Android camera HAL v3 and Video4Linux 2

Sakari Ailus
<sakari.ailus@linux.intel.com>
2014-10-15

Android camera HAL v1

- Modelled against a digital camera
- Three modes
  - Preview
  - Still capture
  - Video

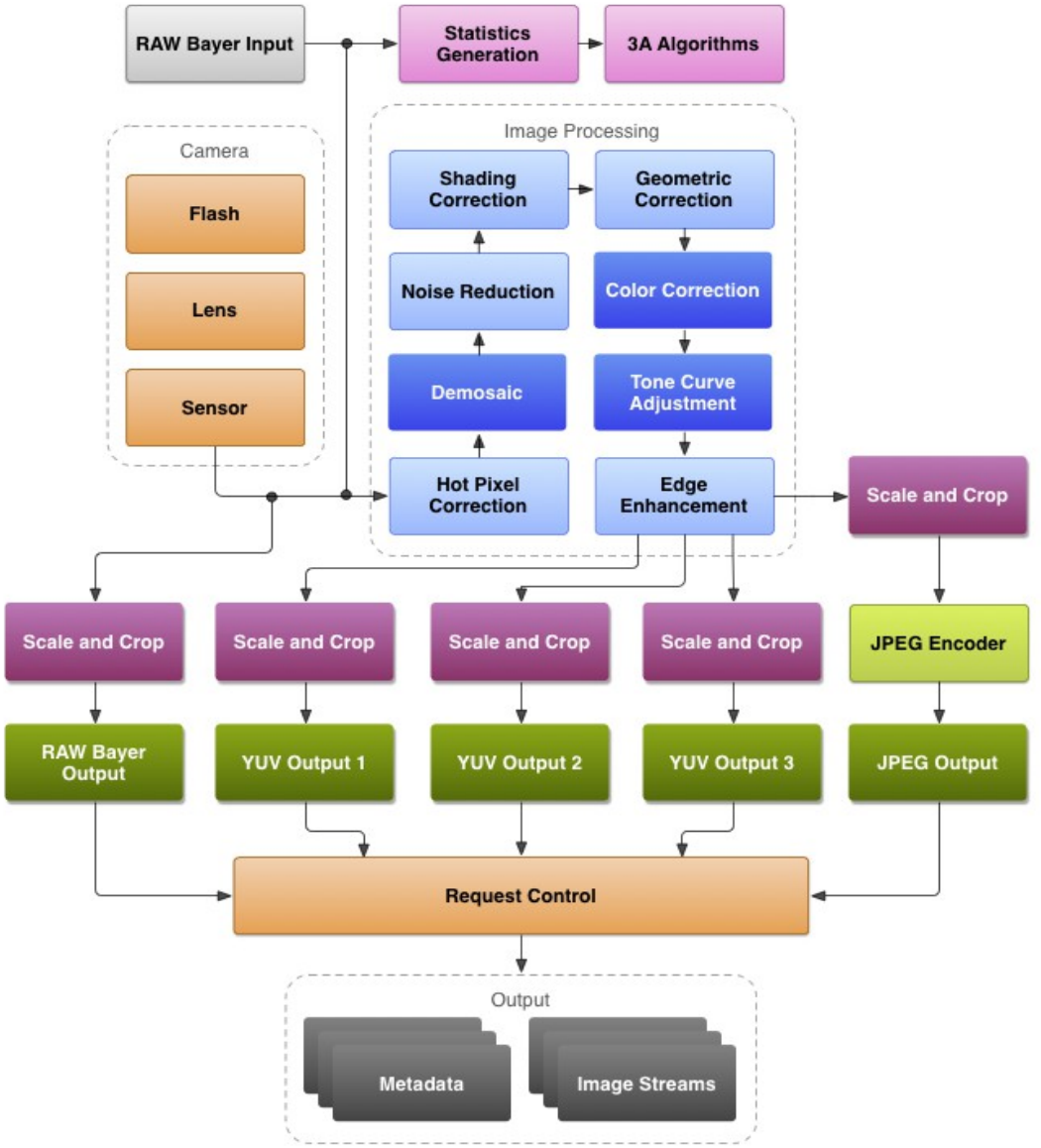
Android camera HAL v3

- Very much not like v1
- Not evolution based on v1
- This is an entirely new API
- Forget what you knew about v1

