

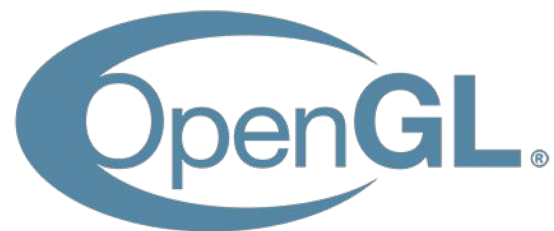
The many faces of virtio-gpu

Sergio López Pascual
Senior Principal Software Engineer
Automotive Team

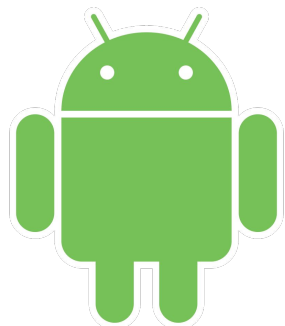
Virtio-gpu, who are you?

This is getting confusing

VirGL



Venus



gfxstream



DRM native context



VirGL2

Virtio-gpu Contexts (I)

Though I prefer calling them “personalities”

- ▶ linux/include/uapi/linux/virtio_gpu.h:

```
#define VIRTIO_GPU_CAPSET_VIRGL 1
#define VIRTIO_GPU_CAPSET_VIRGL2 2
/* 3 is reserved for gfxstream */
#define VIRTIO_GPU_CAPSET_VENUS 4
```

Virtio-gpu Contexts (II)

Though I prefer calling them “personalities”

- ▶ rutabaga_gfx/src/rutabaga_utils.rs:

```
/// Rutabaga capsets.  
pub const RUTABAGA_CAPSET_VIRGL: u32 = 1;  
pub const RUTABAGA_CAPSET_VIRGL2: u32 = 2;  
pub const RUTABAGA_CAPSET_GFXSTREAM_VULKAN: u32 = 3;  
pub const RUTABAGA_CAPSET_VENUS: u32 = 4;  
pub const RUTABAGA_CAPSET_CROSS_DOMAIN: u32 = 5;  
pub const RUTABAGA_CAPSET_DRM: u32 = 6;  
pub const RUTABAGA_CAPSET_GFXSTREAM_MAGMA: u32 = 7;  
pub const RUTABAGA_CAPSET_GFXSTREAM_GLES: u32 = 8;  
pub const RUTABAGA_CAPSET_GFXSTREAM_COMPOSER: u32 = 9;
```

Virtio-gpu Contexts (II)

Though I prefer calling them “personalities”

- ▶ rutabaga_gfx/src/rutabaga_utils.rs:

```
/// Rutabaga capsets.  
pub const RUTABAGA_CAPSET_VIRGL: u32 = 1;  
pub const RUTABAGA_CAPSET_VIRGL2: u32 = 2;  
pub const RUTABAGA_CAPSET_GFXSTREAM_VULKAN: u32 = 3;  
pub const RUTABAGA_CAPSET_VENUS: u32 = 4;  
pub const RUTABAGA_CAPSET_CROSS_DOMAIN: u32 = 5;  
pub const RUTABAGA_CAPSET_DRM: u32 = 6;  
pub const RUTABAGA_CAPSET_GFXSTREAM_MAGMA: u32 = 7;  
pub const RUTABAGA_CAPSET_GFXSTREAM_GLES: u32 = 8;  
pub const RUTABAGA_CAPSET_GFXSTREAM_COMPOSER: u32 = 9;
```

How virtio-gpu became The Faceless Device

A three act story

How virtio-gpu became The Faceless Device

A three act story

▶ EXECBUFFER

How virtio-gpu became The Faceless Device

A three act story

▶ **EXECBUFFER**

- A unidirectional opaque transport.

