COLLABORA

Shifting Media app development into high gear

Using virtual drivers to speed up development

Helen Koike
helen.koike@collabora.com

Open First

COLLABORA

Summary

- Classic V4L2 API → Vivid Driver
- Media API (extension) → Vimc Driver

- Vimc:
  - Submodules
  - Current state
  - Configfs API
  - Future work

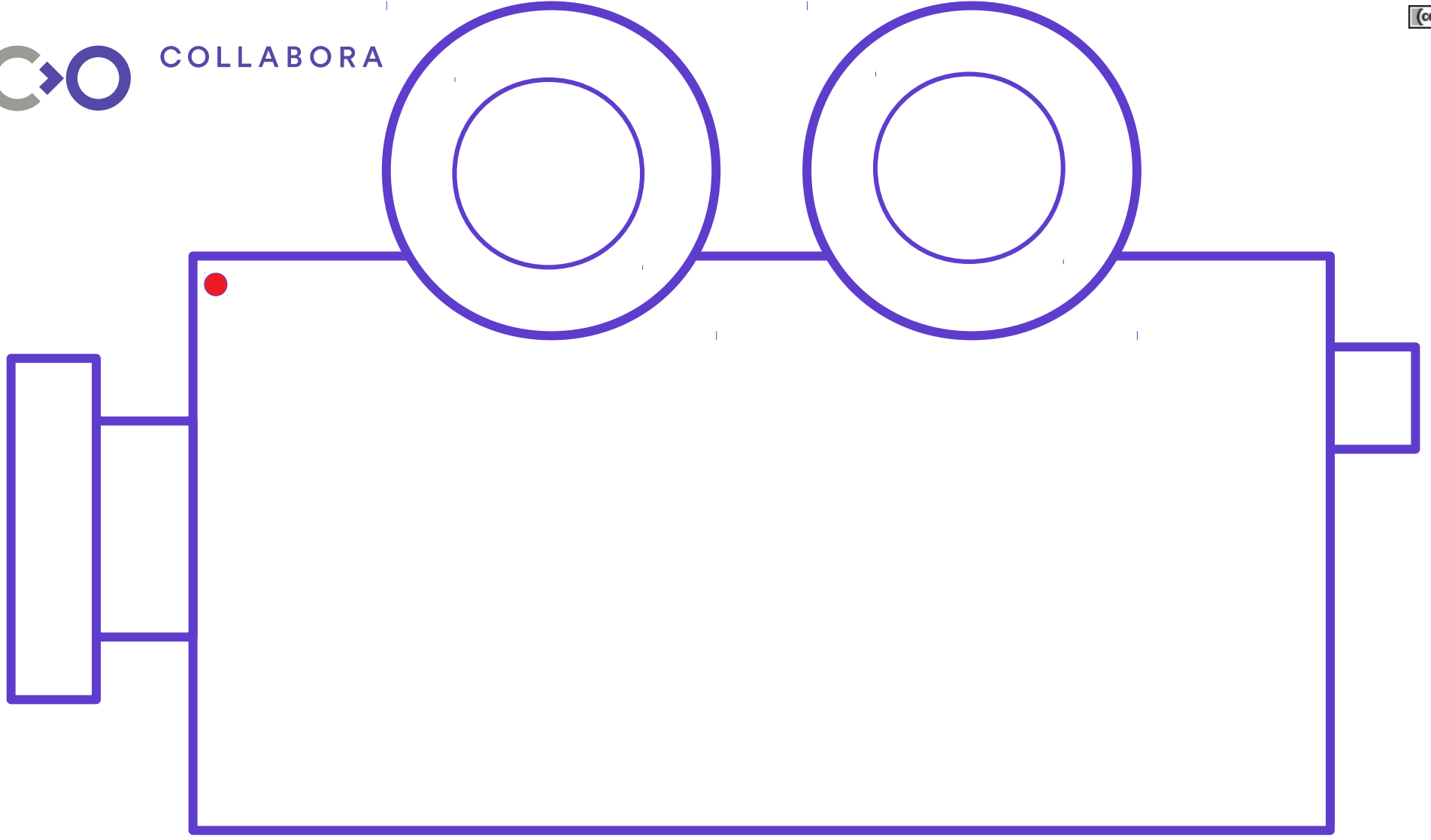
COLLABORA

Summary

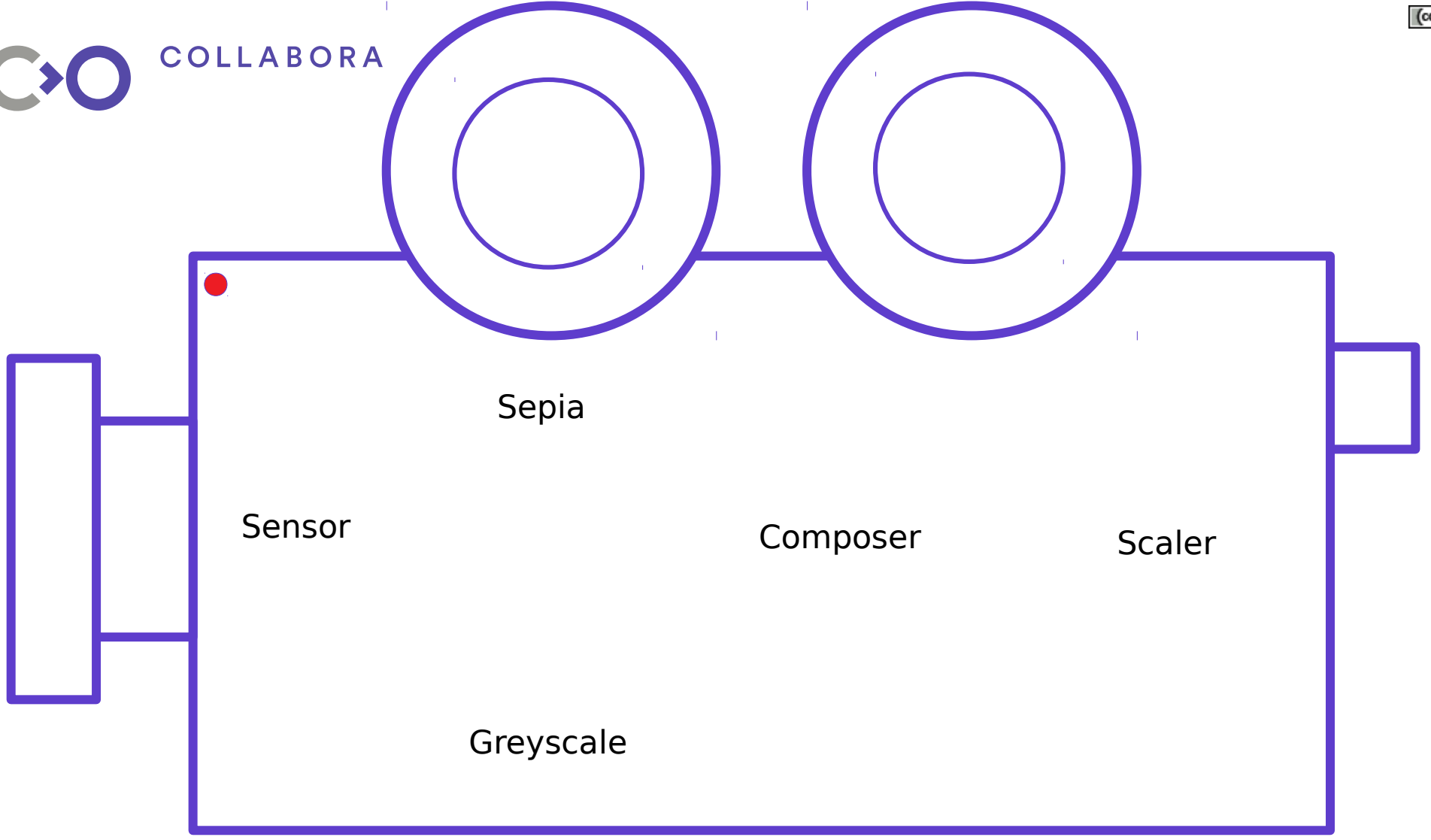
- **Classic V4L2 API → Vivid Driver**
- Media API (extension) → Vimc Driver

