

Project Looking Glass:

3D Desktop Exploration...

Hideya.Kawahara@sun.com

Deron.Johnson@sun.com

X Developer's Conference – Apr. 29th 2004





Just wanted to break the boundary...



The Opportunity

We all know about it...

- 20 year old 2D desktop metaphor
- Significant improvement of 3D capabilities
- How we can capitalize on this?

What would be the right next step?

The Opportunity

Key question...

- **Go beyond accelerating 2D GUIs**
- **But without forcing users into unfamiliar Virtual Reality world**

The Approach...

Evolving the “Paper Paradigm!”

Existing 2D Applications



New 3D Applications

Demo

Proof-of-concept



Explore the New Frontier

Now, how to do it...

- **Integration with existing system is key**
- **Totally new area**
Need to experiment aggressively...
... what we've shown must be a tip of iceberg
- **How we started:**
Leveraged Java™ technology's productivity...
... delivering “a reference platform”

Explore the New Frontier

Goals

Leveraging Project Looking Glass...

- **Add a new dimension to the desktop while following the standards used by well established Linux desktops (GNOME and KDE)**
- **Collaboratively develop additional standards relative to 3D interaction and UI**

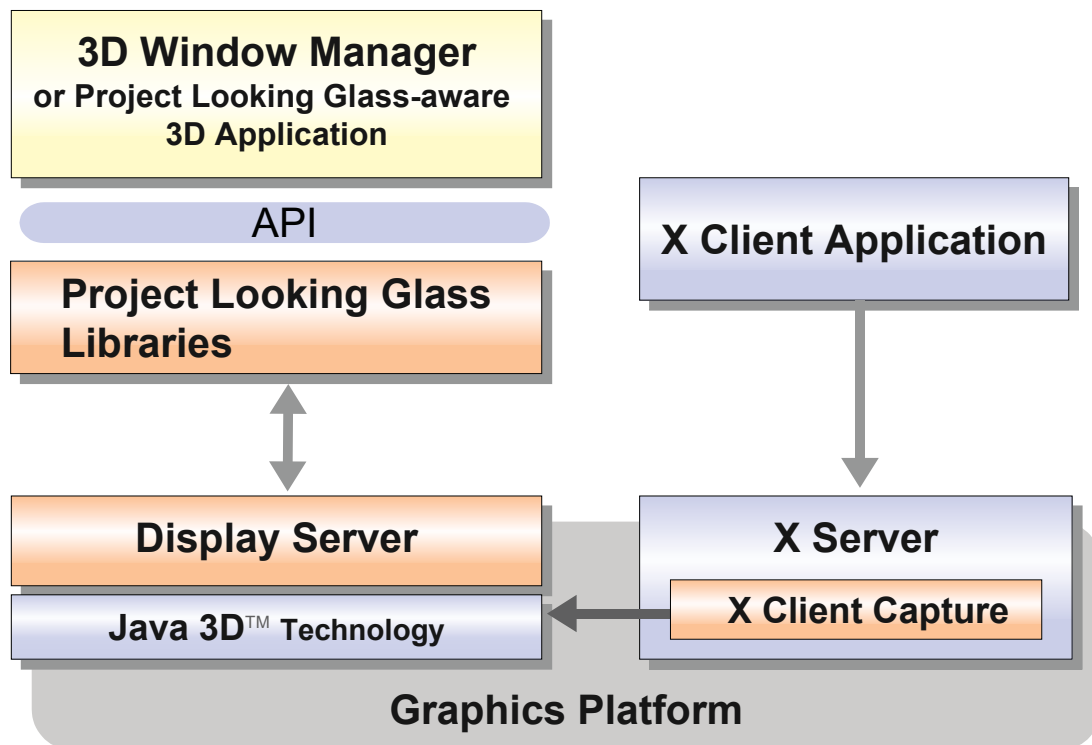
Overview of Technical Info

- **High-level Architecture Overview**
- **Graphics Platform**
- **Client-side API**
- **How All the Pieces Interact**
- **Possible Areas of Collaboration**



