Linux and cameras: past, present and future

Presented at 

2026-05-28, Nice, France

Bonjour

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

IDEAS ON BOARD

Introduction

We make cameras work

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

- 10 developers in 7 countries
- Recognized core contributor to the Linux kernel
- Open, upstream-first philosophy

Introduction

Today we will discuss

- The past
- The present
- The future

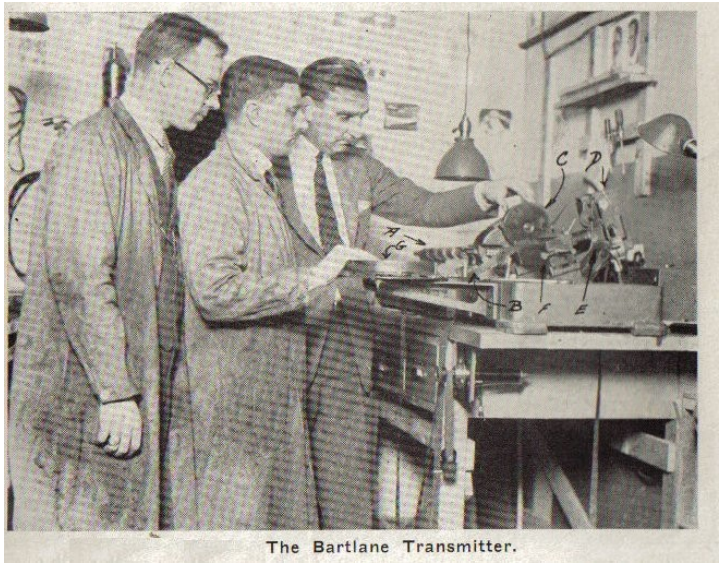
Agenda

Introduction

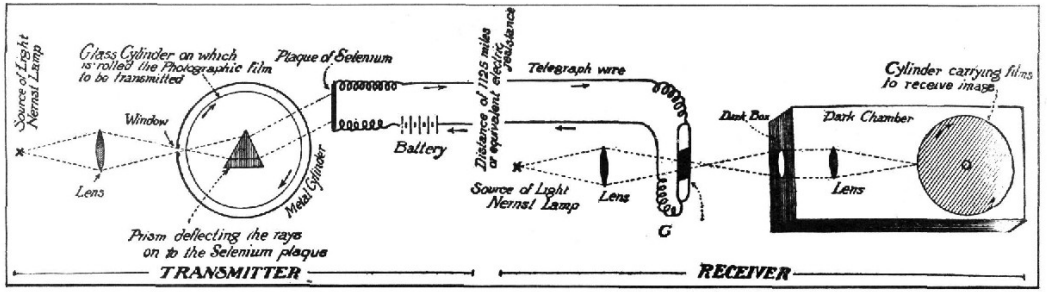
What we will not talk about

The first long-distance image transmission occurred in 1906 using the Korn system. In 1921 the Bartlane system transmitted the first digital image over the Atlantic ocean.

This is interesting history, but out of scope for us today.



https://www.hffax.de/history/html/bartlane.html



http://image.eastmanhouse.org/files/GEH_1953_02_06.pdf

Ancient History

Still out of scope

The CCD image sensor was invented in 1969. The first “portable” digital camera came in 1975. Invention of the pinned photodiode and NMOS image sensors lead to the CMOS image sensor in 1992.

This is material for a trivia game, not for this talk.

https://www.diyphotography.net/worlds-first-digital-camera-introduced-man-invented/

Ancient History

Getting closer

TV capture, camcorders, digital video broadcast are excluded from our agenda. We will focus on cameras.

The V4L2 API was merged in the v2.5.46 version of the Linux kernel in 2002. Anything predating V4L2 is the realm of history books.

Ancient History

IDEAS ON BOARD

Nearly there

Webcam history would make another interesting talk, for another day. USB webcams are still very relevant in the market, but they're boring as they just work[1].

1. Except when they don't, which happen way too often, in which case they become either interesting or frustrating.

IDEAS ON BOARD

Ancient History

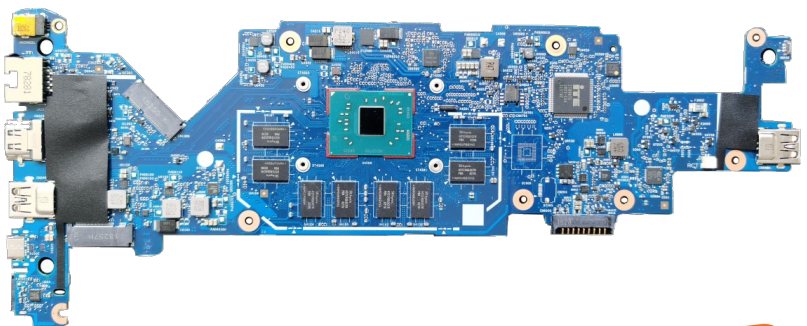
Modern days

The first ISP driver was merged in v2.6.39, released in 2011. This is where we will draw the line: modern camera support in Linux comes with the **Media Controller** and **V4L2 subdev** APIs.

Hardware

We will focus on systems that combine a **raw image sensor** and an Image Signal Processor (**ISP**), where both are exposed to and controlled by the Linux kernel.

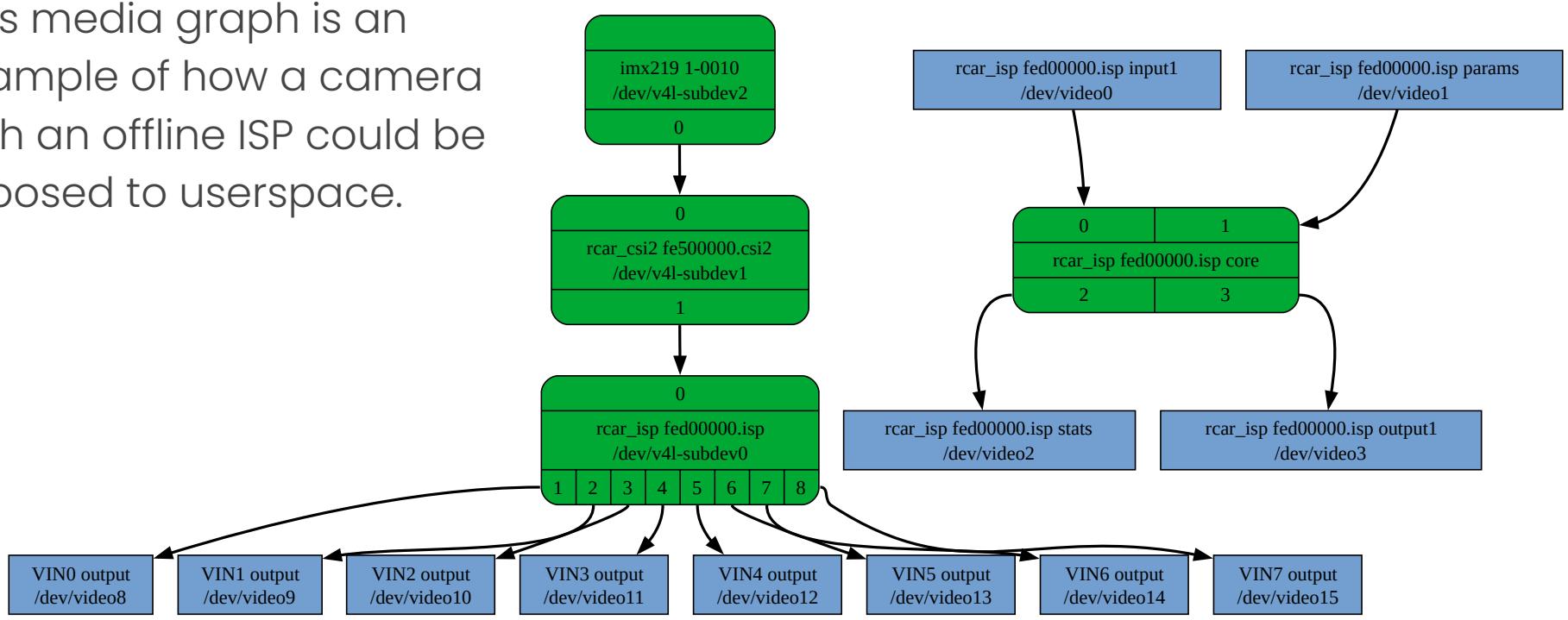
This it not limited to **embedded systems**. Many **laptops** also qualify, as well as nearly all **phones** and **tablets**.



