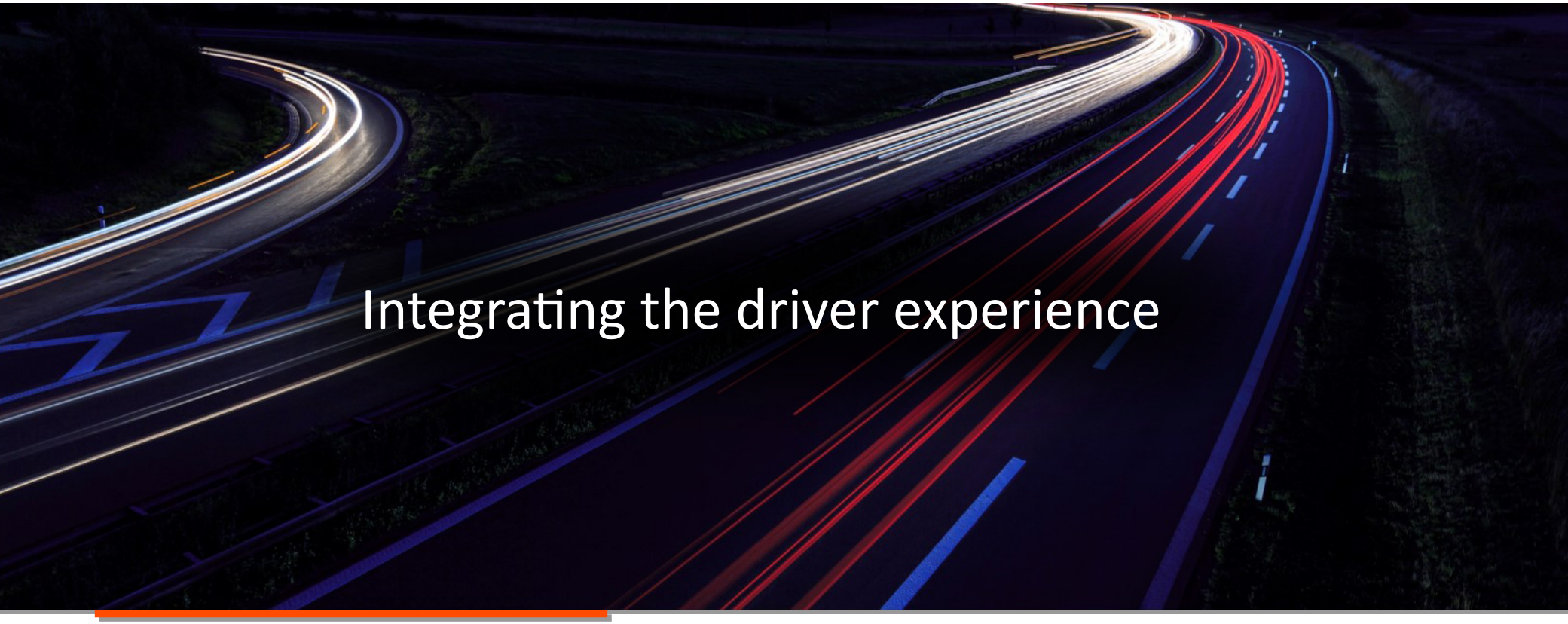
OPENSYNERGY



VirtIO GPU 3D Workshop

How to achieve zero-copy on Renesas SoC?

