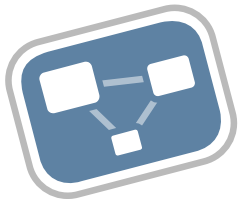
DRM Leases

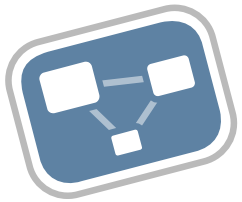
Keith Packard
keithp.com
Valve



VALVE

Head-Mounted Displays

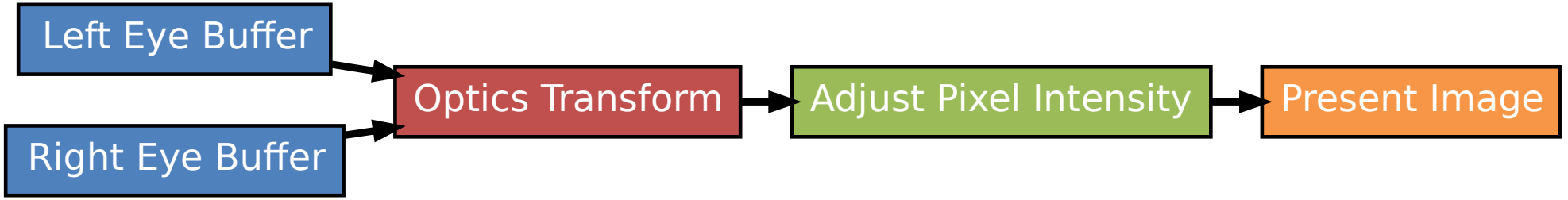
- Used for VR
- IMU for position and orientation tracking
- Display with optics for view

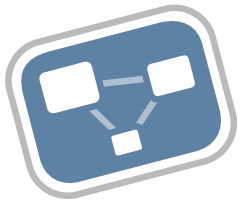


VALVE®

HMD Display Process

- Application generates left/right eye buffers
- VR Compositor constructs unified frame buffer
  - Inverts distortion caused by headset optics
  - Inverts pixel intensity offsets in individual

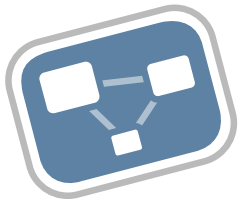




VALVE®

HMD Requirements

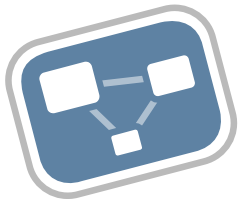
- Hard real-time
  - New frame 90 times per second
  - Bounded latency requirement
  - Vary amount of work to hit target
- Definitely not part of the desktop
  - No window management
  - No “regular” apps
  - Just showing a static image is “hard”



VALVE®

HMD Display Options

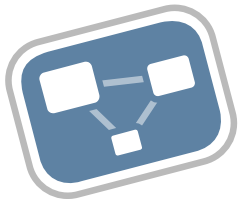
- ICCCM conventions for hiding displays
  - + no kernel or X changes required
  - - Involve the desktop. All of them.
  - - Does not address latency issue.
- RandR changes to hide displays
  - + No kernel changes required
  - - X changes required
  - - Does not address latency issue.
- Meta display server to manage outputs
  - + No new X protocol
  - + No kernel changes
  - - X changes required
  - - Does not address latency issue
- Kernel changes to let apps “borrow” display
  - + Address latency concerns
  - + No visible changes to desktop
  - - Kernel changes
  - - X changes



VALVE®

Get X out of the picture

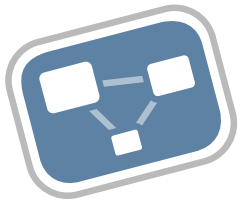
- X isn't helping here
  - Adds latency when flipping
  - Even worse when a compositor is running
- Make HMD app display directly to the device
  - It's already got an FD pointing at the device
  - “just” needs mode setting access.
- Construct a mechanism to pull display out of X and hand to HMD app.
  - Want to leave X able to recover when the app crashes



