
DRI / GL Acceleration 、 Xgl 、 AIGLX 及相關技術發展



Jim Huang (黃敬群 /jserv)

<http://blog.linux.org.tw/jserv/>

May 23, 2006

- User Interface 概況
- 爲何要 3D ？
- 快速釐清觀念
- Xgl 、 AIGLX 與相關技術探討
- 3D ：究竟是 Eye-candy 還是未來？

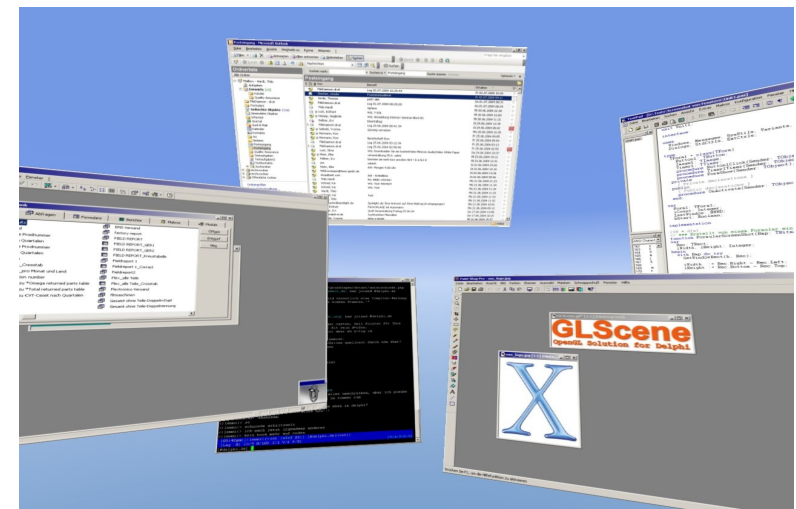


User Interface 概況 (1)

- ▶ 三個世代
 - 批次處理 (mainframe)
 - CLI (Command Line Interface)
 - GUI (Graphical User Interface)
- ▶ 趨勢
 - 變遷並非漸進
 - 革命性突破

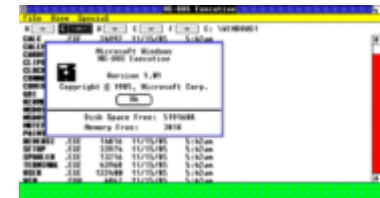
```
Menu 24 - System Maintenance
1. System Status
2. System Information and Console Port Speed
3. Log and Trace
4. Diagnostic
5. Backup Configuration
6. Restore Configuration
7. Upload Firmware
8. Command Interpreter Mode
9. Call Control
10. Time and Date Setting
11. Remote Management Setup

Enter Menu Selection Number: 8
```



User Interface 概況 (2)

- 1972 Xerox PARC
 - Alto
- 1981 Xerox 8180 (Star)
 - 商業化的 Alto
- **WIMP**
 - **Windows**
 - **Icons**
 - **Menus**
 - **Pointers**
- 1983 Apple Lisa
- 1984 MIT X Window System
- 1985 Commodore Amiga
- 1985 Microsoft Windows 1.0



User Interface 概況 (3)

- WIMP 一般來說 ...
 - 標準
 - 延續人類視覺體驗
 - 圖形辨識
- 不足處
 - Not for the disabled
 - 無法大量應用多重感知系統
 - 過度強調任務導向，缺乏一般性
 - 變化性低
 - ...

User Interface 概況 (4)

➤ 邁入後 WIMP 時代

- 廣泛應用多項語音多媒體技術

- 語音辨識
- Headless
- Gesture
- 環繞式介面
- 肢體語言 (Nintendo Wii)
- 3D
- ...

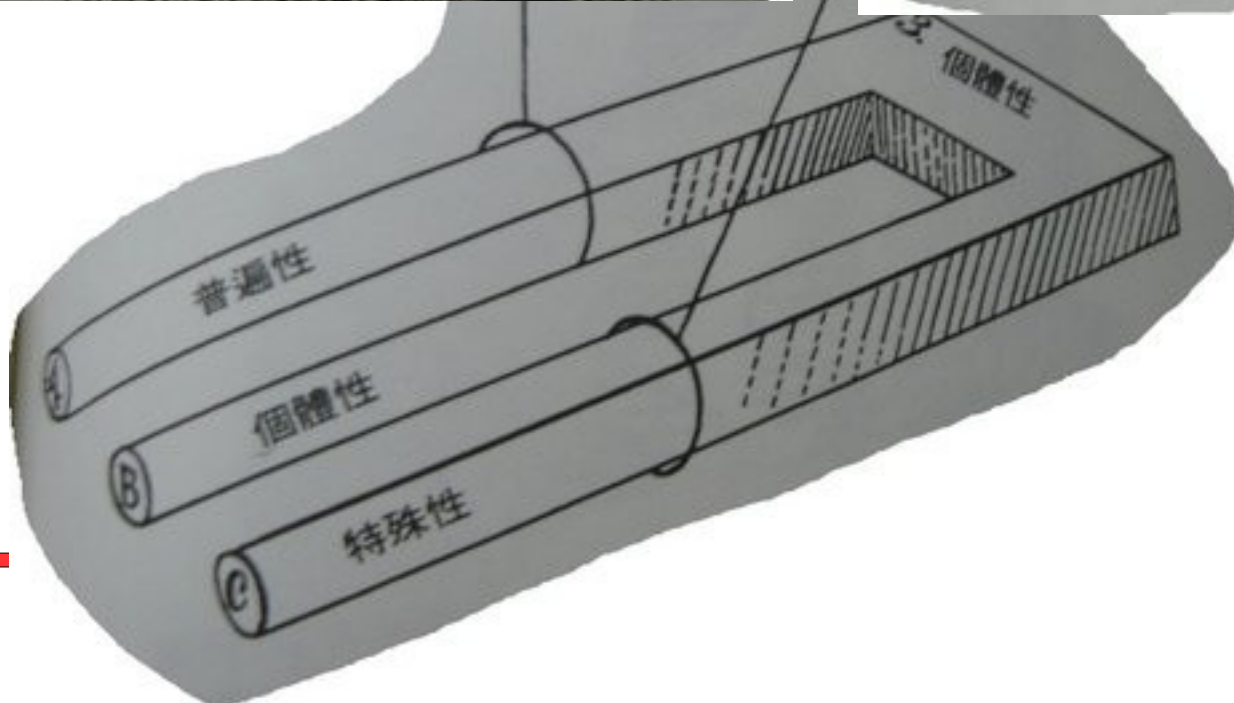
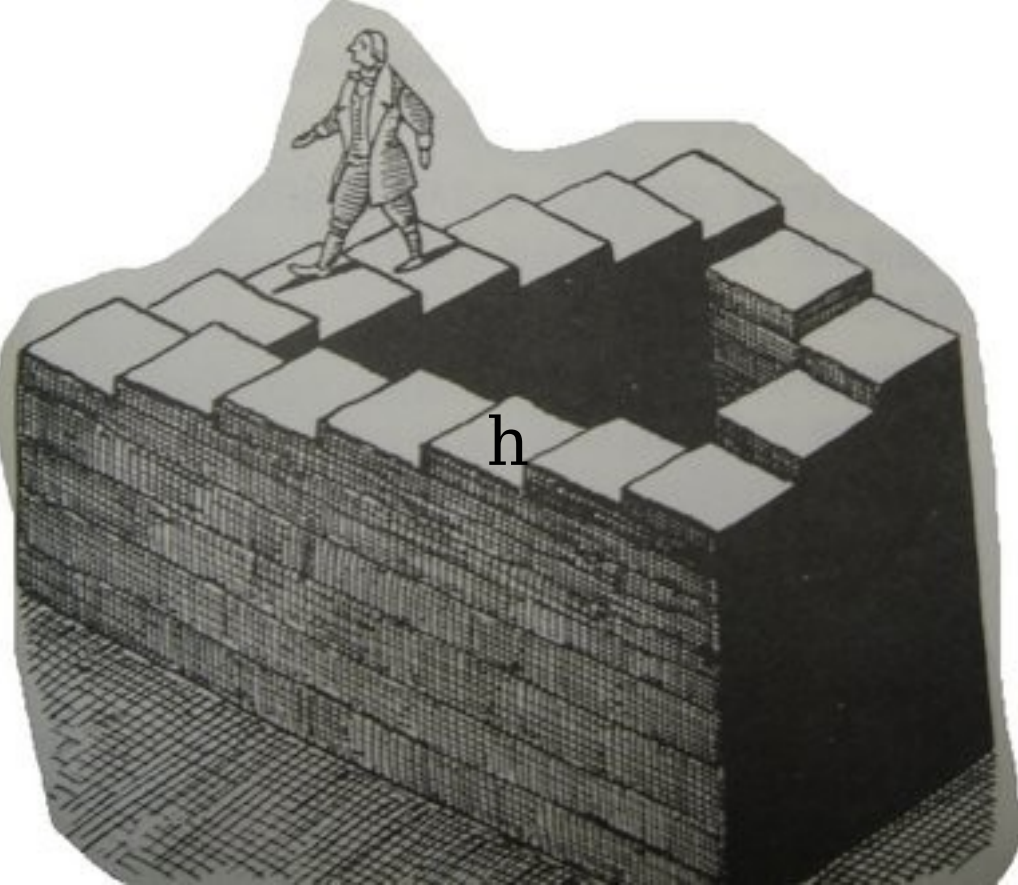


為何要 3D ？

- 只是酷炫？
- 客觀來說
 - 貼近真實體驗
 - 後 WIMP 時代的技術基礎之一
 - 允許大量的感知系統引入
 - 更廣泛的克服 2D WIMP 的技術議題
 - 設計、製模
 - 視覺化
 - 互動式體驗
 - 模擬

3D User Interface 的挑戰 (1)

- 回顧：User Interface 的演變是革命性的突破
- 在挑戰之前 ==> 視錯覺 (optical illusion)
 - 2.5D 的基礎
 - 產生錯覺的原因在生理學上叫視覺位移，又叫視差
 - 用兩只眼睛同時觀察一個物體時，物體上每一點對兩只眼睛都有一個張角
 - 物體離雙眼越近，其上的每一點對雙眼的張角越大，視差也就越大，即會出現「眼見為虛」的現象



3D User Interface 的挑戰 (2)

- 缺乏標準
- 建構於「視錯覺」的假設下
- 缺乏硬體設備的支持（輸入、輸出）
- 缺乏一致性的 3D Toolkit
- 與既有 2D 環境的相容性
- 網路集中性環境
- Session Management

自由軟體架構下的 3D 解決方案

- The past
 - gone / omitted, only for concept-proving
- 以 X11 為基礎的架構
 - 傳統的 client-server 架構已經大幅翻修
 - OpenGL 與 GLX
 - 3D 加速的實現
- 其他：DirectFB/GL 、 VirtualGL 、 ...