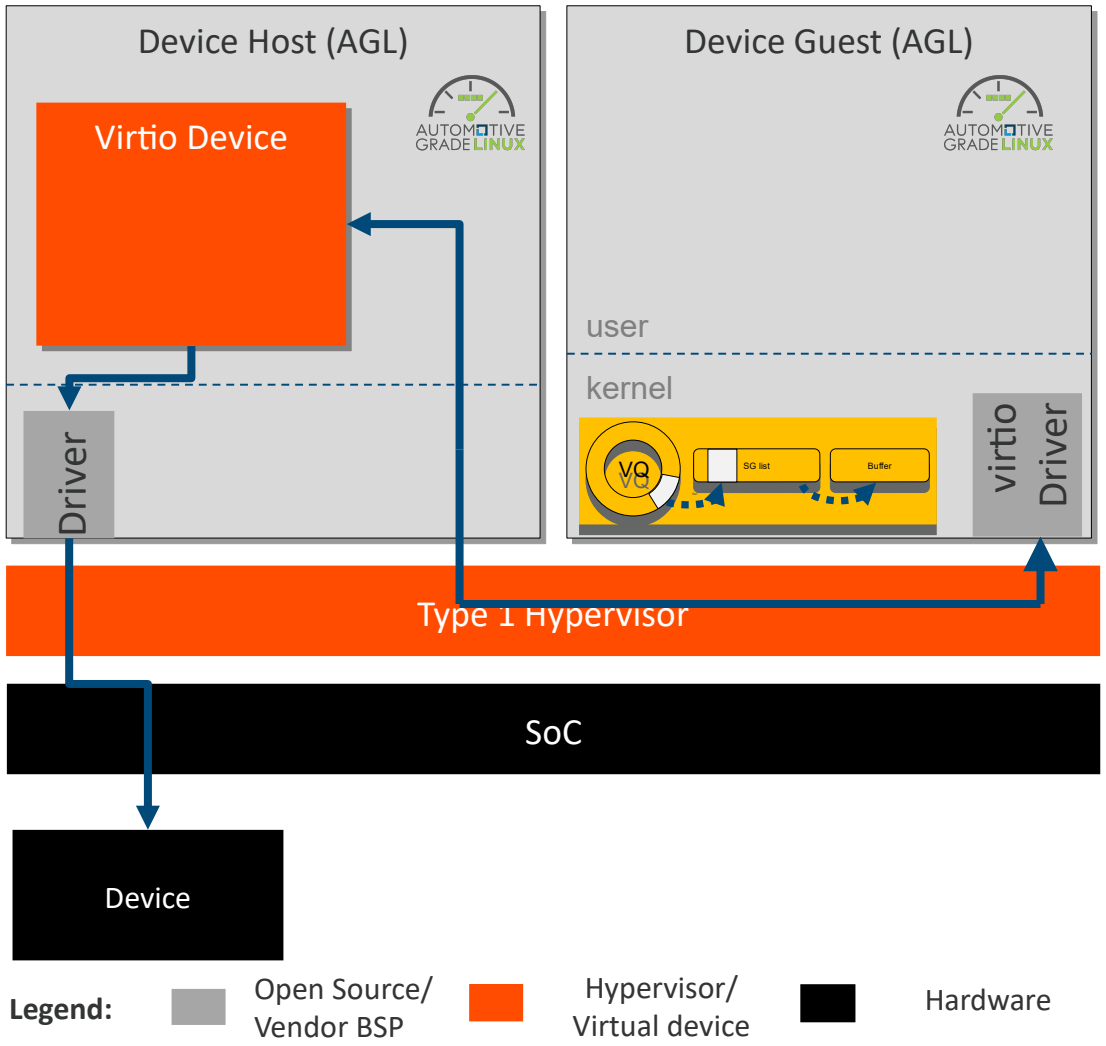
public

Agenda

OPENSYNERGY

1) Virtio Devices Architecture
2) Virtio GPU Architecture
3) Why zero-copy is important
4) Status of zero-copy support available within community
5) How zero copy works in 2D case
6) Resource create in 3D case
7) Zero-copy in 3D case
8) Challenges

