XDC 2018

26-28 Sept. - A Coruña

Android-x86 status update

from lead developer Chih-Wei Huang

Graphics stack evolution

presented by Mauro Rossi (maurossi)

android-x86.org

Android-x86

Agenda - Part I

Android-x86 status update

- Android-x86 project origins
- Android-x86 as seen in analytics
- Porting proceedings
- How to contribute to add new supported Hardware
- Stable and development release status
- Future Plans

Android-x86: what is it?

It is an Android Board Support Package*

For generic x86 platforms, including Desktop, Laptops

Effective Desktop OS alternative, with “Mouse Click as a Touch”

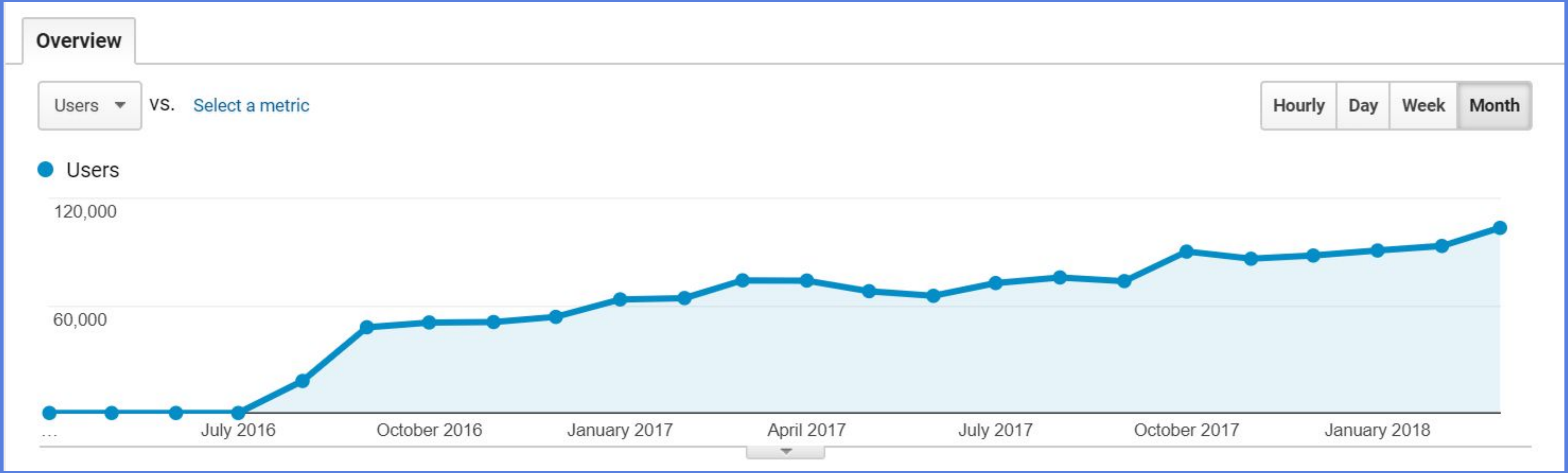
Completely Open Source and tracked as Linux OS in DistroWatch

Started in 2009 by Chih-Wei Huang and Yi Sun (beyounn)

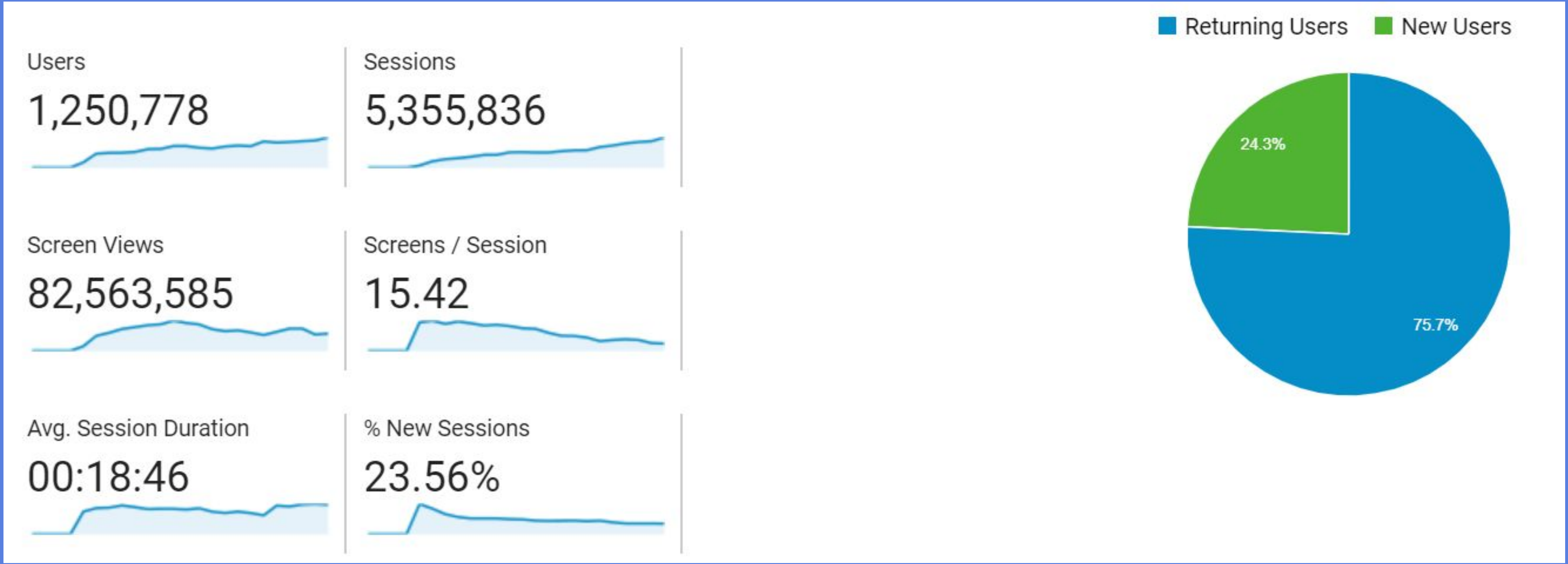
*https://en.wikipedia.org/wiki/Board_support_package

Android-x86 in analytics (1/2)

3 milion downloads reached and users still growing



Android-x86 in analytics (2/2)



Users
1,250,778

Sessions
5,355,836

Screen Views
82,563,585

Screens / Session
15.42

Avg. Session Duration
00:18:46

% New Sessions
23.56%

Porting proceedings

- Prepare the codebase manifest xml
  - AOSP git projects
  - Modified AOSP git projects
    - device/generic/x86
    - device/generic/common (BoardConfig.mk, init.sh)
    - frameworks/base (SurfaceFlinger)
    - hardware/libhardware{,_*}
    - system/core
    - kernel
  - Android-x86 git projects
    - bootable/newinstaller
    - external/drm_gralloc
    - external/libdrm (update versions)
    - external/mesa (updated versions)
    - external/llvm (updated versions)

Chih-Wei Wang

Our development cycle

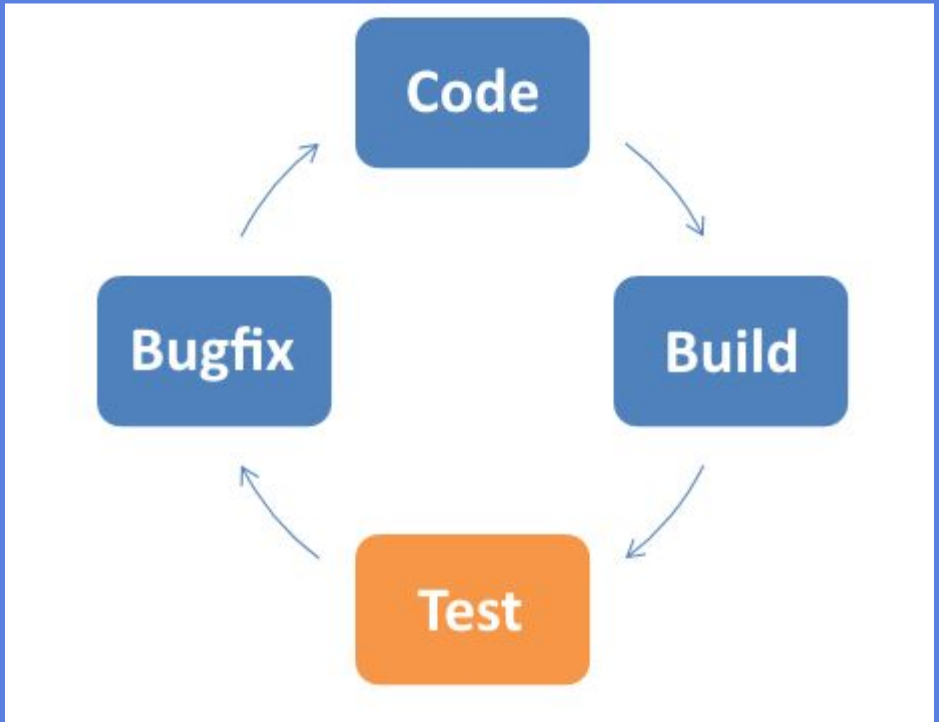
“Agile by necessity”

Apply android-x86 patches (git rebase) to AOSP

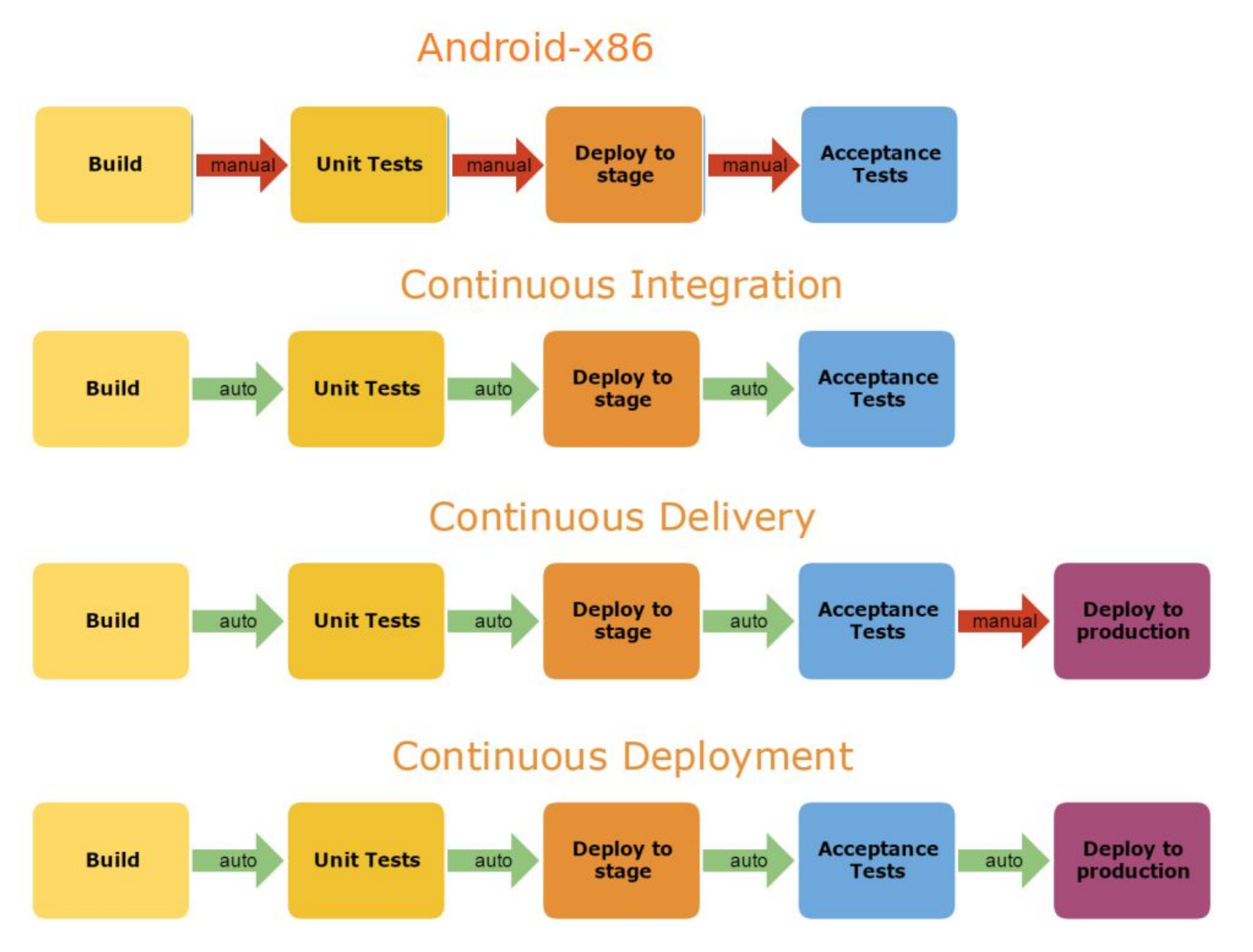
Try to build and fix compiling errors

Try to boot, identify and fix problems

- Display is one the most critical aspects



Android-x86 and CI/CD



Android-x86

Build → manual → Unit Tests → manual → Deploy to stage → manual → Acceptance Tests

Continuous Integration

Build → auto → Unit Tests → auto → Deploy to stage → auto → Acceptance Tests

Continuous Delivery

Build → auto → Unit Tests → auto → Deploy to stage → auto → Acceptance Tests → manual → Deploy to production

Continuous Deployment

Build → auto → Unit Tests → auto → Deploy to stage → auto → Acceptance Tests → auto → Deploy to production