以 X11 為基礎的架構

➤ 歷史背景

- 1984 年 MIT 在 DEC 的贊助下，進行 Athena 計畫，為 X Window System 的前身
- 1987 年第 11 版的 X Window System 為重要里程碑
- 2003 到 2004 年間 X.org 計畫重組，各式新技術紛紛引入，例如 Sun Microsystems 的 Project Looking Glass

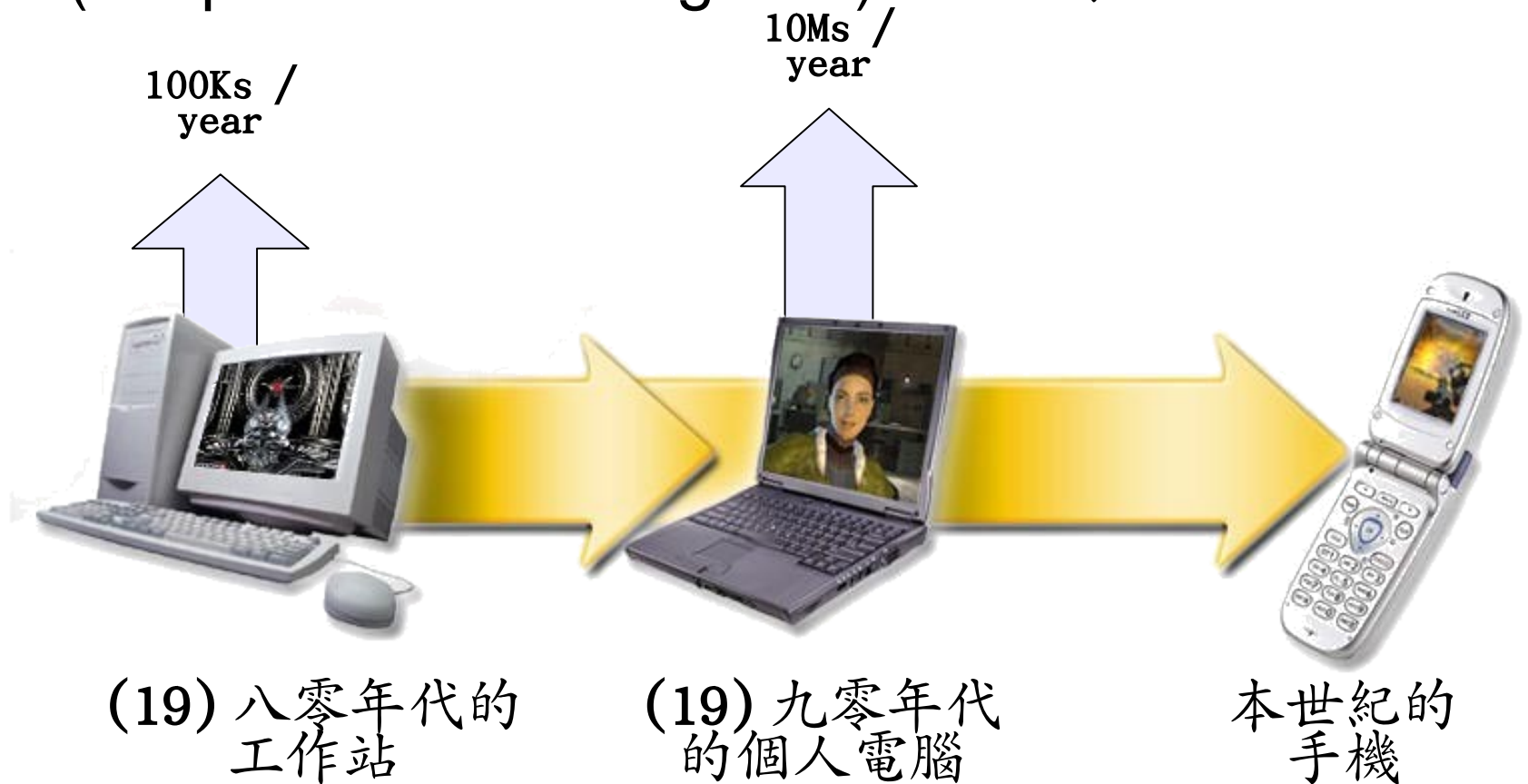
➤ 支援平台廣泛

- *BSD / Linux / Darwin / MINIX / Win32 / ...



3D 與 OpenGL

- ▶ GL = OpenGL = 工業 Graphics 標準
- ▶ GPU (Graphics Processing Unit) 的衝擊



OpenGL 解決方案

- Mesa
- Utah-GLX
- XFree86 4.x/DRI
- NVIDIA
- Xi Graphics
- SGI (Sample Implementation)

所謂「官方」的 OpenGL

➤ 官方 ==> OpenGL 認證

- 購買使用 Trademark 的權利
- 通過大部份的 conformance tests 項目

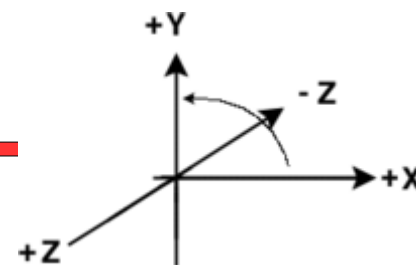
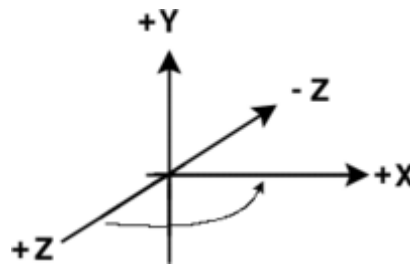
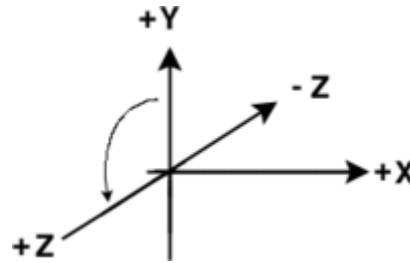


➤ Mesa

- Free/Open Source Software ，歷史悠久
 - 一開始是 software-only 的解決方案 (swraster)
 - 近來引入若干硬體加速實做
- 目前已通過 OpenGL 1.3 spec 大部分的 conformance tests

OpenGL (2)

- glBegin(GL_TRIANGLES);
 - glColor3f(1, 0, 0); /* (R,G,B) */
 - glVertex3f(-1, 1, 0);
 - glColor3f(1, 0, 0);
 - glVertex3f(-1, -1, 0);
 - glColor3f(1, 0, 0);
 - glVertex3f(1, -1, 0);



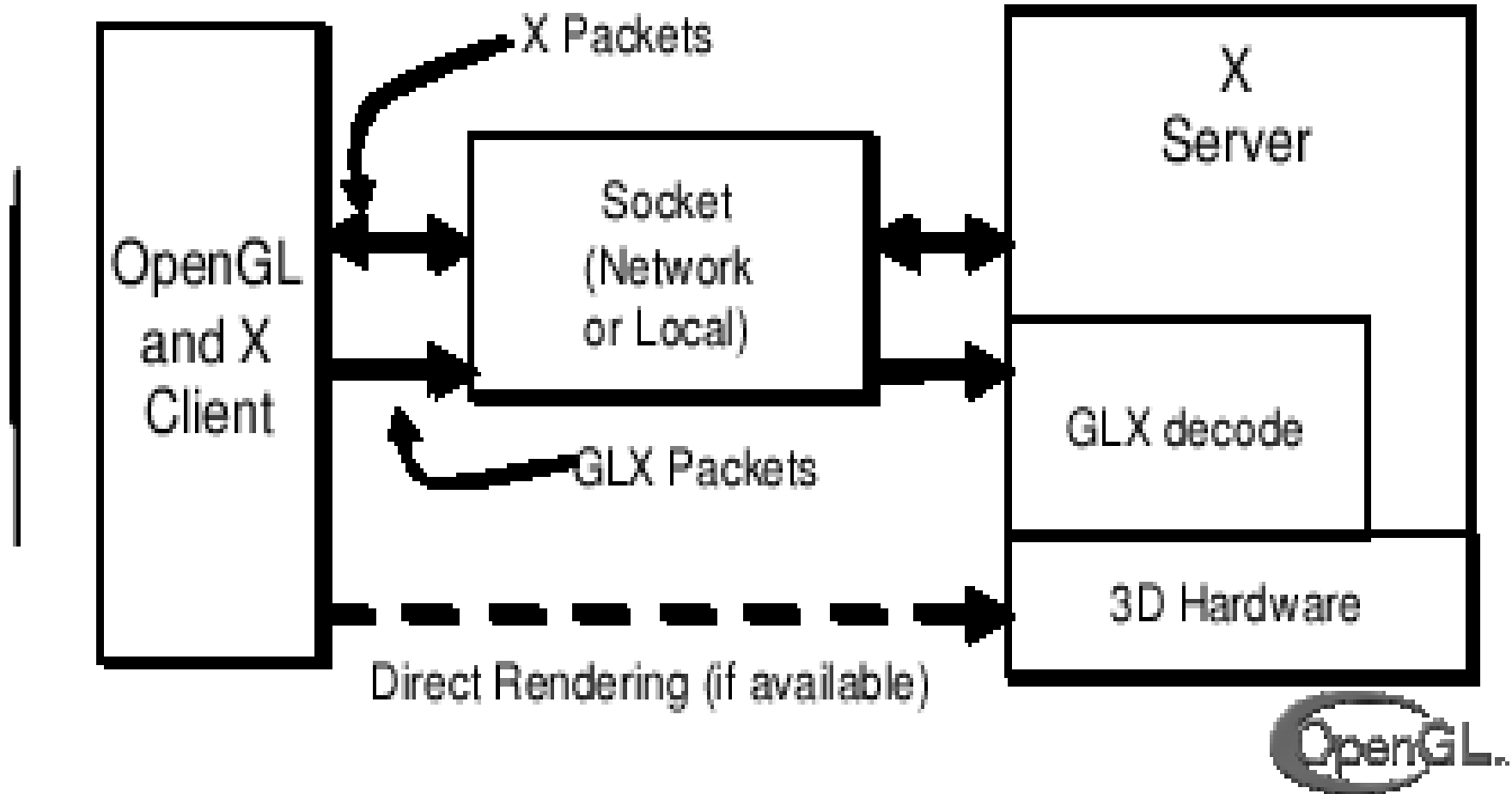
Right-handed coordinate system
(used in OpenGL)

