V4L2 Explicit Synchronization

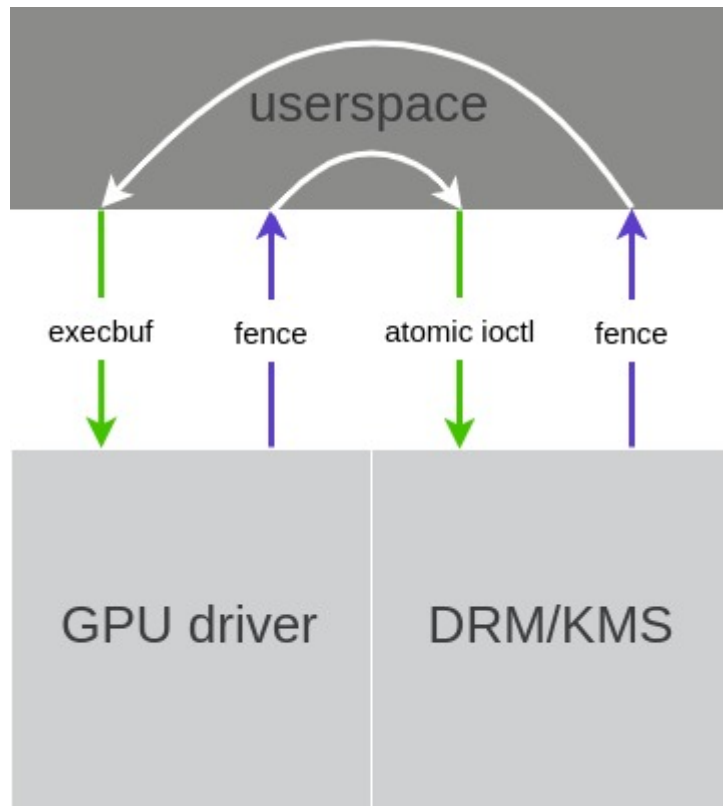## Gustavo Padovan

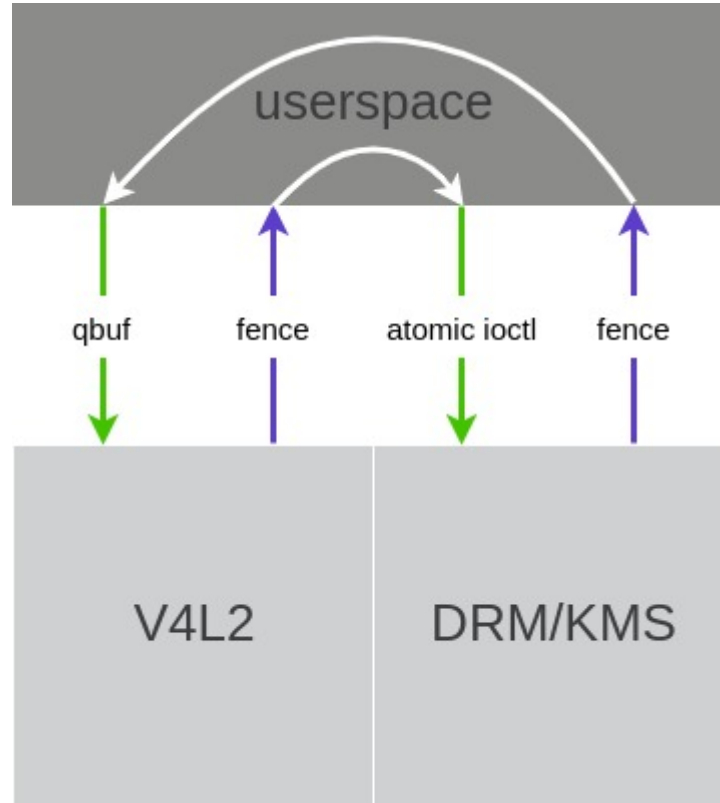