Mostly manual/visual testing
Android CTS for EGL/GLES

We need effortless CI tools
for Build, Boot and GUI tests

Most probably never,
let’s start from CI

Overview of /etc/init.sh

Key part of android-x86 runtime drivers configuration
Based on sh script, with specialized init HAL functions

Adding new hardware support requires:

- linux kernel support (pre-requisite)
- change init_hal_* - e.g. init_hal_bluetooth() for Bluetooth

Patches from community are welcome to extend HW support

nougat-x86 features

- Android 7.1.2_r36 based
- 32-bit and 64-bit systems support
- Kernel 4.9, KMS enabled
- 3D Hardware acceleration (Intel/AMD/Nvidia graphics chipsets)
- Live CD & hard disk installation, Auto Update feature
- Hybrid iso image and EFI image
- Mouse, Wifi, Ethernet, Audio, Bluetooth and Sensors
- Drivers Auto-detect at init
- Virtual Machine installation Qemu, Virtualbox and VMware Player supported

oreo-x86 features

- Android 8.1.0_r46 based
- 32-bit and 64-bit systems support
- Kernel 4.17 wip, KMS enabled
- 3D Hardware acceleration (Intel/AMD/Nvidia graphics chipsets), HWC wip, Vulkan wip
- Live CD & Hard Disk installation, Auto Update feature
- Hybrid iso image and EFI image
- Mouse, Wifi, Ethernet, Audio, Bluetooth and Sensors
- Drivers Auto-detect at init
- Virtual Machine installation supported Qemu, Virtualbox and VMware Player
- Multimedia: Hardware accelerated codecs, HDMI audio
- LineageOS integration

Android-x86 porting status

	nougat-x86	oreo-x86	pie-x86
Display	OK	OK	Starting soon
HW Acceleration	OK	OK*	Starting soon
Audio	OK	OK*	Starting soon
Wifi	OK	OK*	Starting soon
Ethernet	OK	OK	Starting soon
Bluetooth	OK	OK*	Starting soon
Sensors	OK	OK	Starting soon
Camera	OK	OK	Starting soon
Touchscreen	OK	OK	Starting soon
Backlight	OK	OK	Starting soon
Suspend/Resume	OK	OK	Starting soon

*community is asking for additional HW support 3D, Audio, Wifi, Bluetooth

Future plans

Embrace the CI culture, one step at a time

Release HWC + Vulkan enabled iso (atomic & working drivers only)

Improve hardware codec support (NV12, YV12)

Porting Android P/9.0 release to pie-x86

Developers/contributors:

Many thanks to contributors to android-x86 and devs ML:

Chih-Wei Huang - codebase
Jaap Jan Meijer - cyanogenmod branches, LineageOS integration
Jon West - bliss-x86 developer, contributing to pie-x86
Michael Goffioul - multimedia codec acceleration)
Paulo Sergio Tartaglia - nouveau and radeonsi first build and testing
Mauro Rossi - android-x86 mesa lifecycle, kernel RC forward-porting and testing
Emil Velikov - mesa release manager, supporting android-x86 developers Mailing List
Rob Herring - gbm_gralloc, virgl; Professor Chen - virgl
Robert Foss - drm_hwcomposer maintainer
Tapani Pälli - providing insights about android-ia/Intel patches
Tomasz Figa - prime fd fixes in mesa egl/android
Qiang Yu - help with amdgpu Alex Deucher guidance and review of RGBA patches

Join here: https://groups.google.com/forum/#!forum/android-x86-devel

Call for enthusiasts

Ideal platform for HW not supported by new Operating Systems

In many cases Games & Controllers work Out-of-the-Box

Anybody with linux experience and passion can contribute,
for example with improvements to alsa audio configuration,
or to add support for touch screens, x86 tablet sensors, keybuttons

For enthusiasts and supporters, come visit us at:

https://groups.google.com/forum/#!forum/android-x86

End of Part I

For any question: ML and Forum

Agenda - Part II

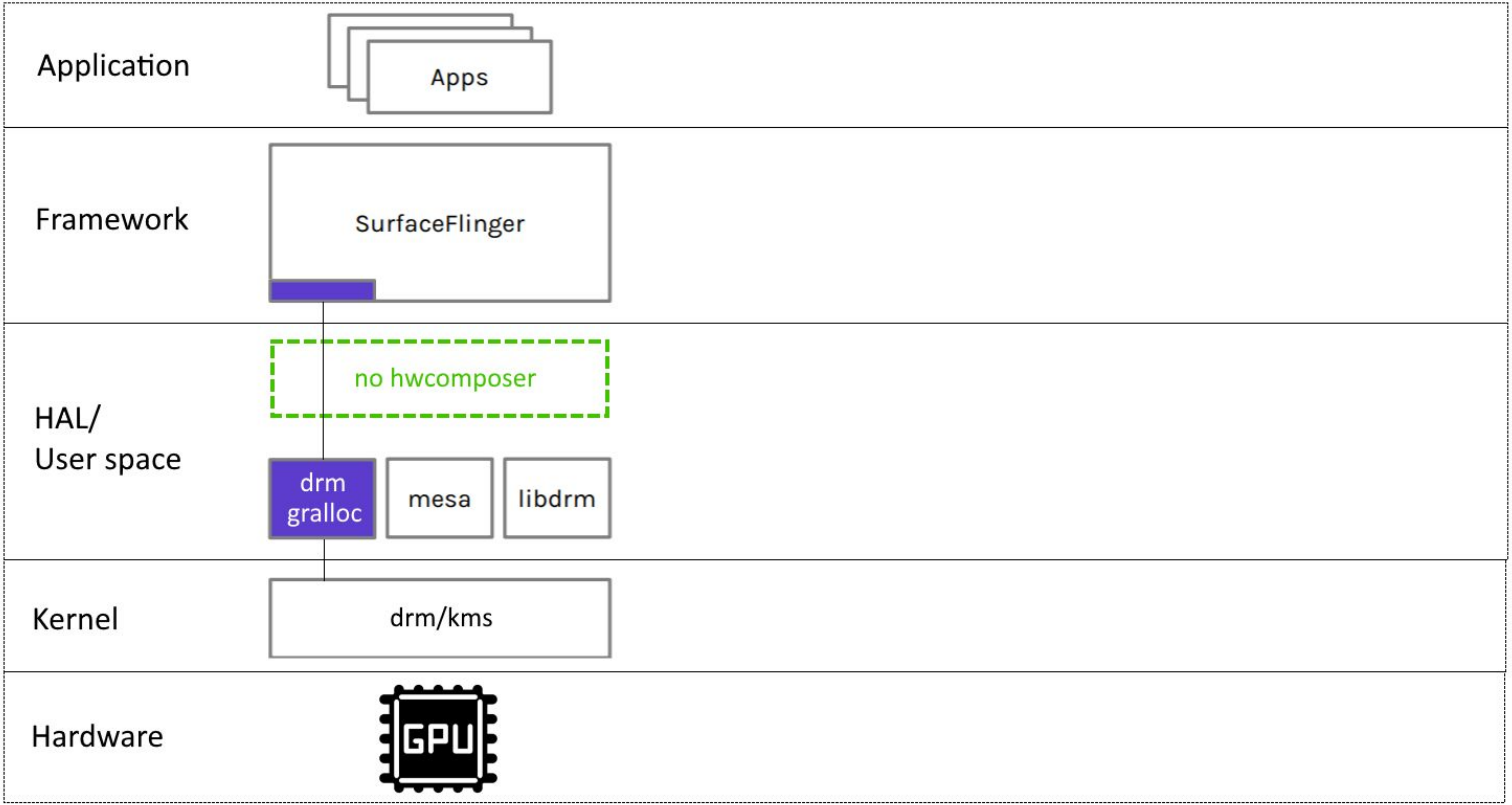
Graphics stack evolution

- Overview of OSS graphic stack in android-x86
- Legacy drm Gralloc: supported HW
- drm HWComposer + gbm Gralloc HAL: test results
- AMDGPU support in android-x86
- Vulkan HAL: anv, radv build integration and testing
- Overview of outstanding work required/Next Steps

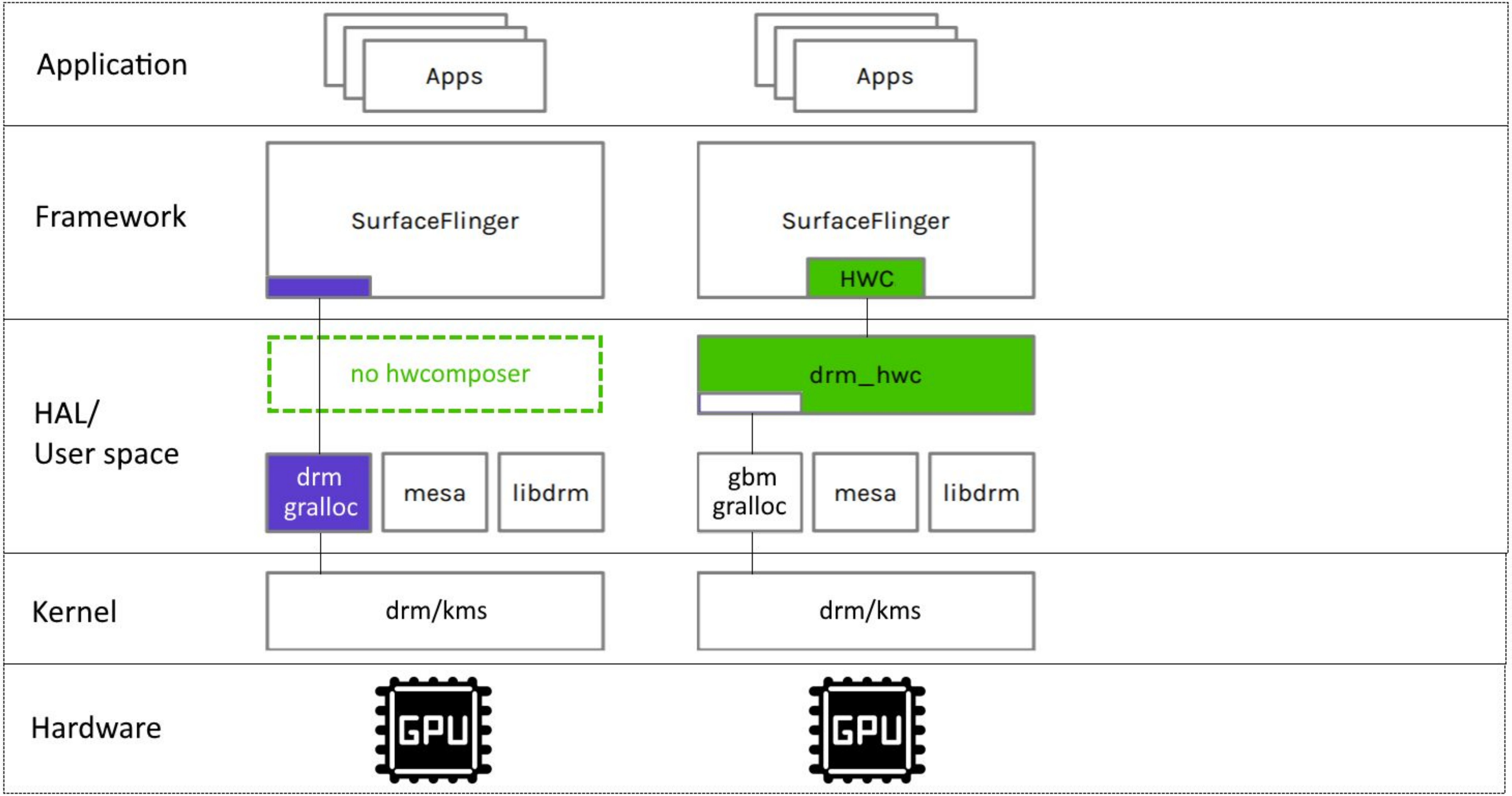
Live Demo with Vulkan in oreo-x86 with HWC1