Android camera HAL v3

- Based on capture requests, each of which translate to a captured image in one or more buffers
  - The full capture configuration is part of the request
- User queues capture requests to the device and receives completed requests back later on

HAL v3 image pipeline model



source: Google [2]

ISP configuration

- Hardware ISPs have low level image processing configuration
  - Lens shading compensation tables
  - Black level correction
  - Linearisation
  - Colour space conversion (RGB to YUV)
  - Statistics configuration (e.g. windows of interest --- location and size)
- This is per-frame configuration, and part of the capture request

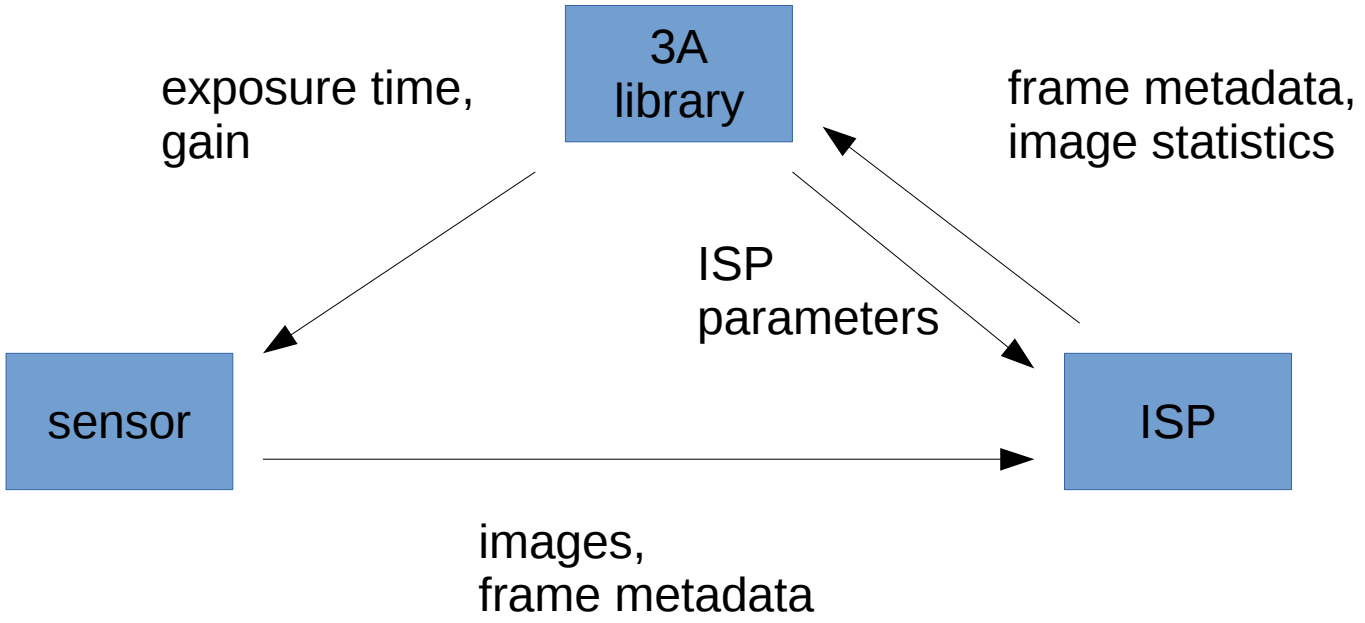
ISP configuration, continued

- The omap3isp driver implements a number of private IOCTLs to implement passing statistics to the user space
- But the functionality is essentially the same as video buffer queues already do
- Discussion in ~ 2010 ended up with a recommendation to use video buffer queues for statistics
  - The statistics formats are device specific
  - Private IOCTL?
  - No implementation yet