- glEnd();

OpenGL (3)

- OpenGL 並沒有限制於特定的 Window System 上
- OpenGL 在 X Window System 的 API 實現：GLX
 - OpenGL rendering 時可挑選合適的 visual format / attribute
 - 將 OpenGL context 對應於特定 X11 Drawable
 - off-screen Rendering
 - rendering 到 Pixmap
 - OpenGL 1.3 有更多此類的 extensions

GLX 概觀



Mesa Software Rendering

- ▶ 全部實做於 client-side ，架構於 xlib 上
- ▶ 允許在所有的模式作 rendering
- ▶ 可於 remote host 作 rendering
 - 不需 GLX
- ▶ Mesa 在此模式可展現 OpenGL 所有的實做項目
- ▶ 低速的 raster-ization

Mesa Hardware Acceleration

- Mesa + Glide ==> 3dfx 硬體
 - single-context / full-screen
 - Linux 上首次的硬體加速繪圖機制
- Utah-GLX ==> XFree86 3.3.6
 - Utah 大學的 GLX protocol 實做
 - 非常受限的 direct rendering
- DRI ==> XFree86 4.x
 - mesa-embedded-branch ==> DirectFB

DRI 的軟體元件

- kernel mode
 - DMA 、 AGP memory ， 以及 MMIO
- 2D XFree86 driver
 - 傳統的 2D X Window System
- 3D DRI driver
 - 3D 硬體的支援
- libGL.so
 - 處理 GLX protocol
 - 載入 DRI driver
- DRI extension
 - 處理 3D 繪圖所需資源管理與通訊
- GLX extension
 - server-side GLX protocol handling / rendering

DRI 的軟體元件 (2)

➤ DRI includes Mesa

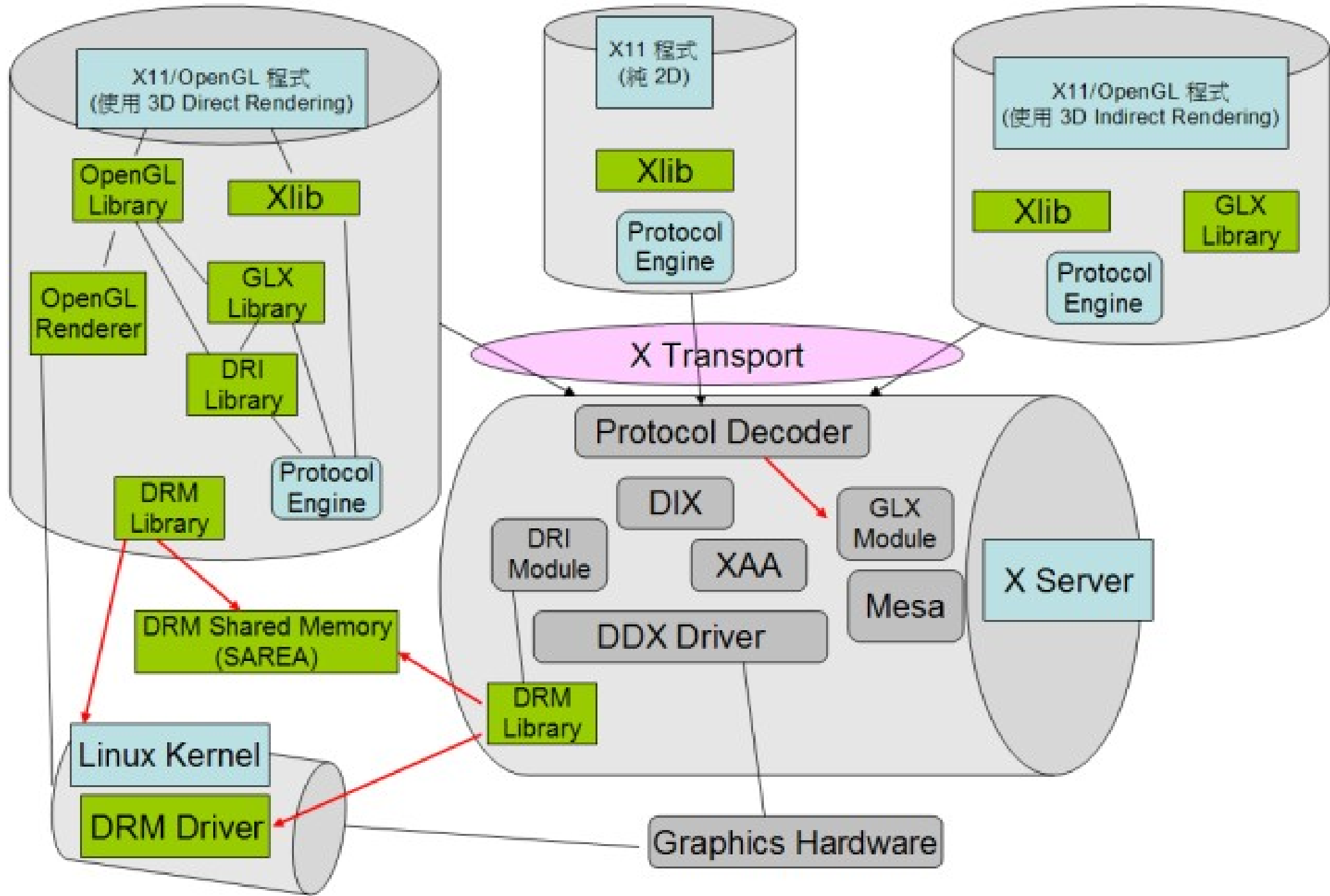
➤ libGL

– from **Mesa**

- 虛擬的 GLX 實做
- 可用於任何 X server 實做

– from **XFree86/DRI**

- 真實的 GLX 實做
- GLX protocol encoder
- 動態載入 3D 硬體 driver



GL 相關名詞解釋 (1)

- GLX extension (libglx)
 - 在 X server 中透過 GLX protocol 對應用程式提供 OpenGL context 服務，保留給硬體廠商實做或 Xorg 自行處理
 - NVIDIA closed source driver 即處理此動作，直接對應到硬體加速呼叫
- libGL
 - 由 X client 所呼叫，可透過 GLX protocol 與 GLX extension

GL 相關名詞解釋 (2)

- Direct OpenGL Context
 - **Direct Rendered** 的 context
 - libGL 可直接與控制硬體的 3D driver 溝通，而不必透過 GLX extension
- Indirect OpenGL Context
 - 必須透過 GLX extension
- Accelerated Rendering
 - CPU(Mesa) → GPU/Hardware

GL 相關名詞解釋 (3)

➤ DRI (Direct Rendering Infrastructure)

- OpenGL Context 的軟體集合

• libGL (透過 Mesa ==> swraster)

• libglx (透過 Mesa)

• 驅動程式的集合

➤ DRM (Direct Rendering Manager)