Mauro Rossi

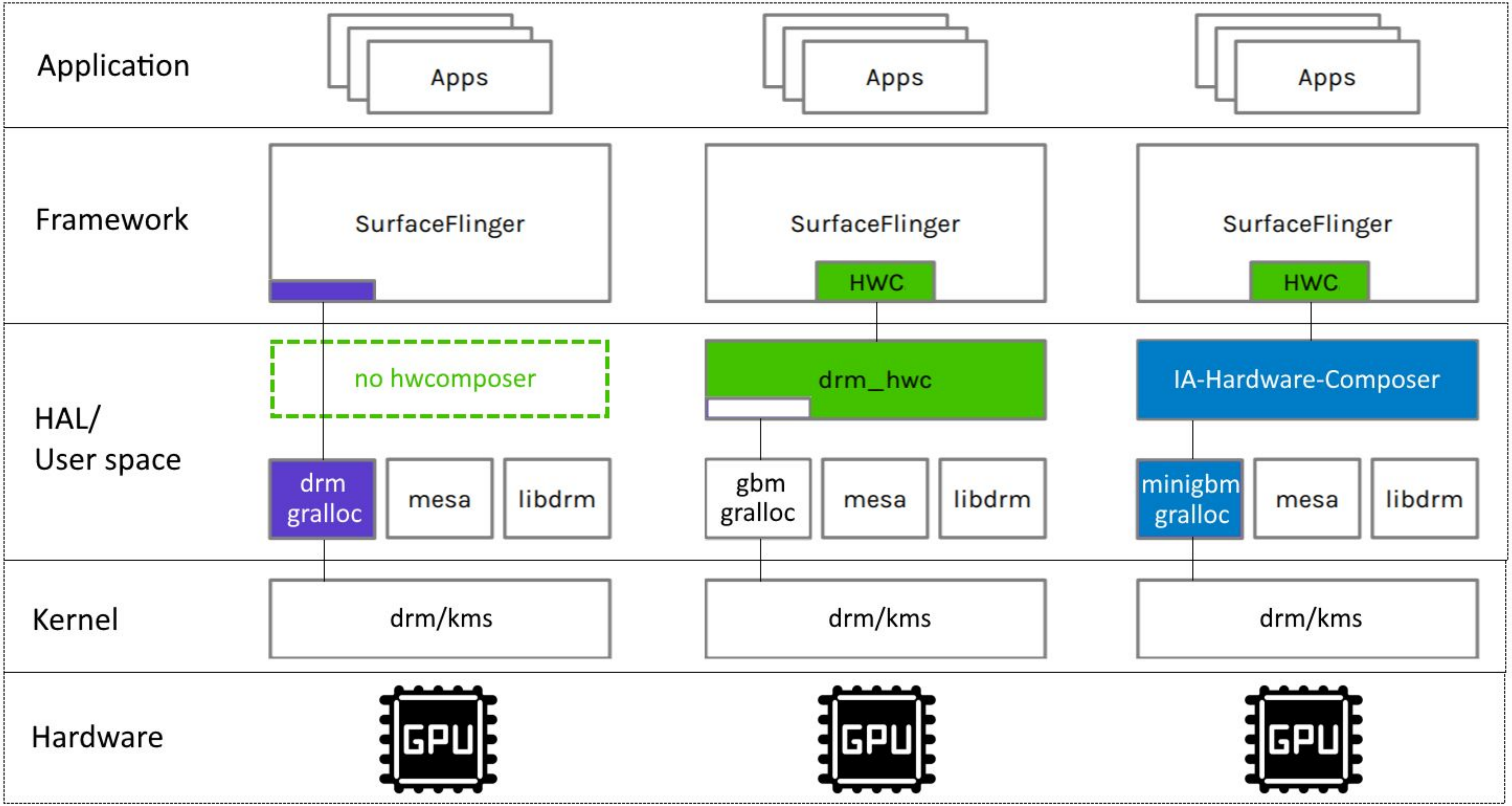
Android-x86 Legacy Graphic stack



Legacy vs HWC reference



OSS Android Graphic stacks



OSS Gralloc HALs capabilities

Gralloc HAL	Gralloc API	GEM / flink names	prime_fd	binderization
drm_gralloc	0	Y	uncomplete untested	N
gbm_gralloc	0	N/A	Y	Y
minigbm	0, 1	N/A	Y	Y

Legacy drm Gralloc HW support

Supported GPU/vGPU drivers [Mesa changes]

- Intel (i915, i965) [null pointer guards to avoid Playstore crashes]
- AMD (r300, r600, radeonsi, amdgpu) [allow non dma texture images]
- Nvidia (nouveau) [mutex locking patches required]
- Vmware (vmwgfx)
- QEMU (virgl)

Not supported

- Intel Atom with PowerVR IGP (no open source driver)

SwitfShader used as fallback for unsupported HW

Legacy drm Gralloc: activities

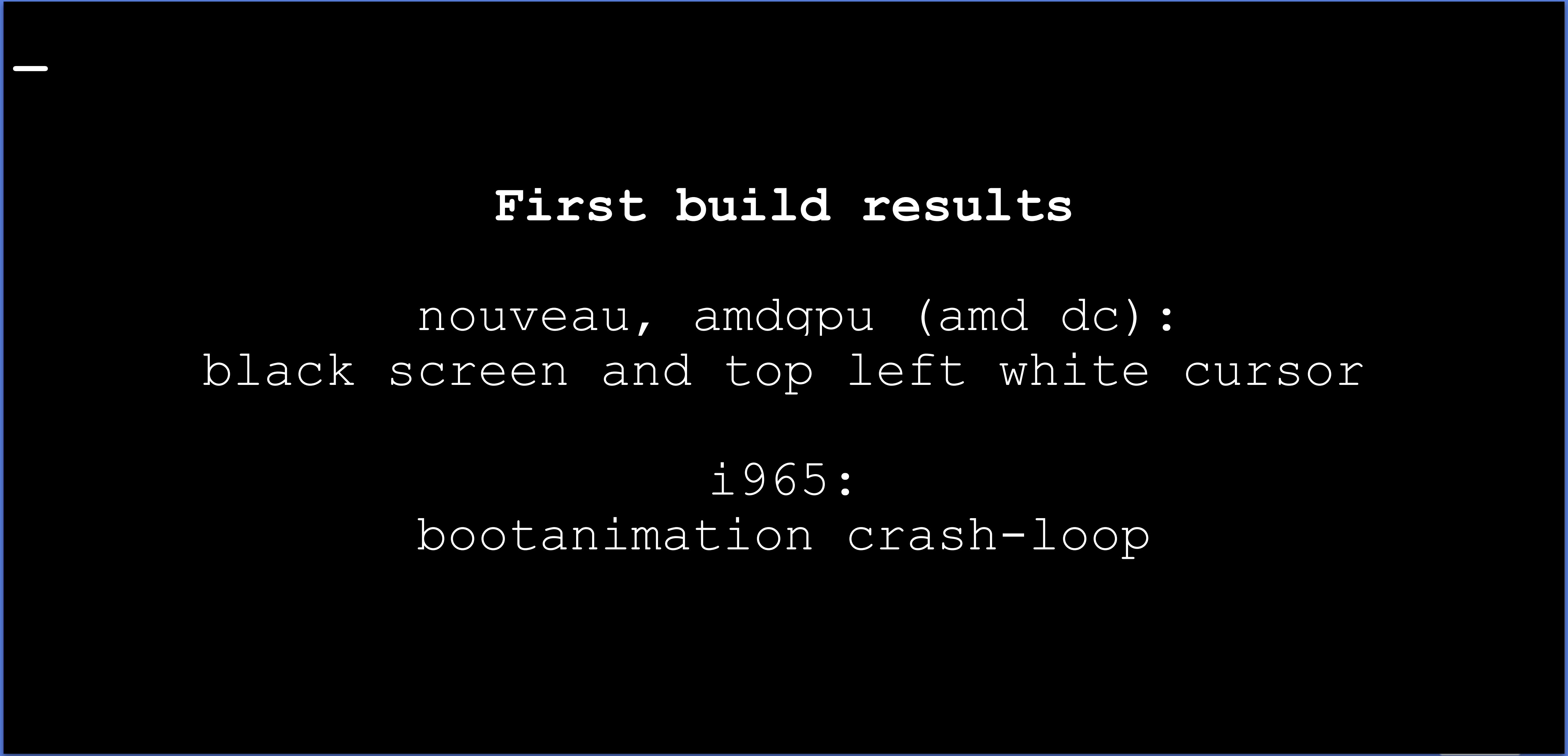
Contributions to drm Gralloc codebase:

- BGRA pixel format for chipsets lacking RGBA format
- Nouveau backend: support and testing for Kepler, Maxwell, Pascal
- Pipe backend: testing with amdgpu (SI, CIK, VI, AI)
- Pipe backend: fix gralloc sw usage flags handling (linear bo)
- Pipe backend: support changes in Mesa 18.2 gallium pipe/winsys
- Mesa 18.2: fixes to egl/android to drm gralloc path

Mesa egl/android periodic testing to verify no regression in drm Gralloc

drm HWComposer + gbm Gralloc path enablers:

- atomic display framework (driver dependent)
- explicit fencing new sync API (Gustavo Padovan - kernel 4.10)
- drm_hwcomposer (Sean Paul, Zach Reizner and Robert Foss)
- gbm gralloc - prime fd and “binderization” (Rob Herring)
- gralloc handle struct harmonized in libdrm (Rob Herring & Robert Foss)
- mesa support for dma-bufs prime_fd in egl/android (Tomasz Figa)



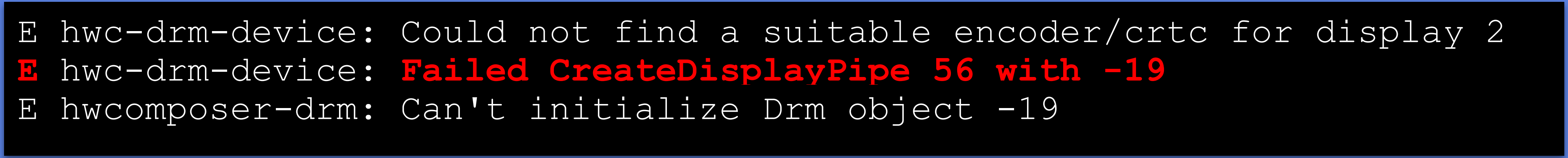
```
_
```

First build results

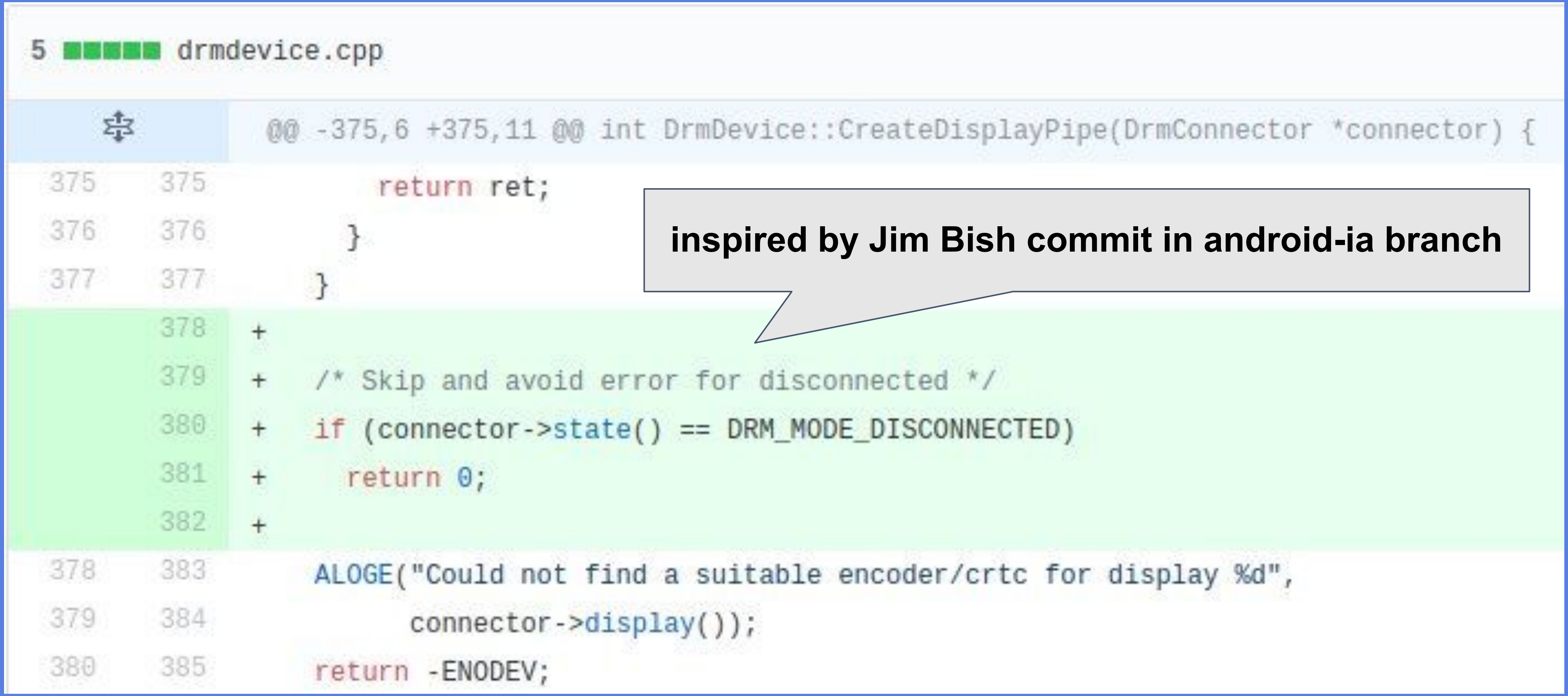
nouveau, amdgpu (amd dc):
black screen and top left white cursor

i965:
bootanimation crash-loop

drm HWComposer HAL



```
E hwc-drm-device: Could not find a suitable encoder/crtc for display 2
E hwc-drm-device: Failed CreateDisplayPipe 56 with -19
E hwcomposer-drm: Can't initialize Drm object -19
```

