



# Linux Graphics Meets the ARM Ecosystem

Jesse Barker <jesse.barker@linaro.org>

April 12, 2011

Find us [here](#) on the Linaro wiki



# Overview

- The Linux Desktop
- The ARM Linux Desktop
- The Subset Approach
- Examples
- Questions
- What's Next

# The Desktop

The screenshot displays a Linux desktop environment with the following elements:

- Terminal Window (Top Left):** Shows the execution of `glmark2 11.01`, which renders a 3D horse model. The terminal output includes:

```
OpenGL Information
GL_VENDOR:    VMware, Inc.
GL_RENDERER:  Gallium 0.4 on softpip
GL_VERSION:   OpenGL ES 2.0 Mesa 7.9

[Suite] Precompilation
[Benchmark] Vertex array
[Benchmark] Vertex buffer object
[Suite] Texture filtering

glmark2

jesse@jesse-desktop:~$ /home/jesse/opt/gl
libEGL warning: use software fallback

glmark2 11.01

OpenGL Information
GL_VENDOR:    VMware, Inc.
GL_RENDERER:  Gallium 0.4 on softpip
GL_VERSION:   OpenGL ES 2.0 Mesa 7.9

[Suite] Precompilation
[Benchmark] Vertex array
```
- Terminal Window (Bottom Left):** Shows the same `glmark2 11.01` command and output as the top window.
- Web Browser (Top Right):** Displays the Linaro website at `https://wiki.linaro.org/Releases/Schedule`. The page features a search bar, navigation links for "Downloads" and "Contact Us", and a section titled "Open engineering" with the text "Discover what we are working on".
- Terminal Window (Bottom Center):** Shows a terminal window with a 3D rendering of a horse model, similar to the one in the top-left window.
- Terminal Window (Bottom Right):** Displays a terminal window with a 3D rendering of a horse model, similar to the one in the top-left window.



# Window system

- Display management
- Resource management
- Session management
- Event handling
- Application programming interface



Protocol Decode

Device Independent X (DIX)

