

Zero-Copy Display of Guest Framebuffers using GEM

John Baboval

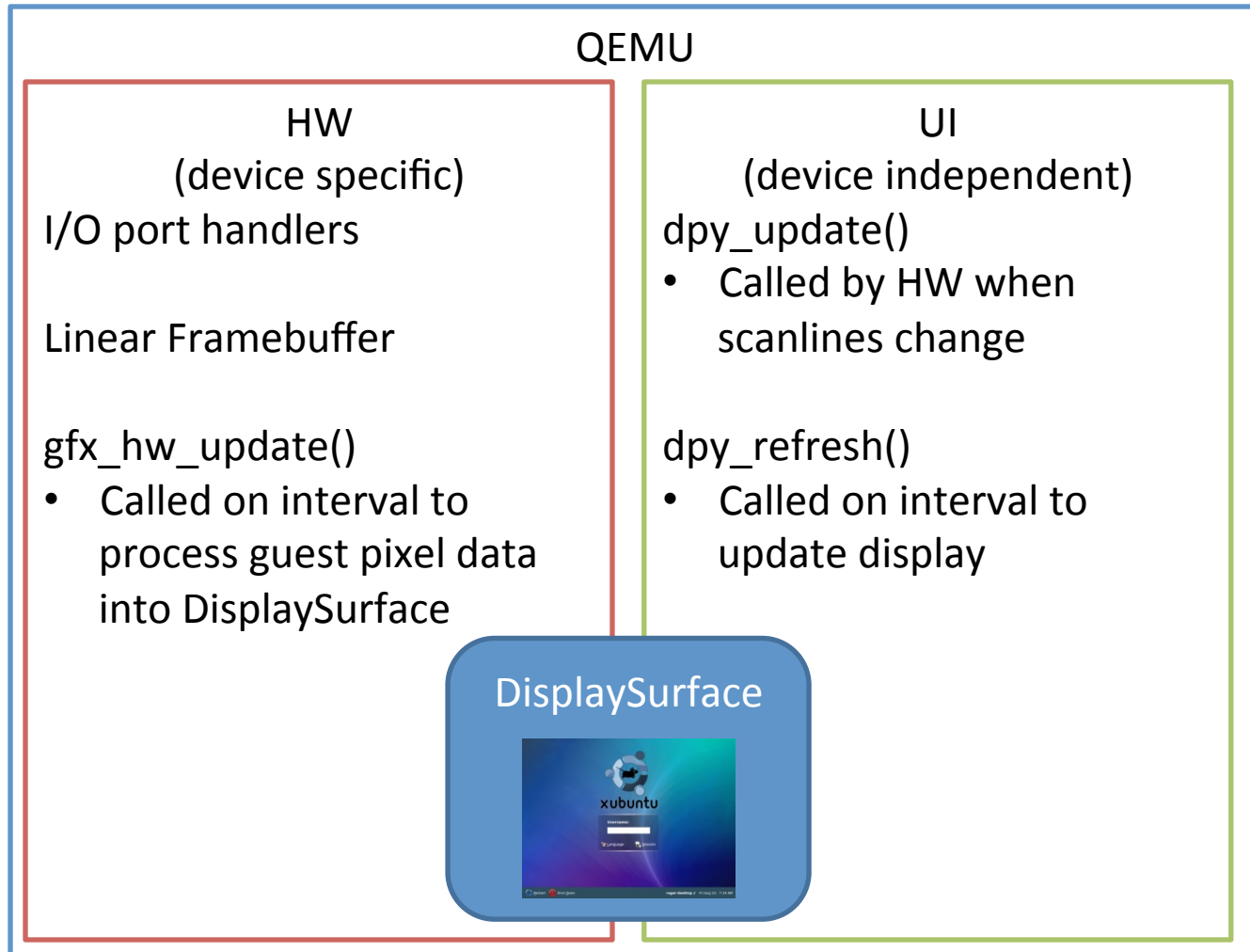
Citrix Systems, Inc.