VALVE®

Leases

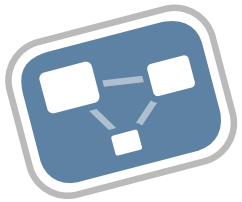
- A “lease” is a contract between X and the VR app naming a set of display resources
- The “lessor” (e.g. X server) is the current controller of the leased resources and promises to leave the resources alone while the lease is active
- The “lessee” (e.g. VR app) is free to set modes, flip frame buffers, DPMS on/off at will.
- When the lesse terminates the lease, the lessor takes the display back.



VALVE®

Side Issue – vblank API

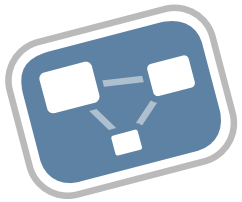
- Current vblank API is a bit of a mess
  - Three functions, one IOCTL
  - Only supports 32-bit frame counter. Wraps in only a few years
  - Only supports microsecond resolution
- Add two new IOCTLs
  - CRTC_GET_SEQUENCE
    - Get last vblank sequence and time
  - CRTC_QUEUE_SEQUENCE
    - Queue event for delivery at specified sequence
  - 64-bit frame counter
  - Nanosecond resolution (ktime_t internally)



VALVE®

Building a Kernel Patch Story

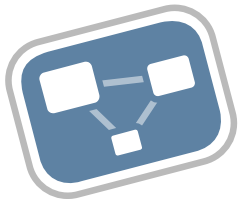
- It doesn't matter how the code was written
  - People don't usually write books from front to back, why should we expect code to be written this way?
- Patches should tell a consistent story
  - Reviewers time is precious; patches should do one thing which is described in the first line of the commit message
- Each patch should be review-able by itself
  - Even if the patch only makes sense in context, it should still be possible to see the commit message and ensure that the patch does exactly that.
- Each patch should compile and run by itself
  - Event if it doesn't do anything useful.
  - Without this, you break bisect, which makes debugging in the future much (much) harder.



VALVE®

Add file parameter to mode resource lookup

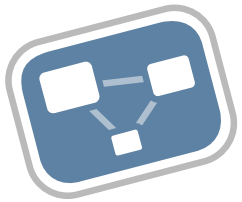
- Significant API churn, no functional change.
- Allows adding access control in future
- Largely mechanical change, easy to review



VALVE®

Let render nodes query mode objects

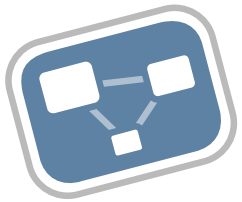
- Permits Vulkan client discovery of resources in AcquireXlibDisplay mode
- Drivers with split render/display files would need access to some kind of read-only version of the display object.
- Just changes access control flags



VALVE®

Lease infrastructure

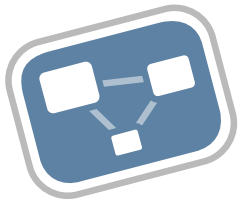
- The heart of the change; touches existing code as little as possible
- Each master has list of lessees
- Each lessee has list of mode resources



VALVE®

Lease access control

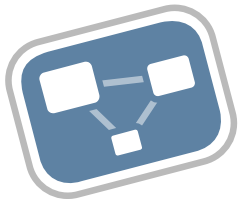
- Only for connector, encoder and crtc resources
- Separated from lease infrastructure patch, touches only the access control code



