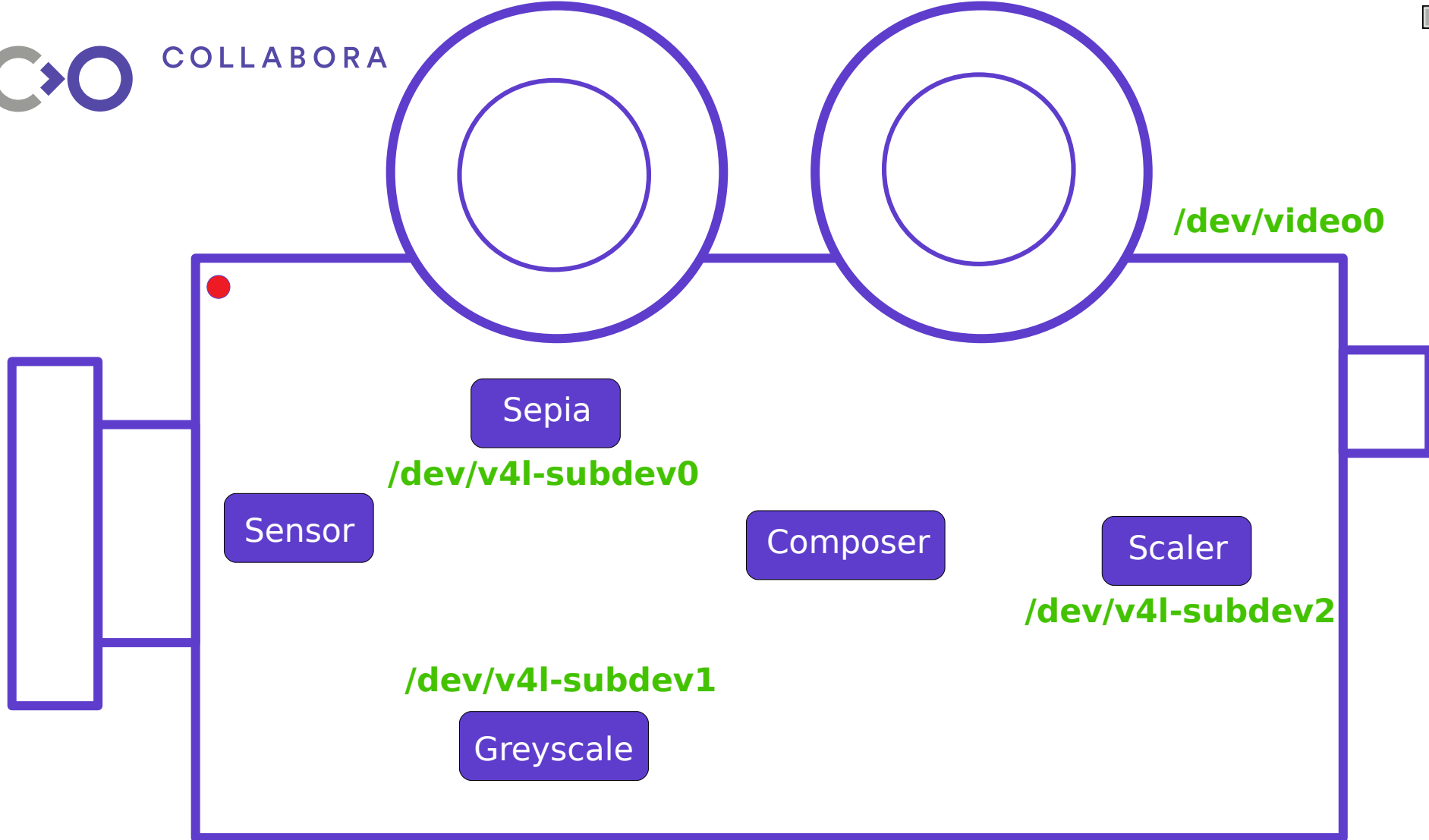
Kernel space

Physical device



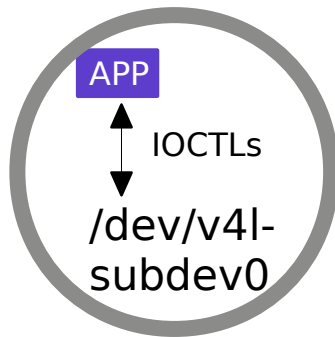
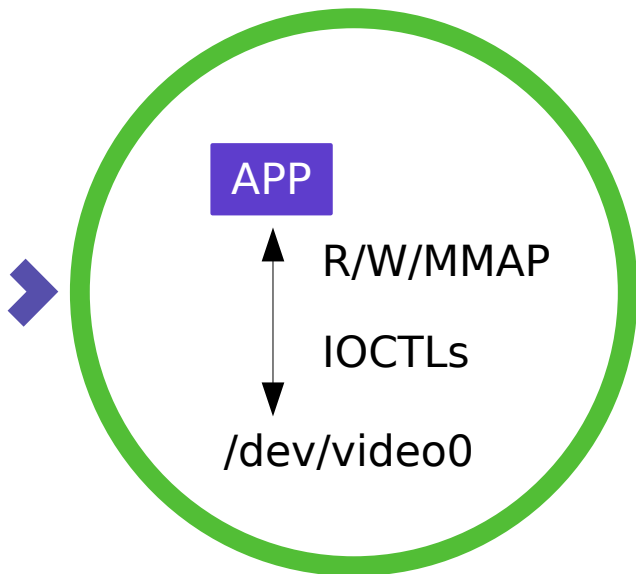




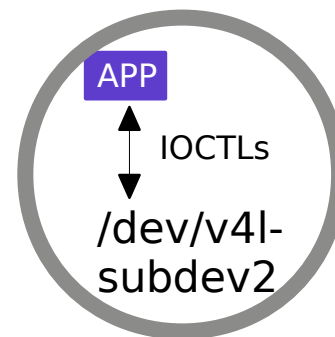




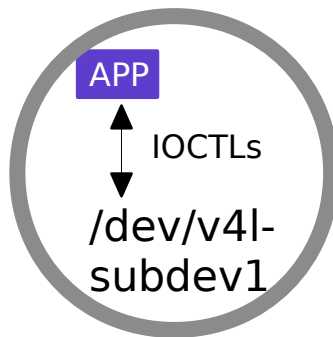
Subdev API



contrast
Sepia



Scaler

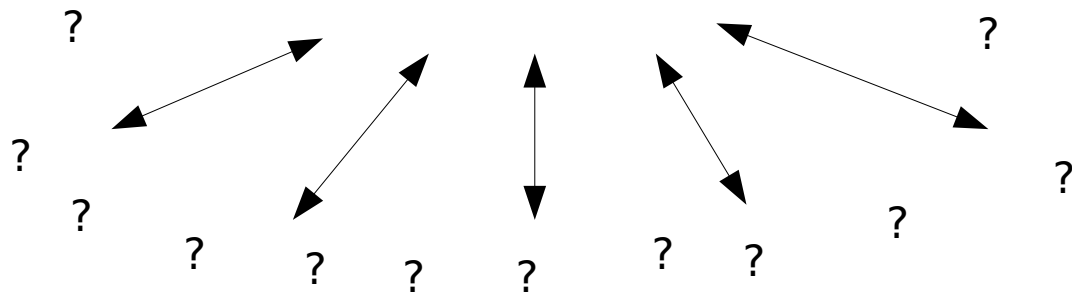


contrast
Greyscale



What is part of the device? What is the topology?