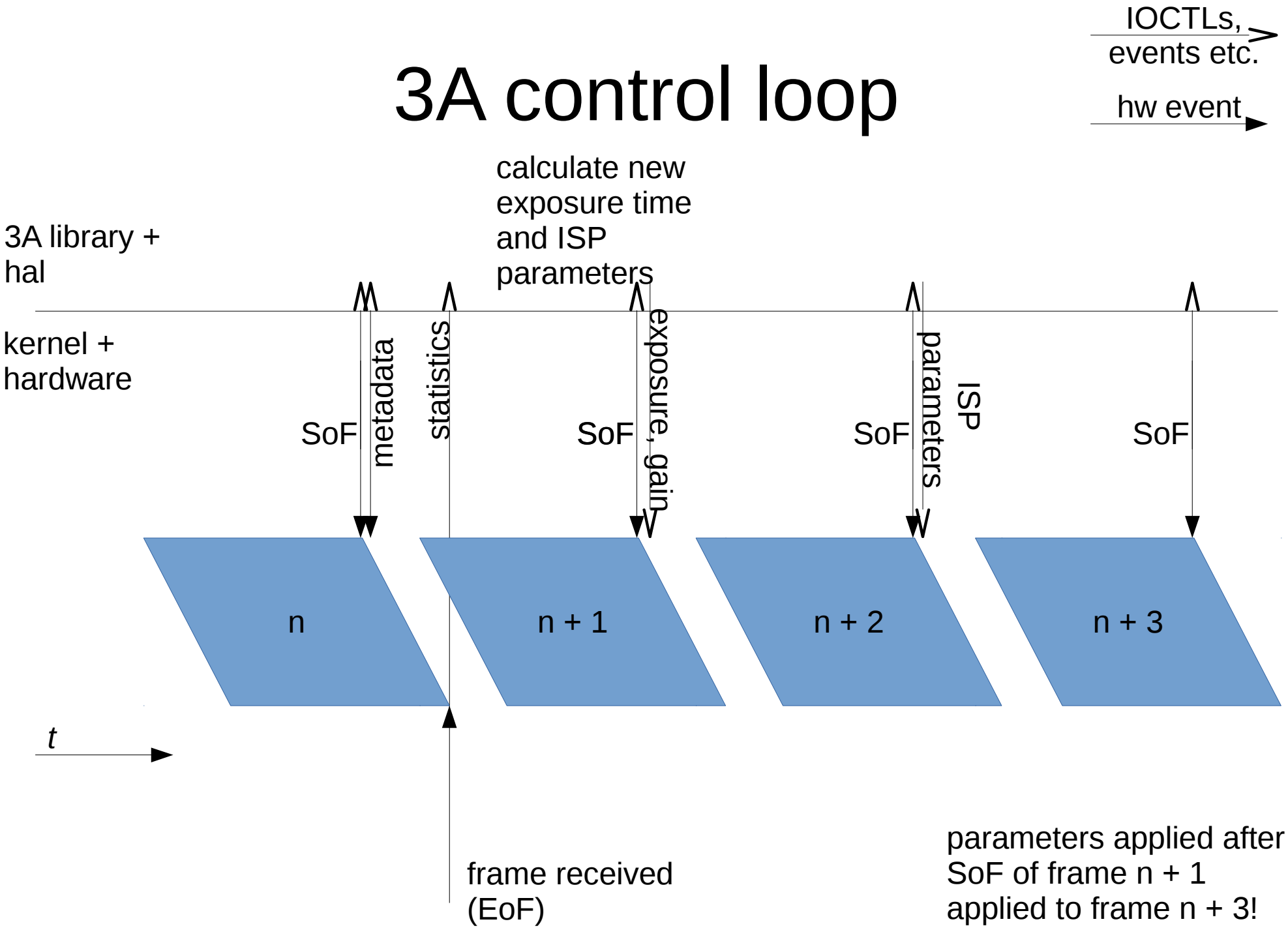
3A library

- Input
  - Exposure and white balance statistics
  - Histogram
  - Frame metadata
- Output
  - Sensor exposure time and gain
  - ISP parameters