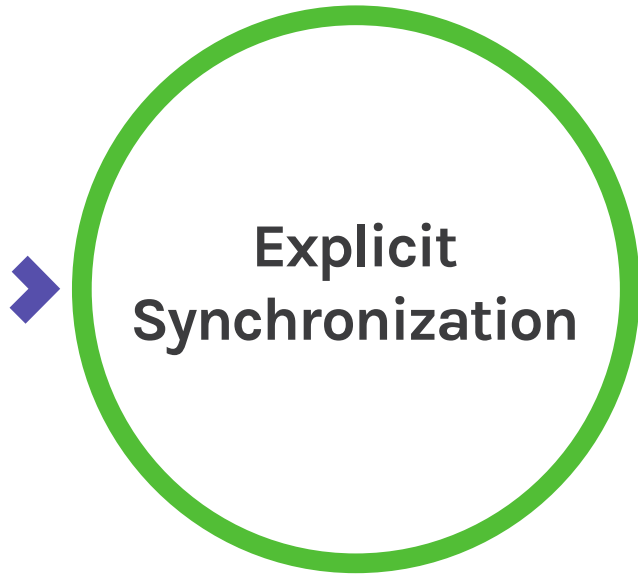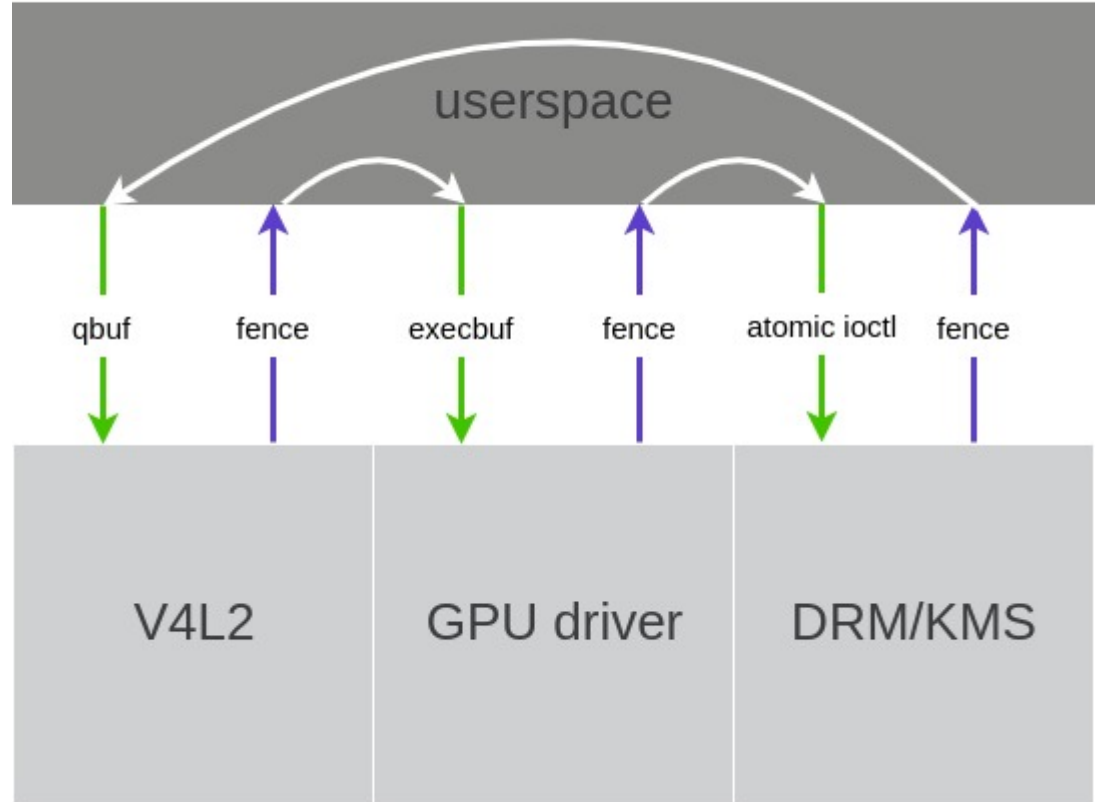**Principal Software Engineer @ Collabora**