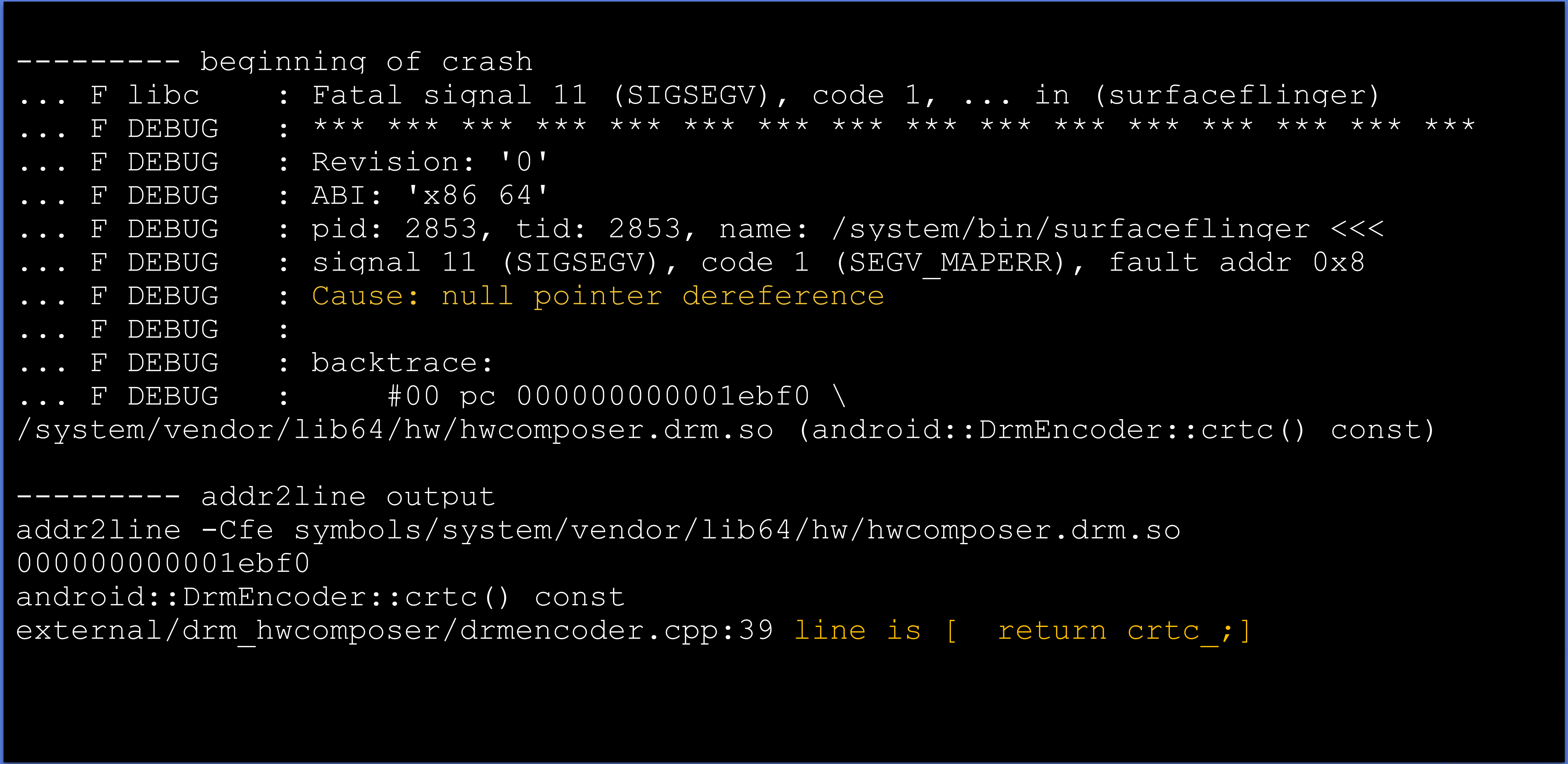


5 🟩🟩🟩🟩🟩 drmdevice.cpp

```
@@ -375,6 +375,11 @@ int DrmDevice::CreateDisplayPipe(DrmConnector *connector) {
          return ret;
        }
      }
+
+     /* Skip and avoid error for disconnected */
+     if (connector->state() == DRM_MODE_DISCONNECTED)
+       return 0;
+
      ALOGE("Could not find a suitable encoder/crtc for display %d",
            connector->display());
      return -ENODEV;
```

inspired by Jim Bish commit in android-ia branch

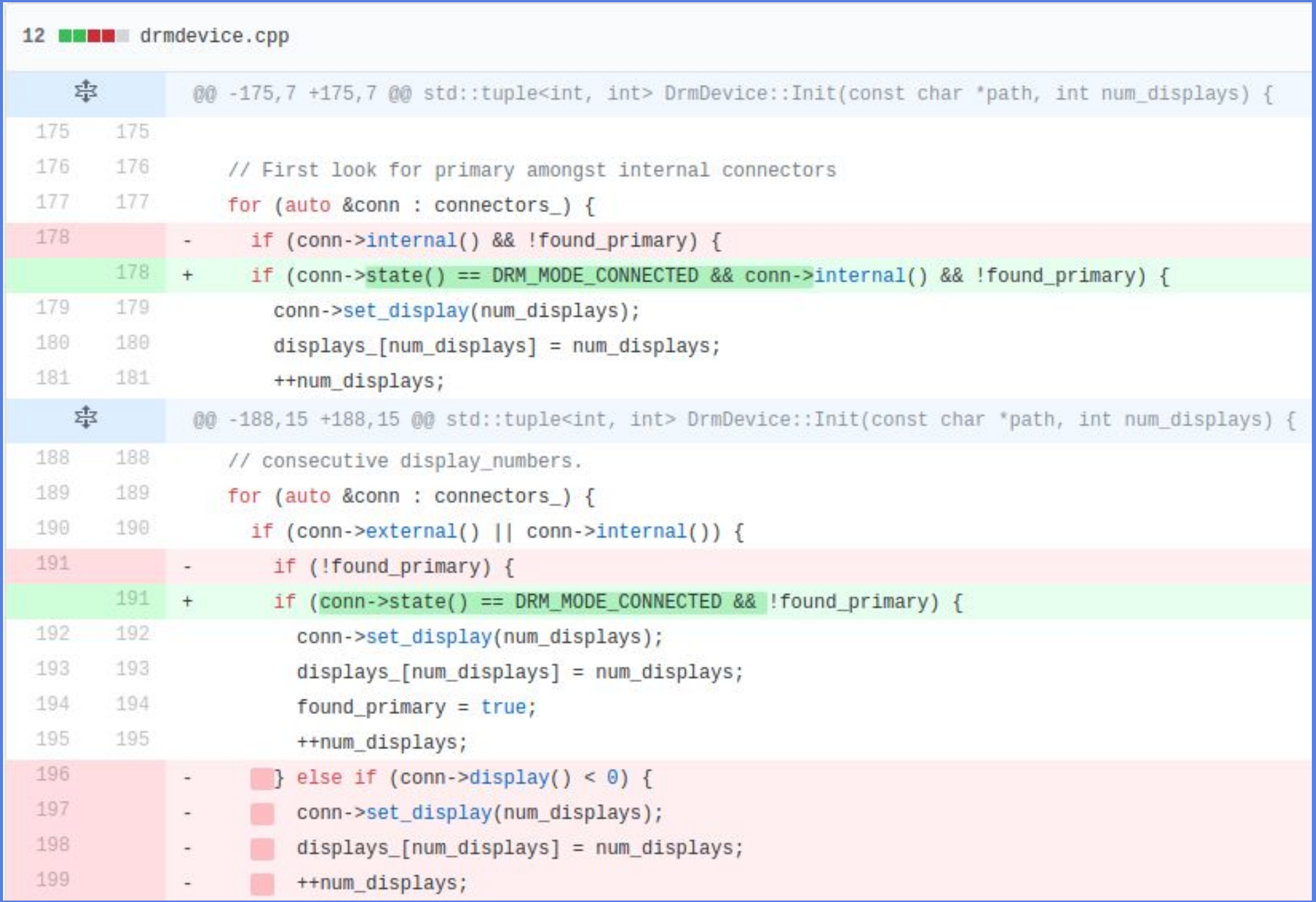
drm HWComposer HAL



```
---------- beginning of crash
... F libc     : Fatal signal 11 (SIGSEGV), code 1, ... in (surfaceflinger)
... F DEBUG    : *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
... F DEBUG    : Revision: '0'
... F DEBUG    : ABI: 'x86_64'
... F DEBUG    : pid: 2853, tid: 2853, name: /system/bin/surfaceflinger <<<
... F DEBUG    : signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x8
... F DEBUG    : Cause: null pointer dereference
... F DEBUG    :
... F DEBUG    : backtrace:
... F DEBUG    :     #00 pc 000000000001ebf0 \
/system/vendor/lib64/hw/hwcomposer.drm.so (android::DrmEncoder::crtc() const)

---------- addr2line output
addr2line -Cfe symbols/system/vendor/lib64/hw/hwcomposer.drm.so
000000000001ebf0
android::DrmEncoder::crtc() const
external/drm_hwcomposer/drmencoder.cpp:39 line is [  return crtc_;]
```

drm HWComposer HAL



12 🟩🟩🟥🟥 drmdevice.cpp

```
@@ -175,7 +175,7 @@ std::tuple<int, int> DrmDevice::Init(const char *path, int num_displays) {

     // First look for primary amongst internal connectors
     for (auto &conn : connectors_) {
-      if (conn->internal() && !found_primary) {
+      if (conn->state() == DRM_MODE_CONNECTED && conn->internal() && !found_primary) {
         conn->set_display(num_displays);
         displays_[num_displays] = num_displays;
         ++num_displays;
@@ -188,15 +188,15 @@ std::tuple<int, int> DrmDevice::Init(const char *path, int num_displays) {
     // consecutive display_numbers.
     for (auto &conn : connectors_) {
       if (conn->external() || conn->internal()) {
-        if (!found_primary) {
+        if (conn->state() == DRM_MODE_CONNECTED && !found_primary) {
           conn->set_display(num_displays);
           displays_[num_displays] = num_displays;
           found_primary = true;
           ++num_displays;
-        } else if (conn->display() < 0) {
-          conn->set_display(num_displays);
-          displays_[num_displays] = num_displays;
-          ++num_displays;
```

Enumerates displays

Regardless of their state

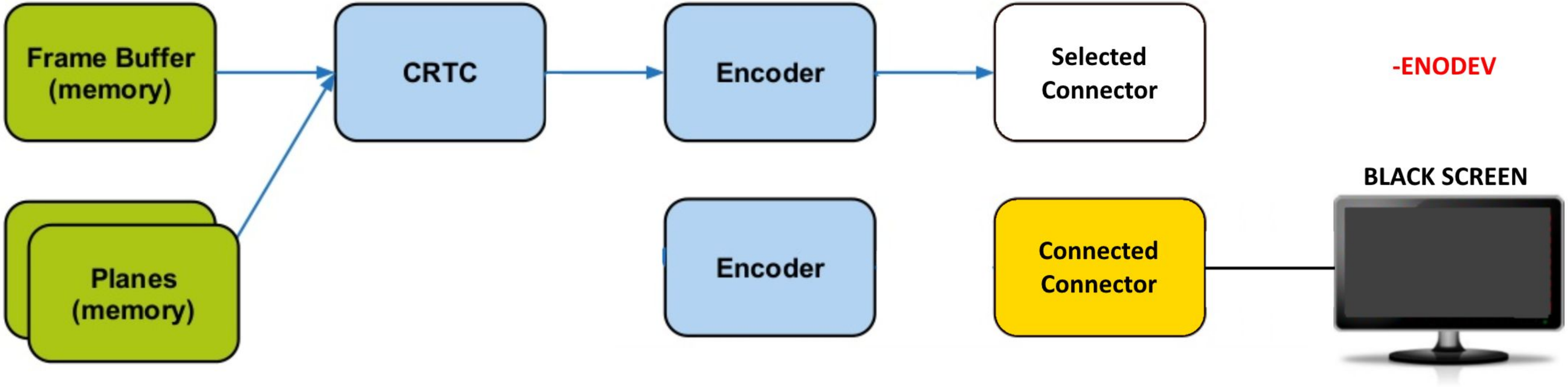
First connector became always the primary

Removed

drm HWComposer HAL

Possible explanation from Android source documentation of HWC

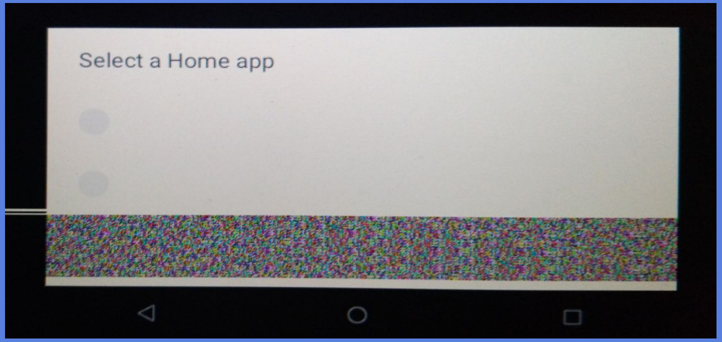
- *It is assumed that there is exactly one primary display; that is, that there is one physical display that will be hotplugged immediately during the initialization*
- *In addition to the primary display, exactly one external display may be hotplugged during normal operation of the device.* (Not yet implemented in drm_hwc)



drm HWComposer: issues

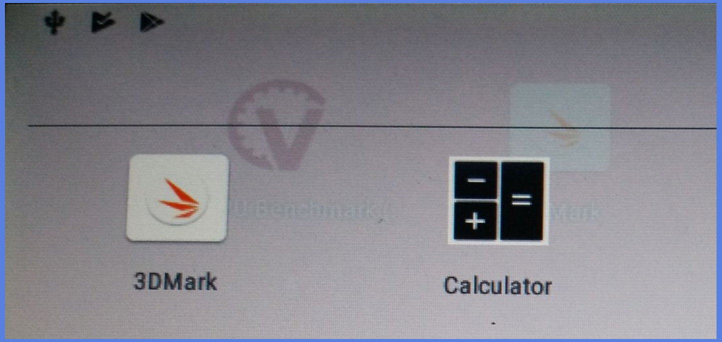
nouveau:

- Google Play crash at Apps search with nouveau
- Cursor tiling artifacts (buffer stride issues)
- Gralloc Usage Read/Write often swap buffer issues (No text)



