

The Architecture of Your Next Linux Desktop: Xorg and XGL

Mat Caughron
Omaha Linux User Group - June 2006

Disclaimer

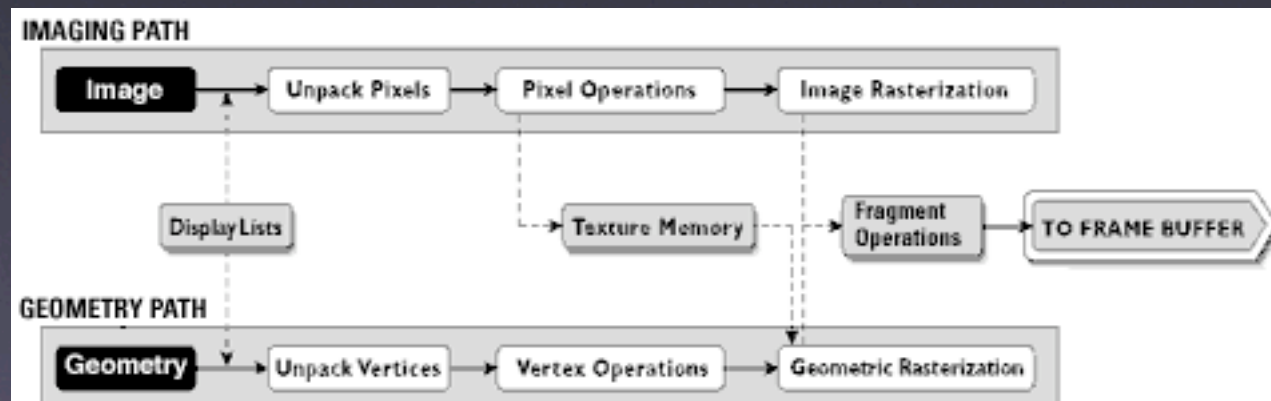
Parts of the following presentation were influenced by people I've never met in person, only on Usenet.

This Talk

- Background - OpenGL and X.org
- Desktop Architecture 202
- How to get Xgl working?
- Distro issues

OpenGL

- Industry standard (for real, not like from Redmond)
- SGI had a graphics language called “IRIS GL”
- The standard continues to evolve, see the Architecture Review Board at OpenGL.org



OpenGL History

- Kurt Akeley said he started OpenGL in 1989.
- first public release of the 1.0 spec was in 1992.
- OpenGL is a direct descendant of SGI's IRIS GL.
- replaced PEX, the PHIGS extension to X
- PHIGS held state and was pretty complex (3 tier)

SGI versus PEX

- SGI viewed PEX as its major competition at the time that OpenGL started.
- PEX was a consortium of big workstation vendors that wanted to elbow in on SGI's hold of the 3D market.
- Apparently SGI posted to the `comp.graphics.api.pex` newsgroup which formally stated why OpenGL was superior to PEX.
- SGI never saw what hit them. OpenGL rapidly became the standard to support because it was less tolerant of subsetting their API compatibility.
- Interesting fact: Now Google has their real estate in Mountain View.