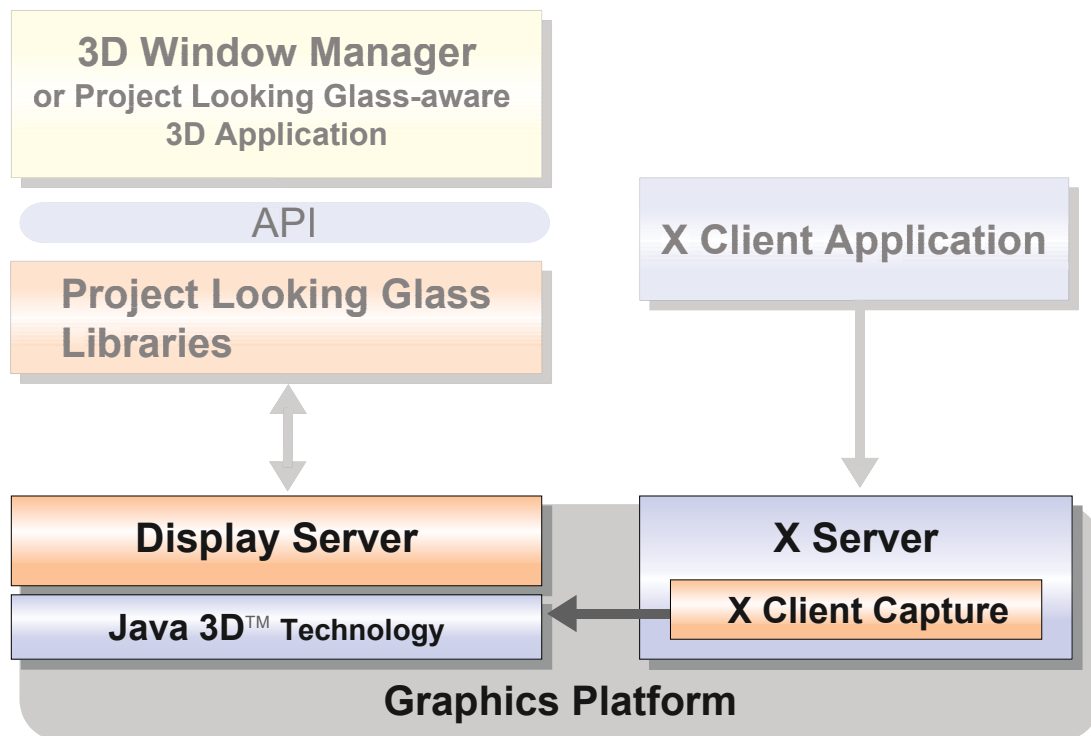
Super High-level Architecture Overview

Platform for further exploration



APIs for 3D Desktop Development

Platform for further exploration



Project Looking Glass Graphics Platform

or “How to Draw 3D X Windows”

Deron.Johnson@sun.com



Presentation Goals

- Reveal the magic behind the pretty pictures
- Overview of Project Looking Glass Graphics architecture
- How Project Looking Glass fits in with X Server
- Graphics Futures

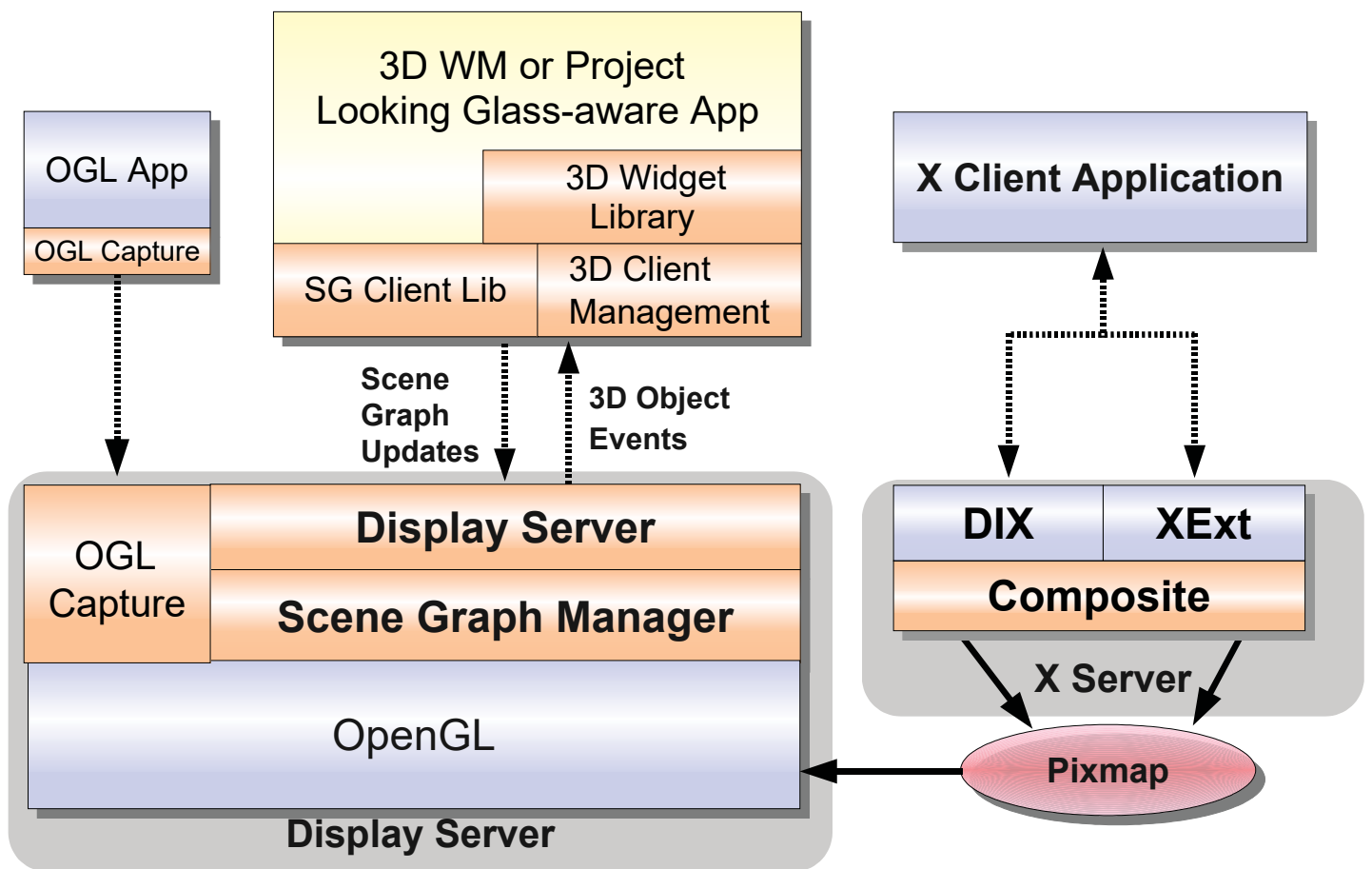
Project Looking Glass – Demo vs. Product

- Currently Available: Proof-of-concept Demo
 - Proof-of-concept only
 - Goal: Stir people's imaginations
 - Graphics layer is very basic -> Needs improvement
- Working On: Refined Graphics Platform
 - Goals:
 - New Architecture: Client-Server, More modular
 - New Features: OpenGL capture
 - Productizable
 - Planned Availability:
 - Mid-2004: Developer's Release (Linux only)
 - Eventual Product for Linux, Solaris™ x86, and Solaris SPARC™