VALVE

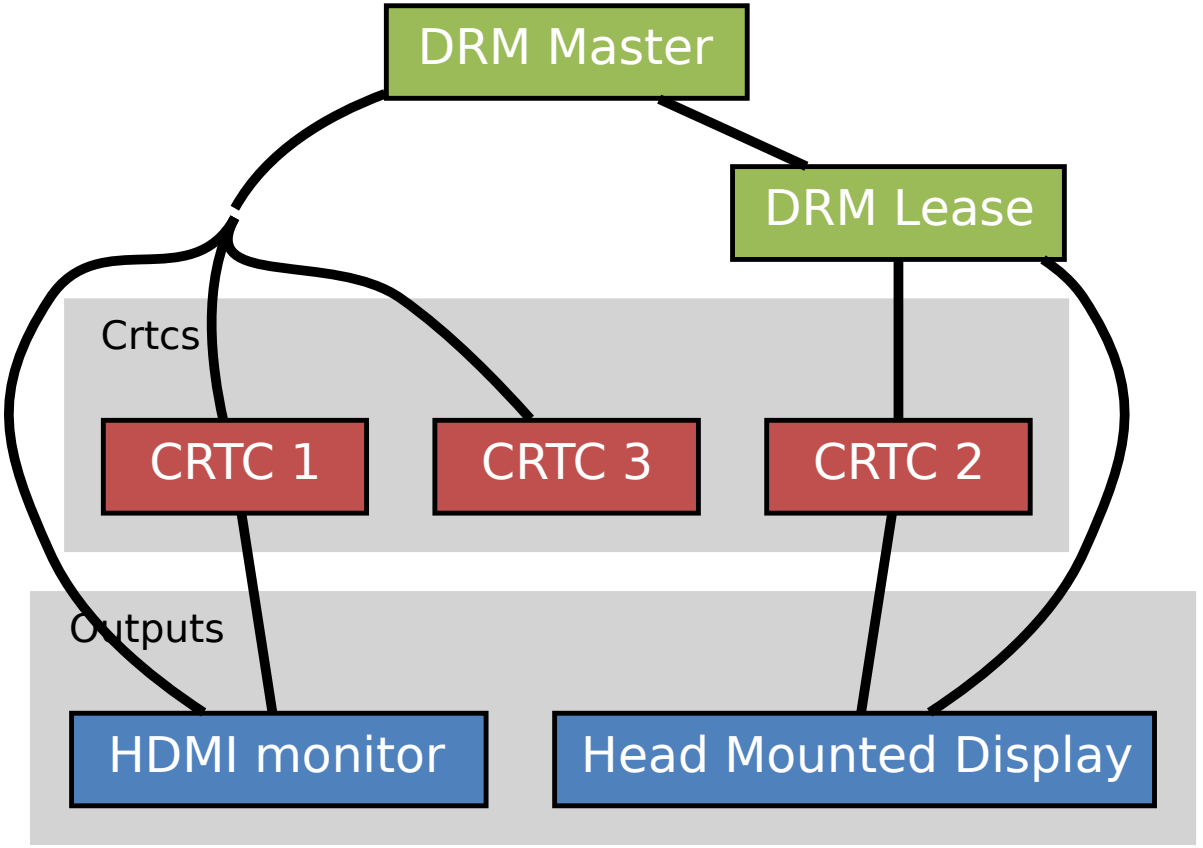
New lease IOCTLs

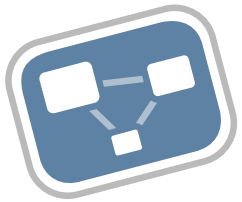
- Hooks up the lease infrastructure to user mode.



VALVE®

Leasing in Action

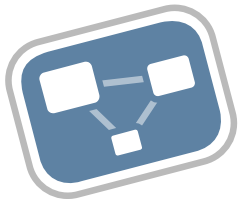




VALVE

X and Leasing

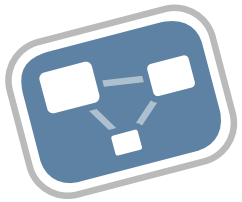
- X Masks leased objects from X clients
  - Leased outputs always appear to be disconnected
  - Leased CRTCs cannot be used with any output
- X clients do not expect RandR resources to come and go on the fly
  - Even though RandR doesn't require they be persistent,
  - Clients now depend on that. Unintentional ABI.
- KMS has connectors and encoders, X has only outputs. X leasing automatically assigns encoders for each output.