Scope

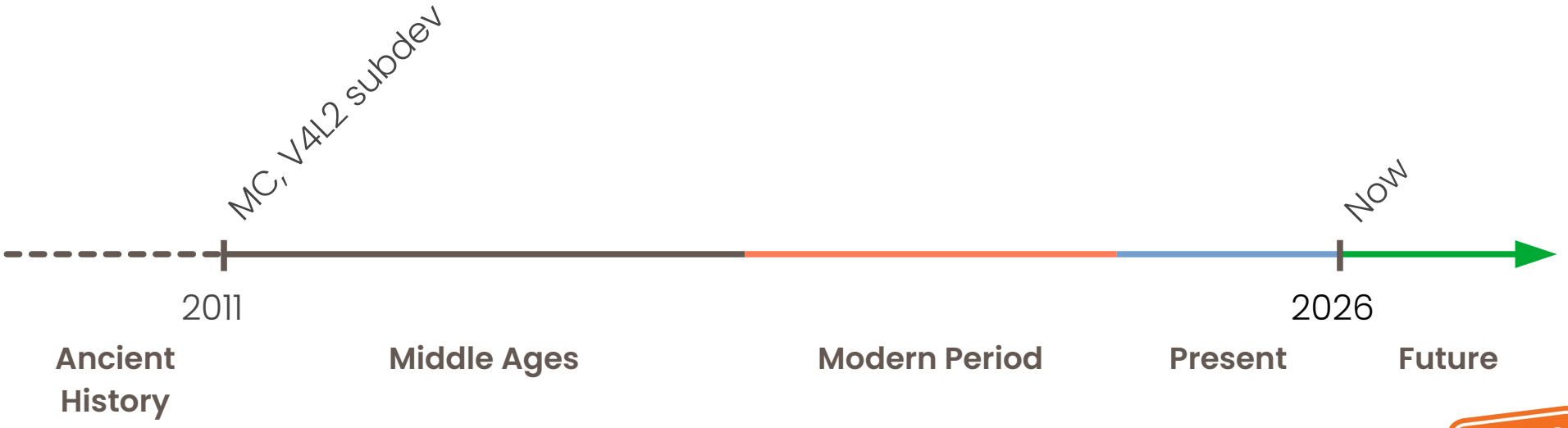
Media graph

This media graph is an example of how a camera with an offline ISP could be exposed to userspace.



Scope

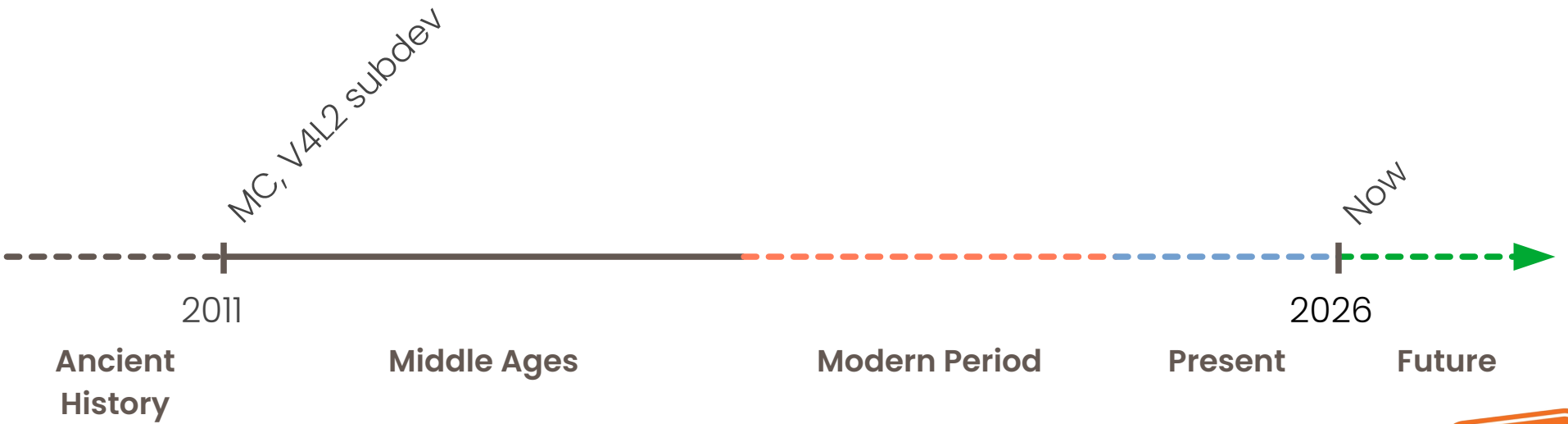
Linux Camera Timeline



History

The Past

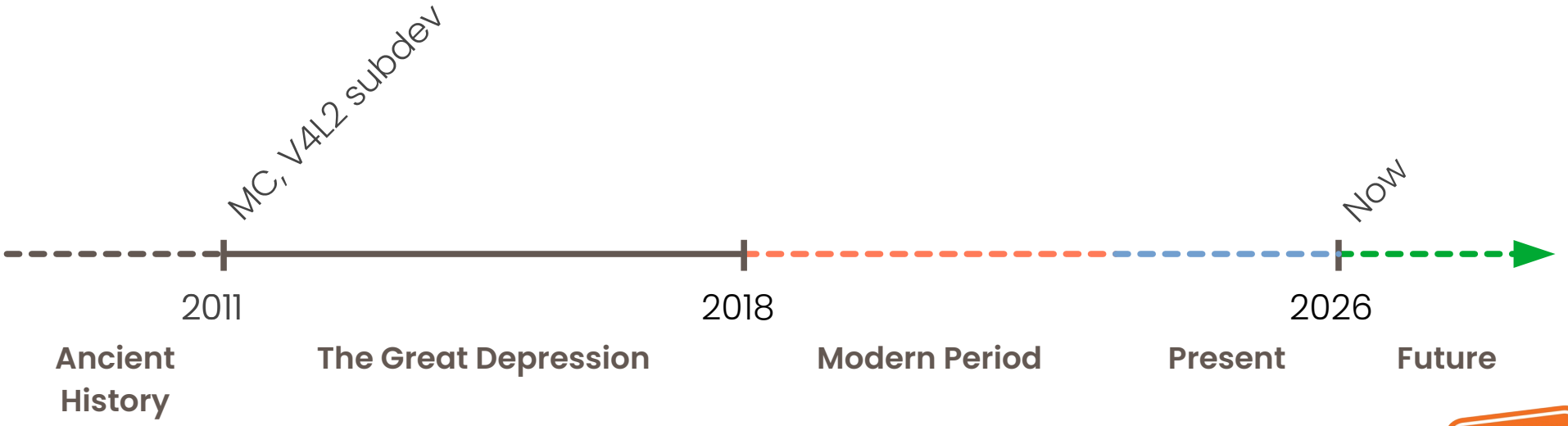
Linux Camera Timeline



The Past

The burning platform

A long period during which nothing happened. This started with “the burning platform memo” in 2011. Linux entered this period with one ISP driver.