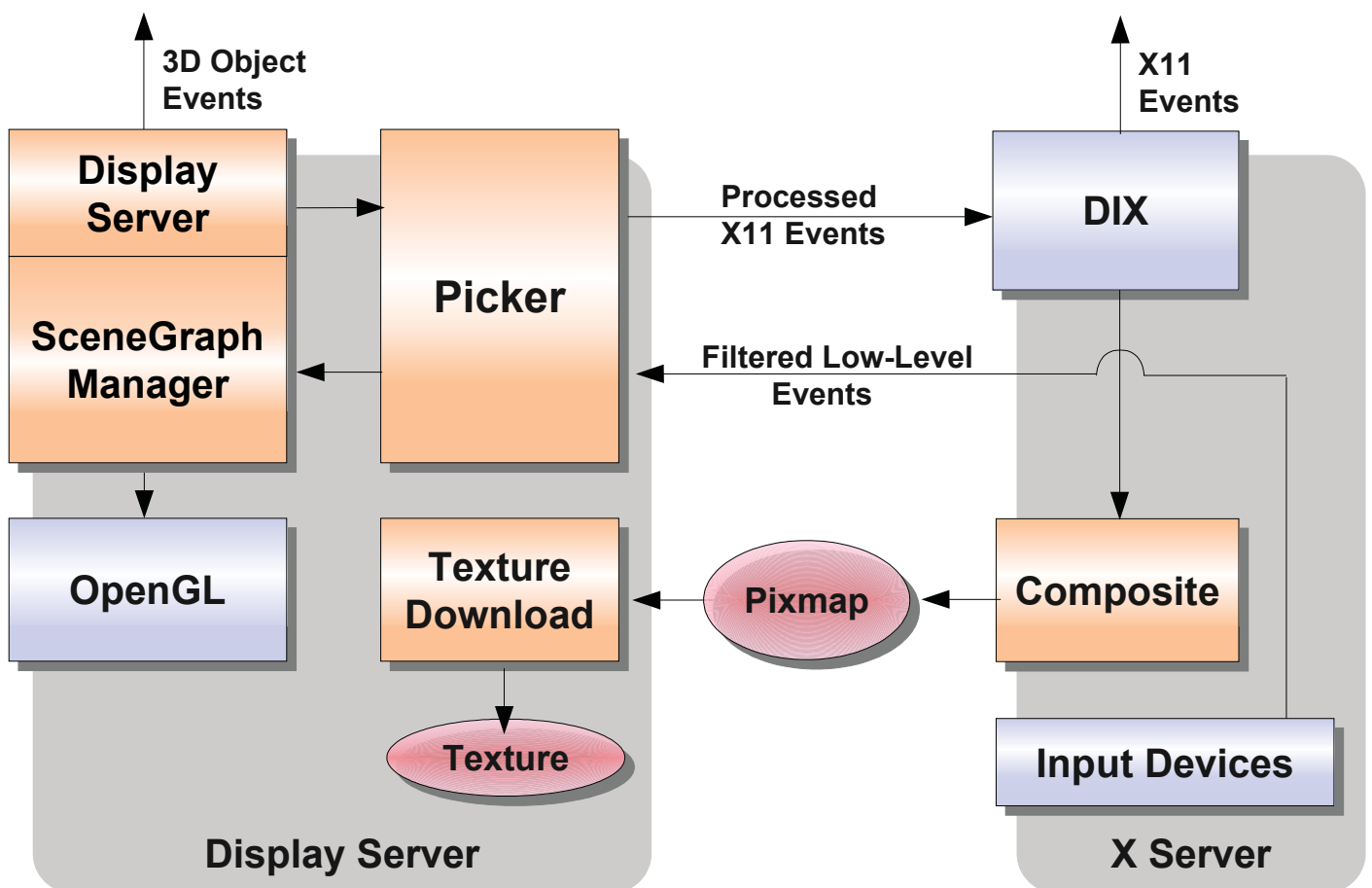
Project Looking Glass Graphics Platform

- X Server (XS) + new 3D Display Server (DS)
- Output Redirection
 - XS: Redirects window rendering -> backing pixmap
 - DS: Loads pixmap into texture
 - Whenever pixmap contents change
 - For example: damage event
- Input Redirection
 - XS: Sends low-level device events to DS
 - DS: Determines which window event is for
 - Also determines coordinates in window space
 - Sends events back to XS for further processing