VALVE®

RandR Additions

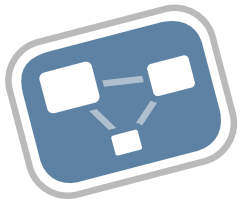
- RRCreateLease
  - Takes list of X outputs and X crtcs
  - Returns XID for lease and FD reference to DRM device
- RRFreeLease
  - Free XID
  - Optionally revoke active lease
- RRLeaseNotifyEvent
  - Notifies of new and terminated leases



VALVE®

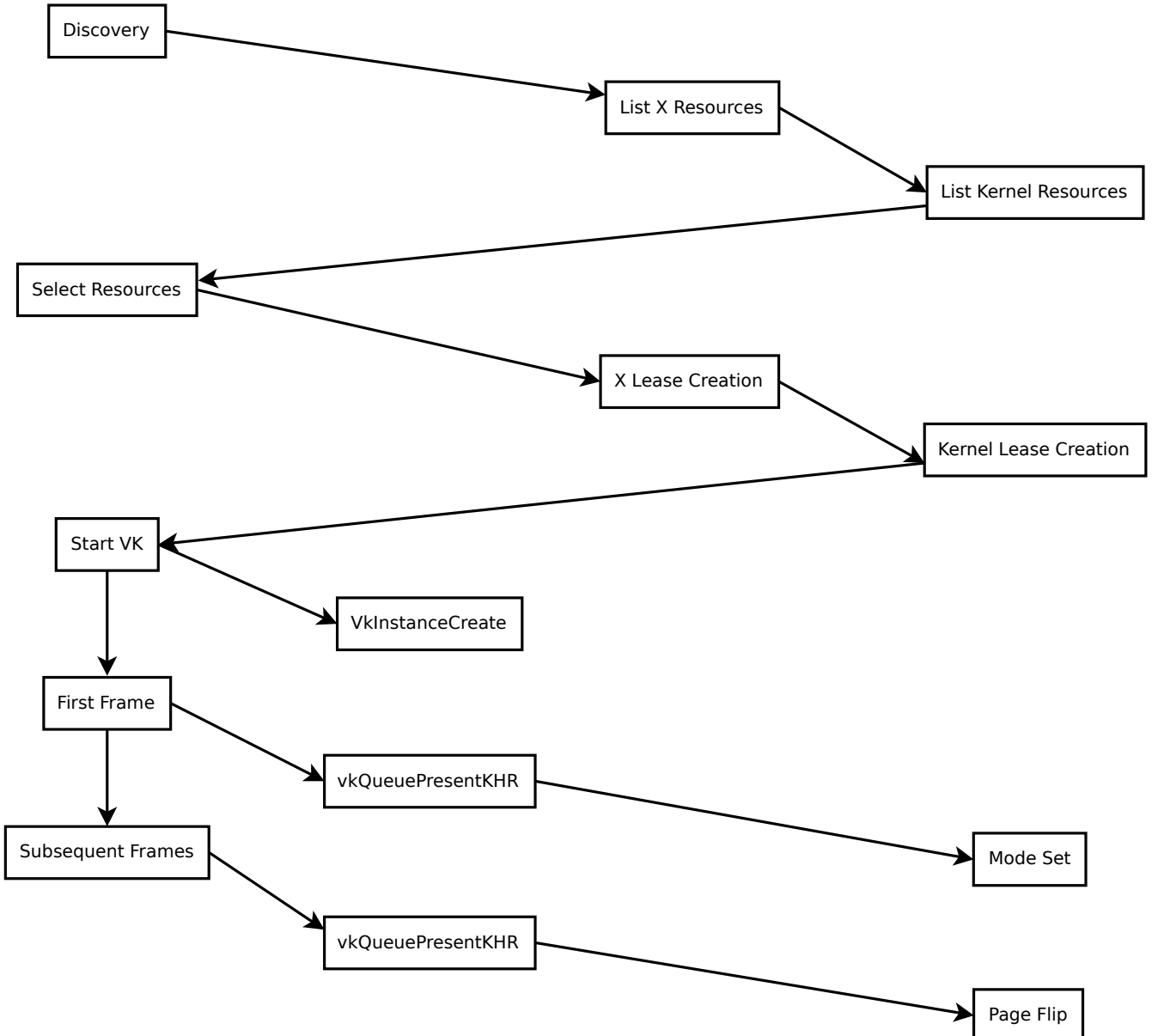
Various X fixups

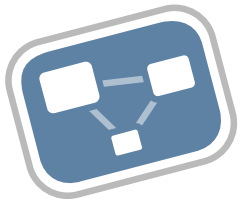
- Turn off cursor when disabling CRTC
  - Otherwise, the lessee will end up with a random cursor on the screen
- Add a CONNECTOR_ID property to each RandR output
  - Lets Vulkan driver map between X and KMS resources



