Supporting complex cameras

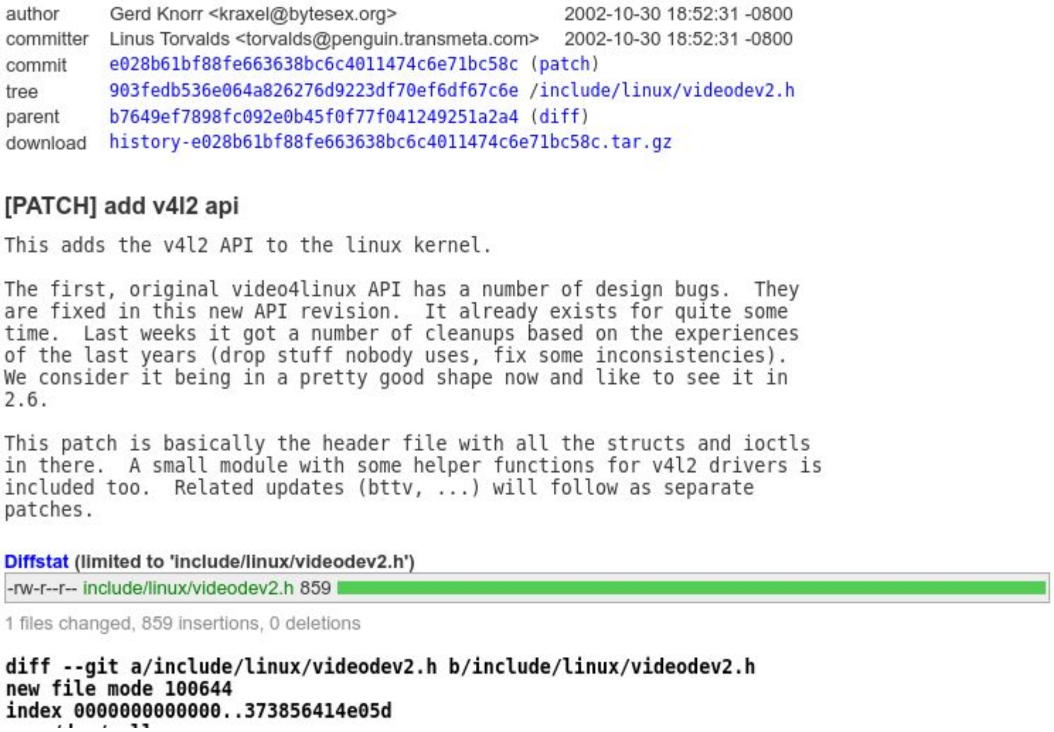
Enabling users and respecting our principles

Ricardo Ribalda

Debian Minidebconf Cambridge 2023, 26th November 2023

Standalone Cameras

Video4Linux 2



author	Gerd Knorr <kraxel@bytesex.org>	2002-10-30 18:52:31 -0800
committer	Linus Torvalds <torvalds@penguin.transmeta.com>	2002-10-30 18:52:31 -0800
commit	e028b61bf88fe663638bc6c4011474c6e71bc58c (patch)	
tree	903fedb536e064a826276d9223df70ef6df67c6e /include/linux/videodev2.h	
parent	b7649ef7898fc092e0b45f0f77f041249251a2a4 (diff)	
download	history-e028b61bf88fe663638bc6c4011474c6e71bc58c.tar.gz	

[PATCH] add v4l2 api

```
This adds the v4l2 API to the linux kernel.

The first, original video4linux API has a number of design bugs.  They
are fixed in this new API revision.  It already exists for quite some
time.  Last weeks it got a number of cleanups based on the experiences
of the last years (drop stuff nobody uses, fix some inconsistencies).
We consider it being in a pretty good shape now and like to see it in
2.6.

This patch is basically the header file with all the structs and ioctls
in there.  A small module with some helper functions for v4l2 drivers is
included too.  Related updates (bttv, ...) will follow as separate
patches.
```

Diffstat (limited to 'include/linux/videodev2.h')

```
-rw-r--r-- include/linux/videodev2.h 859
```

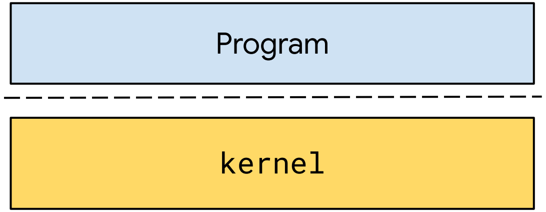
1 files changed, 859 insertions, 0 deletions

```
diff --git a/include/linux/videodev2.h b/include/linux/videodev2.h
new file mode 100644
index 0000000000000..373856414e05d
```

```
cat /dev/video0 > my_holidays_in_hawaii.png
```

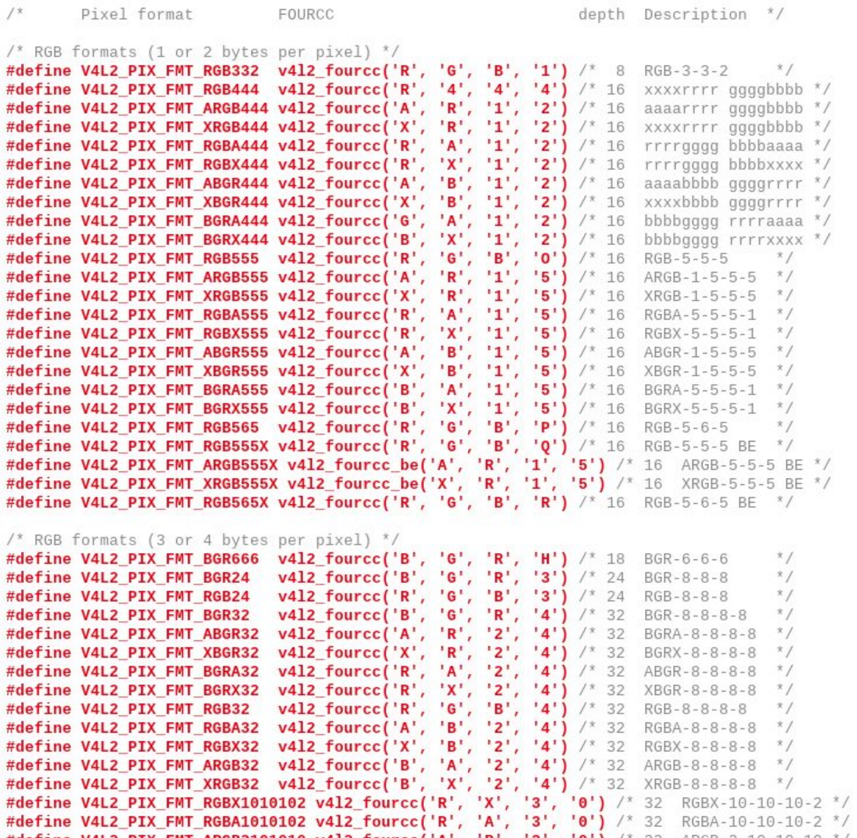
Video4Linux 2

```
fd = open("/dev/video0");
ioctl(fd, VIDIOC_S_FMT, &fmt);
ioctl(fd, VIDIOC_S_PARM, &fps);
ioctl(fd, VIDIOC_S_EXT_CTRL, &exp_time);
ioctl(fd, VIDIOC_REQBUFS, &req)) ;
for i in buf: ioctl(fd, VIDIOC_QUERYBUF, &buf[i]));
for i in buf: ioctl(fd, VIDIOC_QBUF, &buf[i]));
ioctl(fd, VIDIOC_STREAMON, &req)) ;
while true:
    ioctl(fd, VIDIOC_DQBUF, &buf[i]));
    USE IMAGE
    ioctl(fd, VIDIOC_QBUF, &buf[i]));
```



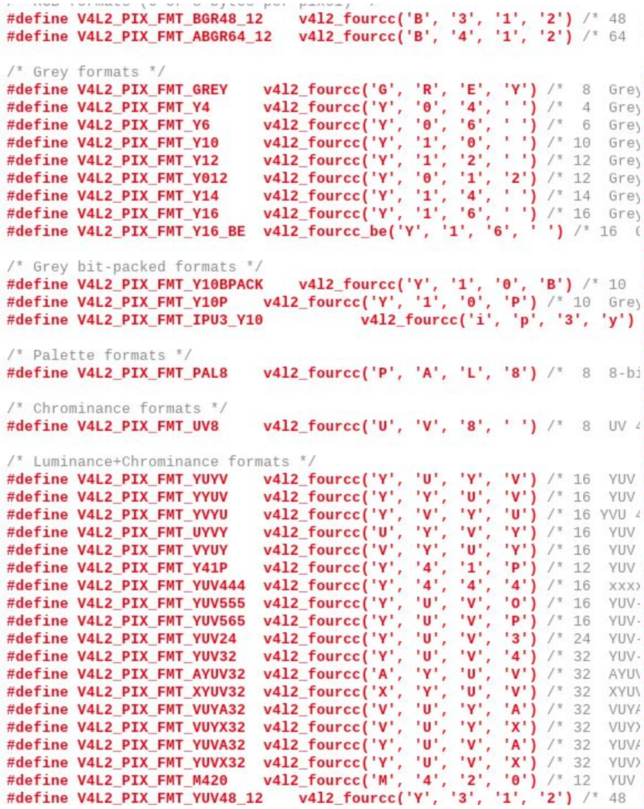
Google

Formats

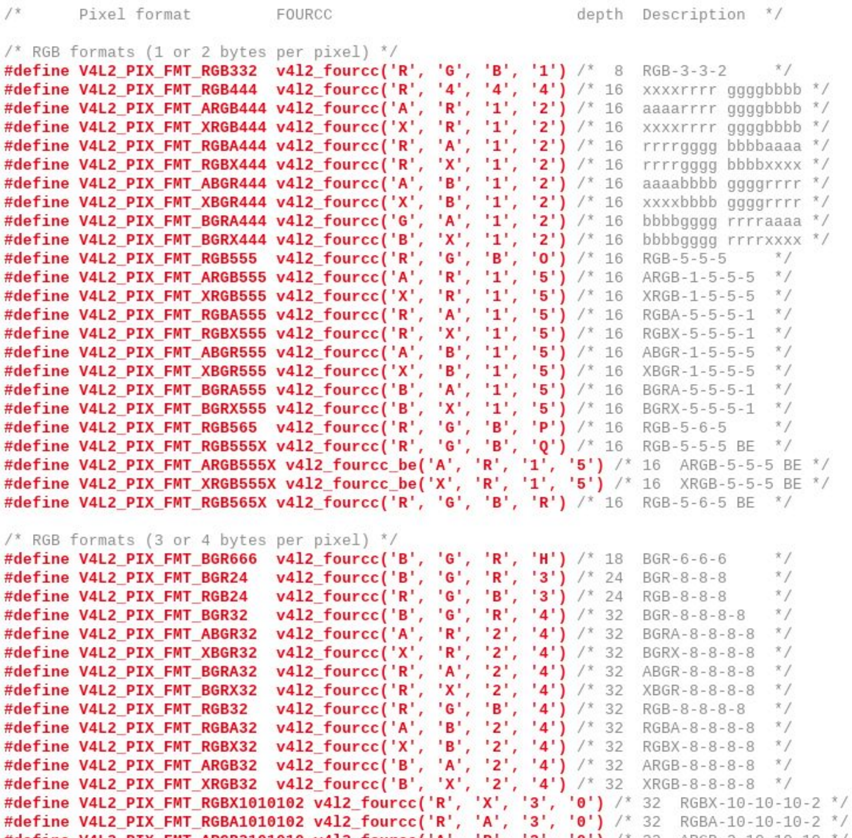


```
/*      Pixel format         FOURCC                          depth  Description  */

/* RGB formats (1 or 2 bytes per pixel) */
#define V4L2_PIX_FMT_RGB332  v4l2_fourcc('R', 'G', 'B', '1') /*  8  RGB-3-3-2     */
#define V4L2_PIX_FMT_RGB444  v4l2_fourcc('R', '4', '4', '4') /* 16  xxxxrrrr ggggbbbb */
#define V4L2_PIX_FMT_ARGB444 v4l2_fourcc('A', 'R', '1', '2') /* 16  aaaarrrr ggggbbbb */
#define V4L2_PIX_FMT_XRGB444 v4l2_fourcc('X', 'R', '1', '2') /* 16  xxxxrrrr ggggbbbb */
#define V4L2_PIX_FMT_RGBA444 v4l2_fourcc('R', 'A', '1', '2') /* 16  rrrrgggg bbbbaaaa */
#define V4L2_PIX_FMT_RGBX444 v4l2_fourcc('R', 'X', '1', '2') /* 16  rrrrgggg bbbbxxxx */
#define V4L2_PIX_FMT_ABGR444 v4l2_fourcc('A', 'B', '1', '2') /* 16  aaaabbbb ggggrrrr */
#define V4L2_PIX_FMT_XBGR444 v4l2_fourcc('X', 'B', '1', '2') /* 16  xxxxbbbb ggggrrrr */
#define V4L2_PIX_FMT_BGRA444 v4l2_fourcc('G', 'A', '1', '2') /* 16  bbbbgggg rrrraaaa */
#define V4L2_PIX_FMT_BGRX444 v4l2_fourcc('B', 'X', '1', '2') /* 16  bbbbgggg rrrrxxxx */
#define V4L2_PIX_FMT_RGB555  v4l2_fourcc('R', 'G', 'B', 'O') /* 16  RGB-5-5-5     */
#define V4L2_PIX_FMT_ARGB555 v4l2_fourcc('A', 'R', '1', '5') /* 16  ARGB-1-5-5-5  */
#define V4L2_PIX_FMT_XRGB555 v4l2_fourcc('X', 'R', '1', '5') /* 16  XRGB-1-5-5-5  */
#define V4L2_PIX_FMT_RGBA555 v4l2_fourcc('R', 'A', '1', '5') /* 16  RGBA-5-5-5-1  */
#define V4L2_PIX_FMT_RGBX555 v4l2_fourcc('R', 'X', '1', '5') /* 16  RGBX-5-5-5-1  */
#define V4L2_PIX_FMT_ABGR555 v4l2_fourcc('A', 'B', '1', '5') /* 16  ABGR-1-5-5-5  */
#define V4L2_PIX_FMT_XBGR555 v4l2_fourcc('X', 'B', '1', '5') /* 16  XBGR-1-5-5-5  */
#define V4L2_PIX_FMT_BGRA555 v4l2_fourcc('B', 'A', '1', '5') /* 16  BGRA-5-5-5-1  */
#define V4L2_PIX_FMT_BGRX555 v4l2_fourcc('B', 'X', '1', '5') /* 16  BGRX-5-5-5-1  */
#define V4L2_PIX_FMT_RGB565  v4l2_fourcc('R', 'G', 'B', 'P') /* 16  RGB-5-6-5     */
#define V4L2_PIX_FMT_RGB555X v4l2_fourcc('R', 'G', 'B', 'Q') /* 16  RGB-5-5-5 BE  */
#define V4L2_PIX_FMT_ARGB555X v4l2_fourcc_be('A', 'R', '1', '5') /* 16  ARGB-5-5-5 BE */
#define V4L2_PIX_FMT_XRGB555X v4l2_fourcc_be('X', 'R', '1', '5') /* 16  XRGB-5-5-5 BE */
#define V4L2_PIX_FMT_RGB565X v4l2_fourcc('R', 'G', 'B', 'R') /* 16  RGB-5-6-5 BE  */

/* RGB formats (3 or 4 bytes per pixel) */
#define V4L2_PIX_FMT_BGR666  v4l2_fourcc('B', 'G', 'R', 'H') /* 18  BGR-6-6-6     */
#define V4L2_PIX_FMT_BGR24   v4l2_fourcc('B', 'G', 'R', '3') /* 24  BGR-8-8-8     */
#define V4L2_PIX_FMT_RGB24   v4l2_fourcc('R', 'G', 'B', '3') /* 24  RGB-8-8-8     */
#define V4L2_PIX_FMT_BGR32   v4l2_fourcc('B', 'G', 'R', '4') /* 32  BGR-8-8-8-8   */
#define V4L2_PIX_FMT_ABGR32  v4l2_fourcc('A', 'R', '2', '4') /* 32  BGRA-8-8-8-8  */
#define V4L2_PIX_FMT_XBGR32  v4l2_fourcc('X', 'R', '2', '4') /* 32  BGRX-8-8-8-8  */
#define V4L2_PIX_FMT_BGRA32  v4l2_fourcc('R', 'A', '2', '4') /* 32  ABGR-8-8-8-8  */
#define V4L2_PIX_FMT_BGRX32  v4l2_fourcc('R', 'X', '2', '4') /* 32  XBGR-8-8-8-8  */
#define V4L2_PIX_FMT_RGB32   v4l2_fourcc('R', 'G', 'B', '4') /* 32  RGB-8-8-8-8   */
#define V4L2_PIX_FMT_RGBA32  v4l2_fourcc('A', 'B', '2', '4') /* 32  RGBA-8-8-8-8  */
#define V4L2_PIX_FMT_RGBX32  v4l2_fourcc('X', 'B', '2', '4') /* 32  RGBX-8-8-8-8  */
#define V4L2_PIX_FMT_ARGB32  v4l2_fourcc('B', 'A', '2', '4') /* 32  ARGB-8-8-8-8  */
#define V4L2_PIX_FMT_XRGB32  v4l2_fourcc('B', 'X', '2', '4') /* 32  XRGB-8-8-8-8  */
#define V4L2_PIX_FMT_RGBX1010102 v4l2_fourcc('R', 'X', '3', '0') /* 32  RGBX-10-10-10-2 */
#define V4L2_PIX_FMT_RGBA1010102 v4l2_fourcc('R', 'A', '3', '0') /* 32  RGBA-10-10-10-2 */
#define V4L2_PIX_FMT_ARGB2101010 v4l2_fourcc('A', 'R', '3', '0') /* 32  ARGB-2-10-10-10 */
```