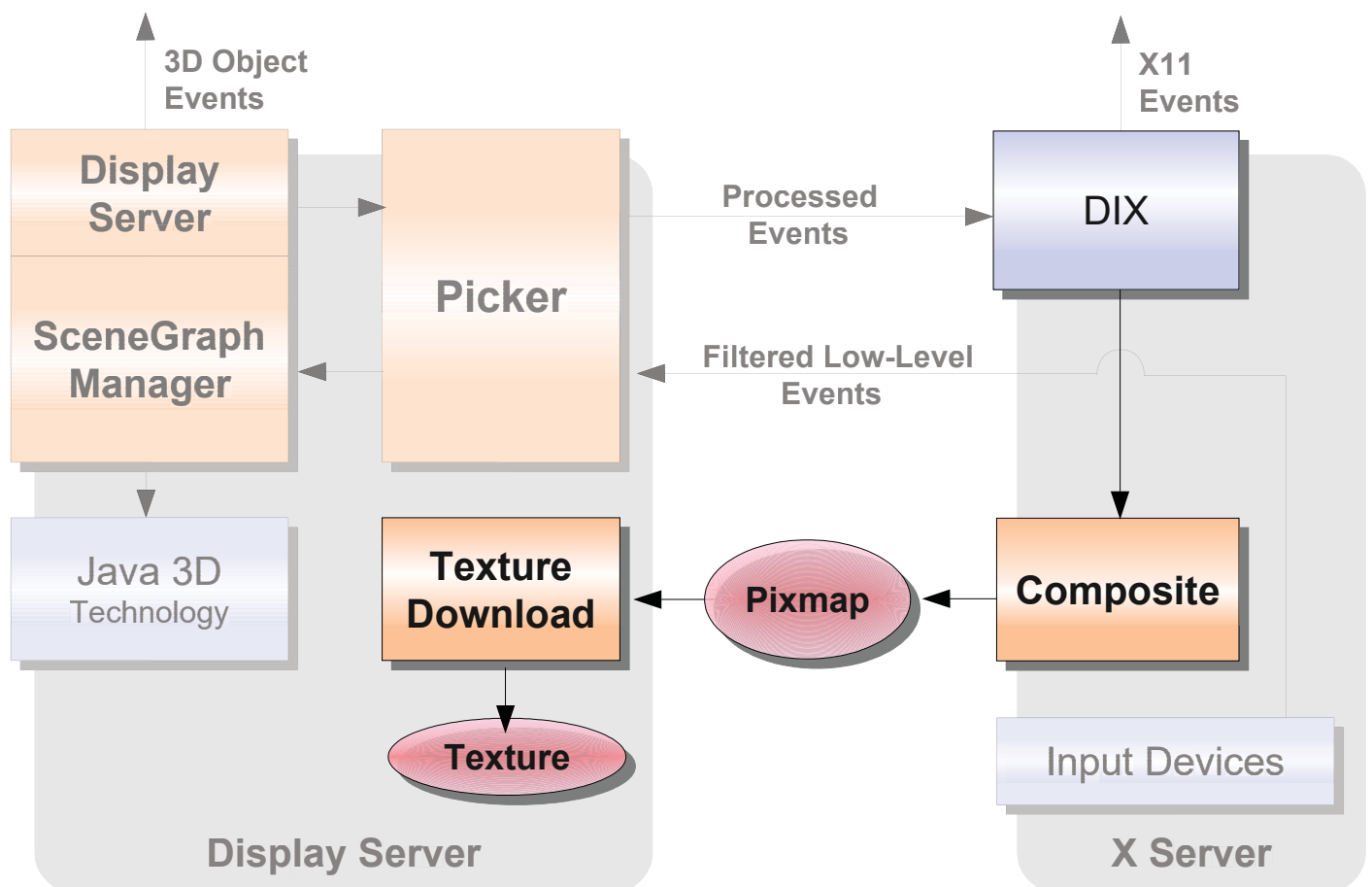
Graphics System and X11 Integration



Graphics Platform Detailed View



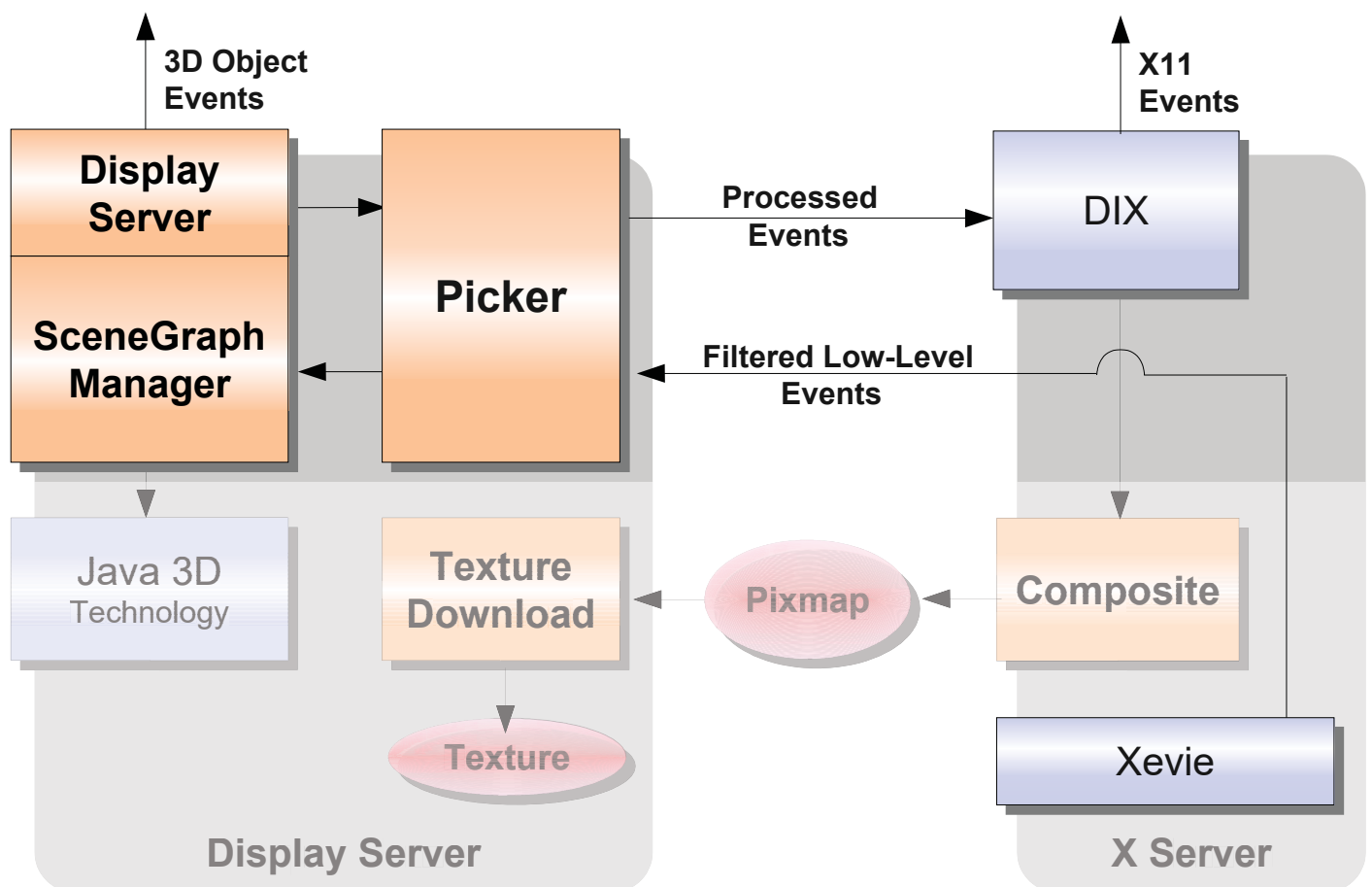
Output Redirection Detailed View



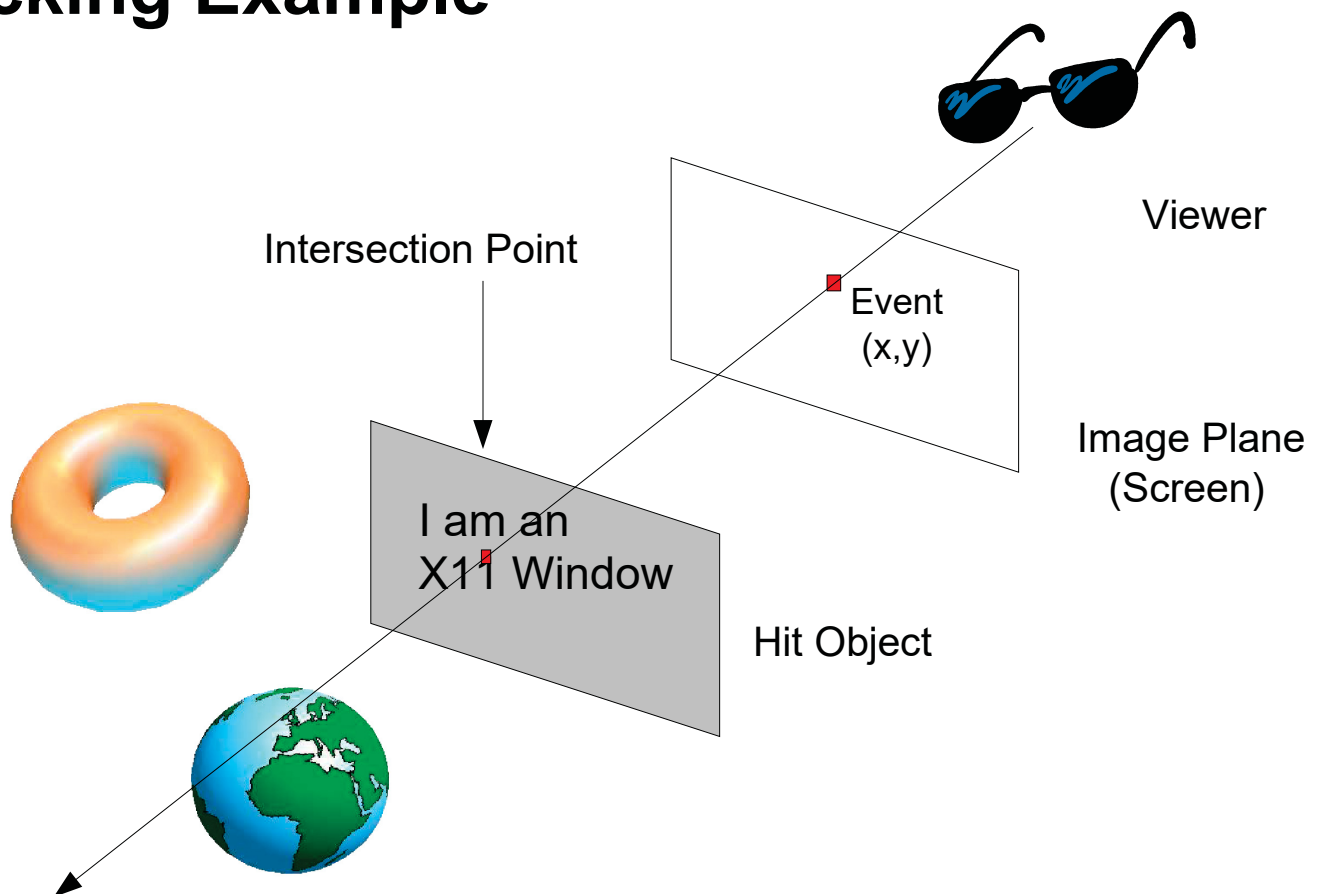
Key: Composite Extension

- Developed by Keith Packard of HP, et al.
- Redirects X11 rendering to a backing pixmap
 - New coordinate space: **Pixmap Space**
 - DDXs: must translate abs screen coords into this space
 - Primitive coordinates
 - GC composite clip
 - This will give best performance
 - But to avoid modifying a DDX
 - **CompTran**: a GC wrapper layer that does the translation for you
 - Can use when you don't have the DDX source

Input Redirection Detailed View



Picking Example



Picker

- X Server device events -> Picker
 - Uses Xevie, a proposed Xorg standard
 - Co-developed by Sun and SUSE
- Fires ray toward event x,y into scene
- Determines hit object (intersection)
- Hit object = X11 window
 - Translates intersection point to window space
 - Updates event x,y
 - Sets event window field
 - Sends event back to X Server
 - DIX: modified to handle events with windows already set
 - XYToWindow()

Future Possibilities

- Window Capture Performance
 - Direct Render-to-Texture (both X11 and OpenGL)
 - Interprocess shared textures
- Input Devices connected to DS?
- Scene Graph
 - Spatialized audio (aural cues for 3D chat windows!)
 - Programmable shaders
 - Wild stuff
 - Physics simulation
 - Collision detection
 - Particle systems
 - What can you imagine?