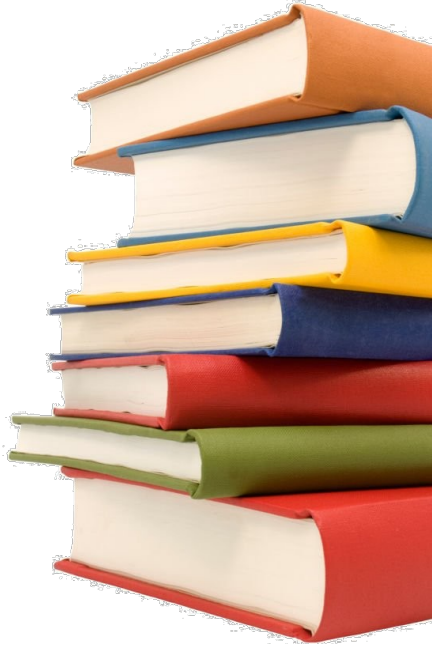


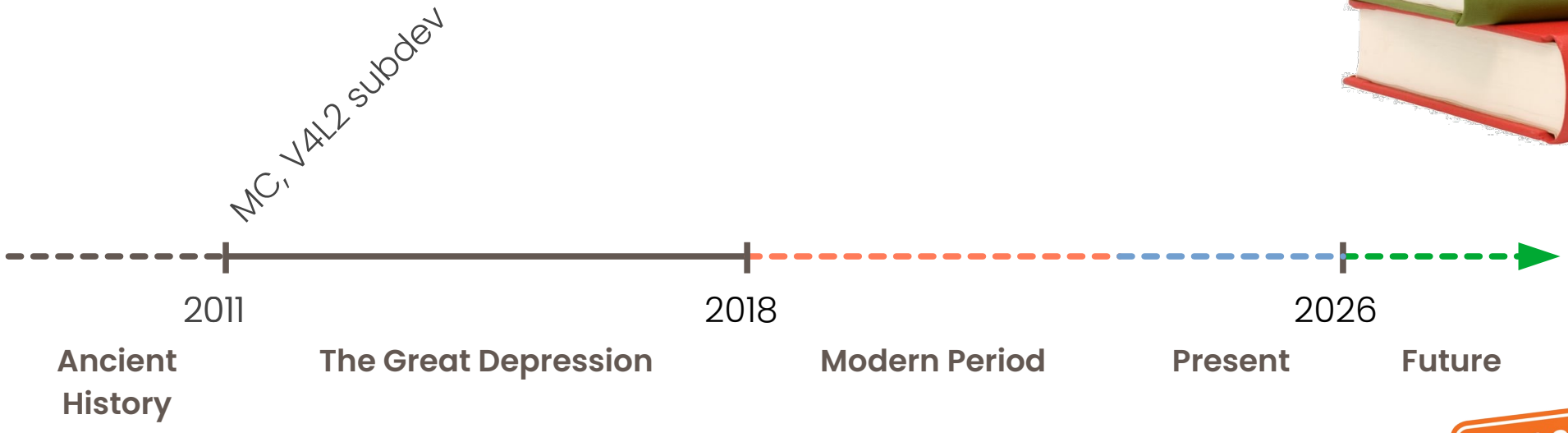
The Great Depression

Well documented APIs

A guide to application usage of media controller devices during this period can be found at

https://git.ideasonboard.org/doc/mc-v4l2.git/

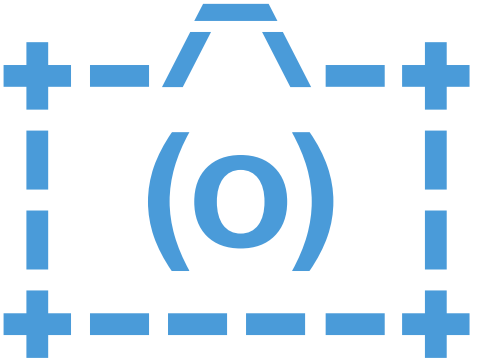


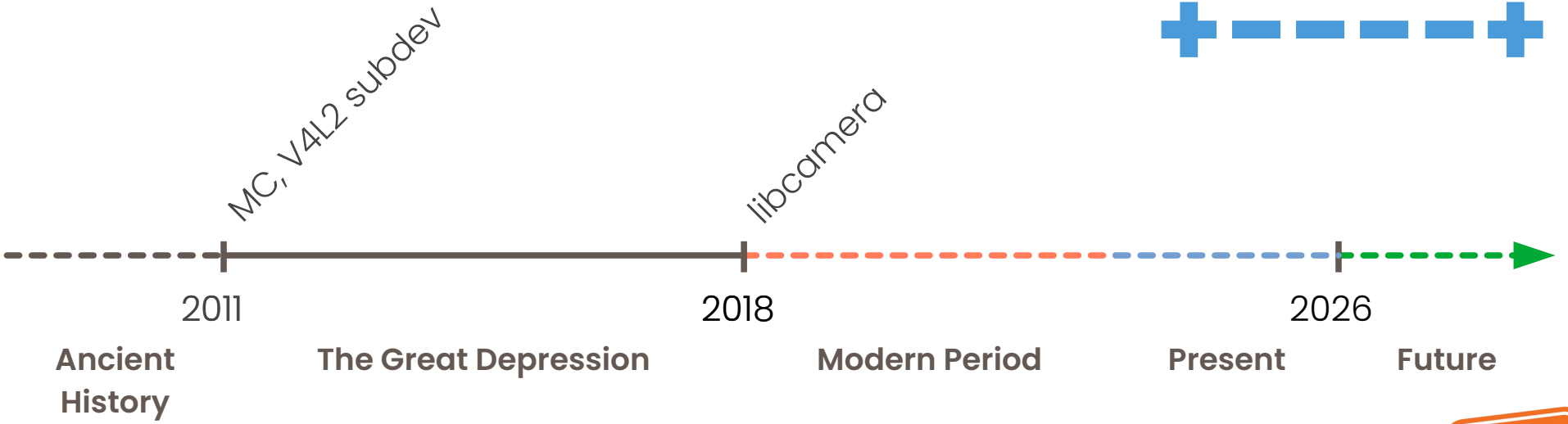


The Great Depression

A new hope

The Great Depression lasted until “the great Tōkyō meeting” in 2018, which led to the creation of libcamera. When libcamera was announced, Linux had gained a second ISP driver (of staging quality).

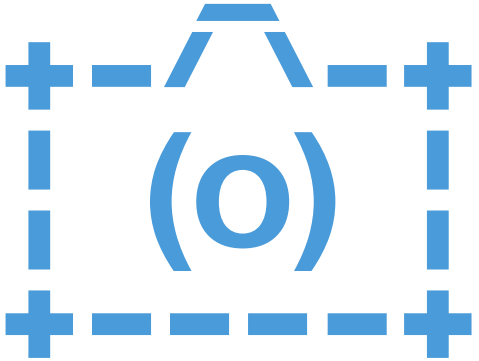




The mesa of the camera world

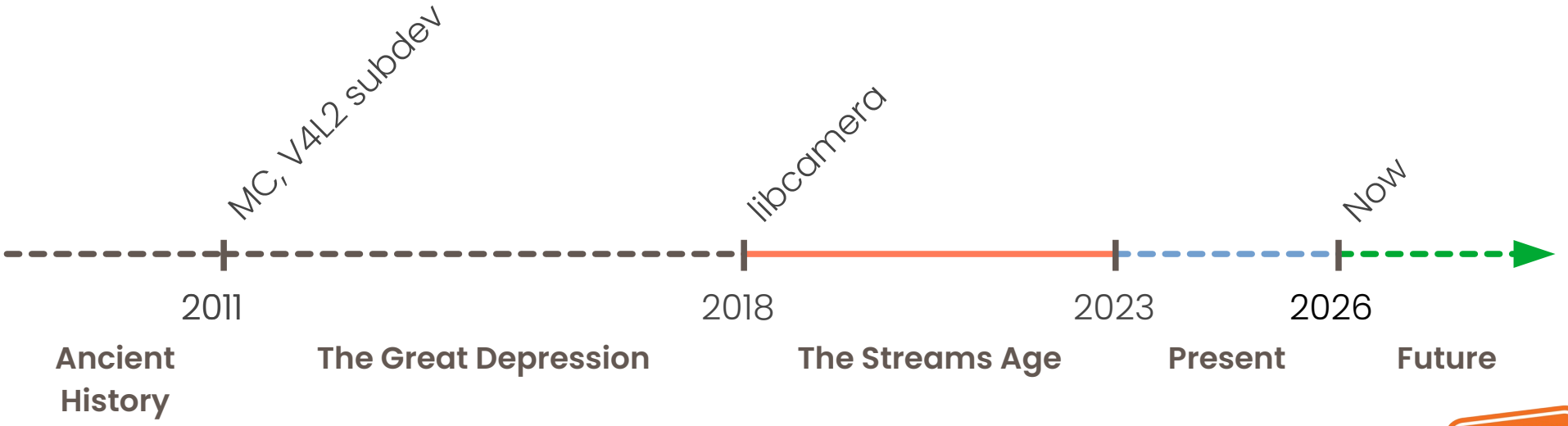
libcamera is an open-source userspace camera framework that controls image sensors, ISPs, and other components in the camera pipeline.