Formats

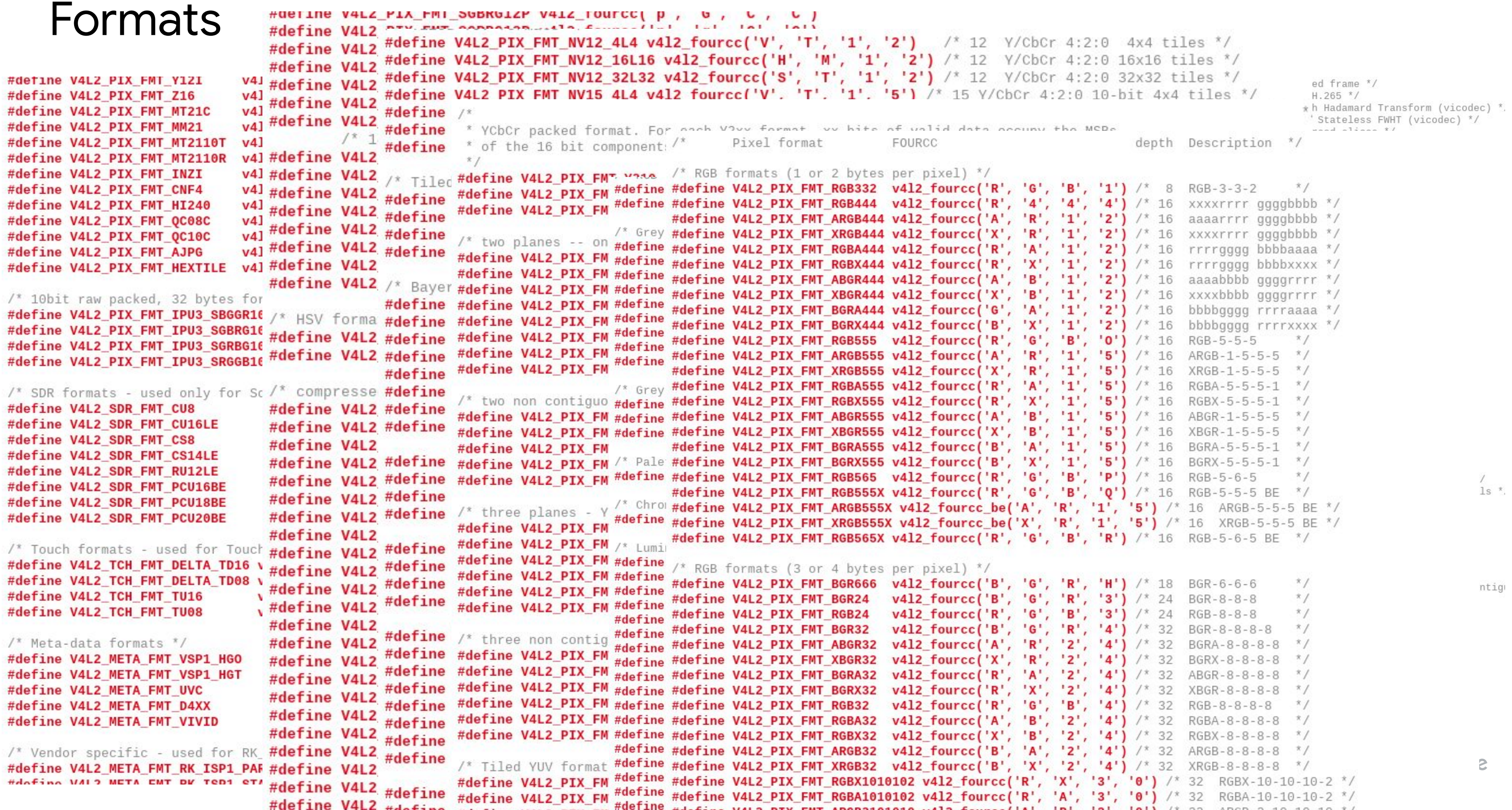


```
/* RGB formats (6 or 8 bytes per pixel) */
#define V4L2_PIX_FMT_BGR48_12    v4l2_fourcc('B', '3', '1', '2') /* 48
#define V4L2_PIX_FMT_ABGR64_12   v4l2_fourcc('B', '4', '1', '2') /* 64

/* Grey formats */
#define V4L2_PIX_FMT_GREY    v4l2_fourcc('G', 'R', 'E', 'Y') /*  8  Grey
#define V4L2_PIX_FMT_Y4      v4l2_fourcc('Y', '0', '4', ' ') /*  4  Grey
#define V4L2_PIX_FMT_Y6      v4l2_fourcc('Y', '0', '6', ' ') /*  6  Grey
#define V4L2_PIX_FMT_Y10     v4l2_fourcc('Y', '1', '0', ' ') /* 10  Grey
#define V4L2_PIX_FMT_Y12     v4l2_fourcc('Y', '1', '2', ' ') /* 12  Grey
#define V4L2_PIX_FMT_Y012    v4l2_fourcc('Y', '0', '1', '2') /* 12  Grey
#define V4L2_PIX_FMT_Y14     v4l2_fourcc('Y', '1', '4', ' ') /* 14  Grey
#define V4L2_PIX_FMT_Y16     v4l2_fourcc('Y', '1', '6', ' ') /* 16  Grey
#define V4L2_PIX_FMT_Y16_BE  v4l2_fourcc_be('Y', '1', '6', ' ') /* 16  (

/* Grey bit-packed formats */
#define V4L2_PIX_FMT_Y10BPACK    v4l2_fourcc('Y', '1', '0', 'B') /* 10
#define V4L2_PIX_FMT_Y10P    v4l2_fourcc('Y', '1', '0', 'P') /* 10  Grey
#define V4L2_PIX_FMT_IPU3_Y10          v4l2_fourcc('i', 'p', '3', 'y')

/* Palette formats */
#define V4L2_PIX_FMT_PAL8    v4l2_fourcc('P', 'A', 'L', '8') /*  8  8-bi

/* Chrominance formats */
#define V4L2_PIX_FMT_UV8     v4l2_fourcc('U', 'V', '8', ' ') /*  8  UV 4

/* Luminance+Chrominance formats */
#define V4L2_PIX_FMT_YUYV    v4l2_fourcc('Y', 'U', 'Y', 'V') /* 16  YUV
#define V4L2_PIX_FMT_YYUV    v4l2_fourcc('Y', 'Y', 'U', 'V') /* 16  YUV
#define V4L2_PIX_FMT_YVYU    v4l2_fourcc('Y', 'V', 'Y', 'U') /* 16 YVU 4
#define V4L2_PIX_FMT_UYVY    v4l2_fourcc('U', 'Y', 'V', 'Y') /* 16  YUV
#define V4L2_PIX_FMT_VYUY    v4l2_fourcc('V', 'Y', 'U', 'Y') /* 16  YUV
#define V4L2_PIX_FMT_Y41P    v4l2_fourcc('Y', '4', '1', 'P') /* 12  YUV
#define V4L2_PIX_FMT_YUV444  v4l2_fourcc('Y', '4', '4', '4') /* 16  xxxx
#define V4L2_PIX_FMT_YUV555  v4l2_fourcc('Y', 'U', 'V', 'O') /* 16  YUV-
#define V4L2_PIX_FMT_YUV565  v4l2_fourcc('Y', 'U', 'V', 'P') /* 16  YUV-
#define V4L2_PIX_FMT_YUV24   v4l2_fourcc('Y', 'U', 'V', '3') /* 24  YUV-
#define V4L2_PIX_FMT_YUV32   v4l2_fourcc('Y', 'U', 'V', '4') /* 32  YUV-
#define V4L2_PIX_FMT_AYUV32  v4l2_fourcc('A', 'Y', 'U', 'V') /* 32  AYUV
#define V4L2_PIX_FMT_XYUV32  v4l2_fourcc('X', 'Y', 'U', 'V') /* 32  XYUV
#define V4L2_PIX_FMT_VUYA32  v4l2_fourcc('V', 'U', 'Y', 'A') /* 32  VUYA
#define V4L2_PIX_FMT_VUYX32  v4l2_fourcc('V', 'U', 'Y', 'X') /* 32  VUYX
#define V4L2_PIX_FMT_YUVA32  v4l2_fourcc('Y', 'U', 'V', 'A') /* 32  YUVA
#define V4L2_PIX_FMT_YUVX32  v4l2_fourcc('Y', 'U', 'V', 'X') /* 32  YUVX
#define V4L2_PIX_FMT_M420    v4l2_fourcc('M', '4', '2', '0') /* 12  YUV
#define V4L2_PIX_FMT_YUV48_12    v4l2_fourcc('Y', '3', '1', '2') /* 48
```



```
/*      Pixel format         FOURCC                          depth  Description  */

/* RGB formats (1 or 2 bytes per pixel) */
#define V4L2_PIX_FMT_RGB332  v4l2_fourcc('R', 'G', 'B', '1') /*  8  RGB-3-3-2     */
#define V4L2_PIX_FMT_RGB444  v4l2_fourcc('R', '4', '4', '4') /* 16  xxxxrrrr ggggbbbb */
#define V4L2_PIX_FMT_ARGB444 v4l2_fourcc('A', 'R', '1', '2') /* 16  aaaarrrr ggggbbbb */
#define V4L2_PIX_FMT_XRGB444 v4l2_fourcc('X', 'R', '1', '2') /* 16  xxxxrrrr ggggbbbb */
#define V4L2_PIX_FMT_RGBA444 v4l2_fourcc('R', 'A', '1', '2') /* 16  rrrrgggg bbbbaaaa */
#define V4L2_PIX_FMT_RGBX444 v4l2_fourcc('R', 'X', '1', '2') /* 16  rrrrgggg bbbbxxxx */
#define V4L2_PIX_FMT_ABGR444 v4l2_fourcc('A', 'B', '1', '2') /* 16  aaaabbbb ggggrrrr */
#define V4L2_PIX_FMT_XBGR444 v4l2_fourcc('X', 'B', '1', '2') /* 16  xxxxbbbb ggggrrrr */
#define V4L2_PIX_FMT_BGRA444 v4l2_fourcc('G', 'A', '1', '2') /* 16  bbbbgggg rrrraaaa */
#define V4L2_PIX_FMT_BGRX444 v4l2_fourcc('B', 'X', '1', '2') /* 16  bbbbgggg rrrrxxxx */
#define V4L2_PIX_FMT_RGB555  v4l2_fourcc('R', 'G', 'B', 'O') /* 16  RGB-5-5-5     */
#define V4L2_PIX_FMT_ARGB555 v4l2_fourcc('A', 'R', '1', '5') /* 16  ARGB-1-5-5-5  */
#define V4L2_PIX_FMT_XRGB555 v4l2_fourcc('X', 'R', '1', '5') /* 16  XRGB-1-5-5-5  */
#define V4L2_PIX_FMT_RGBA555 v4l2_fourcc('R', 'A', '1', '5') /* 16  RGBA-5-5-5-1  */
#define V4L2_PIX_FMT_RGBX555 v4l2_fourcc('R', 'X', '1', '5') /* 16  RGBX-5-5-5-1  */
#define V4L2_PIX_FMT_ABGR555 v4l2_fourcc('A', 'B', '1', '5') /* 16  ABGR-1-5-5-5  */
#define V4L2_PIX_FMT_XBGR555 v4l2_fourcc('X', 'B', '1', '5') /* 16  XBGR-1-5-5-5  */
#define V4L2_PIX_FMT_BGRA555 v4l2_fourcc('B', 'A', '1', '5') /* 16  BGRA-5-5-5-1  */
#define V4L2_PIX_FMT_BGRX555 v4l2_fourcc('B', 'X', '1', '5') /* 16  BGRX-5-5-5-1  */
#define V4L2_PIX_FMT_RGB565  v4l2_fourcc('R', 'G', 'B', 'P') /* 16  RGB-5-6-5     */
#define V4L2_PIX_FMT_RGB555X v4l2_fourcc('R', 'G', 'B', 'Q') /* 16  RGB-5-5-5 BE  */
#define V4L2_PIX_FMT_ARGB555X v4l2_fourcc_be('A', 'R', '1', '5') /* 16  ARGB-5-5-5 BE */
#define V4L2_PIX_FMT_XRGB555X v4l2_fourcc_be('X', 'R', '1', '5') /* 16  XRGB-5-5-5 BE */
#define V4L2_PIX_FMT_RGB565X v4l2_fourcc('R', 'G', 'B', 'R') /* 16  RGB-5-6-5 BE  */

/* RGB formats (3 or 4 bytes per pixel) */
#define V4L2_PIX_FMT_BGR666  v4l2_fourcc('B', 'G', 'R', 'H') /* 18  BGR-6-6-6     */
#define V4L2_PIX_FMT_BGR24   v4l2_fourcc('B', 'G', 'R', '3') /* 24  BGR-8-8-8     */
#define V4L2_PIX_FMT_RGB24   v4l2_fourcc('R', 'G', 'B', '3') /* 24  RGB-8-8-8     */
#define V4L2_PIX_FMT_BGR32   v4l2_fourcc('B', 'G', 'R', '4') /* 32  BGR-8-8-8-8   */
#define V4L2_PIX_FMT_ABGR32  v4l2_fourcc('A', 'R', '2', '4') /* 32  BGRA-8-8-8-8  */
#define V4L2_PIX_FMT_XBGR32  v4l2_fourcc('X', 'R', '2', '4') /* 32  BGRX-8-8-8-8  */
#define V4L2_PIX_FMT_BGRA32  v4l2_fourcc('R', 'A', '2', '4') /* 32  ABGR-8-8-8-8  */
#define V4L2_PIX_FMT_BGRX32  v4l2_fourcc('R', 'X', '2', '4') /* 32  XBGR-8-8-8-8  */
#define V4L2_PIX_FMT_RGB32   v4l2_fourcc('R', 'G', 'B', '4') /* 32  RGB-8-8-8-8   */
#define V4L2_PIX_FMT_RGBA32  v4l2_fourcc('A', 'B', '2', '4') /* 32  RGBA-8-8-8-8  */
#define V4L2_PIX_FMT_RGBX32  v4l2_fourcc('X', 'B', '2', '4') /* 32  RGBX-8-8-8-8  */
#define V4L2_PIX_FMT_ARGB32  v4l2_fourcc('B', 'A', '2', '4') /* 32  ARGB-8-8-8-8  */
#define V4L2_PIX_FMT_XRGB32  v4l2_fourcc('B', 'X', '2', '4') /* 32  XRGB-8-8-8-8  */
#define V4L2_PIX_FMT_RGBX1010102 v4l2_fourcc('R', 'X', '3', '0') /* 32  RGBX-10-10-10-2 */
#define V4L2_PIX_FMT_RGBA1010102 v4l2_fourcc('R', 'A', '3', '0') /* 32  RGBA-10-10-10-2 */
#define V4L2_PIX_FMT_ARGB2101010 v4l2_fourcc('A', 'R', '3', '0') /* 32  ARGB-2-10-10-10 */
```