Let's Collaborate!

- 3D window system area is in its infancy
- Lots of room for collaborative innovation
- Let's collaborate to build a premier 3D window system platform
- Sun is willing to contribute:
 - Ideas
 - Code
 - Engineering

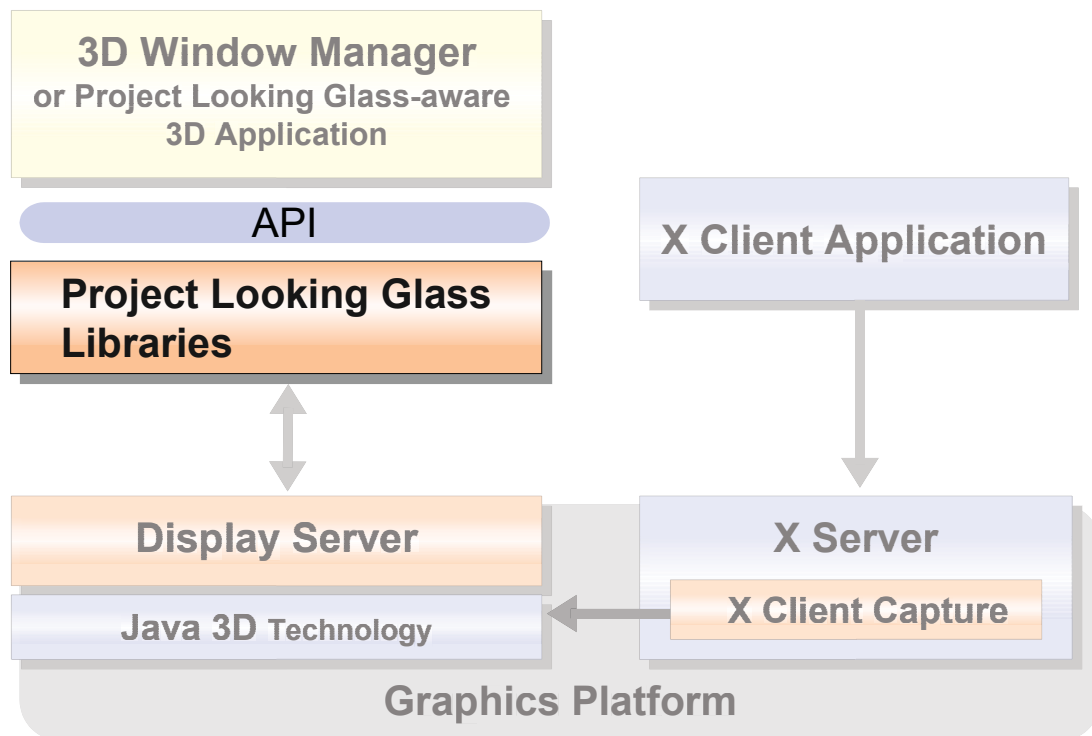
Overview of Technical Info

- High-level Architecture Overview
- Graphics Platform
- **Client-side API**
- How All the Pieces Interact
- Possible Areas of Collaboration



APIs for 3D Desktop Development

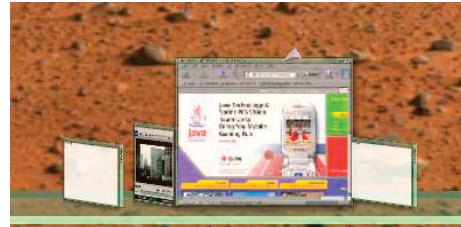
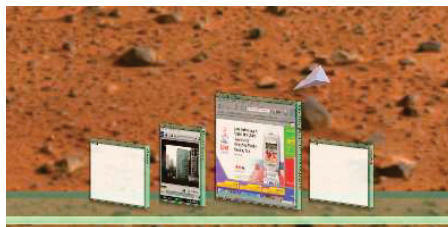
Platform for further exploration