DRI

EXA

DDX

libdrm

evdev

DRM | KMS  
GEM/TTM

Input H/W

CPU

GPU

DC

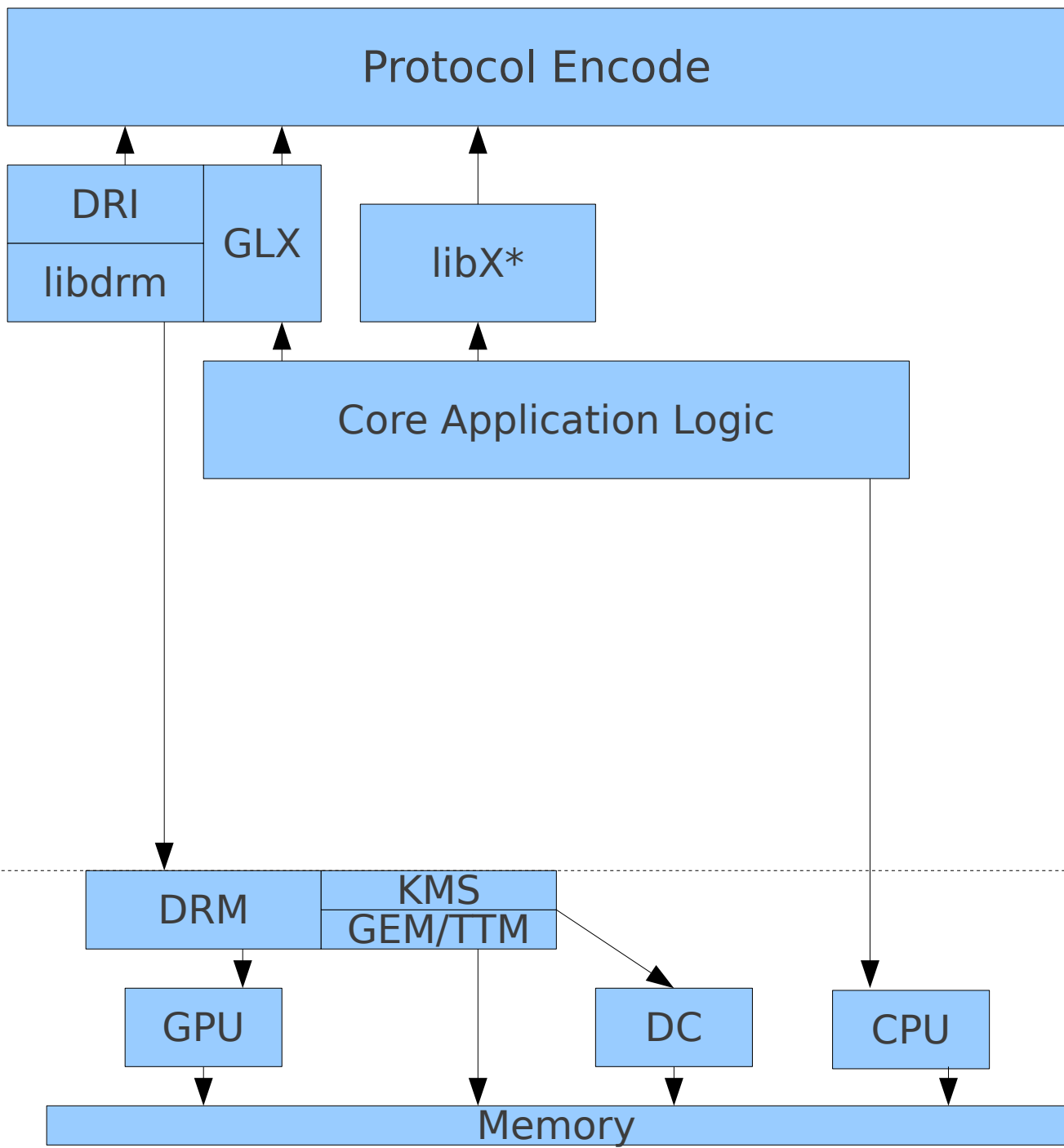
Memory

User space

Kernel space

Linaro





User space

Kernel space





# Toolkits/Frameworks

- Create abstraction layer from the underlying window system.
- Provide uniform look-and-feel across platforms.
- Applications don't have to care which system they are running on.
- New backend to the framework adds a new supported platform for a whole bundle of applications.