Formats



```
#define V4L2_PIX_FMT_NV12_4L4 v4l2_fourcc('V', 'T', '1', '2')   /* 12  Y/CbCr 4:2:0  4x4 tiles */
#define V4L2_PIX_FMT_NV12_16L16 v4l2_fourcc('H', 'M', '1', '2') /* 12  Y/CbCr 4:2:0 16x16 tiles */
#define V4L2_PIX_FMT_NV12_32L32 v4l2_fourcc('S', 'T', '1', '2') /* 12  Y/CbCr 4:2:0 32x32 tiles */
#define V4L2_PIX_FMT_NV15_4L4 v4l2_fourcc('V', 'T', '1', '5') /* 15 Y/CbCr 4:2:0 10-bit 4x4 tiles */
```

```
/*      Pixel format         FOURCC                          depth  Description  */

/* RGB formats (1 or 2 bytes per pixel) */
#define V4L2_PIX_FMT_RGB332  v4l2_fourcc('R', 'G', 'B', '1') /*  8  RGB-3-3-2     */
#define V4L2_PIX_FMT_RGB444  v4l2_fourcc('R', '4', '4', '4') /* 16  xxxxrrrr ggggbbbb */
#define V4L2_PIX_FMT_ARGB444 v4l2_fourcc('A', 'R', '1', '2') /* 16  aaaarrrr ggggbbbb */
#define V4L2_PIX_FMT_XRGB444 v4l2_fourcc('X', 'R', '1', '2') /* 16  xxxxrrrr ggggbbbb */
#define V4L2_PIX_FMT_RGBA444 v4l2_fourcc('R', 'A', '1', '2') /* 16  rrrrgggg bbbbaaaa */
#define V4L2_PIX_FMT_RGBX444 v4l2_fourcc('R', 'X', '1', '2') /* 16  rrrrgggg bbbbxxxx */
#define V4L2_PIX_FMT_ABGR444 v4l2_fourcc('A', 'B', '1', '2') /* 16  aaaabbbb ggggrrrr */
#define V4L2_PIX_FMT_XBGR444 v4l2_fourcc('X', 'B', '1', '2') /* 16  xxxxbbbb ggggrrrr */
#define V4L2_PIX_FMT_BGRA444 v4l2_fourcc('G', 'A', '1', '2') /* 16  bbbbgggg rrrraaaa */
#define V4L2_PIX_FMT_BGRX444 v4l2_fourcc('B', 'X', '1', '2') /* 16  bbbbgggg rrrrxxxx */
#define V4L2_PIX_FMT_RGB555  v4l2_fourcc('R', 'G', 'B', 'O') /* 16  RGB-5-5-5     */
#define V4L2_PIX_FMT_ARGB555 v4l2_fourcc('A', 'R', '1', '5') /* 16  ARGB-1-5-5-5  */
#define V4L2_PIX_FMT_XRGB555 v4l2_fourcc('X', 'R', '1', '5') /* 16  XRGB-1-5-5-5  */
#define V4L2_PIX_FMT_RGBA555 v4l2_fourcc('R', 'A', '1', '5') /* 16  RGBA-5-5-5-1  */
#define V4L2_PIX_FMT_RGBX555 v4l2_fourcc('R', 'X', '1', '5') /* 16  RGBX-5-5-5-1  */
#define V4L2_PIX_FMT_ABGR555 v4l2_fourcc('A', 'B', '1', '5') /* 16  ABGR-1-5-5-5  */
#define V4L2_PIX_FMT_XBGR555 v4l2_fourcc('X', 'B', '1', '5') /* 16  XBGR-1-5-5-5  */
#define V4L2_PIX_FMT_BGRA555 v4l2_fourcc('B', 'A', '1', '5') /* 16  BGRA-5-5-5-1  */
#define V4L2_PIX_FMT_BGRX555 v4l2_fourcc('B', 'X', '1', '5') /* 16  BGRX-5-5-5-1  */
#define V4L2_PIX_FMT_RGB565  v4l2_fourcc('R', 'G', 'B', 'P') /* 16  RGB-5-6-5     */
#define V4L2_PIX_FMT_RGB555X v4l2_fourcc('R', 'G', 'B', 'Q') /* 16  RGB-5-5-5 BE  */
#define V4L2_PIX_FMT_ARGB555X v4l2_fourcc_be('A', 'R', '1', '5') /* 16  ARGB-5-5-5 BE */
#define V4L2_PIX_FMT_XRGB555X v4l2_fourcc_be('X', 'R', '1', '5') /* 16  XRGB-5-5-5 BE */
#define V4L2_PIX_FMT_RGB565X v4l2_fourcc('R', 'G', 'B', 'R') /* 16  RGB-5-6-5 BE  */

/* RGB formats (3 or 4 bytes per pixel) */
#define V4L2_PIX_FMT_BGR666  v4l2_fourcc('B', 'G', 'R', 'H') /* 18  BGR-6-6-6     */
#define V4L2_PIX_FMT_BGR24   v4l2_fourcc('B', 'G', 'R', '3') /* 24  BGR-8-8-8     */
#define V4L2_PIX_FMT_RGB24   v4l2_fourcc('R', 'G', 'B', '3') /* 24  RGB-8-8-8     */
#define V4L2_PIX_FMT_BGR32   v4l2_fourcc('B', 'G', 'R', '4') /* 32  BGR-8-8-8-8   */
#define V4L2_PIX_FMT_ABGR32  v4l2_fourcc('A', 'R', '2', '4') /* 32  BGRA-8-8-8-8  */
#define V4L2_PIX_FMT_XBGR32  v4l2_fourcc('X', 'R', '2', '4') /* 32  BGRX-8-8-8-8  */
#define V4L2_PIX_FMT_BGRA32  v4l2_fourcc('R', 'A', '2', '4') /* 32  ABGR-8-8-8-8  */
#define V4L2_PIX_FMT_BGRX32  v4l2_fourcc('R', 'X', '2', '4') /* 32  XBGR-8-8-8-8  */
#define V4L2_PIX_FMT_RGB32   v4l2_fourcc('R', 'G', 'B', '4') /* 32  RGB-8-8-8-8   */
#define V4L2_PIX_FMT_RGBA32  v4l2_fourcc('A', 'B', '2', '4') /* 32  RGBA-8-8-8-8  */
#define V4L2_PIX_FMT_RGBX32  v4l2_fourcc('X', 'B', '2', '4') /* 32  RGBX-8-8-8-8  */
#define V4L2_PIX_FMT_ARGB32  v4l2_fourcc('B', 'A', '2', '4') /* 32  ARGB-8-8-8-8  */
#define V4L2_PIX_FMT_XRGB32  v4l2_fourcc('B', 'X', '2', '4') /* 32  XRGB-8-8-8-8  */
#define V4L2_PIX_FMT_RGBX1010102 v4l2_fourcc('R', 'X', '3', '0') /* 32  RGBX-10-10-10-2 */
#define V4L2_PIX_FMT_RGBA1010102 v4l2_fourcc('R', 'A', '3', '0') /* 32  RGBA-10-10-10-2 */
```

```
#define V4L2_PIX_FMT_Y12I      v4l
#define V4L2_PIX_FMT_Z16       v4l
#define V4L2_PIX_FMT_MT21C     v4l
#define V4L2_PIX_FMT_MM21      v4l
#define V4L2_PIX_FMT_MT2110T   v4l
#define V4L2_PIX_FMT_MT2110R   v4l
#define V4L2_PIX_FMT_INZI      v4l
#define V4L2_PIX_FMT_CNF4      v4l
#define V4L2_PIX_FMT_HI240     v4l
#define V4L2_PIX_FMT_QC08C     v4l
#define V4L2_PIX_FMT_QC10C     v4l
#define V4L2_PIX_FMT_AJPG      v4l
#define V4L2_PIX_FMT_HEXTILE   v4l

/* 10bit raw packed, 32 bytes for
#define V4L2_PIX_FMT_IPU3_SBGGR10
#define V4L2_PIX_FMT_IPU3_SGBRG10
#define V4L2_PIX_FMT_IPU3_SGRBG10
#define V4L2_PIX_FMT_IPU3_SRGGB10

/* SDR formats - used only for So
#define V4L2_SDR_FMT_CU8
#define V4L2_SDR_FMT_CU16LE
#define V4L2_SDR_FMT_CS8
#define V4L2_SDR_FMT_CS14LE
#define V4L2_SDR_FMT_RU12LE
#define V4L2_SDR_FMT_PCU16BE
#define V4L2_SDR_FMT_PCU18BE
#define V4L2_SDR_FMT_PCU20BE

/* Touch formats - used for Touch
#define V4L2_TCH_FMT_DELTA_TD16  v
#define V4L2_TCH_FMT_DELTA_TD08  v
#define V4L2_TCH_FMT_TU16        v
#define V4L2_TCH_FMT_TU08        v

/* Meta-data formats */
#define V4L2_META_FMT_VSP1_HGO
#define V4L2_META_FMT_VSP1_HGT
#define V4L2_META_FMT_UVC
#define V4L2_META_FMT_D4XX
#define V4L2_META_FMT_VIVID

/* Vendor specific - used for RK_
#define V4L2_META_FMT_RK_ISP1_PAR
#define V4L2_META_FMT_RK_ISP1_STA
```

```
#define V4L2_PIX_FMT_SGBRG12P v4l2_fourcc('p', 'G', 'C', 'C')
#define V4L2_PIX_FMT_SGRBG12P v4l2_fourcc('p', 'g', 'C', 'C')
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
              /* 1
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2

/* HSV forma
#define V4L2
#define V4L2

/* compresse
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
#define V4L2
```

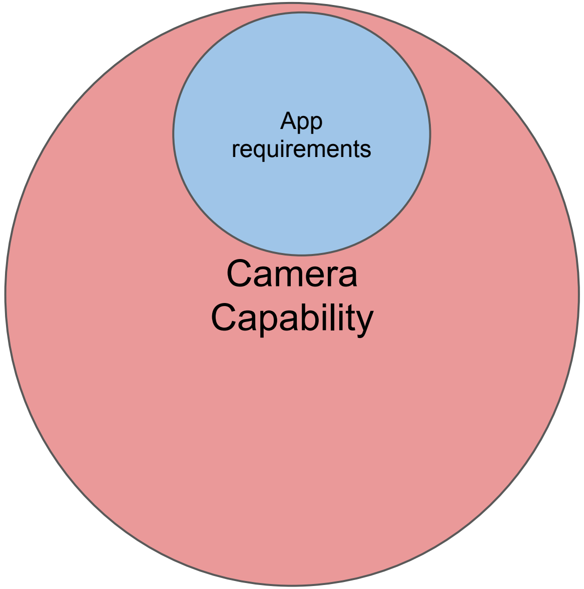
```
#define
#define
#define
/* Tiled
#define
#define
#define
#define

/* Bayer
#define
#define
#define
#define
#define
#define
#define
#define

#define
#define
#define
#define

#define
#define
#define
#define

#define
#define
#define
#define
#define
#define
#define
#define

#define
```

```
/*
 * YCbCr packed format. For each Y2xx format, xx bits of valid data occupy the MSBs
 * of the 16 bit components
 */
#define V4L2_PIX_FMT_Y210
#define V4L2_PIX_FM
#define V4L2_PIX_FM

/* two planes -- on
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM

/* two non contiguo
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM

/* three planes - Y
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM

/* three non contig
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM
#define V4L2_PIX_FM

/* Tiled YUV format
#define V4L2_PIX_FM
#define V4L2_PIX_FM
```