APIs for 3D Desktop Development

Platform for further exploration - Proposal

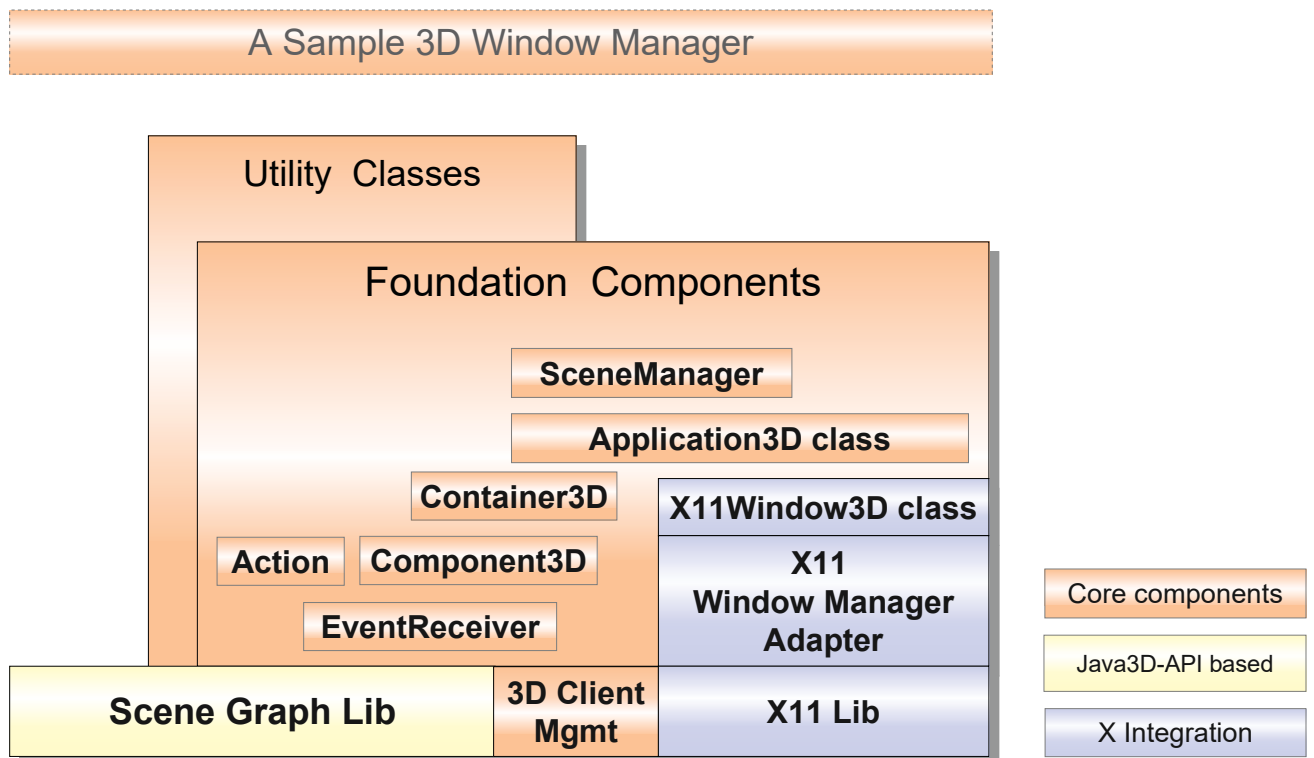
- Core APIs to kick start the exploration
- Java technology based reference implementation
- Java 3D API-based scene graph (feature subset) + Base classes for developing 3D widget set
- Support for integration of existing applications
- Focus on animation support