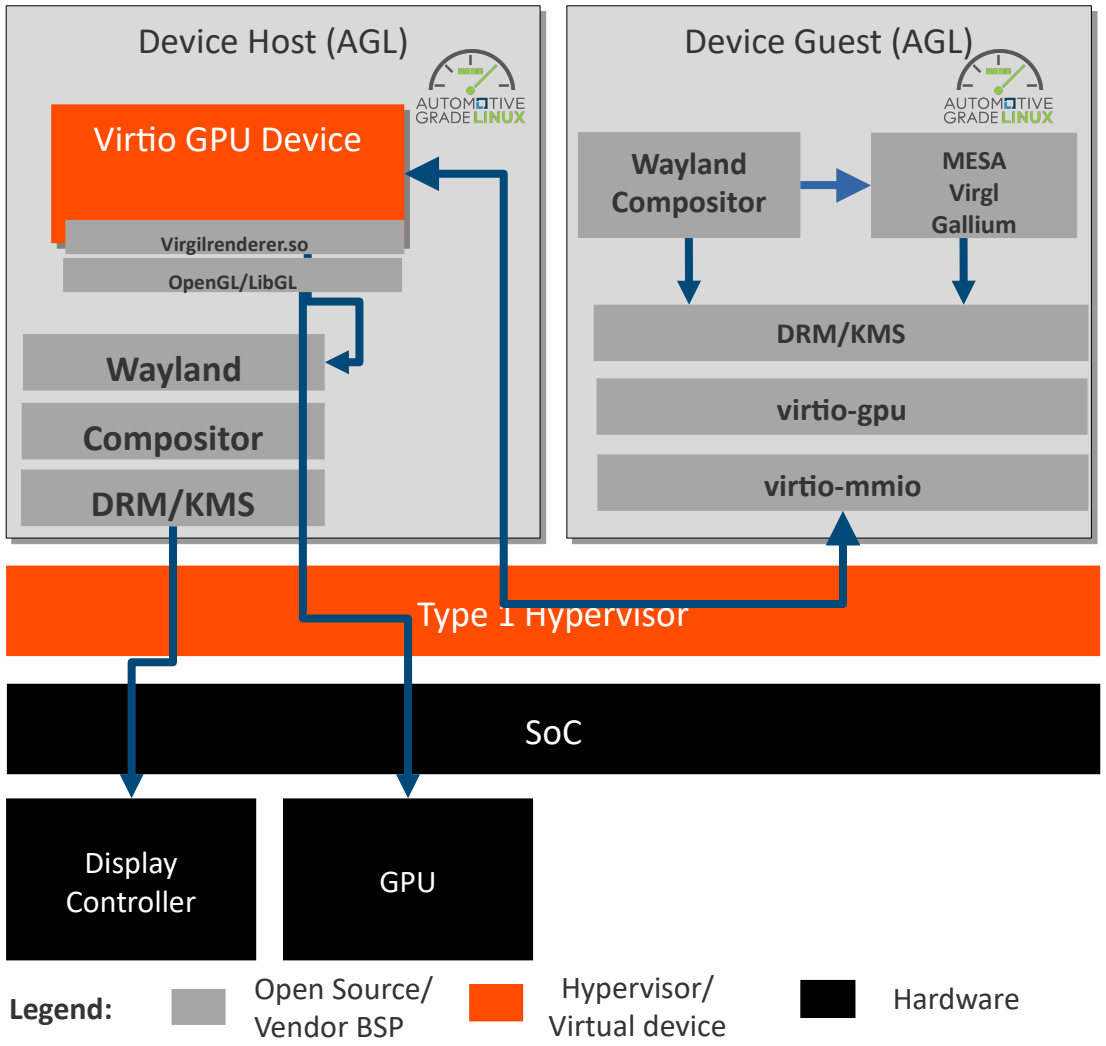
Virtio Devices Architecture



- Virtual devices run as user space process
- Re-use of existing Linux driver infrastructure
- Device guest uses pure open source virtio drivers as part of virtio-aarch64 AGL machine
- Apps running on Device Host may access the native driver directly
- With small change, this illustration is applicable to type 2 hypervisor.

Rusty Russell. virtio: towards a de-facto standard for virtual I/O devices. 2008

Virtio GPU Architecture



- 2D display virtualization
  - Uses native window systems
  - Zero copy by using dma buffers
- 3D GPU virtualization
  - Uses native OpenGL drivers
  - Takes abstract GPU commands from guest employing intermediate shader languages (TGSI)

