Take Back Control of Your Cameras with libcamera

Presented at the embeddedworld Exhibition&Conference

2026-03-10, Nürnberg, Germany

Guten Tag

My name is Laurent Pinchart

- Founder and CEO of **Ideas on Board**
- Electrical engineer and software developer
- 20+ years of experience with **Linux** and cameras
- Linux kernel core contributor and maintainer
- Lead architect of the **libcamera** project

I am  and live in 

✉ laurent.pinchart@ideasonboard.com

9423 1B98 0100 EC61 9AC1 0E10 F045 C2B9 6991 256E

Introduction

We make cameras work

Ideas on Board provides embedded development services for camera and multimedia within the Linux open source ecosystem.

- 11 developers in 8 countries
- Recognized core contributor to the Linux kernel
- Open, upstream-first philosophy

Introduction

Today we will discuss

- Embedded camera architecture
- Linux camera ecosystem
- libcamera
- Real world use cases
- Camera tuning

Agenda

Do not hesitate to ask questions

If anything is unclear in this presentation, please feel free to ask for clarifications during the talk. For other questions, Q&A and discussions are scheduled at the end of the session at 12:30.

Agenda

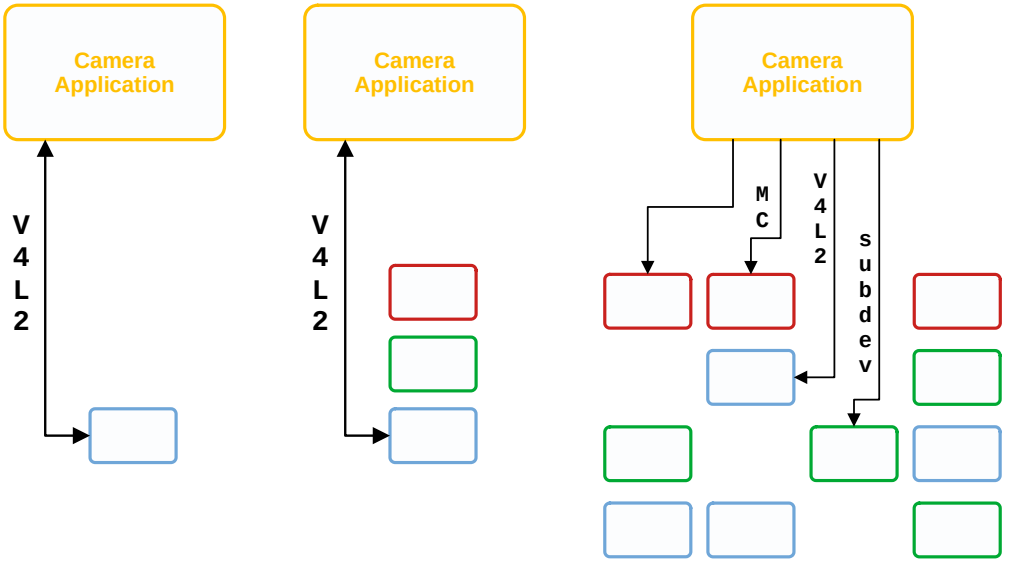
A Bit of History


NOKIA
Nseries

libcamera Genesis

In 2009 Nokia released the N900. It was the first Linux-powered (mostly) open-source smartphone. And it had an ISP.

All of a sudden, userspace applications had to deal with hardware complexity never seen before.



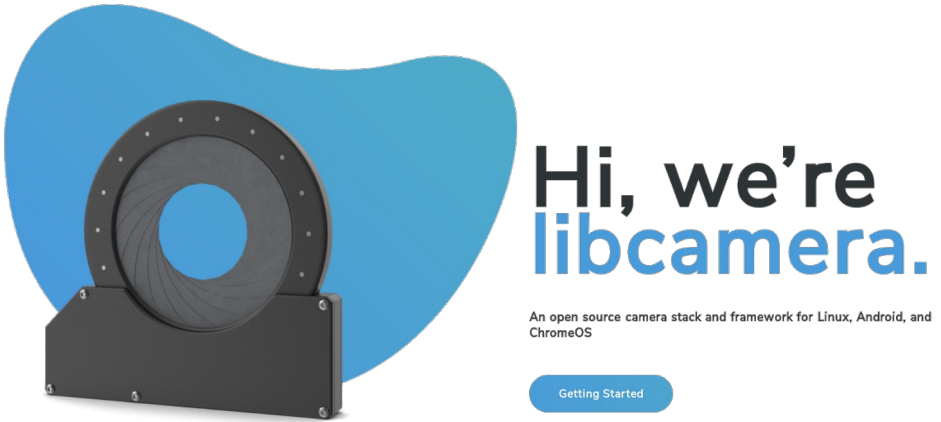
From simple to complex: evolution of the Linux kernel camera APIs

A Bit of History

libcamera Genesis

Due to unfortunate events[1], solutions that were envisioned by Nokia never saw the light of the day.

Nine years went by before the community took action.



1. Search for "Burning platform memo" to see how to destroy a mobile phone business

A Bit of History

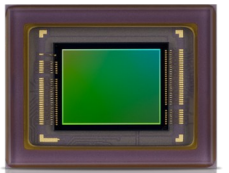
Embedded Camera Architecture

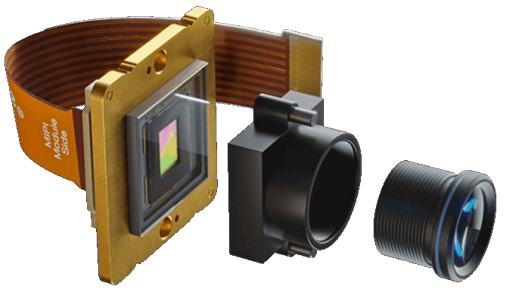
Terminology

An **image sensor**, also known as camera sensor, is an integrated circuit that includes a pixel array sensitive to light.

A **camera module** is an image sensor integrated with a lens, focus motor, IR filter, control electronics and a circuit board.

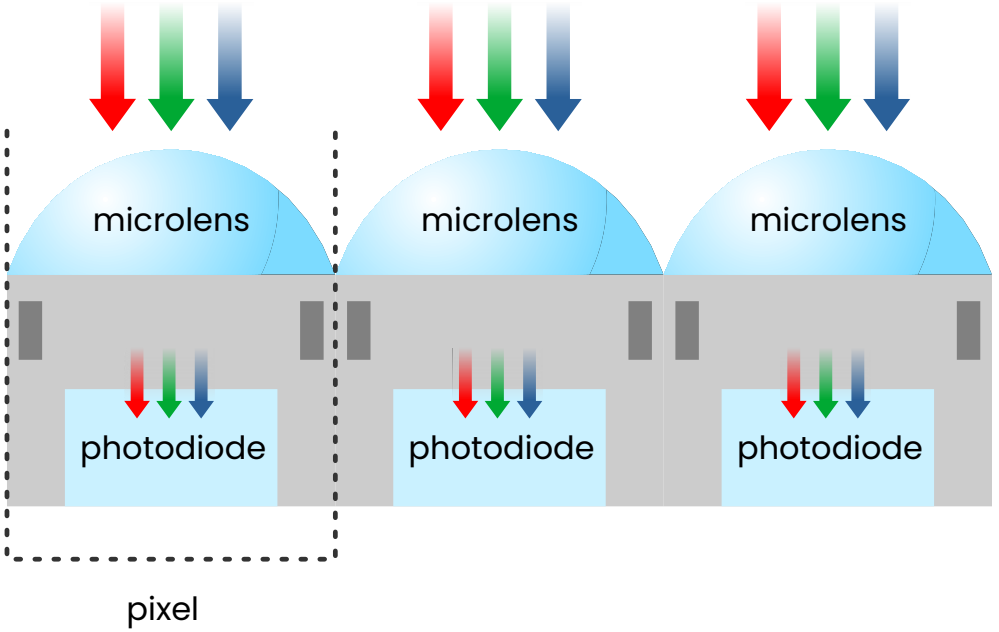
Camera modules come in different shapes and sizes, with different features.





IDEAS ON BOARD

Camera Modules



Pixels

In CMOS image sensors a pixel is implemented as **photodetector** (typically a pinned photodiode) behind a microlens, and active transistors to convert the photodiode charge to a voltage.

The light **incidence angle** causes variations in pixel response, and leakage currents cause **cross-talk**.

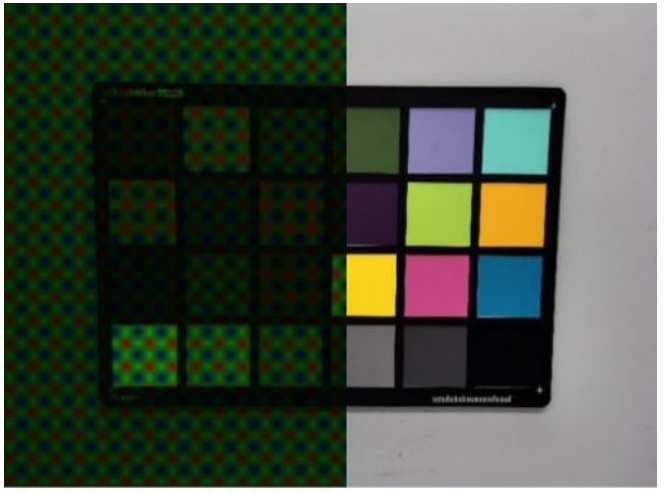
Image Sensor

IDEAS ON BOARD

The Need For Image Processing

Physical properties of the camera modules introduce imperfections in images that need to be corrected through digital processing to obtain usable images.

