Running Android on the Mainline Graphics Stack

Robert Foss
Software Engineer

robert.foss@collabora.com
Agenda

- Android Graphics Stack
- Open Source Graphics Stack
- Current Status
- Why does this matter?
Android Graphics Stack
Android Graphics Stack

- Kernel
- Vendor driver
- SurfaceFlinger
  - HWC2
- Apps
  - HWC2
Android Graphics Stack

The Really Good Stuff™
Android Graphics Stack

The Really Good Stuff™
- No really, this is the whole point!
Android Graphics Stack

Interface between applications and hardware
Android Graphics Stack

Status Bar

- Kernel
- Vendor driver
- SurfaceFlinger
- HWC2
- Apps
Android Graphics Stack

Status Bar

Apps
SurfaceFlinger
HWC2
HWC2
Vendor driver
Kernel
Android Graphics Stack

- Kernel
- Vendor driver
- HWC2
- SurfaceFlinger
- Apps

Status Bar

Navigation Bar

Background
Android Graphics Stack

- Kernel
- Vendor driver
- HWC2
- SurfaceFlinger
- Apps

The non-kernel part of the graphics driver
Android Graphics Stack

The non-kernel part of the graphics driver
- OpenGL, Vulkan, memory allocator, etc.
Android Graphics Stack

The Linux Kernel

- Kernel
- Vendor driver
- HWC2
- SurfaceFlinger
- Apps
Android Graphics Stack

What does the HWC2 API do?

Diagram:
- Kernel
- Vendor driver
- HWC2
- SurfaceFlinger
- Apps
Android Graphics Stack

What does the HWC2 API do?

- API used between SurfaceFlinger and hardware
Android Graphics Stack

What does the HWC2 API do?

- API used between SurfaceFlinger and hardware
- Compose layers to the screen
Android Graphics Stack

What does the HWC2 API do?

- API used between SurfaceFlinger and hardware
- Compose layers to the screen
- Abstract graphical objects
Android Graphics Stack

What does the HWC2 API do?

- API used between SurfaceFlinger and hardware
- Compose layers to the screen
- Abstract graphical objects
- Offload work from GPU to compositor hardware
Mainline Graphics Stack
Mainline Graphics Stack

Where does the OSS stack fit in?
Mainline Graphics Stack

Where does the OSS stack fit in?

- Kernel
- Vendor driver
  - HWC2
  - HWC2
- SurfaceFlinger
- Apps
  

Proprietary
Mainline Graphics Stack

Where does the OSS stack fit in?
Mainline Graphics Stack

Where does the OSS stack fit in?

Kernel

Vendor driver

SurfaceFlinger

Apps

HWC2

HWC2

Driver

Kernel

HWC2

drm_hwc
Mainline Graphics Stack

Where does the OSS stack fit in?
Mainline Graphics Stack
Where does the OSS stack fit in?

[Diagram showing the stack with layers: OSS stack fits above theVendor driver, HWC2, SurfaceFlinger, Apps, and Kernel layers.]
Mainline Graphics Stack

What does drm_hwcomposer do?
Mainline Graphics Stack

What does drm_hwcomposer do?

Get Layers Through HWC2 API
Mainline Graphics Stack

What does drm_hwcomposer do?

Get Layers Through HWC2 API

Optimize Layers for Display
Mainline Graphics Stack

What does drm_hwcomposer do?

- Get Layers Through HWC2 API
- Optimize Layers for Display
- Output Layers To DRM Planes
Current status

Why now?
Current status

Why now?

- Kernel Buffer Synchronization support
  - Inspired by Android kernels, now in mainline
  - Some GPU drivers now support this
Current status

Why now?

• Kernel Buffer Synchronization support
  – Inspired by Android kernels, now in mainline
  – Some GPU drivers now support this

• Atomic Display Framework API
  – Most drivers implement this
Current status

Tested platforms
Current status

Tested platforms

- iMX6
  - GPU: Vivante GC3000
Current status
Tested platforms

- Dragonboard 410c
  - GPU: Adreno 306
Current status

Under Development

- Dragonboard 820c
  - GPU: Adreno 560
Current status
Under Development

- HiKey 960
  - GPU: Mali G71
Current status
Project Hosting
Current status

Previously hosted within ChromiumOS
Current status

Project Hosting

- Previously hosted within ChromiumOS
- Now hosted on Freedesktop.org
Why?
Why?
Long term support
Why?

Long term support

• Deliver products with >10 year lifespan
Why?

Long term support

- Deliver products with >10 year lifespan
- Support can be provided by anyone
Why?
Long term support

- Deliver products with >10 year lifespan
- Support can be provided by anyone
- Hardware support even if the vendor disappears
Why?
Push industry towards Open Source
Why?
Push industry towards Open Source

- Lower driver development costs
Why?
Push industry towards Open Source

- Lower driver development costs
- Increase driver quality
Why?

Push industry towards Open Source

- Lower driver development costs
- Increase driver quality
- Lower cost for device manufacturers
Thank you!