Common Vendors

- nVidia
- ATI
- Matrox
- S3
- Intel

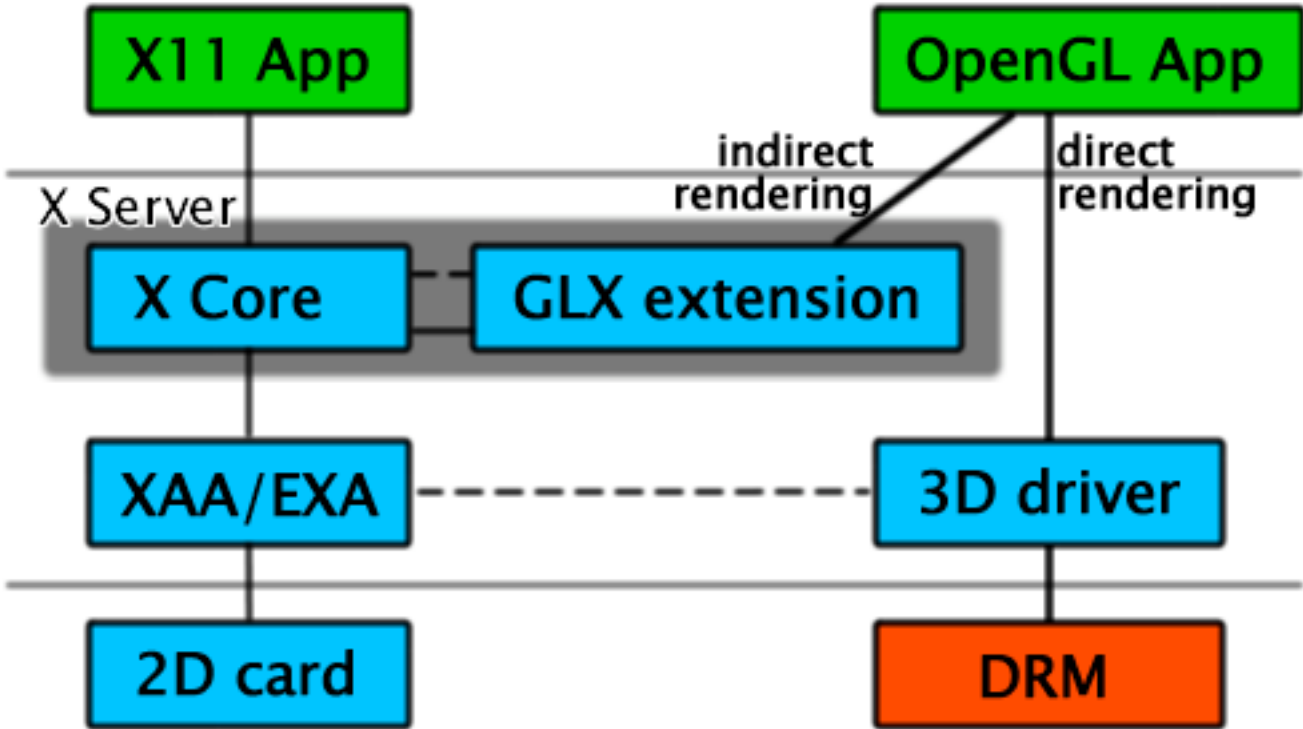
Supporting Distros

- OpenSuSE 10.1 / SuSE (SLED/SLES) 10.1
- Ubuntu - coming soon
- Gentoo, first adopter, as usual
- RedHat?

Mesa, a Software Graphics Lib

- The Mesa 3D graphics library
- API similar to commercial OpenGL
- mimics command syntax
- Brian Paul gives complete history at mesa3d.org
- Mesa 6.x implements OpenGL 1.5
- Mesa 7 to implement OpenGL 2.0 ?

X.org



Step 1:

- You launch Xgl.
- Xgl calls Xorg
- Xorg loads nVidia's libGlx
- Now Xgl can provide direct calls to

Step 2:

- Xgl is linked to nVidia's libGL
- As a client for OpenGL, Xgl needs context
- Direct context is given (straight to GPU)
- Xgl starts rendering from there

Step 3:

- Xgl ready to go: draws root window and interior windows.
- Xgl sets up its own libglx which provides extensions for X
- OpenGL clients can then use libglx (think Xscreensaver) OpenGL clients go through the same process as Xgl did earlier, asking for context for direct rendering access.