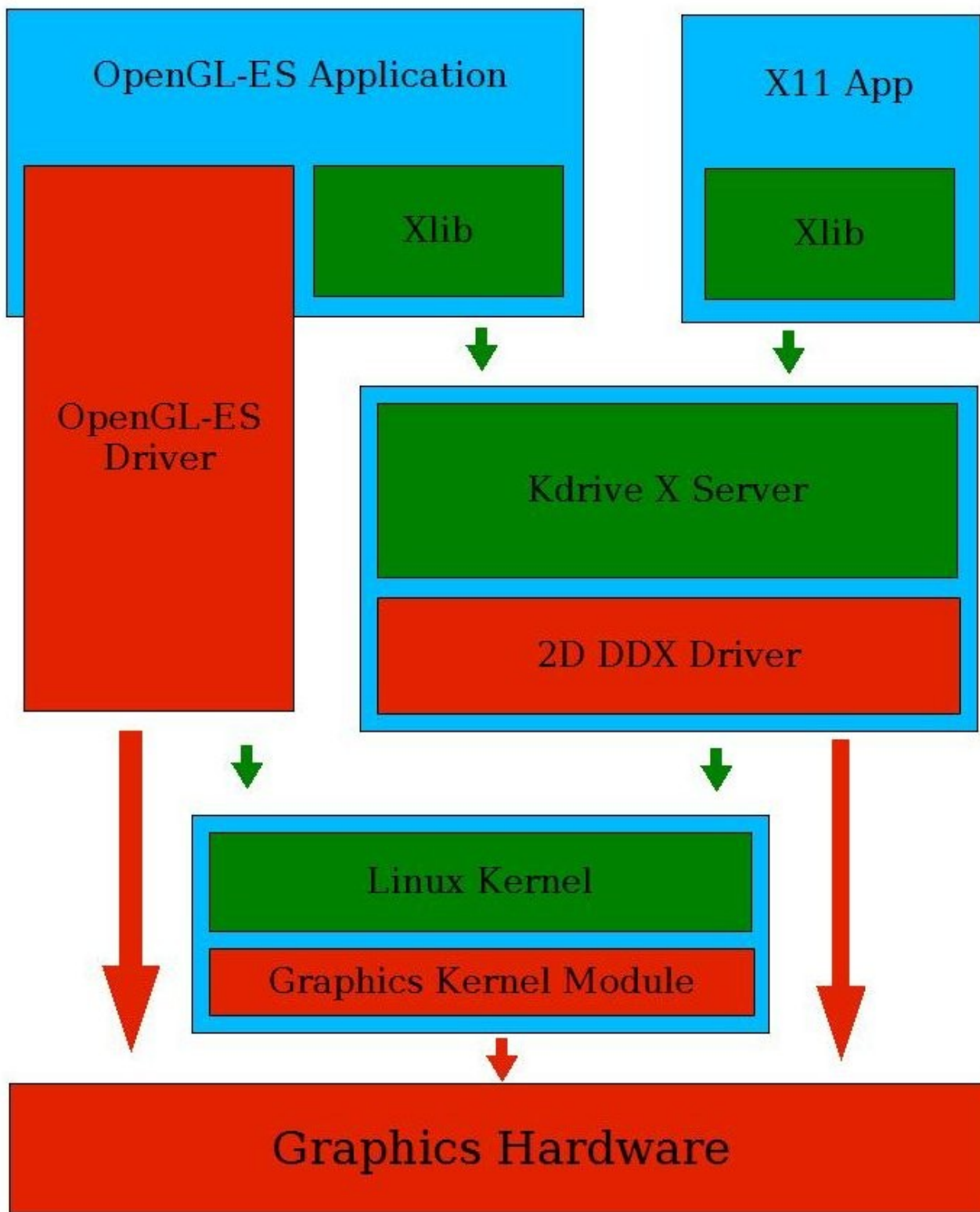
- 本質是 kernel module (char ==> drm)

- 管理對硬體的操控要求

- nVidia 的 Linux driver 設計不透過 DRM (!)

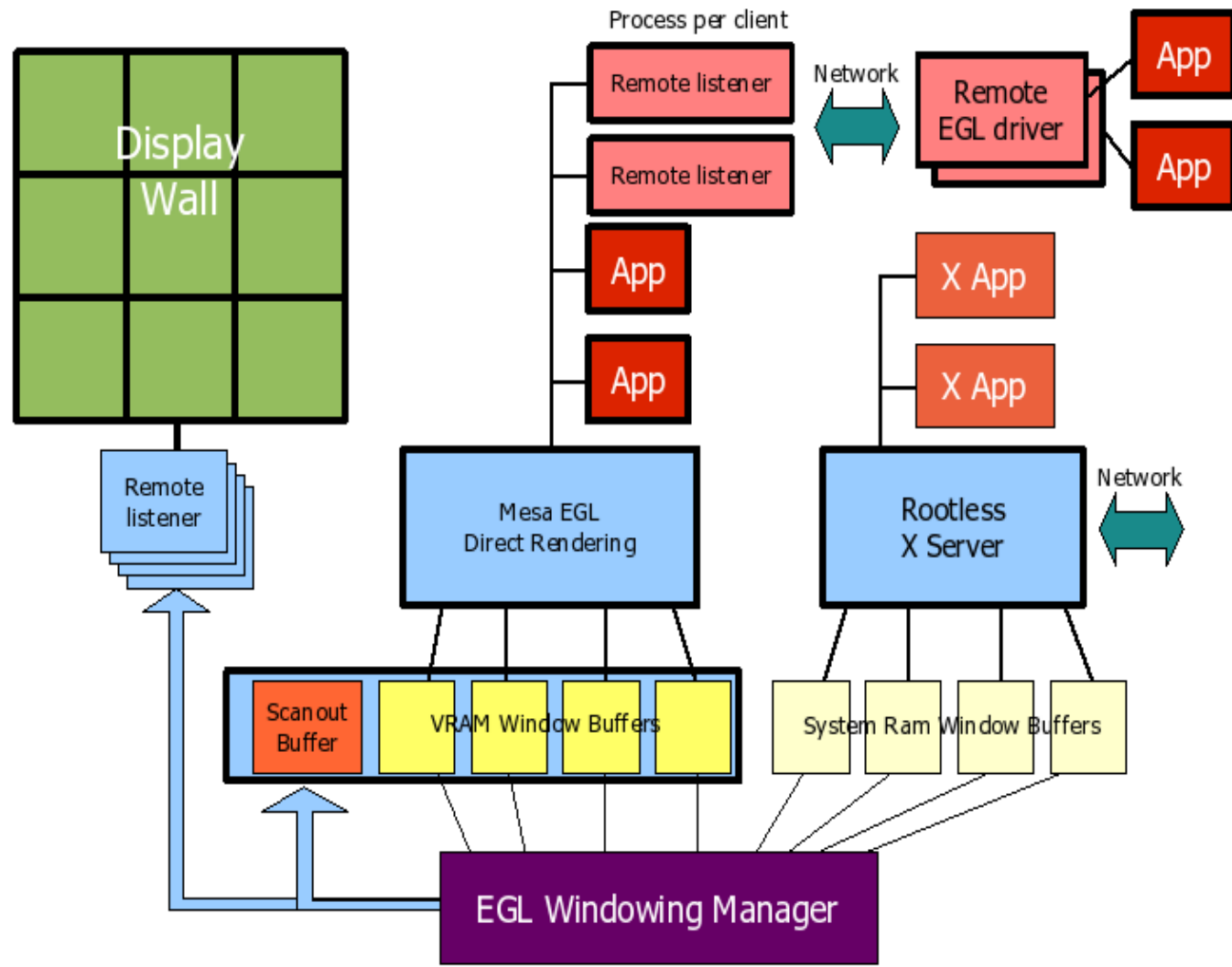
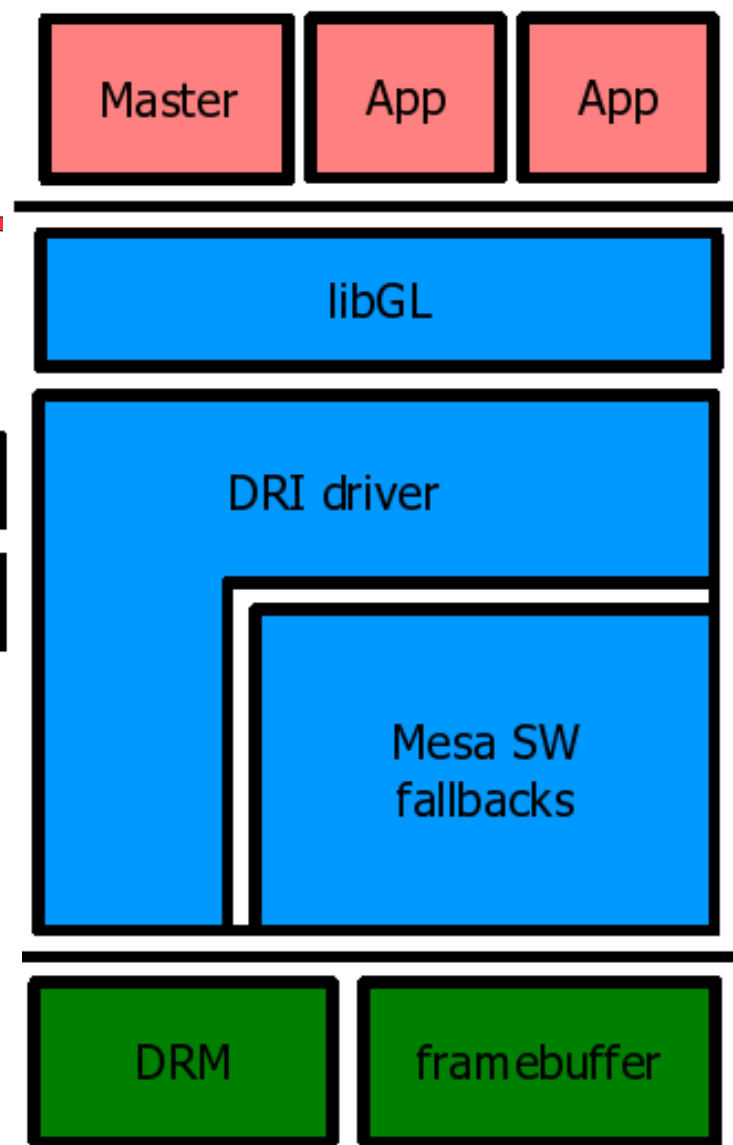
反璞歸真

- 最小化 GLX 的實現
- 透過 Mesa/Solo 實做 MiniGLX



➤ OpenGL/ES (EGL)

針對嵌入式系統精簡化的 OpenGL 規格
OpenGL/ES 2.0 允許 3D 圖形支援



- 爲何要理解 GL 實做細節？
 - 確認 API 相容度
 - Xgl/AIGLX 大量使用新的 OpenGL API
- Vendors
 - SGI Sample Implementation
 - nVidia

GL Vendor: SGI Sample Implementation

- Open sourced in January, 2000
- 重要元件
 - OpenGL 1.2 API
 - GLX 1.3
 - XFree86 4.0 的一部分
 - GLU 1.3
 - 為 Mesa 所採用
 - 工具程式與文件

GL Vendor: nVidia

- Proprietary / closed-source XFree86 4.0 driver for OpenGL 1.2
 - 支援 TNT* 、 GeForce* 、 Quadro 等硬體
 - 特性：
 - Direct Rendering
 - AGP 4x
 - NVIDIA extensions

OpenGL Conformance Issues

- Conformance Testing
 - OpenGL ARB (Architectural Review Board)
Conformance Test
 - Vendor-specific Tests
- ABI 與 Versioning
- Linux/OpenGL Base Standard (oglbase)
 - 類似 LSB 的概念，但涵蓋 ABI 與 OpenGL Runtime

OpenGL Extensions (1)

➤ 選擇性

- 但是高階應用需透過這些 Extensions 來實現

➤ 在 OpenGL 中實現新功能的方式

類似 dlsym
oglbase ABI 的一部分

- GL_ARB_multitexture
- GL_ARB_texture_env_add
- GL_ARB_multisample
- GL_ARB_texture_compression
- GLX_ARB_get_proc_address
- GL_EXT_blend_color
- GL_EXT_blend_subtract
- GLX_EXT_texture_from_pixmap
- ...