October 24th, 2013

Agenda

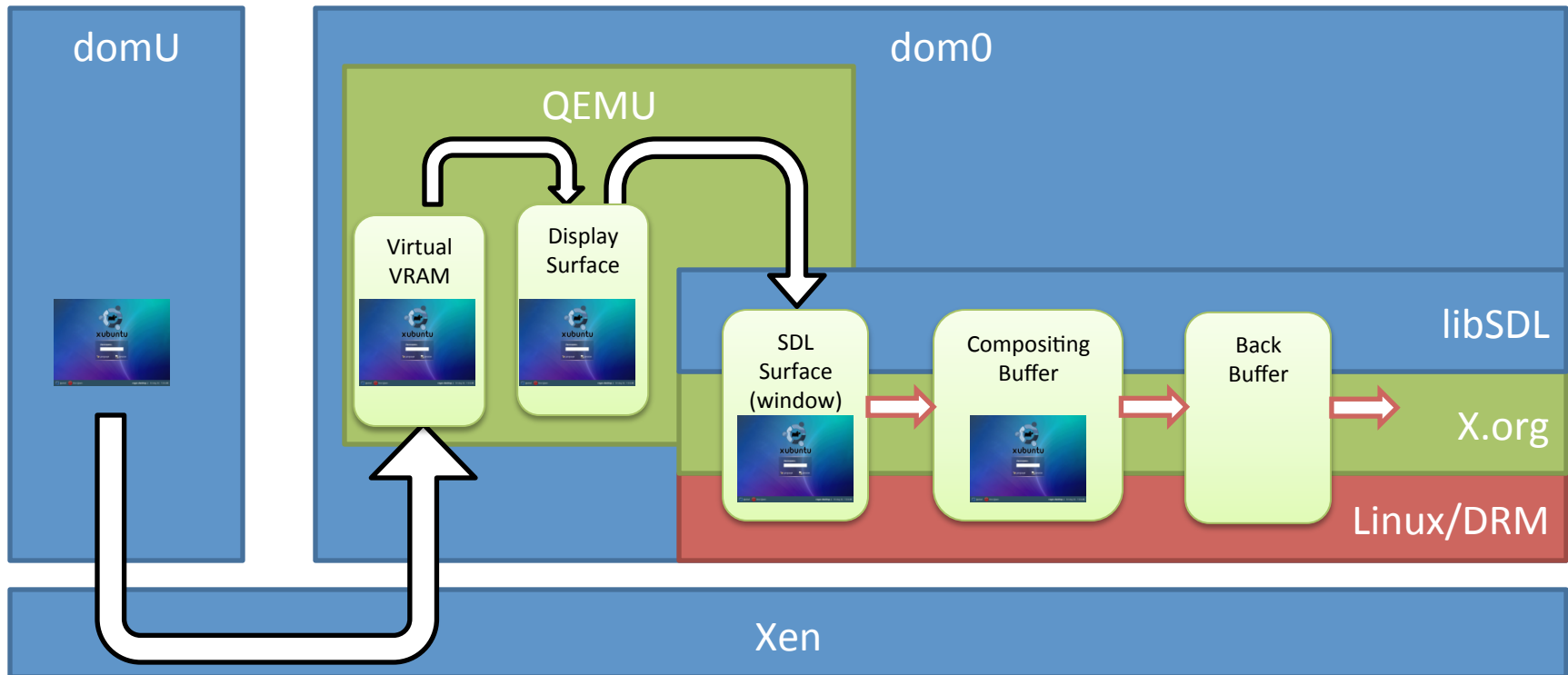
- Xen video basics
- Existing Optimizations
- GEM

Overview of QEMU Graphics (The quick version)