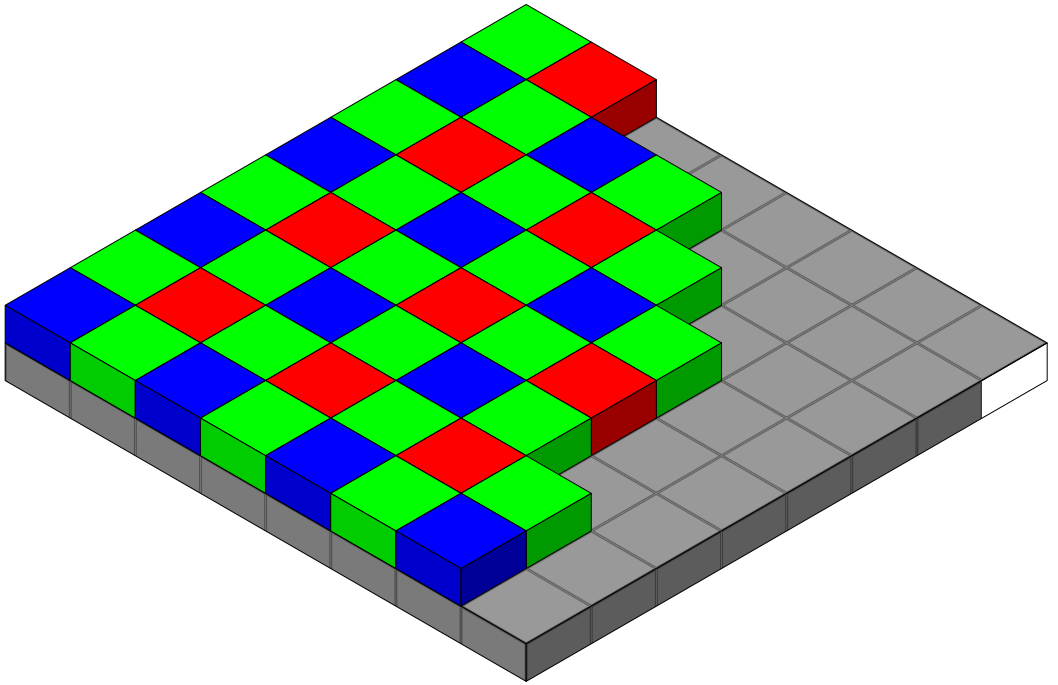
The type and amount of processing depend on the quality of the raw image produced by the camera module, and the intended usage of the images.



Colour Filter Array

Photodiodes are colour-blind by nature. **Colour information** is captured by placing a mosaic of tiny colour filters in front of the pixels.

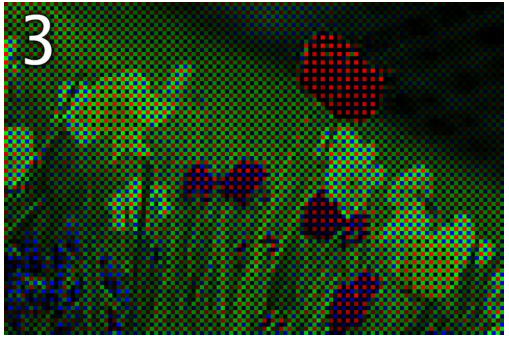
The CFA filters information out that must be reconstructed through **interpolation** of missing data. This can cause artifacts in the image.



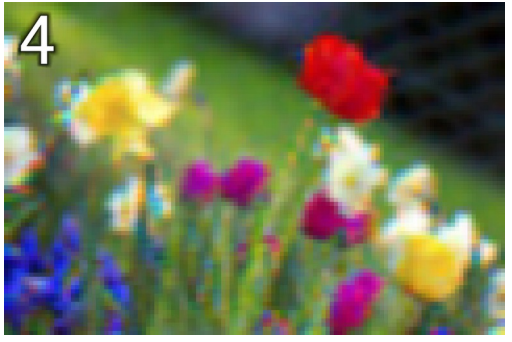
Source: https://en.wikipedia.org/wiki/Color_filter_array

Original

120x80 Bayer image



Colour-coded



Colour interpolation

CFA Interpolation

Images captured through a pixel array are made of a **mosaic** of coloured pixels. The missing colour data needs to be interpolated from neighbouring pixels.

Image Processing

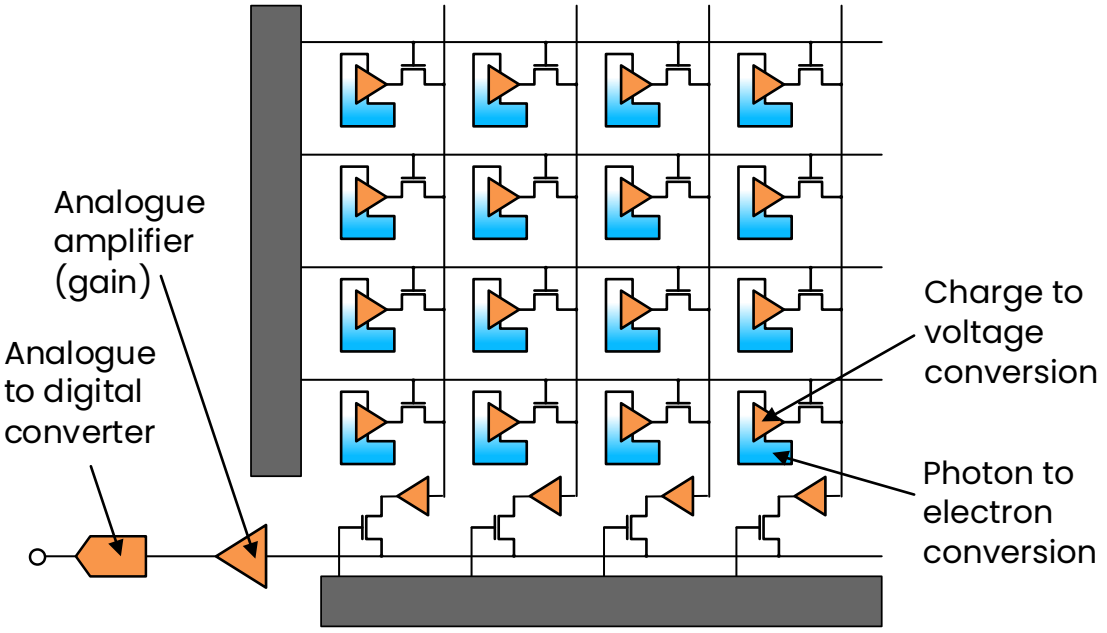
Source: https://en.wikipedia.org/wiki/Bayer_filter

IDEAS ON BOARD

CFA Interpolation

Simple colour filter array interpolation (e.g. bilinear) produces **false colours**. More complex interpolation algorithms can reduce the issue.

Source: https://en.wikipedia.org/wiki/Bayer_filter



Readout

In addition to the pixels, CMOS sensors include **readout circuitry** that transfer and amplify the pixel voltage, and convert it to a digital value.

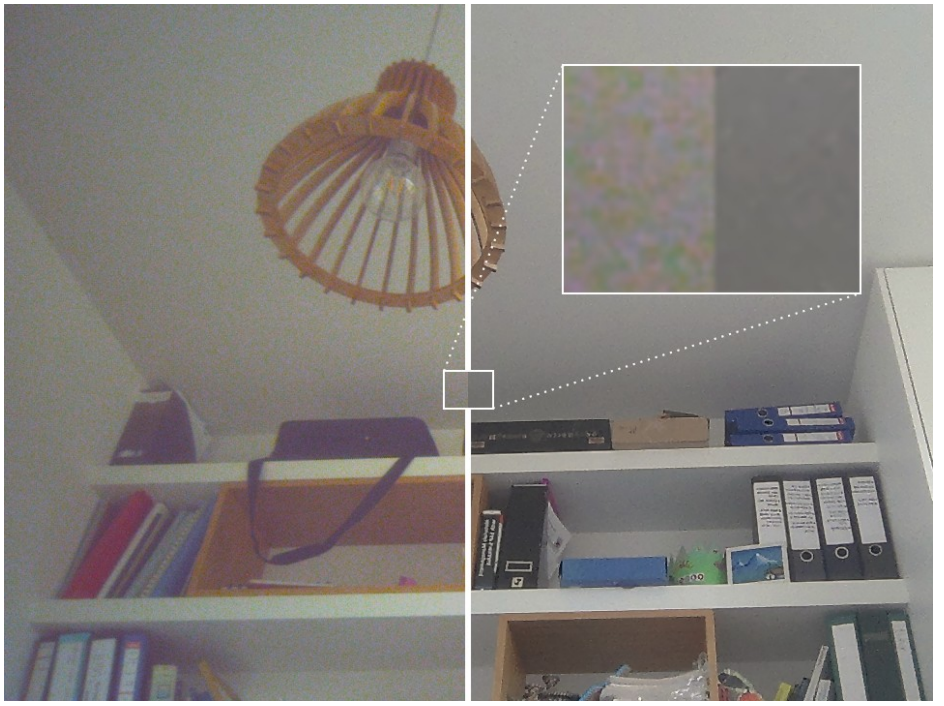
The readout circuitry creates and amplifies **noise**.

Image Sensor

Noise Reduction

Noise is introduced at all analog stages of the camera pipeline, and is amplified by digital processing. It degrades the perceived quality of the image and also affects processing negatively.

Many noise reduction algorithms exist, with spatial and temporal denoising often implemented in hardware.



Lens

Lenses are essentials to produce a focussed image, and can change the field of view. They can cast **lens shading** on the image or cause **distortions** (e.g. fish eye lenses).

Camera Module

Lens Shading

Lens shading, also called vignetting, appears as darker (and often colour-shifted) areas in the image corners. Despite its name, it is not cause by the lens only.

Image Signal Processor

Processing to improve the image quality is performed by an ISP, often implemented as fixed-pipeline hardware.

The ISP processes images and computes statistics. Those are used by the CPU in **real time** to implement **3A algorithms** (AGC, AWB, AF, ...). This **control loop** is specific to each ISP.

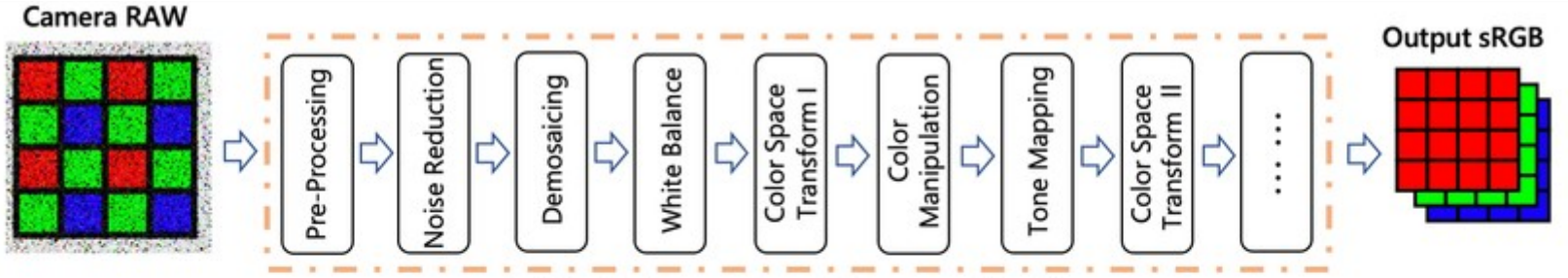


Image Processing

IDEAS ON BOARD

Linux Camera Ecosystem

Overview

The Linux camera ecosystem is made of a variety of independent but related projects, from the **Linux kernel** to applications, going through low-level **middlewares** and high-level **frameworks**.