How virtio-gpu became The Faceless Device

A three act story

- ▶ **EXECBUFFER**
 - A unidirectional opaque transport.
- ▶ **F_RESOURCE_BLOB**

How virtio-gpu became The Faceless Device

A three act story

- ▶ **EXECBUFFER**
 - A unidirectional opaque transport.
- ▶ **F_RESOURCE_BLOB**
 - Rings, queues, mapped FDs...

How virtio-gpu became The Faceless Device

A three act story

- ▶ **EXECBUFFER**
 - A ~~unidirectional~~ bidirectional opaque transport.
- ▶ **F_RESOURCE_BLOB**
 - Rings, queues, mapped FDs...

How virtio-gpu became The Faceless Device

A three act story

- ▶ **EXECBUFFER**
 - A ~~unidirectional~~ bidirectional opaque transport.
- ▶ **F_RESOURCE_BLOB**
 - Rings, queues, mapped FDs...
- ▶ **Fences**

How virtio-gpu became The Faceless Device

A three act story

- ▶ **EXECBUFFER**
 - A ~~unidirectional~~ bidirectional opaque transport.
- ▶ **F_RESOURCE_BLOB**
 - Rings, queues, mapped FDs...
- ▶ **Fences**
 - Pollable objects.

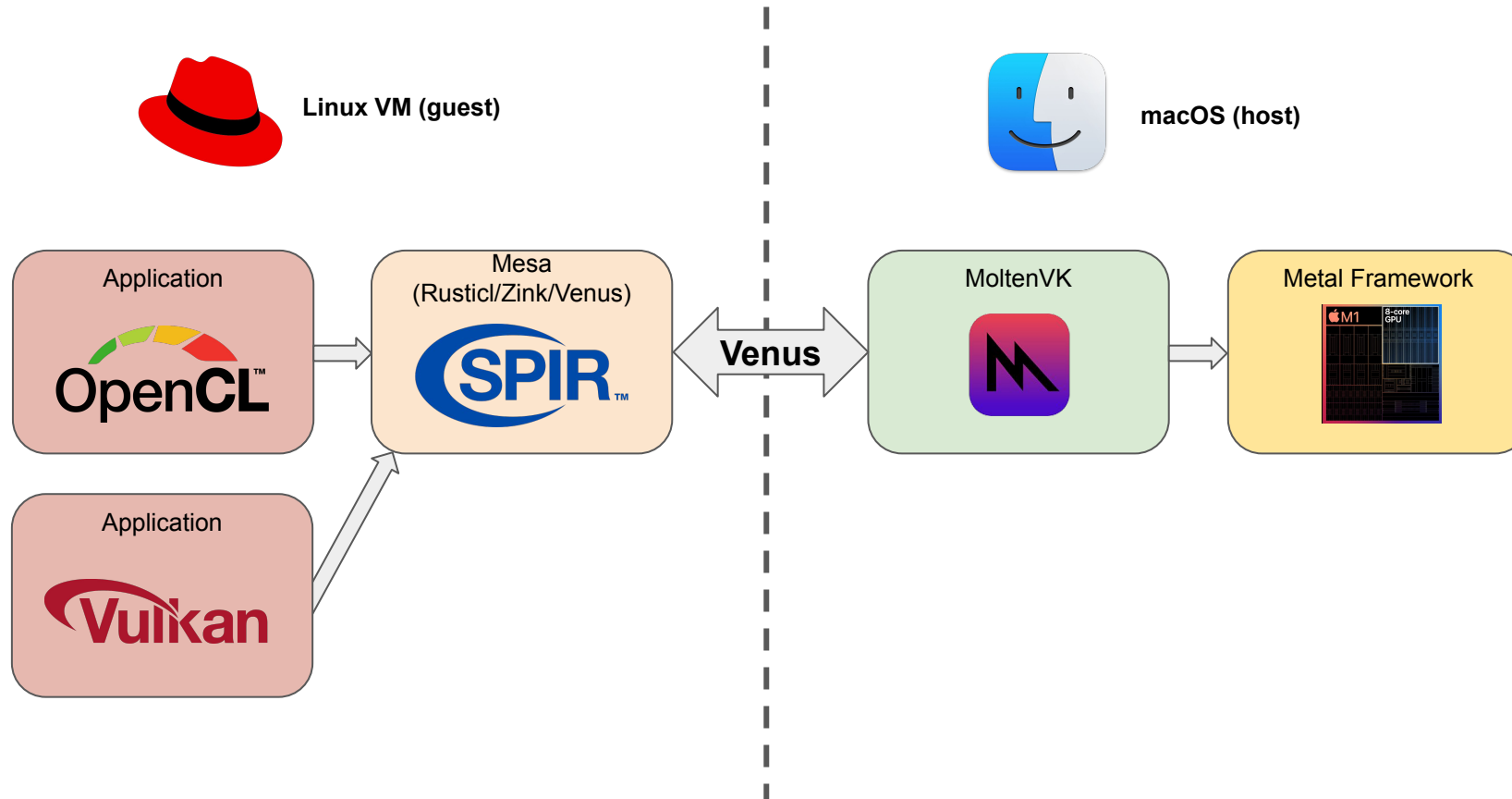
Venus

Serializing Vulkan

- ▶ Serializes Vulkan commands.
 - Easier than GL, since it operates at a less abstract level.
- ▶ Makes heavy use of BLOBs.
- ▶ Use cases:
 - Project Borealis: games on Chromebooks
 - Podman Desktop: run GPU-accelerated containers on macOS for LLM inference.

Podman with Venus on macOS

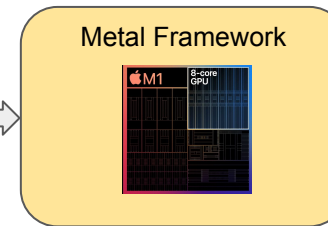
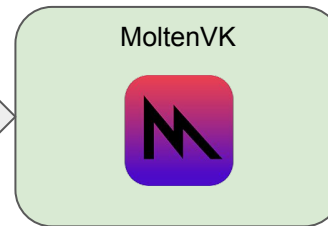
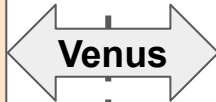
From Vulkan to Metal



Linux VM (guest)

