```
   _
+-/ \-+
| (o) |
+-----+
```

# Application support with libcamera

Ubuntu Developer Summit
Prague 2022

Kieran Bingham
kieran.bingham@ideasonboard.com

```
   _
+-/ \-+
| (o) |
+-----+
```

- Complex Cameras
- Complications
- Existing Solutions
- Community Support
- Future Developments
- Q+A

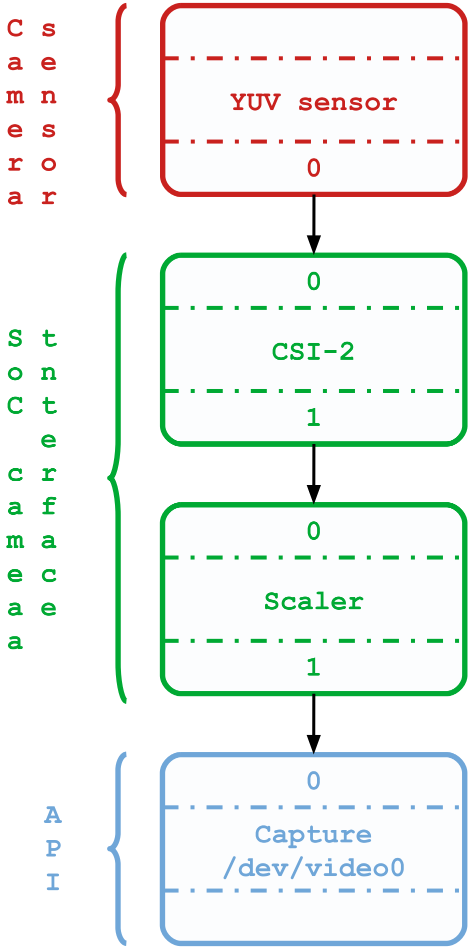
# Legalese…

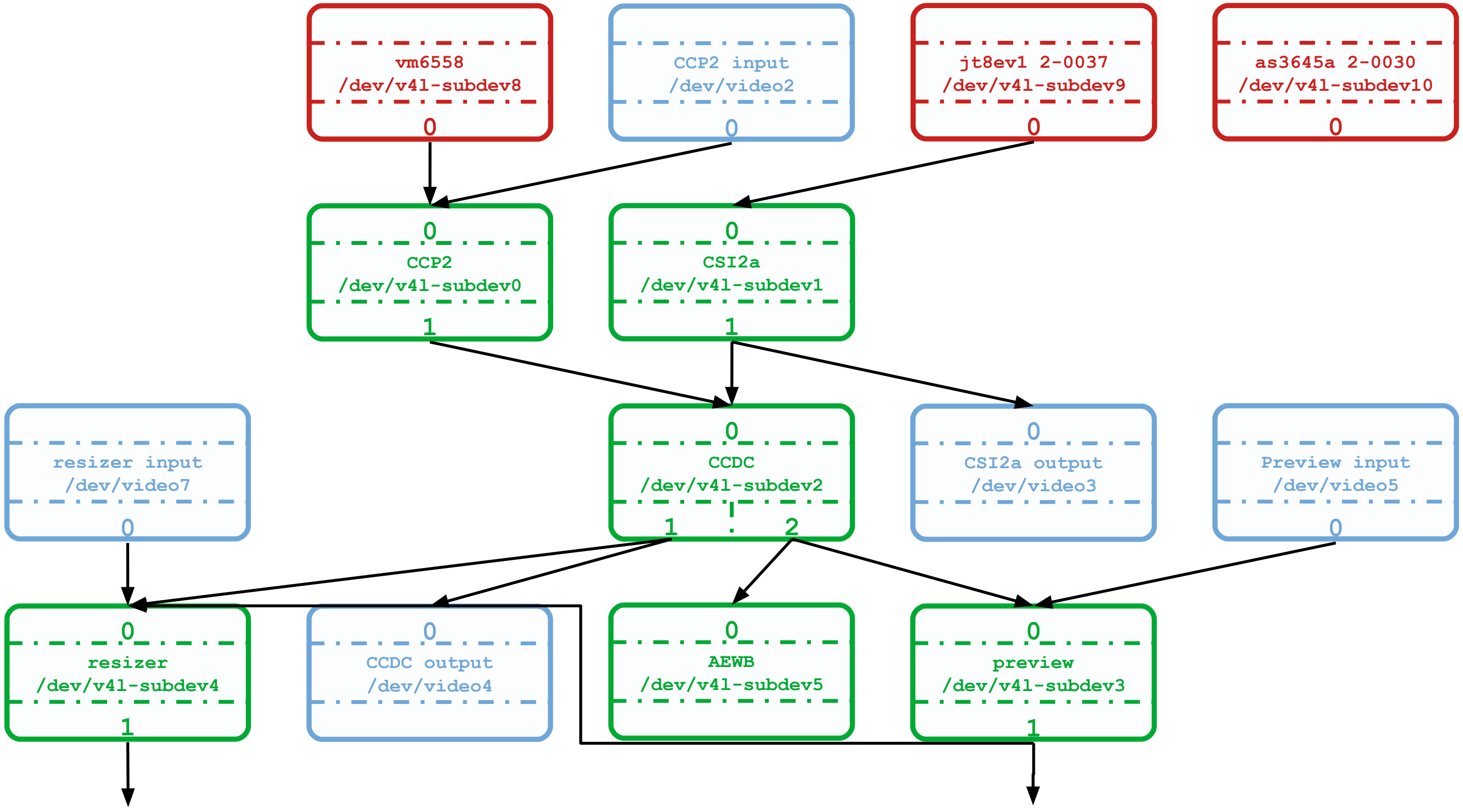
- This talk references many external projects, packages, and software.
- All logos, trademarks, and other identifying marks belong to their respective owners.
- In many places I’m talking about work that /other/ people have done or contributed
    - No intention or desire to take credit for their work

```
   _
+-/ \-+
| (o) |
+-----+
```

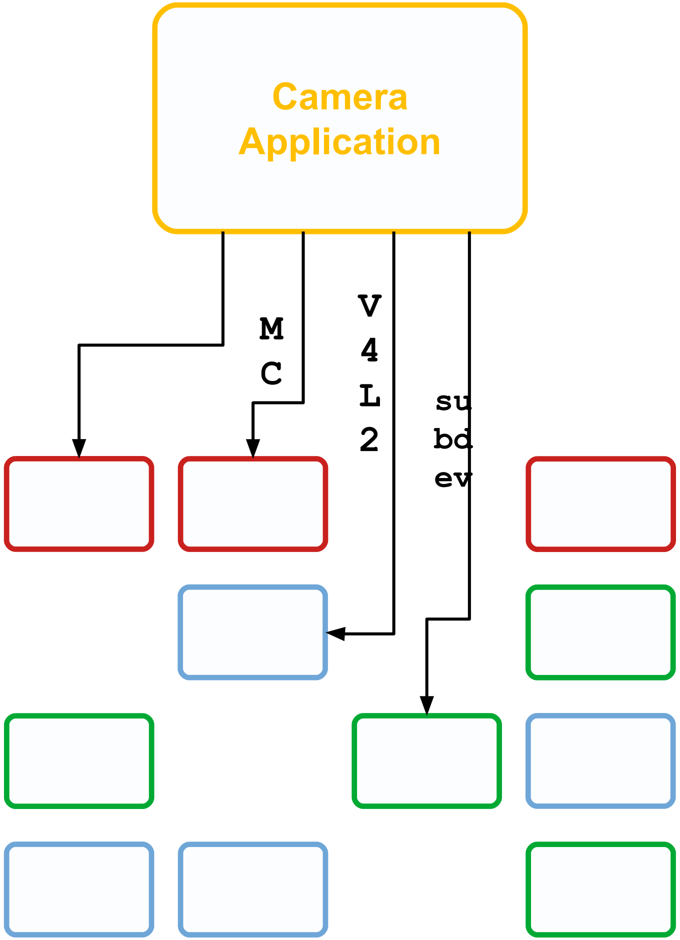
- **Complex Cameras**
- Complications
- Existing Solutions
- Community Support
- Future Developments
- Q+A