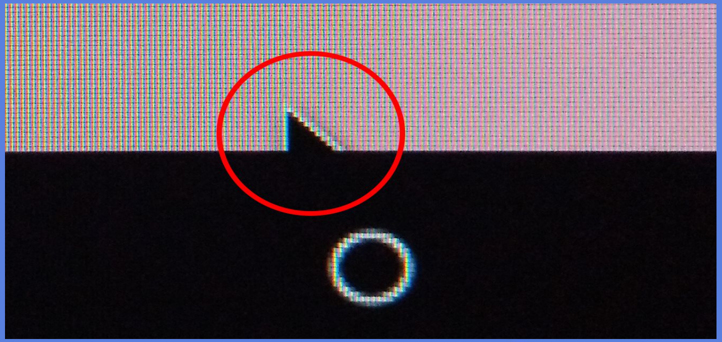
amdgpu:

- Gralloc Usage Read/Write often swap buffer issues (No text)
- Synchronization issues
- GUI opacity issues/Shadowing between layers



intel:

- EGL_swap_buffer_with_damage causes SIGSEGV MAPERR
- Lack of RGBA_8888 in Primary Planes for older HW Gen
- Cursor subject to ordering problem with HWC2 on Skylake



drm HWComposer: current status

Status for HW supported with drm Gralloc:

- Intel booting only specific gen (having RGBA support in Primary Planes)
- AMD booting on GCN 2nd gen and later (Atomic required)
- Nvidia booting but has stability issues (GUI restart, GPU lockup)

Status for vGPU supported with drm Gralloc:

- vmwgfx booting with HWC1, but shows black GUI/no icons
- virgl working with HWC1

AMDGPU support in android-x86

- AMDGPU drm Gralloc path
- AMDGPU LLVM target support
- LLVM build with new Mesa versions
- SurfaceFlinger RGBA Pixel Format constraint

AMDGPU drm Gralloc path:

- gralloc_drm_pipe backend is selected for drivers w/o ad hoc backend
- updated kernel with drm/amd/amdgpu support for SI, CIK, VI, AI
- updated mesa, libdrm
- updated llvm with AMDGPU target support

LLVM versions in android-x86

android-x86 branch	LLVM Version	Achievements
lollipop-x86	3.5	AMDGPU target build fix by Paulo Sergio Travaglia (pstiglia)
marshmallow-x86	3.7	FORCE_AMDGPU cflag to fix function prototypes (maurossi)
nougat-x86	3.8	AMDGPU target build ported to new llvm version (maurossi)
oreo-x86 rc1	3.9	AOSP moves to soong build AMDGPU ported (Rob Herring)
oreo-x86 r1	?	Mesa 18.1 : LLVM 5.0 required but AOSP supports only 3.9

radeonsi and amdgpu drivers support would have been lost in oreo-x86

LLVM new versions problem

PROBLEM : New LLVM version required, but AOSP supports only LLVM 3.9

- Updated LLVM version does not allow to compile core AOSP projects
- Collaboration started with Qiang Yu (AMD)
- Idea: side build libLLVM50 “for Mesa” separate shared library module
- AOSP does not allow duplicate module, variable: rename them all (!)
- Mesa changes: define libLLVM50 dependency and HAVE_LLVM cflag

Module renames can be reverted to get back “regular Android.mk” build

LLVM new versions: soong build

Procedure to add soong building rules to new LLVM versions:

- Started from AOSP LLVM 3.9 (with soong building rules by Rob Herring)
- Rebase to new LLVM version 5.0
- Fix the build with new LLVM versions
- Have ‘sed’ rename LLVM modules and build variables
- One commit to revert to “go back to normal” with documented changes
- Soong building rules tested on LLVM 5.0, 6.0, 7.0

LLVM 7.0 added to android-x86 repo

LLVM 5.0, 6.0 soong branches: https://github.com/maurossi/llvm

SurfaceFlinger RGBA pixel format

A recurrent problem

- AOSP SurfaceFlinger requires RGBA pixel format (mandatory)
- Workaround 10194508 (Fallback to BGRA in SurfaceFlinger) was removed
- GUI breakage for r300, r600, radeonsi, amdgpu, vmwgfx

How to cope with it

- Adding again equivalent workaround “EGL config fallback to simpler query”

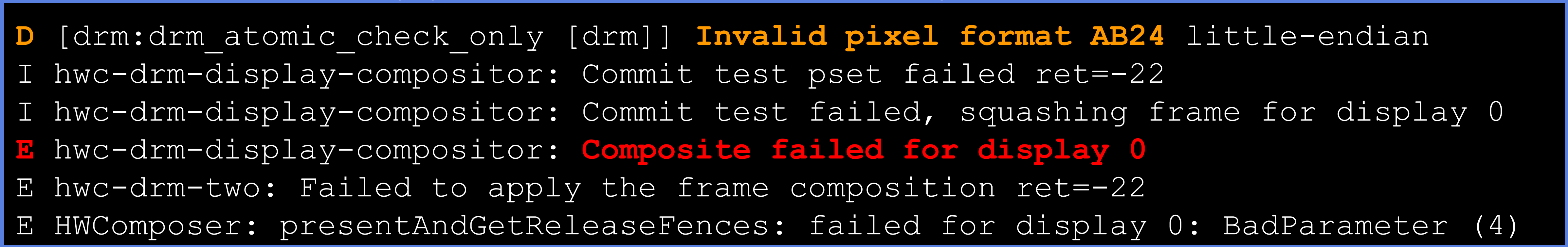
(but that is not a real solution)

How to solve it

- Adding support for RGBA/RGBX when possible in kernel drm driver

AMDGPU: HWC show stopper

HWC: What happens without RGBA pixel format? (AMDGPU)



```
D [drm:drm_atomic_check_only [drm]] Invalid pixel format AB24 little-endian
I hwc-drm-display-compositor: Commit test pset failed ret=-22
I hwc-drm-display-compositor: Commit test failed, squashing frame for display 0
E hwc-drm-display-compositor: Composite failed for display 0
E hwc-drm-two: Failed to apply the frame composition ret=-22
E HWComposer: presentAndGetReleaseFences: failed for display 0: BadParameter (4)
```

SurfaceFlinger HWcomposer throws error

=> no mode set => Black screen

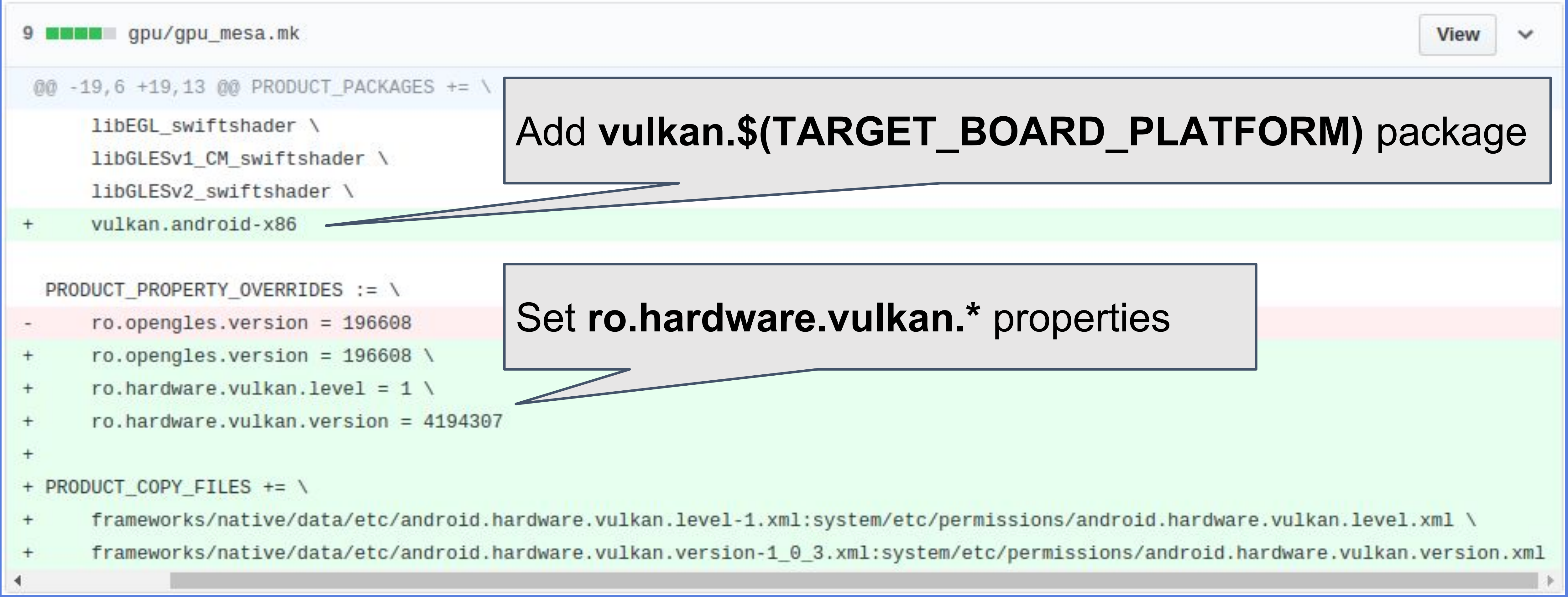
ABGR Patches submitted to amd-gfx, now in drm-next (thanks to Alex Deucher for guidance and review)

- Vulkan HAL build integration and initialization
- Vulkan anv CTS test results on Skylake GT2
- Vulkan radv CTS test results on supported GCN
- Vulkan Apps and Games used for testing
- Vulkan HAL outstanding work and next steps

Vulkan radv build for Android

Android makefiles implemented using anv as a reference
Tested with dEQP-VK tests, Benchmarks, Games
Submitted in Mesa 18.2
AOSP build problem: build radv conditionally to radeonsi
Released as part of Mesa 18.2

Vulkan anv: BoardConfig.mk



```
@@ -19,6 +19,13 @@ PRODUCT_PACKAGES += \
      libEGL_swiftshader \
      libGLESv1_CM_swiftshader \
      libGLESv2_swiftshader \
+     vulkan.android-x86

  PRODUCT_PROPERTY_OVERRIDES := \
-     ro.opengles.version = 196608
+     ro.opengles.version = 196608 \
+     ro.hardware.vulkan.level = 1 \
+     ro.hardware.vulkan.version = 4194307
+
+ PRODUCT_COPY_FILES += \
+     frameworks/native/data/etc/android.hardware.vulkan.level-1.xml:system/etc/permissions/android.hardware.vulkan.level.xml \
+     frameworks/native/data/etc/android.hardware.vulkan.version-1_0_3.xml:system/etc/permissions/android.hardware.vulkan.version.xml
```