Why zero-copy is important

OPENSYNERGY

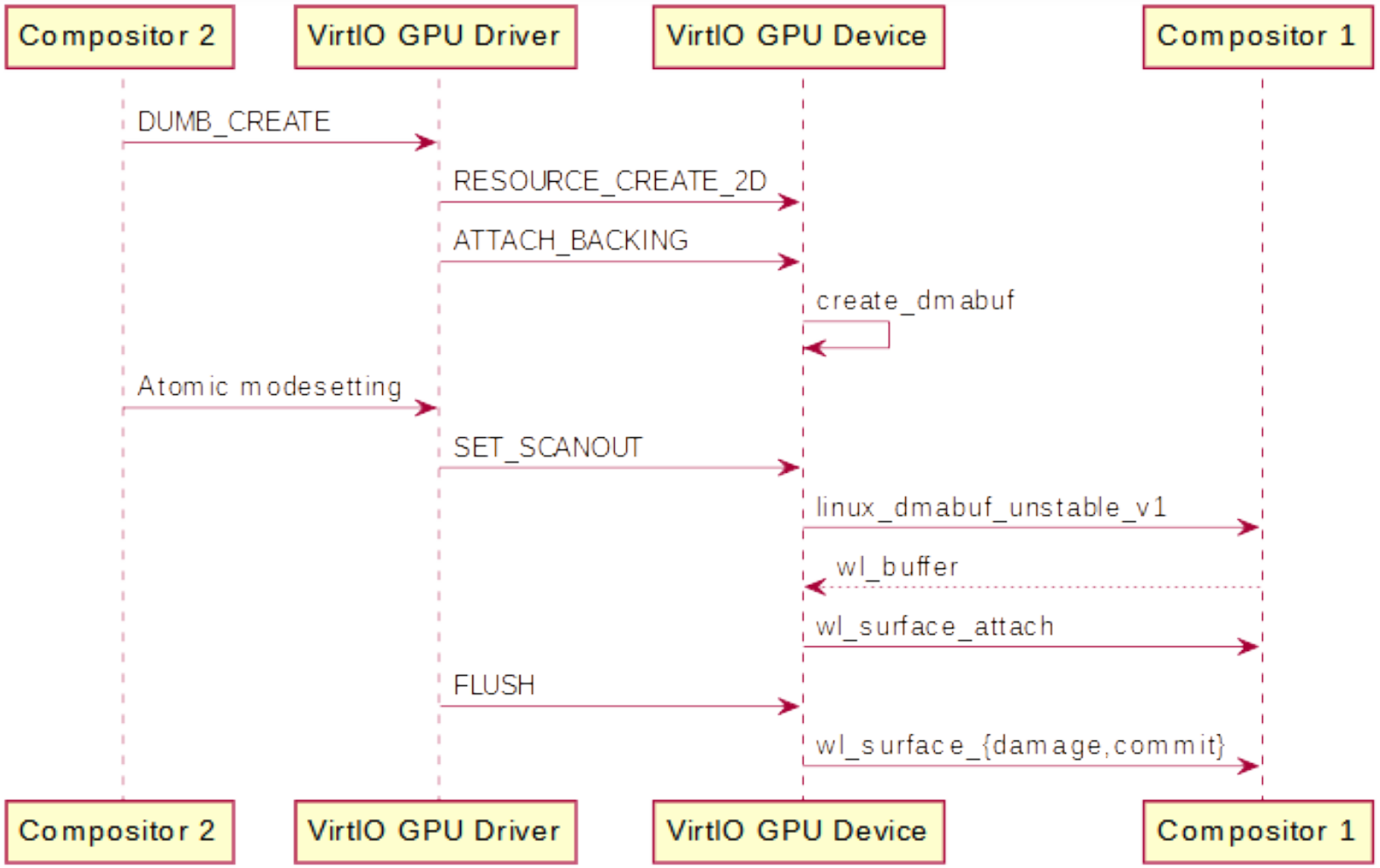
- Performance efficiency: avoid time spent on memcpy in CPU or GPU
- Memory efficiency: avoid resource allocated in device VM and driver VM at the same time

