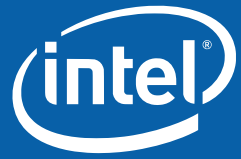


From Click to Pixel: A Tour of the Linux Graphics Stack

Carl Worth

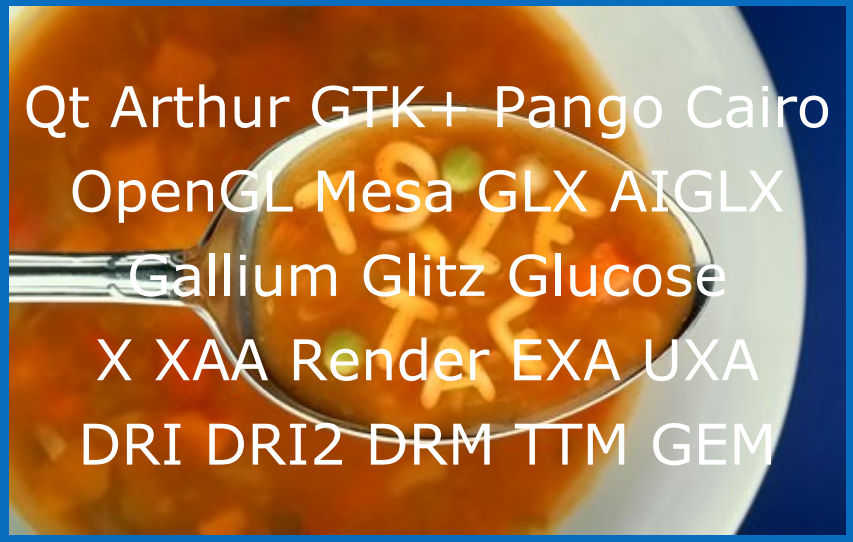
carl.d.worth@intel.com

Overview

- What the stacks looks like (2D and 3D)
- Tutorial: Inspecting rendering layer by layer
- What's changing now