APP



/dev/v4l-subdev0

/dev/v4l-subdev98

/dev/v4l-subdev11

/dev/v4l-subdev14

/dev/video0

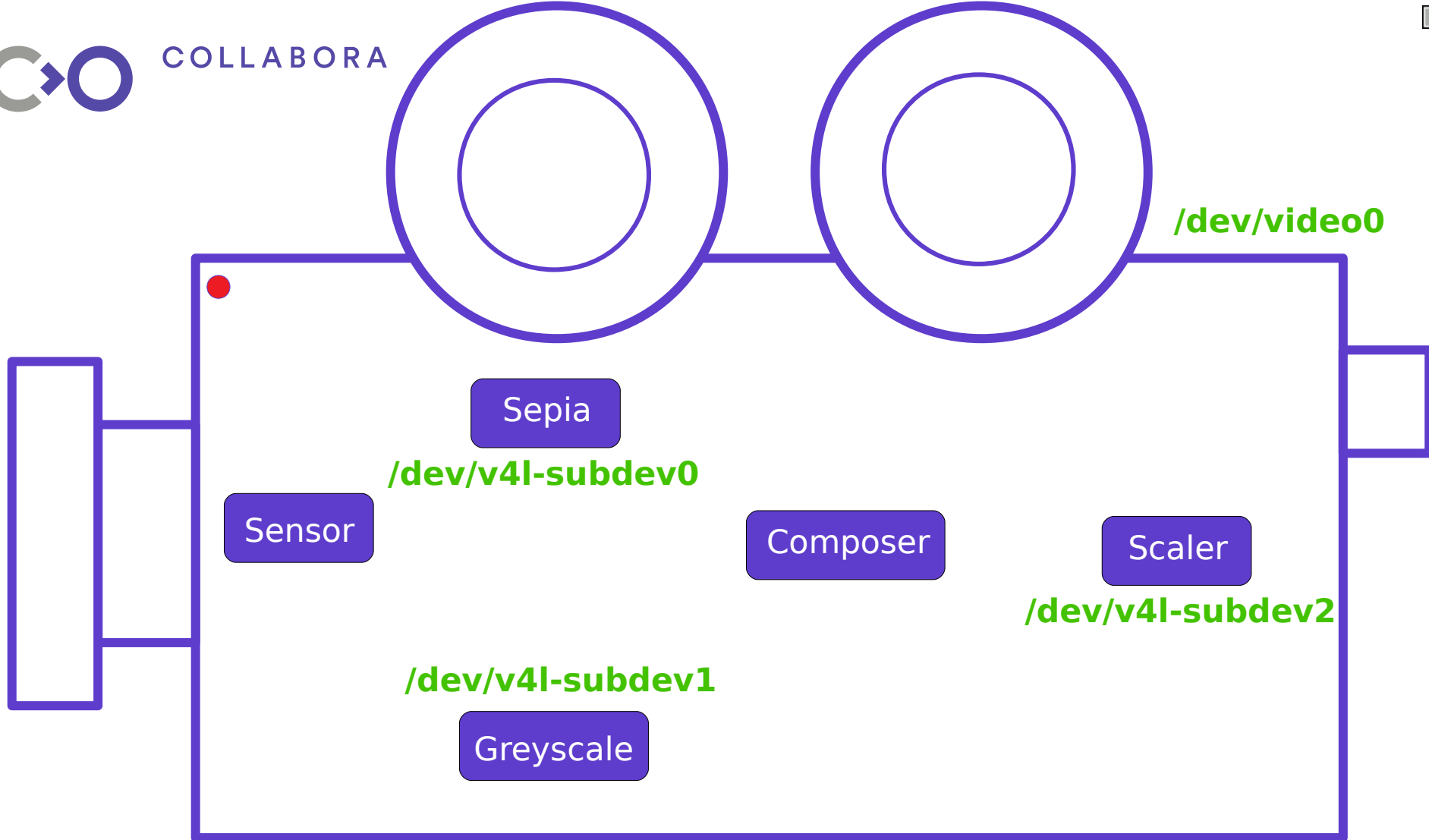
/dev/v4l-subdev6

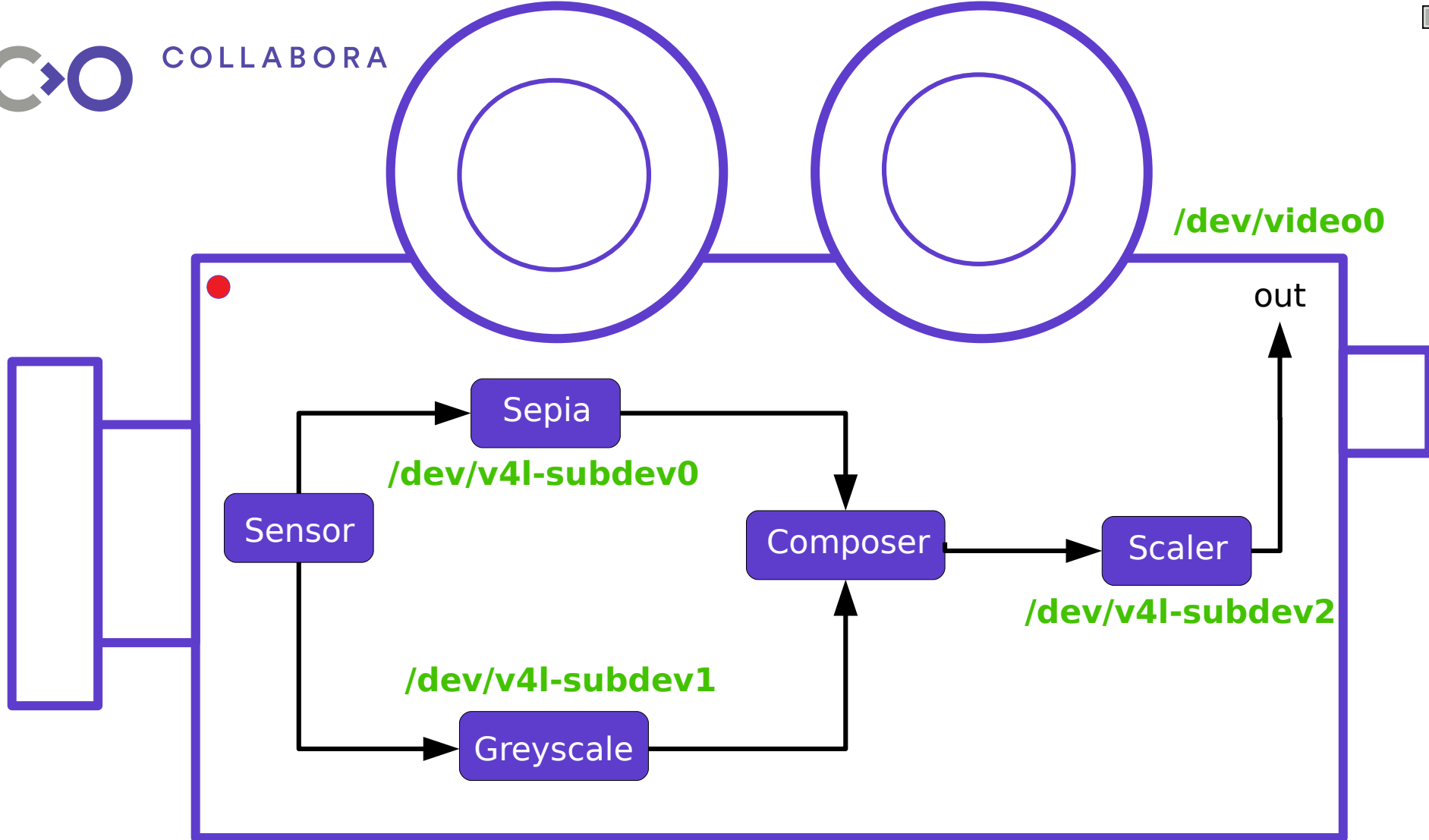