Status of zero-copy support available within community

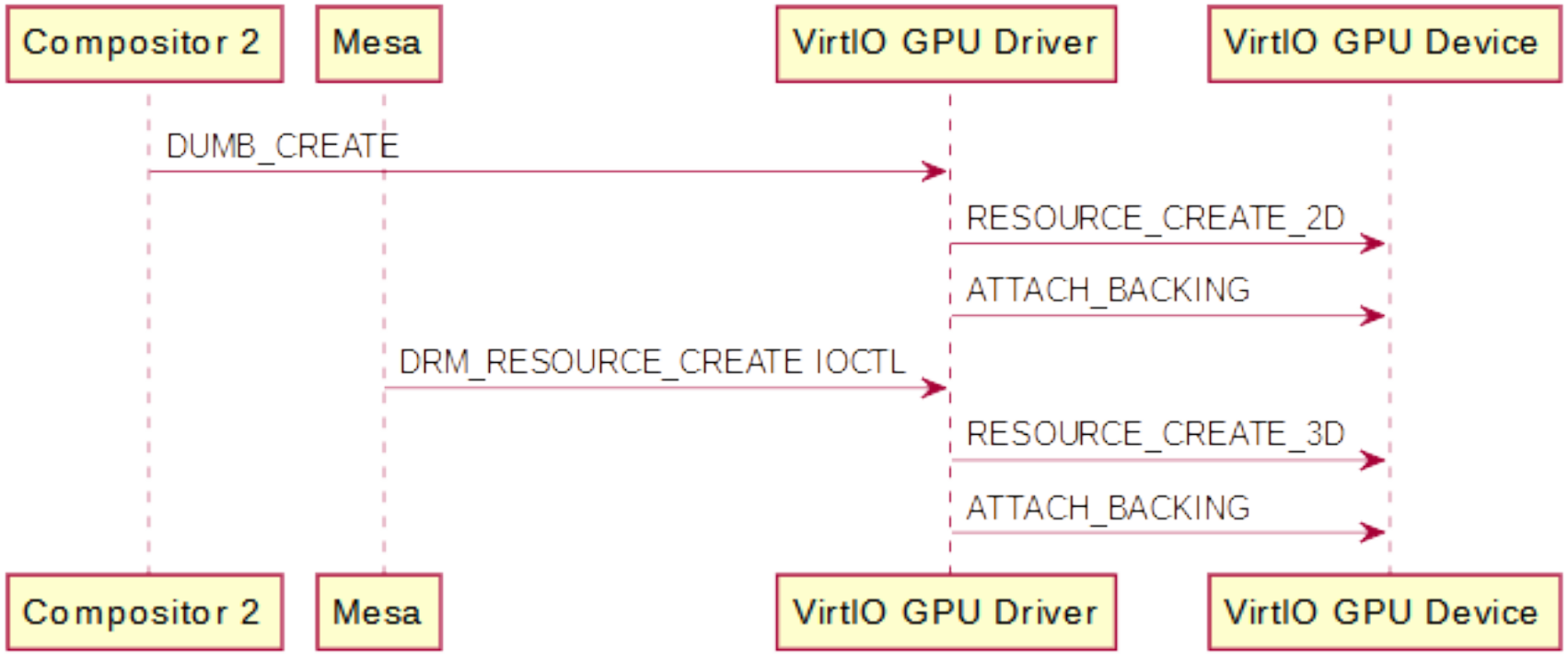
OPENSYNERGY

- Does VirtIO GPU supports zero-copy in QEMU, crosvm, etc.?
  - Crosvm uses host allocations, is it requirement for something?
- Does virglrenderer supports zero-copy of OpenGL buffers (vertex buffer, element buffer, etc.)?