- Vimc:
  - Submodules
  - Current state
  - Configfs API
  - Future work

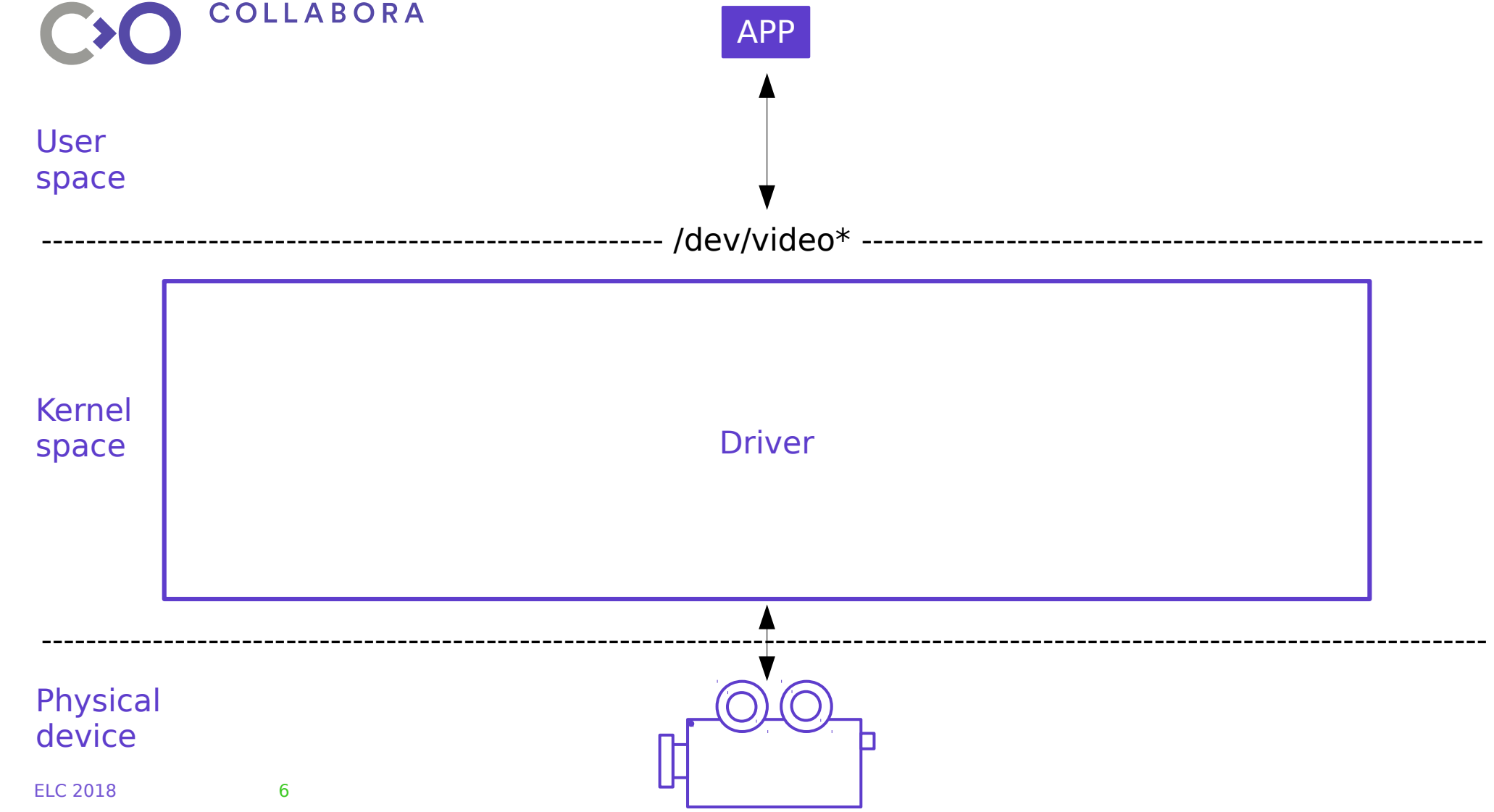
COLLABORA



COLLABORA



COLLABORA



COLLABORA

APP

Read/Write/MMAP

User space

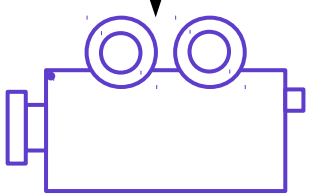
/dev/video*

Driver

Video stream

Kernel space

Physical device



COLLABORA

APP

Read/Write/MMAP

IOCTLs

User space

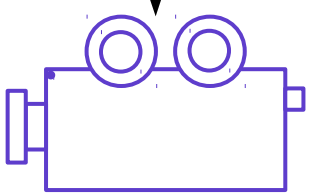
/dev/video*

Kernel space

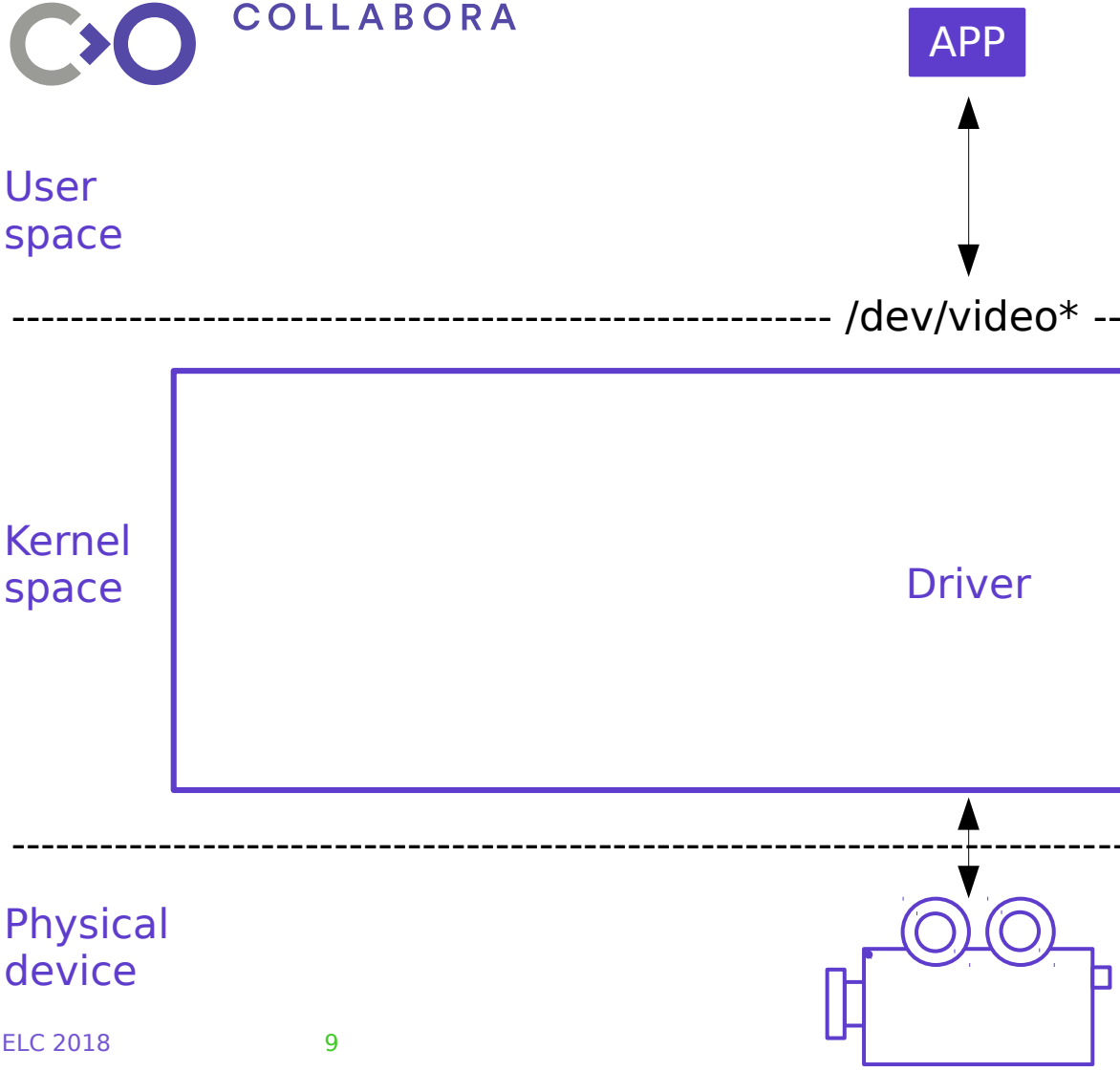
Driver

Video stream	General configs	Standard Controls	Custom Controls
	* Img fmt * Buffers * Video std * Frame rate . .	* Contrast * Brightness * Gamma . . .	* DRV DEF 1 * DRV DEF 2 * DRV DEF 3 . . .

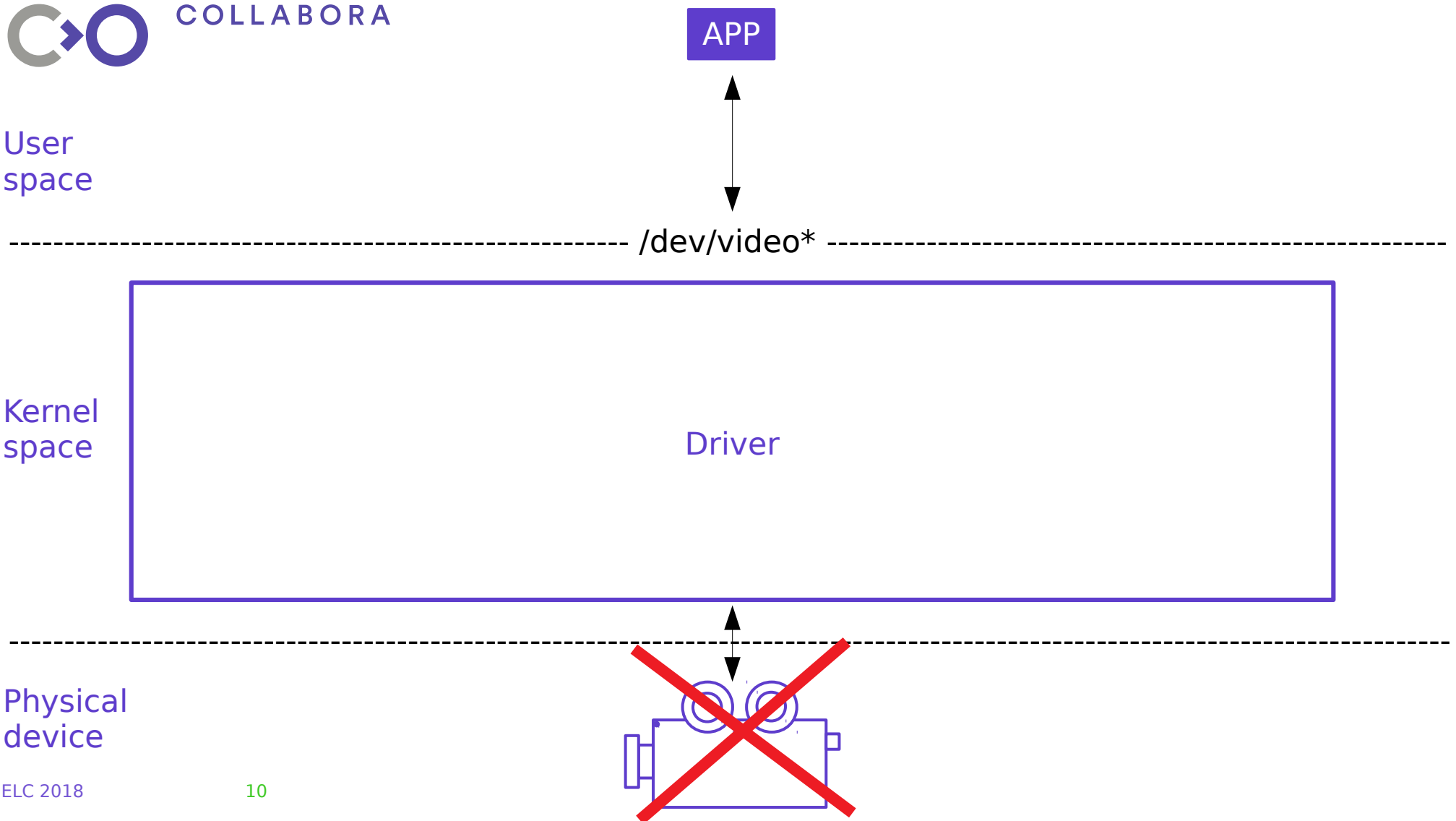
Physical device



COLLABORA



COLLABORA



COLLABORA

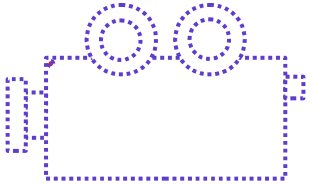
APP

User space

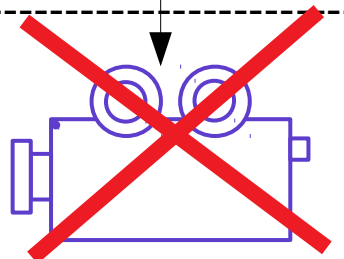
/dev/video*

Kernel space

The Virtual Video Test Driver (VIVID)



Physical device



COLLABORA

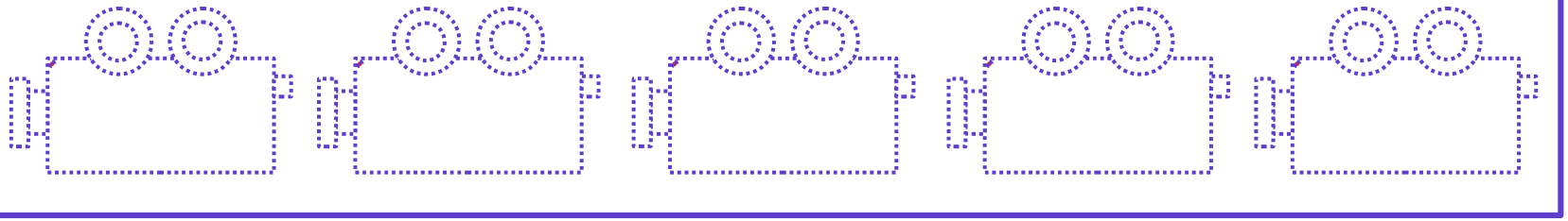
APP

User space

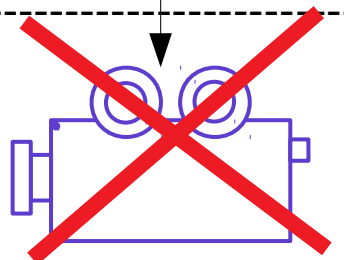
/dev/video*

Kernel space

The Virtual Video Test Driver (VIVID)