# Cameras are Complex

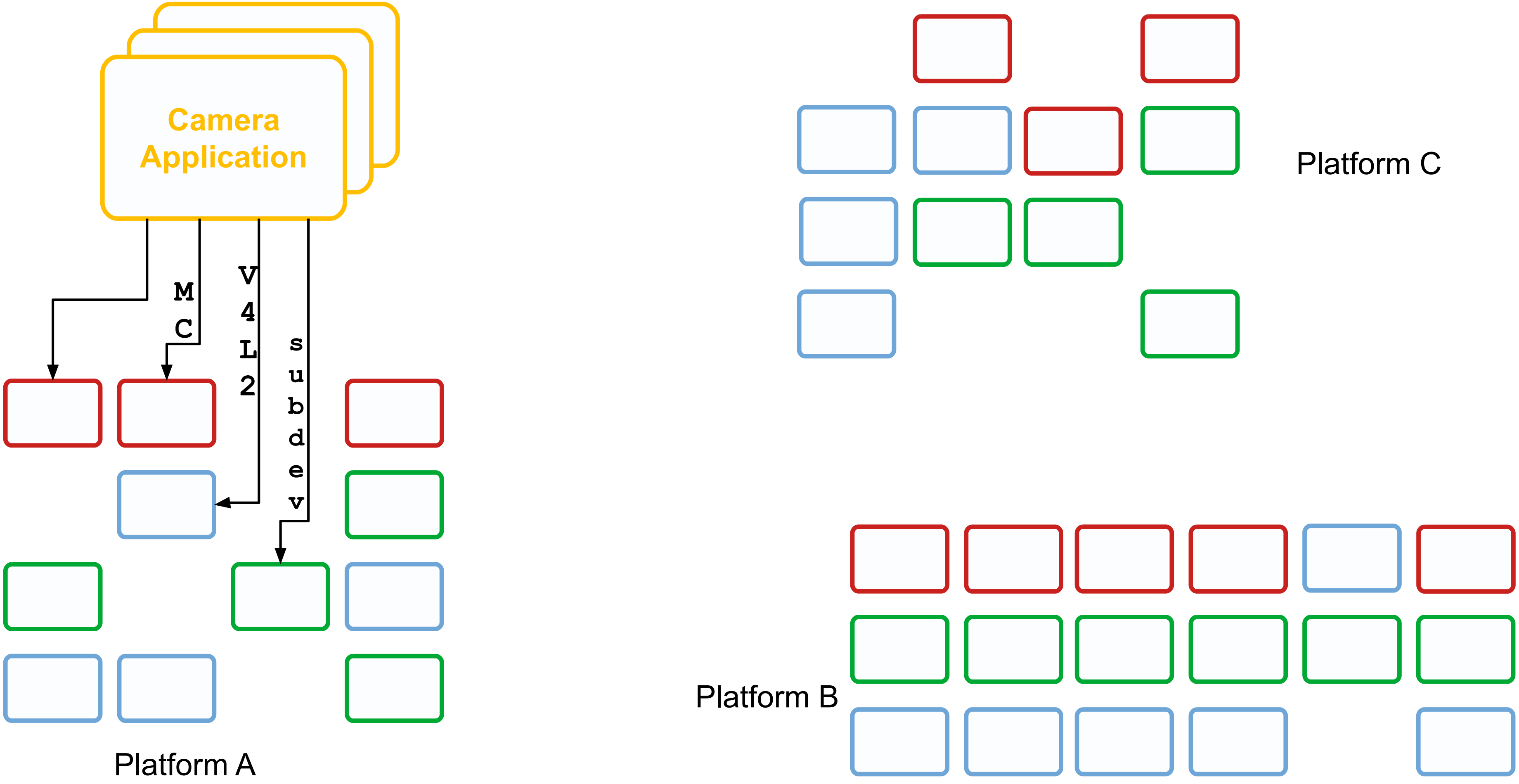




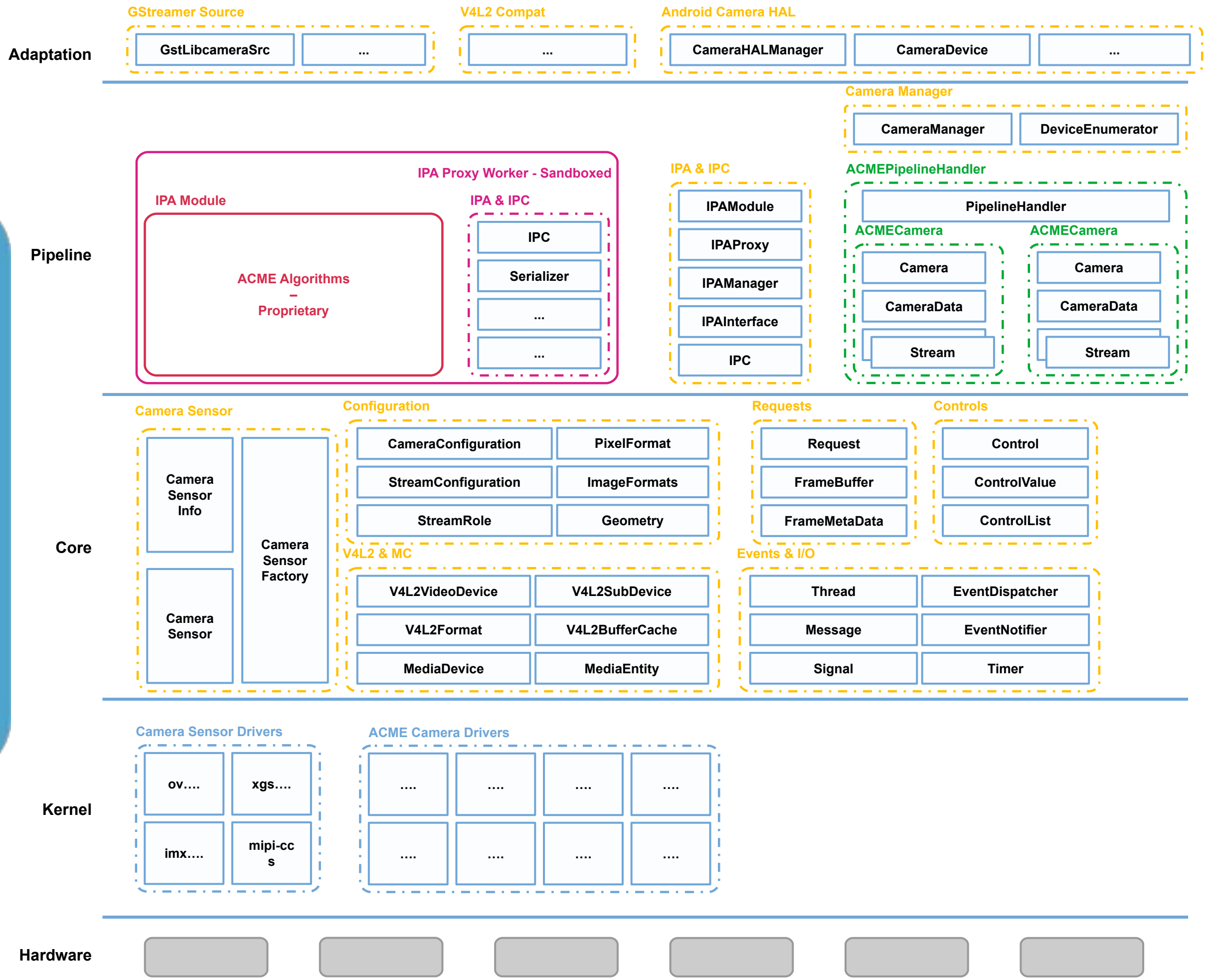
OMAP3 Camera in Nokia N900 - 2009



# Applications can manage those complexities …



# But it doesn’t scale



# libcamera fills that gap

```
   _
+-/ \-+
| (o) |
+-----+
```

- Complex Cameras
- **Complications**
- Existing Solutions
- Community Support
- Future Developments
- Q+A

# Everybody loves V4L2 *

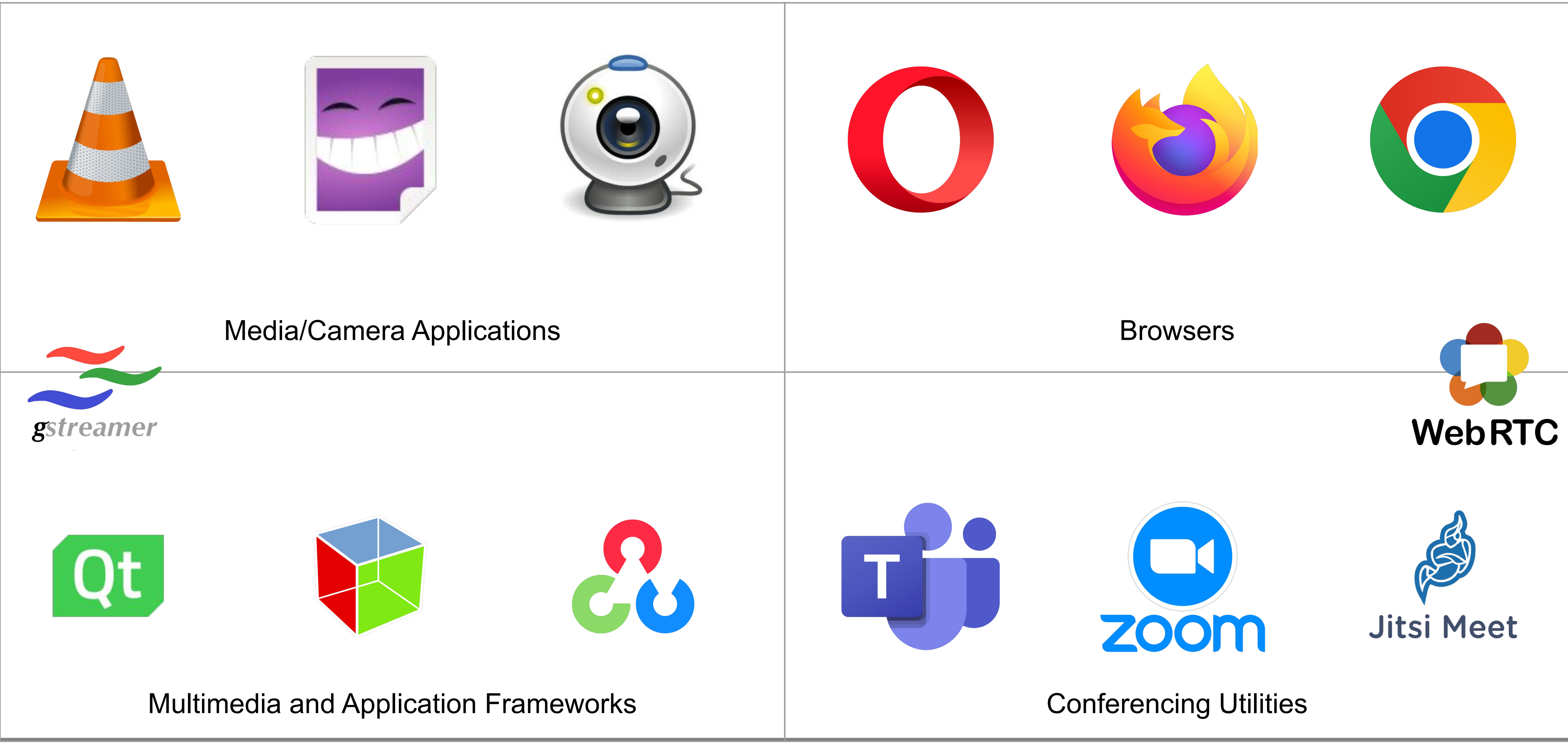
- V4L2 started between 1998, 2002

- Consistent API supports many Video capture devices
  - Wide support of pixel formats …
  - Same API for (existing, simple) Cameras, Digital TV DVB, Set Top Box …

- It exists
  - There hasn’t been anything else at the kernel level (upstream)

- Widely used (thoroughly tested)

All logos subject to copyright by their respective projects and owners, used for reference and indicative purposes only



Media/Camera Applications

Browsers

Multimedia and Application Frameworks

Conferencing Utilities

# V4L2 is already used everywhere

# Not everybody loves V4L2 *

- Really - they love ‘/dev/video0’

- Media Controller
  - Entities, links, pads, format negotiation (propagation)

- Subdevices
  - Direct control over specific internal components
  - Which one do you configure?