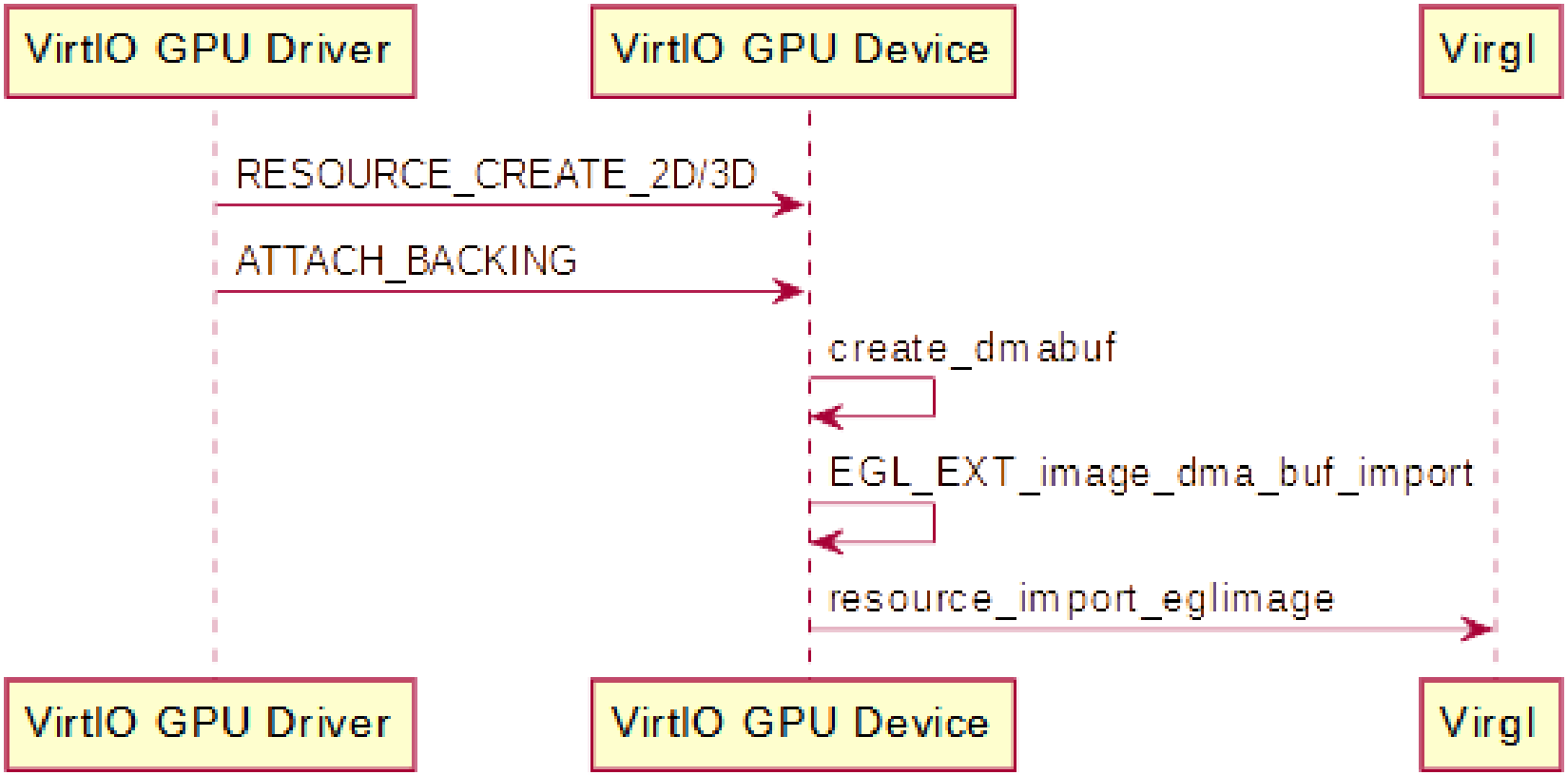
How zero copy works in 2D case



Resource create in 3D case



Zero-copy in 3D case



Challenges

OPENSYNERGY

- EXT_image_dma_buf_import fails with EGL_BAD_ACCESS
  - Background:
    - Documentation: If <target> is EGL_LINUX_DMA_BUF_EXT and one or more of the values specified for a plane's pitch or offset isn't supported by EGL, EGL_BAD_ACCESS is generated.
    - Looking for pitch (stride) requirement for R-Car GPU.
  - Issue:
    - Once they are found, how to put stride specific calculation in Mesa in device guest?
- Avoid bounce-buffering
  - Background: GPU/display supports only 32-bit addresses
  - Issue: ensure device guest (mesa) allocates memory from lower addresses
- Virglrenderer tests fail on R-Car:

```
../git/tests/test_virgl_init.c:78:F:init:virgl_init_egl:0: Assertion 'ret == 0' failed: ret == -1, 0 == 0
../git/tests/test_virgl_init.c:89:F:init:virgl_init_egl_create_ctx:0: Assertion 'ret == 0' failed: ret == -1, 0 == 0
../git/tests/test_virgl_init.c:104:F:init:virgl_init_egl_create_ctx_0:0: Assertion 'ret == 0' failed: ret == -1, 0 == 0
```