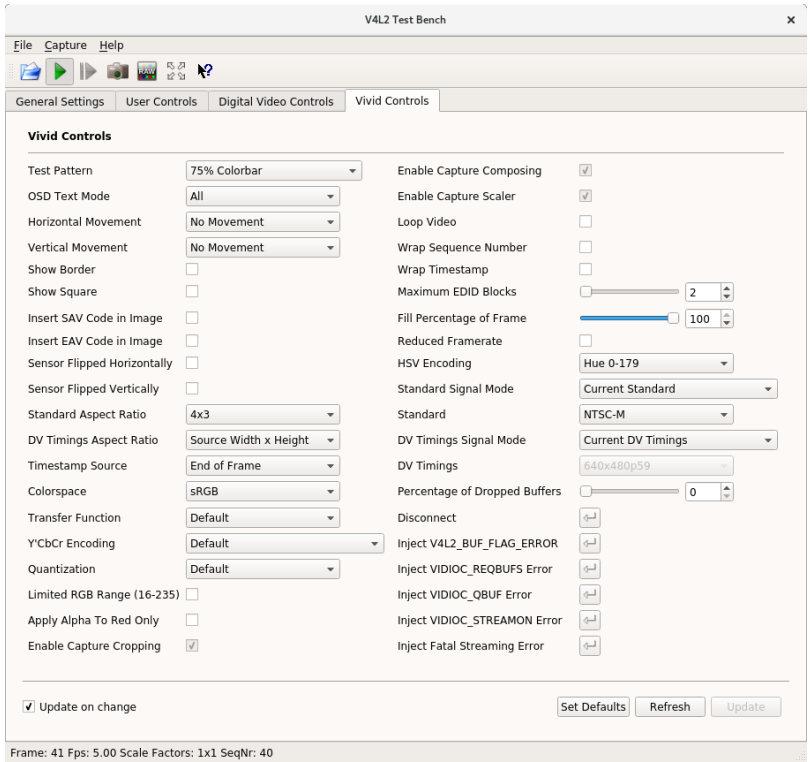


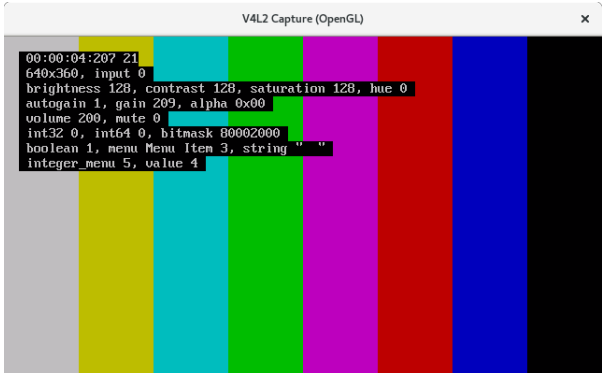
Physical device

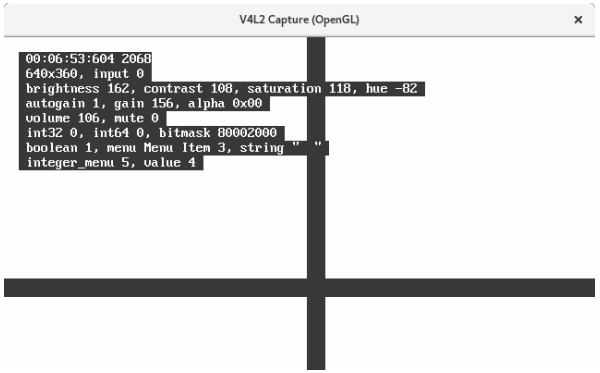


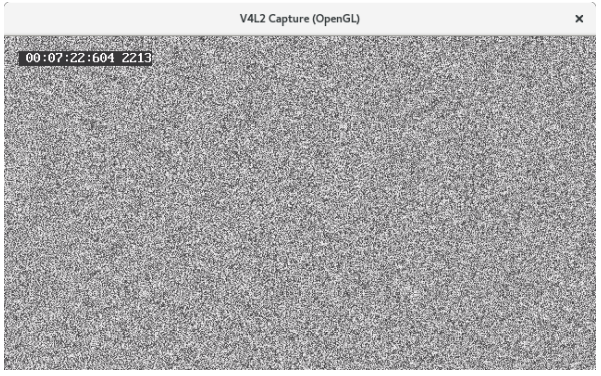
COLLABORA

Vivid driver









COLLABORA

Summary

- Classic V4L2 API → Vivid Driver
- **Media API (extension) → Vimc Driver**

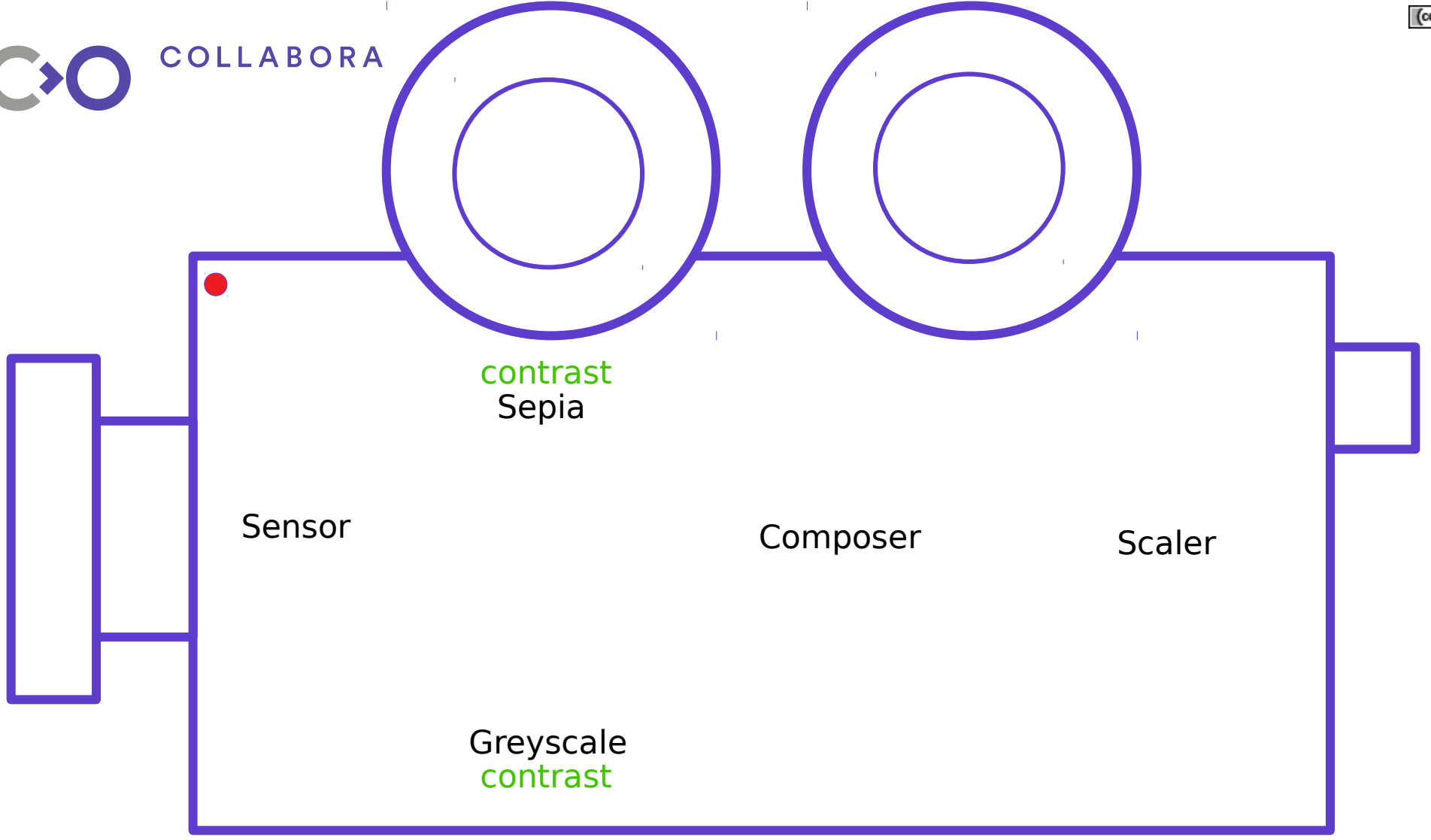
- Vimc:
  - Submodules
  - Current state
  - Configfs API
  - Future work

COLLABORA

Media API Motivation

- Embedded systems: great variety of devices
- Hardware complexity
- Similar configuration for different entities

COLLABORA



COLLABORA

APP

Read/Write/MMAP

IOCTLs

User space

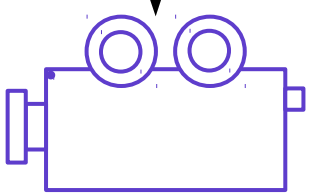
/dev/video*

Kernel space

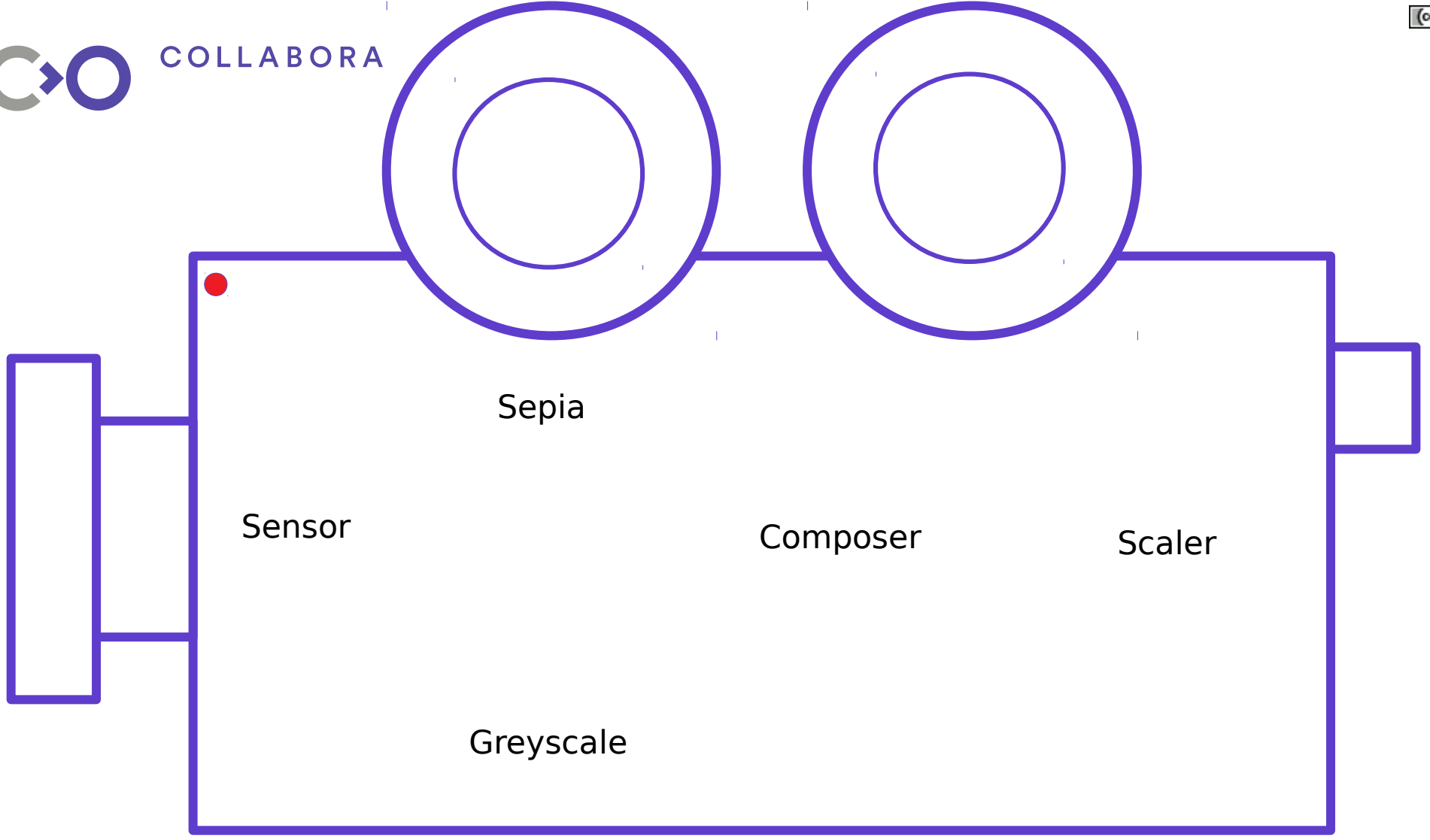
Driver

Video stream	General configs	Standard Controls	Custom Controls
	* Img fmt * Buffers * Video std * Frame rate . .	* **Contrast** * Brightness * Gamma . . .	* DRV DEF 1 * DRV DEF 2 * DRV DEF 3 . . .