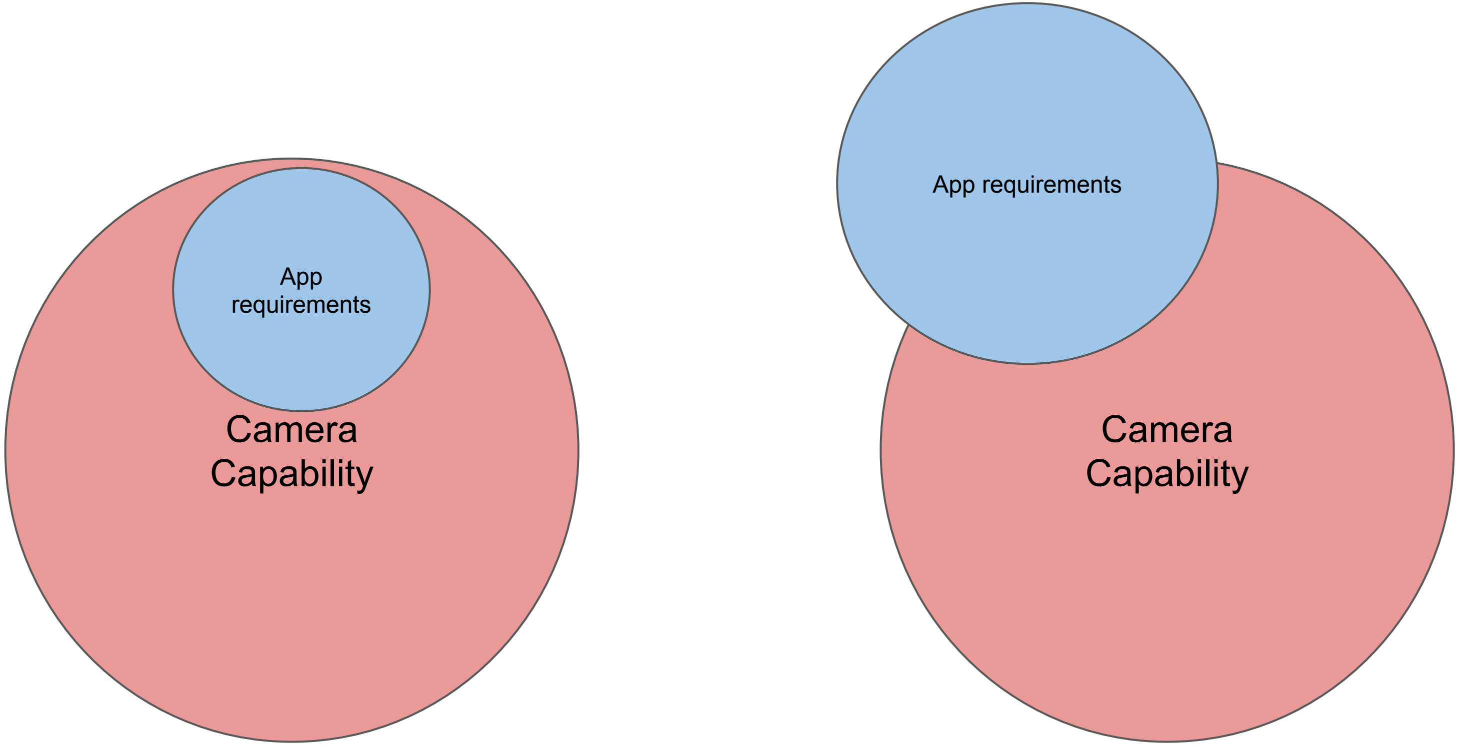
```
#define
#define

/* Grey
#define
#define
#define
#define
#define
#define
#define
#define
#define

/* Grey
#define
#define
#define

/* Pale
#define

/* Chro
#define

/* Lumi
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
#define
```

```
ed frame */
H.265 */
h Hadamard Transform (vicodec) *
' Stateless FWHT (vicodec) */
```

Video4Linux 2



Video4Linux 2



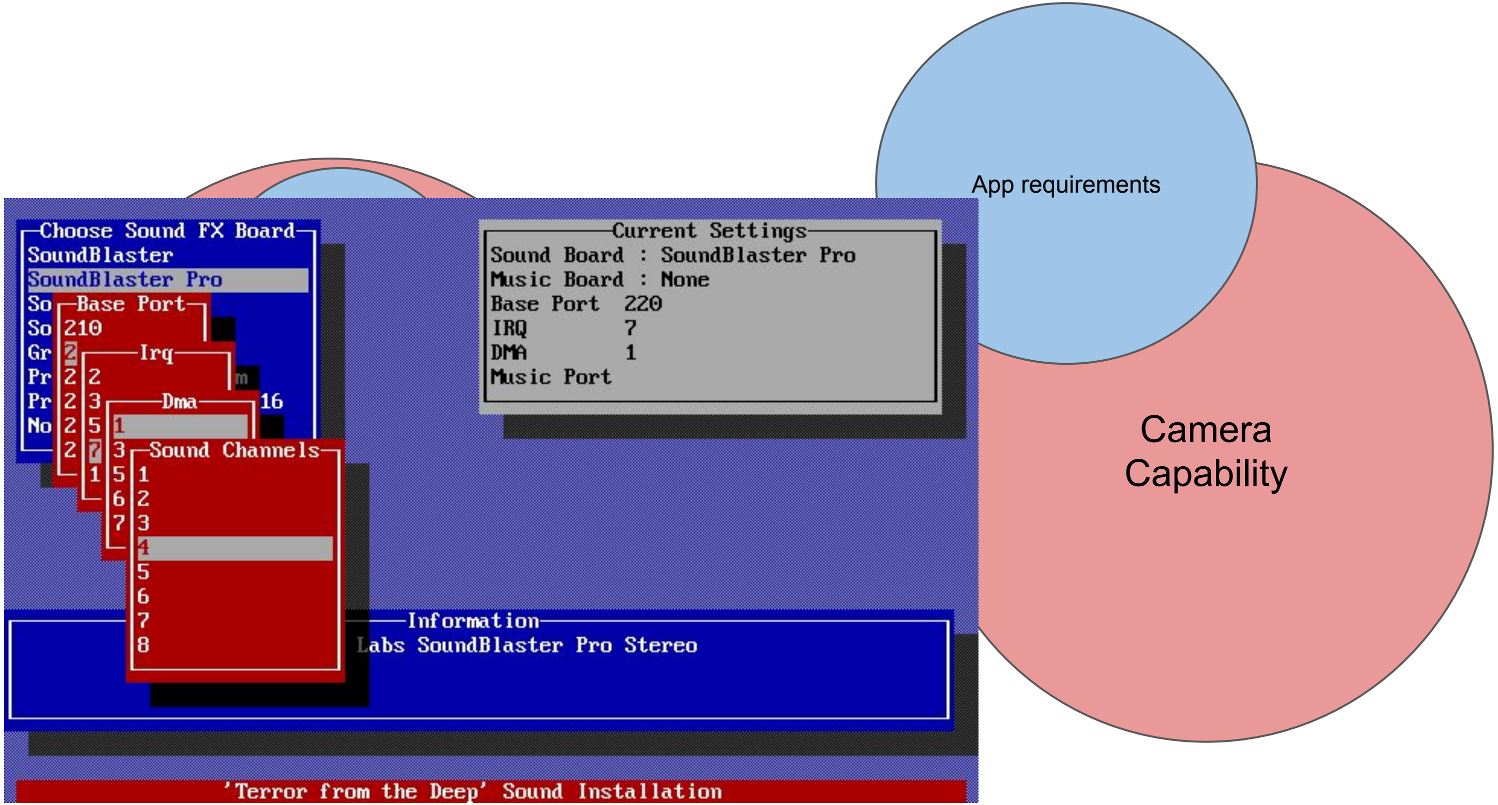
App requirements

Camera Capability

App requirements

Camera Capability

Video4Linux 2



```
Choose Sound FX Board
SoundBlaster
SoundBlaster Pro
So
So
Gr
Pr
Pr
No
```

```
Base Port
210
2
2
2
2
2
2
1
```

```
Irq
2
3
5
7
1
```

```
Dma
1
3
5
6
7
```

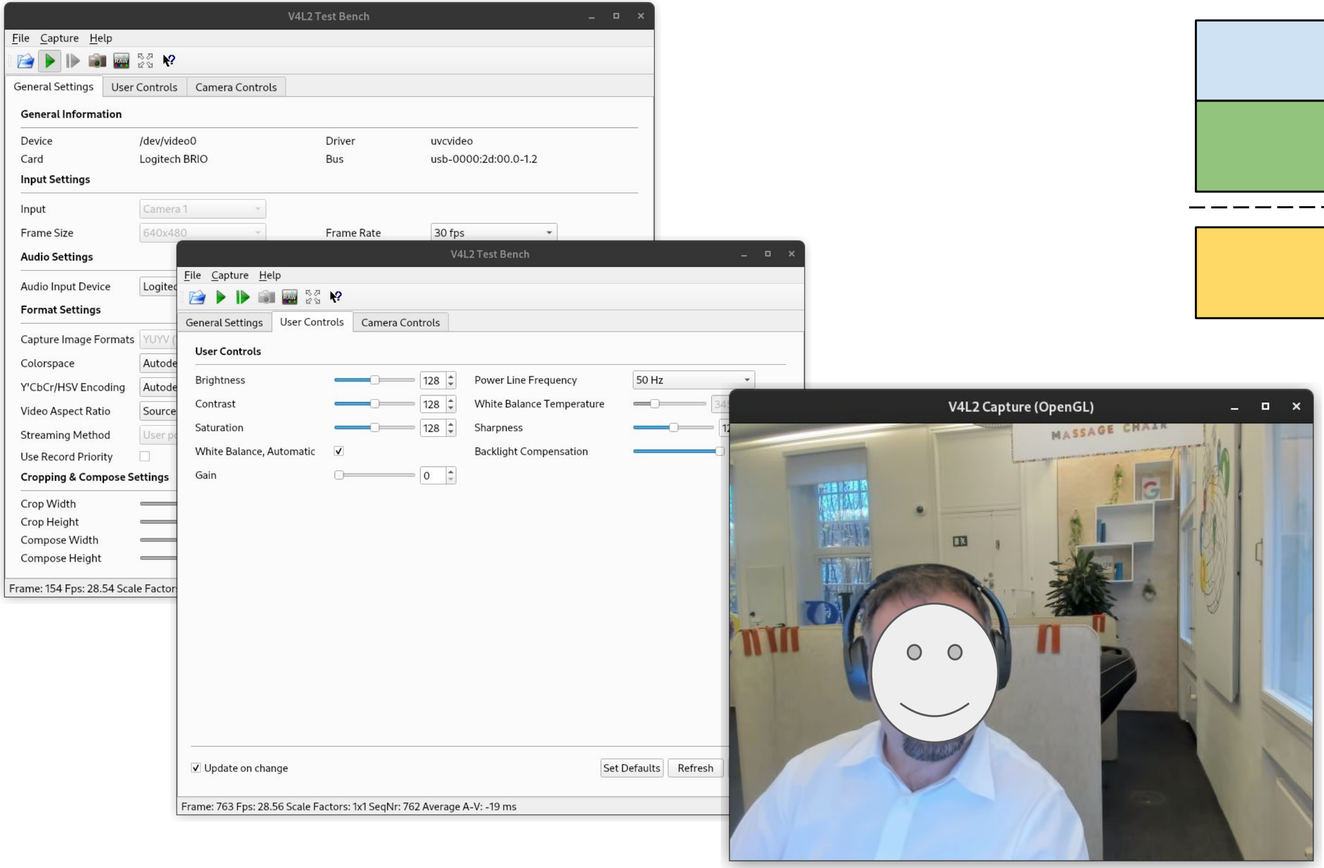
```
Sound Channels
1
2
3
4
5
6
7
8
```

```
Current Settings
Sound Board : SoundBlaster Pro
Music Board : None
Base Port   220
IRQ         7
DMA         1
Music Port
```

```
Information
Labs SoundBlaster Pro Stereo
```

'Terror from the Deep' Sound Installation

qv4l2



General Information

Device /dev/video0 Driver uvcvideo
Card Logitech BRIO Bus usb-0000:2d:00.0-1.2

Input Settings

Input Camera 1
Frame Size 640x480 Frame Rate 30 fps

Audio Settings

Audio Input Device Logitec

Format Settings

Capture Image Formats YUYV (
Colorspace Autode
Y'CbCr/HSV Encoding Autode
Video Aspect Ratio Source
Streaming Method User p
Use Record Priority □

Cropping & Compose Settings

Crop Width
Crop Height
Compose Width
Compose Height

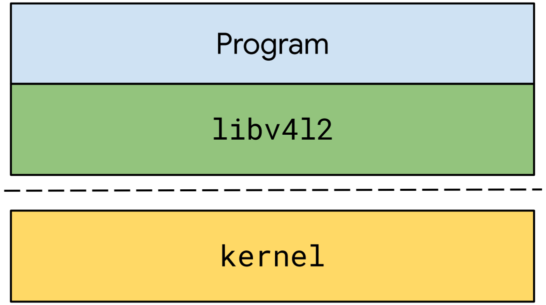
Frame: 154 Fps: 28.54 Scale Factor

User Controls

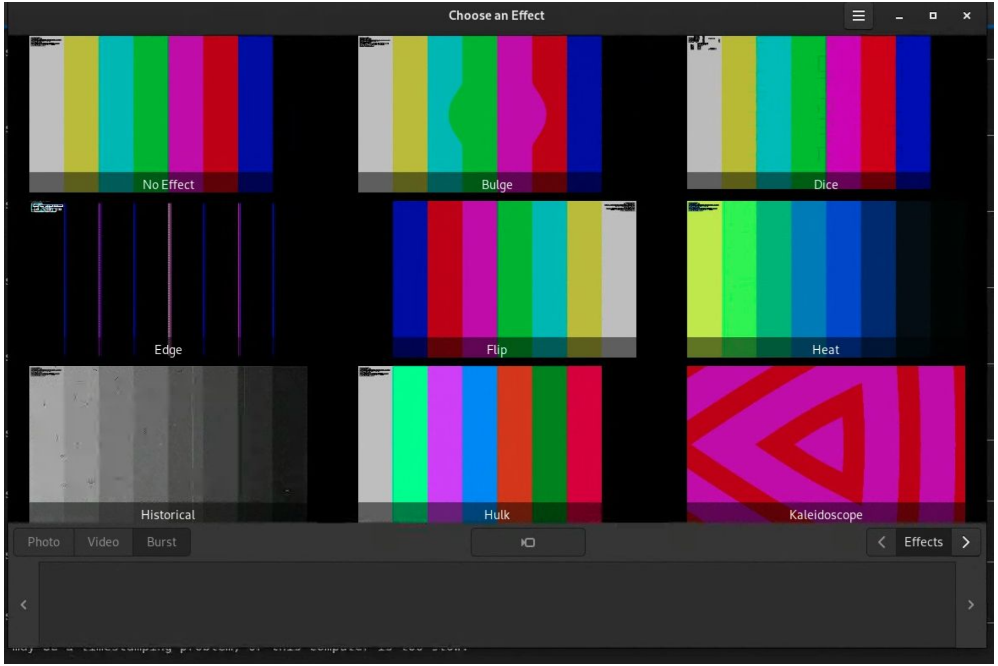
Brightness 128
Contrast 128
Saturation 128
White Balance, Automatic ☑
Gain 0
Power Line Frequency 50 Hz
White Balance Temperature
Sharpness
Backlight Compensation

☑ Update on change

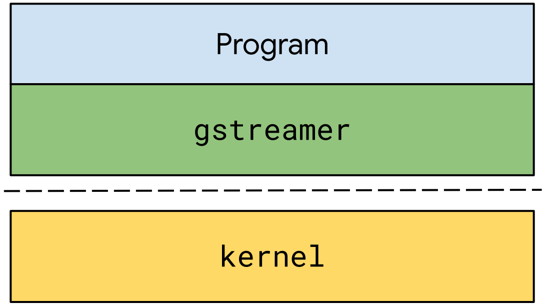
Frame: 763 Fps: 28.56 Scale Factors: 1x1 SeqNr: 762 Average A-V: -19 ms