Those projects collaborate, and receive contributions from a wide number of companies. Many hardware vendors and integrators still do not contribute.

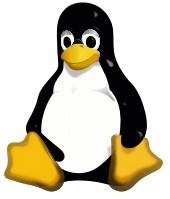
ROS

pipewire

gstreamer

OpenCV

libcamera



Camera Ecosystem

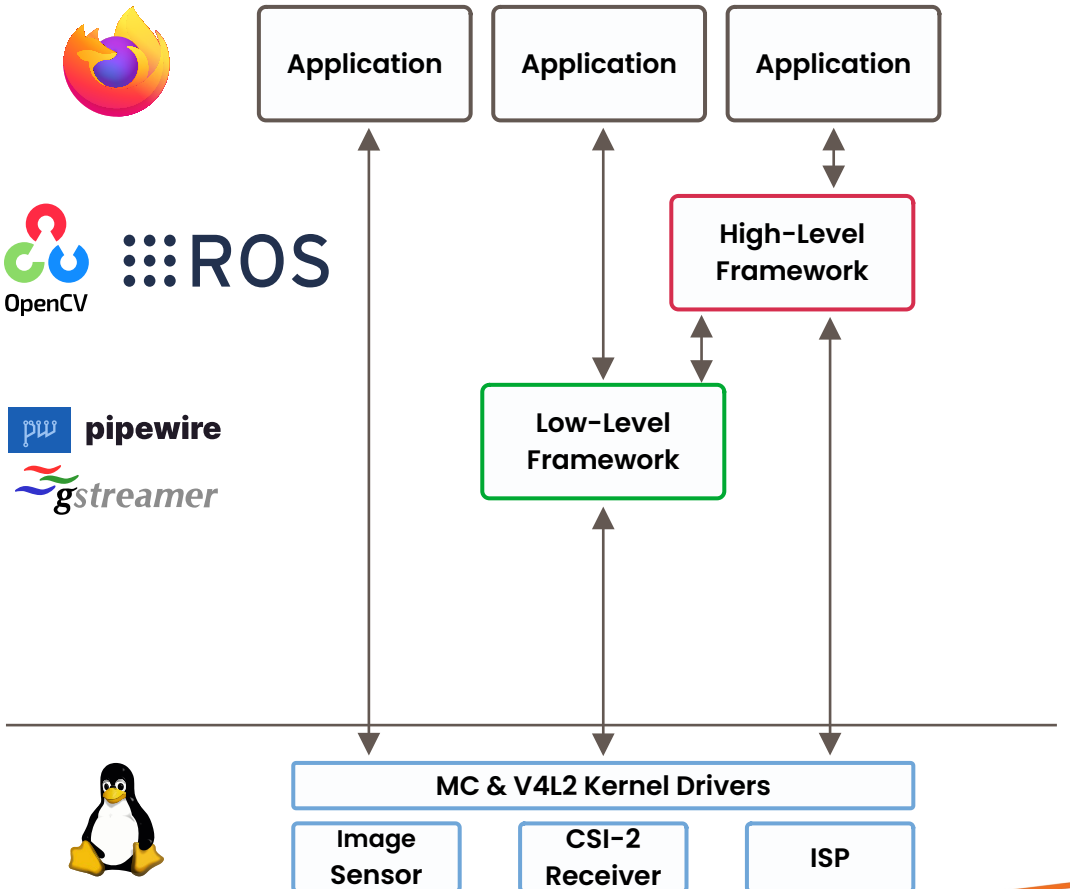
IDEAS ON BOARD

Open-Source Stack

Camera-related kernel drivers implement the **Media Controller** and **V4L2** APIs.

In userspace, applications usually use generic frameworks such as **GStreamer** and **PipeWire**, or specialized middlewares such as **OpenCV** or **ROS**.

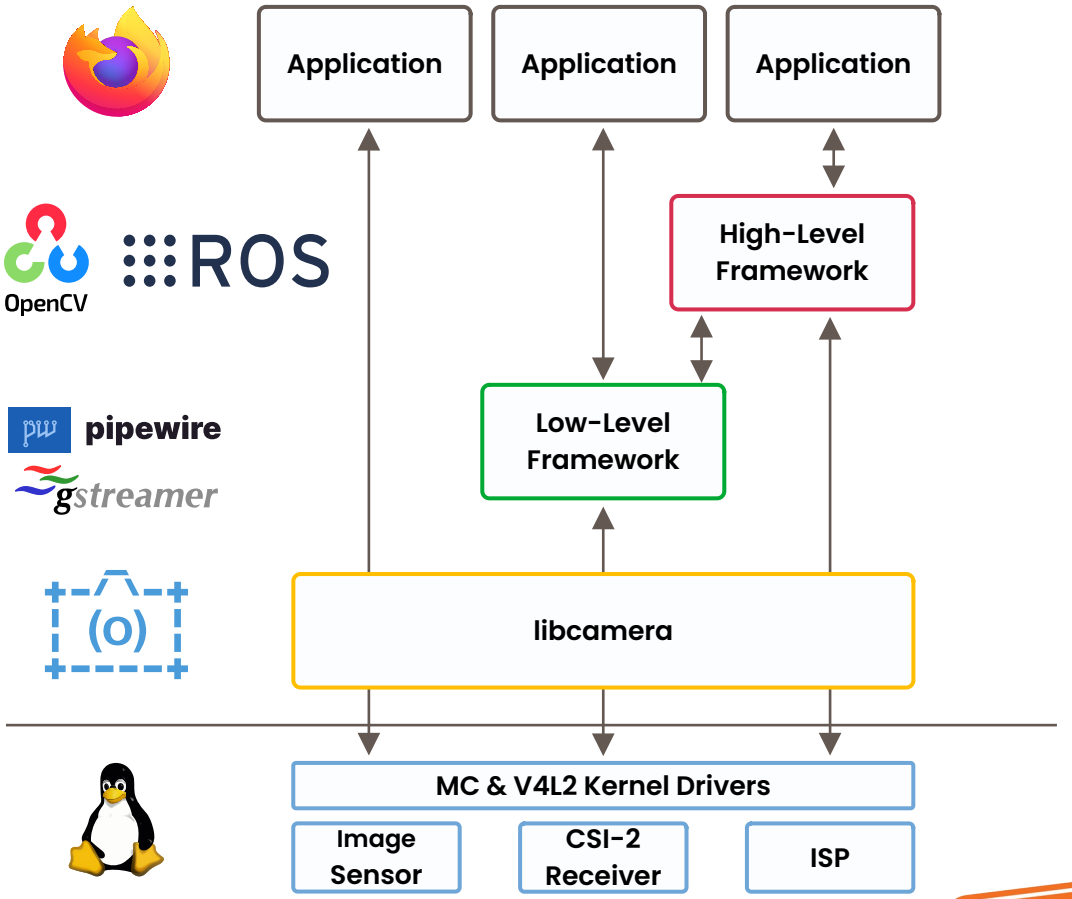
Direct access to the kernel drivers was the norm but is now discouraged.



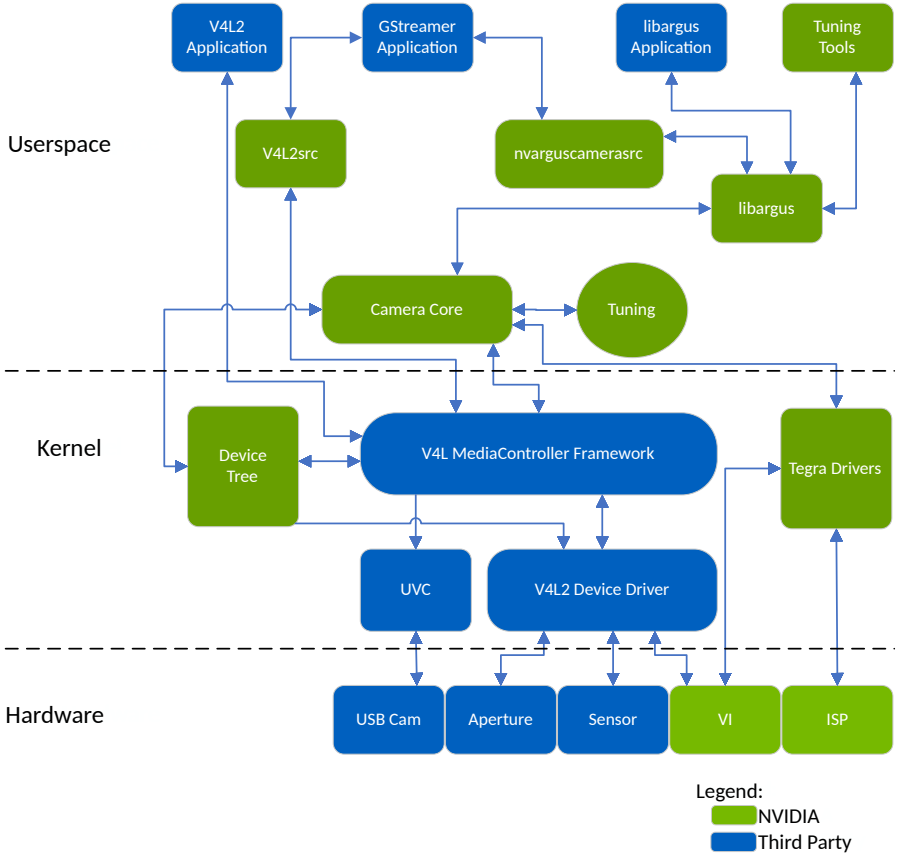
Camera Ecosystem

Open-Source Stack

On systems that require direct control of the image sensor and ISP, those tasks are handled by **libcamera**.



Camera Ecosystem



Source: https://docs.nvidia.com/jetson/archives/r38.4/DeveloperGuide/SD/CameraDevelopment/CameraSoftwareDevelopmentSolution.html

Proprietary Stacks

NVIDIA's camera stack is based on a closed-source library named **libargus**. It provides a proprietary API to applications, and is bridged to frameworks such as GStreamer.

On automotive systems, Nvidia provides an alternative, simpler safety-critical camera stack named **SIPL**.

Kernel drivers expose a vendor-specific API.