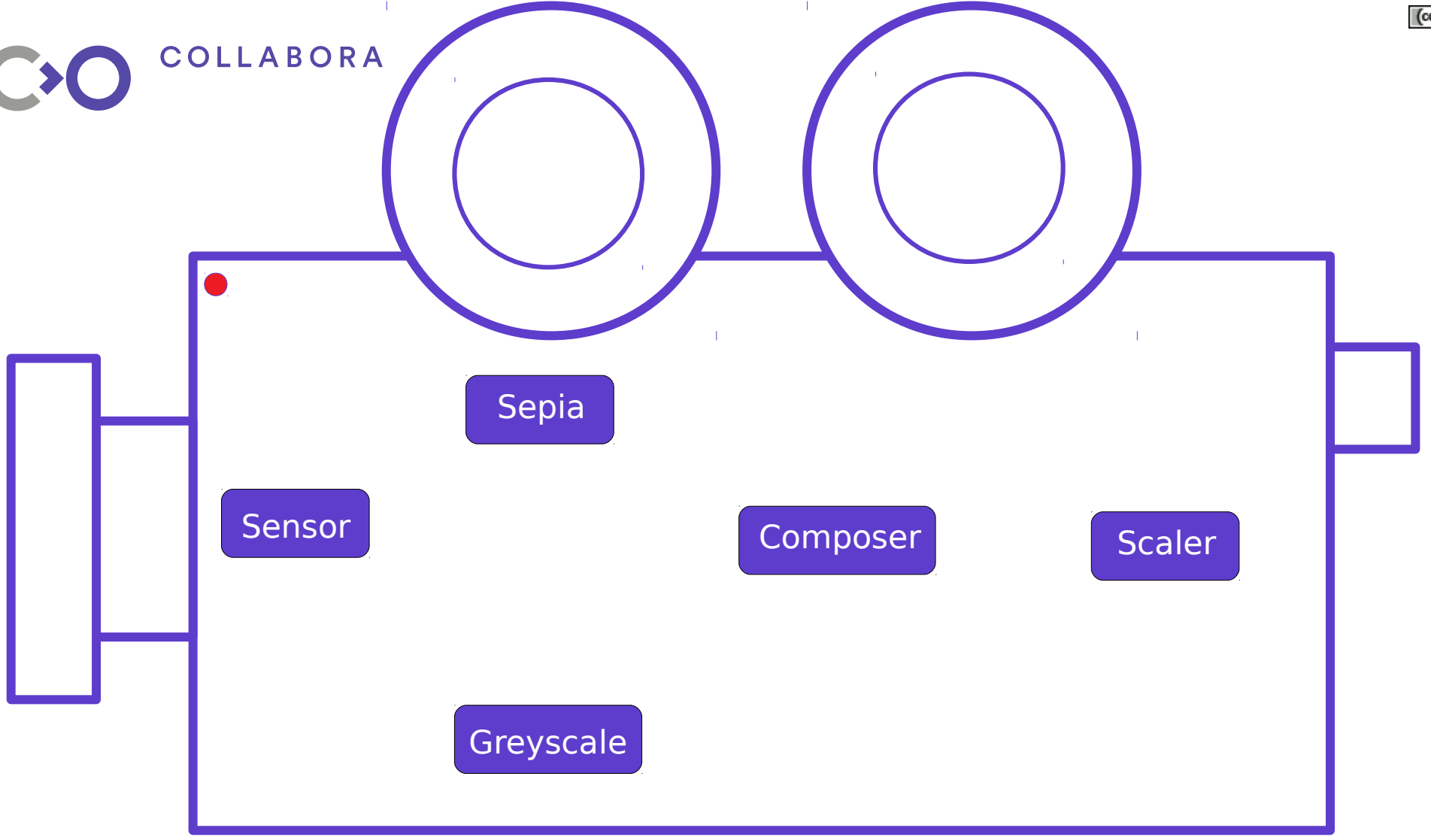
Physical device



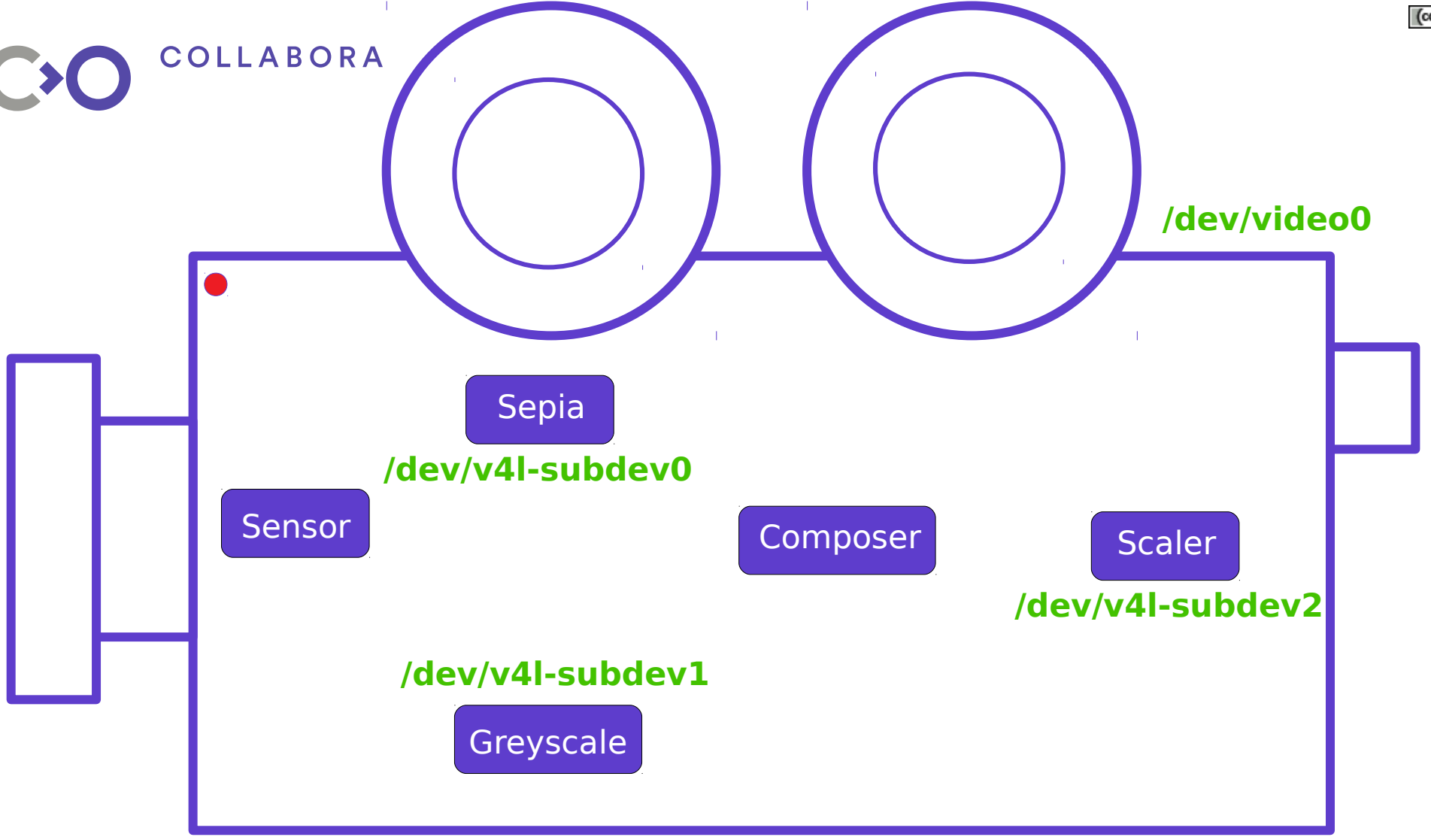
COLLABORA



COLLABORA

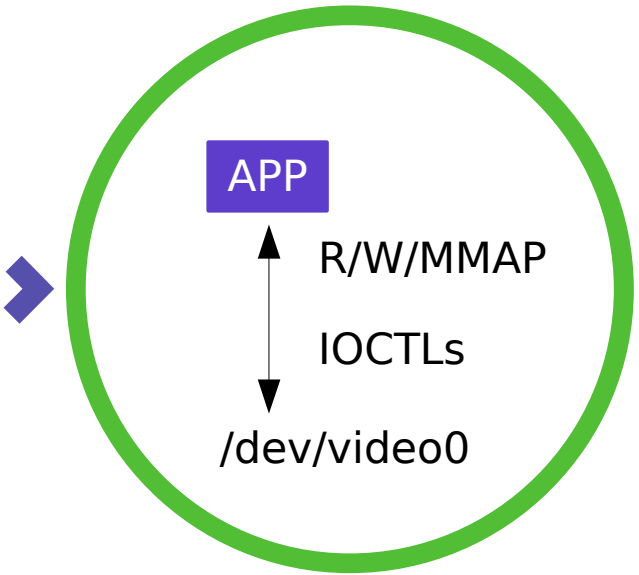


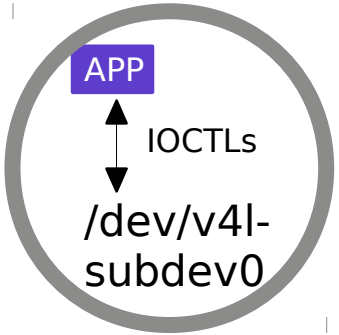
COLLABORA



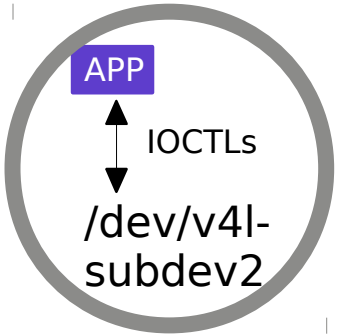
COLLABORA

Subdev API

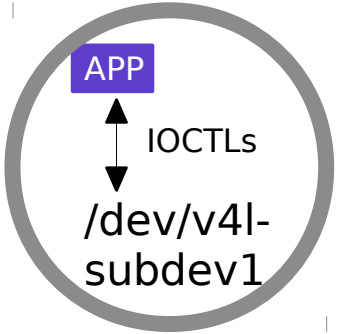




contrast
Sepia



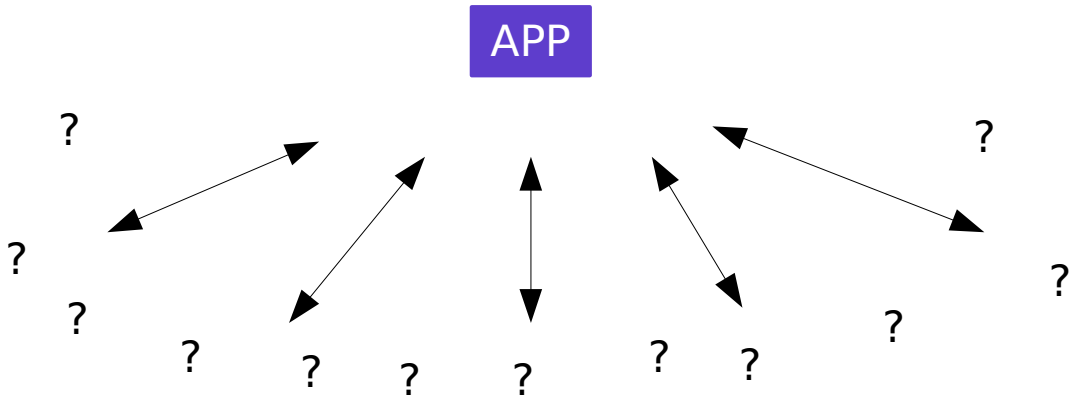
Scaler



contrast
Greyscale

COLLABORA

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



/dev/v4l-subdev0

/dev/v4l-subdev14

/dev/video0

/dev/v4l-subdev31

/dev/video42