Cheese

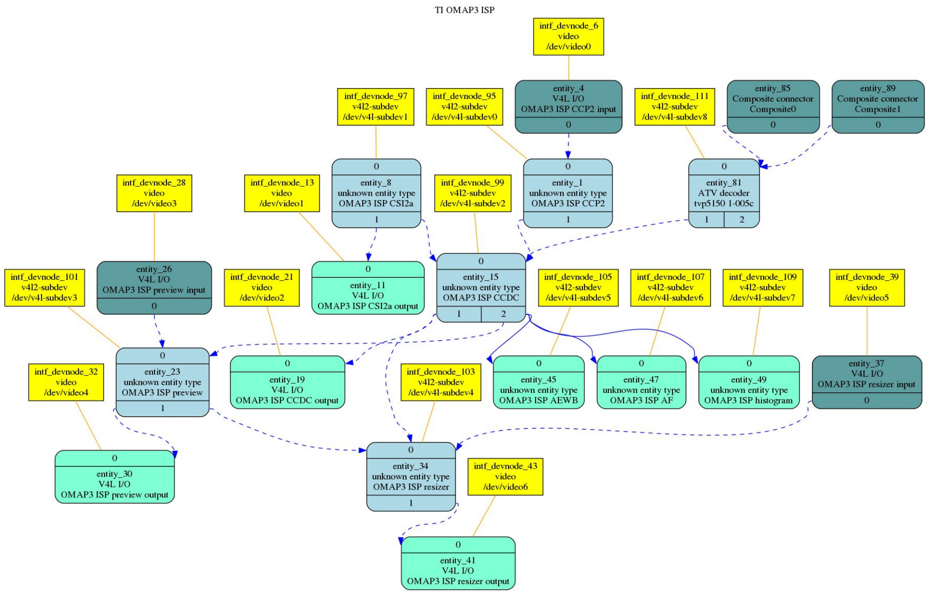


```
may be a timestamping problem, or this computer is too slow.
```

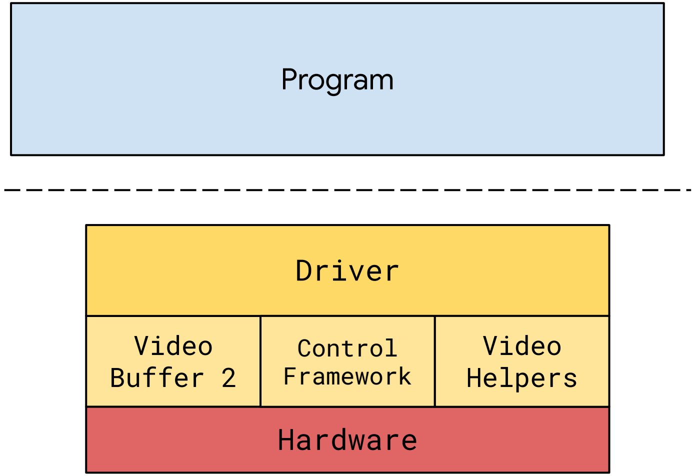


Modular Cameras

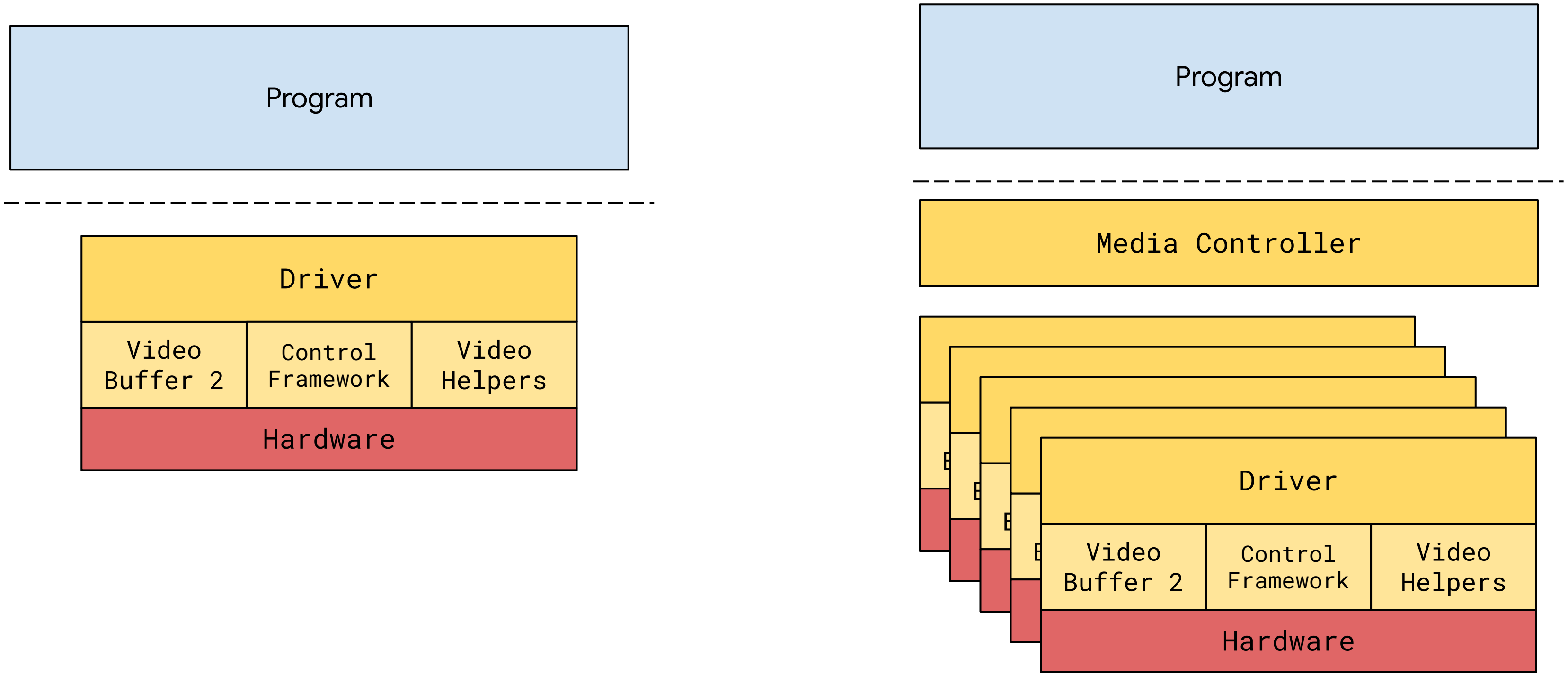
Video4Linux 2 (media controller)


TI OMAP3 ISP
intf_devnode_6 video /dev/video0
entity_4 V4L I/O OMAP3 ISP CCP2 input 0
intf_devnode_97 v4l2-subdev /dev/v4l-subdev1
intf_devnode_95 v4l2-subdev /dev/v4l-subdev0
intf_devnode_111 v4l2-subdev /dev/v4l-subdev8
entity_85 Composite connector Composite0 0
entity_89 Composite connector Composite1 0
0 entity_8 unknown entity type OMAP3 ISP CSI2a 1
0 entity_1 unknown entity type OMAP3 ISP CCP2 1
0 entity_81 ATV decoder tvp5150 1-005c 1 2
intf_devnode_28 video /dev/video3
intf_devnode_13 video /dev/video1
intf_devnode_99 v4l2-subdev /dev/v4l-subdev2
intf_devnode_101 v4l2-subdev /dev/v4l-subdev3
entity_26 V4L I/O OMAP3 ISP preview input 0
intf_devnode_21 video /dev/video2
0 entity_11 V4L I/O OMAP3 ISP CSI2a output
0 entity_15 unknown entity type OMAP3 ISP CCDC 1 2
intf_devnode_105 v4l2-subdev /dev/v4l-subdev5
intf_devnode_107 v4l2-subdev /dev/v4l-subdev6
intf_devnode_109 v4l2-subdev /dev/v4l-subdev7
intf_devnode_39 video /dev/video5
intf_devnode_32 video /dev/video4
0 entity_23 unknown entity type OMAP3 ISP preview 1
0 entity_19 V4L I/O OMAP3 ISP CCDC output
intf_devnode_103 v4l2-subdev /dev/v4l-subdev4
0 entity_45 unknown entity type OMAP3 ISP AEWB
0 entity_47 unknown entity type OMAP3 ISP AF
0 entity_49 unknown entity type OMAP3 ISP histogram
entity_37 V4L I/O OMAP3 ISP resizer input 0
0 entity_30 V4L I/O OMAP3 ISP preview output
0 entity_34 unknown entity type OMAP3 ISP resizer 1
intf_devnode_43 video /dev/video6
0 entity_41 V4L I/O OMAP3 ISP resizer output

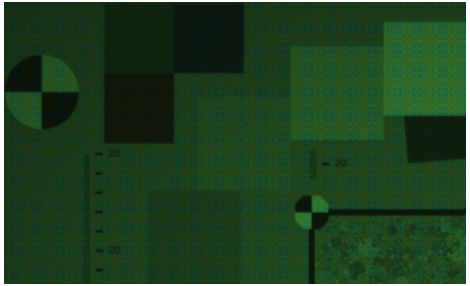
Video4Linux 2 (media controller)

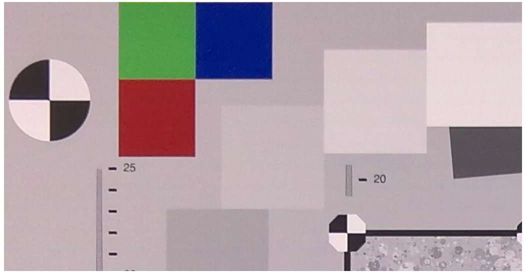


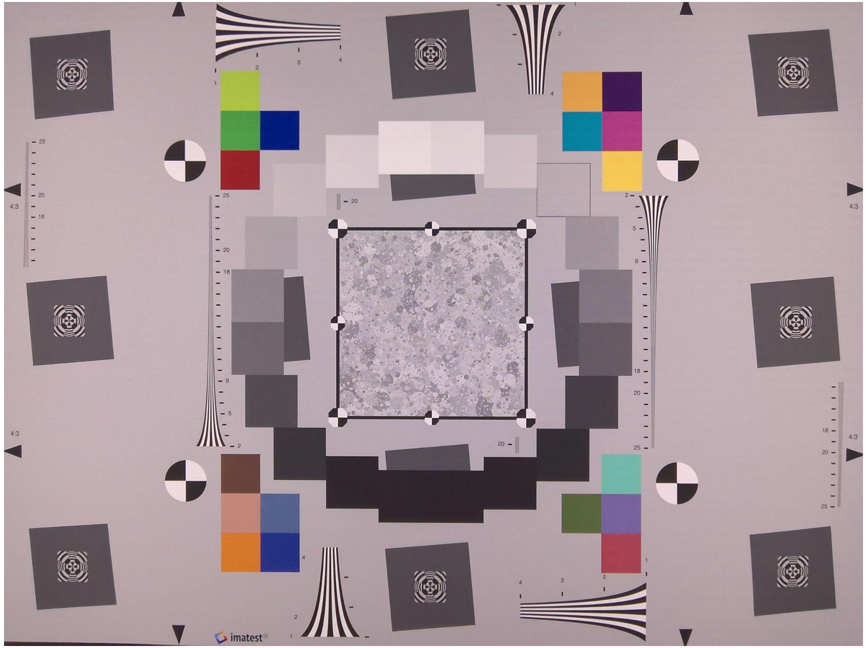
Video4Linux 2 (media controller)