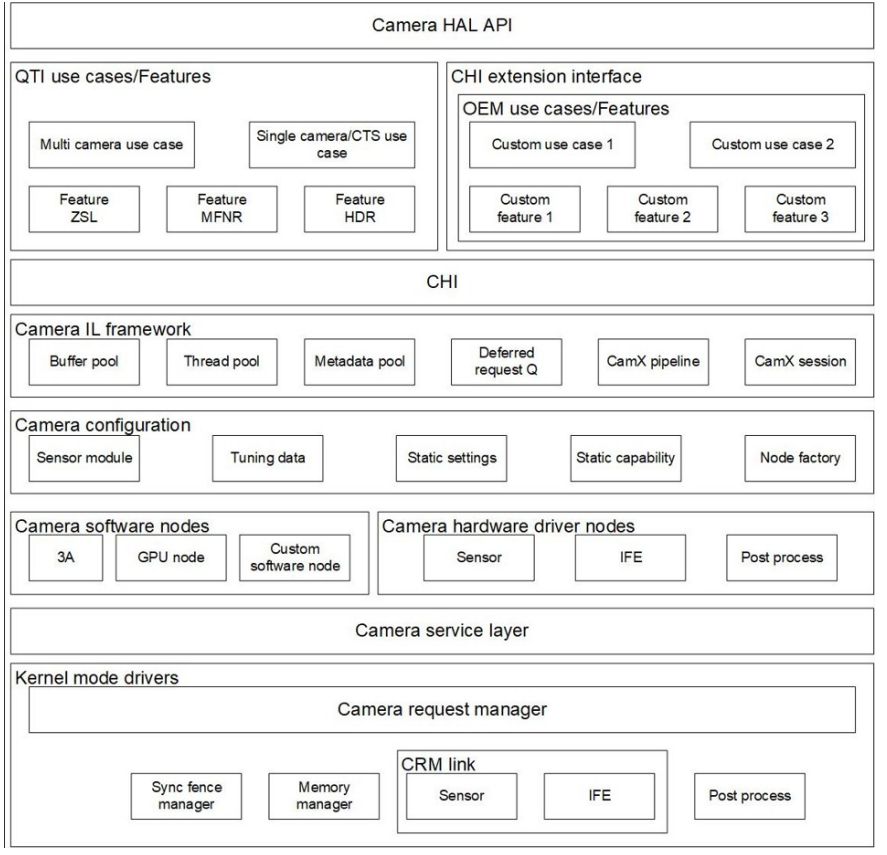
Camera Ecosystem

Proprietary Stacks

Qualcomm's camera stack is based on a closed-source framework named **CamX**, with various higher-level components built on top such as the Qualcomm Multimedia Framework (QMMF) or the Android camera HAL3.

Kernel drivers expose a vendor-specific API.

Public documentation is scarce.



Source: https://docs.qualcomm.com/doc/80-88500-4/topic/129_CamX.html

Camera Ecosystem

Proprietary Stacks

- ✓ Optimized by hardware vendors.
- ✓ Support available from the vendor.
- ✗ Vendor-specific integration, with closed ecosystem of service providers.

libcamera

- ✓ Fully adaptable thanks to source code availability
- ✓ Wide support ecosystem with third-party providers
- ✓ Long term support guaranteed, including security audits (CRA)
- ✗ Not all platforms are supported

Camera Ecosystem

Proprietary Stacks

- ✗ Optimized by hardware vendors. **For their use cases.**
- ✗ Support available from the vendor. **If you buy enough units a year.**
- ✗ Vendor-specific integration, with closed ecosystem of service providers.

libcamera

- ✓ Fully adaptable thanks to source code availability
- ✓ Wide support ecosystem with third-party providers
- ✓ Long term support guaranteed, including security audits (CRA)
- ✗ Not all platforms are supported

Camera Ecosystem

Kamaros

The Khronos Group is developing a new camera API named Kamaros by industry demand.

Current Kamaros design in-flight largely overlaps with libcamera.

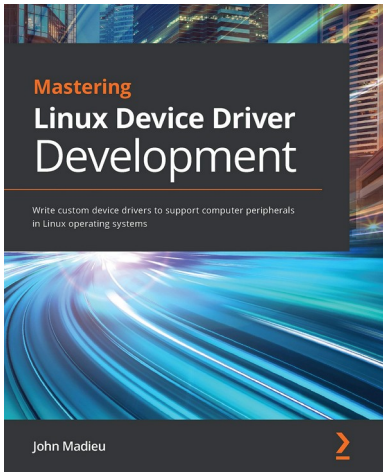
Interesting challenge to identify where API standards can best complement open-source libraries in the camera space.

Khronos welcomes your input.

Camera Ecosystem

Linux Kernel Drivers

Driver availability



Depending on the hardware, drivers can be found

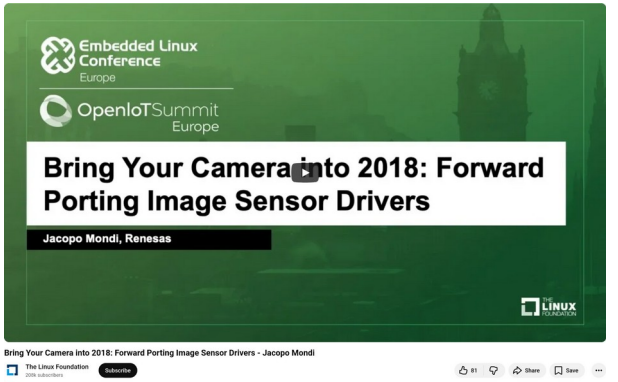
- In the mainline kernel.
- In a BSP kernel (same or different platform vendor).
- In an obscure unmaintained github repository.
- Nowhere.

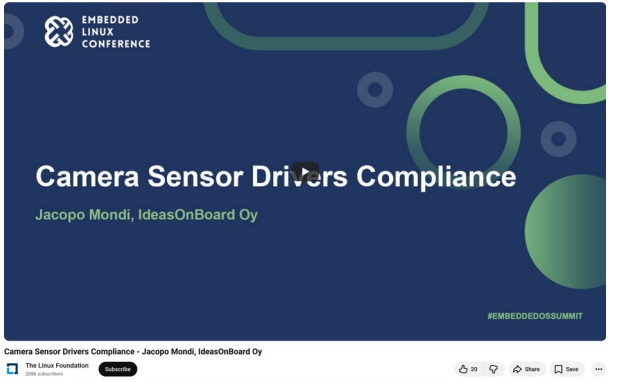
The further a driver is from mainline, the more it risks being incompatible with the target platform or camera module.

Ideally, **vendors should upstream drivers** for their devices.

Image Sensor Driver

How to develop a driver for an image sensor is out of scope for this presentation. The topic has been covered in multiple talks, including **”Bring Your Camera into 2018: Forward Porting Image Sensor Driver”**[1] and **“Camera Sensor Drivers Compliance”**[2] by Jacopo Mondi.





1. https://www.youtube.com/watch?v=PJVlvUf0gP4
2. https://www.youtube.com/watch?v=Cn1cHguVaLk

IDEAS ON BOARD

ISP Driver

How to develop an ISP driver is also out of scope for this presentation. The topic has been covered in the **"Image Signal Processing (ISP) Drivers & How to Merge One Upstream"**[1] talk by Helen Koike.

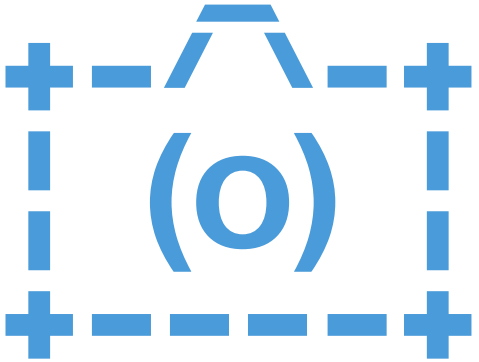
The Linux media community is ready to help, do not hesitate to reach out to the **linux-media@vger.kernel.org** mailing list, or the **#linux-media** IRC channel on oftc.net.

1. https://www.youtube.com/watch?v=SuAiJOtCxQY

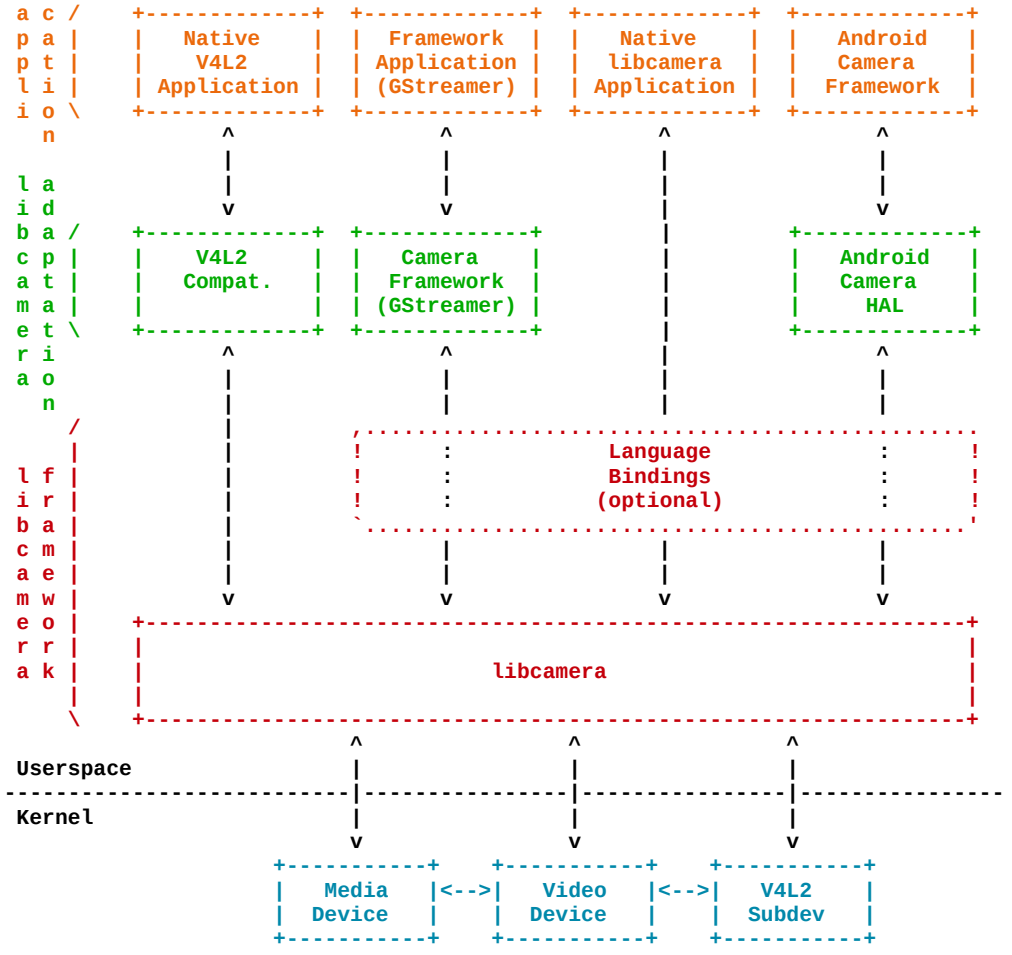
libcamera

libcamera is...