- Multiple video nodes for a single “Camera device”
  - UVC - ‘Metadata video node’
  - CSI2 Receiver
  - ISP - Statistics, Parameter Buffers, Multiple image streams
  - M2M Dewarper

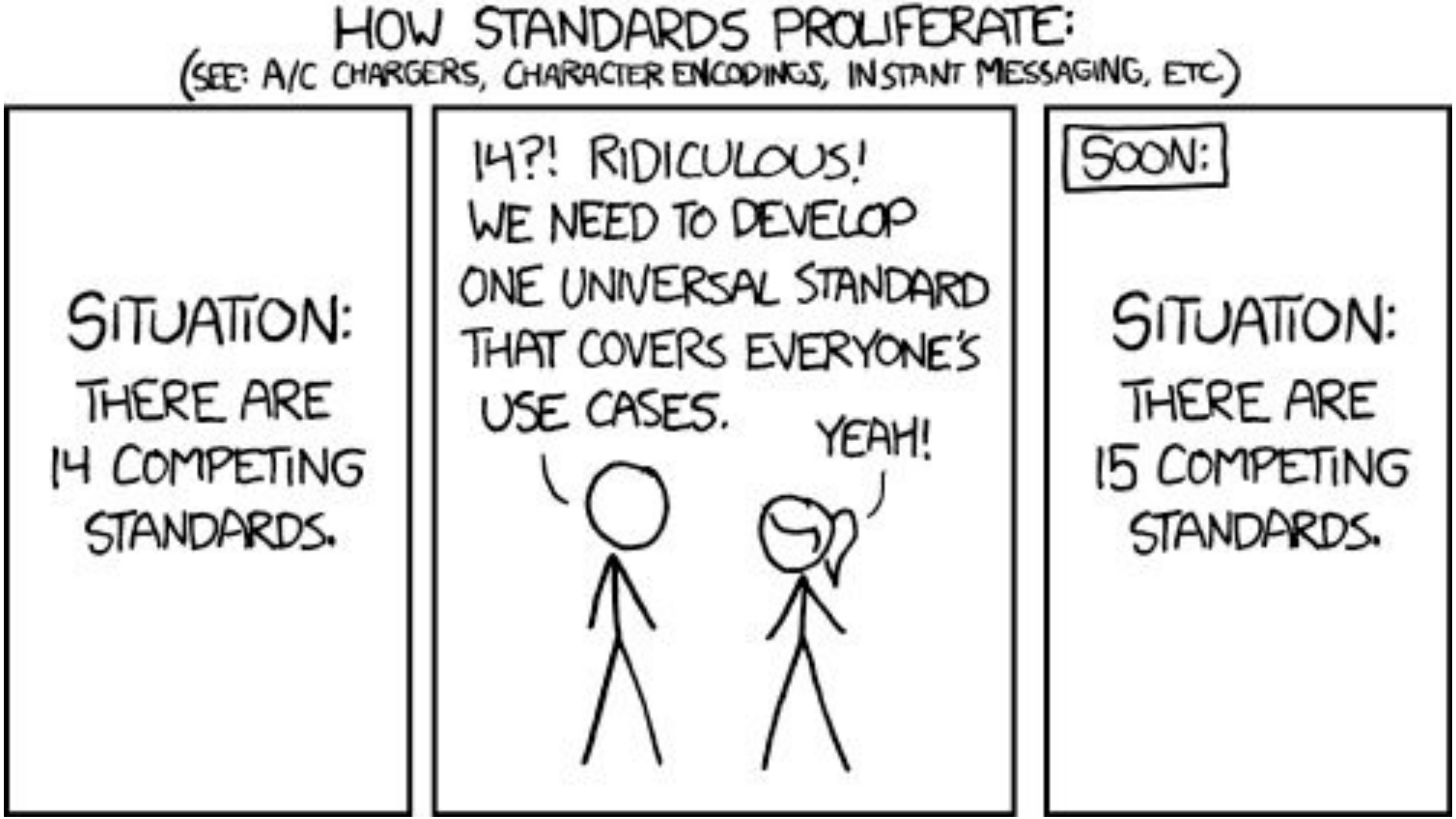
- 3A algorithms need to be handled in userspace
    - Crucial for RAW sensors. YUV sensors are becoming obsolete
- Laptops are now using complex cameras
    - Dell, Lenovo, HP, Surface … (Intel IPU3, IPU4, IPU6)
- Embedded devices already use complex cameras
    - OEM/ODM … need custom solutions to manage each camera
- No portable mobile camera applications.
    - Mobile is dominated by Android, with mostly binary camera stacks
- Ubuntu Touch / UBPorts
    - Targeting a new life for old devices … with cameras

# V4L2 alone isn’t enough

https://www.emersonhc.com/change-management/people-hard-wired-resist-change

- Many applications don’t yet support libcamera
- Adding the support takes effort
  - maintainers haven’t expected this
- C applications don’t want to use C++
  - People can be scared of the ++
  - Development on ‘C / Rust’ bindings begun
- Is it even finished yet?
  - Releases, ABI stability …

# But … now there’s a new API to use


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

# Obligatory XKCD : 927

```
   _
+-/ \-+
| (o) |
+-----+
```

- Complex Cameras
- Complications
- **Existing Solutions**
- Community Support
- Future Developments
- Q+A

# libcamera provides a gstreamer element

Gstlibcamerasrc brings the whole Gstreamer ecosystem to libcamera devices:

- Encoding / Streaming
- Composing / Mixing
- Audio

Camera Viewer

```
gst-launch-1.0 libcamerasrc ! 'video/x-raw,width=1280,height=720' ! glimagesink
```

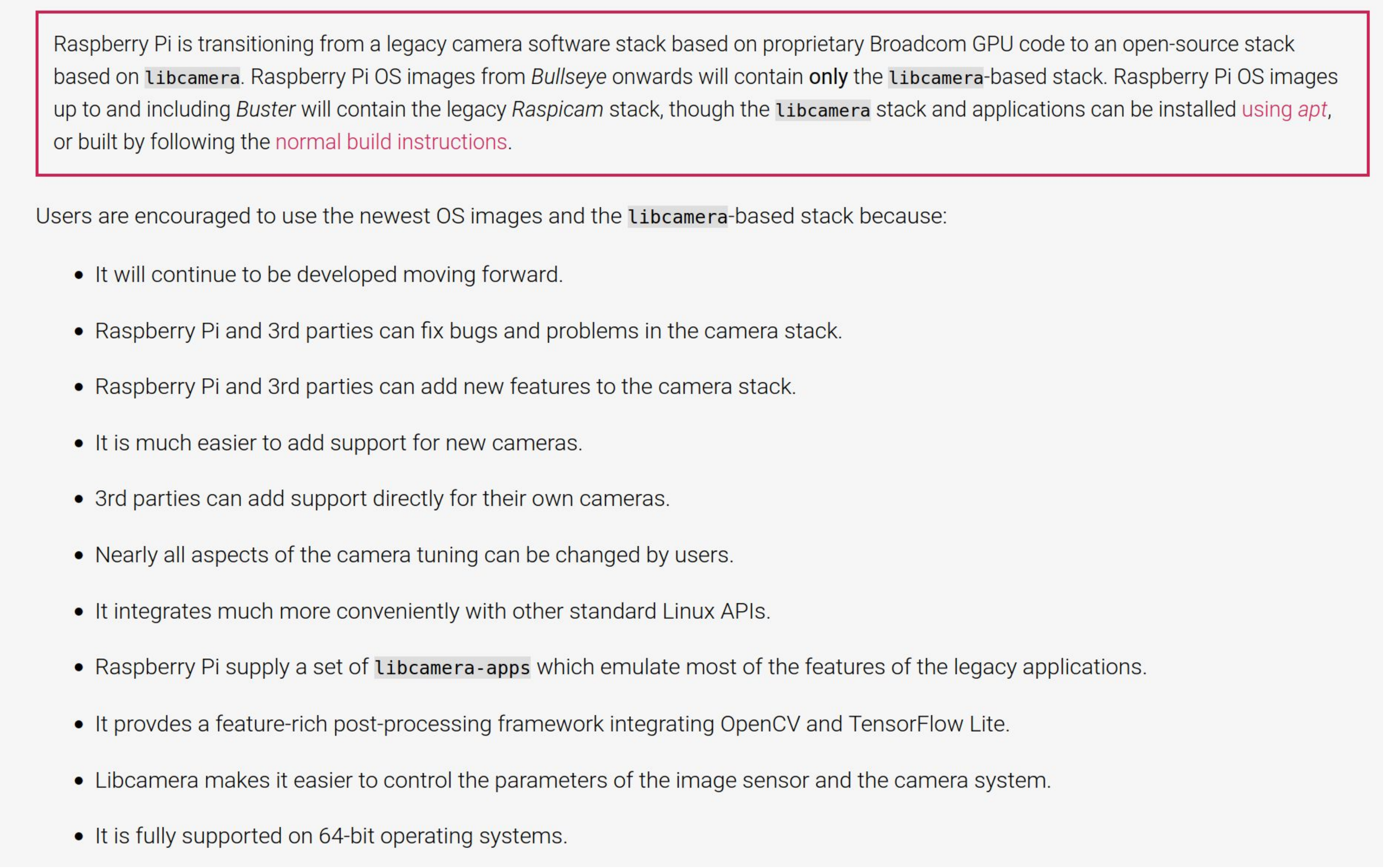
JPEG Network streamer

```
gst-launch-1.0 libcamerasrc ! \
        video/x-raw,colorimetry=bt709,format=NV12,width=1280,height=720,framerate=30/1 ! \
        jpegenc ! multipartmux ! \
        tcpserversink host=0.0.0.0 port=5000
```

JPEG Network Receiver

```
gst-launch-1.0 tcpclientsrc host=$DEVICE_IP port=5000 ! \
        multipartdemux ! jpegdec ! autovideosink
```

# Raspberry Pi provide their own implementation



Raspberry Pi is transitioning from a legacy camera software stack based on proprietary Broadcom GPU code to an open-source stack based on `libcamera`. Raspberry Pi OS images from *Bullseye* onwards will contain **only** the `libcamera`-based stack. Raspberry Pi OS images up to and including *Buster* will contain the legacy *Raspicam* stack, though the `libcamera` stack and applications can be installed using *apt*, or built by following the normal build instructions.