/dev/video2

/dev/video42

/dev/v4l-subdev21

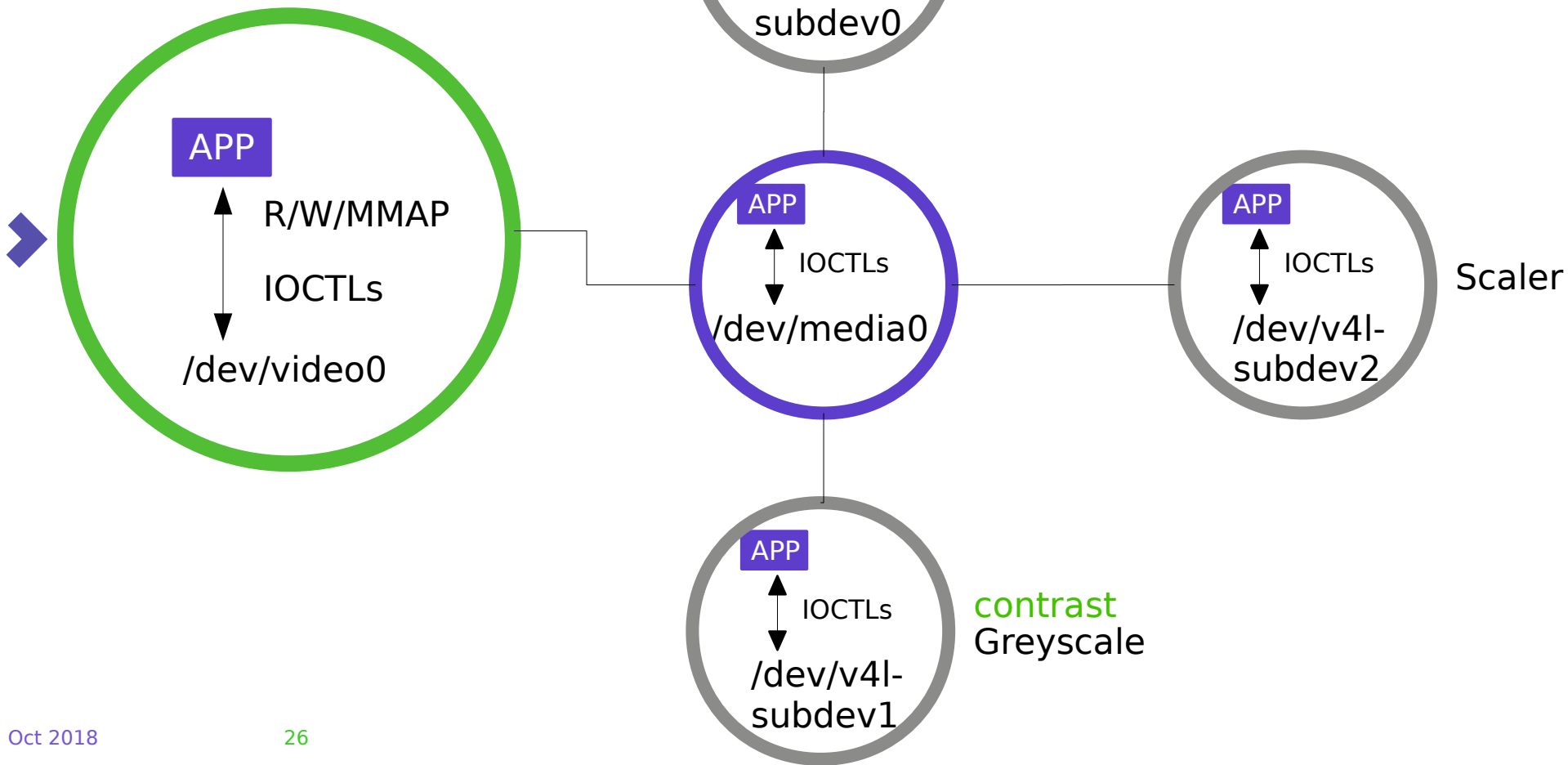
/dev/v4l-subdev31

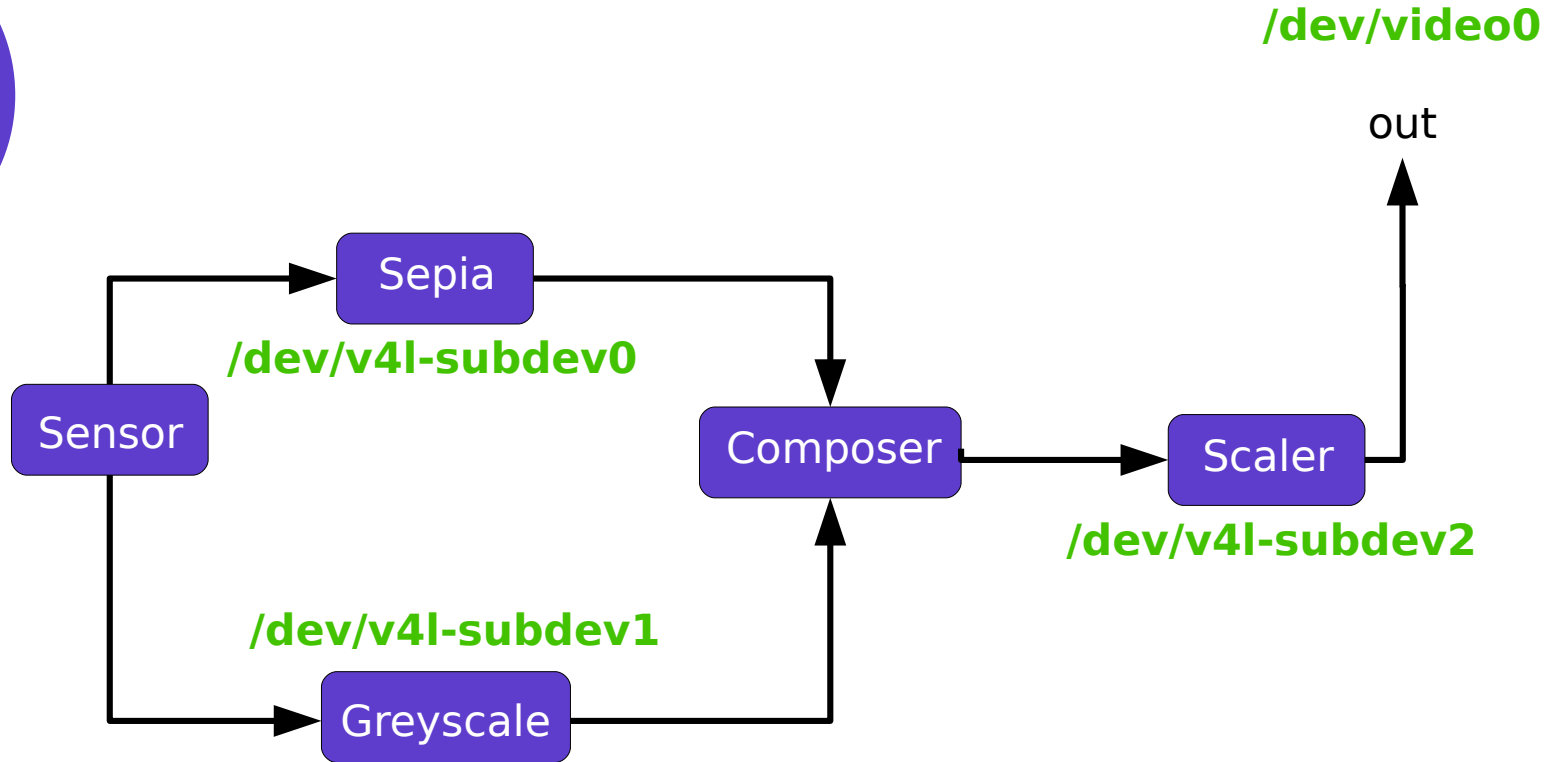
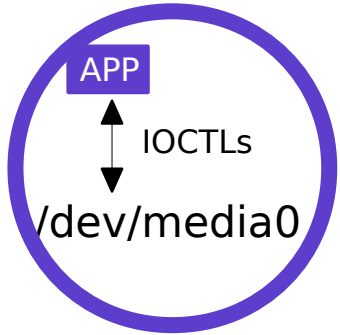