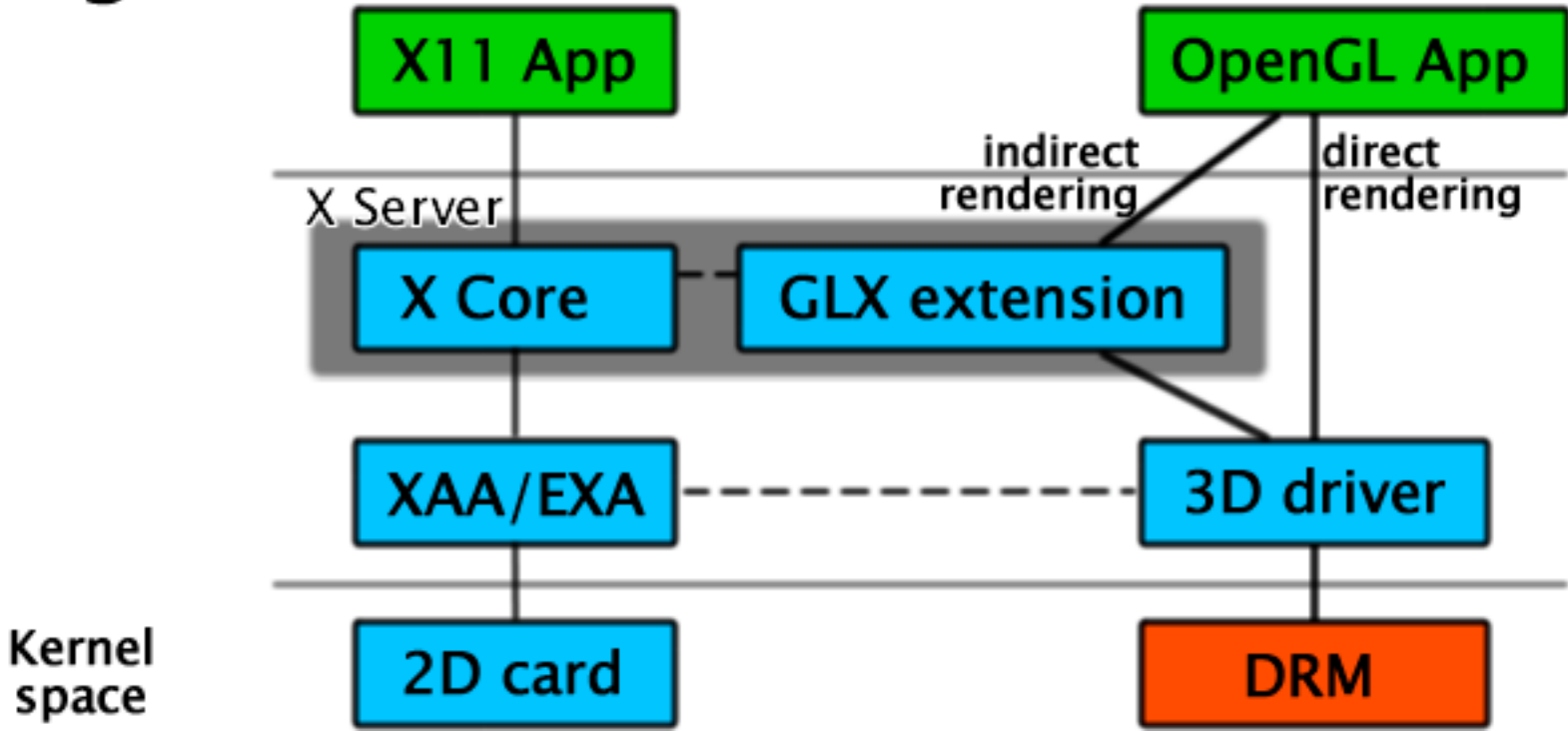
Example Client: compiz

- renders through Mesa libGL Why?
- `GLX_EXT_texture_from_pixmap`
- handles window management

nVidia (ATI) versus DRI

- nVidia provides its own OpenGL lib
- Mesa libGL and libglx are loaded for DRI

Aiglx



Xnest

- a client to the real server
- acts as a server to its clients
- and yes, it'll handle recursive sessions

Live Demo

- Will now switch to OpenSuSE 10.1
- If you want to try this at home: Kororaa 0.2 LiveCDs are available via BitTorrent

Kororaa - GPL Controversy

- Does inclusion of nVidia or ATI drivers cause a GPL violation?
- Drivers are kernel modules, not linked.
- Are distros aggregations, derived works?
- Discuss.
- More after the jump.

What's Next?

- Accelerated indirect GLX
- Stability and more interactivity features.
- multi-desktop browsing like FPS?
- rootless X.org implementation for Win32/
Vista similar to XDarwin for OSX?

Thank you for giving me the opportunity to
speak to you at OLUG.

“In order to make progress, one must leave
the door to the unknown ajar.”

Richard Feynman,

as quoted in *The Beat of a Different Drum*

Mat Caughron

(402) 968-1332

mat@caughron.com