OpenGL Extensions (2)

- ▶ Xgl 在 OpenGL 需要的 Extensions
 - OpenGL/Mesa Off-screen Rendering
 - Framebuffer Object (FBO)
 - GL_EXT_framebuffer_object
 - back buffer / pbuffers (for pixmaps)
 - GLX_SGIX_pbuffers
 - GLX_EXT_texture_from_pixmap

binding redirected top-level windows to texture objects.

3D 世界的特徵

- ▶ 將 2D 視窗視作 “**paint**”
 - Project Looking Glass, Sun Microsystems
 - Croquet Project
- ▶ 共享資源管理機制，並透過硬體 3D 加速來實現
- ▶ 避免使用分離的 2D 驅動程式，亦即在 Framebuffer 層面需要作修正
- ▶ ubiquitous(無所不在的) 3D applications / object



淺談 Linux 3D 技術

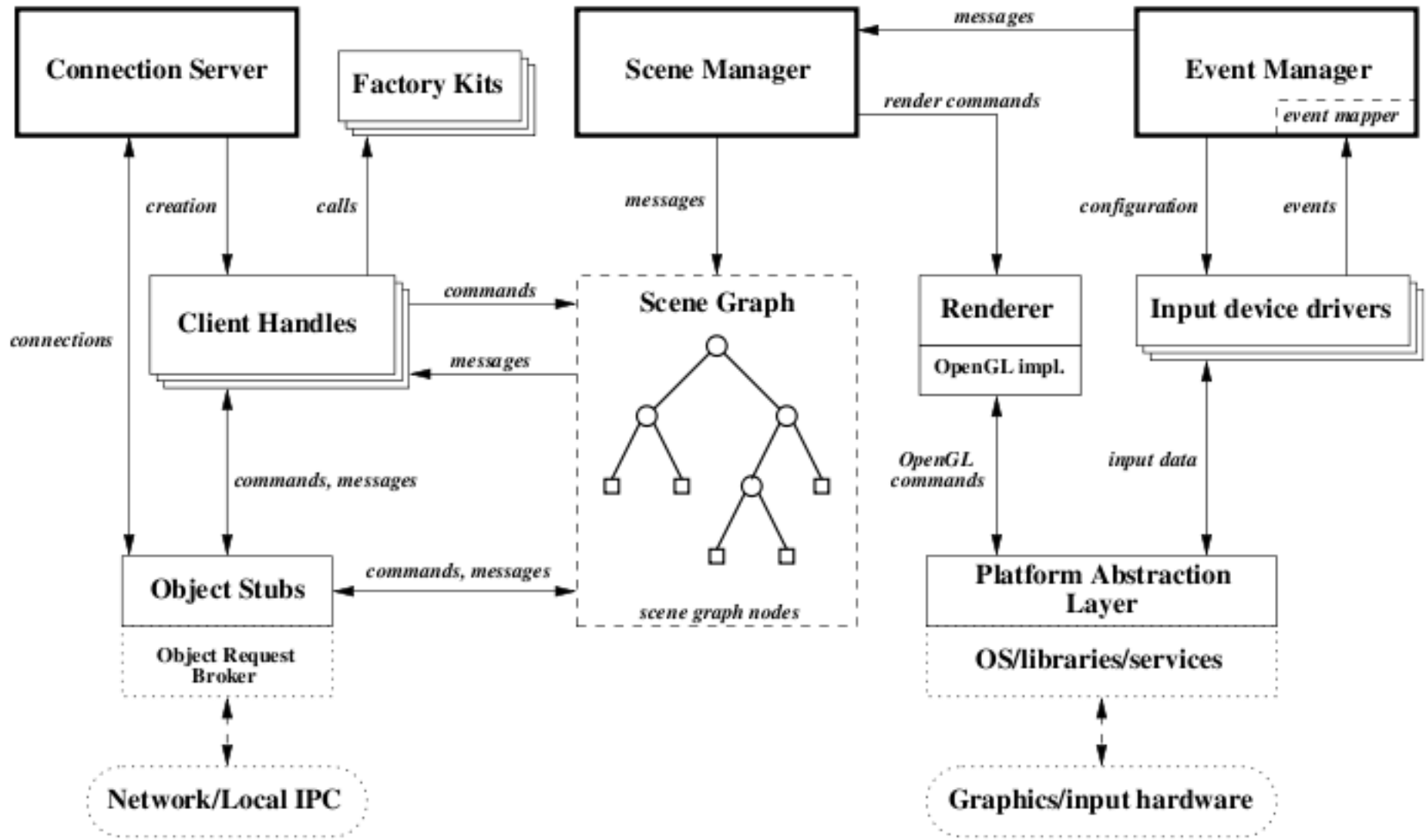
2D 與 3D 的整合

- 現況
 - Window system 還是 2D 架構
 - 3D 應用程式「活」在 2D 視窗中
 - 無法通透的共享資源
- 目標
 - ✓ 將 X 視同一個 OpenGL 應用程式
 - ✓ 2D 資源在 3D Living World 是可見的 (visible)
 - ✓ 3D 應用程式可被「投影」或「描繪」予 2D 繪圖內容

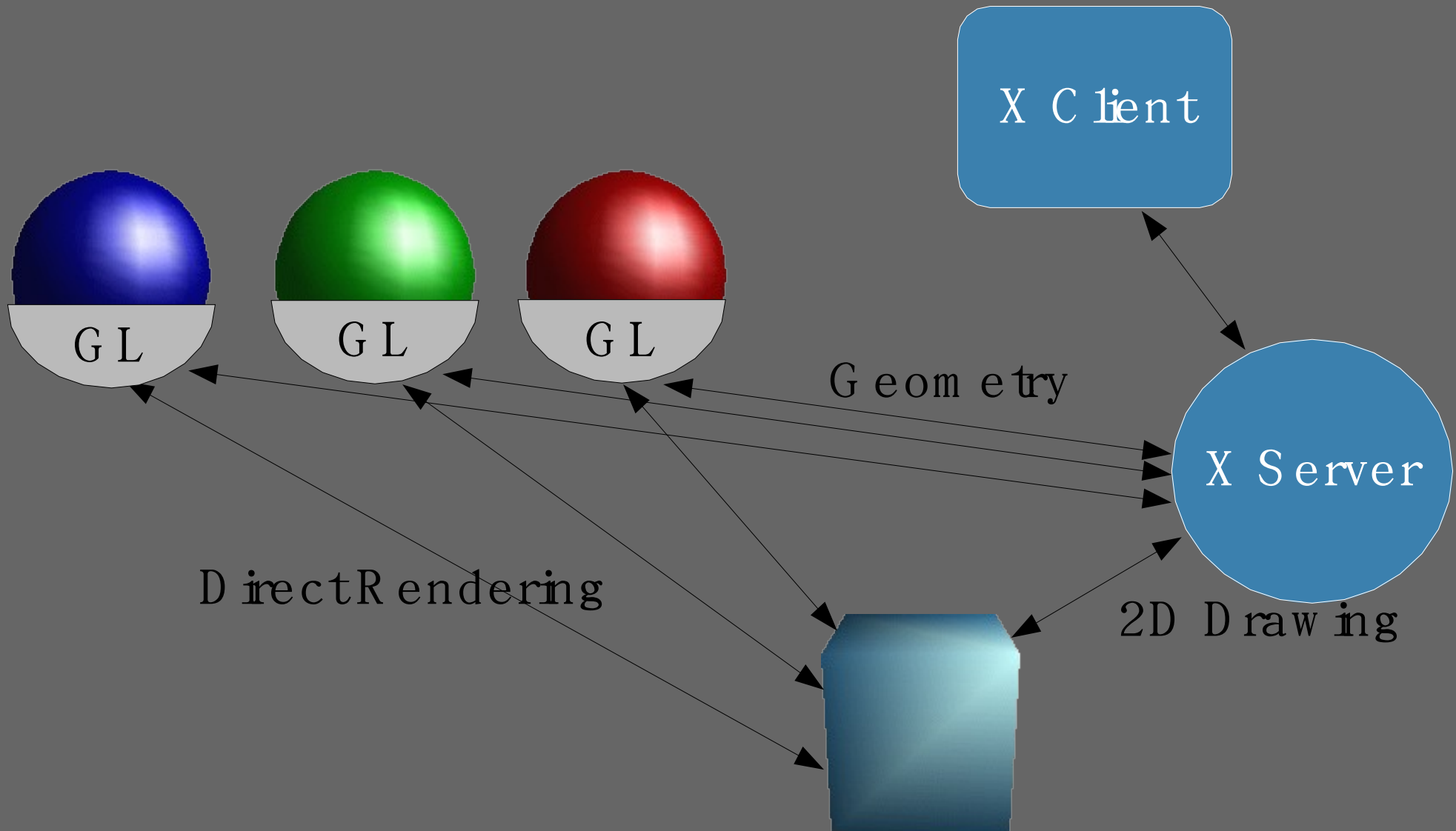
3D Window System 面臨的技術挑戰

- 一致性的 3D 高階概念
- Retarget-able
 - 在多樣化的 3D 環境中，應用程式應該要能再不需要修改的前提下運作
- Backwards-compatible
 - 傳統 2D Window System 在 3D 環境中要能被存取與交互運作
- Multitasking