- a vendor-neutral, open project to implement full computational camera support in Linux
- a framework to **control ISPs**, imaging sensors, and other components
- a **standard API** for applications to use the camera
- a solution for **all Linux-based systems**, including embedded, Android and desktop environments

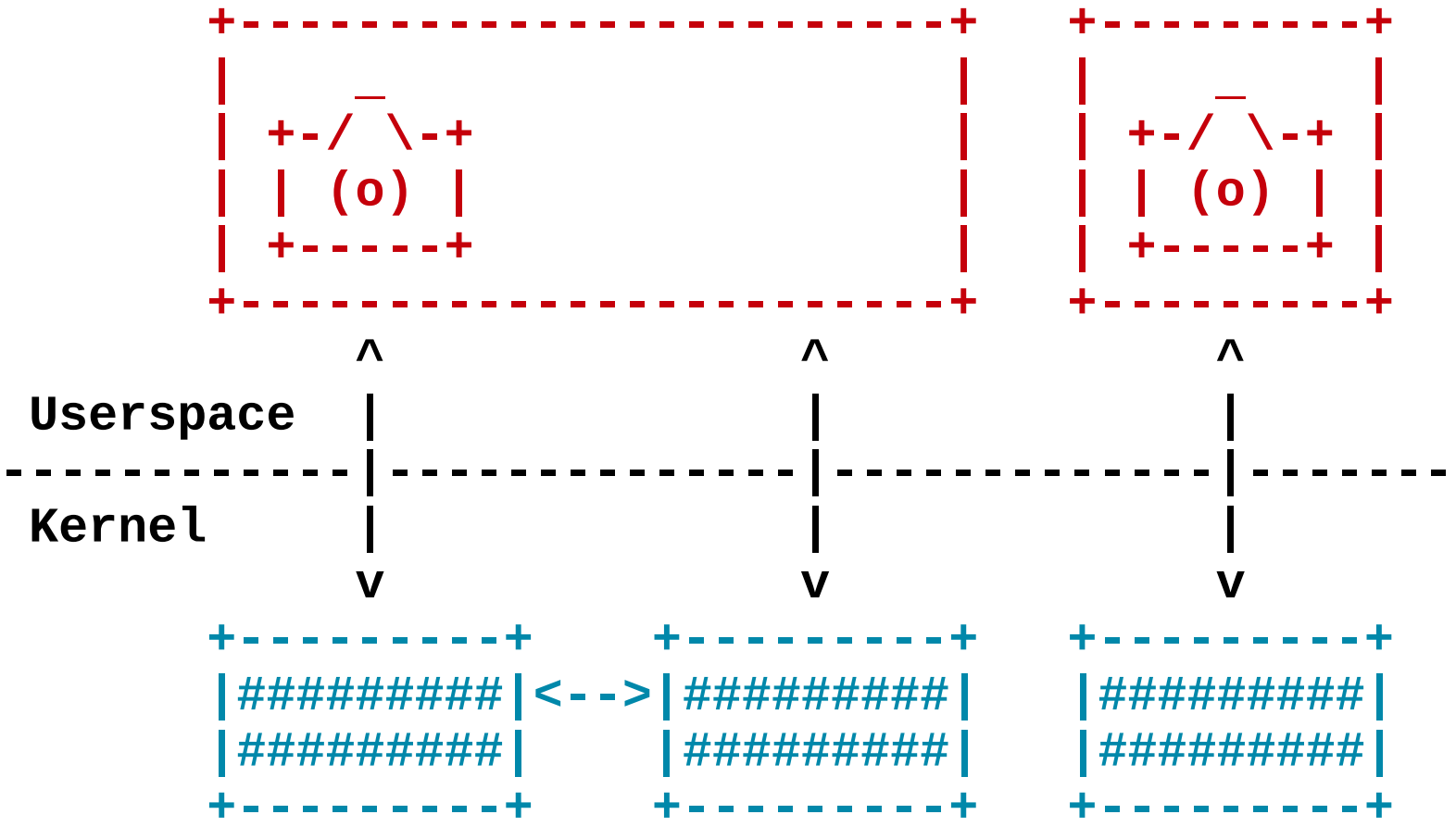


libcamera provides a complete userspace camera stack.



The 'mesa' of the camera world.

libcamera

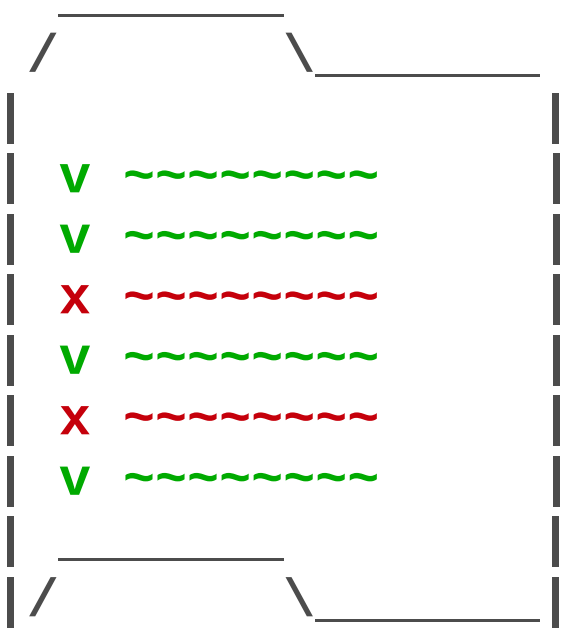


libcamera enumerates cameras...

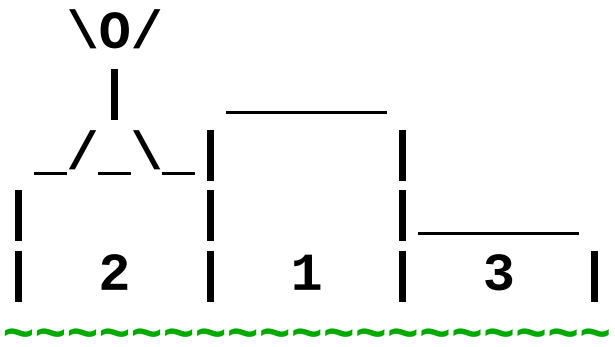
libcamera

IDEAS ON BOARD

... and exposes their capabilities.

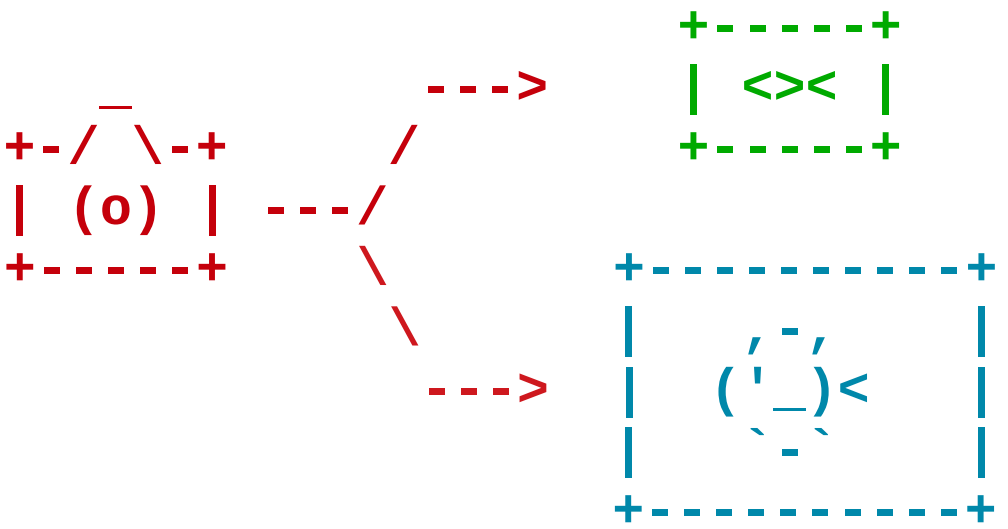


Capabilities



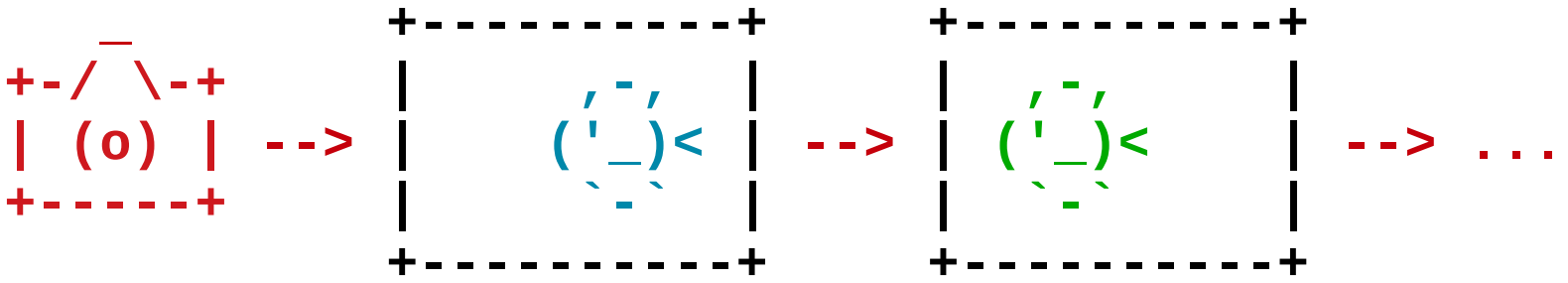
Profiles

It supports multiple concurrent streams for the same camera...