Secret Sauce

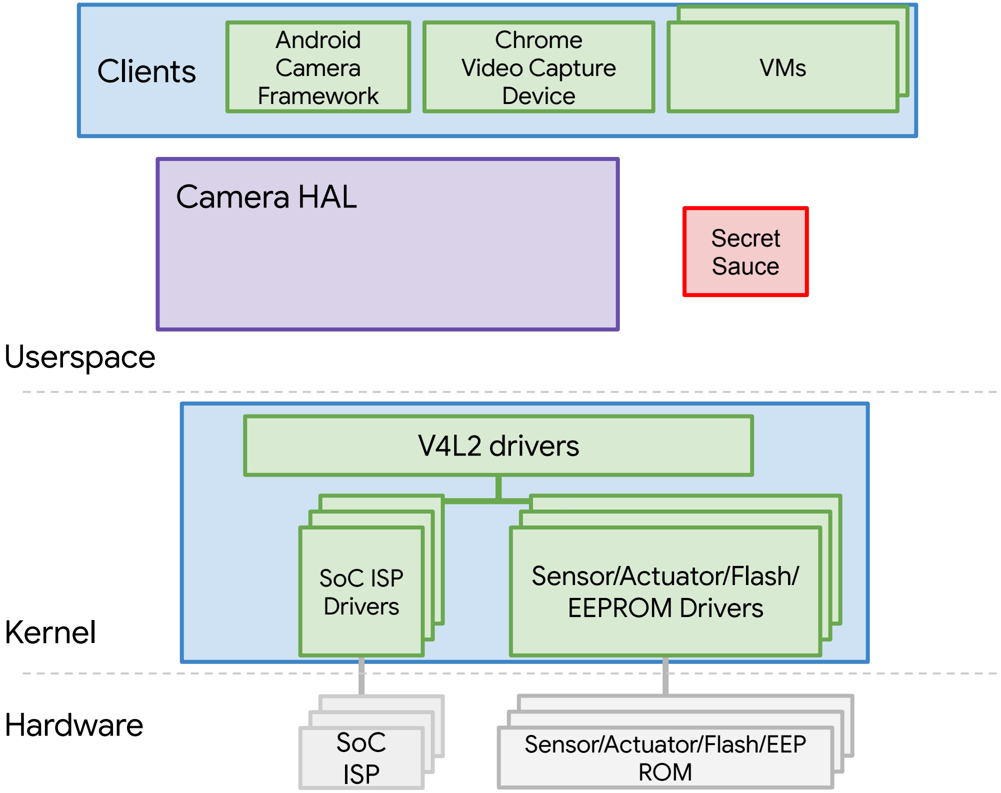




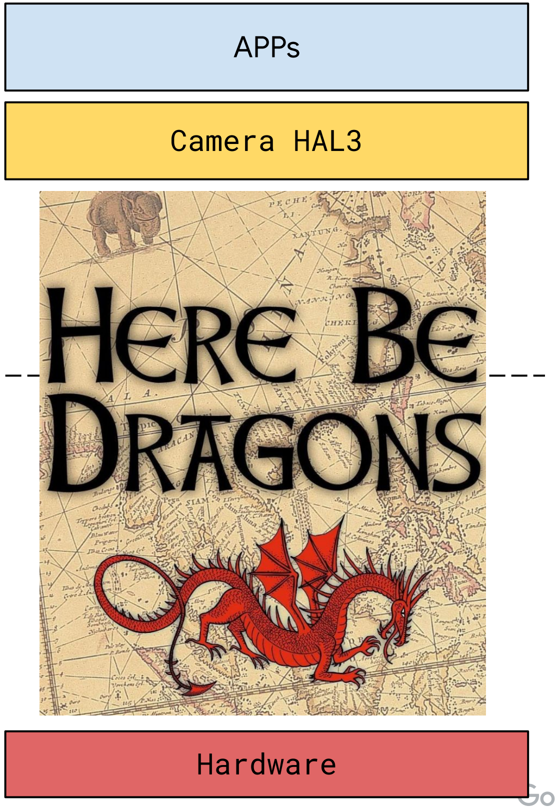


Stack

chromeOS

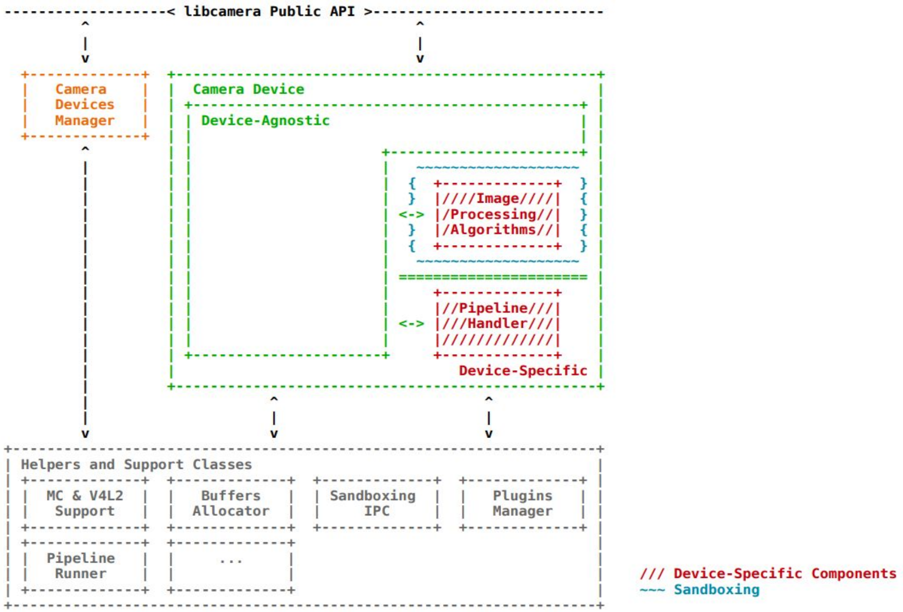


android



Stack

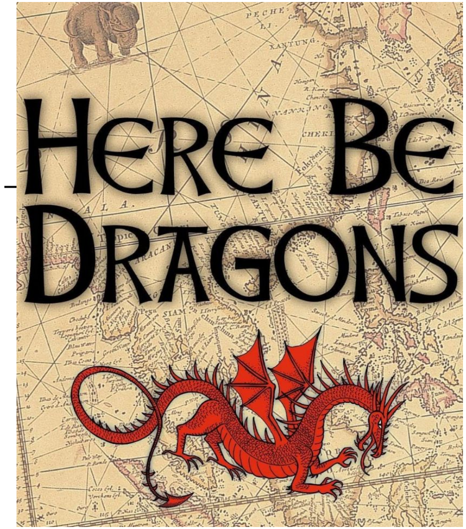
```
   _
+-/ \-+
| (o) |  libcamera
+-----+
```



```
-------------------< libcamera Public API >---------------------------
         ^                                      ^
         |                                      |
         v                                      v
  +-------------+  +-------------------------------------------------+
  |   Camera    |  |  Camera Device                                  |
  |   Devices   |  | +---------------------------------------------+ |
  |   Manager   |  | | Device-Agnostic                             | |
  +-------------+  | |                                             | |
         ^         | |                      +----------------------+ |
         |         | |                      |   ~~~~~~~~~~~~~~~~~~~  |
         |         | |                      |  {  +-------------+  } |
         |         | |                      |  }  |////Image////|  { |
         |         | |                      | <-> |/Processing//|  } |
         |         | |                      |  }  |/Algorithms//|  { |
         |         | |                      |  {  +-------------+  } |
         |         | |                      |   ~~~~~~~~~~~~~~~~~~~  |
         |         | |                      | ====================== |
         |         | |                      |     +-------------+    |
         |         | |                      |     |//Pipeline///|    |
         |         | |                      | <-> |///Handler///|    |
         |         | |                      |     |/////////////|    |
         |         | +----------------------+     +-------------+    |
         |         |                                 Device-Specific |
         |         +-------------------------------------------------+
         |                     ^                        ^
         |                     |                        |
         v                     v                        v
+--------------------------------------------------------------------+
| Helpers and Support Classes                                        |
| +-------------+  +-------------+  +-------------+  +-------------+ |
| |  MC & V4L2  |  |   Buffers   |  | Sandboxing  |  |   Plugins   | |
| |   Support   |  |  Allocator  |  |     IPC     |  |   Manager   | |
| +-------------+  +-------------+  +-------------+  +-------------+ |
| +-------------+  +-------------+                                   |
| |  Pipeline   |  |     ...     |                                   |
| |   Runner    |  |             |                                   |    /// Device-Specific Components
| +-------------+  +-------------+                                   |    ~~~ Sandboxing
+--------------------------------------------------------------------+
```

APPs

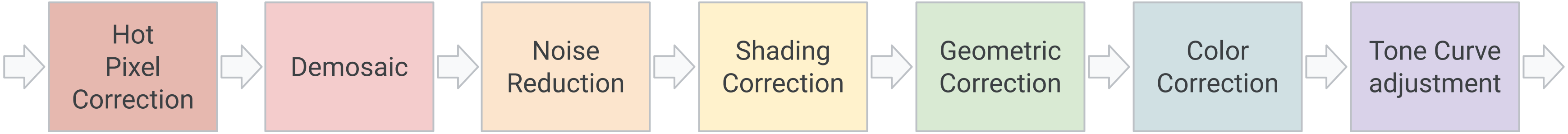
Gstreamer



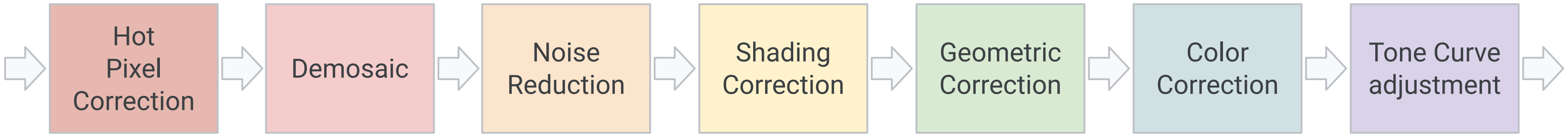
Hardware

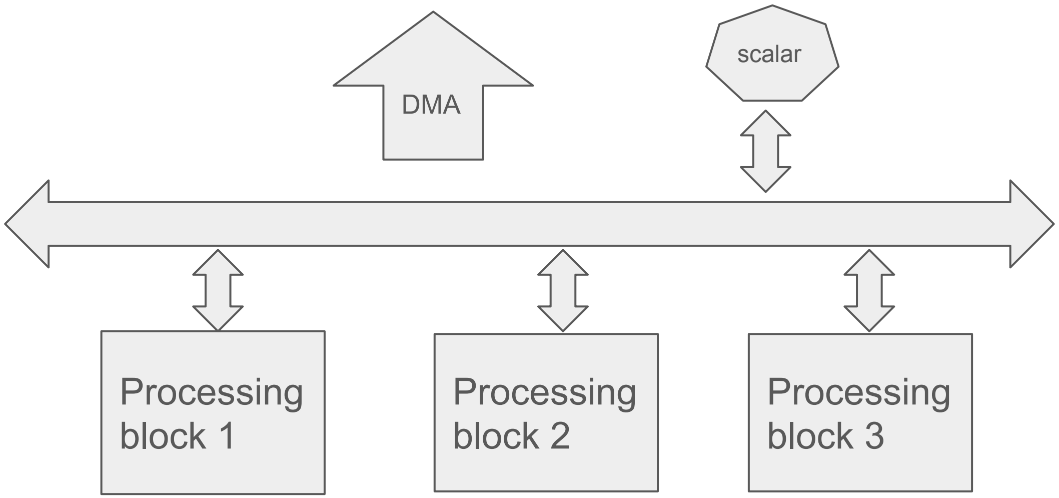
Complex Cameras

Complex cameras



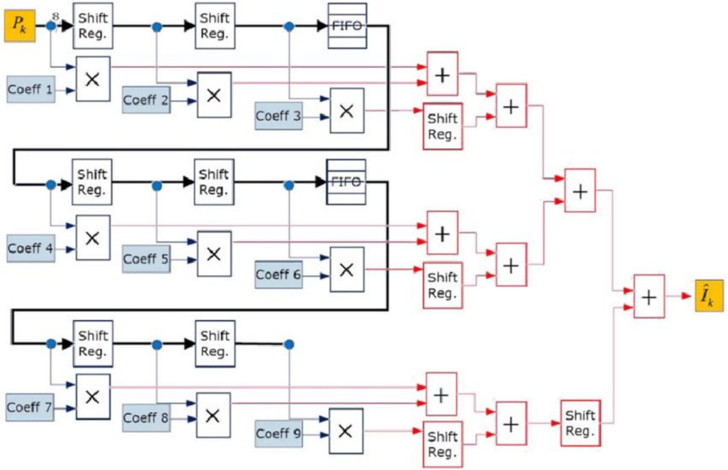
Complex cameras





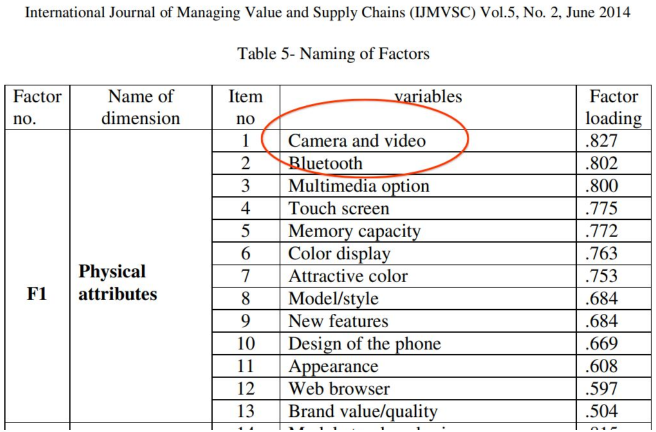
- Super HDR
- Zero Shutter Lag
- Ultra high FPS

IP



Lapray, Pierre-Jean, Luc Gendre, Alban Foulonneau, and Laurent Bigué. "An FPGA-based pipeline for micropolarizer array imaging." *International Journal of Circuit Theory and Applications* 46, no. 9 (2018): 1675-1689.

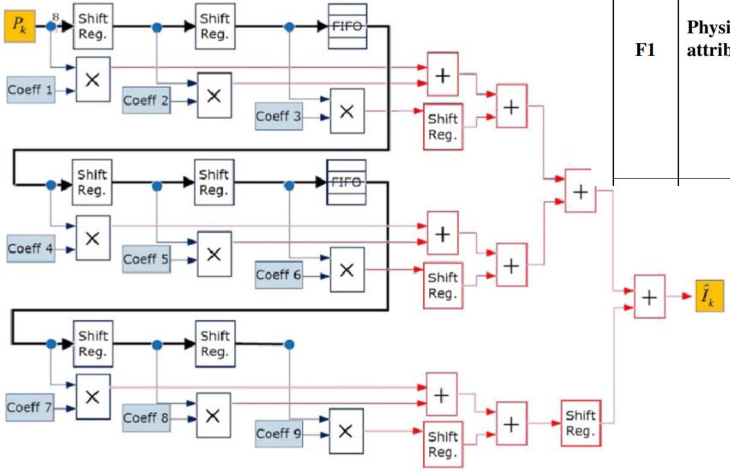
IP



International Journal of Managing Value and Supply Chains (IJMVSC) Vol.5, No. 2, June 2014

Table 5- Naming of Factors

Factor no.	Name of dimension	Item no	variables	Factor loading
F1	**Physical attributes**	1	Camera and video	.827
		2	Bluetooth	.802
		3	Multimedia option	.800
		4	Touch screen	.775
		5	Memory capacity	.772
		6	Color display	.763
		7	Attractive color	.753
		8	Model/style	.684
		9	New features	.684
		10	Design of the phone	.669
		11	Appearance	.608
		12	Web browser	.597
		13	Brand value/quality	.504

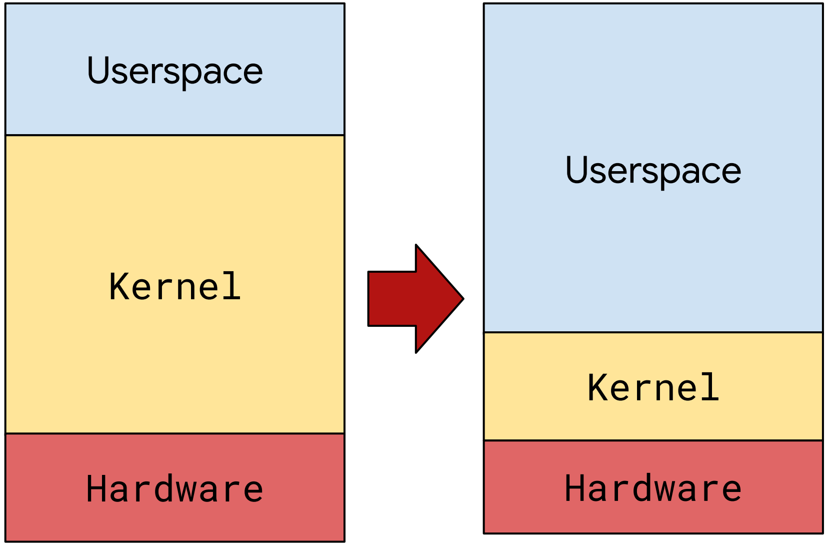


Lapray, Pierre-Jean, Luc Gendre, Alban Foulonneau, and Laurent Bigué. "An FPGA-based pipeline for micropolarizer array imaging." *International Journal of Circuit Theory and Applications* 46, no. 9 (2018): 1675-1689.

~~KCAM~~ ISP

Create a new kernel subsystem designed to support specifically “complex cameras”.

It is NOT a replacement for V4L2.



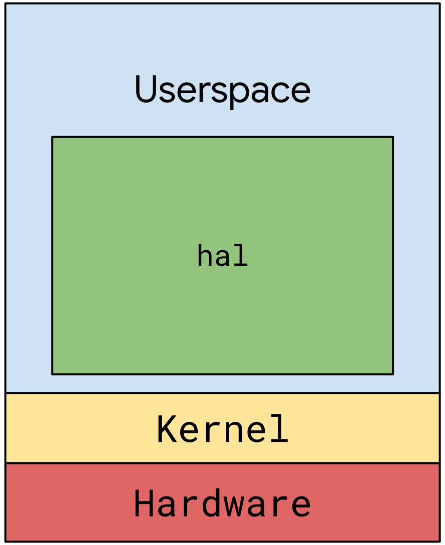
Kcam follows a DRM-like model where the kernel provides basic functionality:

- Scheduling
- Discovery

Everything else is provided by userspace

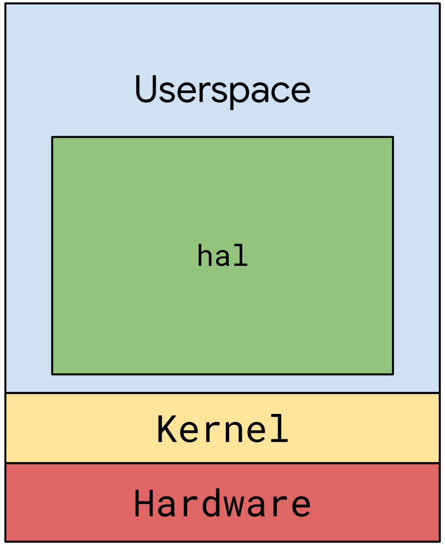
ISP openness

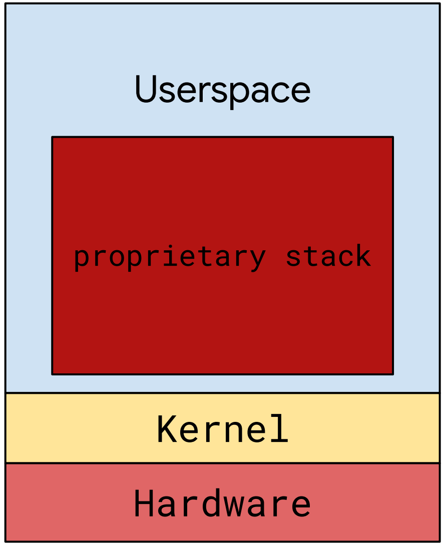
For a driver to be upstreamed, there must be an open source camera stack. That enables standard use of the camera: video conferencing, single-shot photo....