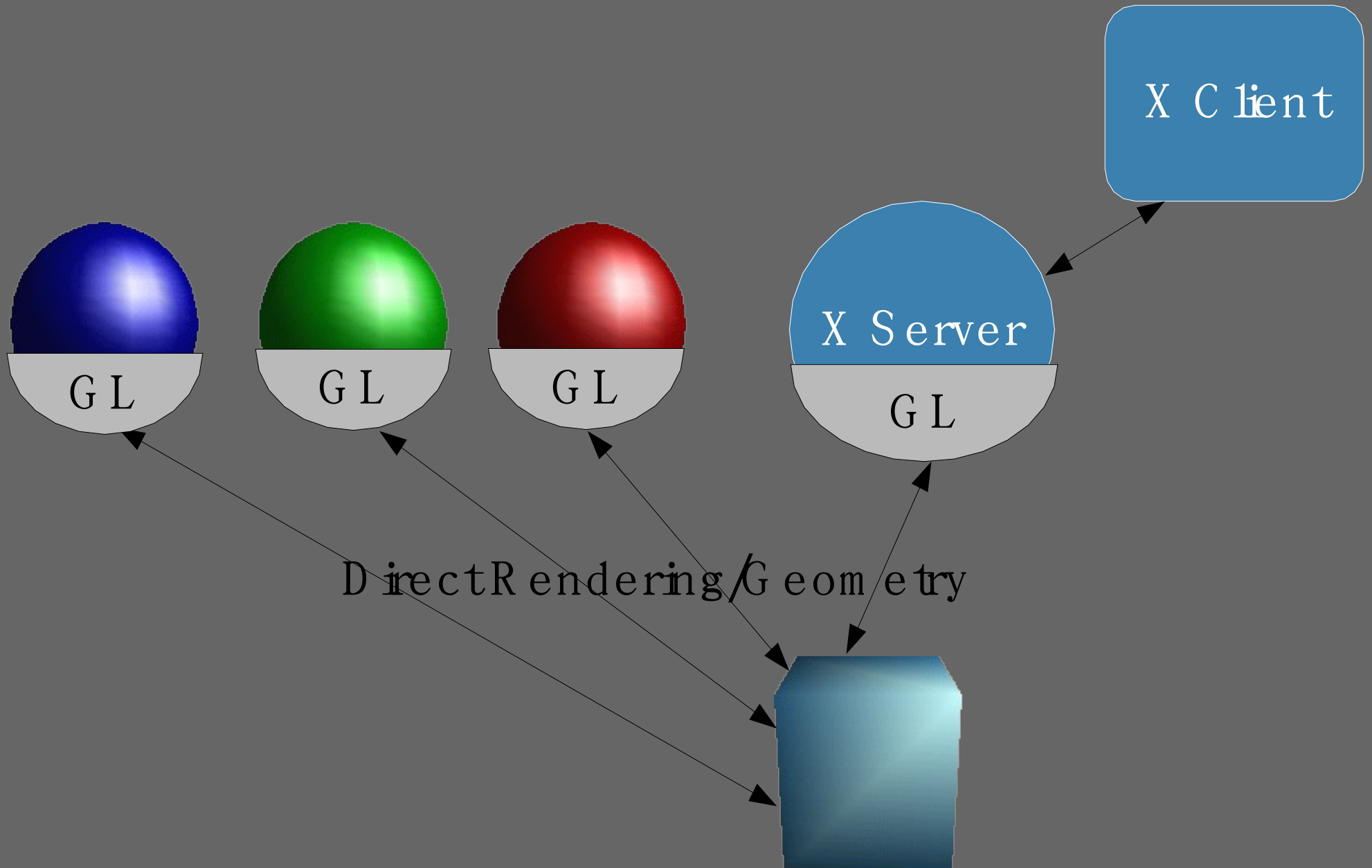
3D 系統典型架構



Existing Architecture

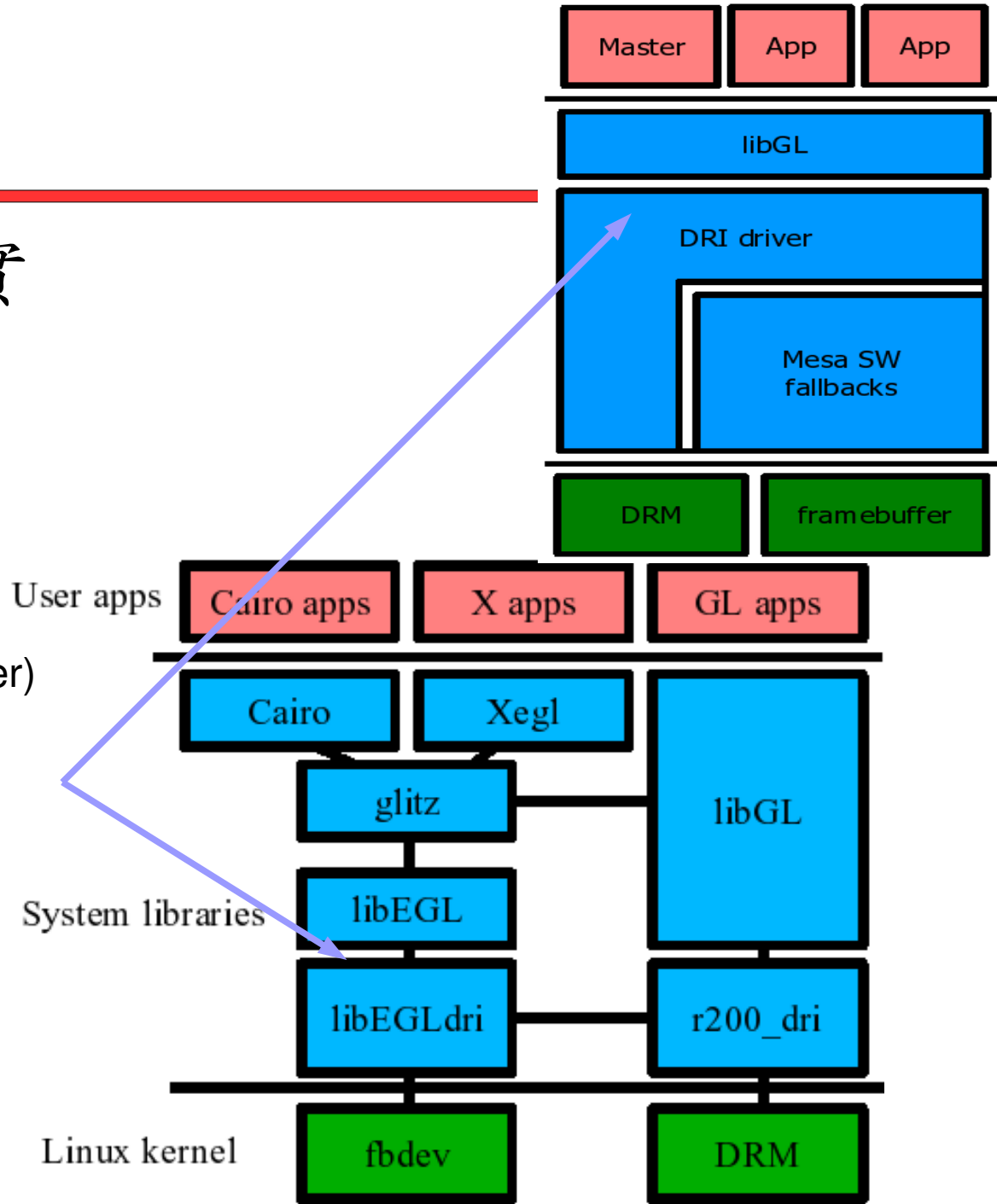


Proposed Architecture



Xegl

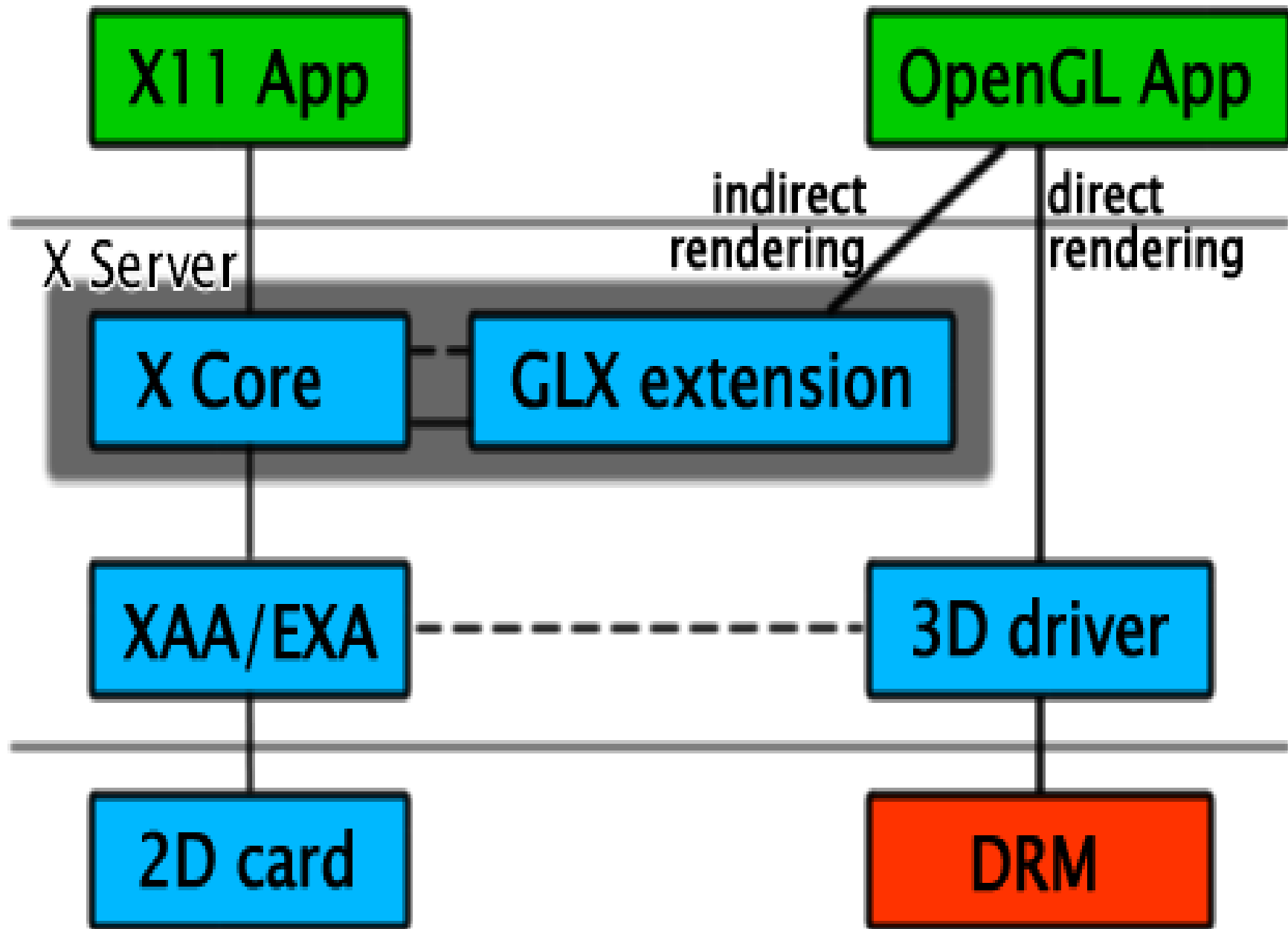
- 最小化 X on GL 的實現
- 目前實做
 - Linux Framebuffer
 - DRM (Direct Rendering Manager)
 - ATI Radeon
 - Mesa/Solo