COLLABORA
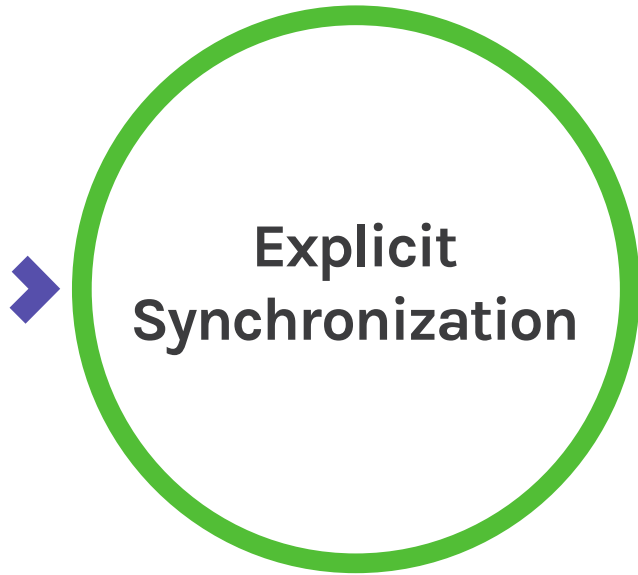
Explicit Synchronization

Explicit Synchronization

userspace

qbuf   fence   atomic ioctl   fence

V4L2   DRM/KMS

Explicit Synchronization

userspace

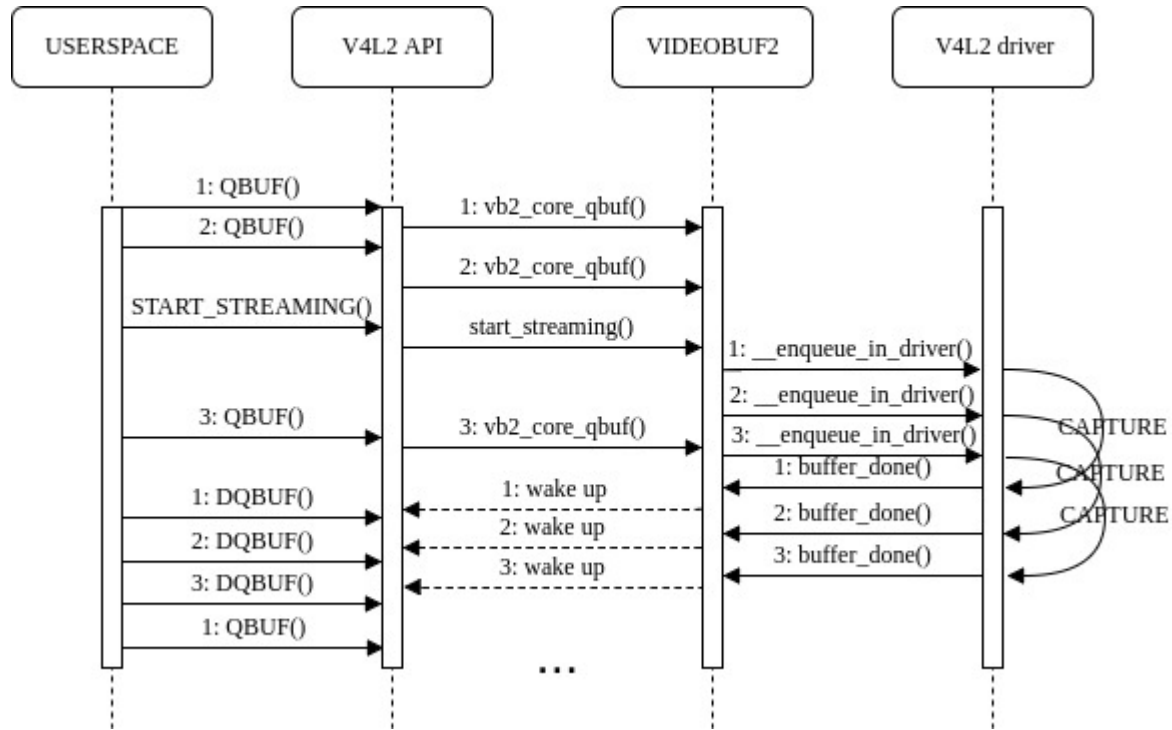qbuf | fence | execbuf | fence | atomic ioctl | fence