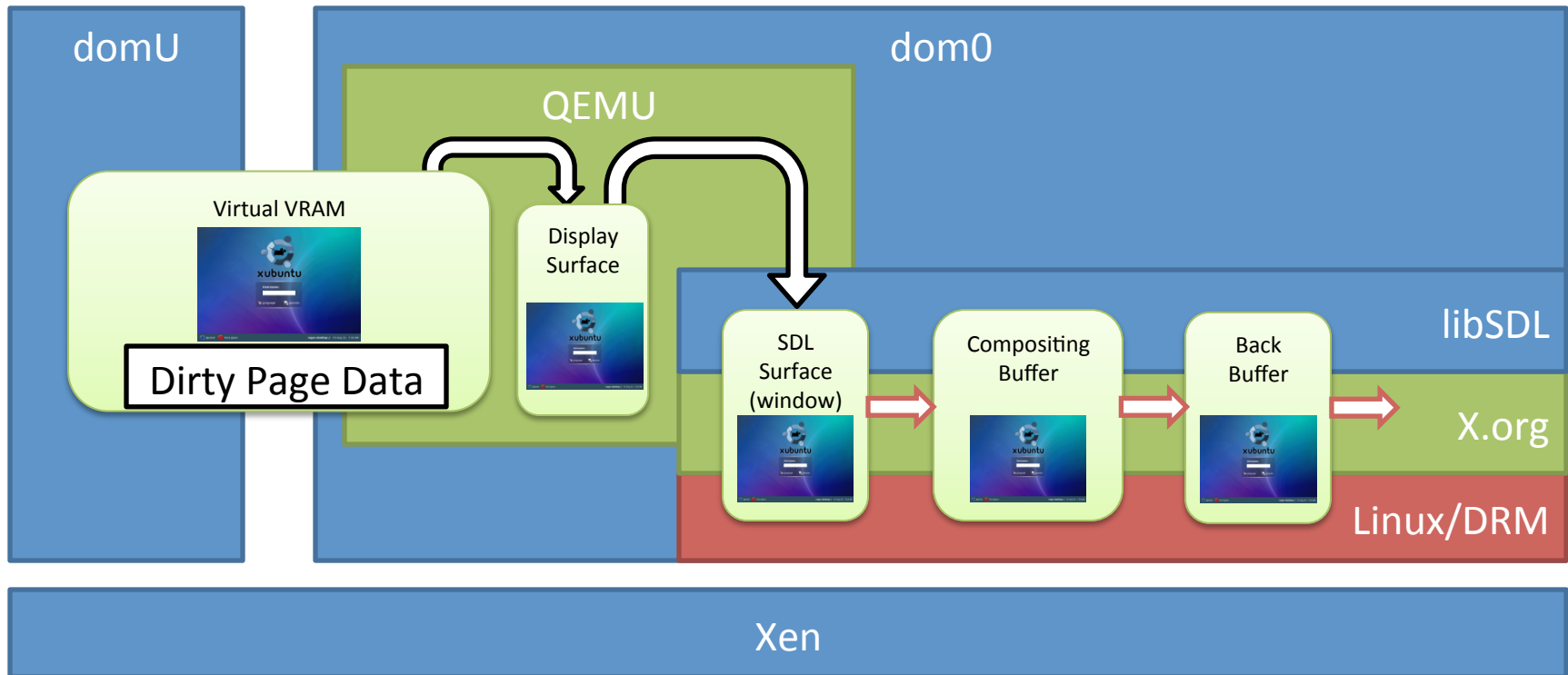
Worst Case Scenario

- Emulated VRAM



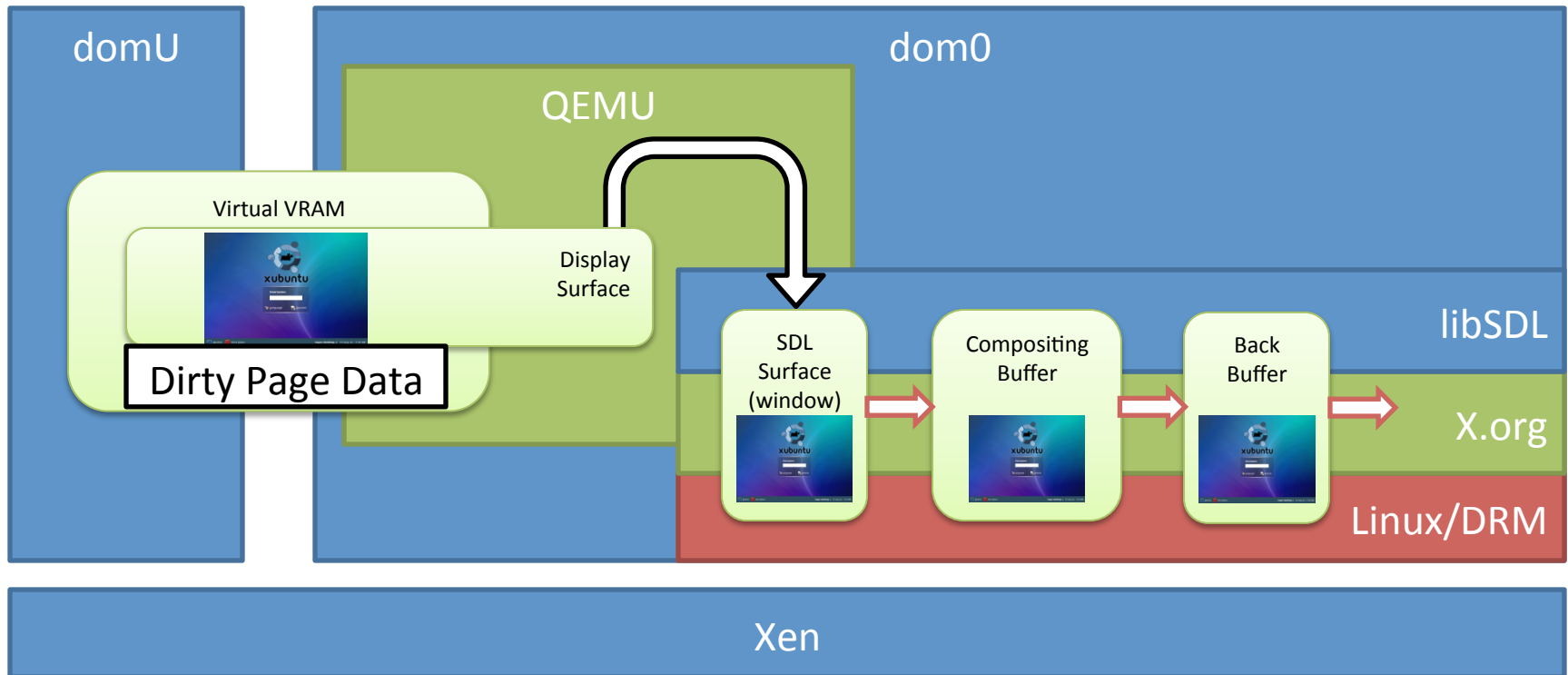
Existing Optimizations

- Foreign Page Mapping (Shared Memory)



Existing Optimizations

- Shared Buffer Mode + Foreign Page Mapping



Graphics Hardware

(The REALLY Short Version)

- Systems commonly have a unified memory architecture (UMA)
 - GPU is connected to the same memory bus as the CPU
 - Can scan directly out of system RAM
- UMA GPUs have their own virtual address space
 - Can access any domain's memory if the GPU's page table is appropriately programmed...
 - ...and... well, it's the REALLY short version

The Obvious Solution!

- Map the linear framebuffer into the GPU page table
- Program the CRTC base address so that the GPU scans out of the framebuffer
- Simple!

Unpleasant Reality

- GPUs are complicated
- GPU vendors are secretive
- GPUs change constantly

- You probably want to use your GPU for more than one thing at a time

GEM

- Linux already knows how to program your GPU
- The API is “standardized”

- ...but we want to tell it WHICH pages to use.

GEM Objects

- Basically a bunch of GPU specific state tracking and a scatter-gather list of pages
- Add an interface that fills in the scatter-gather list with the right pages
- Pass the object to KMS

- ... but which pages?

Foreign pages?

- Mapping the guest's foreign pages provides virtual addresses to the memory
- No machine addresses
- Not Linux memory, so no page structs to add to the scatter gather list

Grants?

- domU allocates a big grant table
- Creates a grantref for each page in the framebuffer
- Passes the table address and size to the device model
- Grant mappings get added to the m2p_override table, so we have page structs

- Requires a cooperative guest
- Uses a lot of grant refs
- Sets up mappings that never get used

Manual m2p_override

- Use the `translate_gpfn_list` hypercall to get a list of MFNs
- Use the `m2p_override` infrastructure from the grant code to redirect pages
- Use redirected pages in the scatter gather list
- No guest knowledge
- No unnecessary mappings