Add **vulkan.$(TARGET_BOARD_PLATFORM)** package

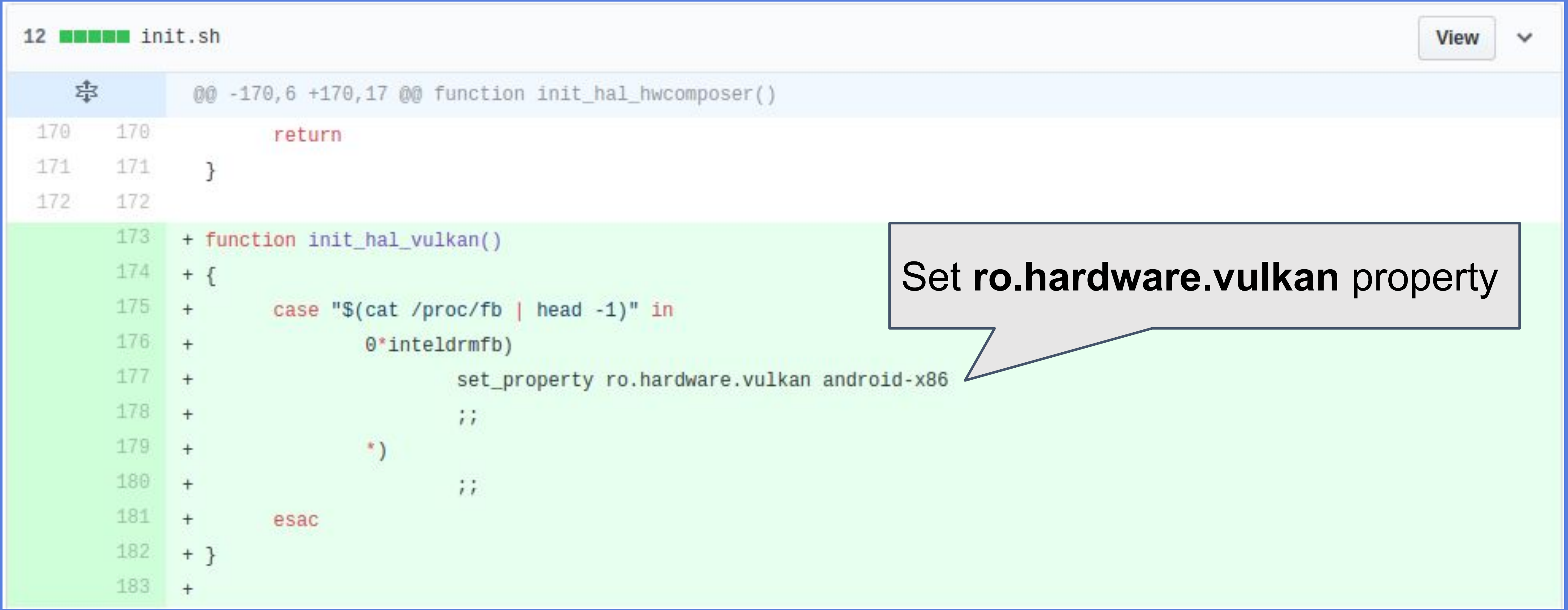
Set **ro.hardware.vulkan.*** properties

NOTE: Vulkan HAL anv package, properties and copy files in android-x86

android-x86.org

Android-x86

Vulkan anv: init.sh



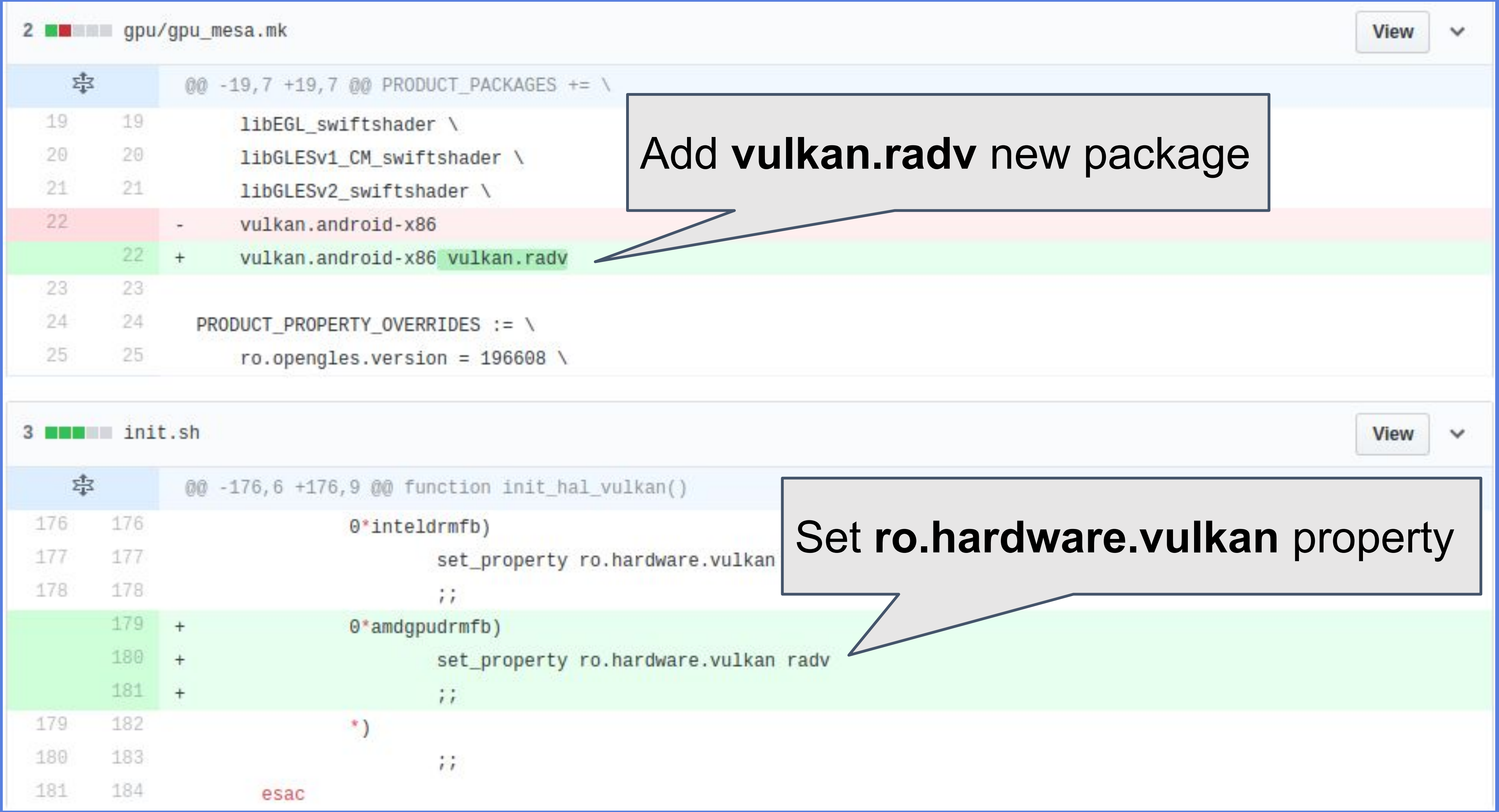
```
@@ -170,6 +170,17 @@ function init_hal_hwcomposer()
         return
 }

+function init_hal_vulkan()
+{
+        case "$(cat /proc/fb | head -1)" in
+                0*inteldrmfb)
+                        set_property ro.hardware.vulkan android-x86
+                        ;;
+                *)
+                        ;;
+        esac
+}
+
```

Set **ro.hardware.vulkan** property

NOTE: init_hal_vulkan() is added to the sequence of init_hal_*() functions

Vulkan radv: add support



gpu/gpu_mesa.mk

```
@@ -19,7 +19,7 @@ PRODUCT_PACKAGES += \
    libEGL_swiftshader \
    libGLESv1_CM_swiftshader \
    libGLESv2_swiftshader \
-   vulkan.android-x86
+   vulkan.android-x86 vulkan.radv

PRODUCT_PROPERTY_OVERRIDES := \
    ro.opengles.version = 196608 \
```

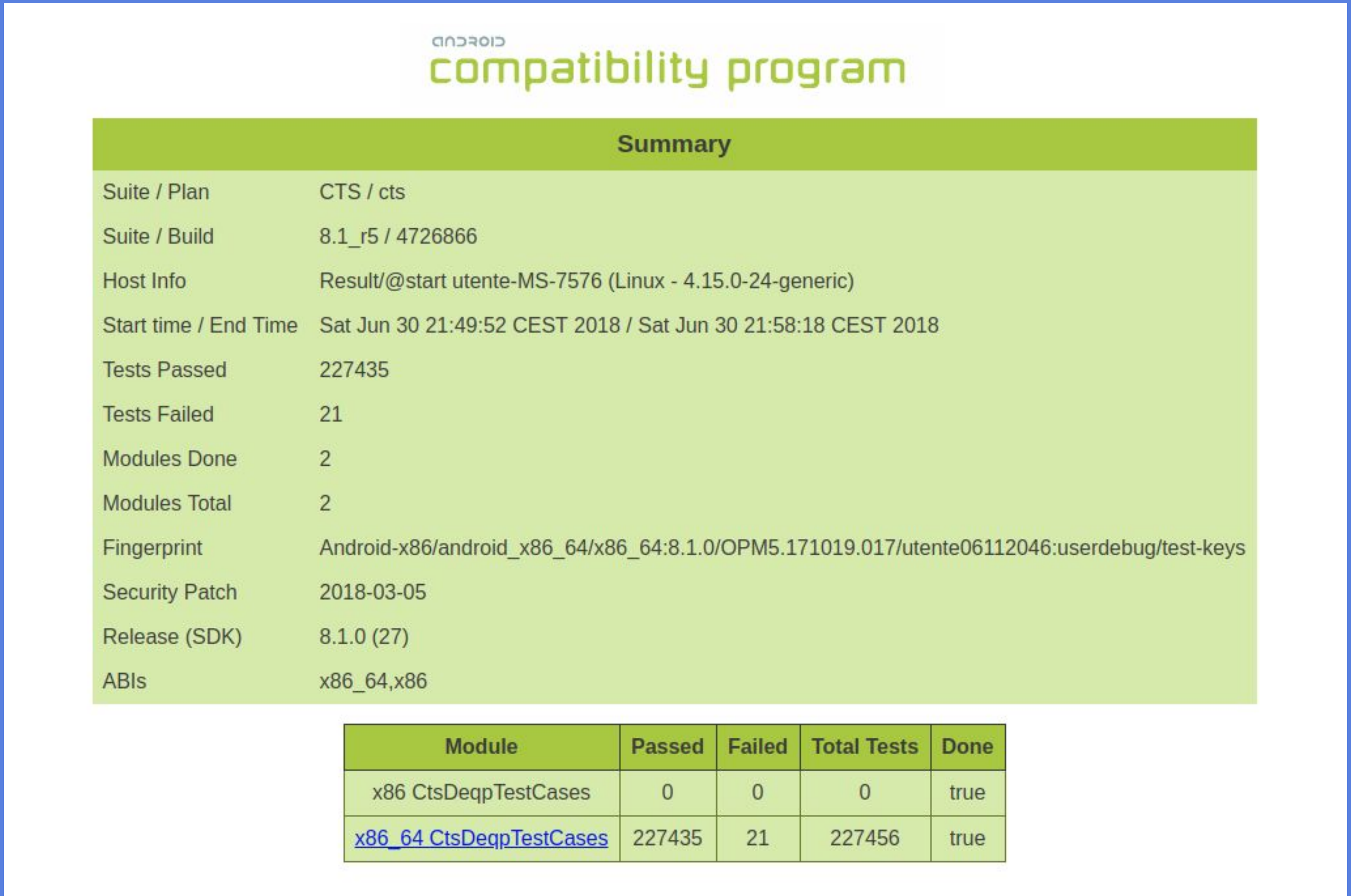
Add **vulkan.radv** new package

init.sh

```
@@ -176,6 +176,9 @@ function init_hal_vulkan()
                0*inteldrmfb)
                        set_property ro.hardware.vulkan
                        ;;
+               0*amdgpudrmfb)
+                       set_property ro.hardware.vulkan radv
+                       ;;
                *)
                        ;;
        esac
```

Set **ro.hardware.vulkan** property

anv CTS dEQP-VK results



android
compatibility program

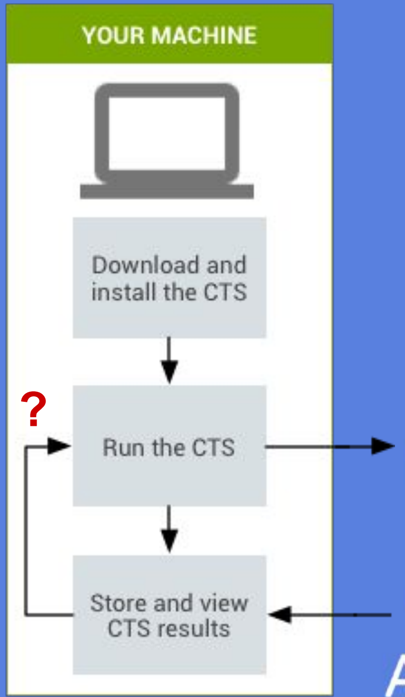
Summary	
Suite / Plan	CTS / cts
Suite / Build	8.1_r5 / 4726866
Host Info	Result/@start utente-MS-7576 (Linux - 4.15.0-24-generic)
Start time / End Time	Sat Jun 30 21:49:52 CEST 2018 / Sat Jun 30 21:58:18 CEST 2018
Tests Passed	227435
Tests Failed	21
Modules Done	2
Modules Total	2
Fingerprint	Android-x86/android_x86_64/x86_64:8.1.0/OPM5.171019.017/utente06112046:userdebug/test-keys
Security Patch	2018-03-05
Release (SDK)	8.1.0 (27)
ABIs	x86_64,x86

Module	Passed	Failed	Total Tests	Done
x86 CtsDeqpTestCases	0	0	0	true
x86_64 CtsDeqpTestCases	227435	21	227456	true

Mesa 18.2.0-devel (30-Jun-2018)