Users are encouraged to use the newest OS images and the `libcamera`-based stack because:

- It will continue to be developed moving forward.
- Raspberry Pi and 3rd parties can fix bugs and problems in the camera stack.
- Raspberry Pi and 3rd parties can add new features to the camera stack.
- It is much easier to add support for new cameras.
- 3rd parties can add support directly for their own cameras.
- Nearly all aspects of the camera tuning can be changed by users.
- It integrates much more conveniently with other standard Linux APIs.
- Raspberry Pi supply a set of `libcamera-apps` which emulate most of the features of the legacy applications.
- It provdes a feature-rich post-processing framework integrating OpenCV and TensorFlow Lite.
- Libcamera makes it easier to control the parameters of the image sensor and the camera system.
- It is fully supported on 64-bit operating systems.

https://www.raspberrypi.com/documentation/accessories/camera.html

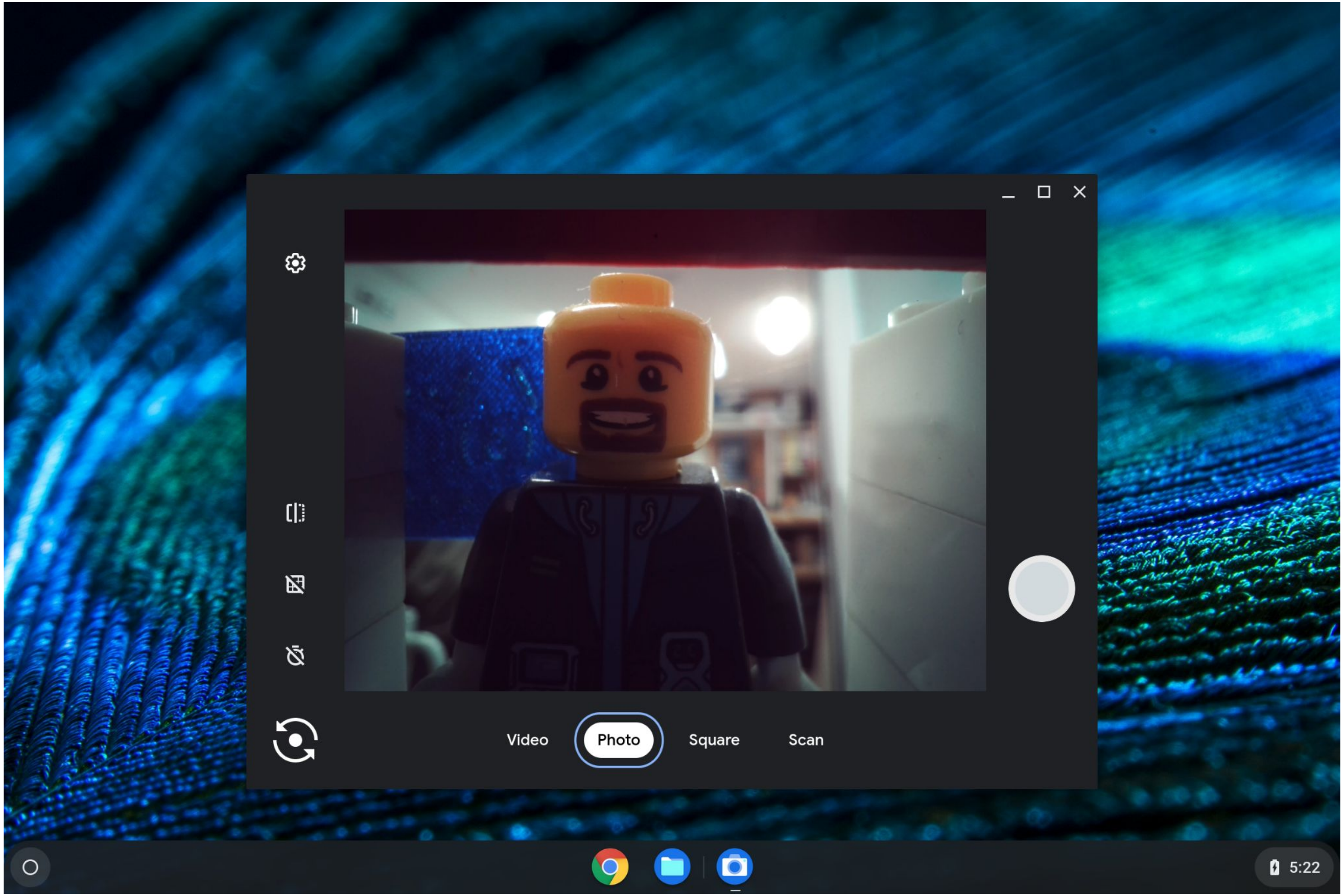
# We have python support

- Few users like C++

- Python is “friendly”
- Fast to prototype

- ‘Picamera2’ simplifies the libcamera python API

```python
#!/usr/bin/python3

# Capture a JPEG while still running in the preview mode. When you
# capture to a file, the return value is the metadata for that image.

import time

from picamera2 import Picamera2, Preview

picam2 = Picamera2()

preview_config = picam2.create_preview_configuration(main ={"size": (800, 600)})
picam2.configure(preview_config)

picam2.start_preview(Preview.QTGL)

picam2.start()
time.sleep(2)

metadata = picam2.capture_file( "test.jpg")
print(metadata)

picam2.close()
```

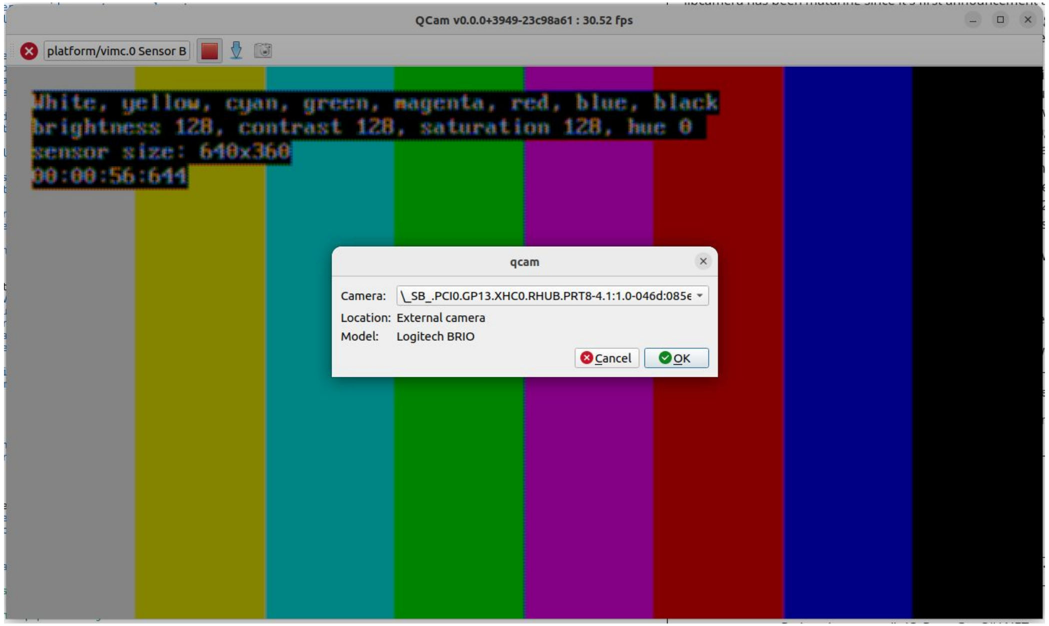


# An android HAL implementation

- LD_PRELOAD solution
  - LIBCAMERA_PUBLIC int **open**(const char *path, int oflag, ...)
  - LIBCAMERA_PUBLIC int open64(const char *path, int oflag, ...)
  - LIBCAMERA_PUBLIC int openat(int dirfd, const char *path, int oflag, ...)
  - LIBCAMERA_PUBLIC int __openat_2(int dirfd, const char *path, int oflag)
  - LIBCAMERA_PUBLIC int openat64(int dirfd, const char *path, int oflag, ...)
  - LIBCAMERA_PUBLIC int **dup**(int oldfd)
  - LIBCAMERA_PUBLIC int **close**(int fd)
  - LIBCAMERA_PUBLIC void ***mmap**(void *addr, size_t length, int prot, int flags,
    int fd, off_t offset)
  - LIBCAMERA_PUBLIC void *mmap64(void *addr, size_t length, int prot, int flags,
    int fd, off64_t offset)
  - LIBCAMERA_PUBLIC int **munmap**(void *addr, size_t length)
  - LIBCAMERA_PUBLIC int **ioctl**(int fd, unsigned long request, ...)
- **‘libcamerify’**
  - $ libcamerify -d -d myV4L2Application –myArgs

# … and a V4L2 compatibility layer

# As well as test applications

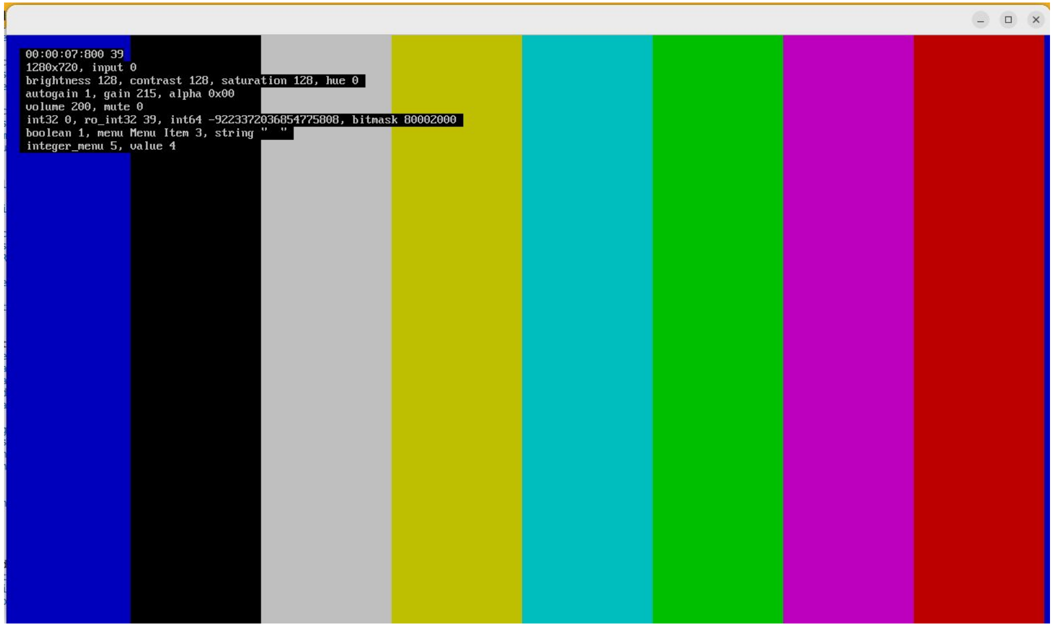


```
White, yellow, cyan, green, magenta, red, blue, black
brightness 128, contrast 128, saturation 128, hue 0
sensor size: 640x360
00:00:56:644
```