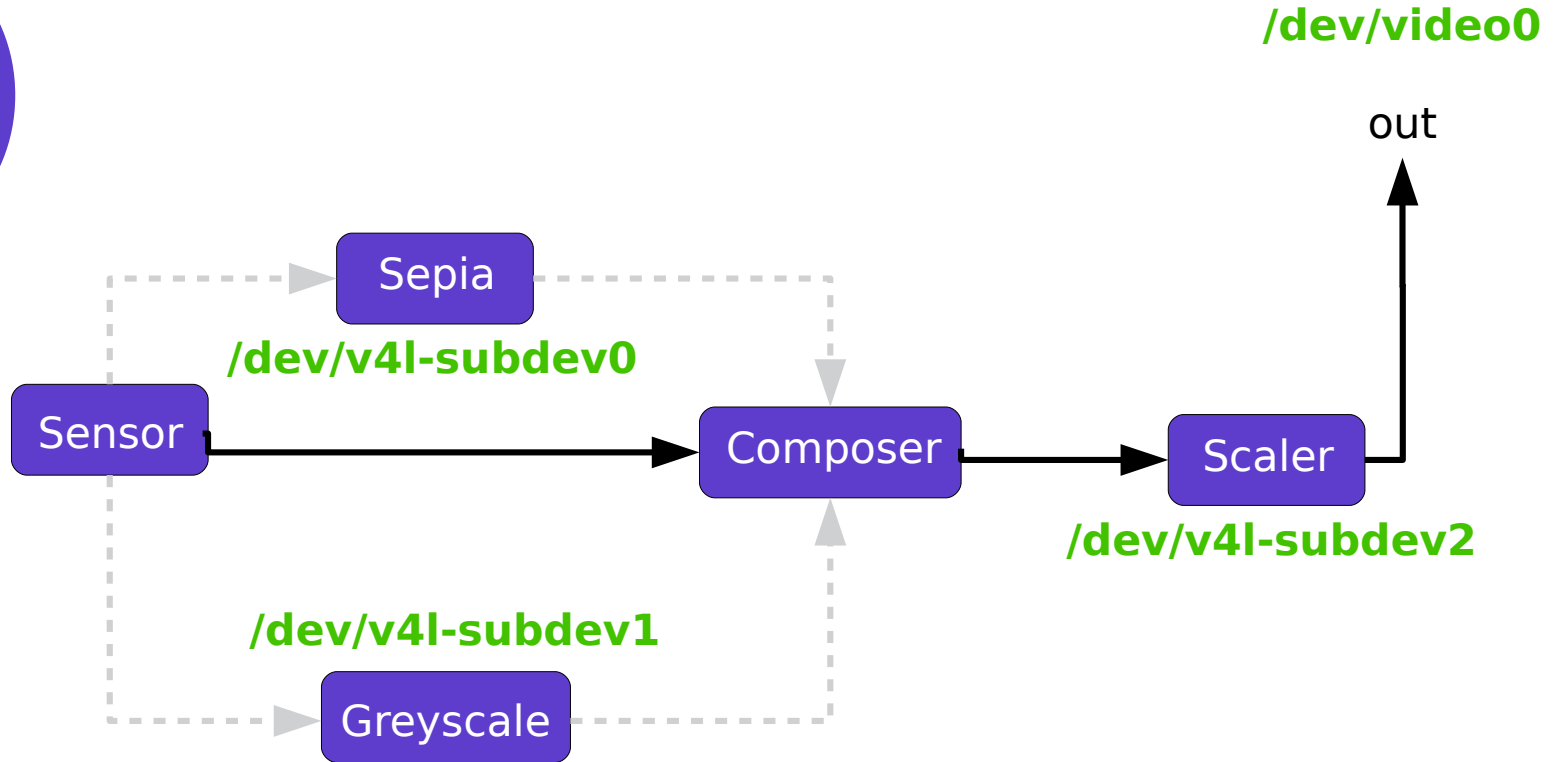
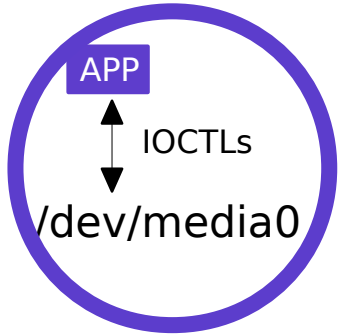




Media API



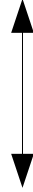






User space

APP



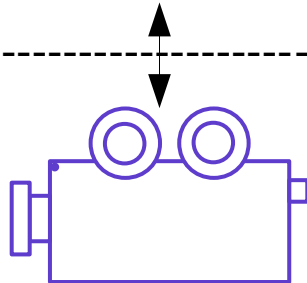
----- /dev/media* -----

Kernel space

Driver



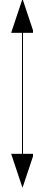
Physical device





User space

APP

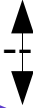


----- /dev/media* -----

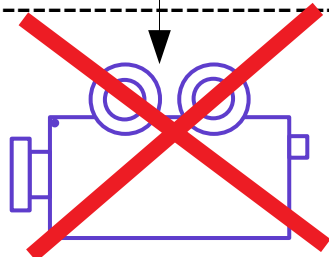
Kernel space



Driver



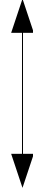
Physical device





User space

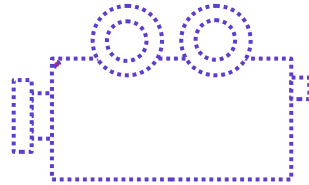
APP



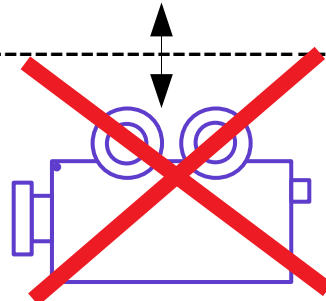
----- /dev/media* -----

Kernel space

The Virtual Media Controller Driver (VIMC)