Very good results

Strange issue with interrupted tests runs



YOUR MACHINE

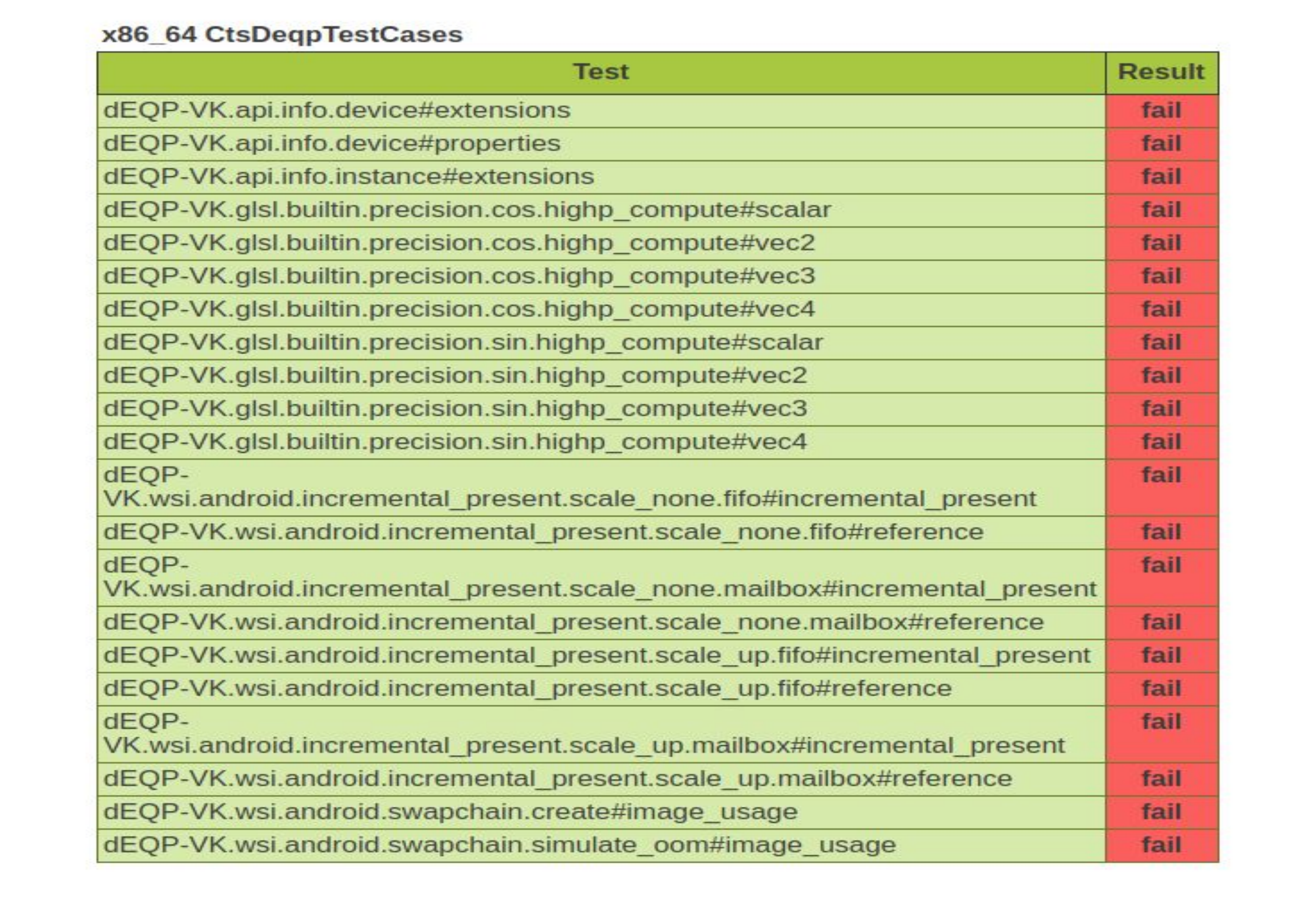
Download and install the CTS

Run the CTS

Store and view CTS results

Android-x86

anv CTS dEQP-VK failed tests



x86_64 CtsDeqpTestCases

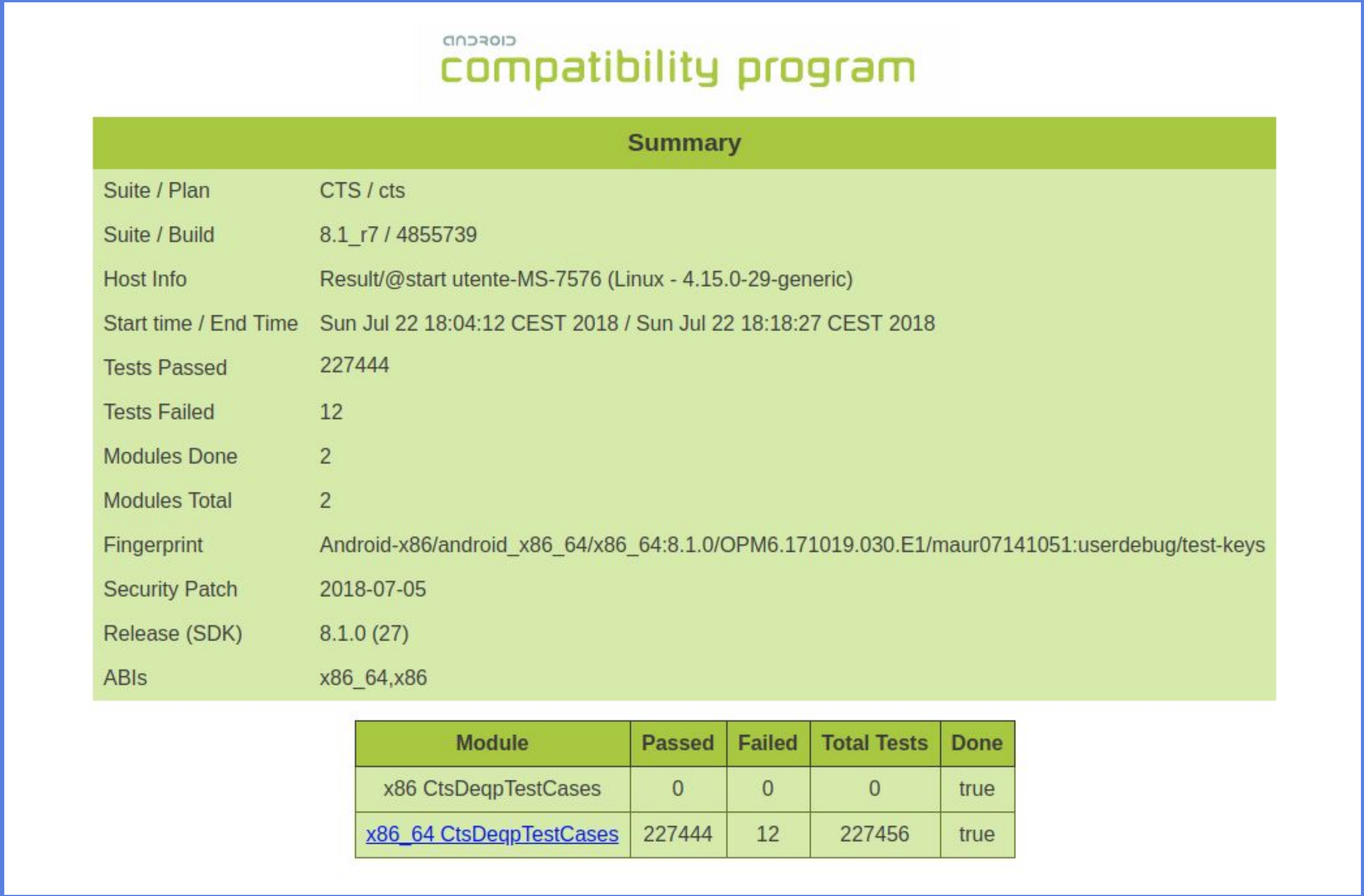
Test	Result
dEQP-VK.api.info.device#extensions	fail
dEQP-VK.api.info.device#properties	fail
dEQP-VK.api.info.instance#extensions	fail
dEQP-VK.glsl.builtin.precision.cos.highp_compute#scalar	fail
dEQP-VK.glsl.builtin.precision.cos.highp_compute#vec2	fail
dEQP-VK.glsl.builtin.precision.cos.highp_compute#vec3	fail
dEQP-VK.glsl.builtin.precision.cos.highp_compute#vec4	fail
dEQP-VK.glsl.builtin.precision.sin.highp_compute#scalar	fail
dEQP-VK.glsl.builtin.precision.sin.highp_compute#vec2	fail
dEQP-VK.glsl.builtin.precision.sin.highp_compute#vec3	fail
dEQP-VK.glsl.builtin.precision.sin.highp_compute#vec4	fail
dEQP-VK.wsi.android.incremental_present.scale_none.fifo#incremental_present	fail
dEQP-VK.wsi.android.incremental_present.scale_none.fifo#reference	fail
dEQP-VK.wsi.android.incremental_present.scale_none.mailbox#incremental_present	fail
dEQP-VK.wsi.android.incremental_present.scale_none.mailbox#reference	fail
dEQP-VK.wsi.android.incremental_present.scale_up.fifo#incremental_present	fail
dEQP-VK.wsi.android.incremental_present.scale_up.fifo#reference	fail
dEQP-VK.wsi.android.incremental_present.scale_up.mailbox#incremental_present	fail
dEQP-VK.wsi.android.incremental_present.scale_up.mailbox#reference	fail
dEQP-VK.wsi.android.swapchain.create#image_usage	fail
dEQP-VK.wsi.android.swapchain.simulate_oom#image_usage	fail

Mesa 18.2.0-devel
(30-Jun-2018)

dEQP-VK.wsi.*	10
dEQP-VK.glsl.*	8
dEQP-VK.api.*	3
Total	21

github.com/intel/mesa
solves many of these

radv CTS dEQP-VK results



android compatibility program

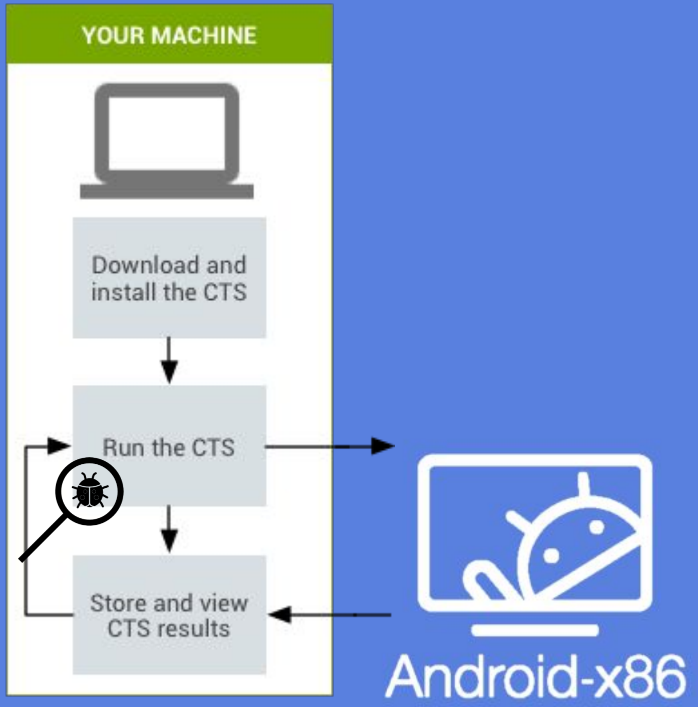
Summary	
Suite / Plan	CTS / cts
Suite / Build	8.1_r7 / 4855739
Host Info	Result/@start utente-MS-7576 (Linux - 4.15.0-29-generic)
Start time / End Time	Sun Jul 22 18:04:12 CEST 2018 / Sun Jul 22 18:18:27 CEST 2018
Tests Passed	227444
Tests Failed	12
Modules Done	2
Modules Total	2
Fingerprint	Android-x86/android_x86_64/x86_64:8.1.0/OPM6.171019.030.E1/maur07141051:userdebug/test-keys
Security Patch	2018-07-05
Release (SDK)	8.1.0 (27)
ABIs	x86_64,x86

Module	Passed	Failed	Total Tests	Done
x86 CtsDeqpTestCases	0	0	0	true
x86_64 CtsDeqpTestCases	227444	12	227456	true

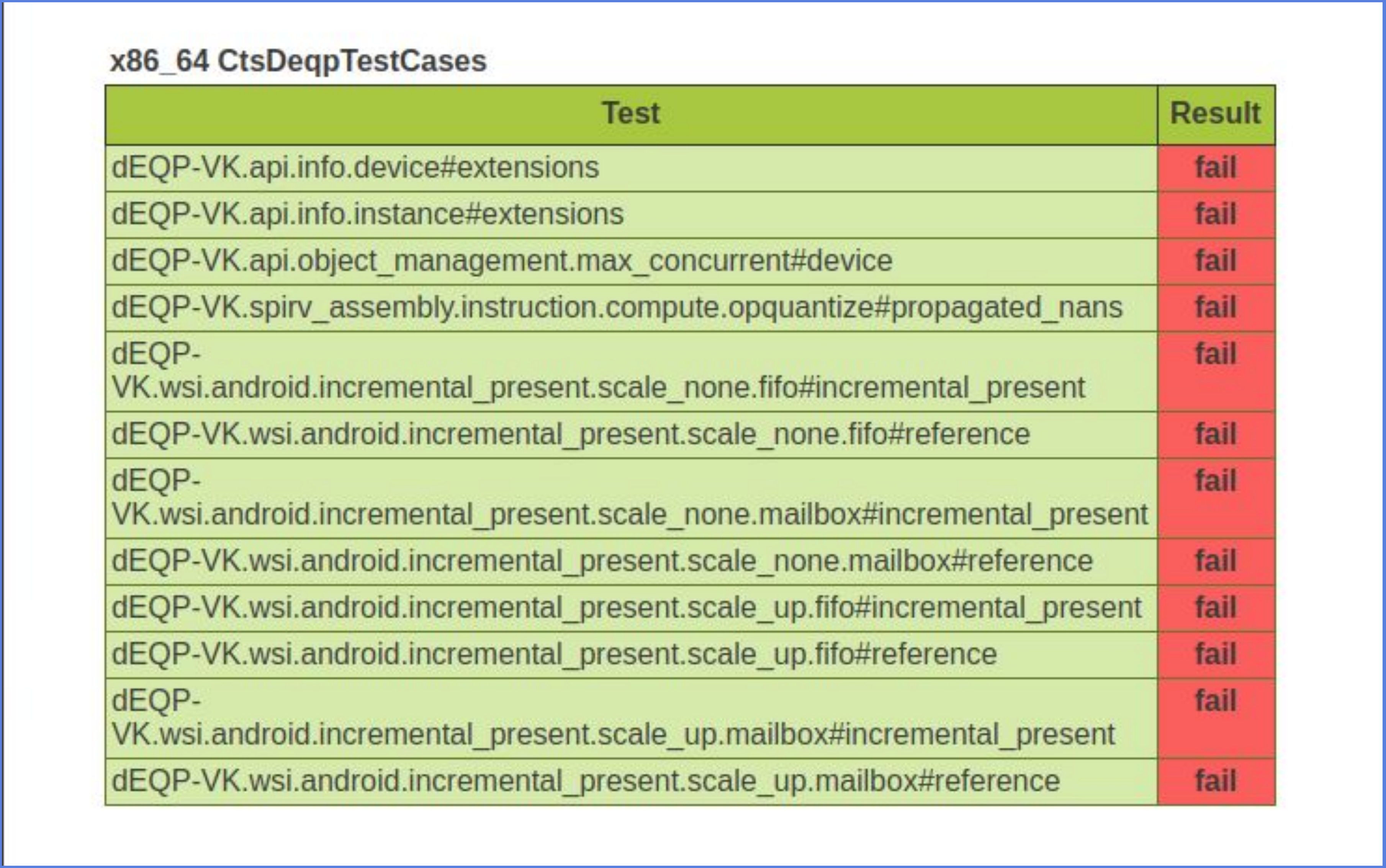
Mesa 18.2.0-devel (22-Jul-2018)