Camera: _SB_.PCI0.GP13.XHC0.RHUB.PRT8-4.1:1.0-046d:085e

Location: External camera

Model: Logitech BRIO



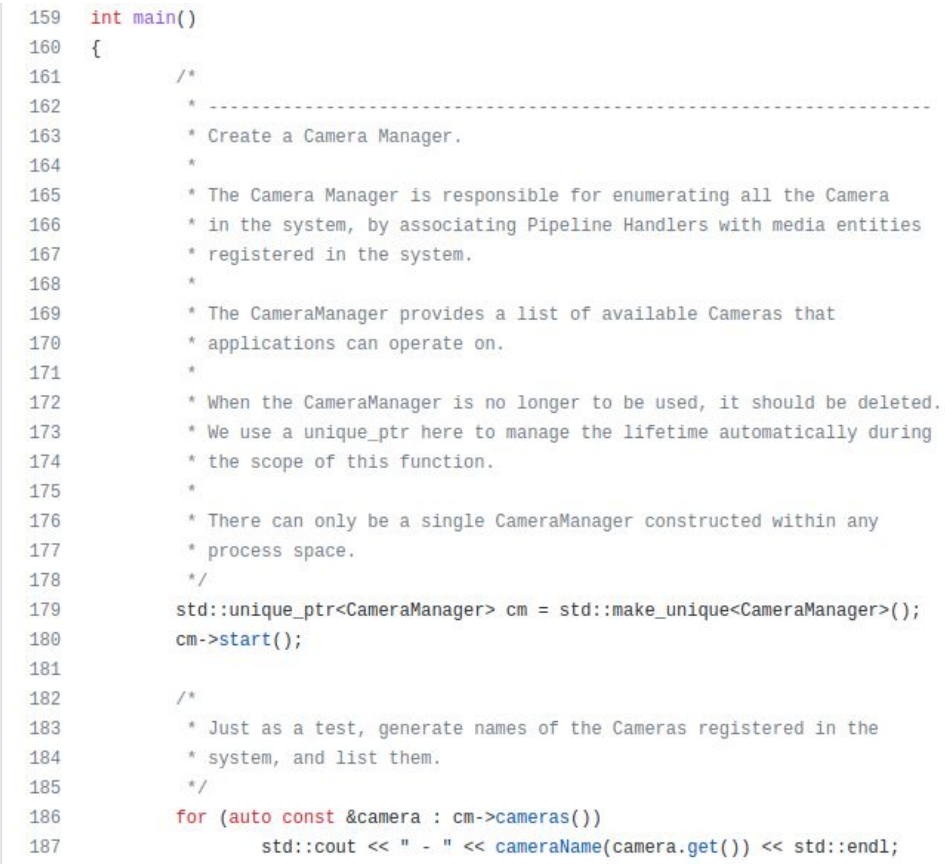
```
00:00:07:800 39
1280x720, input 0
brightness 128, contrast 128, saturation 128, hue 0
autogain 1, gain 215, alpha 0x00
volume 200, mute 0
int32 0, ro_int32 39, int64 -9223372036854775808, bitmask 80002000
boolean 1, menu Menu Item 3, string "  "
integer_menu 5, value 4
```

```
$ qcam -r gles
```

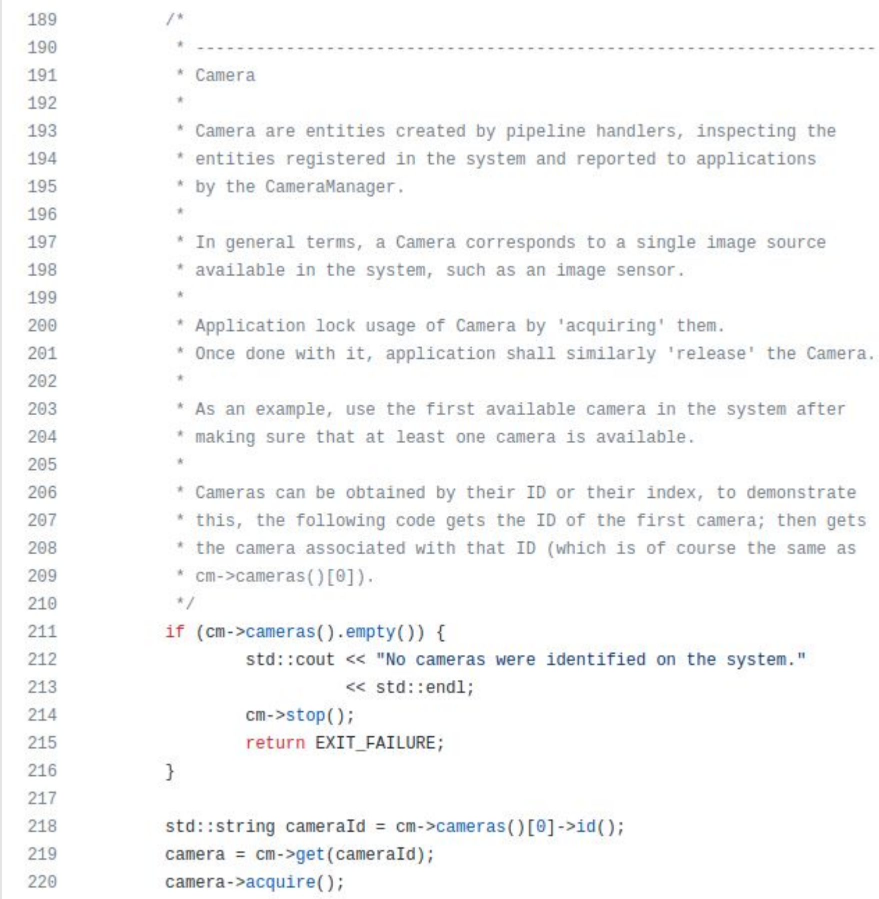
(Using VIMC)

```
$ cam -c3 -C -S –stream pixelformat=YUYV
```

(Using VIVID)



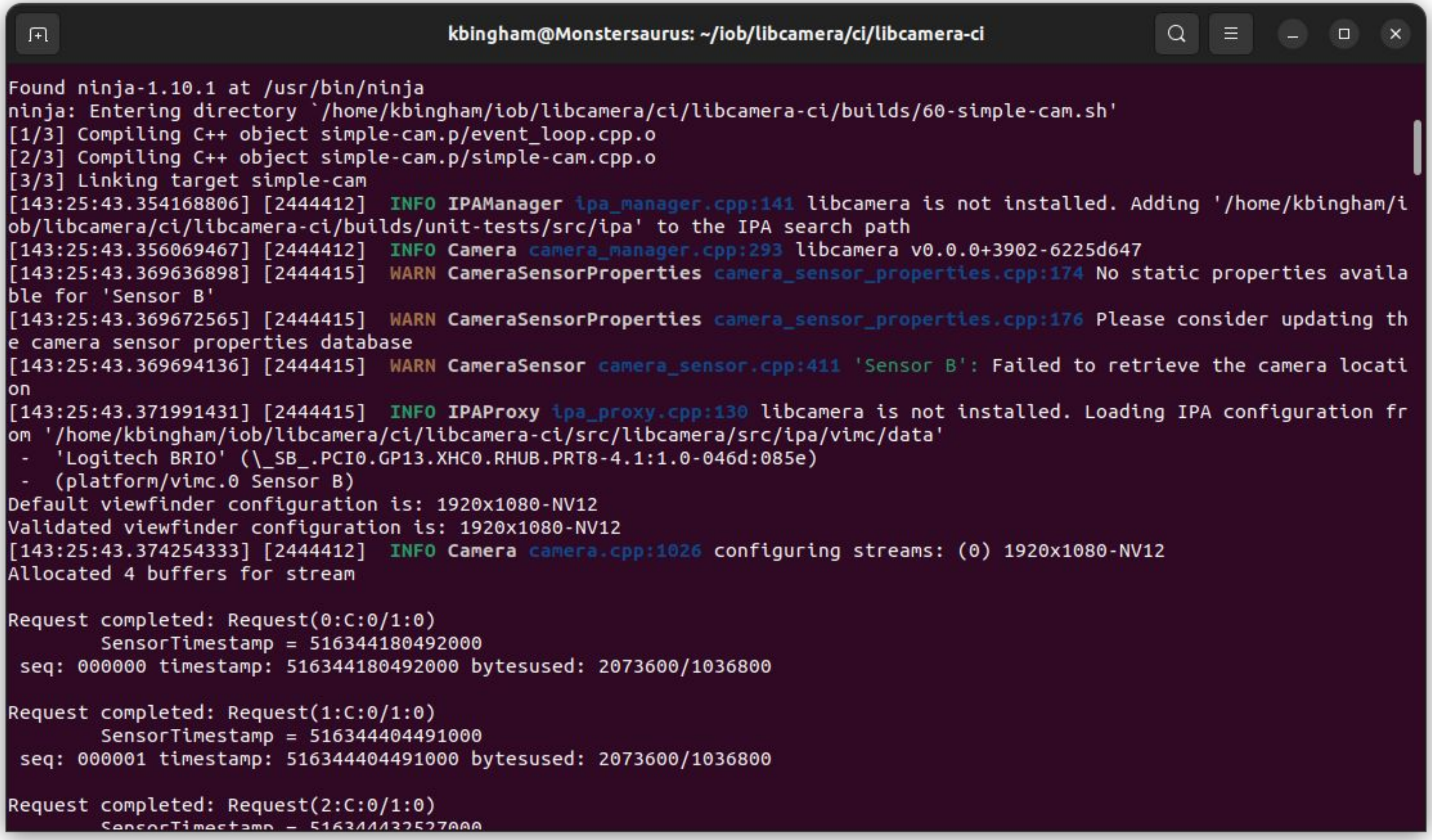
```
int main()
{
	/*
	 * --------------------------------------------------------------------
	 * Create a Camera Manager.
	 *
	 * The Camera Manager is responsible for enumerating all the Camera
	 * in the system, by associating Pipeline Handlers with media entities
	 * registered in the system.
	 *
	 * The CameraManager provides a list of available Cameras that
	 * applications can operate on.
	 *
	 * When the CameraManager is no longer to be used, it should be deleted.
	 * We use a unique_ptr here to manage the lifetime automatically during
	 * the scope of this function.
	 *
	 * There can only be a single CameraManager constructed within any
	 * process space.
	 */
	std::unique_ptr<CameraManager> cm = std::make_unique<CameraManager>();
	cm->start();

	/*
	 * Just as a test, generate names of the Cameras registered in the
	 * system, and list them.
	 */
	for (auto const &camera : cm->cameras())
		std::cout << " - " << cameraName(camera.get()) << std::endl;
```



```
	/*
	 * --------------------------------------------------------------------
	 * Camera
	 *
	 * Camera are entities created by pipeline handlers, inspecting the
	 * entities registered in the system and reported to applications
	 * by the CameraManager.
	 *
	 * In general terms, a Camera corresponds to a single image source
	 * available in the system, such as an image sensor.
	 *
	 * Application lock usage of Camera by 'acquiring' them.
	 * Once done with it, application shall similarly 'release' the Camera.
	 *
	 * As an example, use the first available camera in the system after
	 * making sure that at least one camera is available.
	 *
	 * Cameras can be obtained by their ID or their index, to demonstrate
	 * this, the following code gets the ID of the first camera; then gets
	 * the camera associated with that ID (which is of course the same as
	 * cm->cameras()[0]).
	 */
	if (cm->cameras().empty()) {
		std::cout << "No cameras were identified on the system."
			  << std::endl;
		cm->stop();
		return EXIT_FAILURE;
	}

	std::string cameraId = cm->cameras()[0]->id();
	camera = cm->get(cameraId);
	camera->acquire();
```