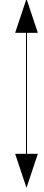
Physical device





User space

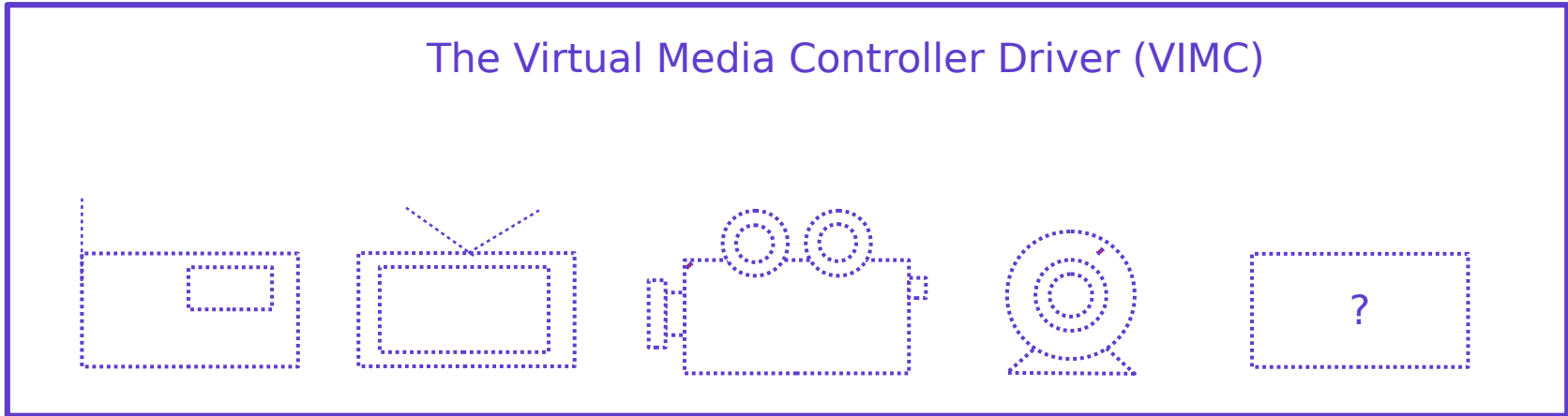
APP



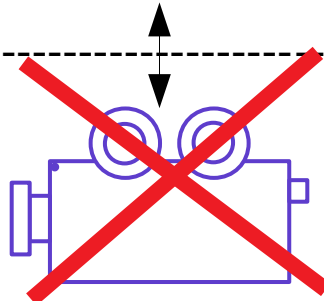
----- /dev/media* -----

Kernel space

The Virtual Media Controller Driver (VIMC)



Physical device



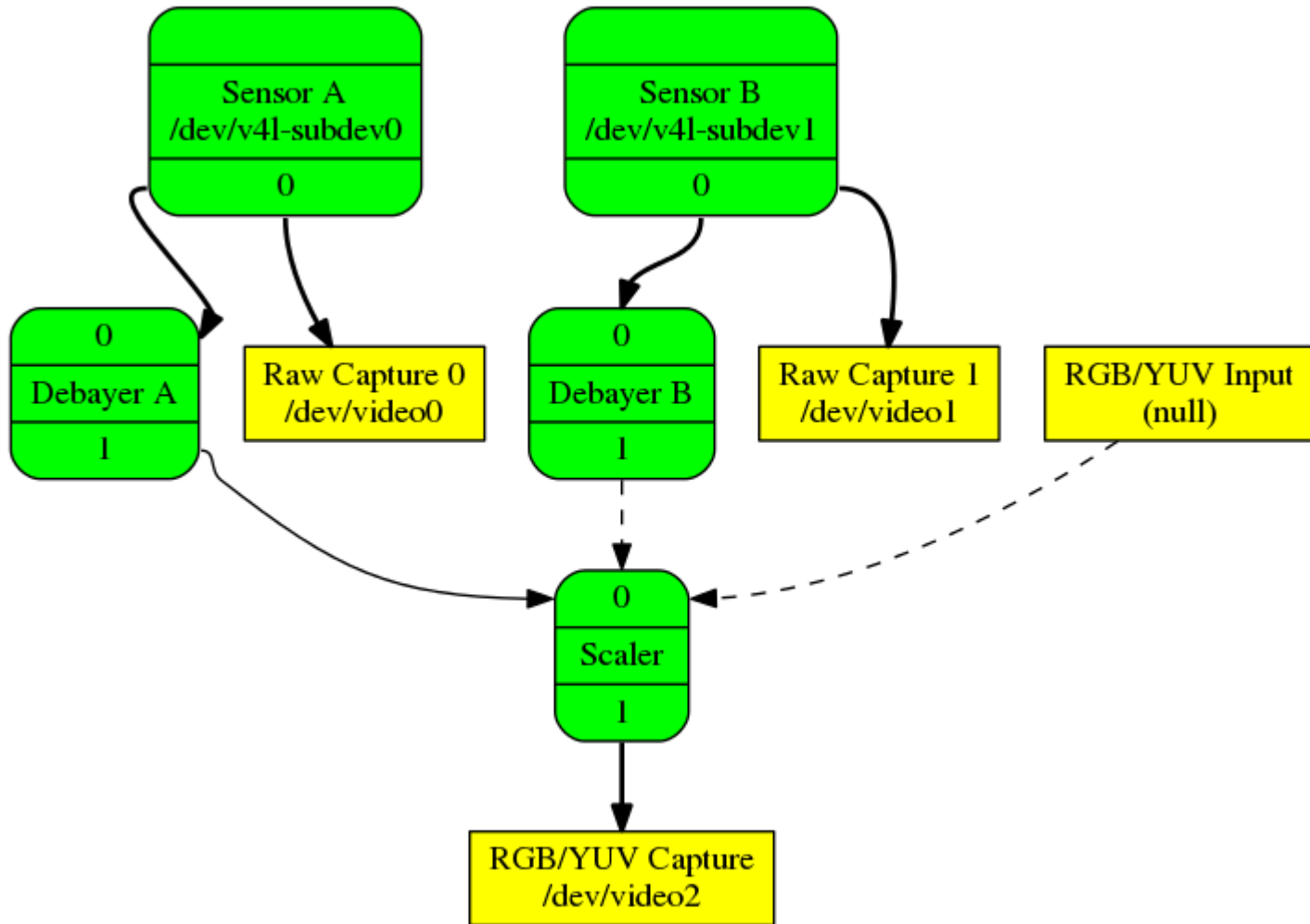
Vimc Driver

- Proposed by Laurent Pinchart for Outreachy in 2015
- Merged in Kernel 4.12
- Moving slowly



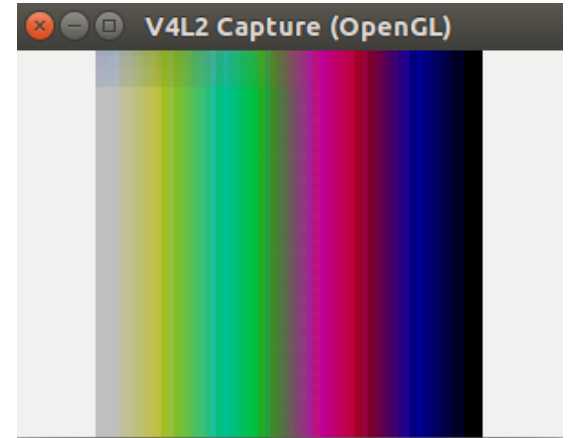
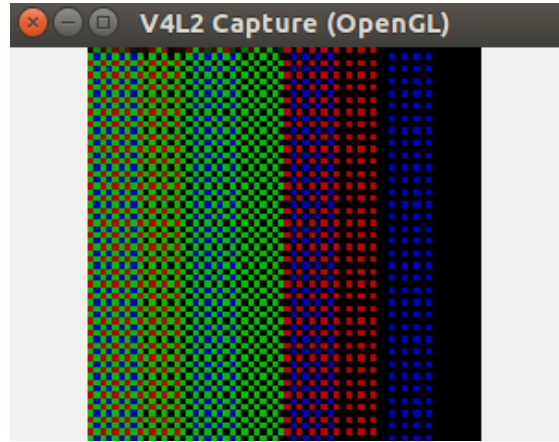
Vimc Driver: current state