3A control loop



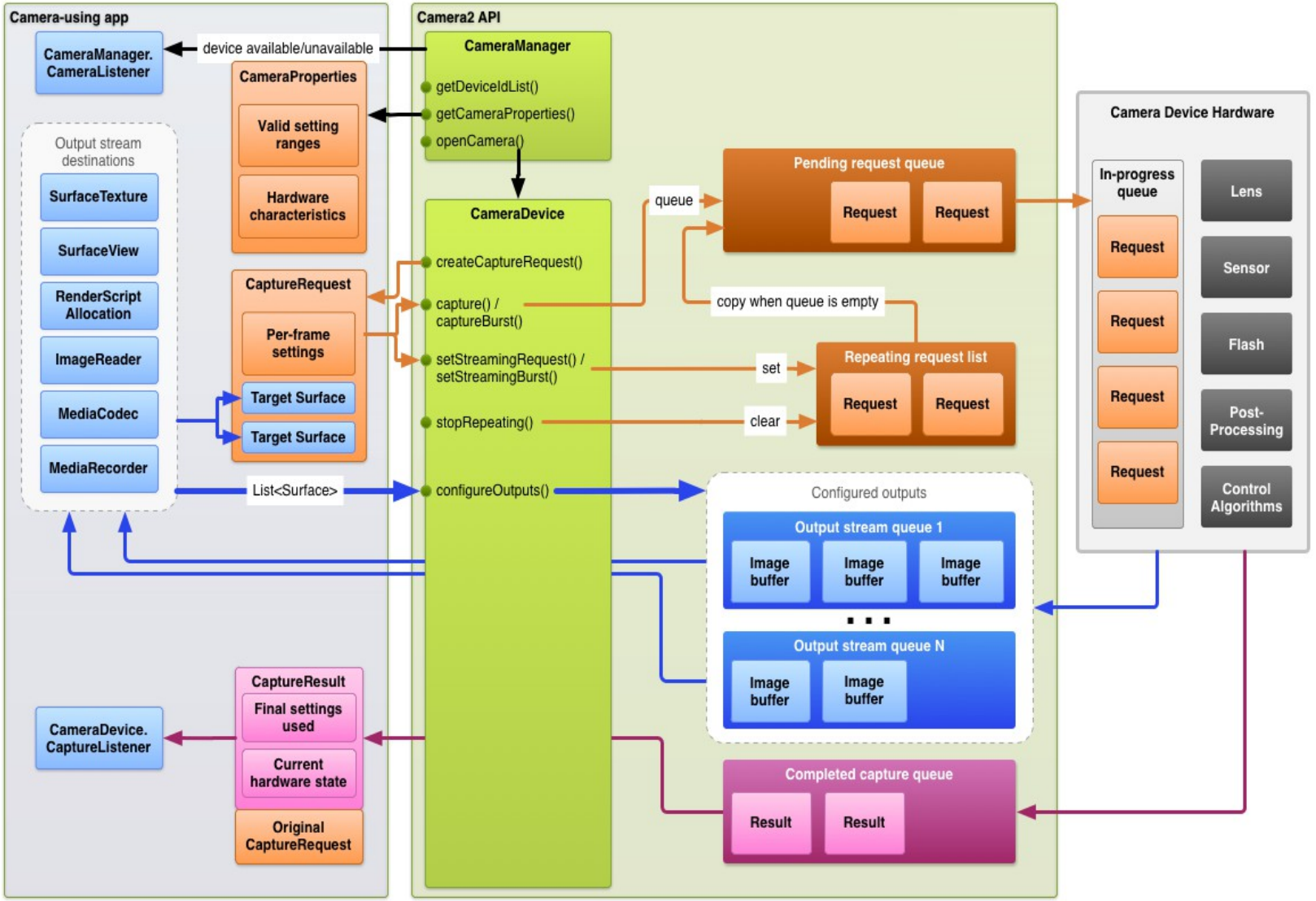
3A control loop



Capture requests

- A capture requrest is about a single frame
  - At most one buffer per stream
- Practical implementations have multiple DMA engines to write the buffers into memory
  - Multiple video buffer queues
- Exactly the same buffers must be returned to the user than were in the capture request

Android camera HAL v3



source: Google [2]

Capture requests

- Include **all** parameters related to capturing a frame, including that calculated by the 3A library
  - ISP configuration
    - Lens shading tables
    - Bayer to GRB conversion parameters
  - Sensor exposure time and gain
  - Lens focus value

Metadata tags

- A metadata tag consists of a single or an array of integer or floating point number(s)
  - Much like controls in V4L2
- A set of metadata tags is related to a single capture request

Metadata tags

- Metadata tags are consumed by HAL
  - HAL configuration
  - 3A library configuration
    - E.g. AWB mode
  - Indirect effects on hardware configuration
- Or produced by HAL
  - Hardware produced statistics conversion to a hardware independent format
    - Such as the histogram

