



COLLABORA

# Mainline Explicit Fencing

**Gustavo Padovan**

**Open First**



# Agenda

- **Android Sync Framework**
- **Mainline Explicit Fencing**
- **Current Status**



# Android Sync Framework

- Android's Explicit Fencing implementation
- Use fd for fence passing
- **Sync Timeline** to control ordering
- **Sync Point** to represent a fence
- **Sync Fence** for fd passing



# Sync Timeline



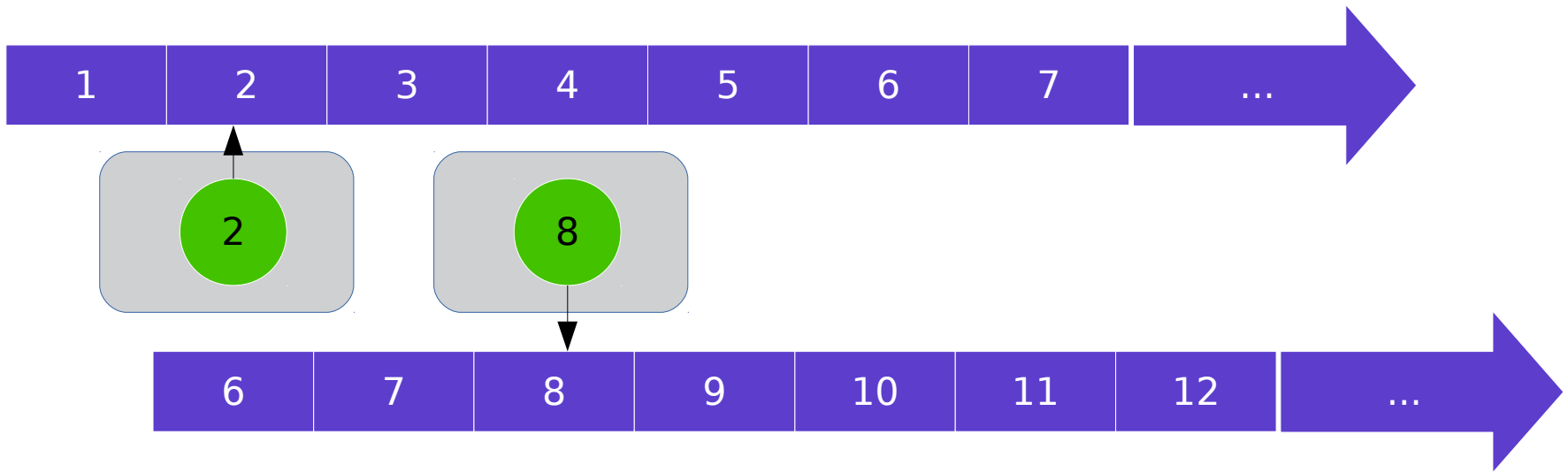
- Monotonically increasing counter
- Usually one timeline per driver context

# Sync Point



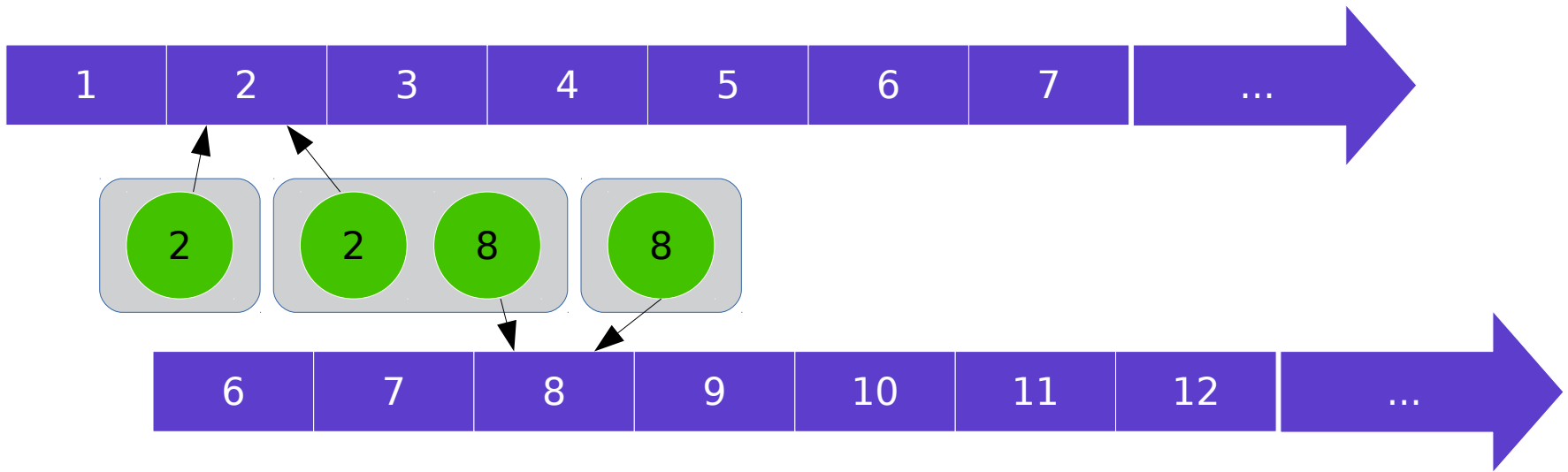
- It is the fence
- Represents a value on the timeline
- Three states: active, signaled and error