- Basic set of emulation units:
 - capture (no output e.g. HDMI/S-Video)
 - sensor (generates images in different formats)
 - debayer (converts bayer to non-bayer format)
 - scaler (scaling down is not supported)
- Several optimizations and controls are still missing
- Hard-coded topology (re-compilation required)





Vimc Driver





Summary

- Classic V4L2 API
 - Vivid Driver
 - Vimc Driver
- Codecs
 - Vicodec Driver
- Vimc:
 - Submodules
 - Configfs API
 - Future work



Codecs (encoder/decoder)

- Encoder:
 - raw → compressed (e.g. RGB → H.264)
- Decoder:
 - compressed → raw (e.g. H.264 → RGB)

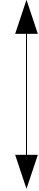


Stateful vs Stateless codecs

- Stateful codecs:
 - Driver keeps the current state
(definition of state depends on the image format)
 - Requires `/dev/video*`
- Stateless codecs:
 - Userspace keeps the state sending it in every frame
 - Requires `/dev/video*` `/dev/media*`



APP

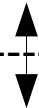


User space

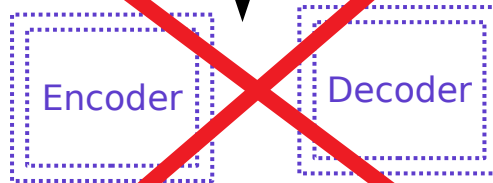
----- /dev/video* (/dev/media*) -----

The Virtual Codec Driver (Vicodec)

Kernel space



Physical device





Vicodec Driver: current state

- Relatively new (4.18 by Tom aan de Wiel)
- FWHT codec (Fast Walsh Hadamard Transform)
- Supports stateful only
- Stateless support: proposed as an Outreachy project



Use cases for virtual drivers

- Hardware is not ready / Board bring-up phase
- Convenient to make tests on a PC instead of an embedded device
- Test application in different hardware
 - e.g. Gstreamer



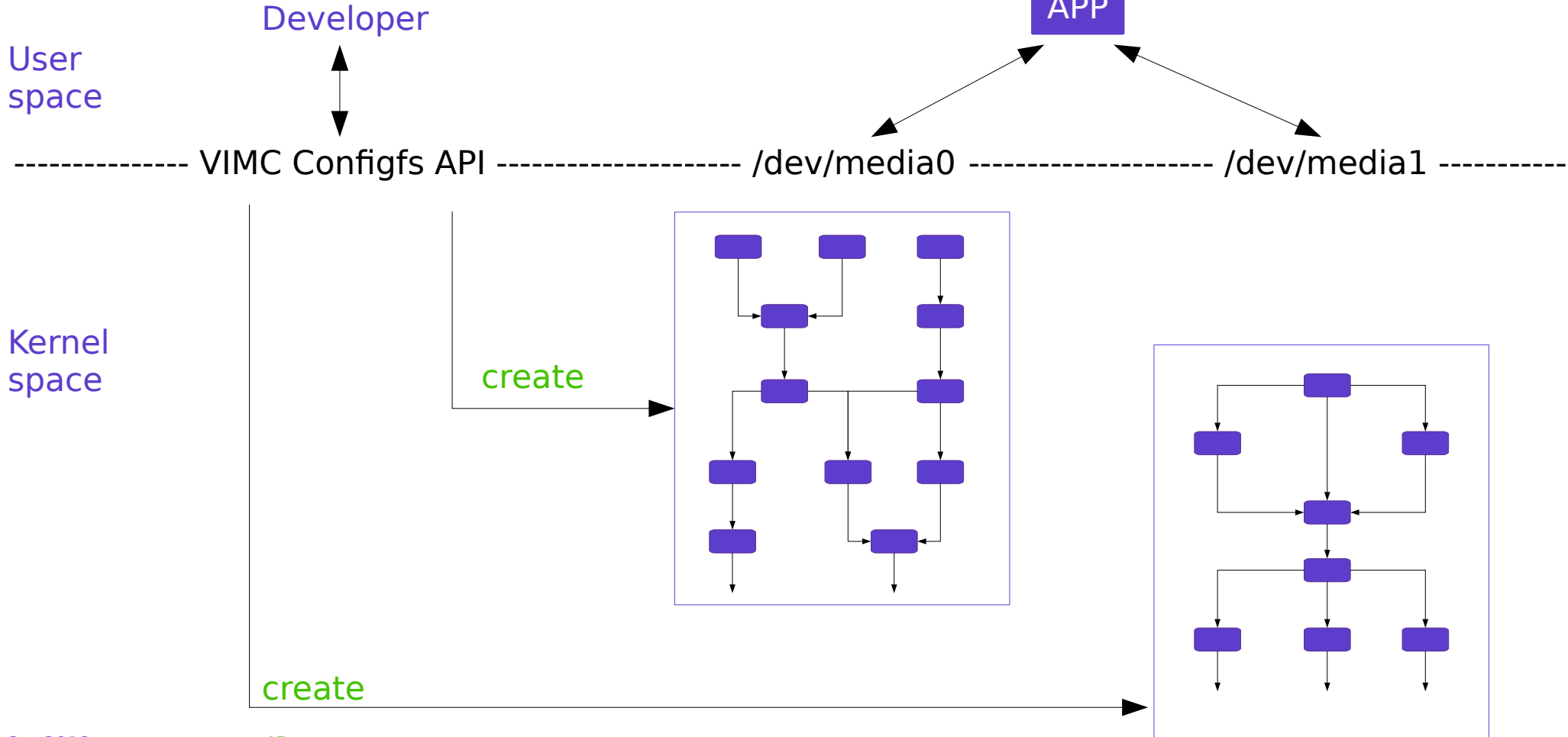
Use cases for virtual drivers (cont)

- Loop back device
 - Adding layers of effects
 - Test streams
- Test APIs
- Reference code
- Improve compliance tests



Summary

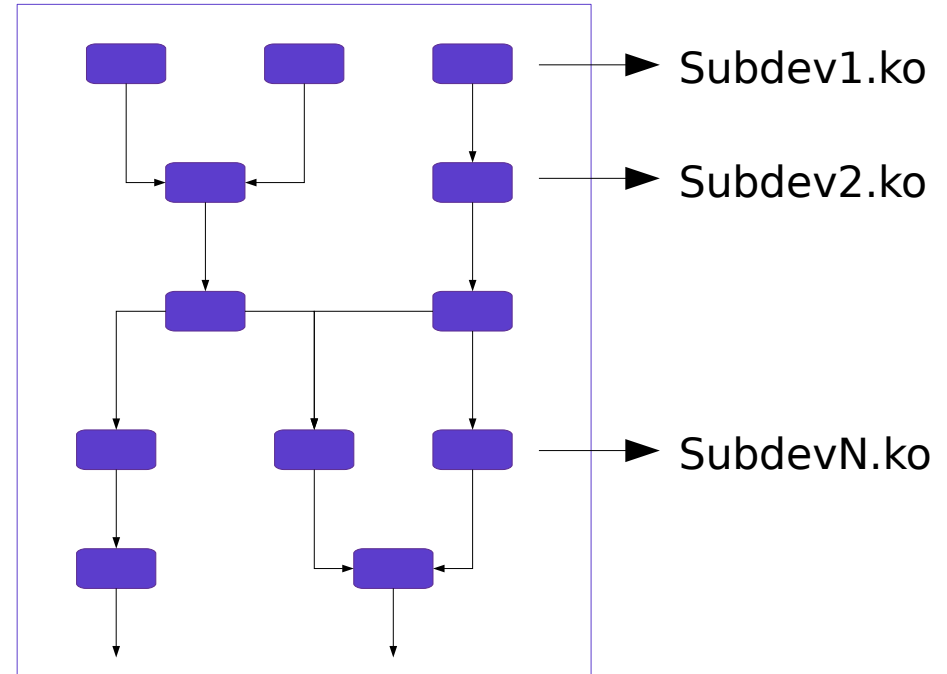
- Classic V4L2 API
 - Vivid Driver
 - Vimc Driver
 - Vicodec Driver
- Media API
- Codecs
- Vimc:
 - Submodules
 - Configfs API
 - Future work