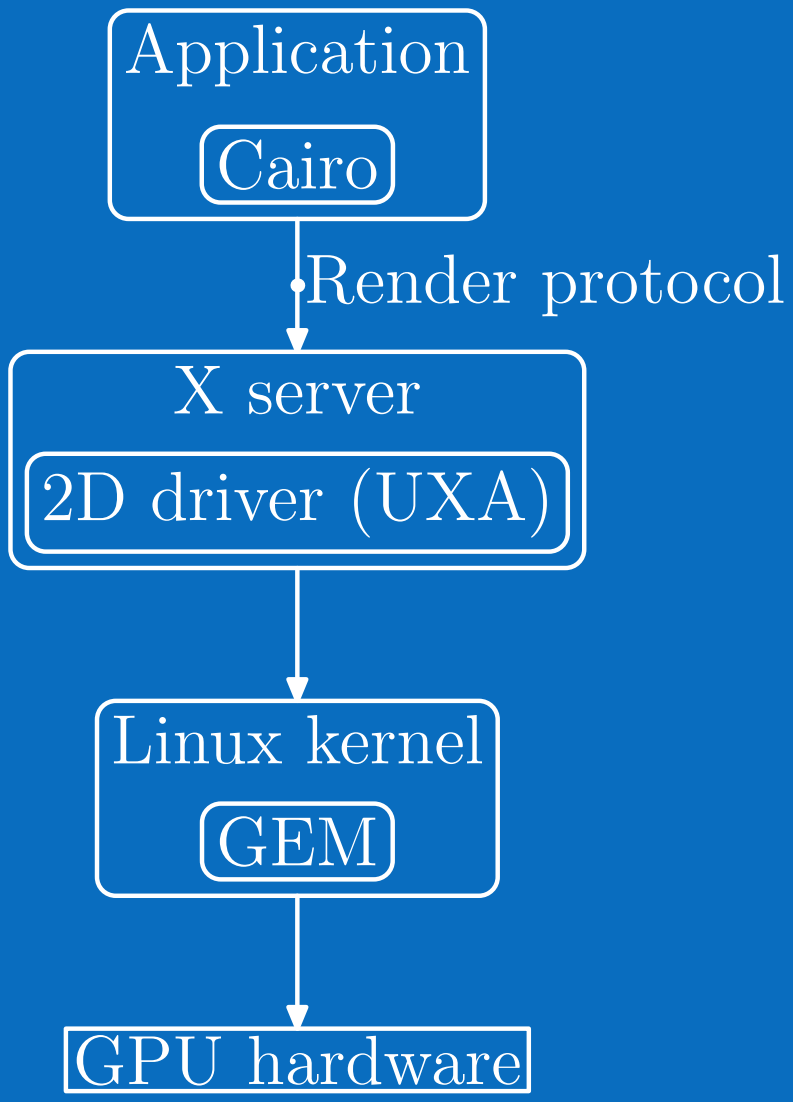
Alphabet Soup

Alphabet Soup

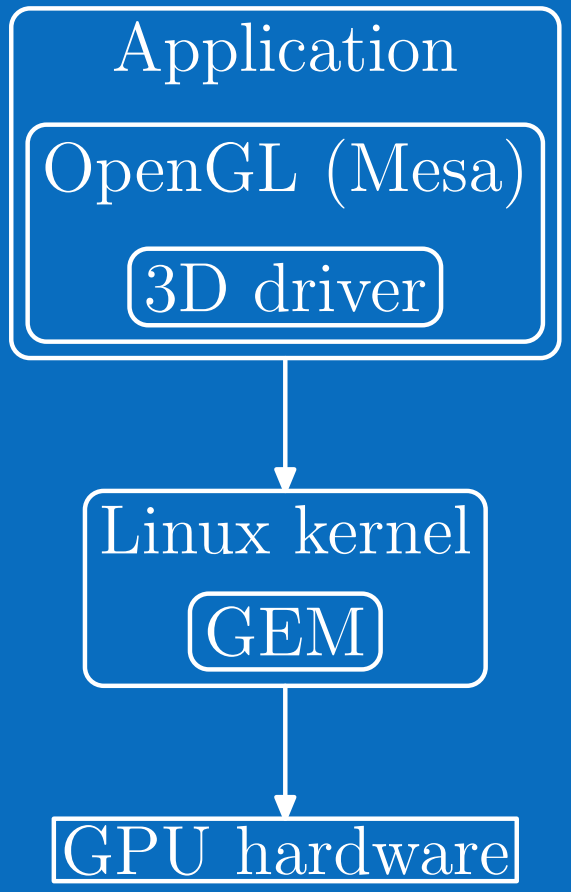


Stack Overview

2D Graphics Stack



3D Graphics Stack



Inspecting Layers

Profile first

- oprofile
- sysprof

Profile first

- oprofile
- sysprof

- Use them.

Profile first

- oprofile
- sysprof

- Use them. First.

Visually inspecting GTK+ updates

- Useful for visually identifying excessive redraws
- Does require recompilation of GTK+

- HOWTO:

```
./configure --enable-debug=yes # for GTK+
GTK_DEBUG=updates ./my-program
```

Tracing cairo calls

- Most useful for debugging application
- Enables robust capture of all rendering
- Trace makes an ideal test case for cairo community
- No modifications to application or cairo required
- HOWTO:

  Install cairo 1.9 or later

  cairo-trace ./my-program

  See results in my-program.$PID.trace

Inspecting Render protocol

- Much lower-level than cairo-trace
- No recompilation required

- HOWTO:

```
xtrace -D :5 > my-program.xtrace
DISPLAY=:5 my-program
```

Finding software fallbacks in EXA

- Very useful for identifying unexpected slowness
- HOWTO:
  - Edit xserver/exa/exa_priv.h:

    #define DEBUG_TRACE_FALL 1
  - Recompile xserver
  - Examine Xorg.0.log file

Finding software fallbacks in xf86-video-intel

- Very useful for identifying unexpected slowness
- HOWTO:
  - In "device" section of xorg.conf:

    Option "FallbackDebug" "true"
  - Examine Xorg.0.log file

Inspecting 3D state (for Intel)

- CPU is pegged?
  - Start with a profiler, then INTEL_DEBUG=fall
- CPU is idle?
  - Buffer management has gone wrong
- Chip is hanging?
  - State likely not getting re-emitted, INTEL_DEBUG=batch,sync
- HOWTO:

  INTEL_DEBUG=<comma-separated list of flags>

  fall: Show software fallbacks

  batch: Show decoded batchbuffers

  sync: Wait for idle after each batchbuffer

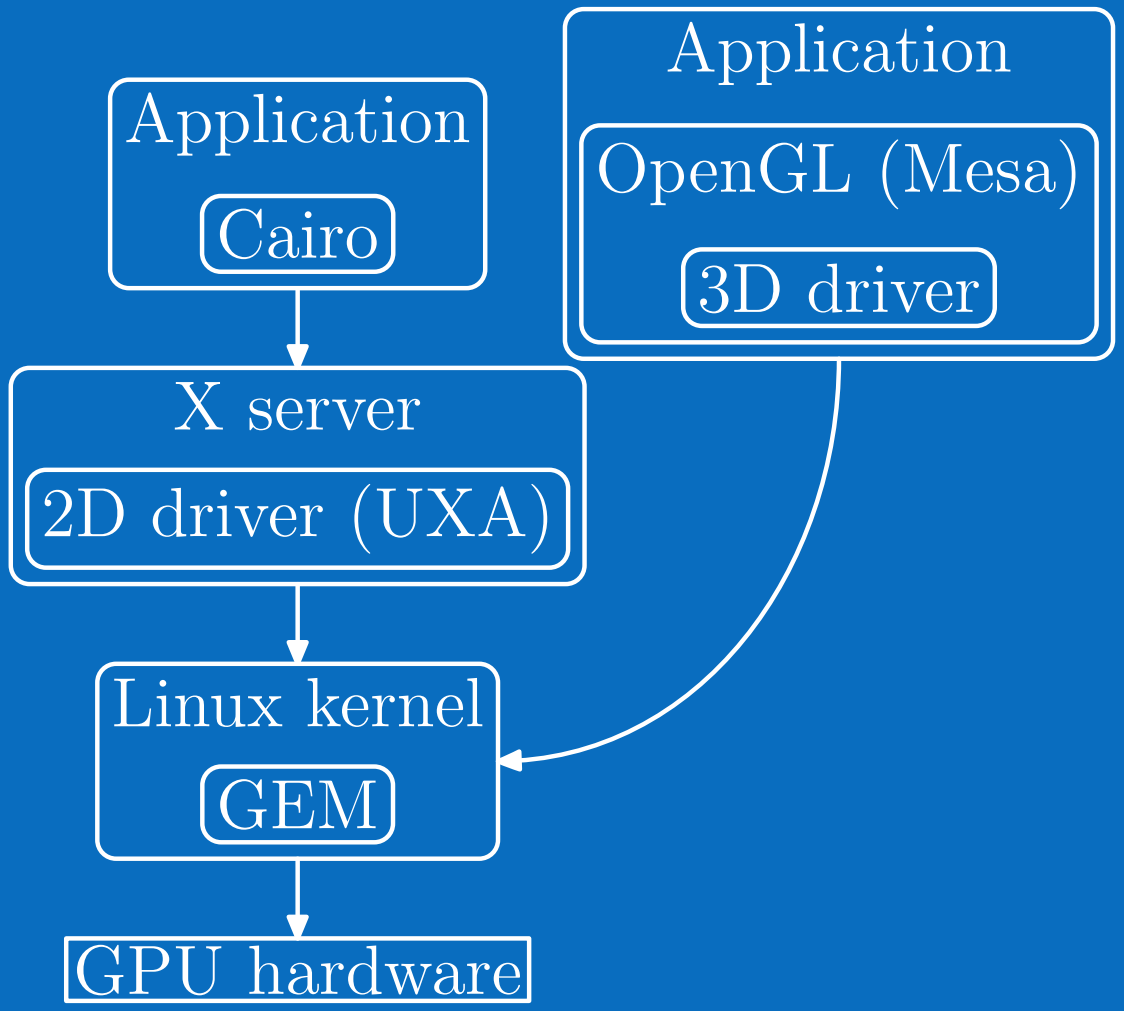
  (see intel_context.c debug_control[] for more)