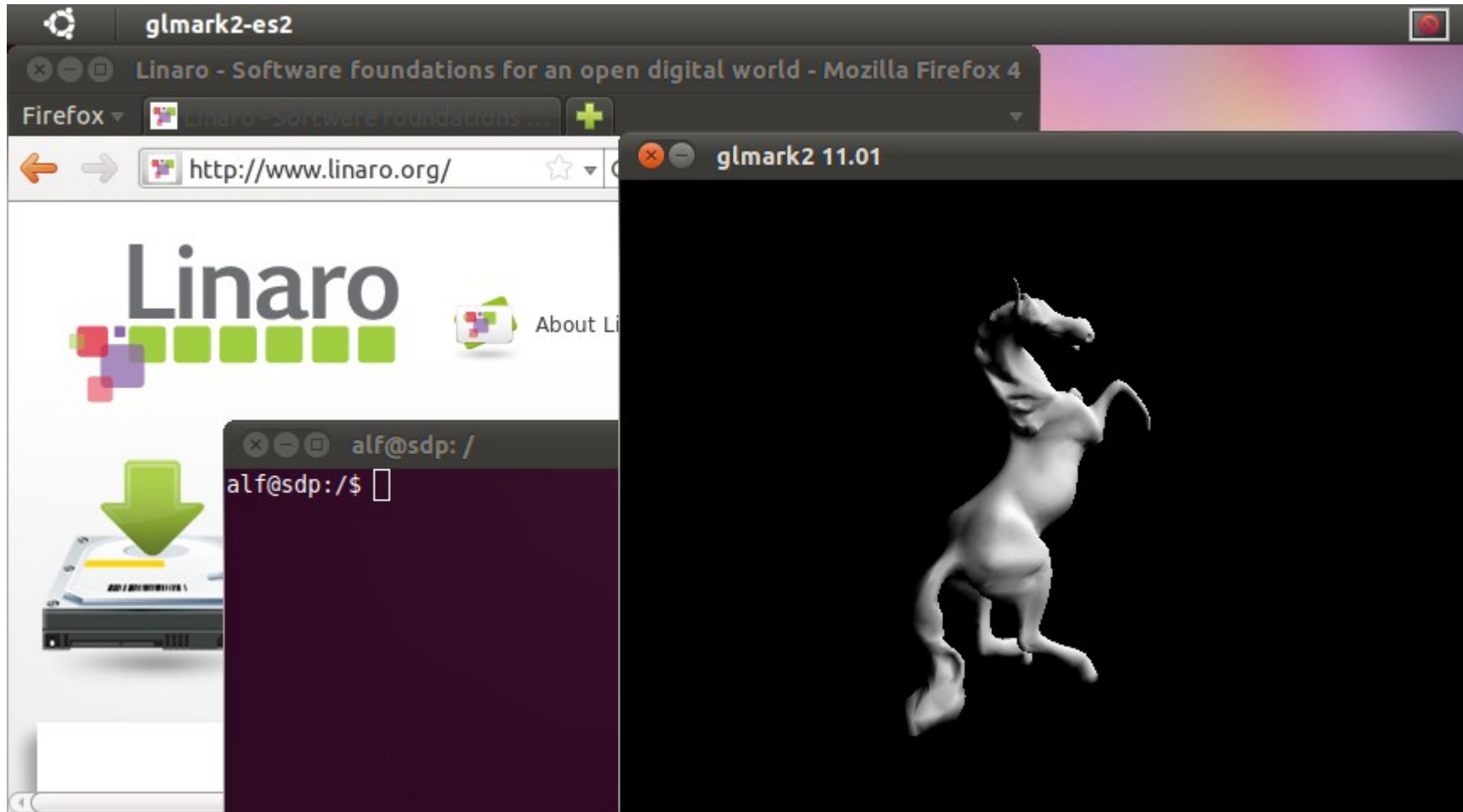


# Bells and Whistles

- OpenGL
- Video
- Audio
- Compositing window managers
- Animation



# The ARM Desktop





# What's the difference?

- Most differences are “physical”
  - Screen size and resolution
  - Unified memory pool
  - Power vs. raw performance
- Some API (not necessarily, though)
  - Window system interfaces
  - Rendering interfaces



# The Subset Approach

- OpenGL ES 2.0 is explicitly defined as a subset of OpenGL 2.1.
- Both have diverged since the original definition.
- Minimize specialized code (e.g., window system interfaces).



# The “big-ticket” items

- Immediate mode
- Fixed-function vertex processing
- Fixed-function fragment processing
- EGL vs. GLX



# Examples

- glmark2
- cairo-gles
- compiz



# glmark2

- Based upon opensource glmark by Ben Smith.
- Uses 3D Studio Max for model content.
- Uses SDL for window system abstraction.
- Only real differences between OpenGL and OpenGL ES is in initialization of SDL objects.



# cairo

- Cairo is a 2D vector graphics library.
- Used by GTK, Webkit, Mozilla, and others.
- Leverages the desktop OpenGL backend.
- Highlights differences between OpenGL and OpenGL ES.



# compiz

- Popular compositing window manager
- Uses an extensible architecture to amend functionality.
- Core handles context and resource management as well as event handling.
- Plugins handle bells and whistles.





Questions?



# What's next?

- Android
- Wayland
- ???



Thank You