It exposes an API to applications, and integrates with GStreamer, PipeWire and Android.



V4L2 evolves again

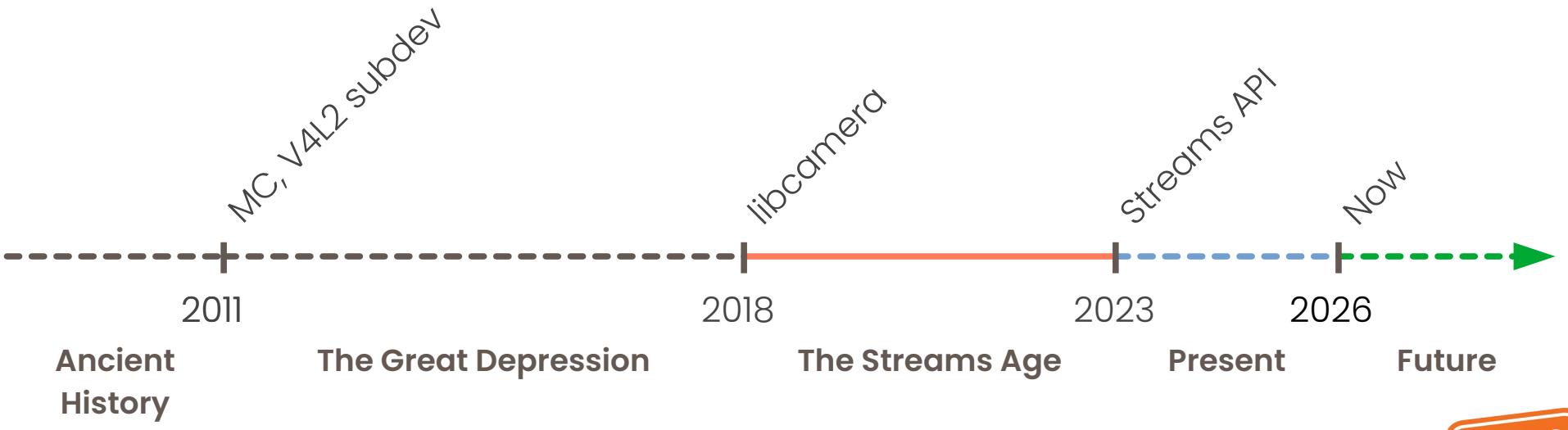
The Streams Age started five years of combined kernel and userspace development effort to upstream ISP support in Linux and libcamera. This has led to the discovery of many V4L2 shortcomings.



The Streams Age

The Streams API

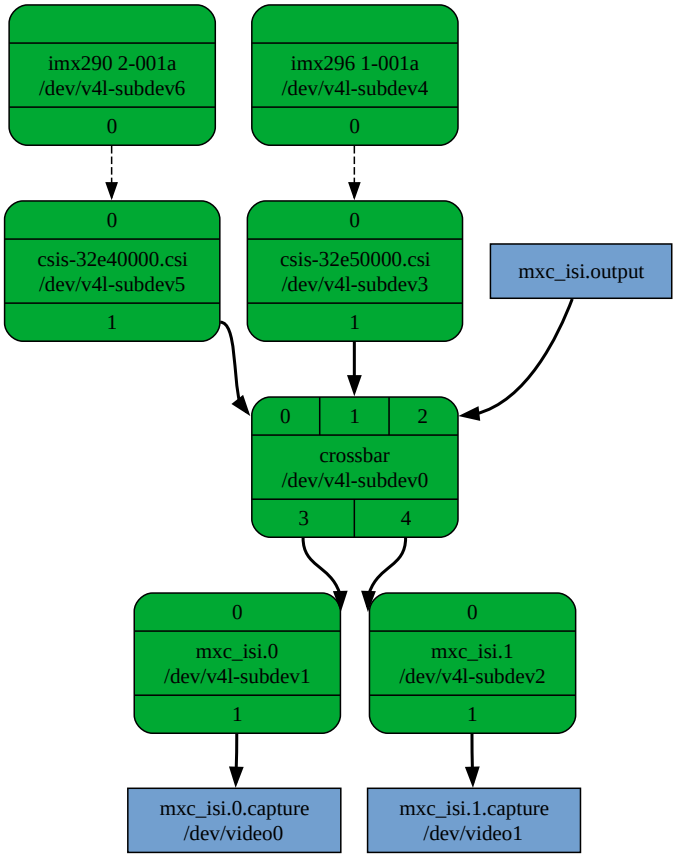
The Streams Age culminated with the merge of the V4L2 Streams API.



The Streams Age

Four years of development

More than 4 years of development was necessary to develop the 16 versions[1] of the streams API and merge it. Three developers took turn driving this project.



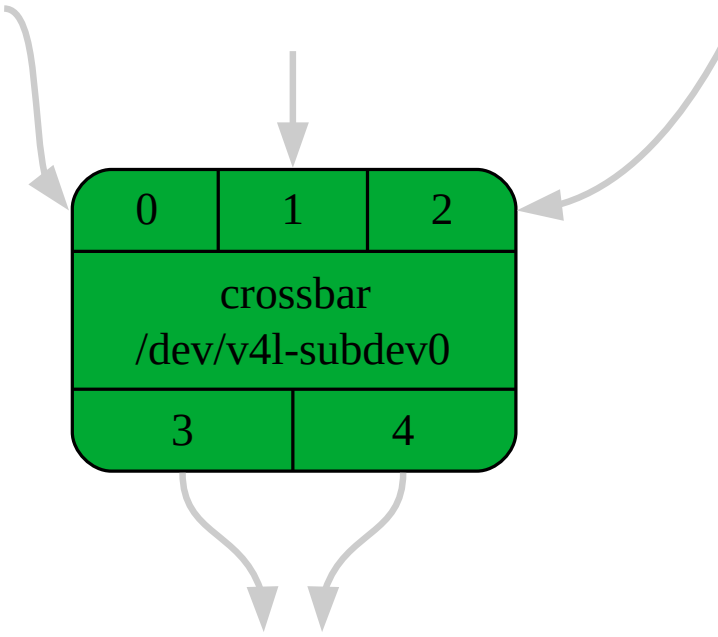
1. Far from the record of 28 versions for a linux-media patch series

Pads, streams & routing

"**A stream is a stream** of content (e.g. pixel data or metadata) flowing through the media pipeline from a source (e.g. a sensor) towards the final sink (e.g. a receiver and demultiplexer in a SoC)."

Streams extend pads to model multiple data flows carried over a link. Each stream on a pad has its own format. Streams routing within a subdev is now exposed and configurable.

Major use cases for streams are CSI-2 DT & VC support and crossbar switches.