VALVE

Original Plan

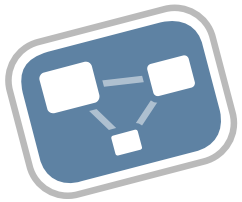




VALVE

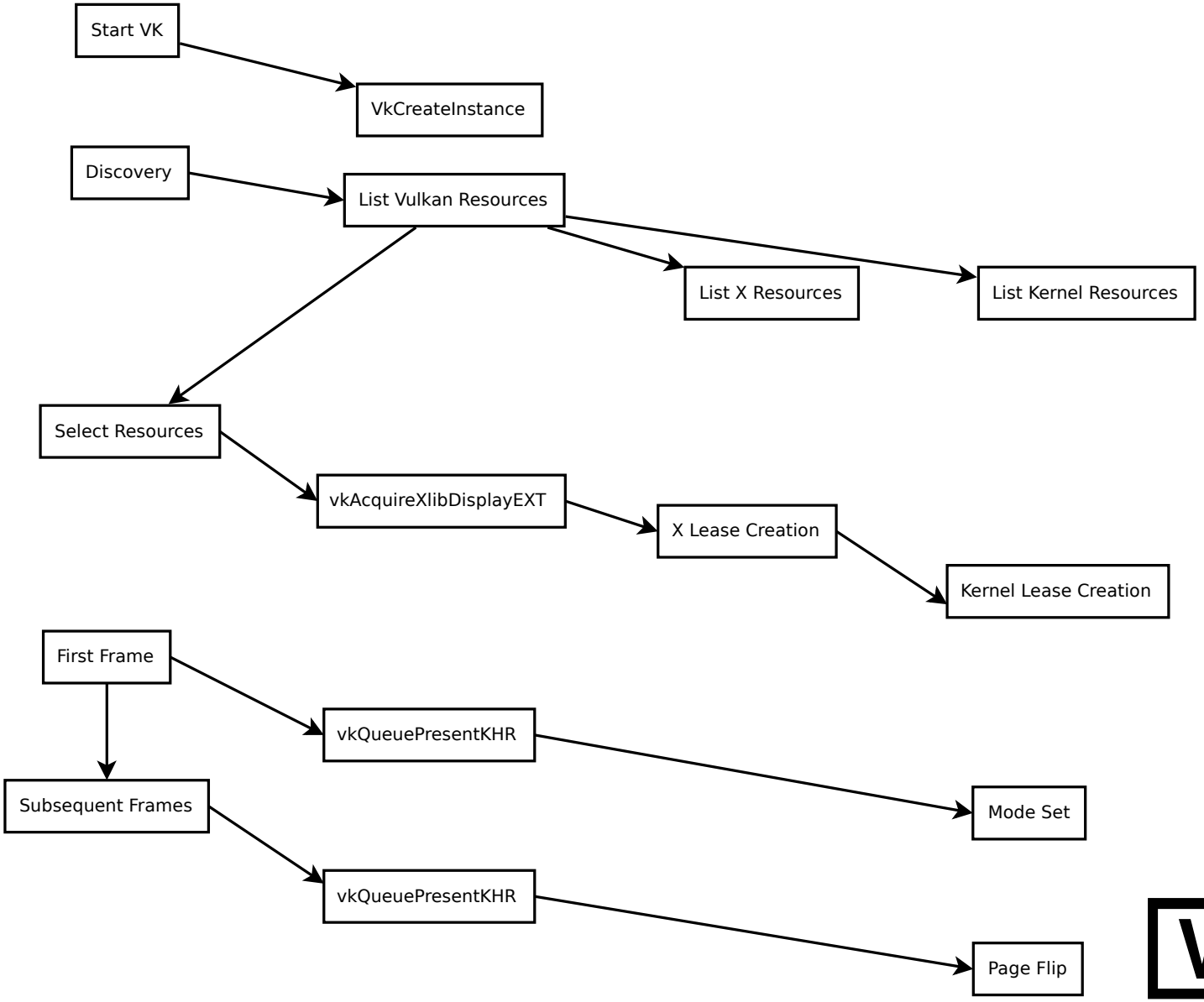
Vulkan Bits for original plan

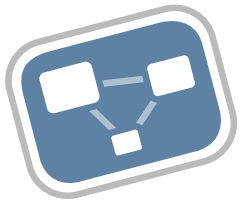
- VK_KEITHP_kms_display
  - New instance extension
  - Adds structure to pass device fd into VkCreateInstance
  - Drivers can use it instead of their own fd
- Implement VK_KHR_display and VK_EXT_direct_mode_display extensions
  - Create surfaces and swapchains on physical devices



VALVE

vkAcquireXlibDisplayEXT

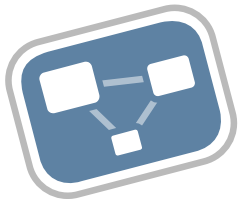




VALVE®

Additions for VK_EXT_acquire_xlib_display

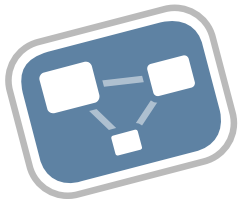
- Instance already has render node fd opened
- Must create X lease to get master fd
- Code must now deal with two fds
- Map RandR output to KMS connector using CONNECTOR_ID property



VALVE®

VK_EXT_display_control

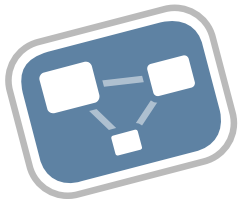
- Needed for vblank fences on direct mode displays
- Uses new kernel IOCTLs to provide 64-bit sequences and nano-second resolution times.
- Adds new fence types inside drivers, so now we have WINSYS (original) and WSI (new).
- Also adds DPMS support for direct displays



VALVE®

Remaining Issues

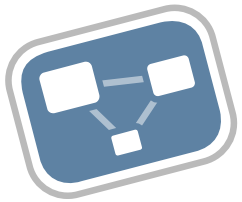
- Hide HMD from apps even when no VR app is running.
  - Even fbdev.
  - Maybe some kind of EDID registry that the kernel knows about?
- Wire up hotplug events for VK_EXT_display_control
  - Part of the spec, but I haven't needed it yet.



VALVE®

Thanks!

- git://people.freedesktop.org/~keithp/linux drm-lease-v3 drm-sequence-64-v2
- git://people.freedesktop.org/~keithp/xserver drm-lease
- git://people.freedesktop.org/~keithp/randrproto drm-lease
- git://people.freedesktop.org/~keithp/xcb/proto drm-lease
- git://people.freedesktop.org/~keithp/drm drm-lease
- git://people.freedesktop.org/~keithp/mesa drm-lease drm-lease-intel



VALVE®