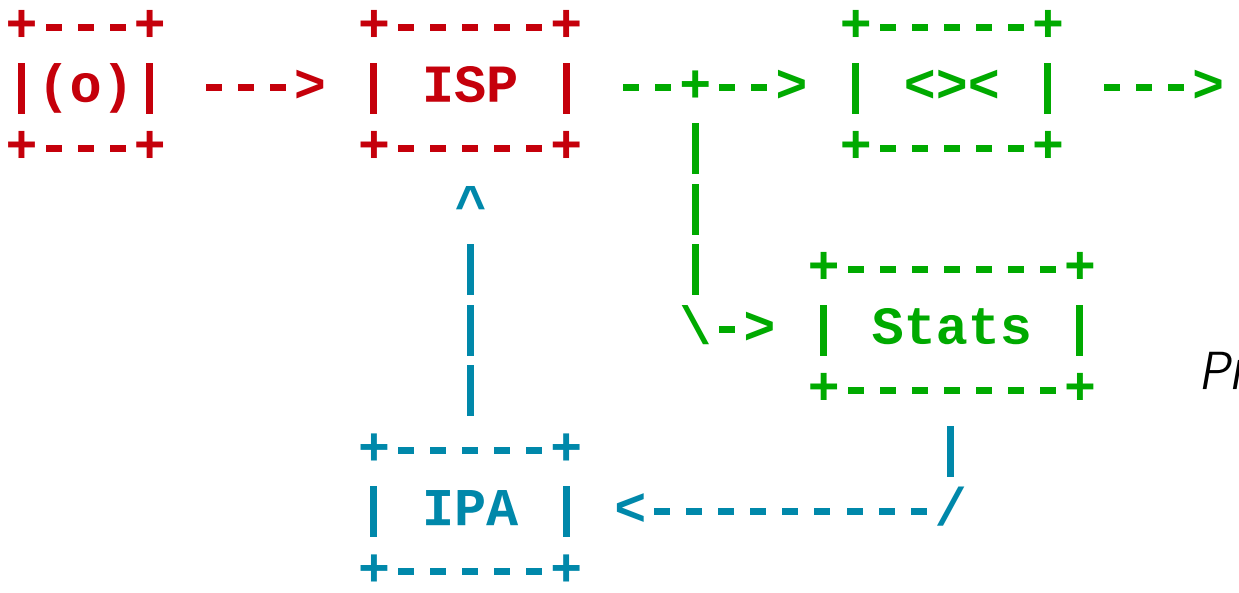


libcamera

... and per-frame controls.

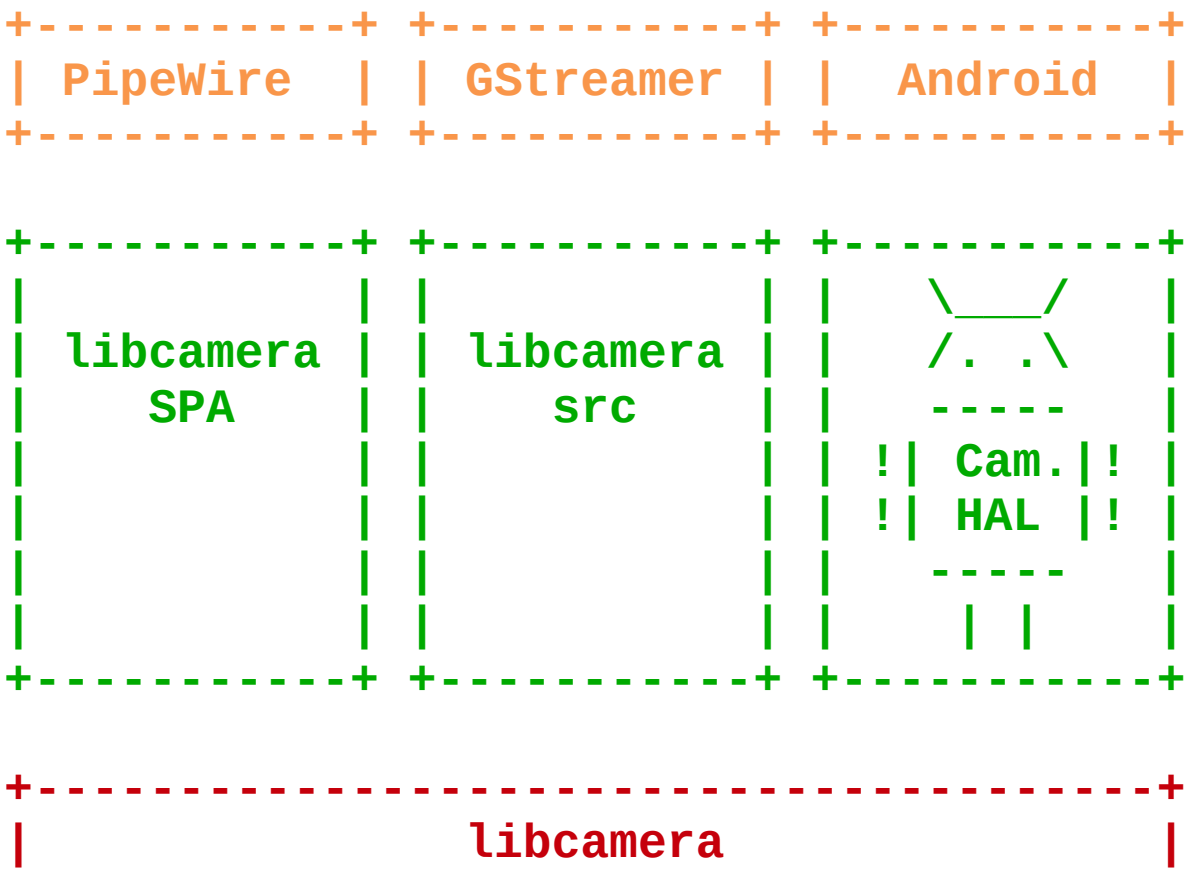


libcamera



It implements Image Processing Algorithms, loaded as external modules to offer IP protection.

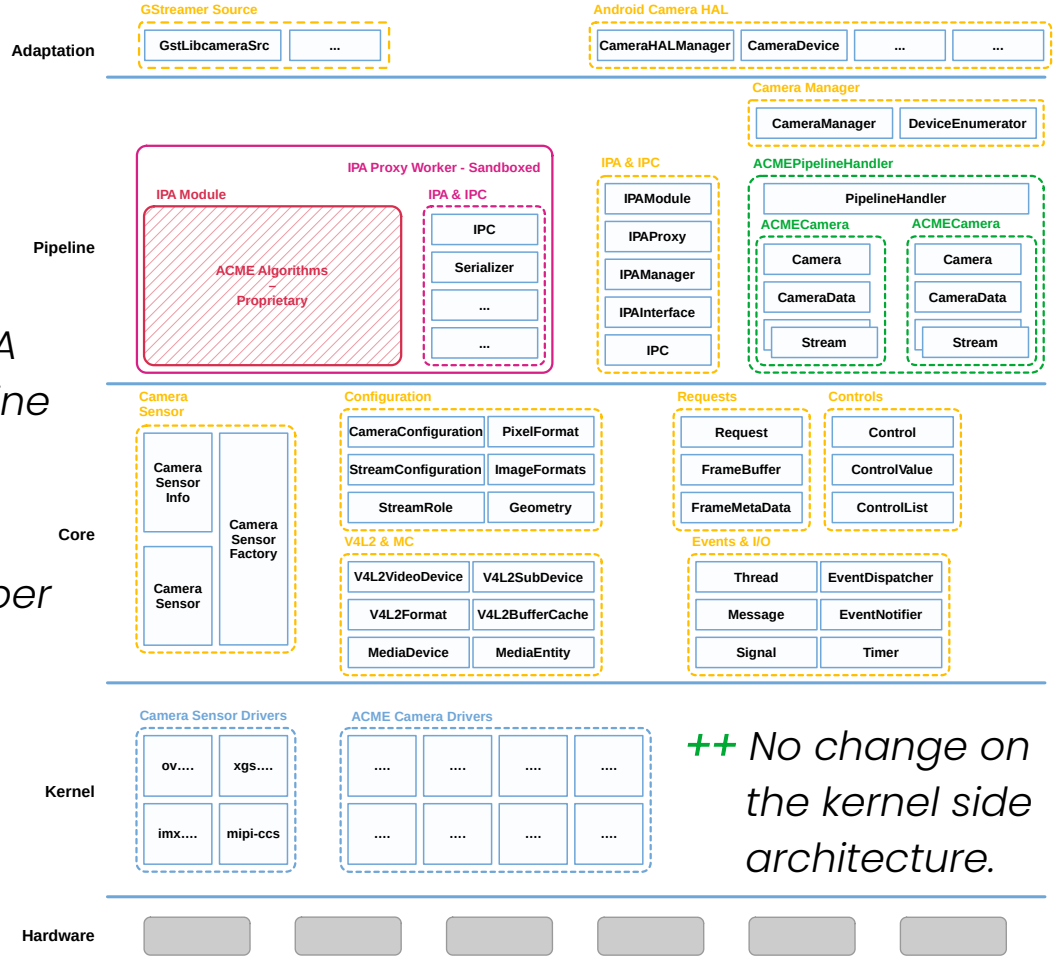
Adaptation layers integrate libcamera with existing APIs.



++ *IPA module sandboxing for safety.*

/!\ *Custom API for IPA module <-> pipeline handler communication.*

++ *libcamera wrapper classes reduce custom code.*



++ *No change on the kernel side architecture.*

++ *Standard Android Camera HAL Implementation.*

++ *GStreamer, V4L2, ...*

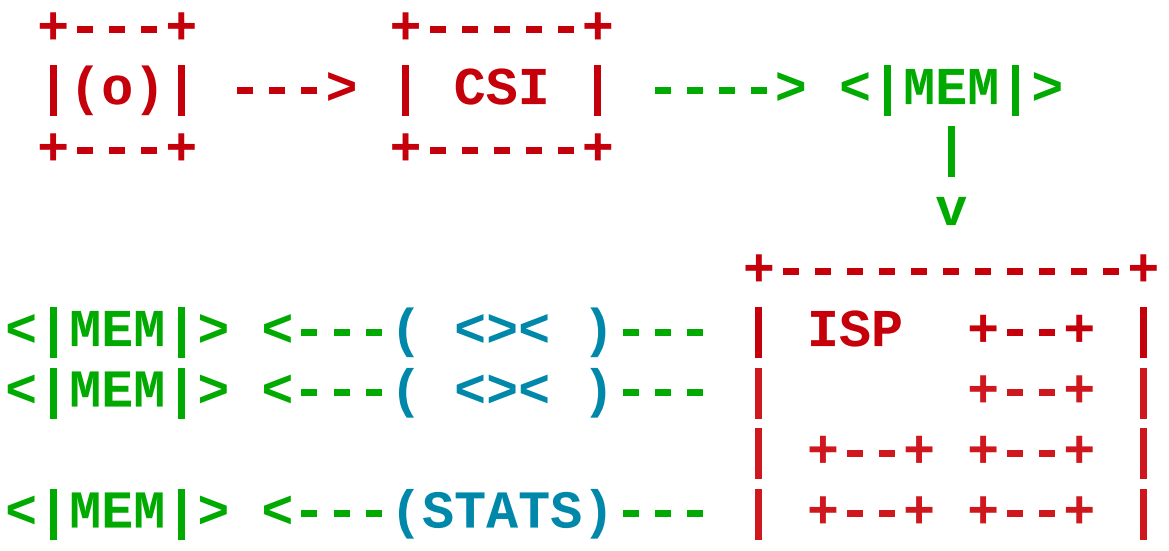
/!\ *Pipeline handler is vendor-specific development.*

++ *Development support available.*

/!\ *Implementation changes may be required to mainline drivers.*

libcamera

IDEAS ON BOARD



The pipeline handler interfaces with all kernel devices. It abstracts them and exposes video streams to upper layers.