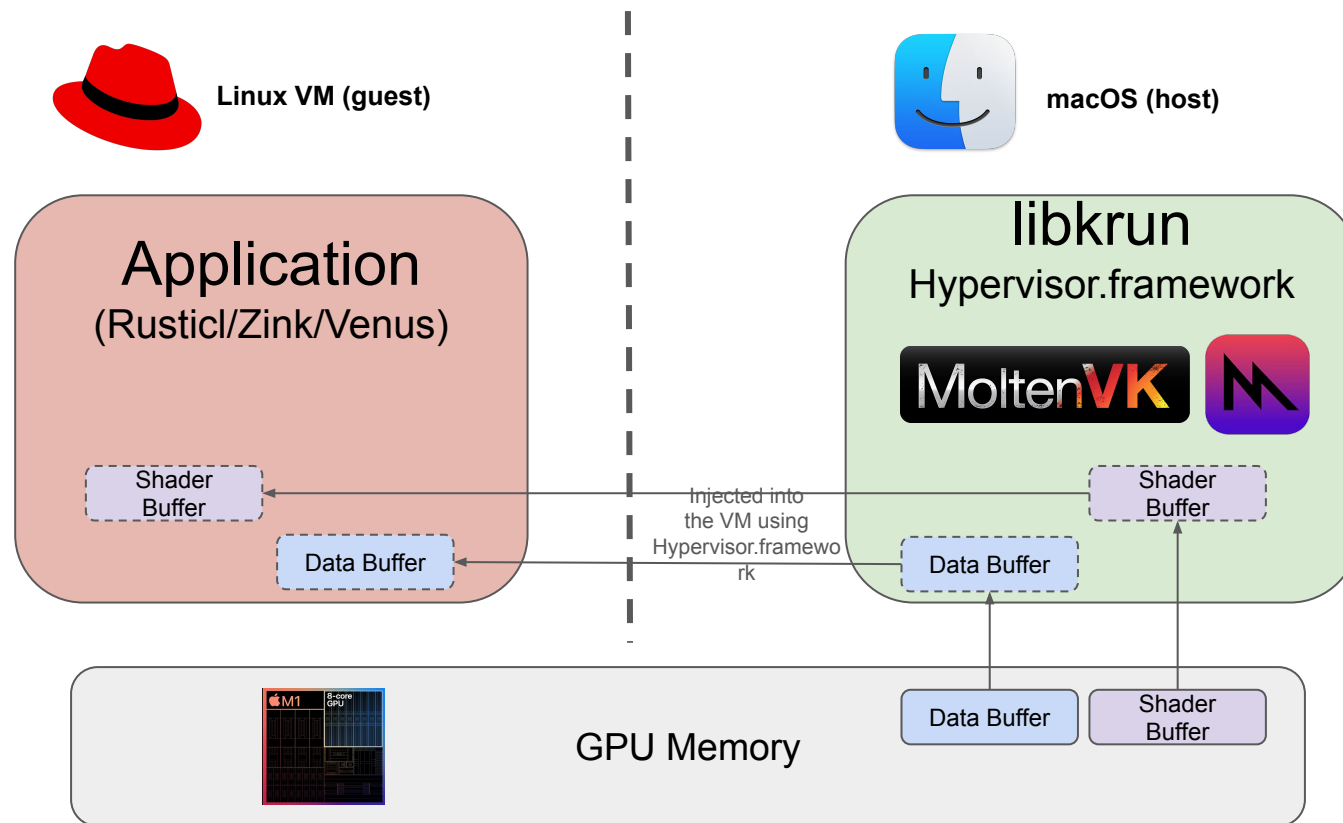


macOS (host)



Podman with Venus on macOS

From Vulkan to Metal



Cross-domain

A compositor's game

- ▶ Drawing in the framebuffer isn't always the most desirable approach.
 - Worse UI
 - Worse Performance
- ▶ But we need a way for apps running in the guest to talk to the compositor in the host.
- ▶ Offers primitives to support a nested Wayland compositor
 - Replaces virtio-wl

DRM native context

Let's go lower!

- ▶ Serializes DRM ioctls!
 - Requires specific support in the Mesa driver.
 - Requires a dedicated controller in virglrenderer for each GPU driver.
 - msm (Freedreno), amdgpu, intel (WIP), asahi (WIP)
- ▶ Use cases
 - Borealis?
 - Commercial games on Asahi Linux

Asahi Linux's problem with Commercial Games

I think we just need one more layer

- ▶ Apple Silicon is ARM64, while most commercial games only provide x86_64 binaries.
 - Use a (fast!) userspace x86_64 emulator.
- ▶ Asahi Linux is a 16K page distribution.
 - Run a 4K kernel in a VM.

