

Embrace the Atomic Display Age

LCA 2016, Geelong
Daniel Vetter



Accidents of history

- userspace mode setting in X, maybe
- FBDEV just a framebuffer ...
- ... combined with unsuitable acceleration
- hence DRM

7 or 8 years ago ... KMS

- real kernel driver
- in-kernel memory manager for GPU buffers
- supporting multiple screens
- desktop compositing using GL

Problem No. 1

- 3 displays, only 2 clock generators
- intermediate state could have 3 different clocks
- ... and your compositor dies
- avoidable, but atomic switching is better

Why Suddenly Overlays Again?

- 1080p YUV: ~4.1 MB, 4k RGB: ~33MB per frame
- 4.1MB + 33MB + 33MB with compositing per frame
- 4.1MB with overlay
- ... or split screen, UI overlay, ...
- ~ 4.2 GB/s at 60fps for compositing

It's all About Mobile!

- can't draw&composit entire screen
- video, because more idle = less power
- atomic required to avoid tearing

3 years ago ... Android's ADF

- replaced FBDEV + custom hacks horror show
- only one update queue
- kept old mistakes like DPMS
- resurrected old mistakes like midlayers

ADF? Not for Upstream

- not extensible, not generic
- only atomic for plane updates
- complete new subsystem incompatible with existing drivers
- ... and incompatible with existing userspace

Prep Work: Universal Planes

- merged 2 years ago
- cursor, primary, overlay planes ...
- it's all the same

Upstream Atomic ABI

- partial updates, for backwards compatibility
- properties as generic transport
- standardized sets of properties as extensions
- simple value, object, blob properties

More Upstream Atomic ABI

- cursor hack to appease X
- TEST_ONLY for discovery of constraints
- ALLOW_MODESET flag

Internal Driver Interface

- standardized properties decoded in core
- state duplication for partial updates
- subclassable for private properties
- check/commit split

Concurrent Updates

- per-object state structures
- per-object locks
- plus magic wait/wound locking
<https://lwn.net/Articles/548909/>

Say No to Midlayers

- completely new atomic helper library
- much more modular
- strict state transition guarantees for driver hooks
- implements all legacy IOCTL
- intermediate plane helpers for smooth conversion of existing drivers

Just in the Last Few Months

- suspend/resume helpers
- atomic fbdev emulation
- better support for runtime PM in general
- thousands lines of documentation

Missing & Wanted

- generic async commit
- explicit fencing support
- faster than vblank
- testsuite, based on i-g-t
- much extensions: color management, blending, ...

Documentation

- conversion HOWTO for legacy drivers:
<http://blog.ffwll.ch/2014/11/atomic-modeset-support-for-kms-drivers.html>
- design overview on LWN: <https://lwn.net/Articles/653071/>
<https://lwn.net/Articles/653466/>
- DRM DocBook: <https://01.org/linuxgraphics/gfx-docs/drm/>
- presentation for driver writers:
<http://blog.ffwll.ch/2015/09/xdc-2015-atomic-modesetting-for-drivers.html>

One Atomic Display API

DRM atomic core&helper libraries developed by:

Ville Syrjälä, Daniel Vetter, Maarten Lankhorst, Ander Conselvan de Oliveira (all Intel), Rob Clark (Redhat), Laurent Pinchart (ideasonboard), Daniel Stone, Gustavo Padovan (all Collabora), Thierry Reding (Nvidia), Sean Paul (Google), Inki Dae (Samsung), ...

To Rule them All

- drm_hwcomposer for Android
- Ozone on CrOS
- weston/wayland
- xf86-video-modesetting for X
- FBDEV emulation

And Forever Bind Them

i915 (Intel), tegra (Nvidia), msm (Qualcomm), exynos (Samsung), omap (TI), rockchip, vc4 (Broadcomm), virtio (QEMU), rcar-du (Renesas), fsl-du (Freescale), atmel, ...

And more to come with every release!