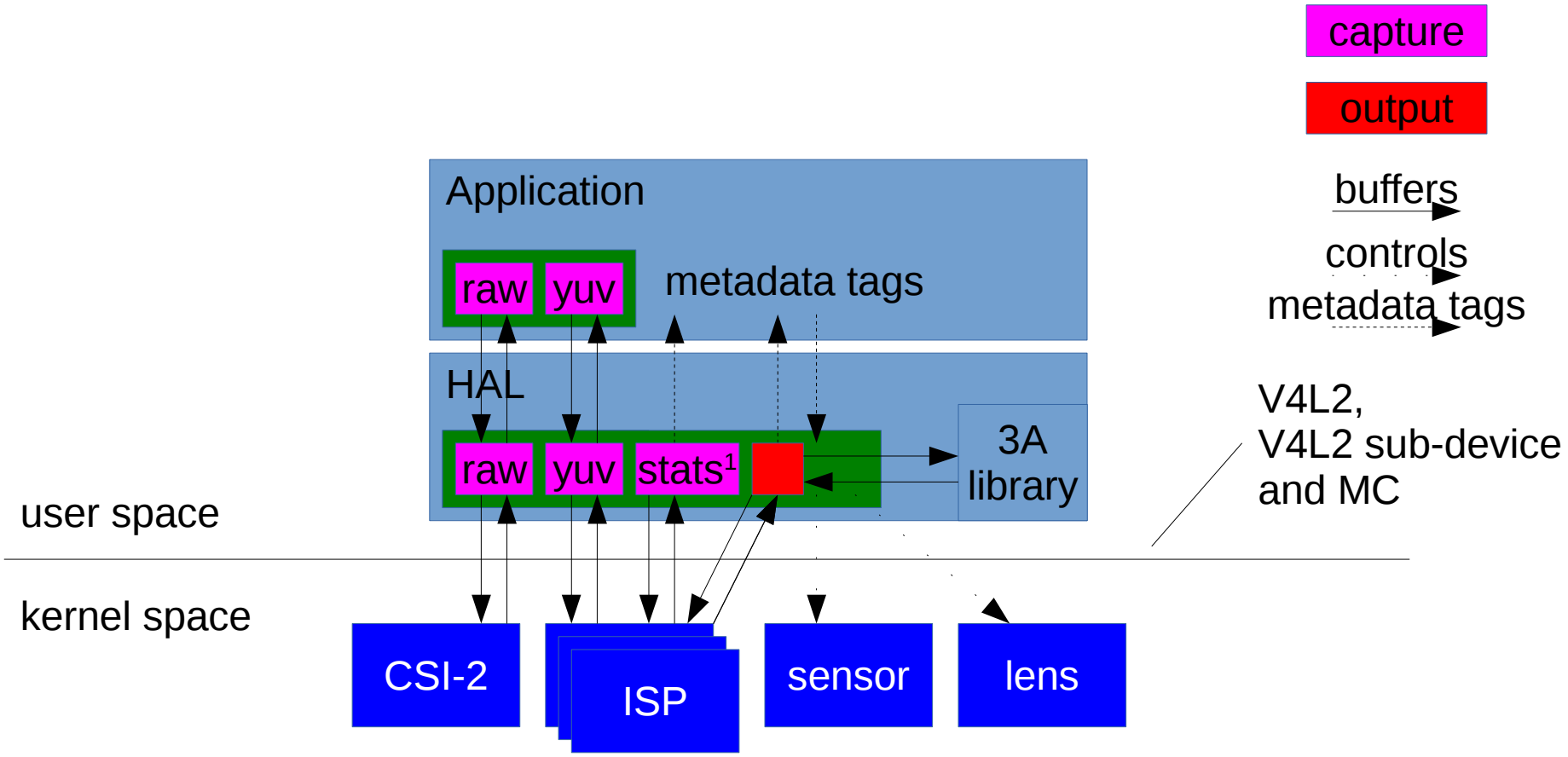
Device usage flow from camera framework towards HAL

1. Open the device.
   camera_module_t.common->open()
2. camera3_device_t->ops->initialize()
3. Configure the streams.
   camera3_device_t->configure_streams()
4. Allocate stream buffers.
   camera3_device_t->ops->register_stream_buffers()