/dev/v4l-subdev98

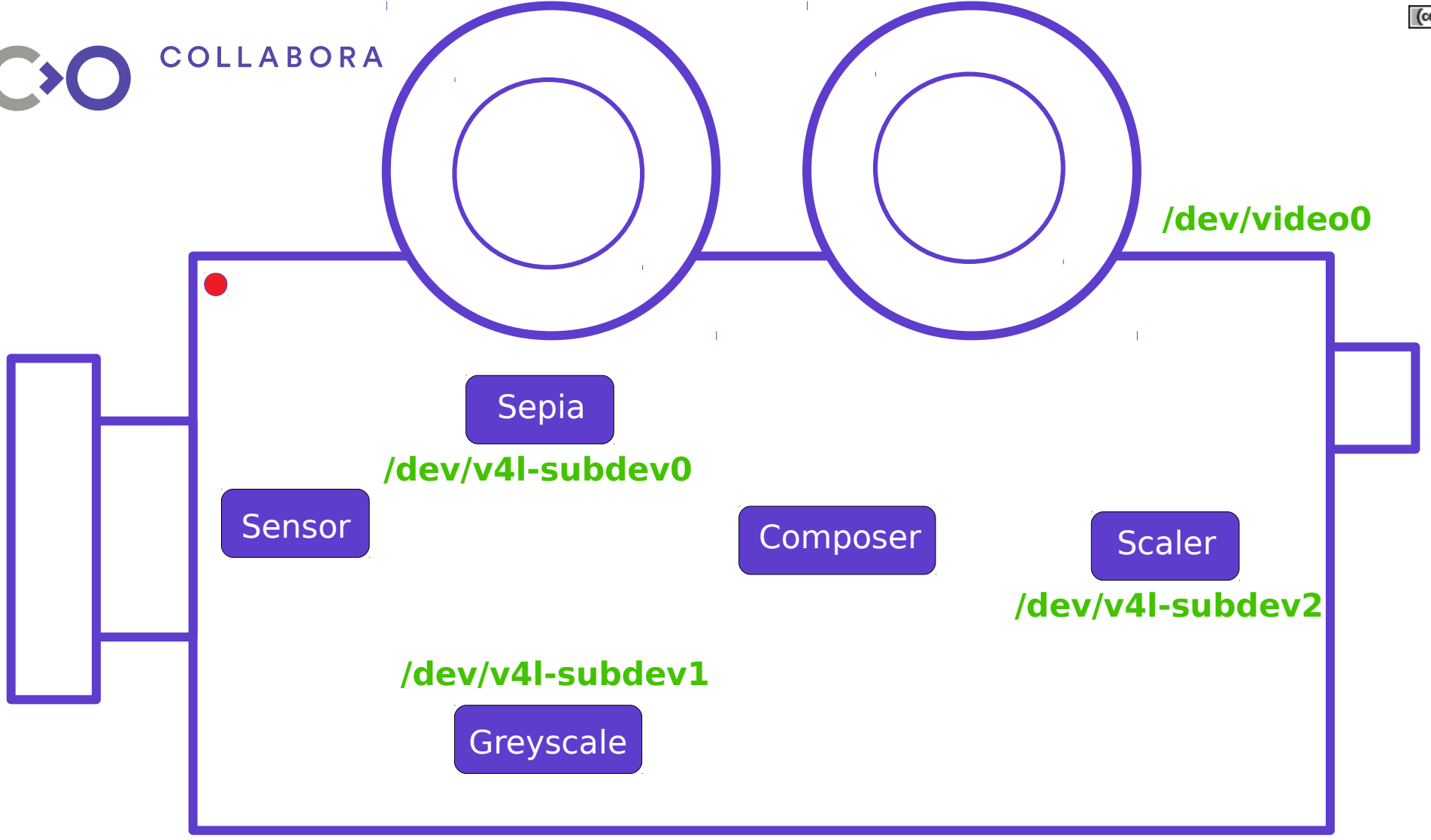
/dev/v4l-subdev6

/dev/v4l-subdev11

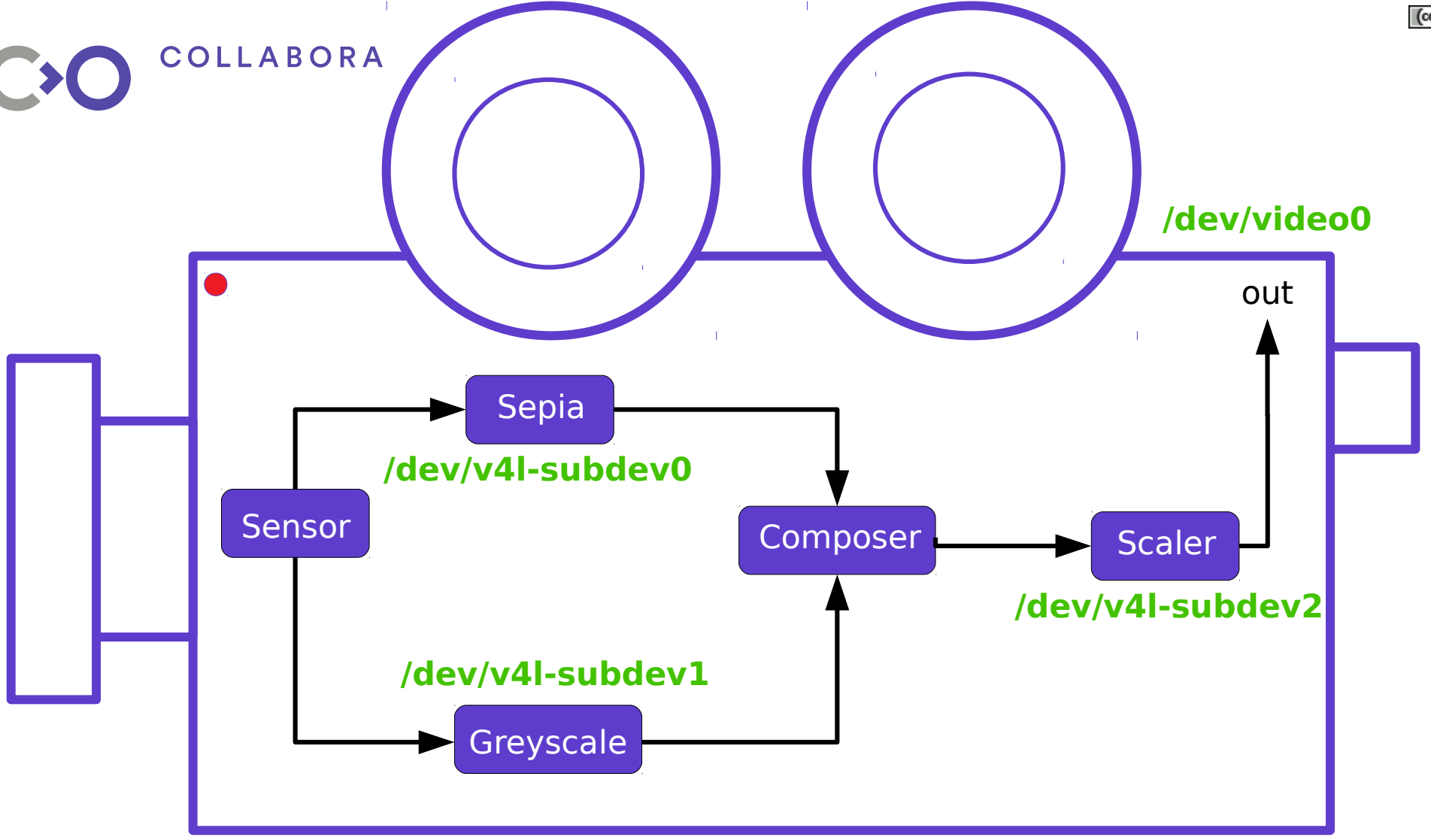
/dev/video2

/dev/v4l-subdev21

COLLABORA

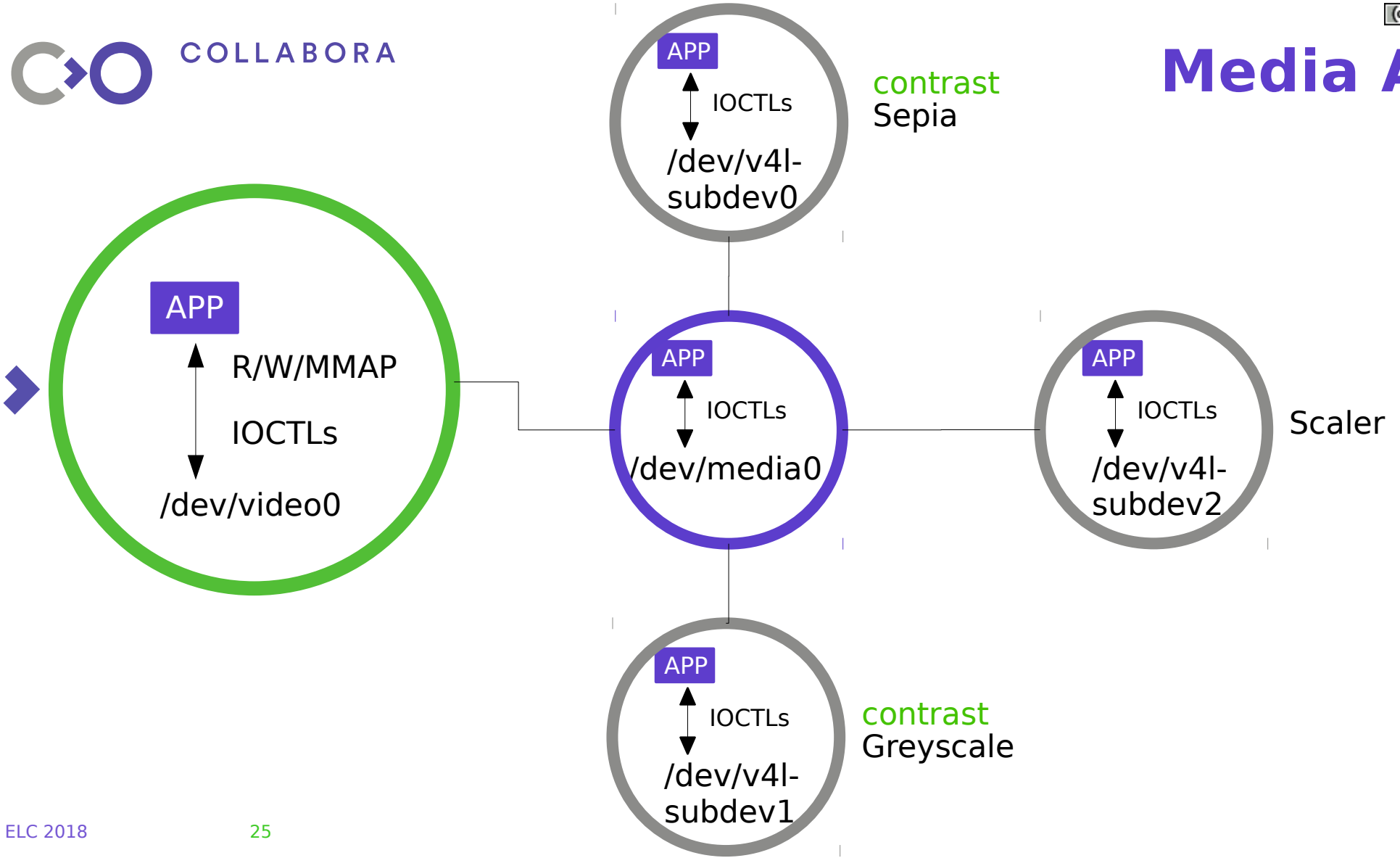


COLLABORA

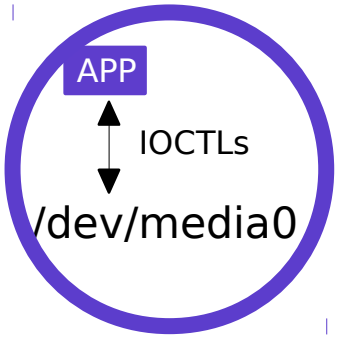


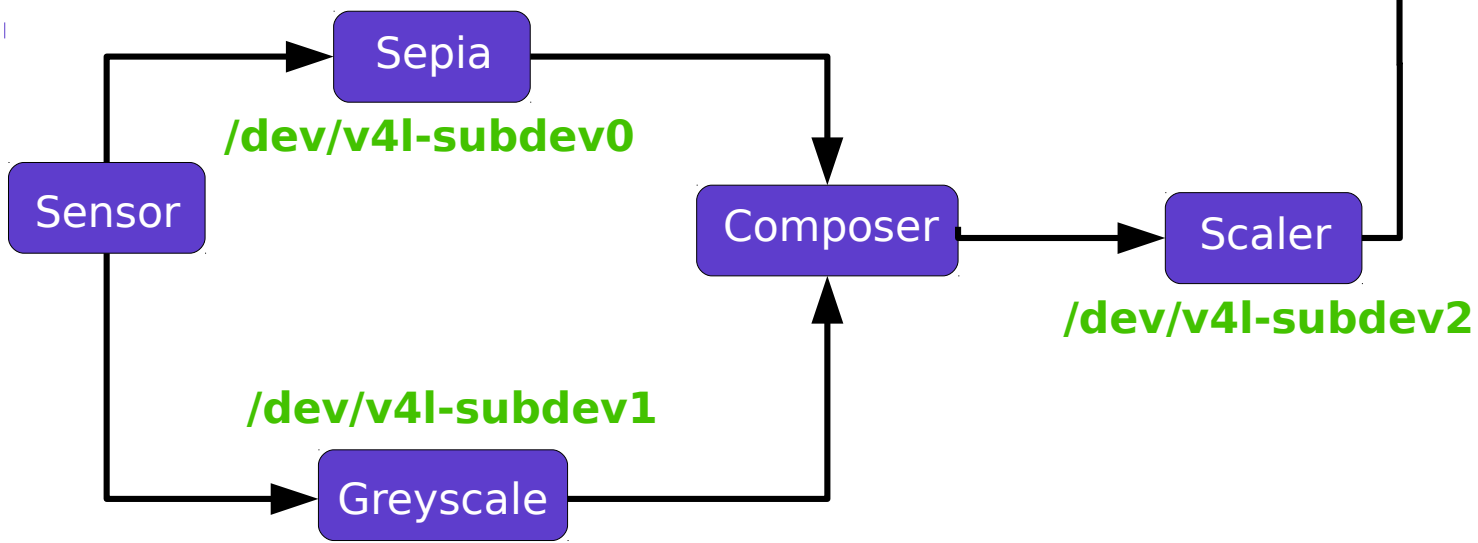
COLLABORA

Media API

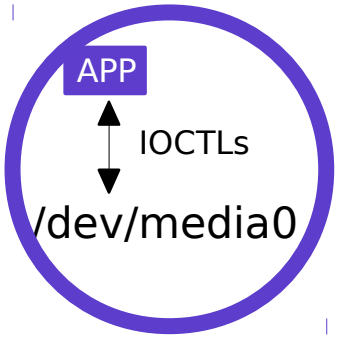


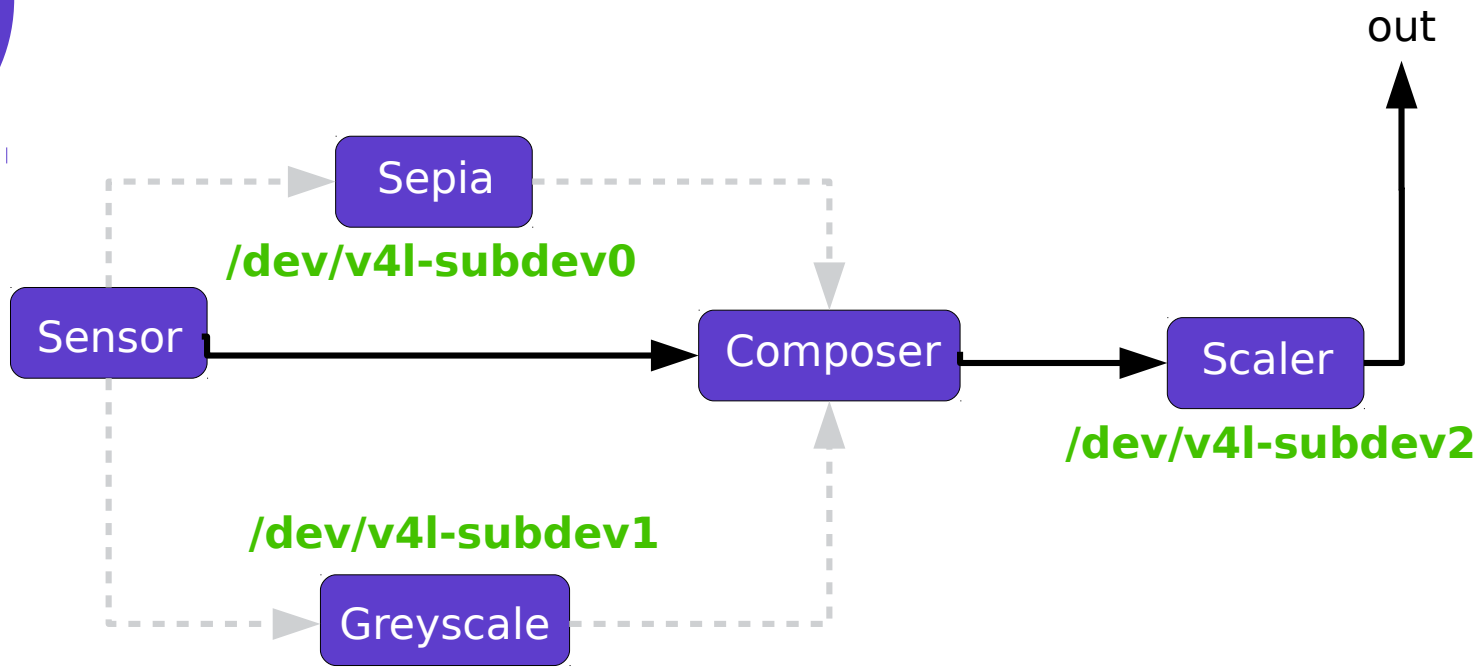
COLLABORA



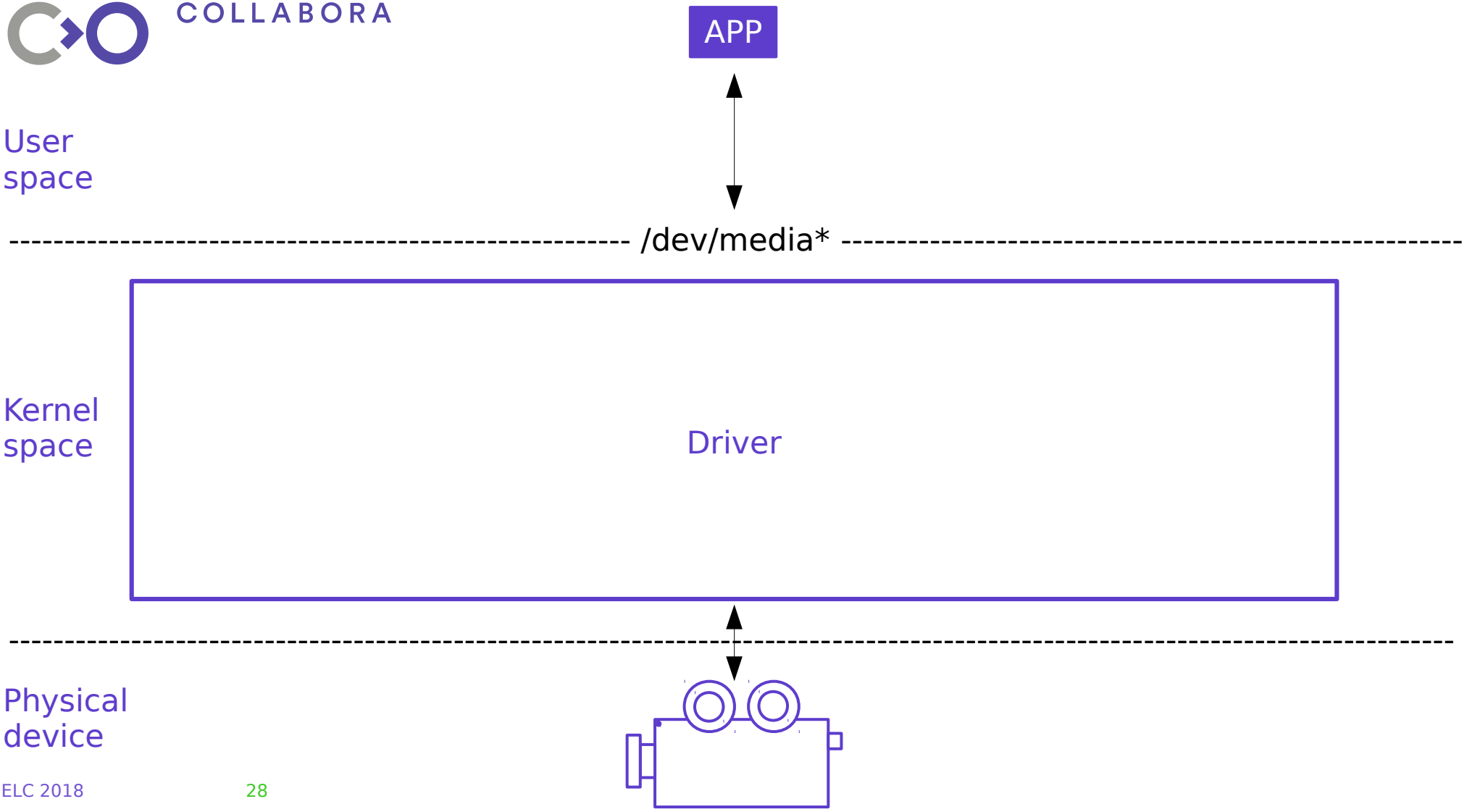


COLLABORA

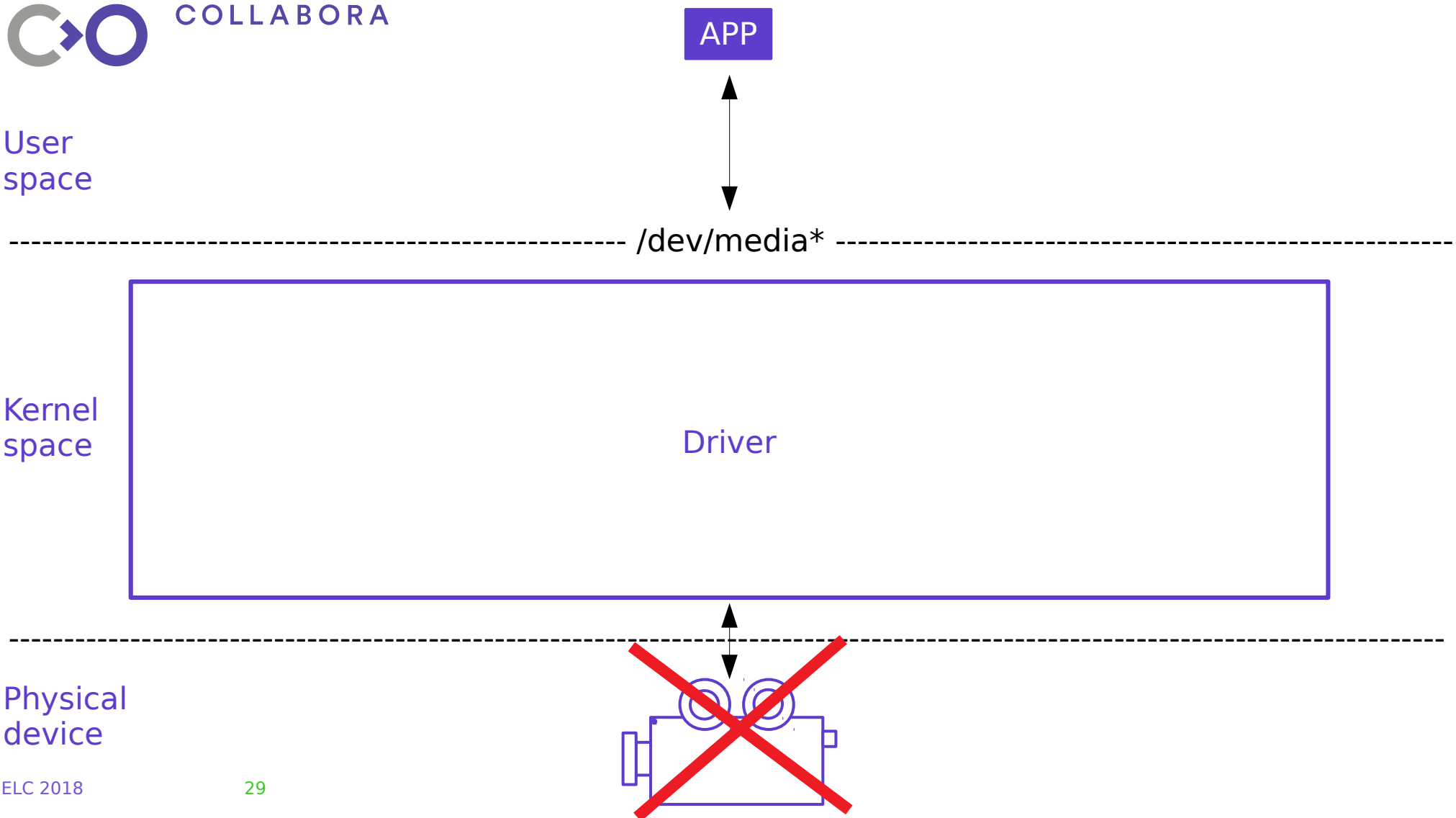




COLLABORA



COLLABORA



COLLABORA