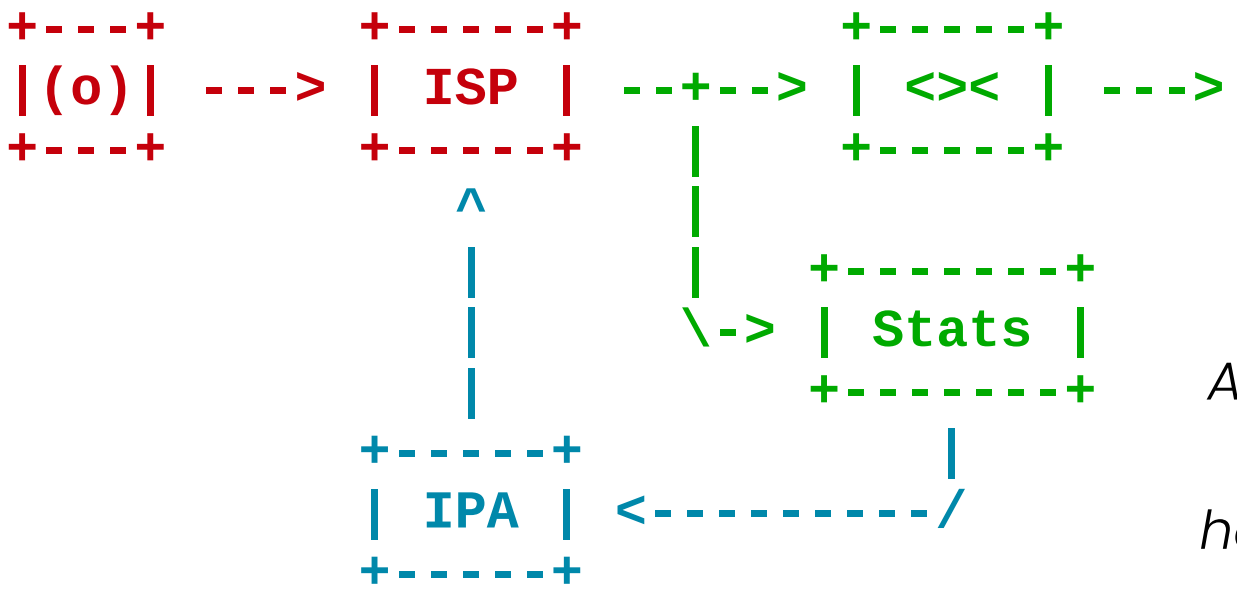
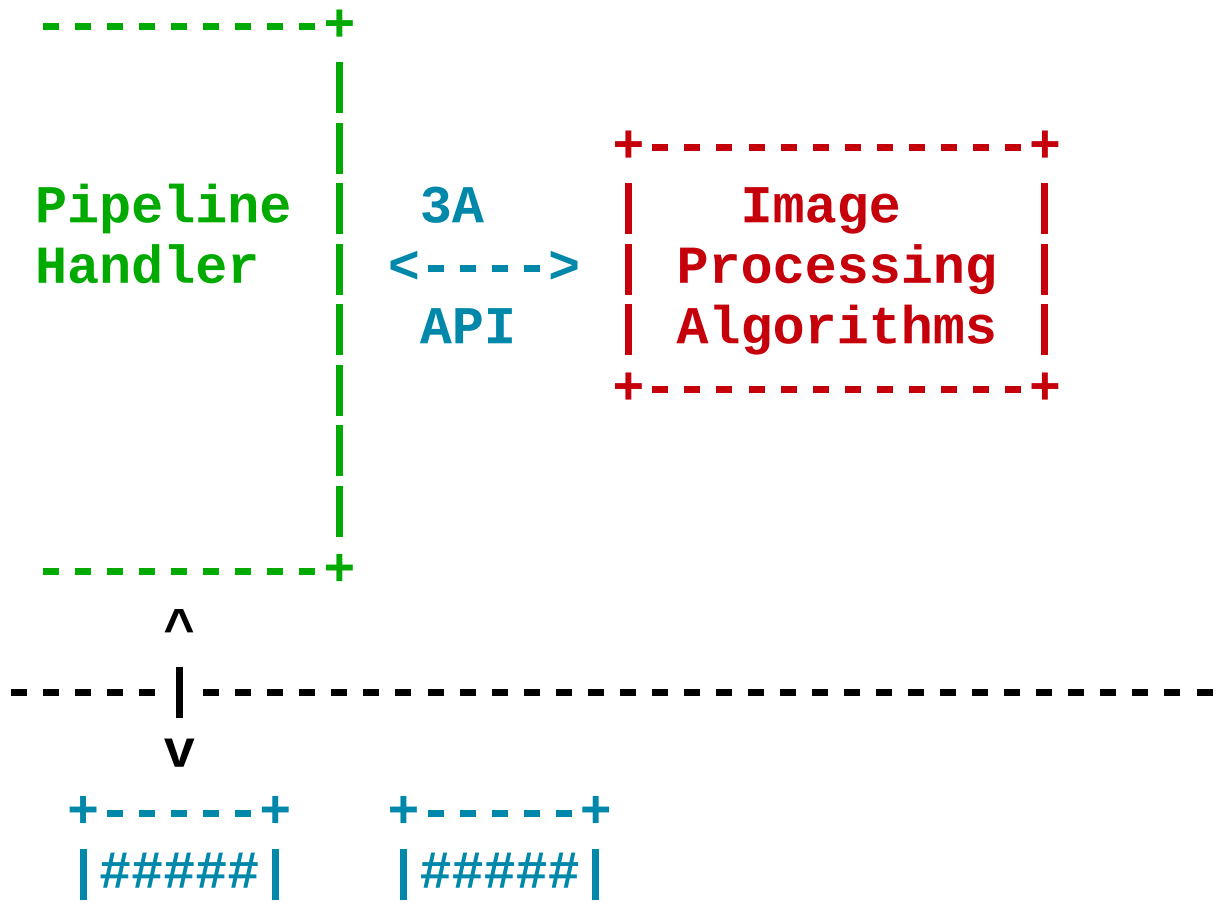
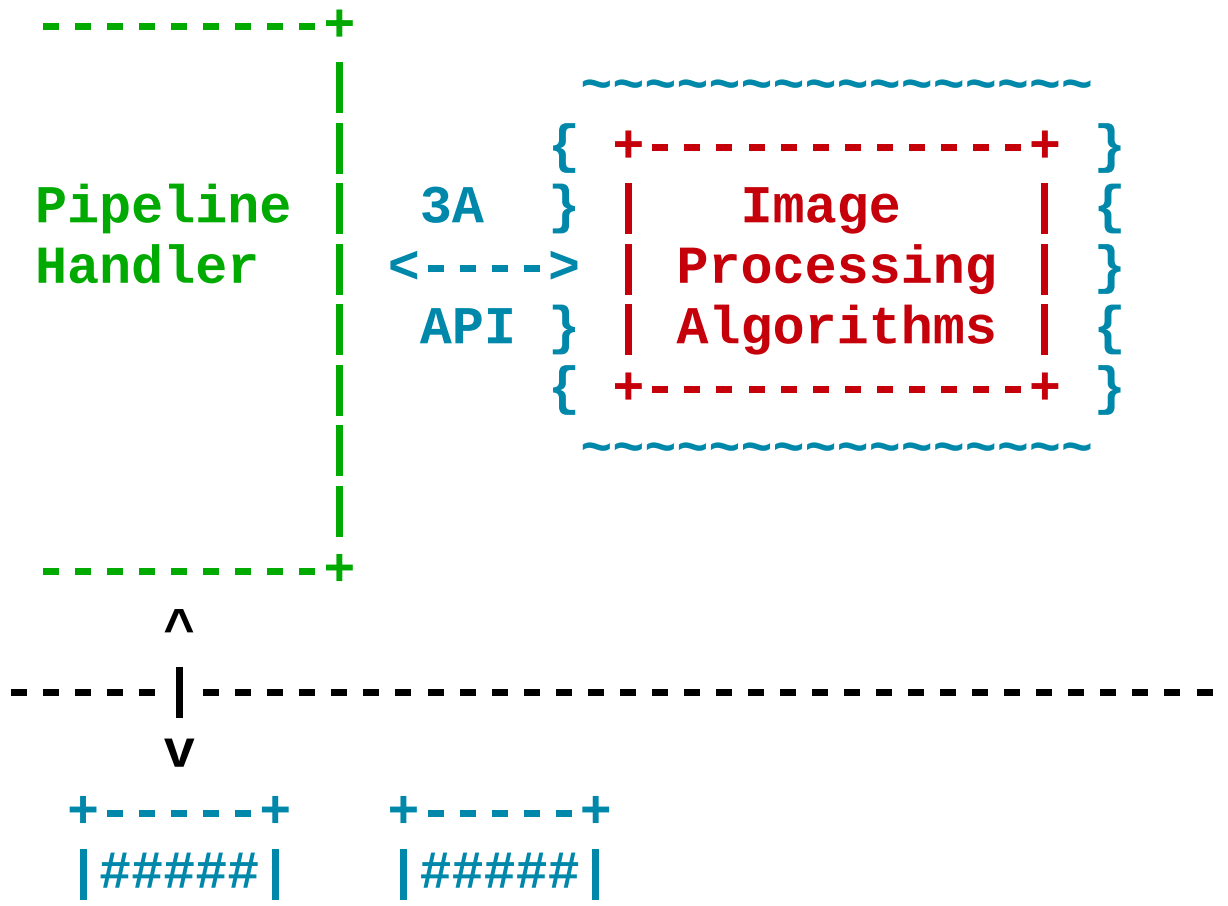


Image Processing Algorithms (IPA) receive statistics from the hardware and compute optimal image parameters.