Vimc: submodules

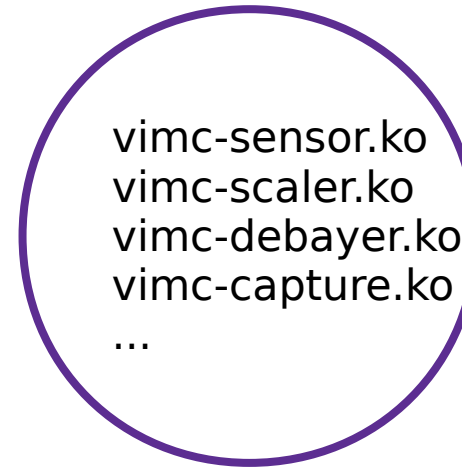
Implementation of subdevice drivers are modularized and doesn't need to alter Vimc's core code.





APP Developer

Emulate different topologies to test the app in several scenarios



Standard



Manufacturer

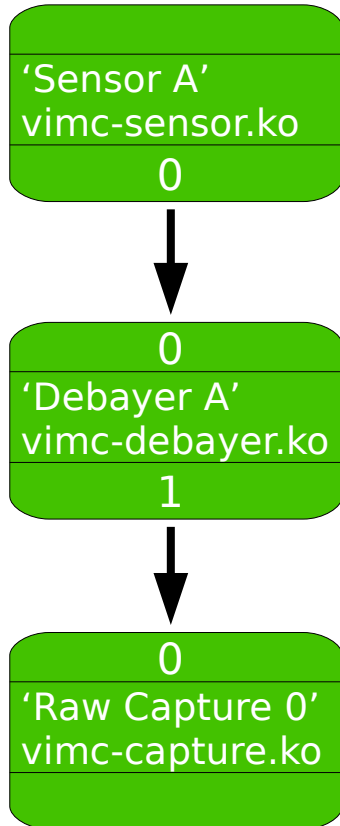


Summary

- Classic V4L2 API
 - Vivid Driver
 - Vimc Driver
 - Vicodec Driver
- Media API
- Codecs
- Vimc:
 - Submodules
 - Configfs API
 - Future work



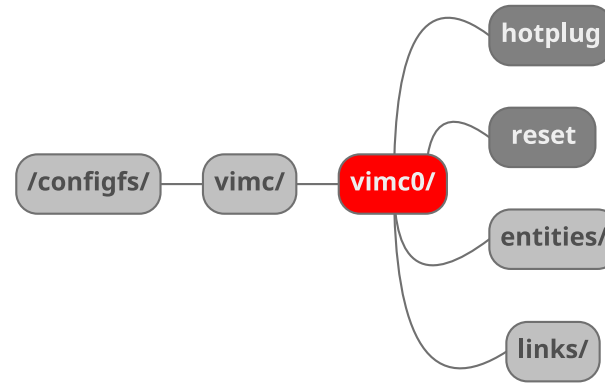
Configs API: Topology



- Entities
 - Name
 - Submodule
- Pads
 - Source
 - Sink
- Links

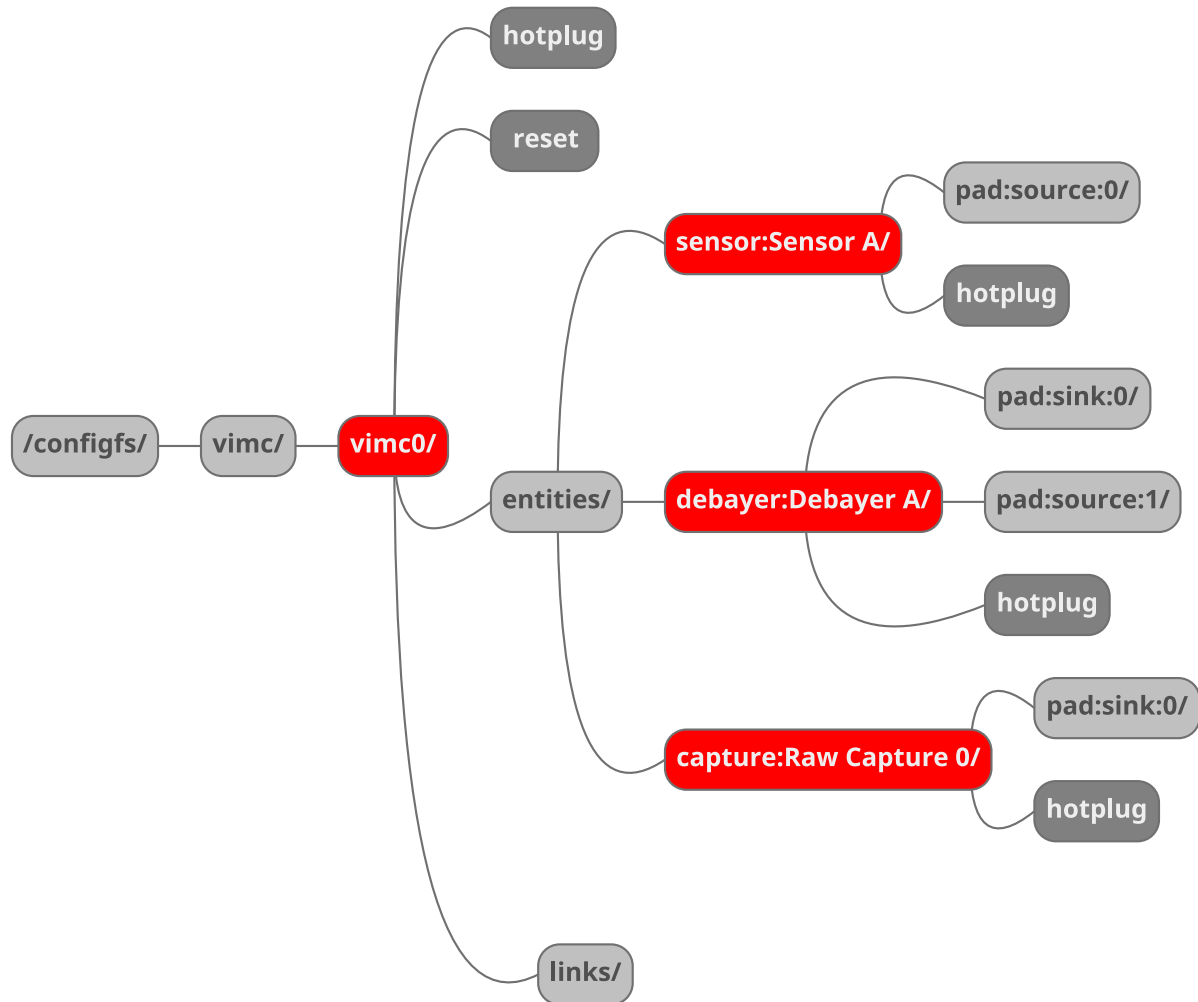
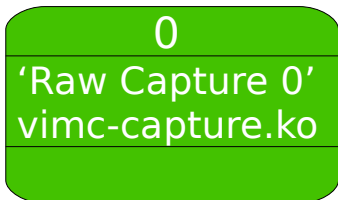
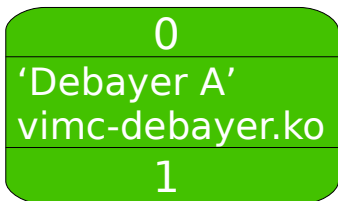
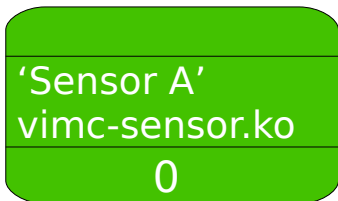


mkdir "MEDIA_NAME"