 - Cairo/glitz



分歧：XGL 與 AIGLX

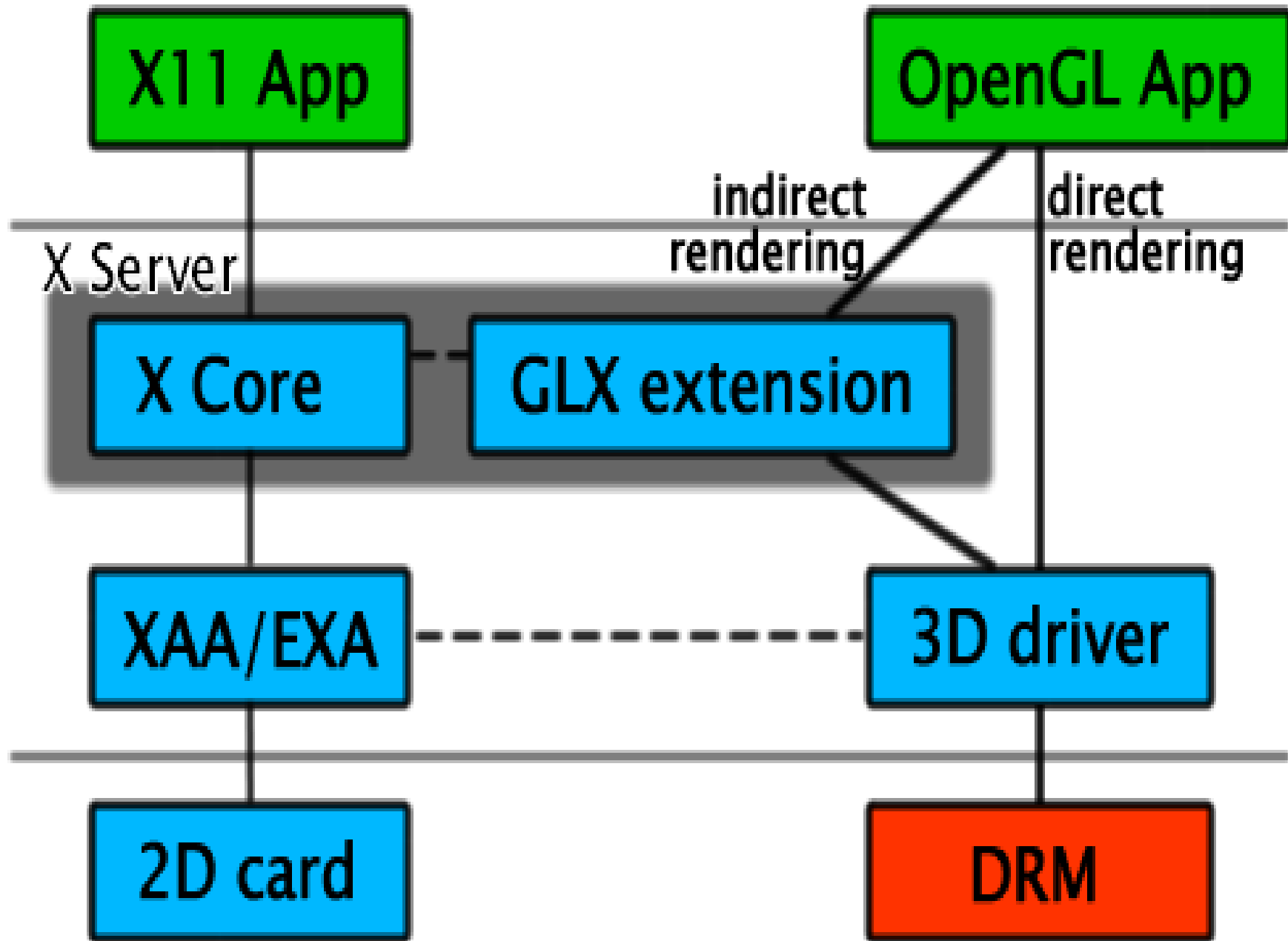
- Xegl 對於 OpenGL 硬體加速機制來說，是簡單又有效率的解法（一個 driver 同時適用 2D 與 3D）
 - 然而現在的 Driver 都需經過大量修改
- XGL (by Novell)
 - 最終目標：**Xegl**
 - 過渡時期：透過 **Xglx**
- AIGLX (by RedHat)
 - Accelerated Rendering to Indirect OpenGL Contexts