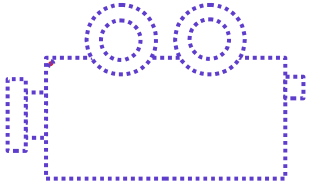
APP

User space

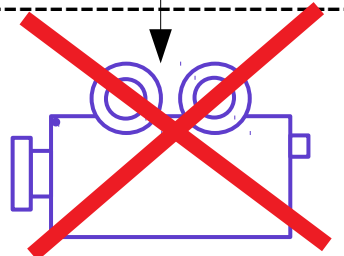
/dev/media*

Kernel space

The Virtual Media Controller Driver (VIMC)



Physical device



COLLABORA

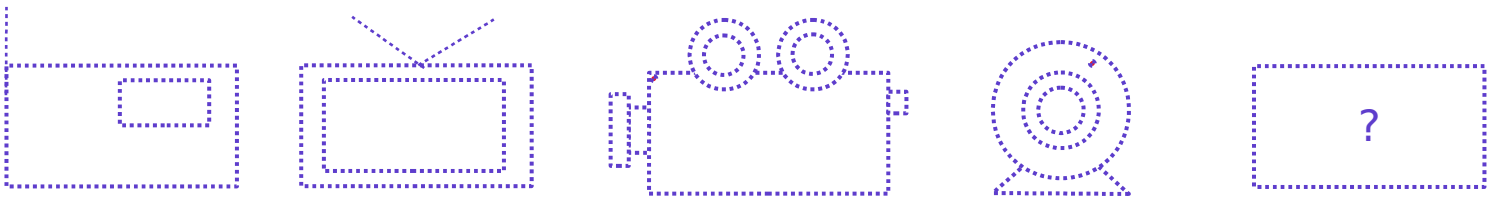
APP

User space

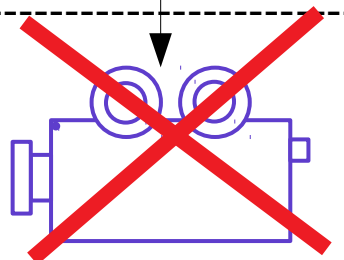
/dev/media*

Kernel space

The Virtual Media Controller Driver (VIMC)



Physical device



COLLABORA

Summary

- Classic V4L2 API → Vivid Driver
- Media API (extension) → Vimc Driver

- **Vimc:**
  - **Submodules**
  - Current state
  - Configfs API
  - Future work