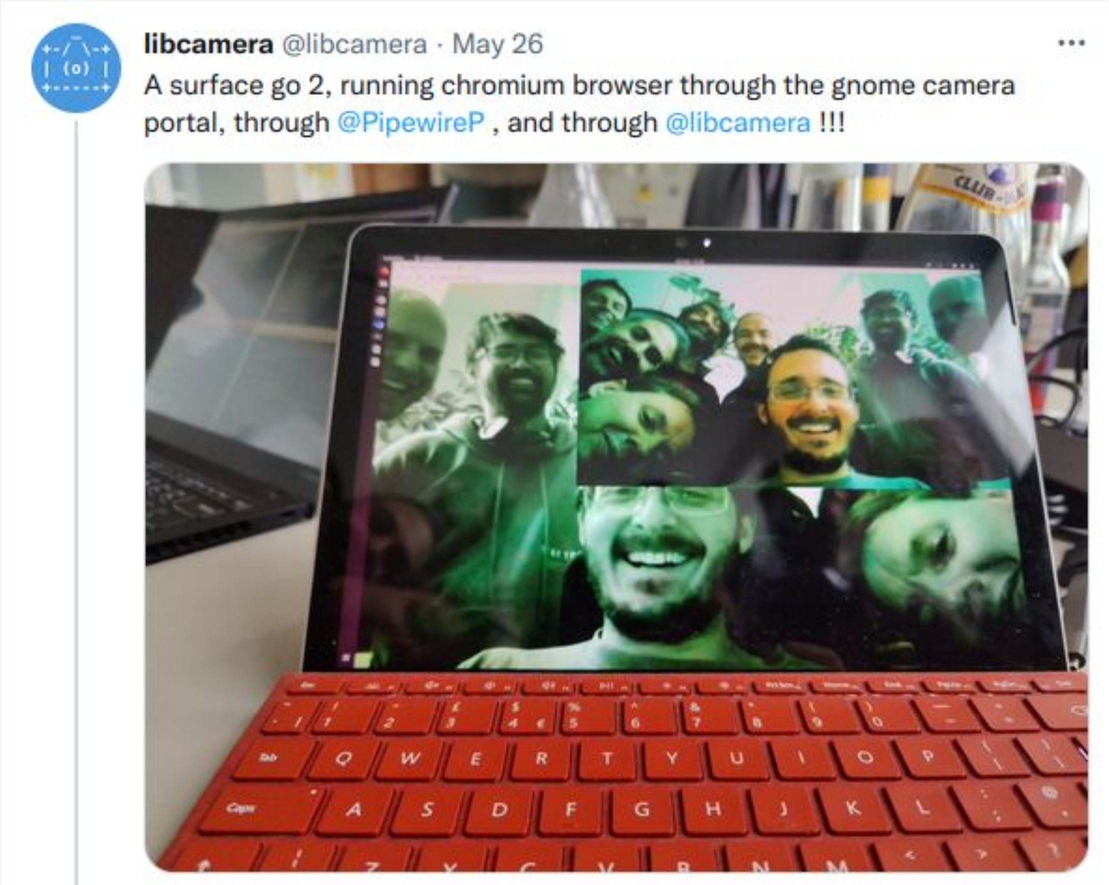
https://git.libcamera.org/libcamera/simple-cam.git/

# And a sample ‘hello world’ for the API



```
Found ninja-1.10.1 at /usr/bin/ninja
ninja: Entering directory `/home/kbingham/iob/libcamera/ci/libcamera-ci/builds/60-simple-cam.sh'
[1/3] Compiling C++ object simple-cam.p/event_loop.cpp.o
[2/3] Compiling C++ object simple-cam.p/simple-cam.cpp.o
[3/3] Linking target simple-cam
[143:25:43.354168806] [2444412]  INFO IPAManager ipa_manager.cpp:141 libcamera is not installed. Adding '/home/kbingham/i
ob/libcamera/ci/libcamera-ci/builds/unit-tests/src/ipa' to the IPA search path
[143:25:43.356069467] [2444412]  INFO Camera camera_manager.cpp:293 libcamera v0.0.0+3902-6225d647
[143:25:43.369636898] [2444415]  WARN CameraSensorProperties camera_sensor_properties.cpp:174 No static properties availa
ble for 'Sensor B'
[143:25:43.369672565] [2444415]  WARN CameraSensorProperties camera_sensor_properties.cpp:176 Please consider updating th
e camera sensor properties database
[143:25:43.369694136] [2444415]  WARN CameraSensor camera_sensor.cpp:411 'Sensor B': Failed to retrieve the camera locati
on
[143:25:43.371991431] [2444415]  INFO IPAProxy ipa_proxy.cpp:130 libcamera is not installed. Loading IPA configuration fr
om '/home/kbingham/iob/libcamera/ci/libcamera-ci/src/libcamera/src/ipa/vimc/data'
 -  'Logitech BRIO' (\_SB_.PCI0.GP13.XHC0.RHUB.PRT8-4.1:1.0-046d:085e)
 -  (platform/vimc.0 Sensor B)
Default viewfinder configuration is: 1920x1080-NV12
Validated viewfinder configuration is: 1920x1080-NV12
[143:25:43.374254333] [2444412]  INFO Camera camera.cpp:1026 configuring streams: (0) 1920x1080-NV12
Allocated 4 buffers for stream

Request completed: Request(0:C:0/1:0)
        SensorTimestamp = 516344180492000
 seq: 000000 timestamp: 516344180492000 bytesused: 2073600/1036800

Request completed: Request(1:C:0/1:0)
        SensorTimestamp = 516344404491000
 seq: 000001 timestamp: 516344404491000 bytesused: 2073600/1036800

Request completed: Request(2:C:0/1:0)
        SensorTimestamp = 516344432527000
```

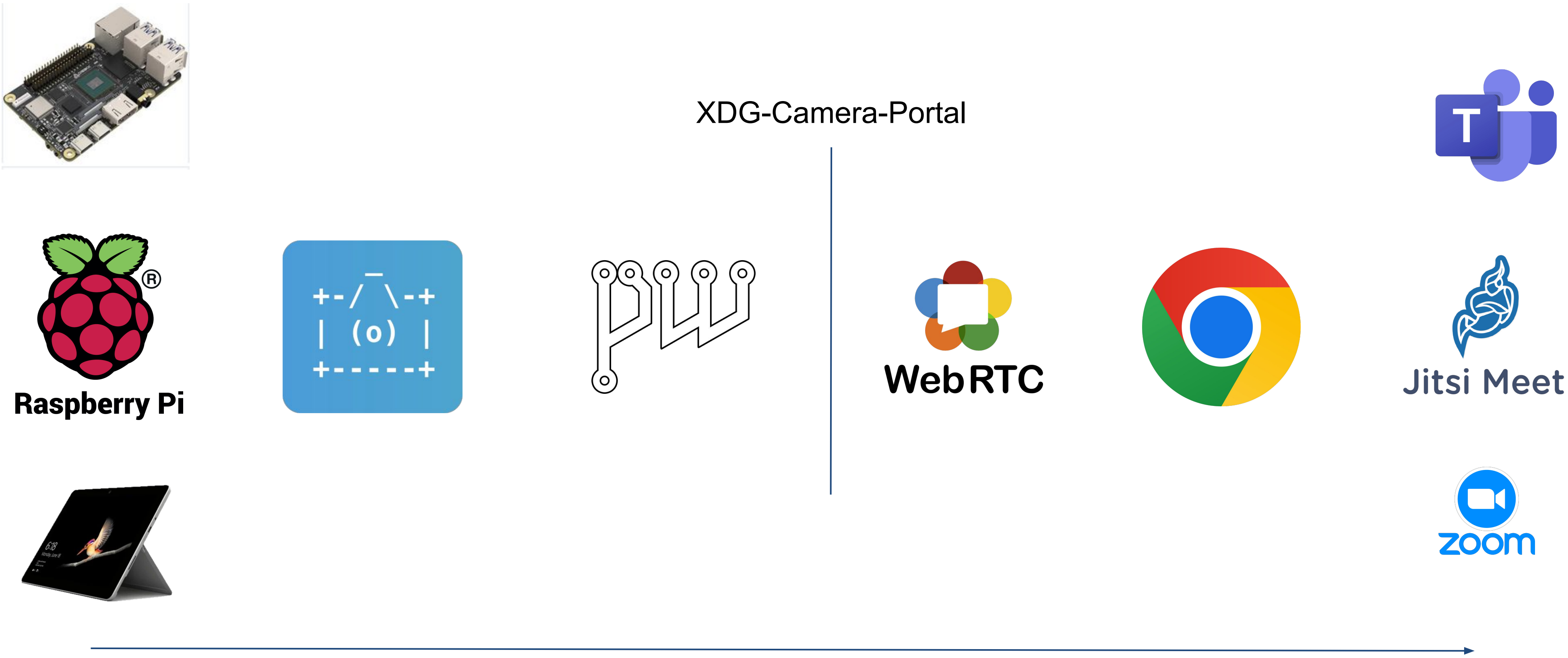
# And a sample ‘hello world’ for the API



libcamera @libcamera · May 26

A surface go 2, running chromium browser through the gnome camera portal, through @PipewireP , and through @libcamera !!!