# Sync Fence



- Wrap Sync Point into a file
- Also have active and signaled states
- Shared via fd-passing to/from userspace

# Sync Fence



- Sync fences can be merged!
- It can contain many Sync Points



# Android Sync Framework - ioctls

- `sync_wait(fd, timeout)`
- `fd3 = sync_merge(fd1, fd2)`
- `sync_fence_info(fd, info)`





## Sync Framework de-staging

- Android Sync added to staging in 2013
- Mainly need for fd-passing
- Removed Sync Timeline
- Removed Sync Point
- Reworked Sync Fence



## Sync File

- Renamed Sync Fence to Sync File
- Changed ioctl API
  - Provided patch to Android's libsync
- Removed internal kernel API
- Used strictly for fd-passing
  - `sync_file = sync_file_create(fence)`
  - `fence = sync_file_get_fence(fd)`



## struct fence\_array

- Subclass of struct fence
- Store multiple fences
- Useful for merged Sync File
- Hide complexity from the drivers
- fence\_is\_array(fence)



## DRM/KMS

- Only available for Atomic Modesetting
- Entirely in DRM Core
- Extended DRM properties
- in-fences: fences to wait on
- out-fences: fences to be signaled by KMS



## DRM/KMS: in-fences

- in-fences: fences to wait on
- IN\_FENCE\_FD property on each DRM Plane
- Receives sync\_file fds carrying fences
- `drm_atomic_helper_wait_for_fences()` helper



## DRM/KMS: out-fences

- One out-fence per DRM CRTC
- `OUT_FENCE_PTR` property on each DRM CRTC
- It signals at CRTC scanout
  - It means the **previous** buffer can be reused.
  - New semantic supported by HWC2



## DRM/renderer

- Similar to KMS side
- Extends execbuf ioctl args on each driver
- Every driver needs sync\_file/fences support
- Done on freedreno
- WIP on i915 and virgl



## Mesa

- `EGL_ANDROID_native_fence_sync`
  - Receive out fence fd
- `EGL_ANDROID_wait_sync`
  - Make the GPU wait for in fence to signal
- WIP by Rob Clark
- kmscube running on explicit fencing





## drm\_hwcomposer

- Already support DRM fences semantic
- Use it as example userspace for upstream
- WIP by Sean Paul and Robert Foss
- Already working with padovan/fences



## Current Status Summary

- Sync File synchronization de-stage: DONE
- SW\_SYNC validation de-stage: DONE
- fence\_array: DONE
- DRM/KMS: WIP – 4.10?
- DRM/renderer: WIP – 4.10?
- MESA: WIP
- igt-gpu-tools: WIP
- drm\_hwcomposer: WIP
- Wayland: WIP
- V4L explicit fences: TODO

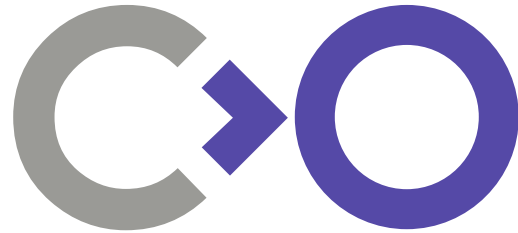


# Thank you to everyone involved

Daniel Vetter, Rob Clark, Greg KH, Daniel Stone, Robert Foss, Sean Paul, Stéphane Marchesin, Maarten Lankhorst, Chris Wilson, Christian König, David Reveman, Kristian Høgsberg and others.



COLLABORA



**Thank you!**

Gustavo Padovan  
[gustavo@padovan.org](mailto:gustavo@padovan.org)  
[www.padovan.org](http://www.padovan.org)  
[www.collabora.com](http://www.collabora.com)