The Streams Age

We have more ISPs

- omap3isp (v2.6.39)
- atomisp2 (v4.12)
- ipu3-imgu (v5.0)
- rkisp1 (v5.10)
- microchip-isc (v6.10)
- stf-camss (v6.10)
- sun6i-isp (v6.10)

The Streams Age

IDEAS ON BOARD

We need better ISPs

- omap3isp (v2.6.39) – **no userspace**
- atomisp2 (v4.12) – **staging**, dropped and restored in v5.10
- ipu3-imgu (v5.0) – **staging**
- rkisp1 (v5.10) – support for i.MX8MP in v6.10
- microchip-isc (v6.10) – **in-kernel algorithms**
- stf-camss (v6.10) – **dropped in v7.1**
- sun6i-isp (v6.10) – **no userspace**

The Streams Age

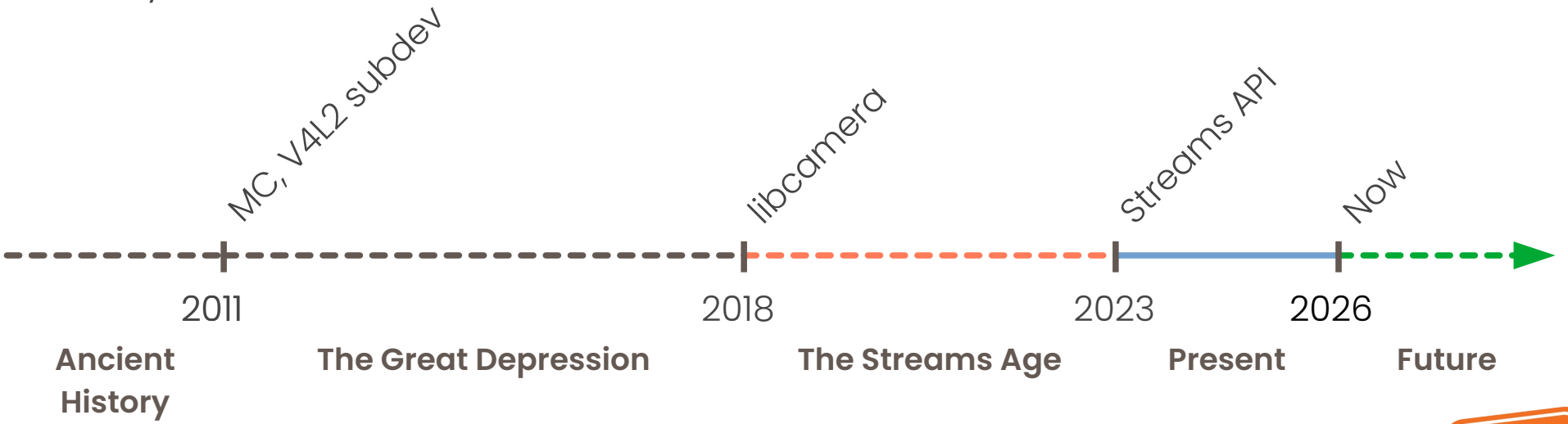
IDEAS ON BOARD

The Present

The present is now

Arbitrarily, the present covers the period from the end of the Streams Age to now.

We are actively working on turning the present into the past, stay tuned.



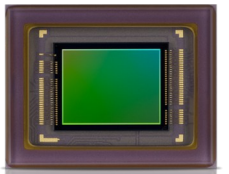
The Present

Image sensors

There is a slow but steady flow of new drivers for image sensors. From v6.10 to now:

- GalaxyCore: GC0310, GC05A2, GC08A3
- Sony: IMX111, IMX214, IMX283
- OmniVision: OG0VE1B, OS05B10, OV02C10, OV02E10, OV2732, OV2735, OV6211
- Samsung: S5K3M5, S5KJN1
- Toshiba: T4KA3
- ST: VD55G1, VD56G3

Most drivers hardcode a **small list of modes**.



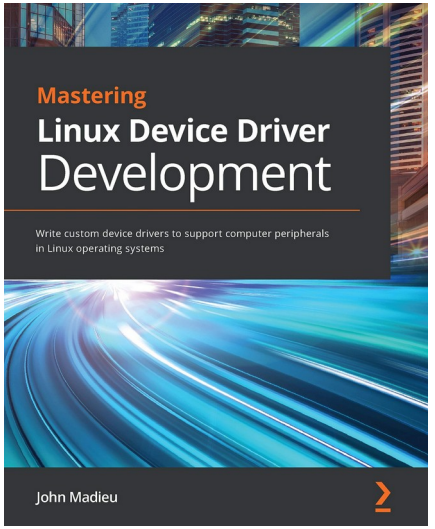
Hardware Support

Image sensor vendors

Traditionally image sensor drivers have been developed and upstreamed by users, with limited (if any) support from vendors (with a notable exception for STMicroelectronics who upstreamed drivers directly).

Over the last few years we have engaged directly with sensor vendors to help them develop and upstream **dynamically configurable** drivers. Results will become increasingly visible over time.

This outreach effort needs to expand, especially towards vendors from Asian countries.

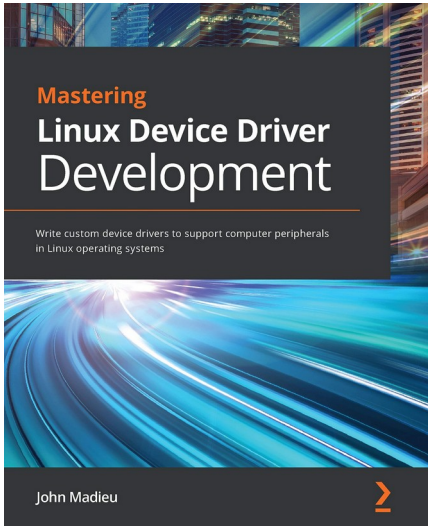


Hardware Support

IDEAS ON BOARD

Image sensor vendors

If sensor vendors can develop and upstream dynamically configurable drivers, we could start rejecting mode-based drivers from other contributors.



Community effort for ISPs

Support for new ISPs is accelerating. Community efforts, largely led by Ideas on Board, resulted in the Arm **mali-c55** driver merged in v6.19 (for the Renesas RZ/V2H SoC).

Two more drivers have been posted to the linux-media mailing list:

- Dream Chip: **rpp-x1** (for Renesas R-Car V4H)
- Rockchip: **rkisp2** (for RK3588)

All these developments received support from the vendors. **libcamera** integration is also available.

Hardware Support

Vendor efforts for ISPs

Following their support for the Pi 4, **Raspberry Pi** has developed drivers for the Pi 5 ISP backend (pisp-be, v6.11) and frontend (rp1-cfe, v6.13). Upstreaming was handled by Ideas on Board.

Amlogic has developed and upstream the C3 ISP driver (c3-isp, v6.16), also with our help.

Development of the i.MX95 neoisp driver was handled autonomously by **NXP**. The driver has been posted and is under review.

libcamera support is included in all those projects.

Vendor efforts for ISPs

In a move that took many by surprise, **Qualcomm** has posted a driver for the CAMSS ISP (camss-isp-ope). Supported features are minimal so far, with high hopes of incremental improvements.

It is fair to say that interest from vendors is growing significantly, but far from being a done deal. More outreach is required, and more outreach is ongoing.

Software ISP

The libcamera software ISP is both a way to support platforms without a software ISP, and a stop-gap solution on platforms where vendors don't want to provide or support an open solution. We have also found it can then provide an incentive for vendors to be more open.

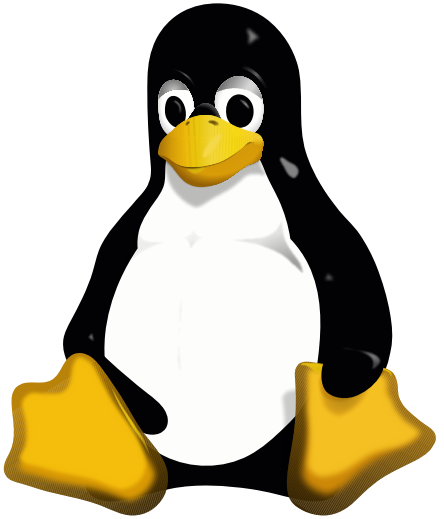
Development of the software ISP is continuing, with a focus on **more algorithms with GPU acceleration**.