# Pipewire integration brings desktop use cases



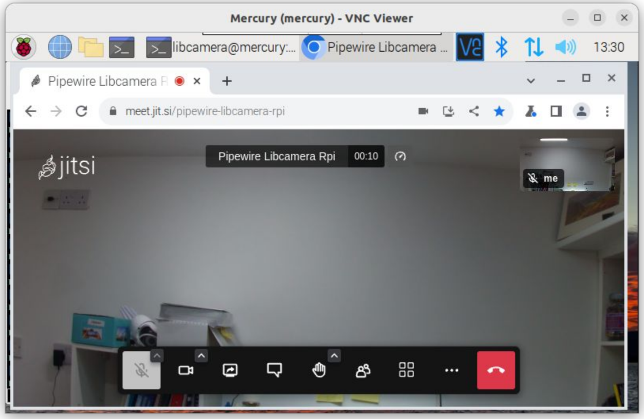
https://flatpak.github.io/xdg-desktop-portal/#gdbus-org.freedesktop.portal.Camera

# Such as video conferencing through chromium

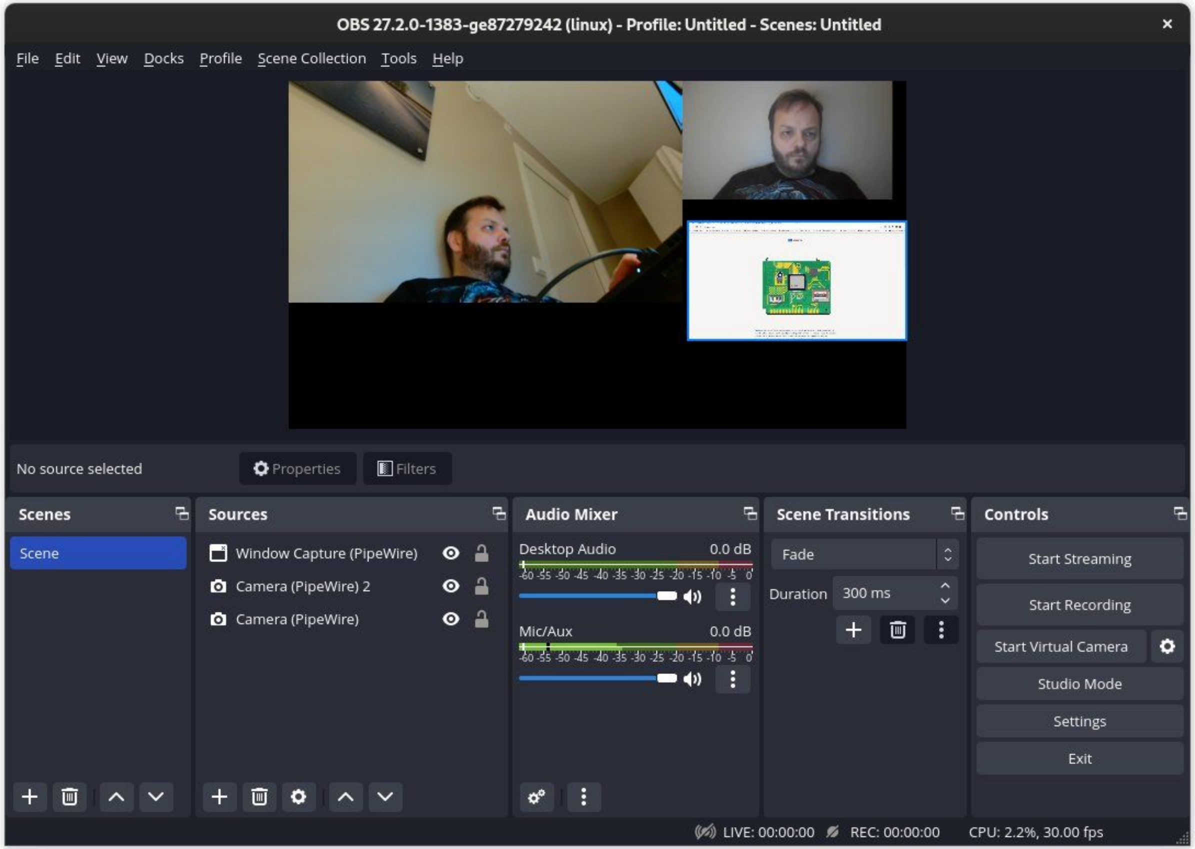
- Lacks correct format negotiation
- Stride not correctly managed
  - Affects frame sizes that are not a multiple of 32, in width. 1280x720 is usable
- ~~NV12 support has format configuration faults in WebRTC~~
- PipeWire issue with multiple cameras.
  - Fixed to single camera for the moment
- Segmentation faults with RPi chromium build with the V4L2 M2M decoders.
  - A pain - but not related to the camera work

Technical Showcase stand :

- Embedded Linux conference Europe (Dublin 2022)



# This is still in development and has a way to go



OBS 27.2.0-1383-ge87279242 (linux) - Profile: Untitled - Scenes: Untitled

No source selected

Scenes: Scene

Sources: Window Capture (PipeWire), Camera (PipeWire) 2, Camera (PipeWire)

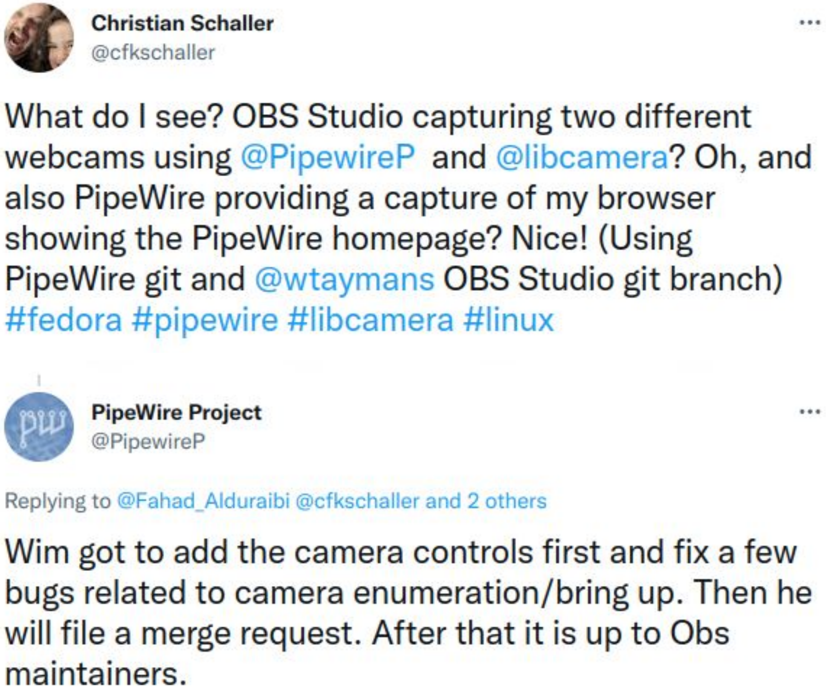
Desktop Audio 0.0 dB

Mic/Aux 0.0 dB

Fade

Duration 300 ms

LIVE: 00:00:00 REC: 00:00:00 CPU: 2.2%, 30.00 fps



**Christian Schaller**
@cfkschaller

What do I see? OBS Studio capturing two different webcams using @PipewireP and @libcamera? Oh, and also PipeWire providing a capture of my browser showing the PipeWire homepage? Nice! (Using PipeWire git and @wtaymans OBS Studio git branch) #fedora #pipewire #libcamera #linux

**PipeWire Project**
@PipewireP

Replying to @Fahad_Alduraibi @cfkschaller and 2 others

Wim got to add the camera controls first and fix a few bugs related to camera enumeration/bring up. Then he will file a merge request. After that it is up to Obs maintainers.

# But it’s moving fast

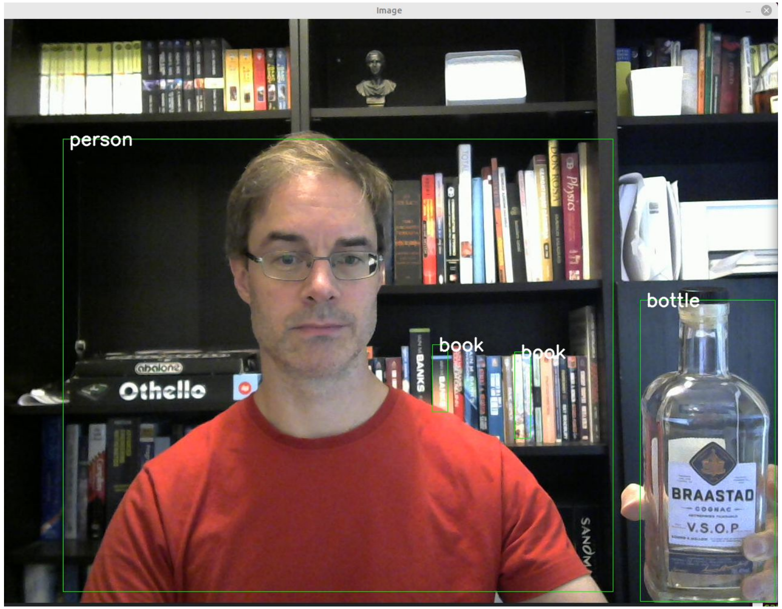
```
   _
+-/ \-+
| (o) |
+-----+
```

- Complex Cameras
- Complications
- Existing Solutions
- **Community Support**
- Future Developments
- Q+A

- I want to hear use cases and requirements
  - Platforms (RPi, NXP, Rockchip, … Other)
  - Cameras (Sony IMX, Omnivision, IP/Genicam?)
  - Frameworks (OpenCV / Tensorflow … others?)