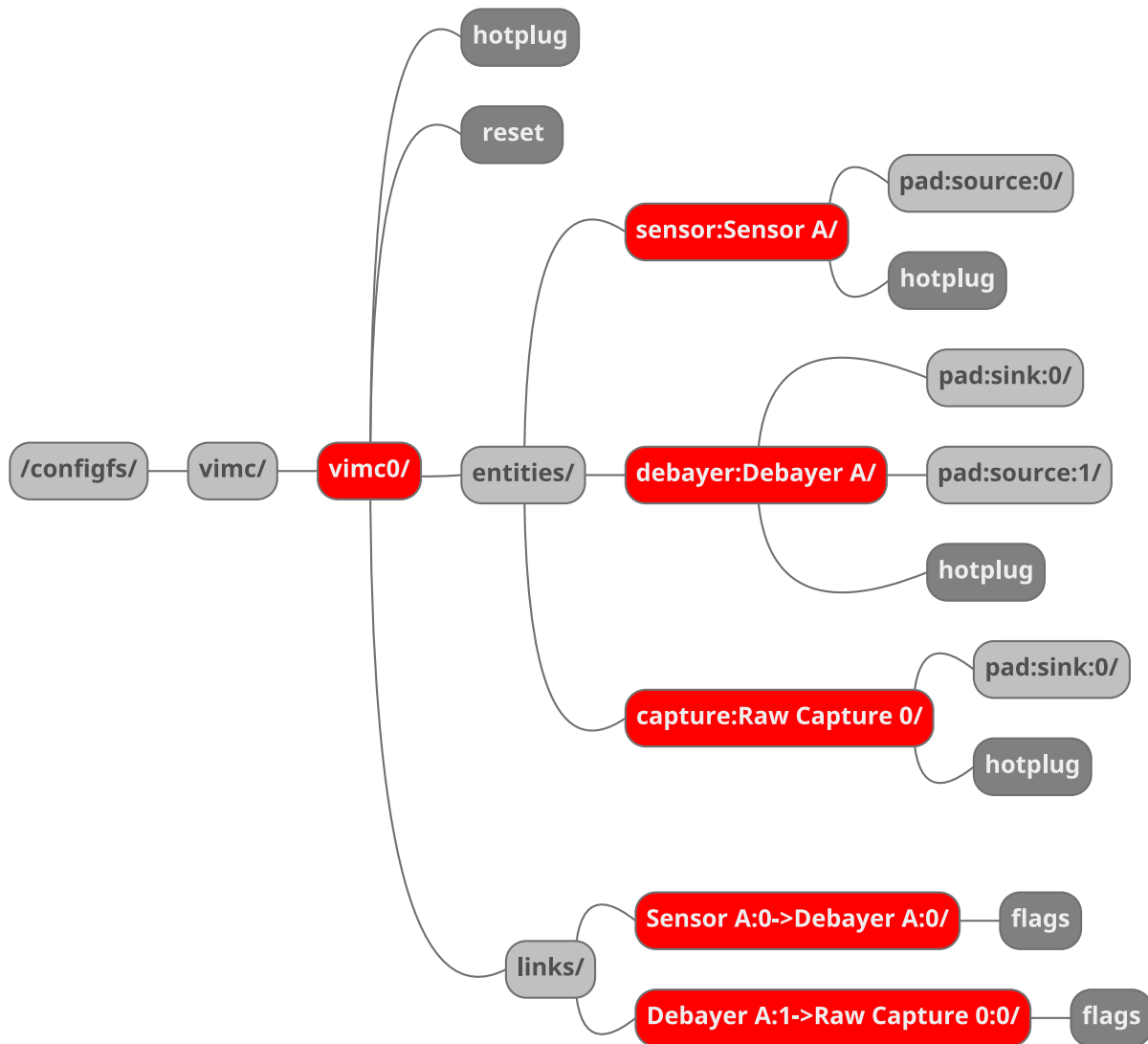
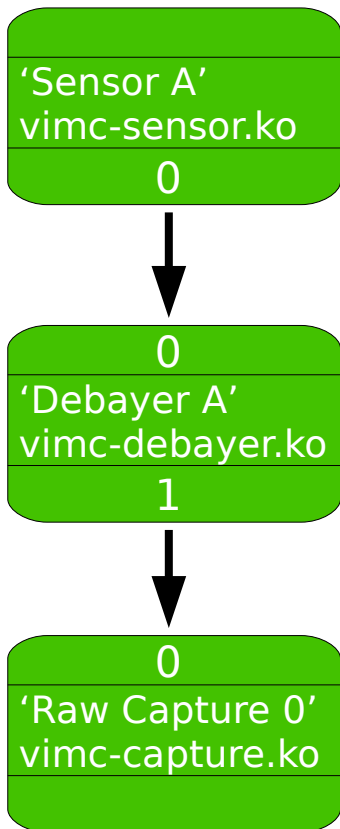


mkdir "SUBMOD:NAME"





mkdir "NAME1:PAD->NAME2:PAD"





Summary

- Classic V4L2 API
 - Vivid Driver
 - Vimc Driver
 - Vicodec Driver
- Media API
- Codecs
- Vimc:
 - Submodules
 - Configfs API
 - Future work



Vimc Driver: future work

- API in Configfs (WIP)
- More controls:
 - LKCAMP – Linux kernel study group
- Image generation directly from capture node (WIP)
 - LKCAMP
- Frame rate control

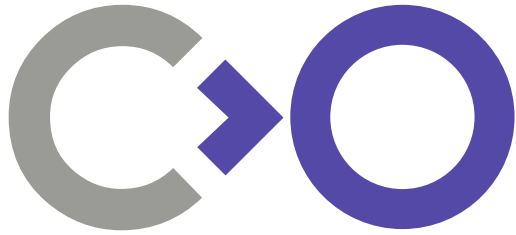


Vimc Driver: future work

- Optimizations of img processing in the pipeline
 - Calculate final format based on the full pipeline configuration
 - Multi-threaded processing
- Submodules for output: HDMI / S-Video
- More standard submodules
- Add more V4L2 mechanisms / controls / options



COLLABORA



Thank you!

Helen Koike

helen.koike@collabora.com

This work is licensed under <https://creativecommons.org/licenses/by-sa/4.0/>



Userspace tools

- v4l-utils

```
media-ctl -d /dev/media0 --print-dot > /tmp/out.dot && xdot /tmp/out.dot
```

```
media-ctl -d /dev/media0 -V "'Sensor A':0[fmt:RGB888_1X24/600x600]"
```

```
media-ctl -v -d /dev/media0 --links "'Debayer A':1->'Scaler':0 [0]"
```

- Yavta (Yet Another V4L2 Test Application)

```
yavta --format RGB24 --size 600x600 /dev/video0
```



Summary

- Classic V4L2 API
 - Vivid Driver
 - Vimc Driver
 - Vicodec Driver
- Media API
- Codecs
- Vimc:
 - Submodules
 - Configfs API
 - Future work