X.org

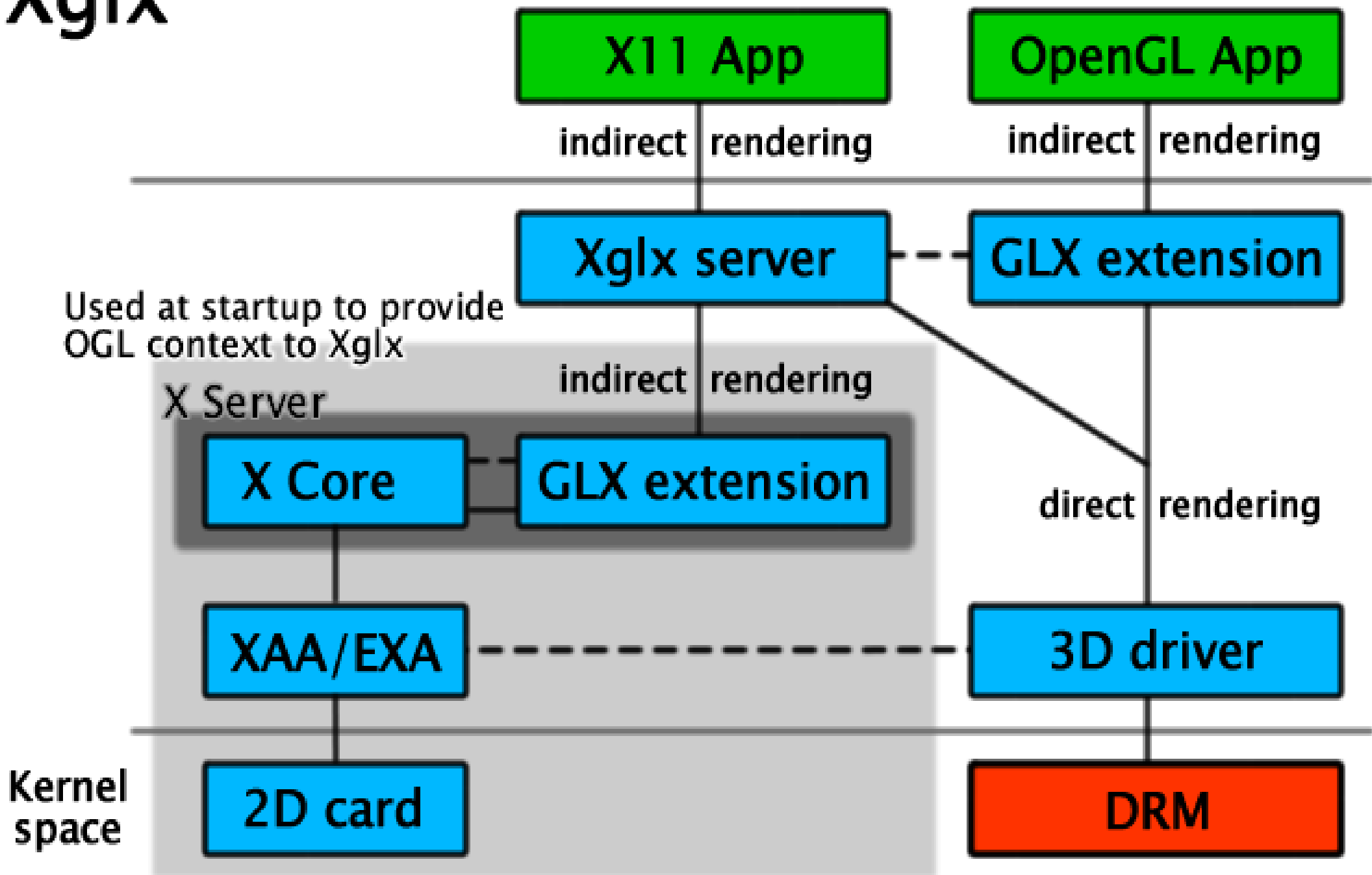


Kernel space

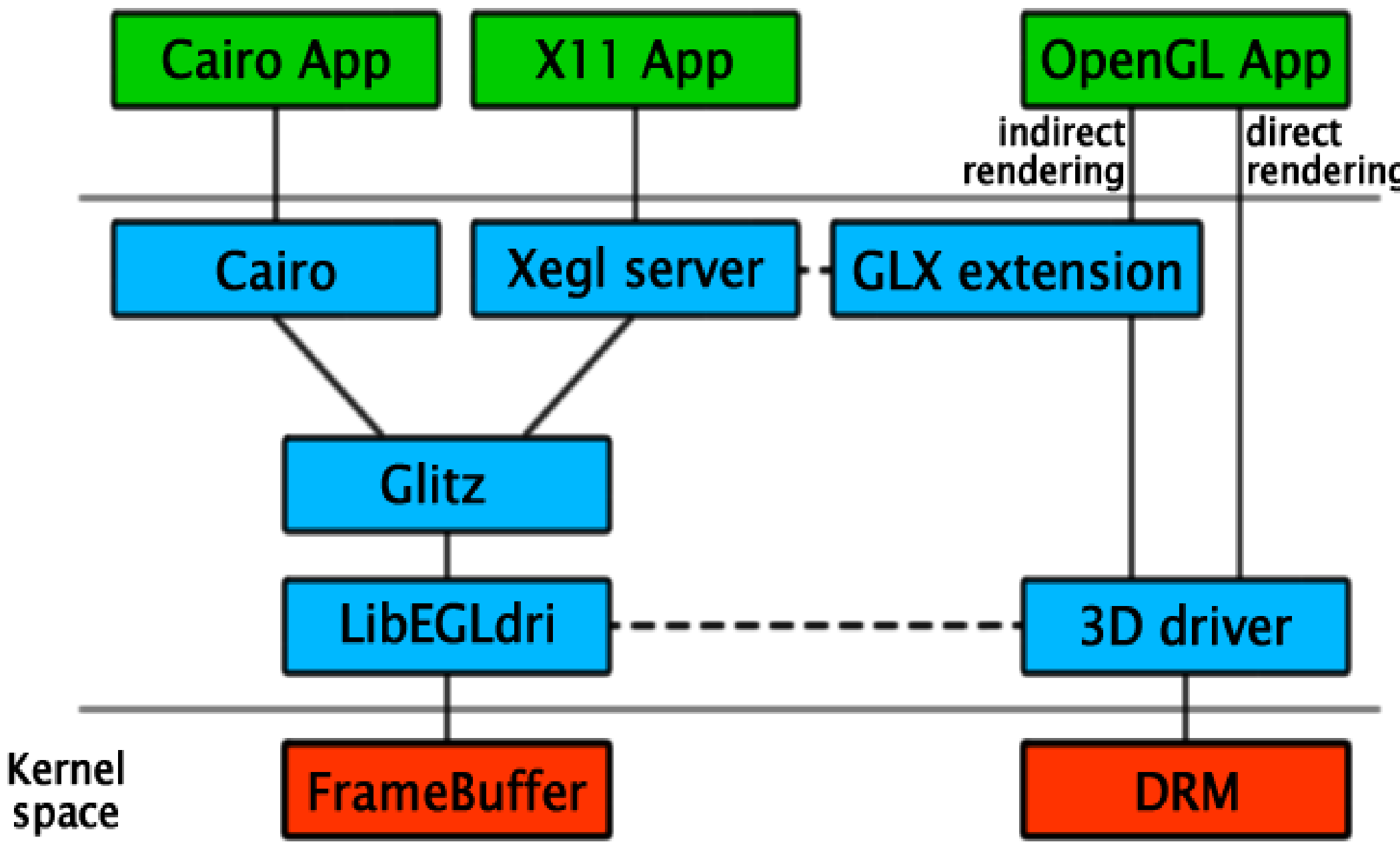
Aiglx



Xglx



Xegl



~~Xglx~~ Xorz



indirect rendering



indirect rendering



indirect rendering



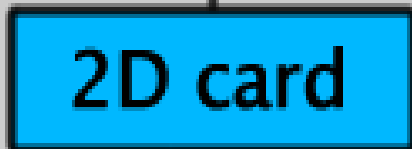
Used at startup to provide OGL context to Xglx

X Server



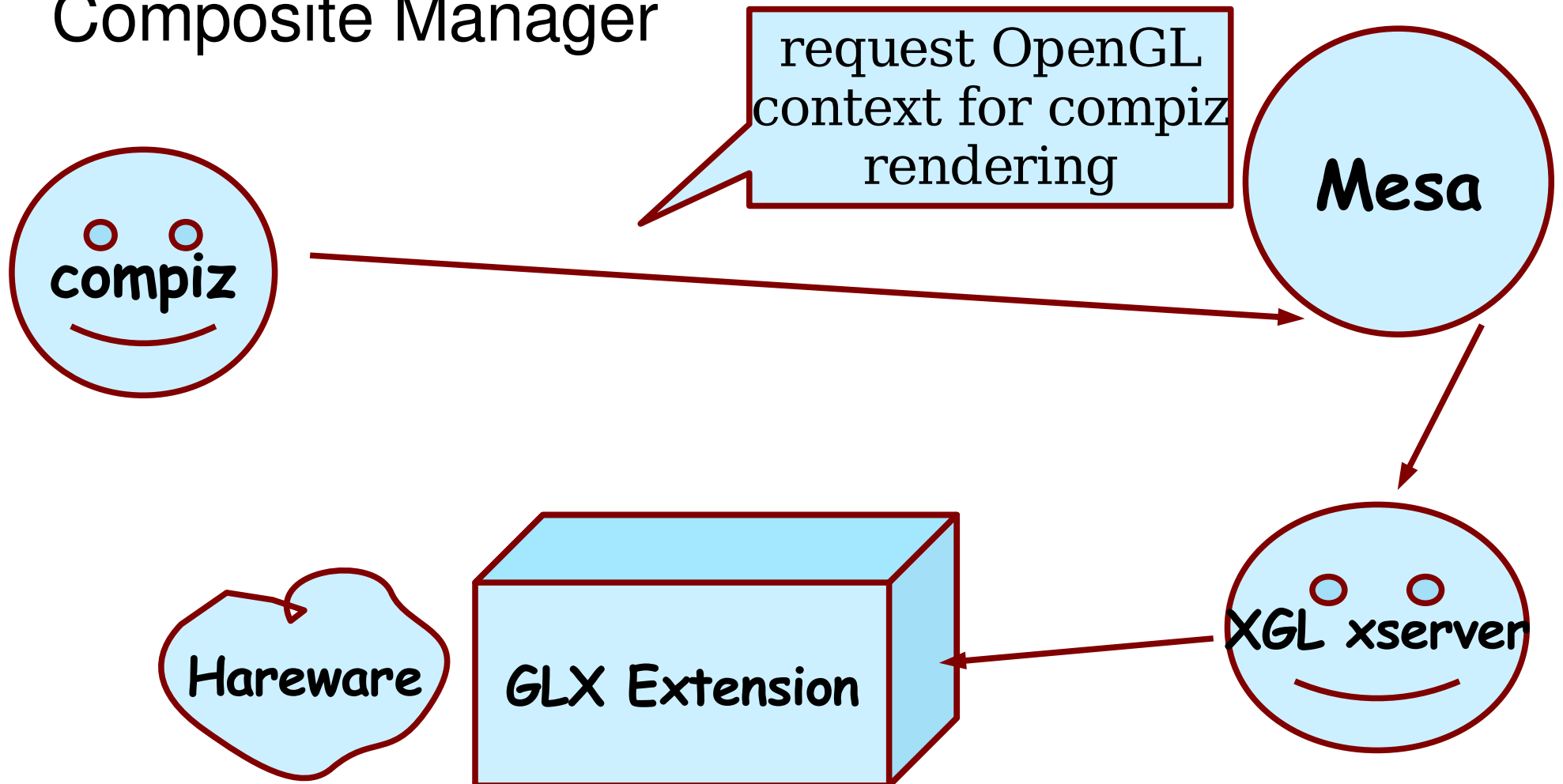
direct rendering

Kernel space



在 XGL 下 GL client 的運作

- compiz ==> OpenGL 撰寫的 Window Manager 兼 Composite Manager



3D：究竟是 3D 還是未來？

- 我們無法預知未來！
- XGL / Compiz / AIGLX / 3DWM / Project Looking Glass / ... 不全然是「酷炫玩具」
 - 驗證在延續既有架構的前提下，作概念性的實驗
- 跟 Microsoft Vista 相比？跟 Apple MacOS X 較勁？
 - 並非作業系統之間的戰爭
 - 相互激盪優秀的概念
 - 自由、開放

- Xgl - <http://www.freedesktop.org/wiki/Software/Xgl>
- AIGLX -
<http://fedoraproject.org/wiki/RenderingProject/aiglx>
- Novell Compiz - <http://www.novell.com/linux/xglrelease/>
- jserv's slides - <http://jserv.sayya.org/freedesktop/>

感謝您的蒞臨指教！

Just Enjoy It!