# ROS integration

# OpenCV



**[libcamera-devel] py: examples: Add simple-capture-opencv.py**

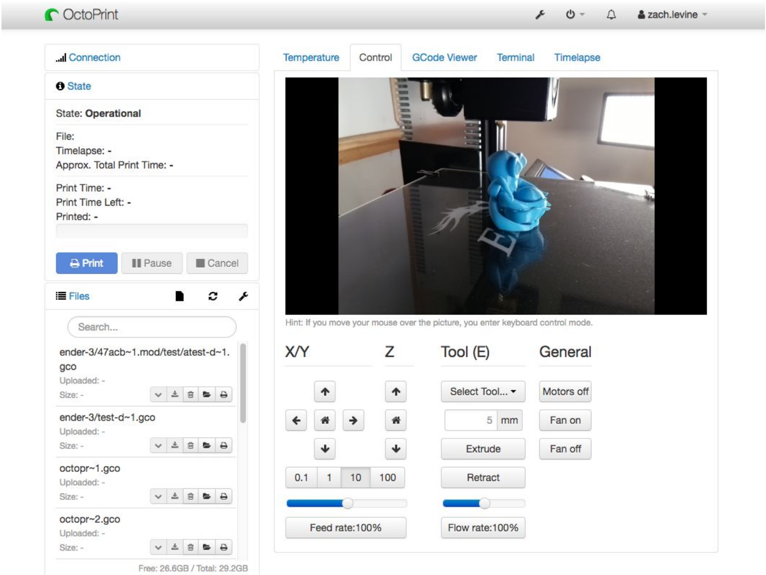
- libcamerify
  - Frame rate support to be added to the V4L2 adaptation layer.
  - **[libcamera-devel,v3] v4l2: Support setting frame rate in the V4L2 Adaptation layer**
  - [https://patchwork.libcamera.org/patch/15392/](https://patchwork.libcamera.org/patch/15392/)
- Likely a good candidate for a direct libcamera implementation
  - GSoC open
- GStreamer pipeline support already possible through the gstlibcamerasrc
- Direct python example now available

# Motion

- https://github.com/Motion-Project/motion/
- Widely used with Raspberry Pi devices
- Working with ‘libcamerify’
- Proposed native libcamera integration - but C++ rejected

- https://github.com/Motion-Project/motionplus
  - “MotionPlus is a break at version 4.2.2 from the Motion application. MotionPlus removes some of the outdated processes and features of the Motion application and introduces new functionalities.
  - Could be suitable for someone to write native libcamera integration

# mjpeg-streamer



State: Operational

File:
Timelapse: -
Approx. Total Print Time: -

Print Time: -
Print Time Left: -
Printed: -

ender-3/47acb~1.mod/test/atest-d~1.gco
Uploaded: -
Size: -

ender-3/test-d~1.gco
Uploaded: -
Size: -

octopr~1.gco
Uploaded: -
Size: -

octopr~2.gco
Uploaded: -
Size: -

Free: 26.6GB / Total: 29.2GB

Hint: If you move your mouse over the picture, you enter keyboard control mode.

- https://sourceforge.net/projects/mjpg-streamer/
- https://github.com/jacksonliam/mjpg-streamer
- MJPG-streamer takes JPGs from Linux-UVC compatible webcams, filesystem or other input plugins and streams them as M-JPEG via HTTP to webbrowsers, VLC and other software. It is the successor of uvc-streamer, a Linux-UVC streaming application with Pan/Tilt
- https://github.com/ArduCAM/mjpg-streamer

Commits on Feb 21, 2022

Add input_libcamera plugin

tom-ard committed on 21 Feb

e503302

# And more users, or potential users …

- https://github.com/ericcurtin/twincam
  - A lightweight camera application, designed to start quickly in a bare environment. It is named twincam as it is built with automotive in mind like a twin-cam engine, it is simply the name of the application.
- https://github.com/folkertvanheusden/constatus
  - Constatus monitors, converts, transforms, filters and multiplexes video-feeds. Feeds like IP-cameras, "video4linux"-devices, pixelflut, VNC-servers, Raspberry Pi-cameras, etc. It is an NVR (network video recorder) with special features.
- https://github.com/aler9/rtsp-simple-server
- https://github.com/ayufan-research/camera-streamer

Potential (upcoming?) Users

- Megapixels / Millipixels
  - Pinephone … Custom 'ini' files to handle media controller.
- Your Camera based app here …
  - Please come and talk to me if you have a use case/app already.
  - I want to hear about more users and use cases!

- #libcamera on irc.oftc.net / Matrix Bridge / Mailing list.

```
    _
+-/ \-+
| (o) |
+-----+
```

- Complex Cameras
- Complications
- Existing Solutions
- Community Support
- **Future Developments**
- Q+A

# The pain of loss is greater than the power of gain.

**To overcome the resistance of change:**

- **Dissatisfaction** with the way things are now
- A positive **vision** of the future
- Concrete **steps** to make the vision a reality

https://www.emersonhc.com/change-management/people-hard-wired-resist-change

https://www.emersonhc.com/change-management/people-hard-wired-resist-change

IDEAS ON BOARD

## Applications (developers) need to see the benefits

# Chromium and libwebrtc needs to be upstream

- libwebrtc changes need more review and testing
- Many common use cases will be opened up
- Pengutronix are already leading the way on this

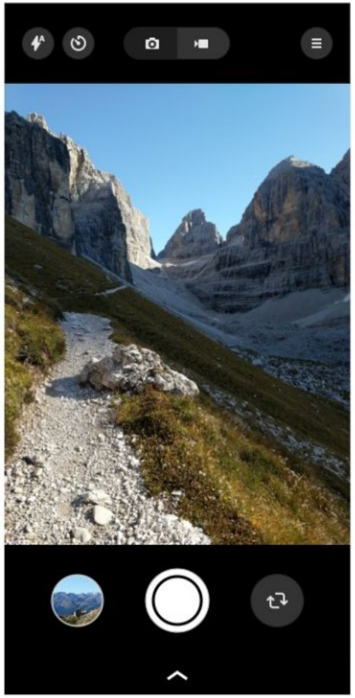
**WebRTC**
Issue 13177: Use Pipewire for camera/webcam access as well : https://bugs.chromium.org/p/webrtc/issues/detail?id=13177
Gerrit: Split out generic portal / pipewire code : https://webrtc-review.googlesource.com/c/src/+/263721
Gerrit: Add pipewire/portal video capture support : https://webrtc-review.googlesource.com/c/src/+/261620
Gerrit: Add callback for raw frames for video capture : https://webrtc-review.googlesource.com/c/src/+/264548

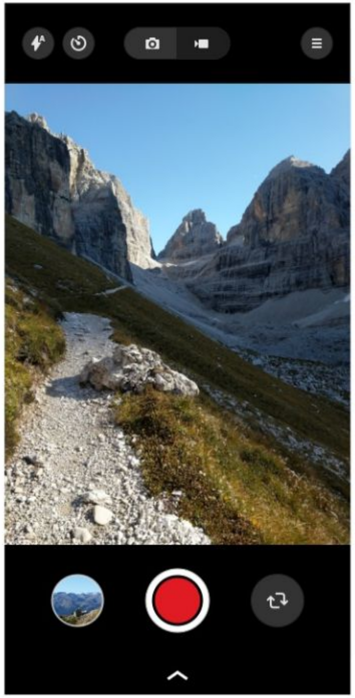
**Chromium**
Video Capture Linux: add backend for portal / pipewire cameras : https://chromium-review.googlesource.com/c/chromium/src/+/3308882
Video Capture Linux: factor out v4l2 camera support : https://chromium-review.googlesource.com/c/chromium/src/+/3634526

https://www.omgubuntu.co.uk/2020/02/gnome-camera-app-mockup

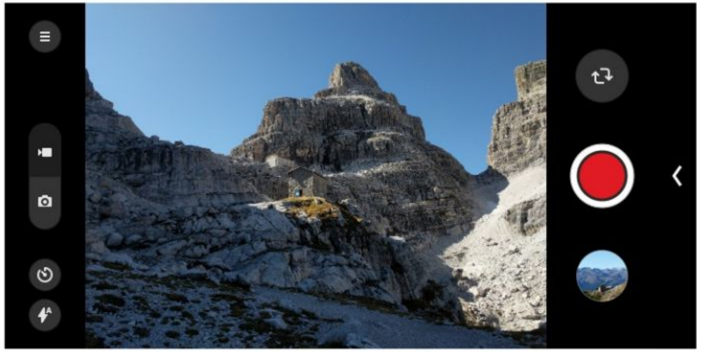
## Phone

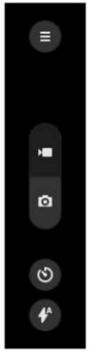
Because the entire 4:3 photo needs to fit onto the view and the screen is quite tall, we need to do something with the additional height. This is done by making the headerbar taller than usual and vertically centering the buttons.





On phones in landscape mode, the overall layout remains the same, but the icons/content rotates





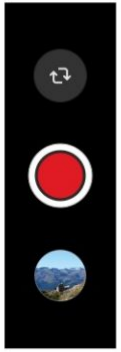
Primary Menu

Switches between photo and video mode

TODO: Should there be a horizontal swipe gesture to switch?

Enables photo timer

Three-state toggle for flash (**On**, **Off**, **Auto** based on brightness)



Switch between Back/Front-facing camera

Shutter

Opens most recently taken photo in Photos
Thumbnail shows the most recently taken picture

# Gnome Camera App

- Which version of libcamera should we use?
  - libcamera$ ./utils/gen-version.sh
  - 0.0.1+50-aa7b3740

- We’re still actively developing. We can’t guarantee ABI/API stability. ?
  - Soname fixes allows packaging the differences

- Distributions are /screaming/ for a ‘tag’
  - *DONE* 0.0.1
  - Aiming for monthly or two monthly release cadence.
  - Major.Minor.Patch
    - Patch increments at 0.0.x until we have automatic ABI breakage detect
    - Minor patch increments will then imply ABI/API breakage, with stable patch
    - Major release bump when we declare stable ABI/API.

- We need more (available) CI
  - Freedesktop ?

# libcamera releases

- Canonical
- Collabora
- Redhat
- Pengutronix
- Raspberry Pi
- NXP
- ChromeOS

## Thank you

- Camera’s Complexities have led to libcamera development becoming a necessity on consumer devices (laptops, linux-phones, embedded)

- Application support with libcamera is increasing but still a long way to go.

- Pipewire support can already provide desktop integration
  - device security through the XDG Camera Portal

- Python bindings are now available.
  - picamera2 now released by RPi

- GStreamer support has had a lot of improvements

# TL;DR

# ? !

# Contact

https://www.libcamera.org

kieran.bingham@ideasonboard.com

# Thank you!

```
   _
+-/ \-+
| (o) |
+-----+
```

libcamera