COLLABORA

User space

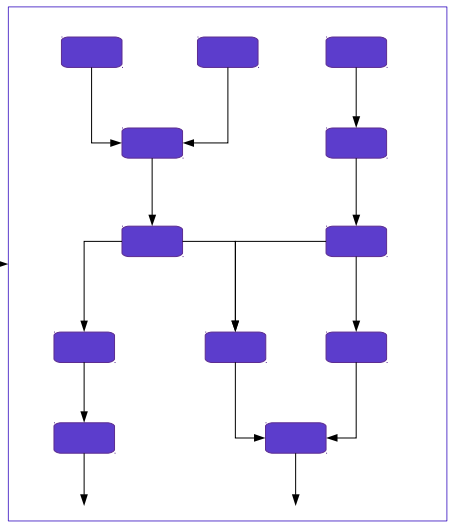
Developer

APP

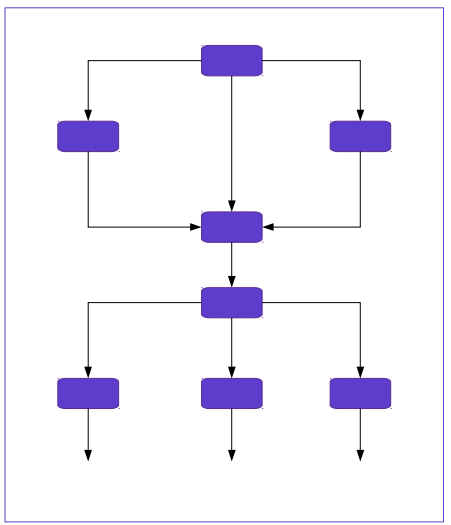
---------- VIMC Configfs API ---------- /dev/media0 ---------- /dev/media1 ----------

Kernel space

create



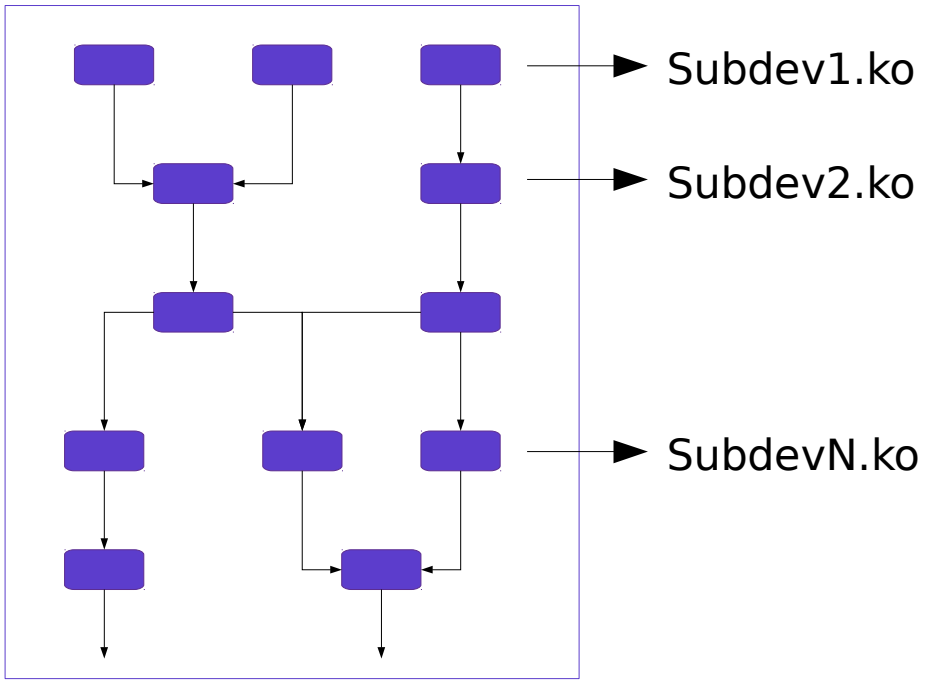
create



COLLABORA

Vimc: submodules

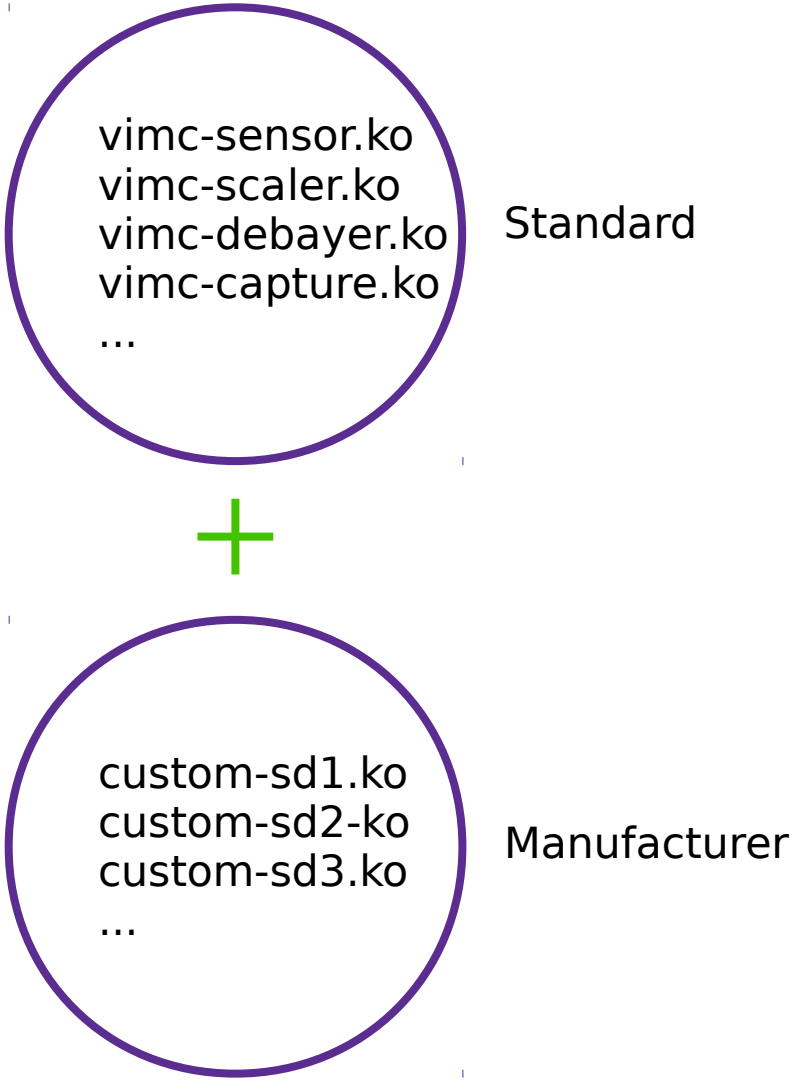
Implementation of subdevice drivers are modularized and doesn't need to alter Vimc's core code.



COLLABORA

APP Developer

Emulate different topologies to test the app in several scenarios



COLLABORA

Summary

- Classic V4L2 API → Vivid Driver
- Media API (extension) → Vimc Driver

- **Vimc:**
  - Submodules
  - **Current state**
  - Configfs API
  - Future work

COLLABORA

Vimc Driver

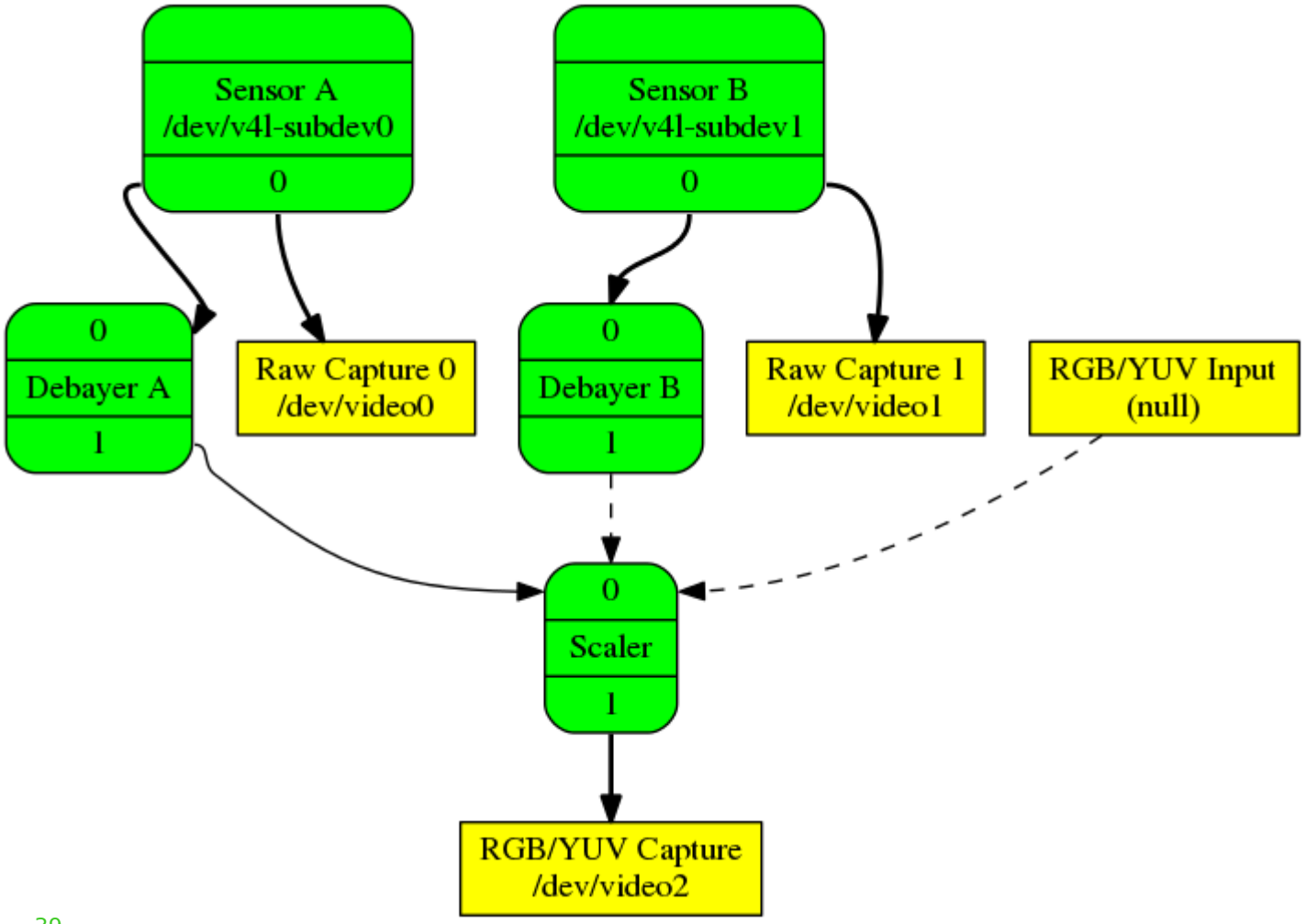
- Proposed by Laurent Pinchart for Outreachy in 2015
- Merged in Kernel 4.12
- Moving slowly

COLLABORA

Vimc Driver: current state

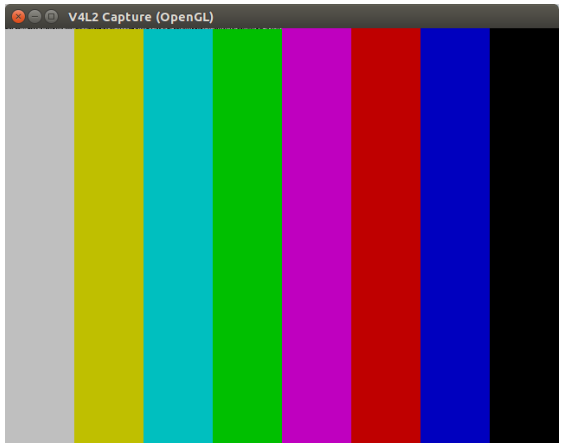
- Basic standard submodules:
  - vimc-capture.ko
  - vimc-sensor.ko
  - vimc-debayer.ko
  - vimc-scaler.ko
- Hard-coded topology (re-compilation required)