Inspecting GEM state

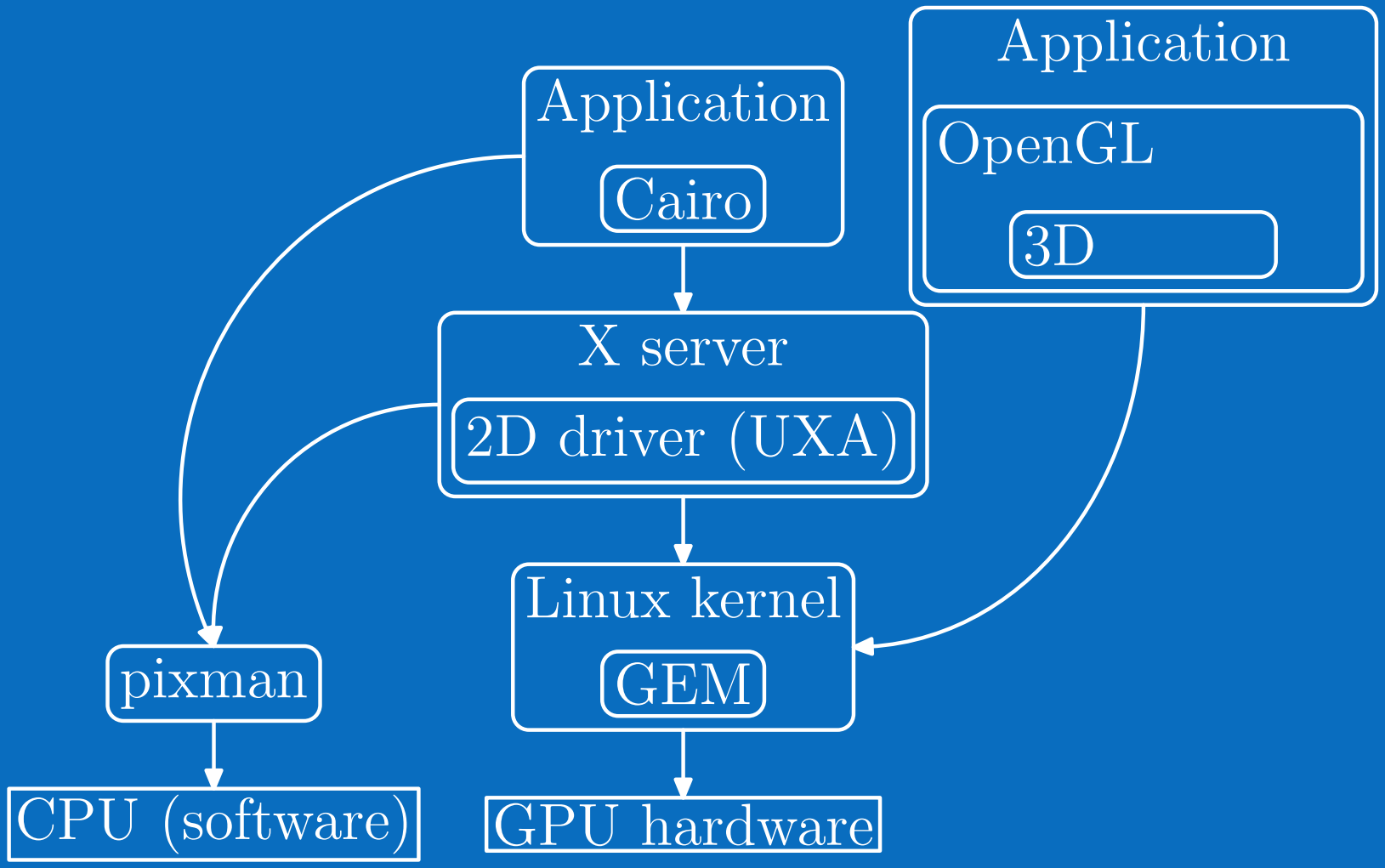
- Useful for debugging graphics driver
- HOWTO:
  - cat /proc/dri/0/gem_objects
  - cat /proc/dri/0/i915_gem_interrupt