KCAM openness

For a driver to be upstreamed, there must be an open source camera stack. That enables standard use of the camera: video conferencing, single-shot photo....





394th Judicial District Court

Recording of this hearing or live stream is prohibited.

Violation may constitute contempt of court and result in a fine of up to $500 and a jail term of up to 180 days.

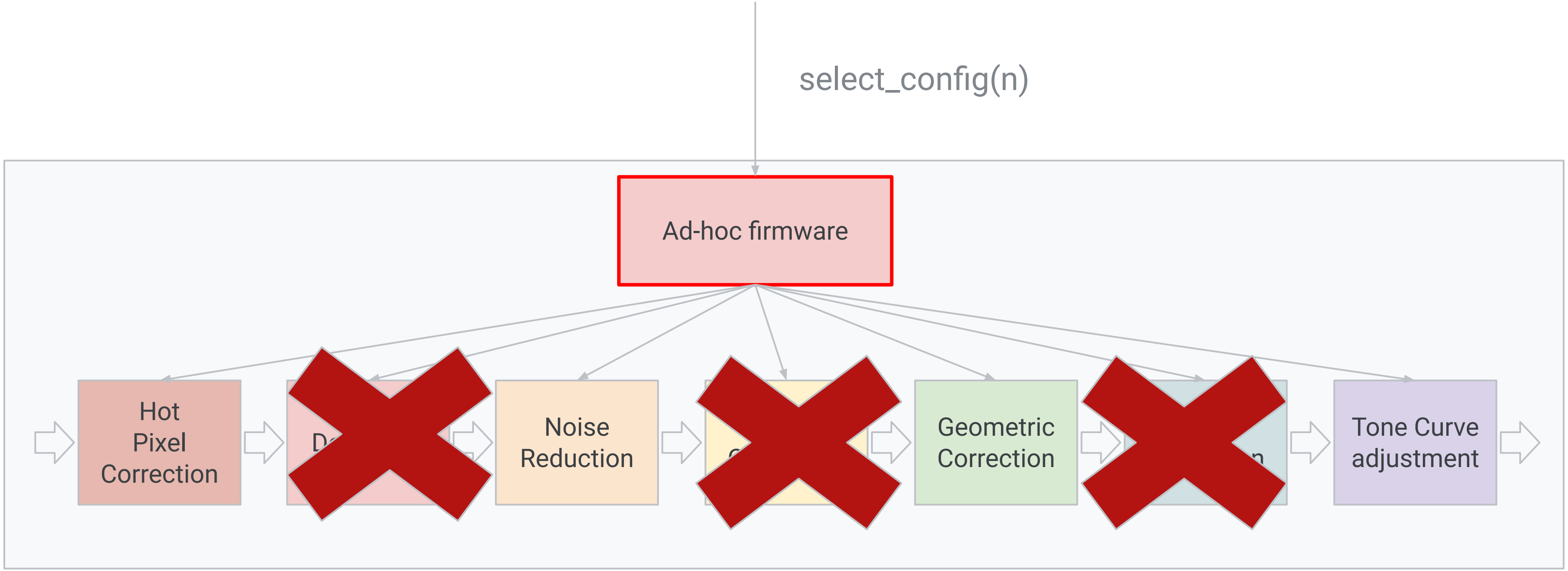
Challenges

The vendor access

- Media maintainers expect that:

  A fully open source upstream driver. Does require opening up hardware access and documenting the API, but not the algorithms used to configure the HW optimally, those can remain closed. However, enough information must be available so an open source implementation can be made.

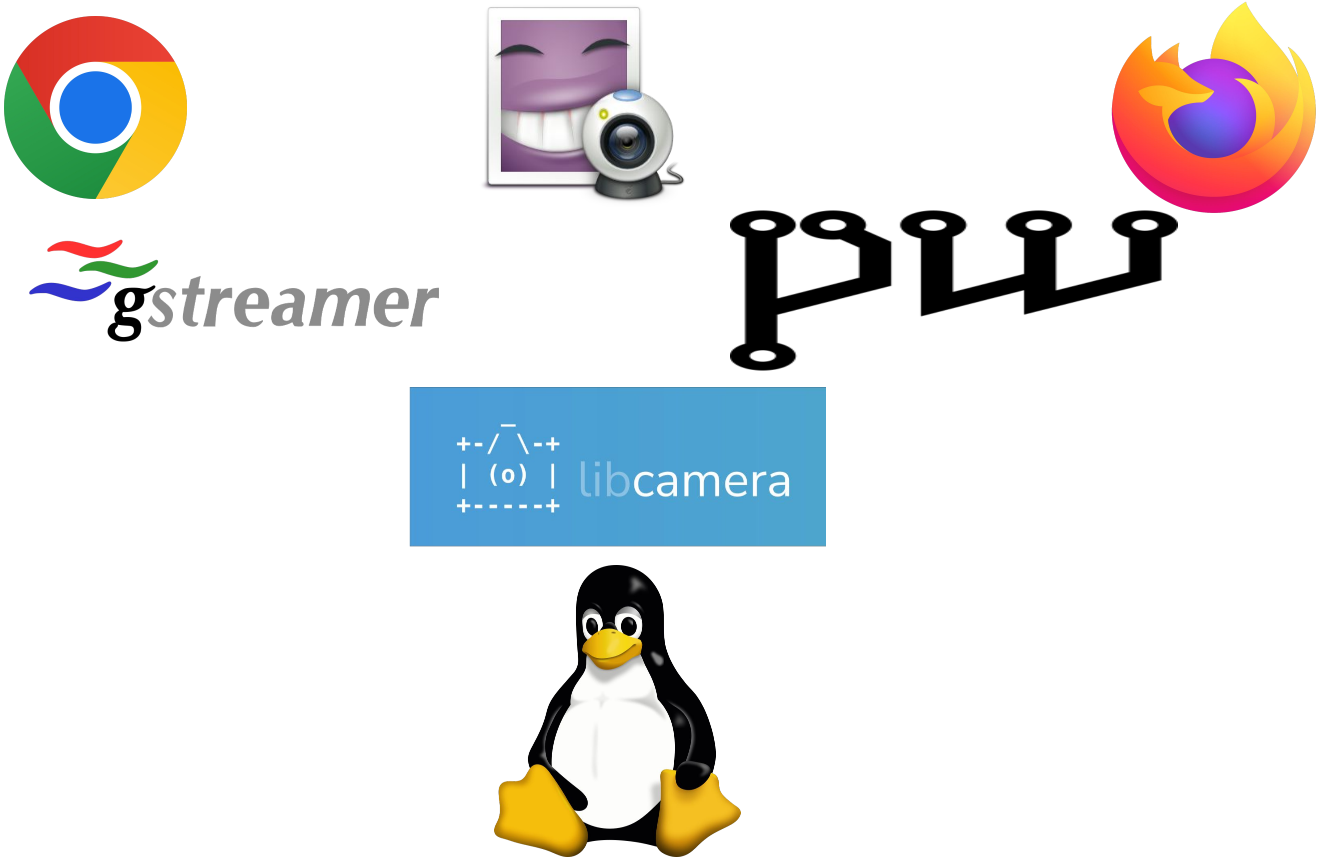
Black box hardware



Unusable hardware

Proposal

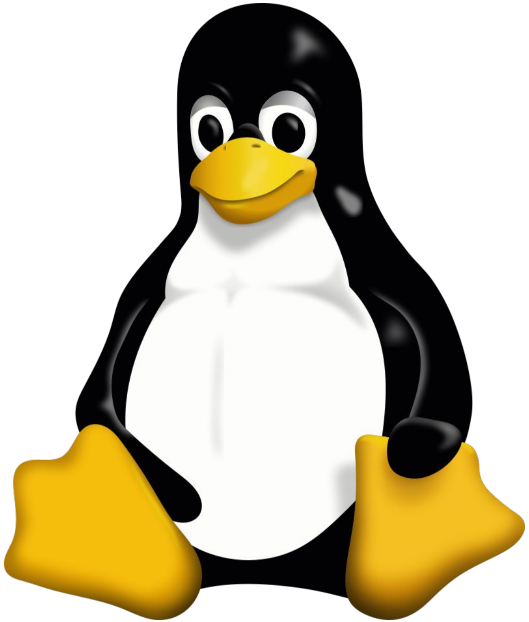
End goal



Chicken and egg problem

- Vendors do not want to invest in a new framework if it does not have a chance to succeed
- Upstream community do not trust vendors until they deliver an complete solution
- Users cannot commit to an open OS if it does not cover their use cases

Kernel component



DKMS package in main

- GPL Licence

Documentation:

https://chromium.googlesource.com/chromiumos/third_party/kernel/+/refs/heads/kcam-6.1/Documentation/userspace-api/isp

https://chromium.googlesource.com/chromiumos/third_party/kernel/+/refs/heads/kcam-6.1/Documentation/driver-api/isp.rst

Source code:

https://chromium.googlesource.com/chromiumos/third_party/kernel/+/refs/heads/kcam-6.1/include/uapi/linux/isp.h

https://chromium.googlesource.com/chromiumos/third_party/kernel/+/refs/heads/kcam-6.1/include/linux/isp/

Camera framework

Alternatives:

- Libcamera based
- Vendor code based

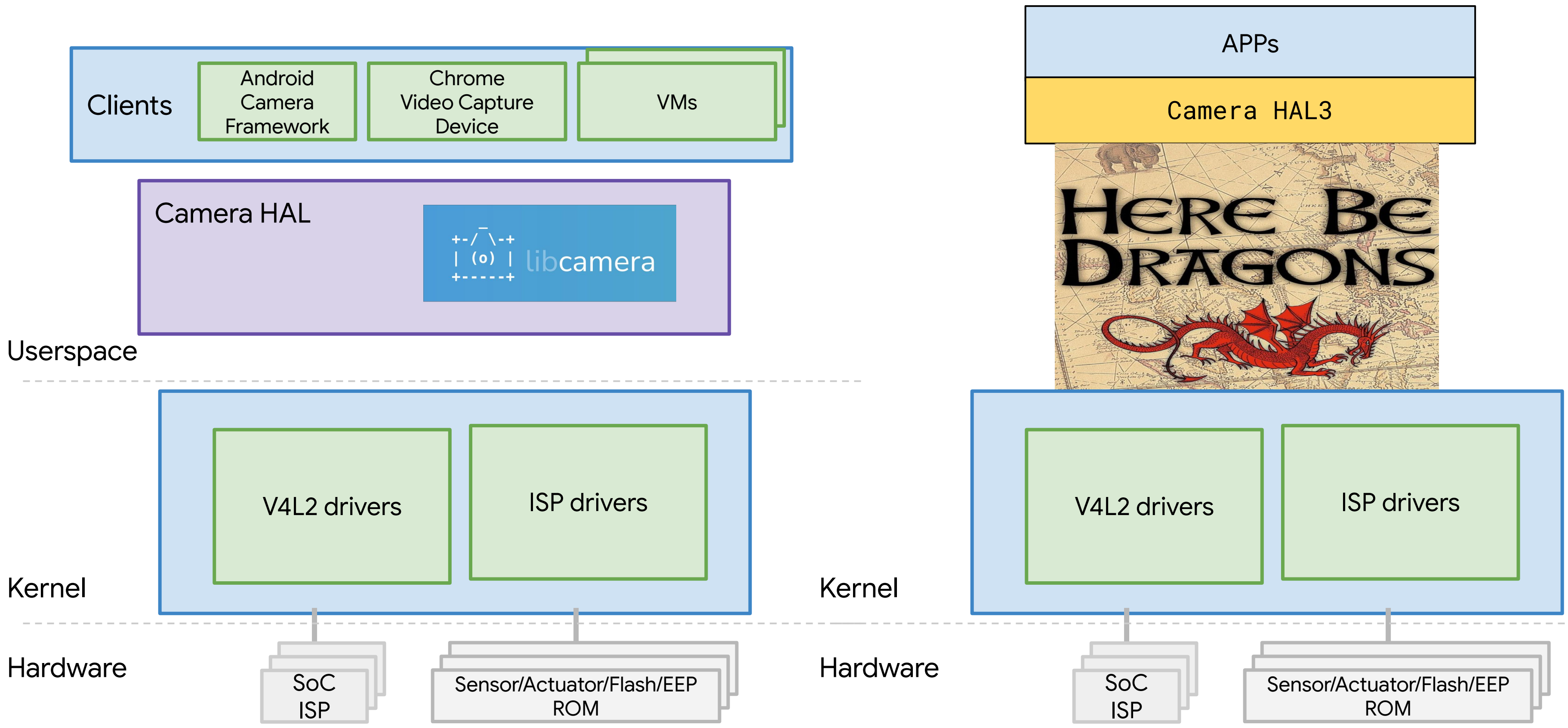
Two components:

- Open License (LGPL, MIT) for main components
- Closed license for "Secret sauce"