Hardware Support

We need more

As before, all those developments have identified many shortcomings of the MC and V4L2 APIs. Several new important API extensions have been proposed:

- Generic line-based metadata
- Internal pads
- Raw sensor model
- Multi-context
- Media jobs

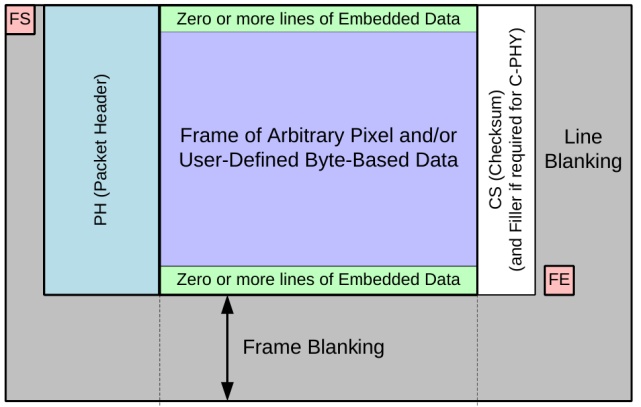


Generic line-based metadata

A raw image sensor can produce embedded data in addition to image data. For CSI-2 sensors, they are two separate streams carried over the same physical output bus.

The embedded data format is sensor-specific, but the data is carried as raw bytes through the camera pipeline. Using sensor-specific media bus codes would require explicit support for all those formats in every CSI-2 receiver driver.

Generic line-based metadata formats solve this issue.

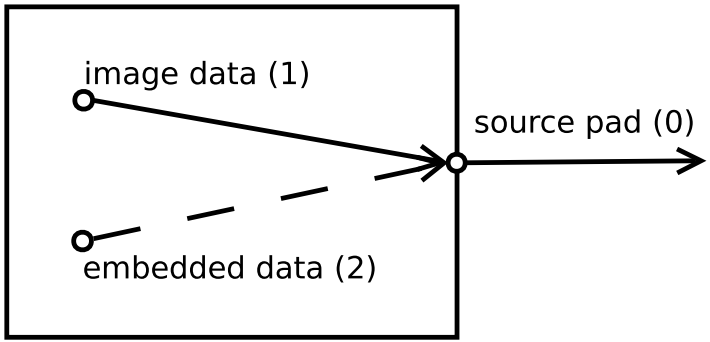


Internal pads

When an entity in a media graph generates multiple streams of data, its single output pad is often not able to expose all the configuration parameters of the device.

Internal pads are an extension of the Media Controller that allows entities to **expose the internal source of each stream**.

A raw image sensor can use internal pads to expose configuration of the image data stream and the embedded data stream.

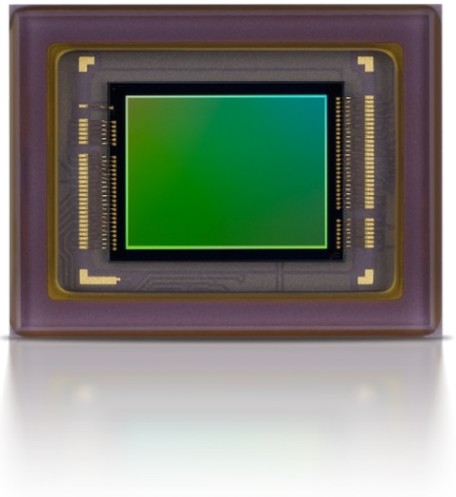


Raw sensor model

The V4L2 subdev API exposes versatile elements such as formats or crop rectangles. The V4L2 specification does not standardise how those elements map to particular device features.

This has led to drivers exposing image sensor features in different ways. The addition of internal pads will only make the issue worse.

The raw sensor model **standardises usage of V4L2 API** elements for raw image sensors, allowing development of generic userspace code.

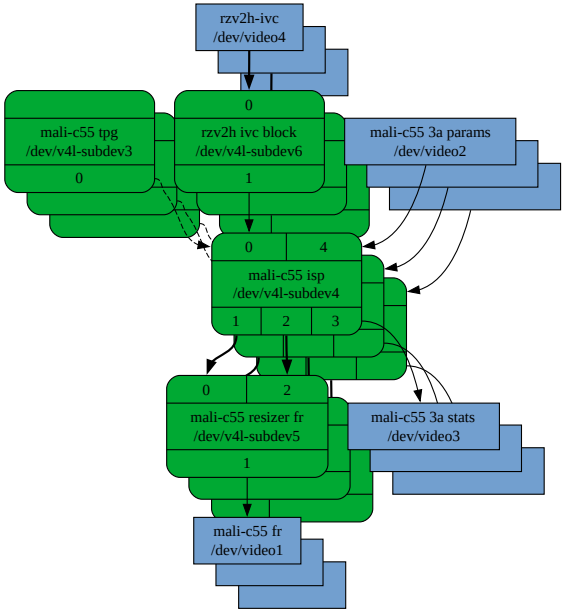


Multi-context support

ISPs operating in memory-to-memory mode are often designed to process multiple cameras in a time-multiplexed fashion.

The V4L2 M2M framework handles time-multiplexing in memory-to-memory operation, but only for drivers that expose a single video device node.

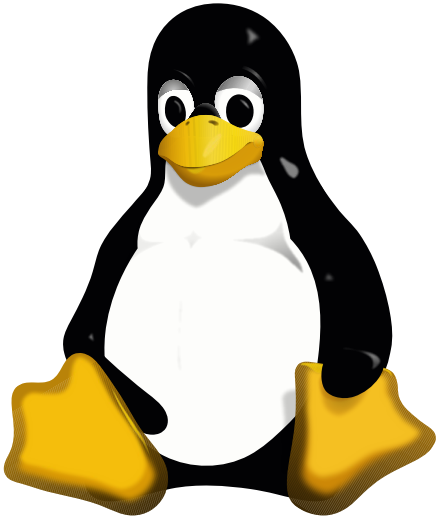
The multi-context API extends time-multiplexing support to a complete media graph with multiple video devices and subdevs.



We need more, faster

Progress to merge those is very slow. More V4L2 shortcomings have been identified already.

V4L2 needs more champions to take ownership of important features and bring them to mainline.



Market-Ready

Despite all the previous shortcomings, ISP support in the Linux kernel and libcamera is market-ready. The user base and community are large, and professional-grade support is available.