Foreign-backed GEM Objects

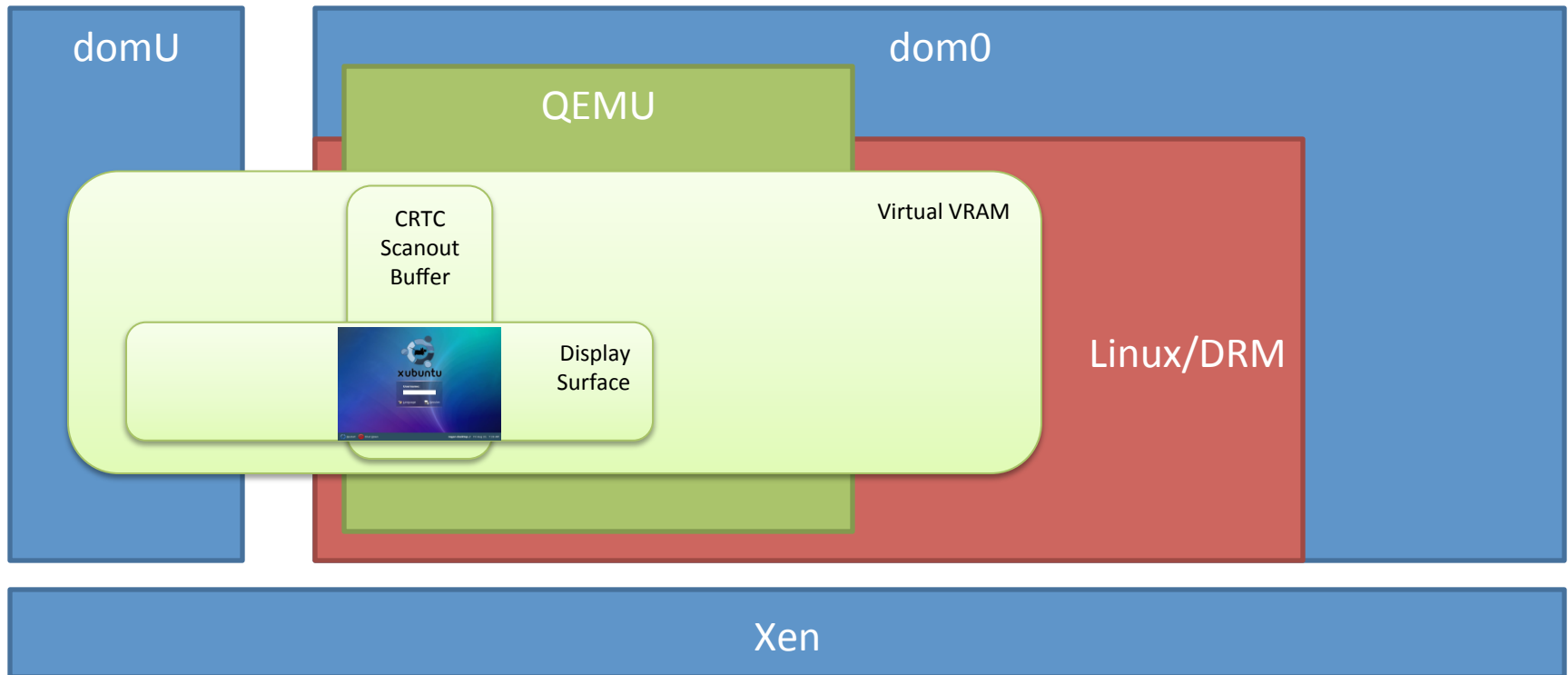
- Add new ioctl to create `i915_gem_foreign_object`
- Standard GEM object, but override `get_pages` and `put_pages` handlers using existing hooks
- Create `m2p_override` mappings for framebuffer pages
- Fill in the scatter-gather list with the overridden pages
- Pass the object to KMS
- Success!

New DRM ioctl

- (intel)
- I915_GEM_FOREIGN
 - In:
 - GFN
 - Size
 - Domain ID
 - Out:
 - GEM handle

Existing Optimizations

- GEM Object + libDRM UI



What to do With Your GEM Object

- Take over the scanout buffer of an entire vt
 - Lowest runtime overhead (none)
 - Requires source of input events (/dev/event/*)
- Convert it to a Prime object for use with DRI2/DRI3 X extensions
- Convert it to an EGL named image, and bind it to a texture
 - Hardware accelerated framebuffer tricks
 - Scaling
 - Rotation
 - Compositing
 - Effects
- Use it to display XenGT hardware accelerated framebuffers
 - Be sure to see Haitao Shan's XenGT presentation tomorrow...

Demo