Very good results

CTS Bug reported on multiarch tests re-runs



radv CTS dEQP-VK failed tests



x86_64 CtsDeqpTestCases

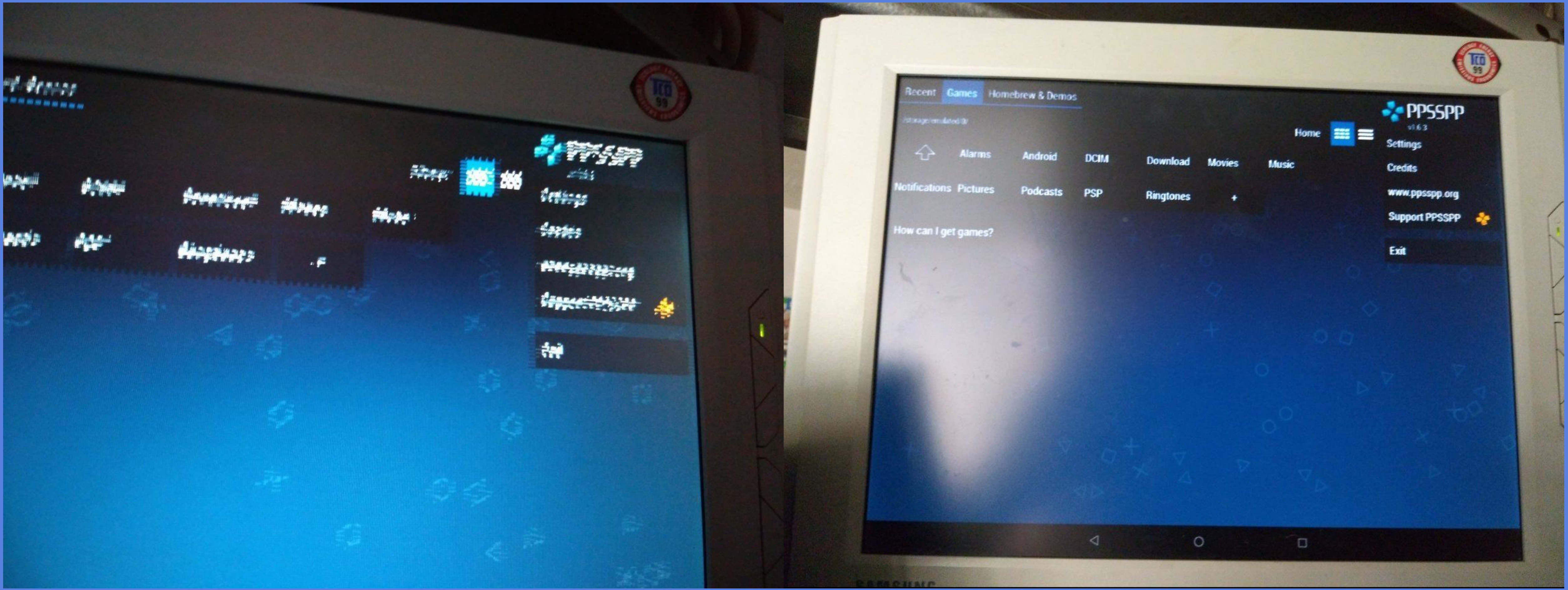
Test	Result
dEQP-VK.api.info.device#extensions	fail
dEQP-VK.api.info.instance#extensions	fail
dEQP-VK.api.object_management.max_concurrent#device	fail
dEQP-VK.spirv_assembly.instruction.compute.opquantize#propagated_nans	fail
dEQP-VK.wsi.android.incremental_present.scale_none.fifo#incremental_present	fail
dEQP-VK.wsi.android.incremental_present.scale_none.fifo#reference	fail
dEQP-VK.wsi.android.incremental_present.scale_none.mailbox#incremental_present	fail
dEQP-VK.wsi.android.incremental_present.scale_none.mailbox#reference	fail
dEQP-VK.wsi.android.incremental_present.scale_up.fifo#incremental_present	fail
dEQP-VK.wsi.android.incremental_present.scale_up.fifo#reference	fail
dEQP-VK.wsi.android.incremental_present.scale_up.mailbox#incremental_present	fail
dEQP-VK.wsi.android.incremental_present.scale_up.mailbox#reference	fail

Mesa 18.2.0-devel
(22-Jul-2018)

dEQP-VK.wsi.*	8
dEQP-VK.api.*	3
dEQP-VK.spirv.*	1
Total	12

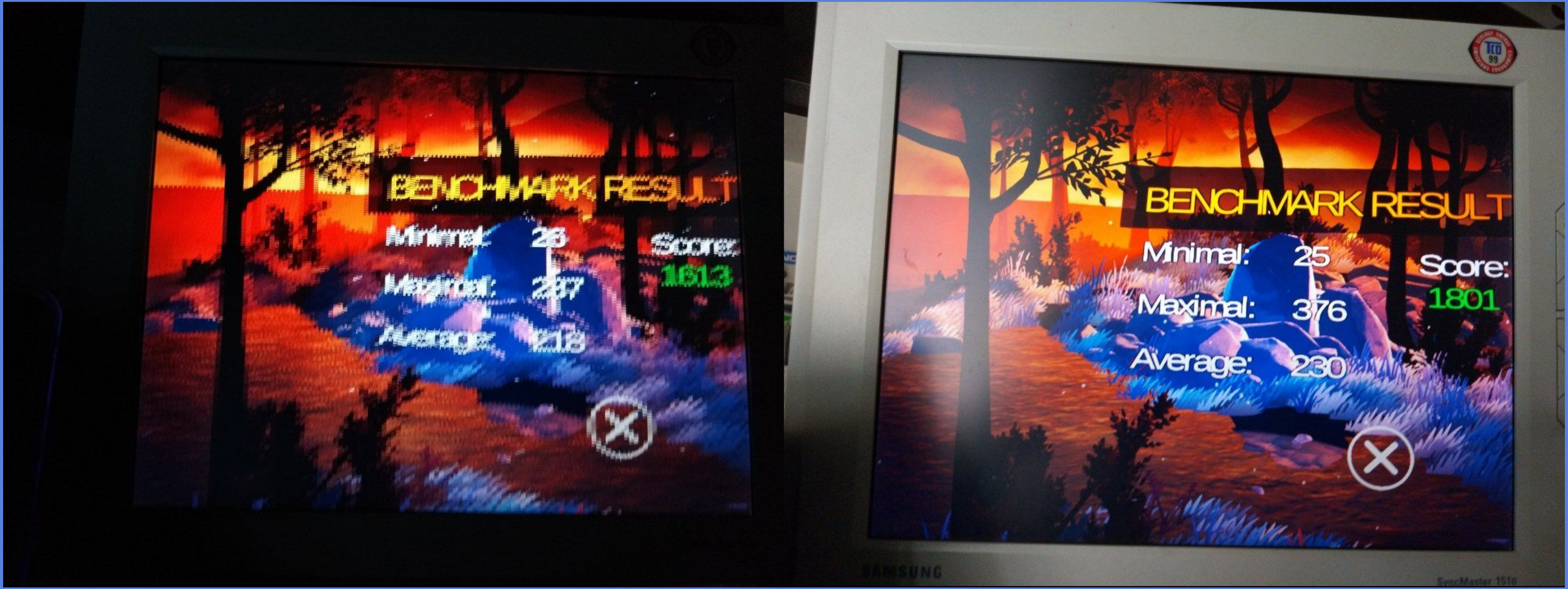
intel/mesa anv patches could be ported

Vulkan apps screen artifacts (1/2)



Root cause identified in gbm gralloc Usage Flags for HWC buffer

Vulkan apps screen artifacts (2/2)



Patch is now available in gbm gralloc master (robherring github)

Vulkan: Android Apps and Games

Android Apps
3DMark - API Overhead
3DMark - Slingshot Extreme
V1 Benchmark Pro
GFXbench 5.0
Google samples tutorials
LunarG examples
Sascha Willems examples
VulkanCapsViewer
Dolphin emulator
PPSSPP emulator

Android Games
Breakneck
Dirt Road Transport
Dream League Soccer
Heroes of Incredible Tales
Olympus Rising
Roblox
Score! Hero
Sky Force Reloaded
Toy Zombies Lite
Vainglory

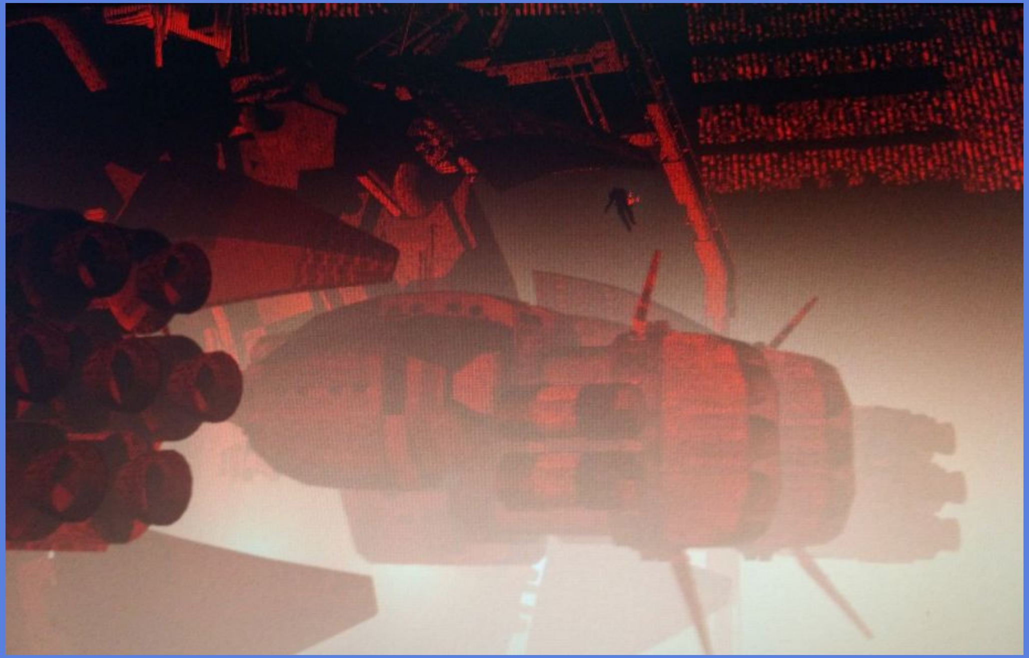
Vulkan app issues (test cases)

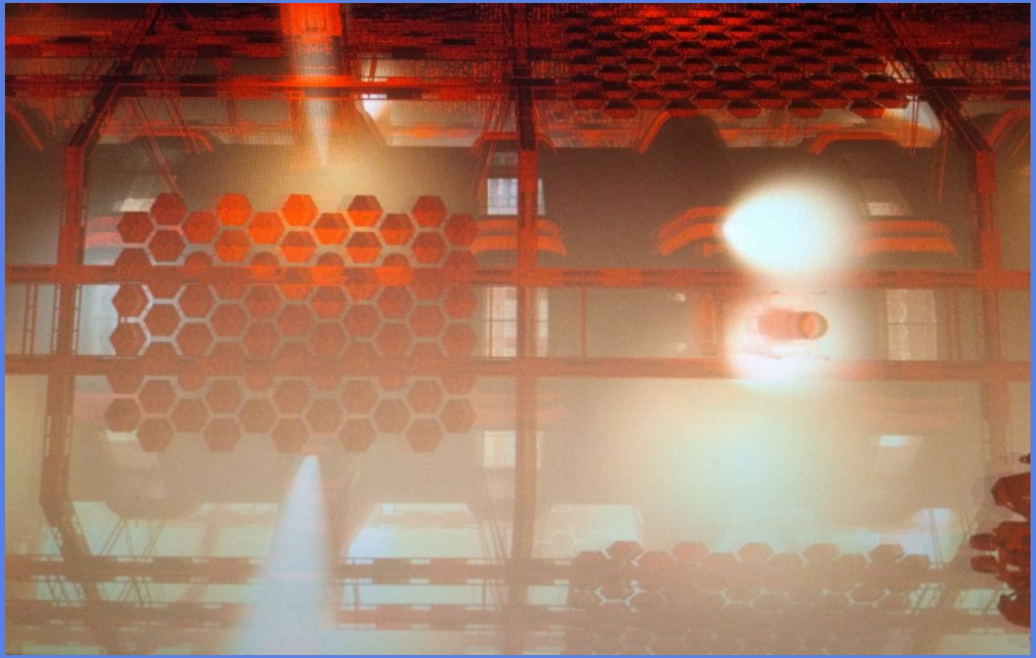
GFXbench 5.0 Vulkan tests not executed “OUT_OF_MEMORY” (anv and radv)

V1 Vulkan Pro display sync issues (radv on Mullins)

3DMark “API Overhead” freeze (radv on GCN 2nd gen “GPU fault detected: 146”)

3DMark “Slingshot Extreeme” orange/black renders (radv GCN 2nd and 4th gen)





HWC/Vulkan on android-x86: next steps

Public release of xml manifest, to accelerate fix of pending issues

Complete testing with HWC2

Official iso release when sufficiently stable/usable

Testing of new drivers supporting Atomic Display Framework

Vulkan nouveau, Vulkan virgl, Vulkan SwiftShader (when available)

Questions?

Thank you!