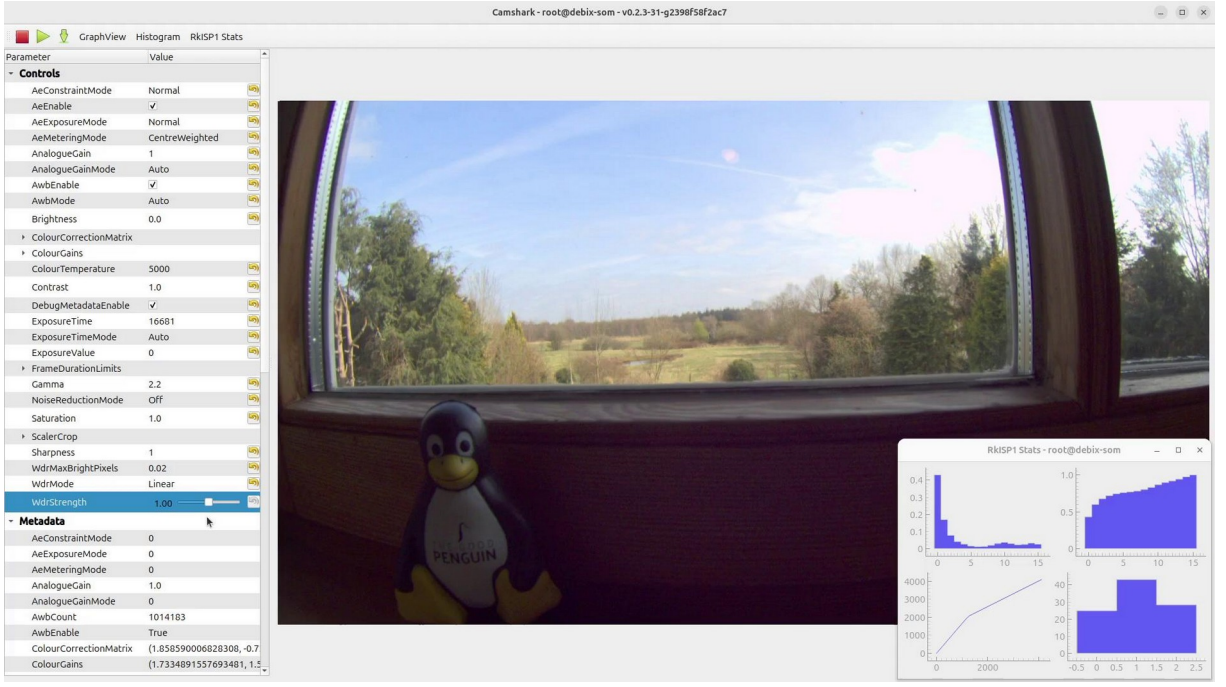
IDEAS ON BOARD

The Present

Camshark

Camshark is a test, debug and development application for libcamera.

Stay for the next talk for a presentation of this awesome tool by Stefan Klug.

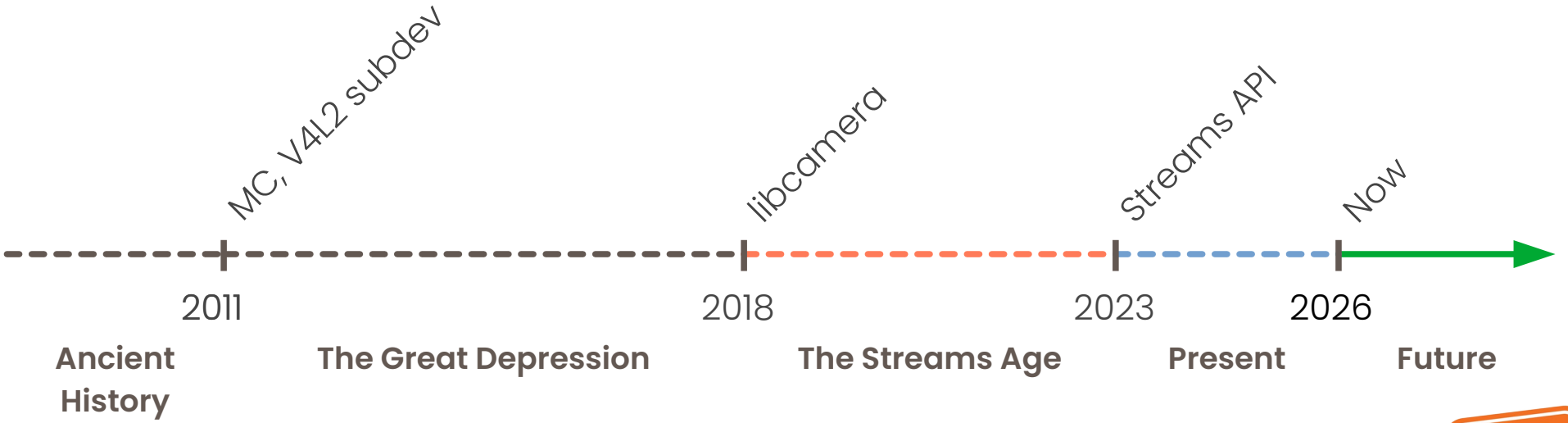


Tooling

The Future

The future starts now

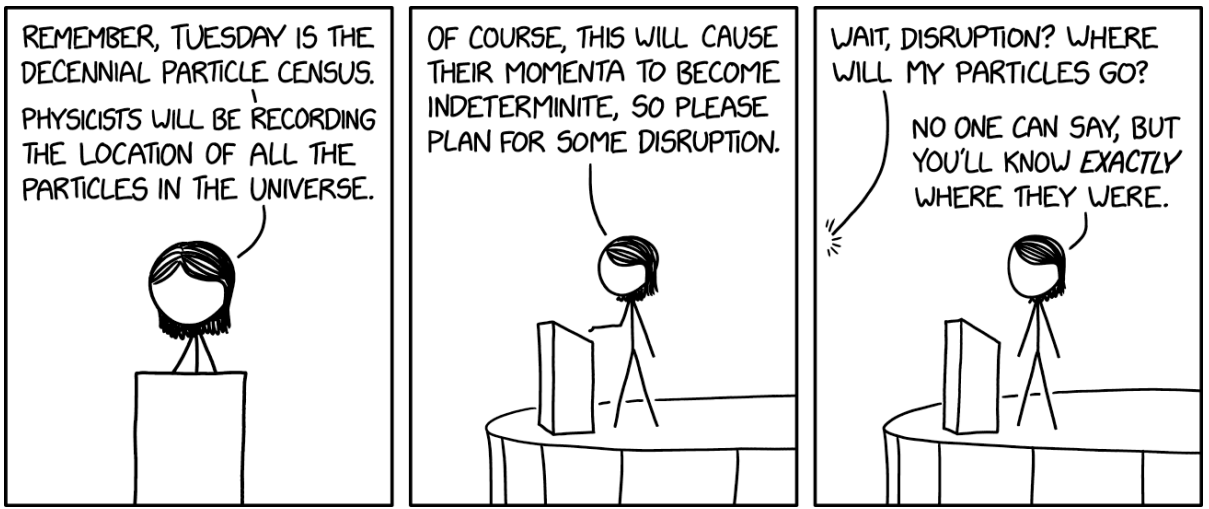
This title contradicts the previous slide that stated “The present is now”. We apologize for any negative impact this may have on space-time, causality and the fabric of the universe.



The Future

A divination exercise

Information in this section of the talk ranges from correct by sheer luck to complete misjudgment. Use at your own risk.



https://xkcd.com/3247/

“New” MIPI standards

Most MIPI CSI-2 camera systems support in Linux are based on MIPI D-PHY. Other MIPI standards exist:

- C-PHY
- I3C
- A-PHY

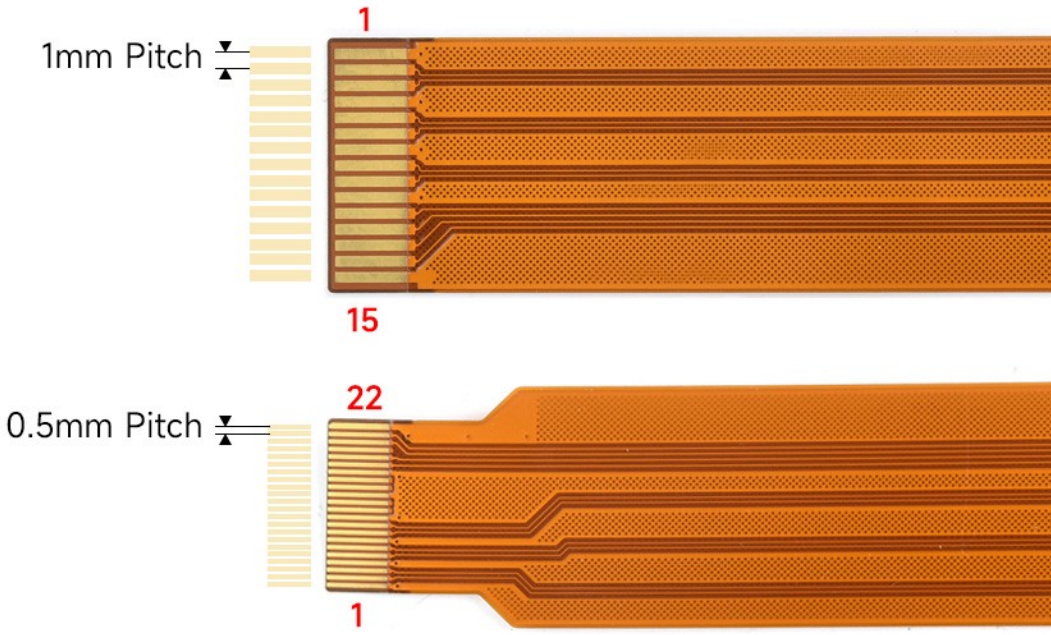
We will see a rapid increase in support for C-PHY. Adoption of I3C or A-PHY for image transfer will be slower.