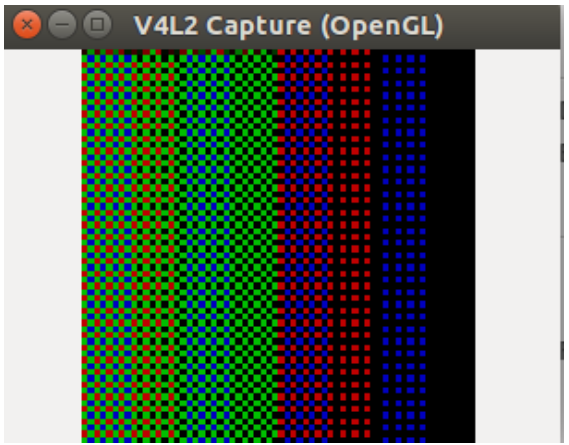
COLLABORA

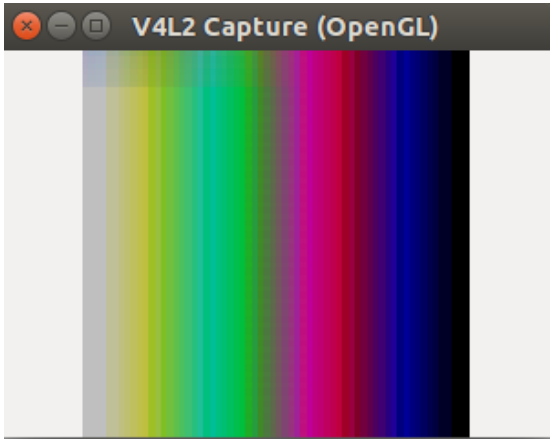


COLLABORA

Vimc Driver







COLLABORA

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**

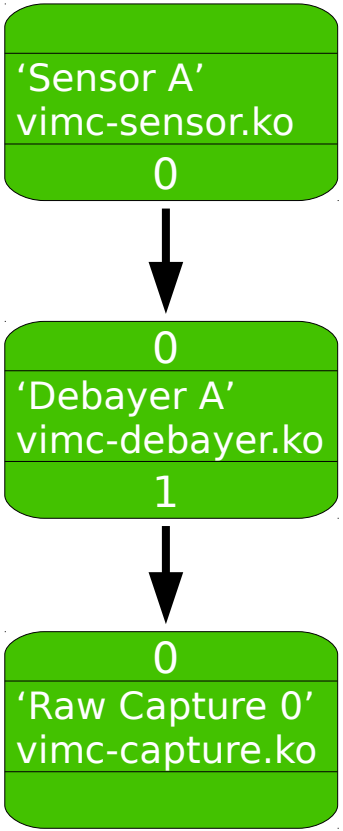
COLLABORA

Summary

- Classic V4L2 API → Vivid Driver
- Media API (extension) → Vimc Driver

- **Vimc:**
  - Submodules
  - Current state
  - **Configfs API**
  - Future work

COLLABORA

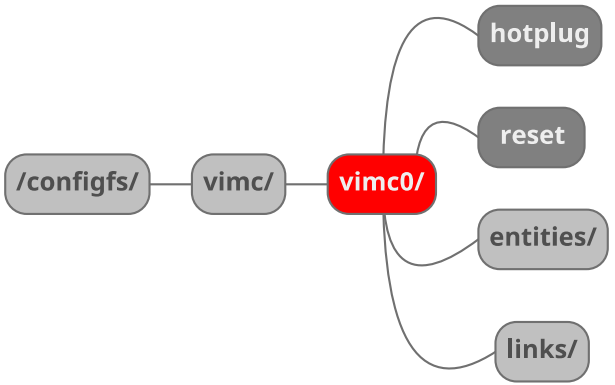


Configfs API: Topology

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

COLLABORA

mkdir “MEDIA_NAME”



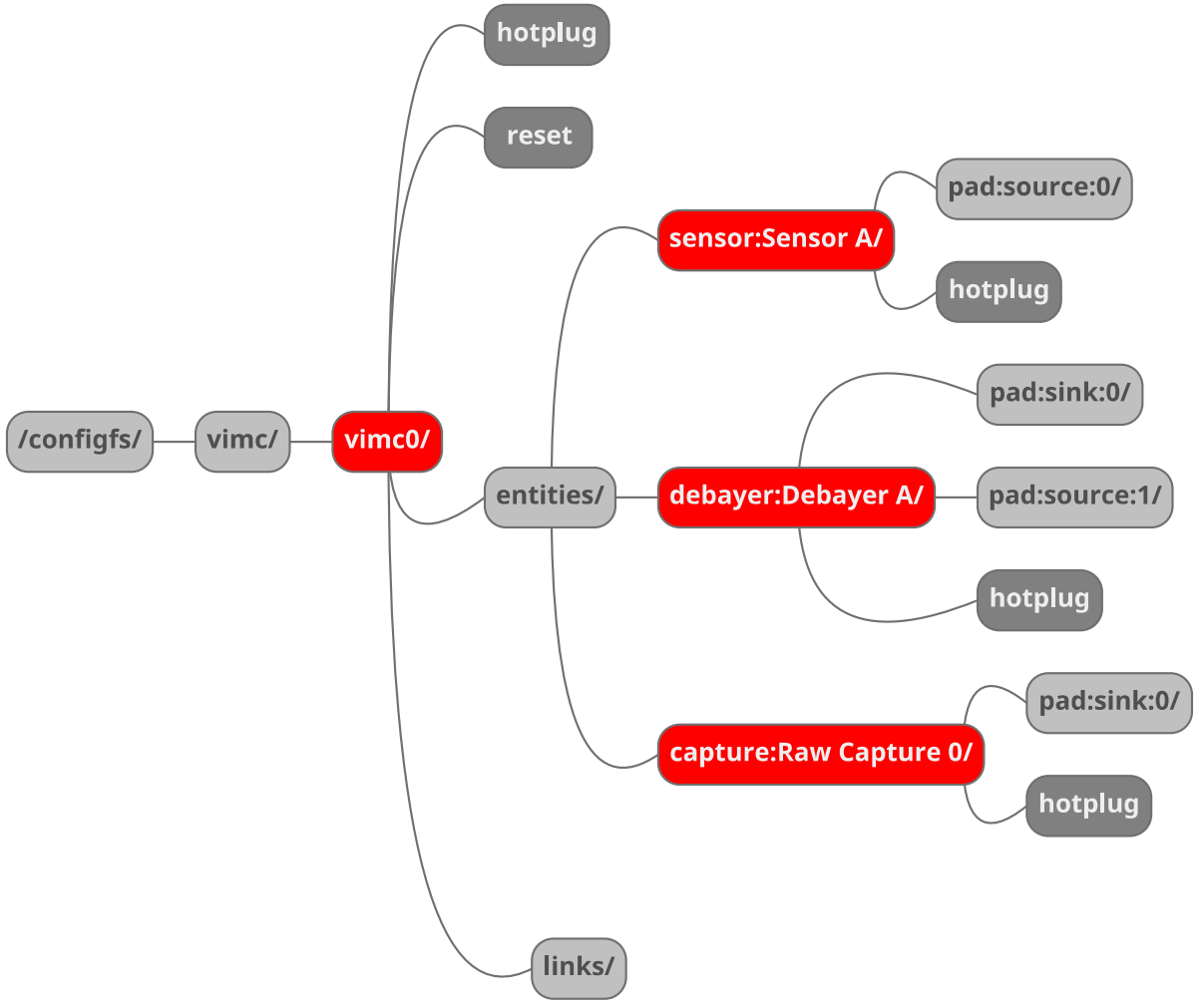
COLLABORA

mkdir “SUBMOD:NAME”

‘Sensor A’
vimc-sensor.ko
0

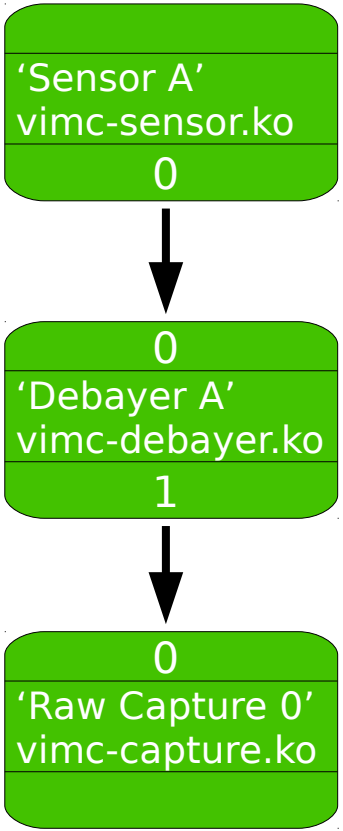
0
‘Debayer A’
vimc-debayer.ko
1

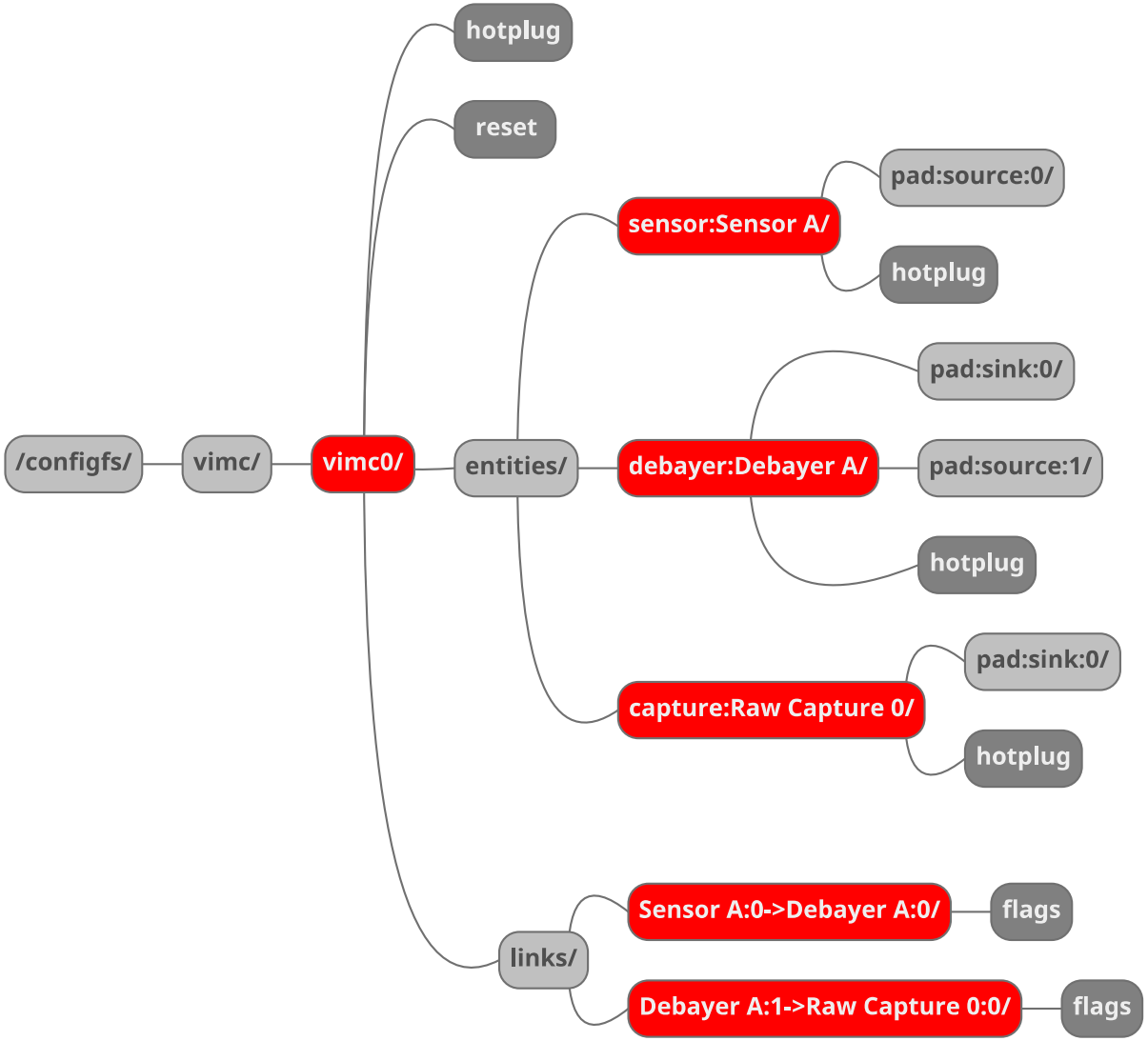
0
‘Raw Capture 0’
vimc-capture.ko



COLLABORA

mkdir “NAME1:PAD->NAME2:PAD”





COLLABORA

Summary

- Classic V4L2 API → Vivid Driver
- Media API (extension) → Vimc Driver

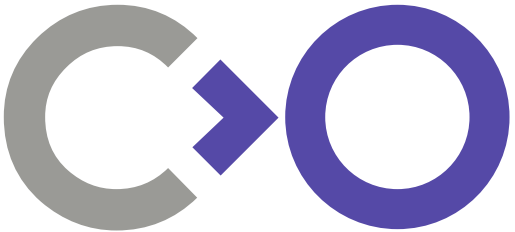
- **Vimc:**
  - Submodules
  - Current state
  - Configfs API
  - **Future work**

COLLABORA

Vimc Driver: future work

- API in Configfs (WIP)
- vimc-input.ko (WIP)
- Optimizations of img processing in the pipeline (WIP)
- GUI User space tool
- More standard submodules
- Add more V4L2 mechanisms / controls / options

COLLABORA



Thank you!

Helen Koike
helen.koike@collabora.com