V4L2 | GPU driver | DRM/KMS

V4L2 Explicit Synchronization

- Add fences to CAPTURE and OUTPUT queues
- In-fence: fences to wait before using the buffer
- Out-fence: signals when the buffer is ready

V4L2 – Before Fences

V4L2 – in-fences

- in-fence: Passed in `QBUF()` fence_fd field
- `V4L2_BUF_FLAG_IN_FENCE` should be set
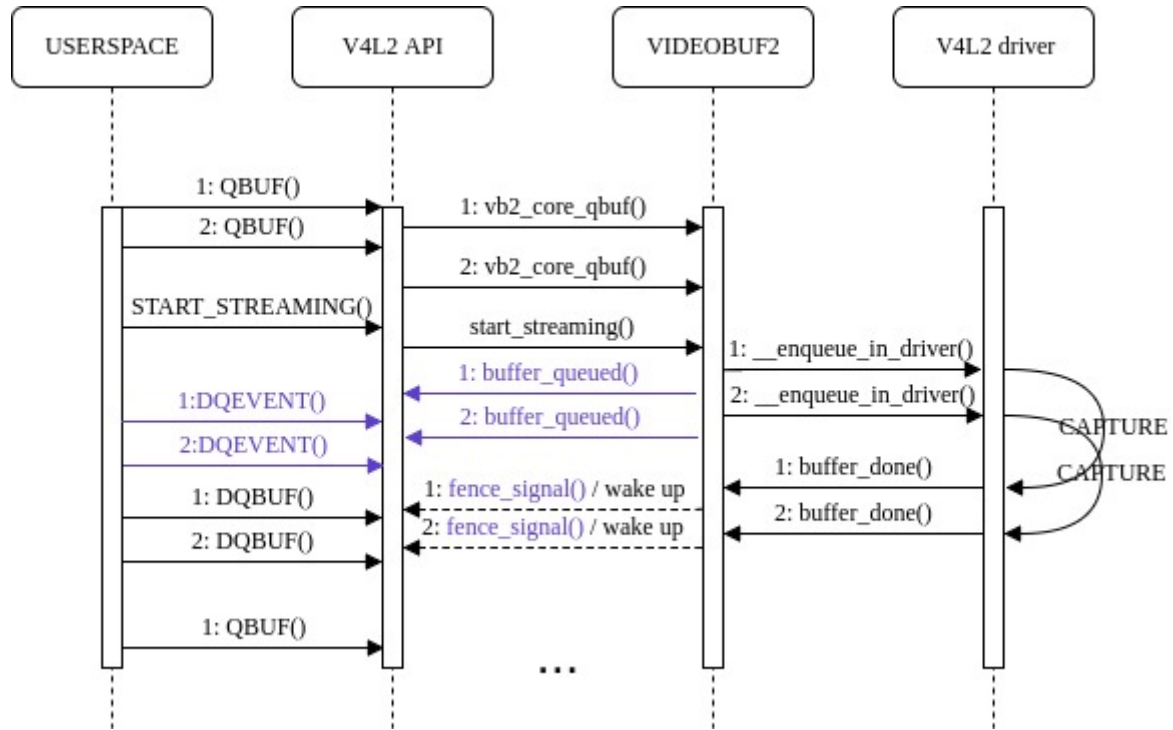- Can't be queued to driver before fence signal

V4L2 – out-fences

- QBUF() should have V4L2_BUF_FLAG_OUT_FENCE
- Remember: No guarantee of ordering
- new V4L2 event: V4L2_EVENT_OUT_FENCE
- Call VIDIOC_SUBSCRIBE_EVENT()
- Receive event with DQEVENT()
- Event provide buffer *index* and *out_fence_fd*

V4L2 – After fences

V4L2 – current RFC

- V4L2-event changes
- ordered_in_driver flag (requirement for now)
- ordered_in_vb2 flag (OUTPUT/some m2m queues)

V4L2 - usecases

- Android/ChromeOS HAL3/Camera App
- Color converter and scalers (ordered)
- m2m/encoders
- Synchronize audio/video - Gstreamer
- Capture to networking (partial fences)

Thank you!

Gustavo Padovan
gustavo.padovan@collabora.com
www.padovan.org
www.collabora.com