Device usage flow, continued

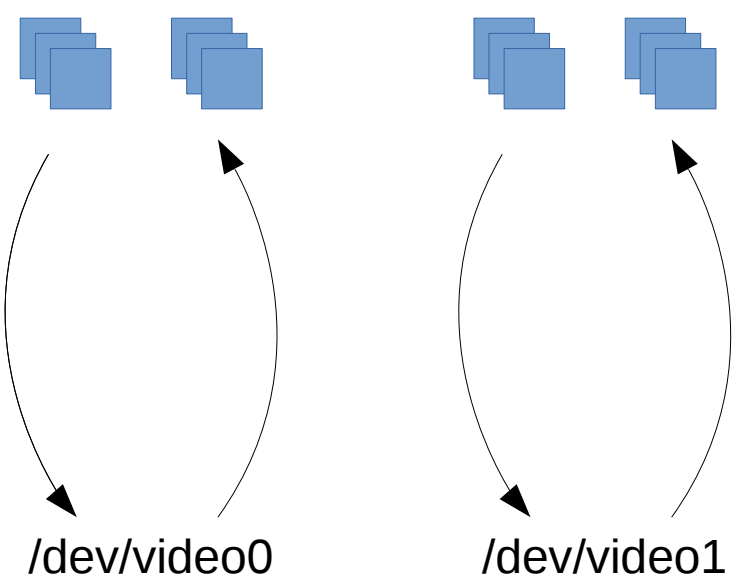
5. Construct a capture request and send it to HAL.
   camera3_device_t->ops->process_capture_request()
6. HAL notifies the framework of a started capture request.
   camera3_callback_ops->notify()
7. HAL notifies the framework of a finished capture request.
   camera3_callback_ops->process_capture_result()
8. Jump back to 5 or continue.
9. camera3_device_t->common->close() may be called to close the device.

Capture requests, HAL and V4L2

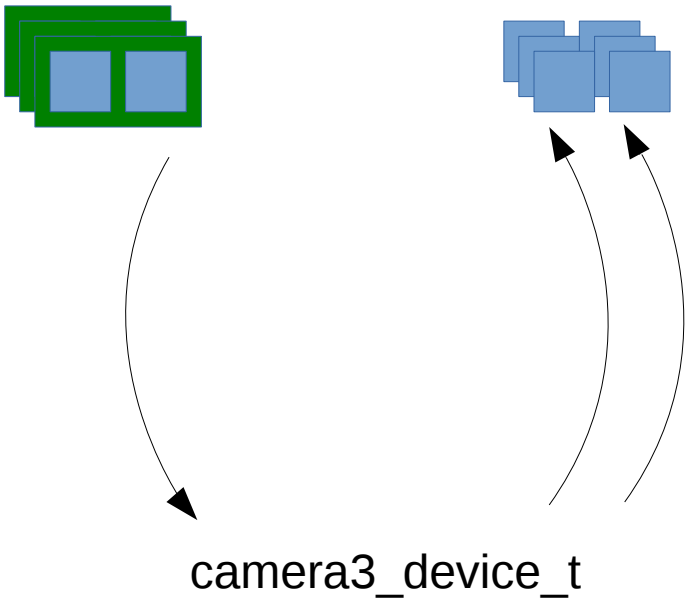


[1]Image statistics (AEWB, histogram and AF) and image metadata from the sensor

Capture requests vs. video buffers



V4L2



Android camera HAL v3

Capture requests and video buffer queues

- There's no standard way in V4L2 to queue multiple buffers to independent video buffer queues with the guarantee of capturing the frame from a single image
  - Not all capture requests come with a buffer for every stream
- The sensor and lens devices already depend on timing
  - Still unreliable and difficult to implement for video buffers
- A new video buffer flag could be used to signal not to pass the buffer to the device yet
  - E.g. V4L2_BUF_FLAG_POSTPONE
- Together with the sequence number

Capture requests and video buffer queues

- No guarantee on buffer ordering in V4L2, but individual drivers could guarantee this
  - A small piece in the puzzle, so probably not worth spending much attention now
- Buffer index can be used to connect buffers related to a queued request to those that are dequeued by HAL

References

[1] https://source.android.com/devices/camera/camera3.html

[2] https://source.android.com/devices/camera/camera3_requests_methods.html

[3] https://android.googlesource.com/platform/hardware/libhardware/+/master/include/hardware/camera3.h

[4] https://android.googlesource.com/platform/system/media/+archive/master/camera/docs.tar.gz

[5] http://developer.android.com/reference/android/hardware/Camera.html