Cursed connectors

The industry has not standardized any connector for MIPI CSI-2 camera modules.

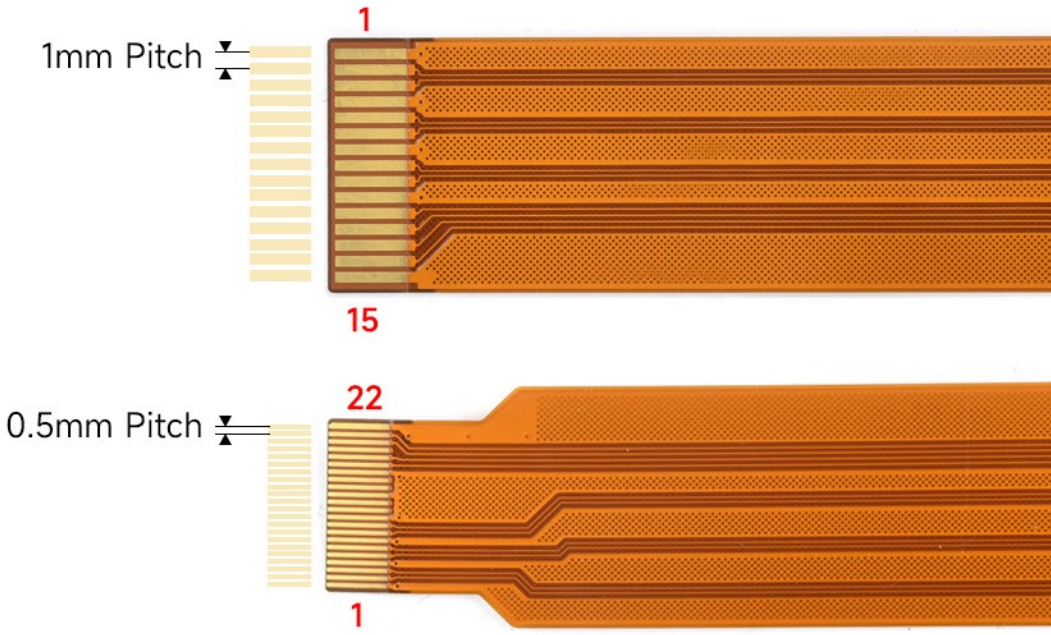
Many vendors have adopted the Raspberry Pi **22-pin camera connector**.

We will likely see this trend continuing, but not for all cameras as more pins are needed to support trigger and flash.



Cursed connectors

Mind the pinout: The Raspberry Pi schematics pin numbers are inverted compared to the connector vendor pin numbers.

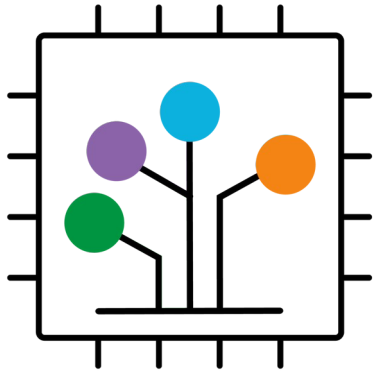


IDEAS ON BOARD

Cursed connectors, take 2

Devices that support interchangeable camera modules often use DT overlays to describe the camera. The same camera module connected to a different board needs a different overlay, leading to combinatorial explosion of overlays.

Hervé Codina and Luca Ceresoli have proposed a **connector abstraction**[1,2] for devicetree that would solve this problem. Progress is slow as the problem is complex, but everybody wants a solution.



1. https://bootlin.com/pub/conferences/2025/elce/ceresoli-hotplug-status.pdf
2. https://lore.kernel.org/devicetree-compiler/20260112142009.1006236-1-herve.codina@bootlin.com/

Serializers

As the number of cameras in a system increases, serialization technologies (**GMSL, FPD-Link, MIPI A-PHY**, ...) will be used more widely. Better support in the kernel and in libcamera will be required.

Fault-tolerance will also become a more important topic.

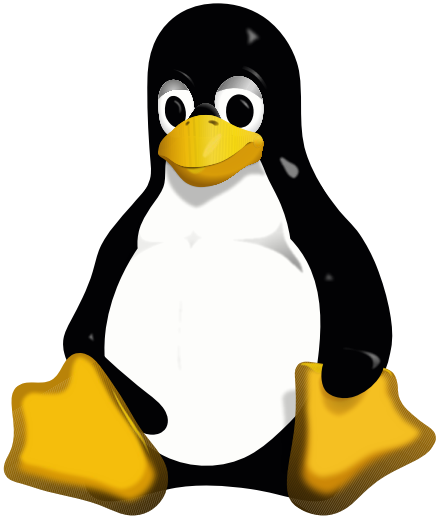
Integration

Kernel Frameworks & APIs

More shortcomings of V4L2 have already been identified:

- Time-multiplexed operation will require a job scheduler
- V4L2 controls will need to be stored in state objects alongside formats
- Extending subdev states will pave the way to an atomic API with a single ioctl
- Media graph will expand to cover multiple devices.

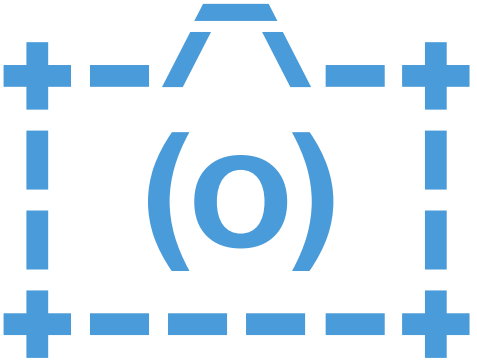
V4L2 **really** needs more champions.



Major refactoring

libcamera is seeing major refactoring of pipeline handlers and IPA modules to make them more modular and facilitate integration of new platforms.

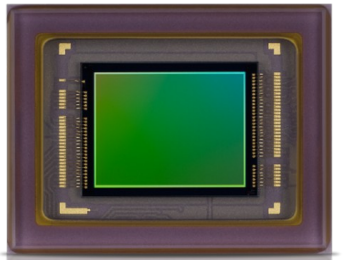
We will start seeing **heterogenous pipelines** that integrate ISP, DSP, GPU and NPU.

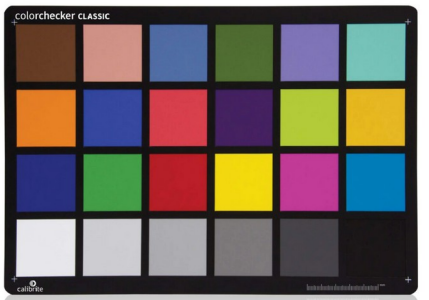


Collaboration with vendors

Vendor interest in upstreaming will increase. Outreach efforts will expand, and many people will wonder why they haven't embraced libcamera much earlier.

Emerging areas (such as MIPI A-PHY) will require time to establish communication channels and trust with vendors.





Conclusion

A brighter future

Mainline drivers help integrators, they help vendors too.

Generic and modular frameworks, in the kernel and libcamera, are of tremendous value.

Mainline camera stacks have shipped in devices

The future of cameras on Linux is libcamera.

Conclusion

Demo

IDEAS ON BOARD