libcamera

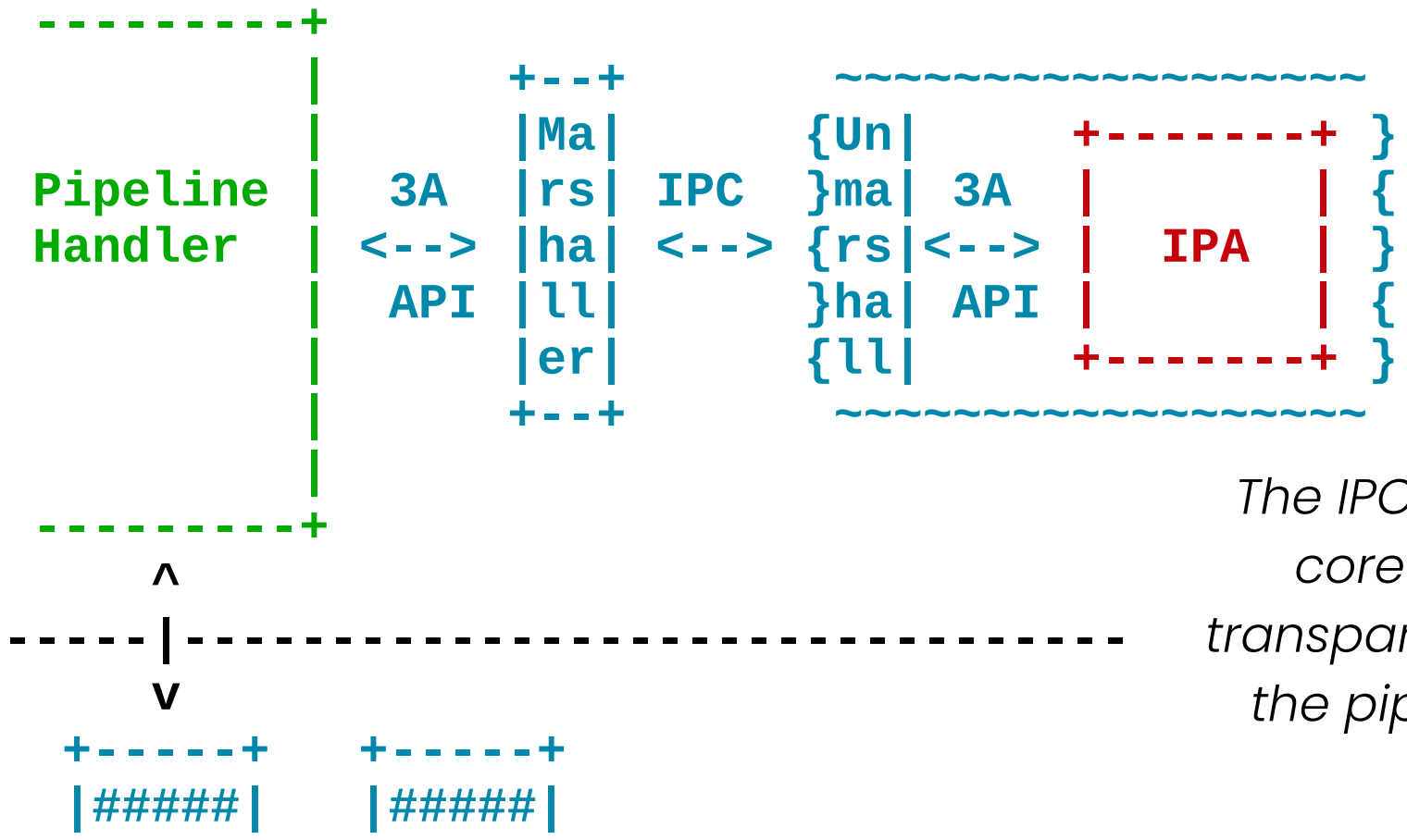


IPAs are separate modules that don't access kernel devices directly. They only have access to their pipeline handler through the IPA API.



Out-of-tree (including closed-source) IPAs are sandboxed in a separate process. They communicate with the pipeline handler through IPC.

libcamera


Pipeline Handler
3A API
IPC
Main thread
Unmanaged threads
3A API
IPA
The IPC is handled in core components, transparently for both the pipeline handler and the IPA.

Key Features

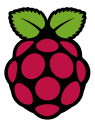
- Abstraction of the image sensor and ISP from applications
- Multi-stream capture
- Request-based per-frame control
- Fully open-source, fully adaptable
- Growing hardware ecosystem (Allwinner, Amlogic, Arm, Intel, MediaTek, NXP, Qualcomm, Raspberry Pi, Renesas, Rockchip, ST, TI)
- No vendor lock-in
- Library of reusable ISP control algorithms
- Software-based ISP (with GPU acceleration) for platforms with no usable ISP

Work in Progress & Future Work

- Per-stream controls
- Reprocessing API
- Improvements to the GPU-based ISP
- Better noise reduction support
- HDR and local tone mapping
- GPU-based post-processing

Contributing

An increasing number of hardware vendors and integrators are contributing directly or supporting development, including




libcamera sets development priorities based on user feedback. Too many integrators evaluate the project for their use cases without reporting missing features. **Get in touch** to report issues or we won't be able to fix them.

Real World Use Cases


Leica
TS20
E-portal
Find & add codes
Electricity lin
Pillar
Temp wall
Temp wall_002
Temp wall_001
Store
Stop
Define
Multi on
Page
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Fn
OK

Surveying Total Station

- Two image sensors (wide angle and telelens)
- Complex hardware pipeline in front of the SoC
- Power consumption, performance and accuracy constraints

libcamera helped support the complex hardware design, achieving the performance targets (camera switch <500ms) with pixel-perfect image alignment.

Use Cases

IDEAS ON BOARD


KARL
STORZ
Germany

Medical Endoscope

- Image sensor hotplug
- Boot time optimization
- Image processing customization (AGC, AWB, ...)
- Medical certification with very long term support

libcamera helped adapt image processing to light conditions inside the human body, and meet the medical industry safety standards.

Use Cases

Camera Tuning

Image quality tuning

Camera tuning is an required step to achieve good image quality.

libcamera provides a fully open-source tuning infrastructure. Work is ongoing to extend it to all the ISPs that libcamera supports, with multiple improvements, and a new tuning guide.

For more information about tuning, watch the **"Raw to Real and Green to Great: Open Source Camera Tuning for Linux Devices with libcamera"**[1] talk that Kieran Bingham has presented at FOSDEM 2026.

1. https://fosdem.org/2026/schedule/event/BQJPUP-libcamera-tuning/

Camshark

Camshark is a test, debug and development application for libcamera. It runs on a host machine and connects to the target device through ssh.

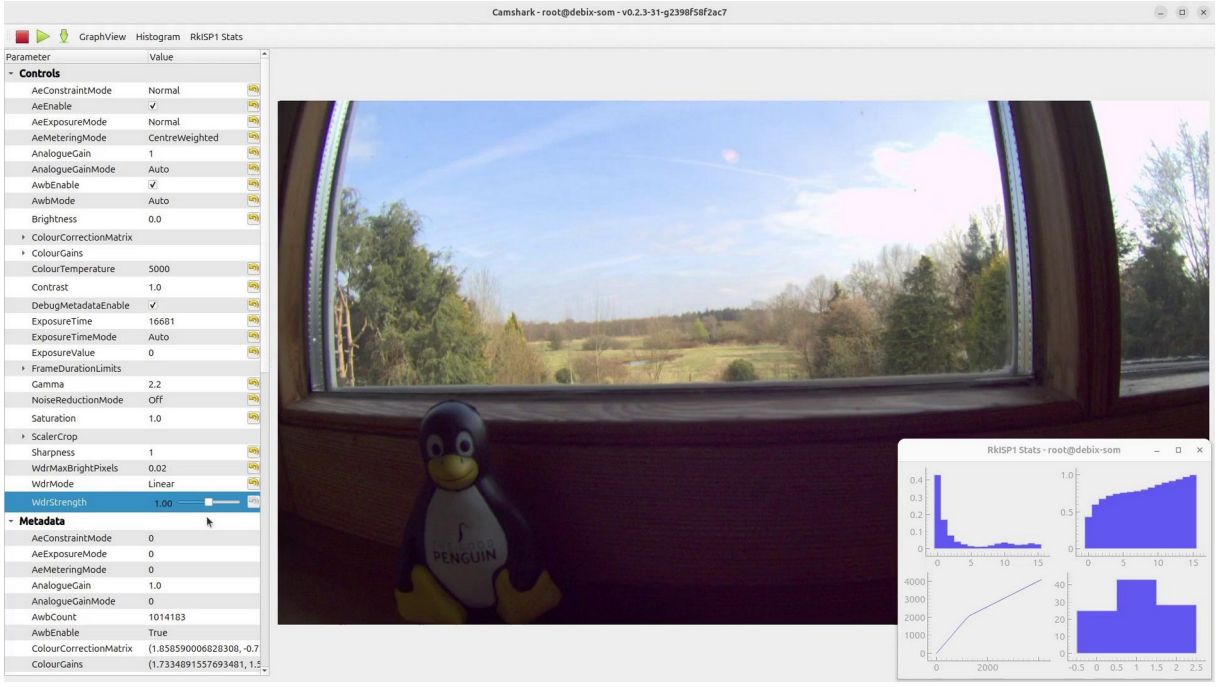
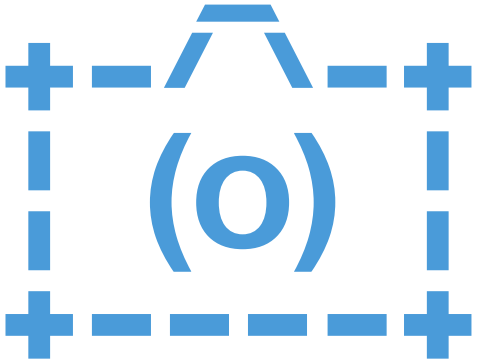


Image Capture

Conclusion

libcamera is...

- **fully adaptable** thanks to its open-source nature
- highly **maintainable** with a clean code base
- vendor-neutral, with **no vendor lock-in**
- ready for **long-term support** from an ecosystem of service providers
- a proven solution **deployed in products**
- continuously improved – **get in touch** with us

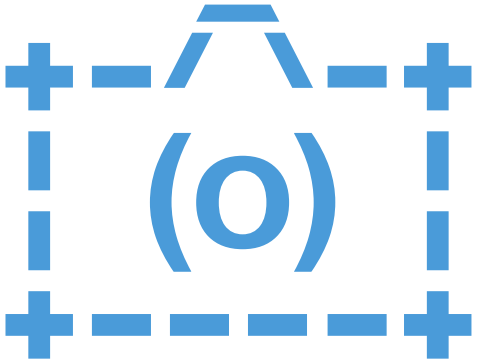


Conclusion

IDEAS ON BOARD

libcamera is...

- **fully adaptable** thanks to its open-source nature
- highly **maintainable** with a clean code base
- vendor-neutral, with **no vendor lock-in**
- ready for **long-term support** from an ecosystem of service providers
- a proven solution **deployed in products**
- continuously improved – **get in touch** with us



libcamera gives you back control of your cameras.
Demand it from your vendors!

Conclusion

IDEAS ON BOARD

Vielen Dank für Ihre Aufmerksamkeit