Mixing things up

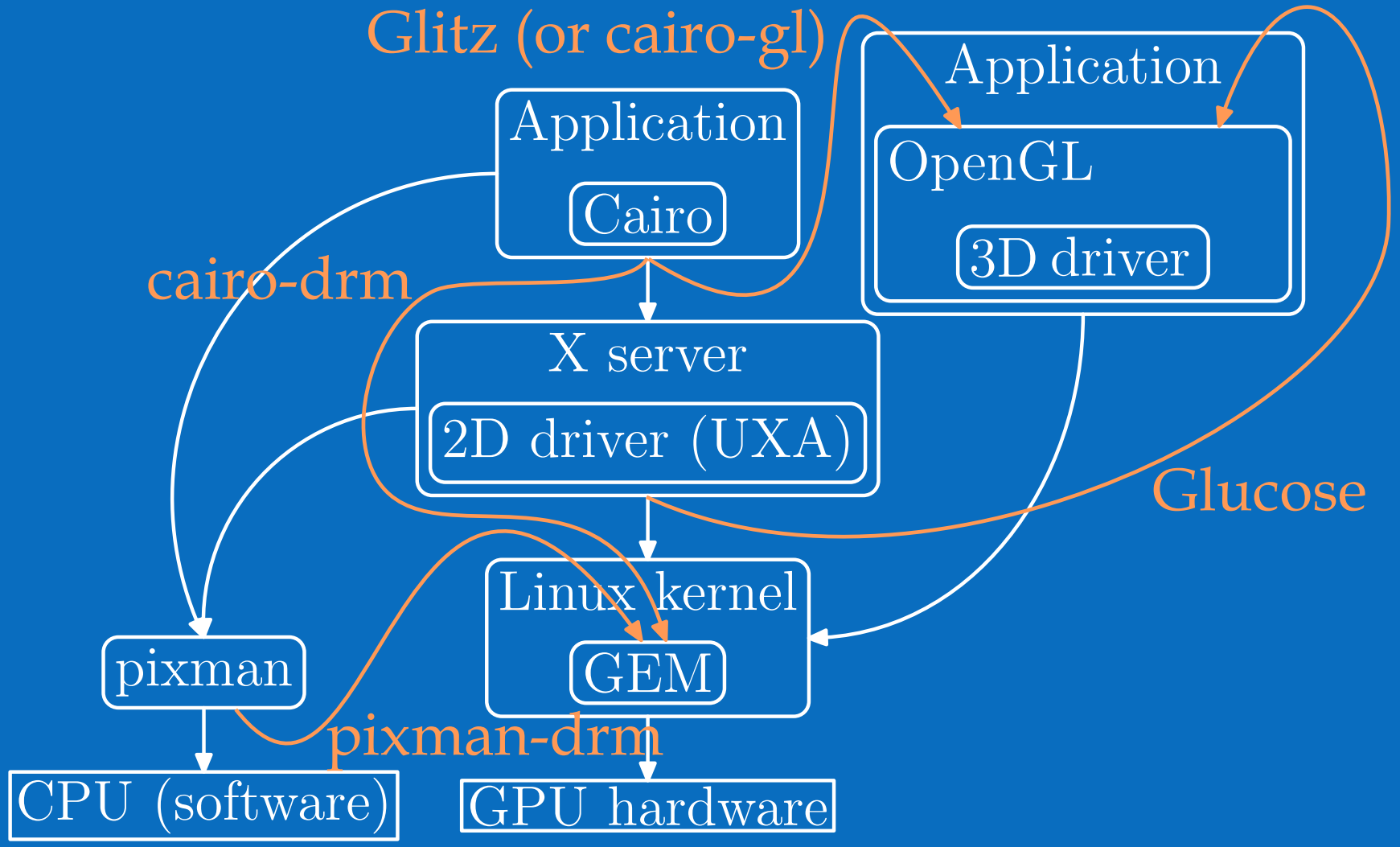
Combined stack



Software fallbacks



Current/Future changes



Everything through OpenGL?

- At which level?
  - Cairo? (see Glitz)
  - X server? (see Glucose)
- Potential problems
  - Quality concerns (overblown?)
  - Missing operations (just add extensions?)

Direct-rendering with cairo (cairo-drm)

- Chris Wilson has been playing around with this (on i915/i945)
- Results are already very promising
  - Gradients are 100 - 120x faster
  - Some painting operations are 50x faster (why?)
  - Text is 4x faster (needs more testing)
- Brings XVideo and XRender together (using video as source)
- Some of the work can be folded back to X server 2D driver
- Longer-term makes more sense in pixman-drm