Asahi Linux's problem with Commercial Games

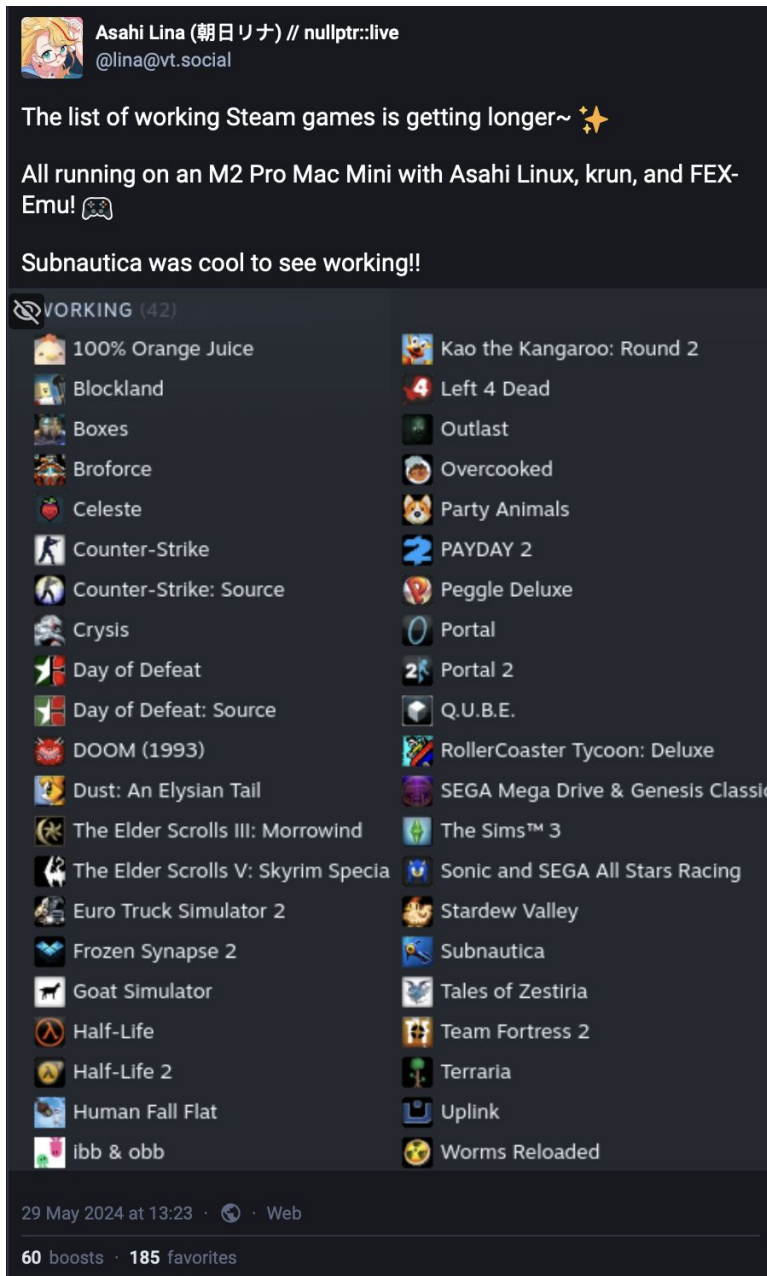
I think we just need one more layer

- ▶ Apple Silicon is ARM64, while most commercial games only provide x86_64 binaries.
 - Use a (fast!) userspace x86_64 emulator.
- ▶ Asahi Linux is a 16K page distribution.
 - Run a 4K kernel in a VM.
- ▶ VMs doesn't have direct access to graphics.
 - Use DRM native context.

DRM native-context + krun + fex-emu on Asahi Linux

Using the marvelous reverse-engineered Asahi GPU driver

DEMO



- ▶ Asahi Lina’s Mastodon post:
 - <https://vt.social/@lina/112524118075585601>
- ▶ Asahi Lina’s Live Stream trying out games:
 - https://youtube.com/live/JT9a_MrFV18

Questions?

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



twitter.com/RedHat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



twitter.com/RedHat

VirGL and VirGL2

The ones who started it all

- ▶ Encodes GL operations in the guest, decodes them in the host.
 - In the guest, it's implemented as a Gallium driver in Mesa.
 - In the host, it's supported by the virglrenderer library linked against the virtio-gpu device implementation.
 - Written by hand
- ▶ Use case: run GPU-accelerated VMs pretty much everywhere.

GfxStream

Android's preferred solution

- ▶ Auto-generates GLES and Vulkan calls.
- ▶ Was developed as part of Android Studio.
- ▶ AFAWK, AOSP is going to deprecate VirGL (no longer enabled by default in the builds) in favor of GfxStream.
- ▶ It's upstream status beyond Android is so-so.
- ▶ Multiple flavors:
 - GLES, VK, MAGMA?, COMPOSITOR?