Stack

chromeOS

android



Clients

Android Camera Framework

Chrome Video Capture Device

VMs

Camera HAL

libcamera

Userspace

Kernel

V4L2 drivers

ISP drivers

Hardware

SoC ISP

Sensor/Actuator/Flash/EEPROM

APPs

Camera HAL3

Kernel

V4L2 drivers

ISP drivers

Hardware

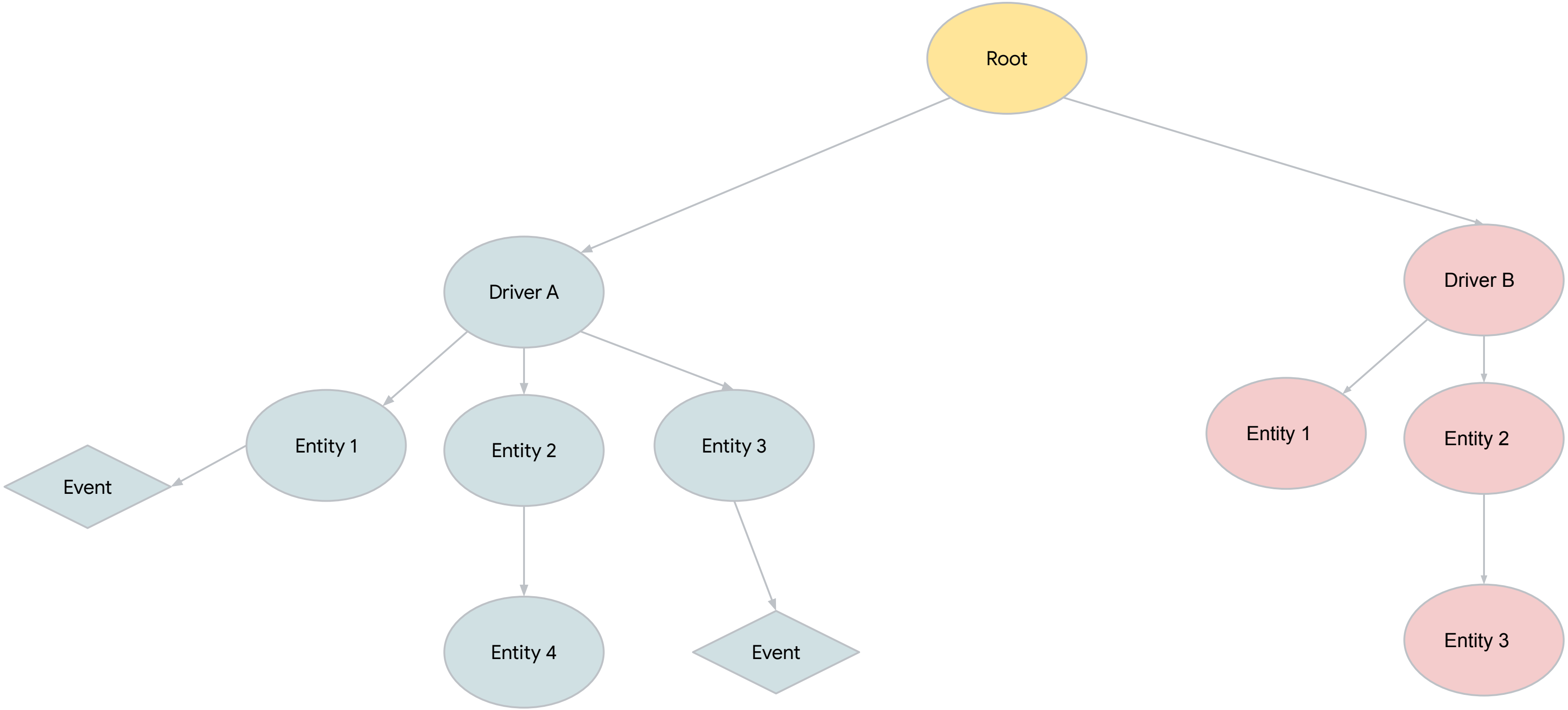
SoC ISP

Sensor/Actuator/Flash/EEPROM

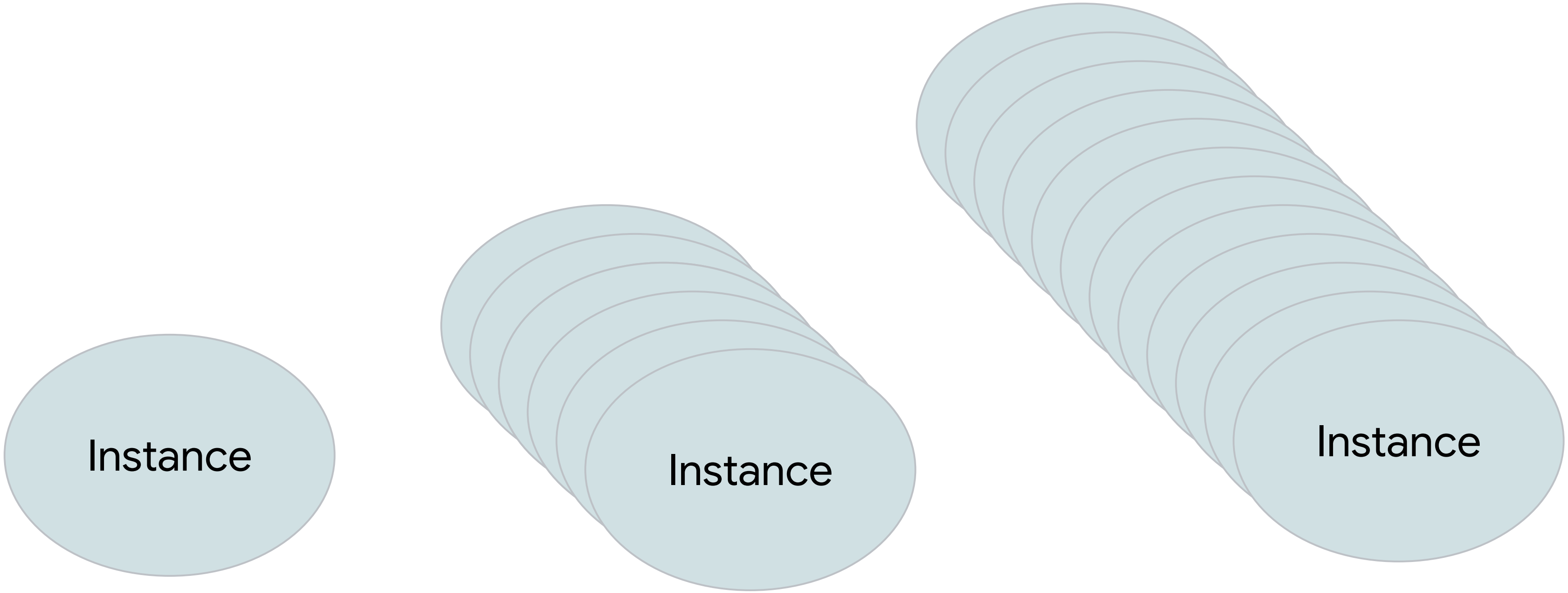
Thank you!

Sergey Senozhatsky, Hidenori Kobayashi, Tomasz Figa, Ricardo Ribalda

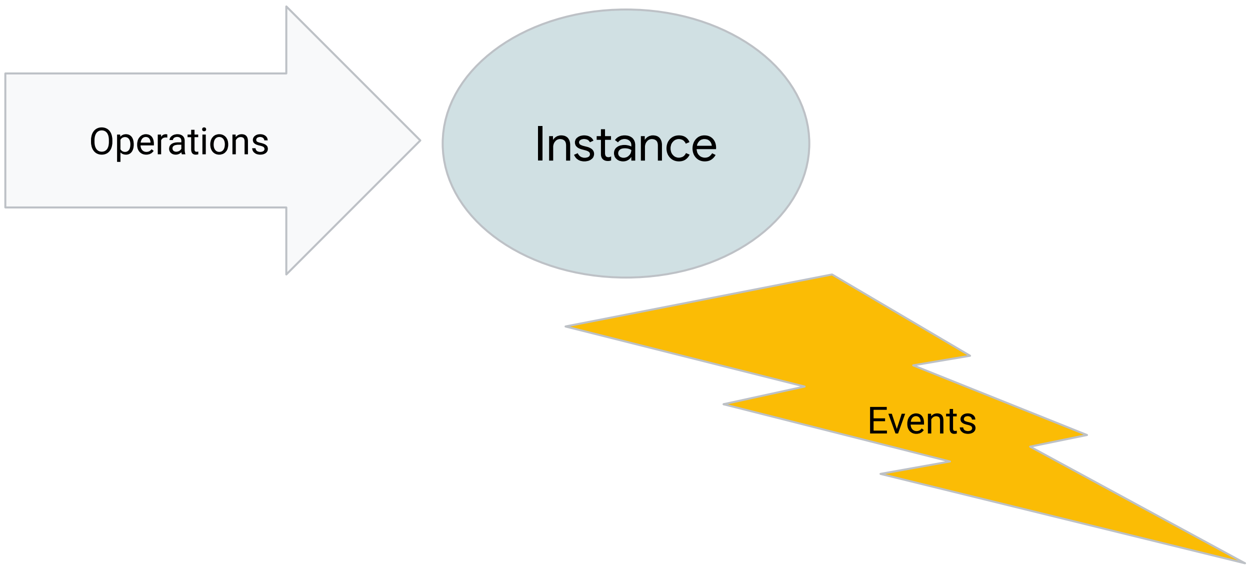
KCAM Internal - Tree



KCAM Internal - Entities



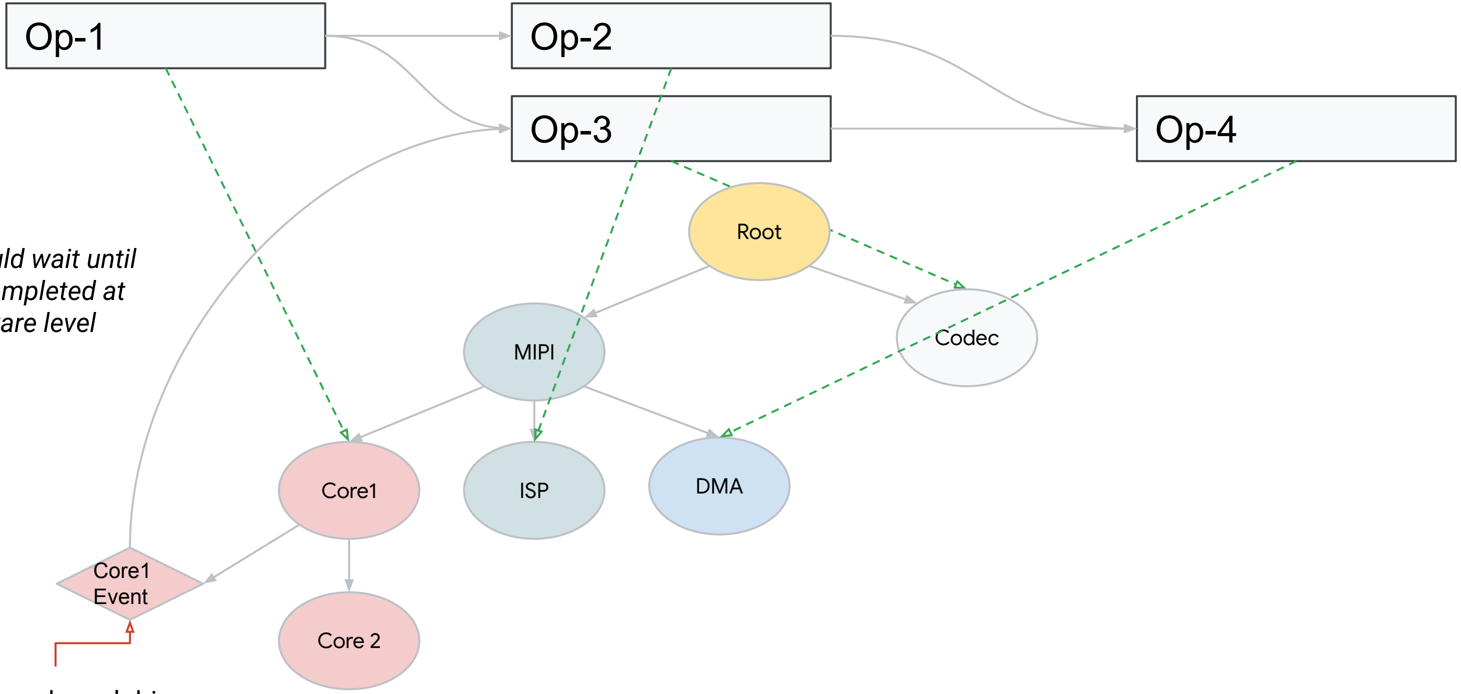
KCAM Internal - Instances



KCAM Internal - Operations

Asynchronous execution of operations

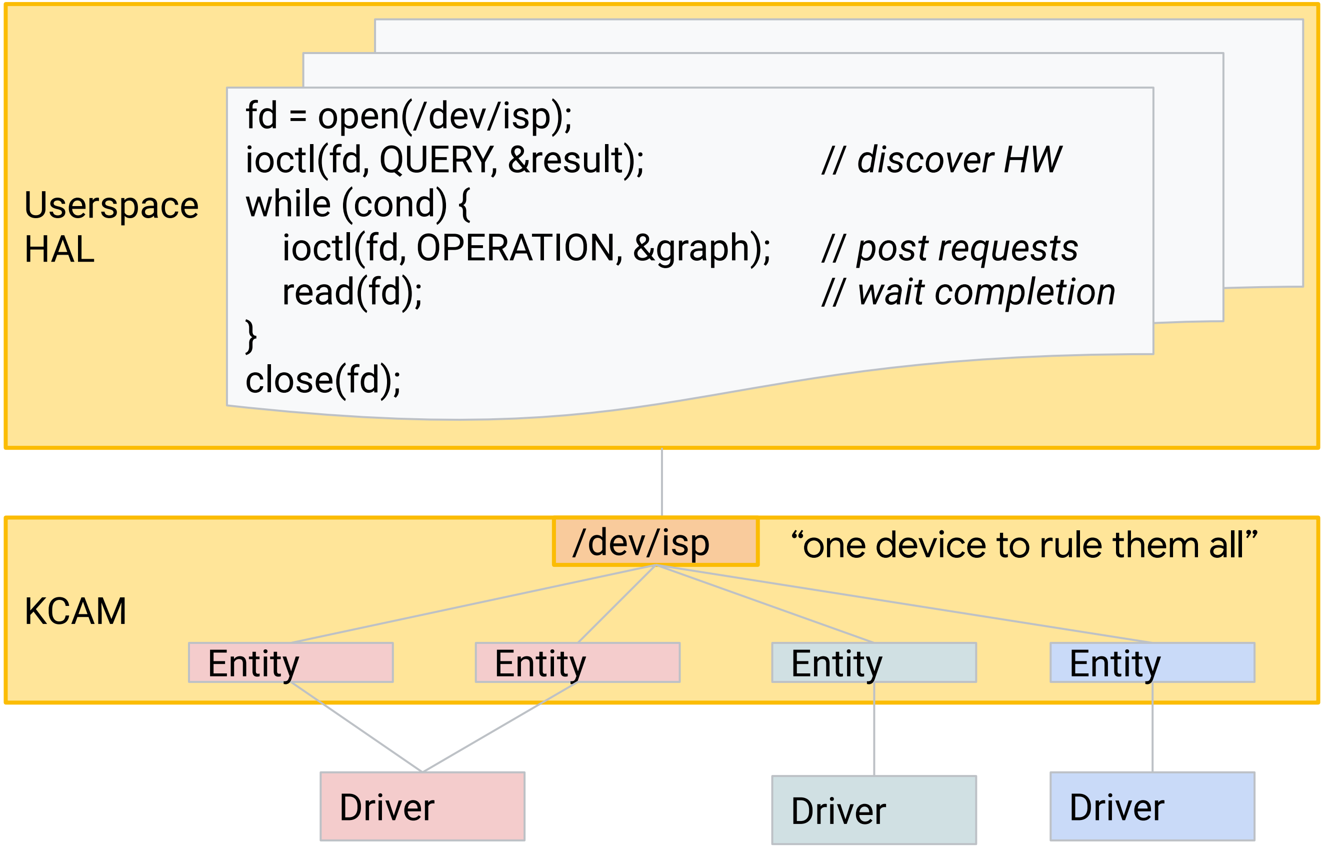
- userspace submits a graph of operations
- dependencies between operations are automatically taken care of



Op-3 should wait until Op-1 is completed at the hardware level

signal from kernel driver

ISP UAPI



```
fd = open(/dev/isp);
ioctl(fd, QUERY, &result);              // discover HW
while (cond) {
    ioctl(fd, OPERATION, &graph);       // post requests
    read(fd);                           // wait completion
}
close(fd);
```

“one device to rule them all”

Entity = abstraction of hardware components driver provides

Tight collaboration between heterogeneous devices! (e.g. stereo cameras, post-processing)