An example of 3D Container with animation

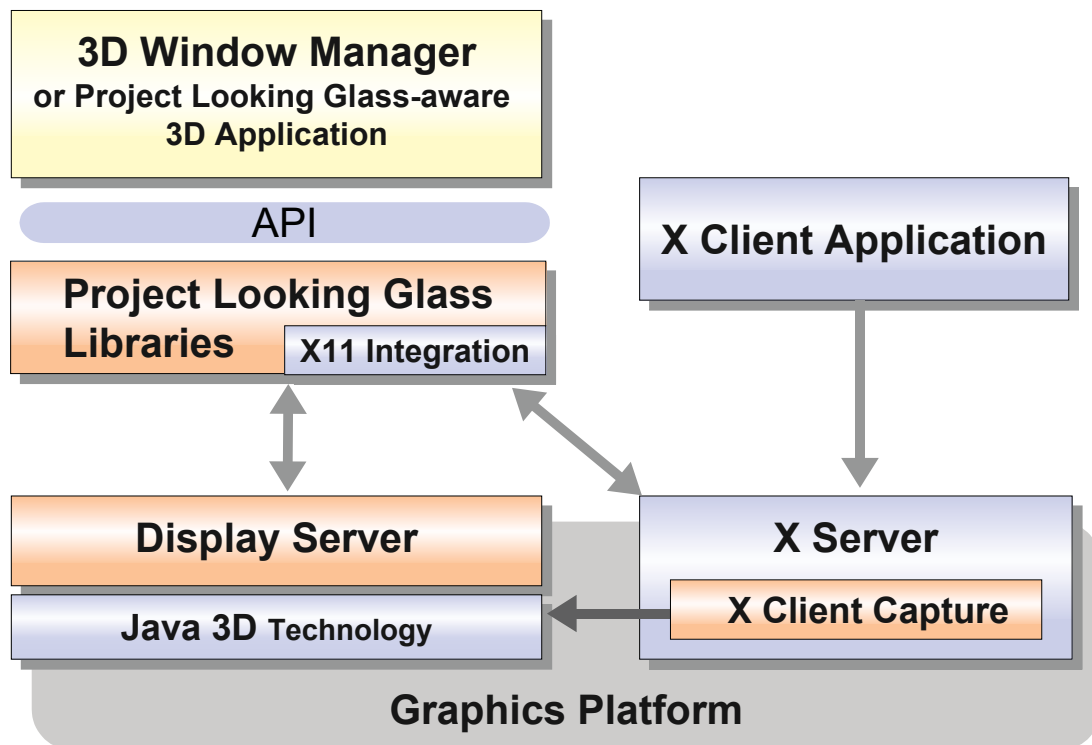
APIs for 3D Desktop Development

High-level block diagram - Proposal



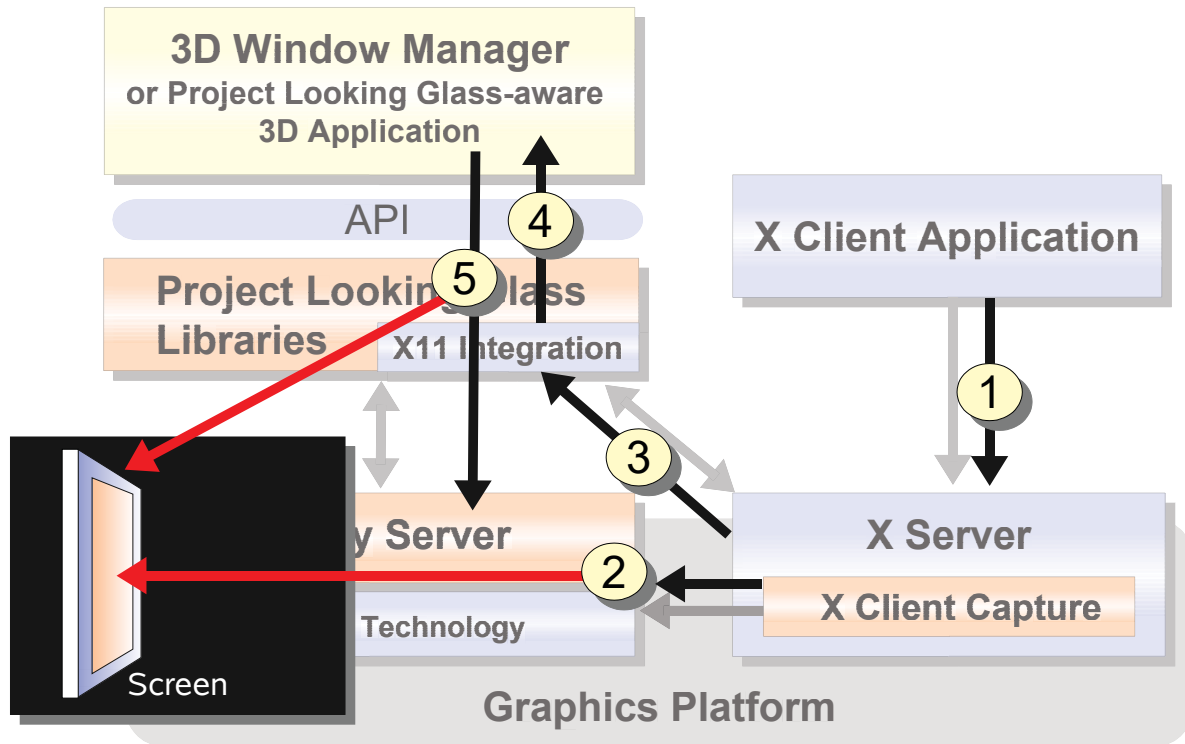
How All the Pieces Interact...

Example: Window creation and resize



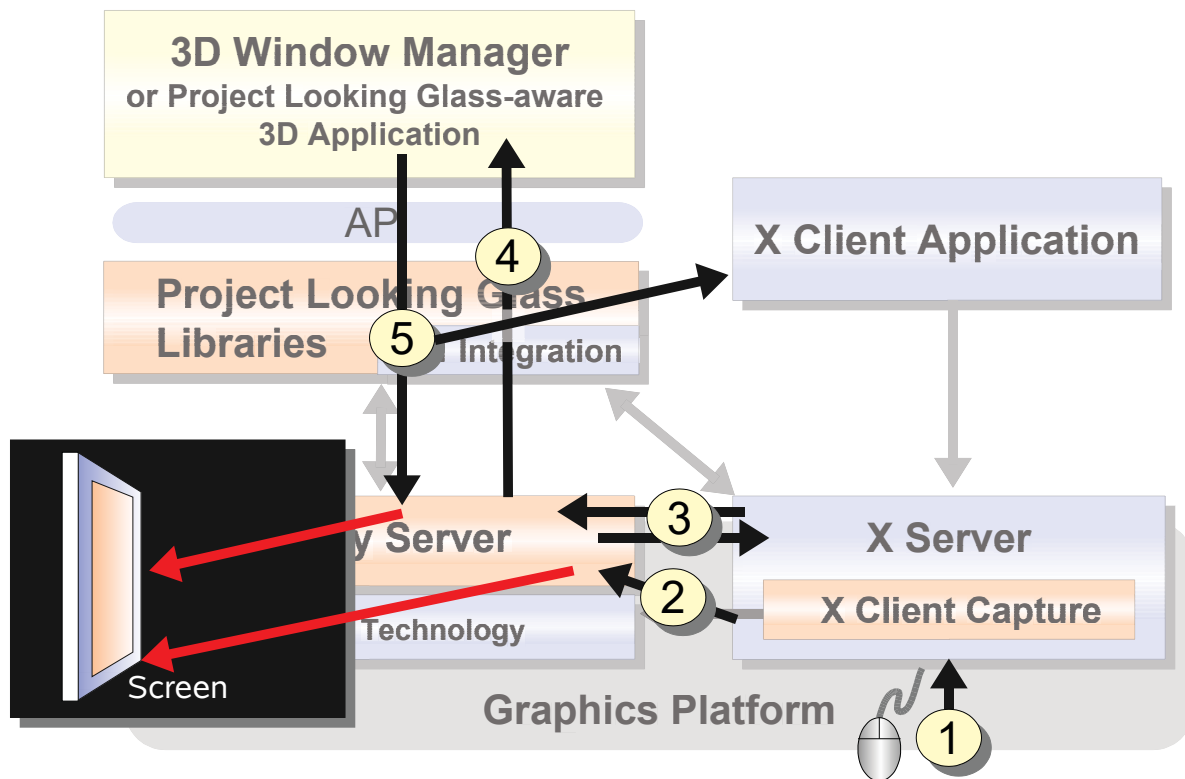
How All the Pieces Interact...

Example: Window Creation



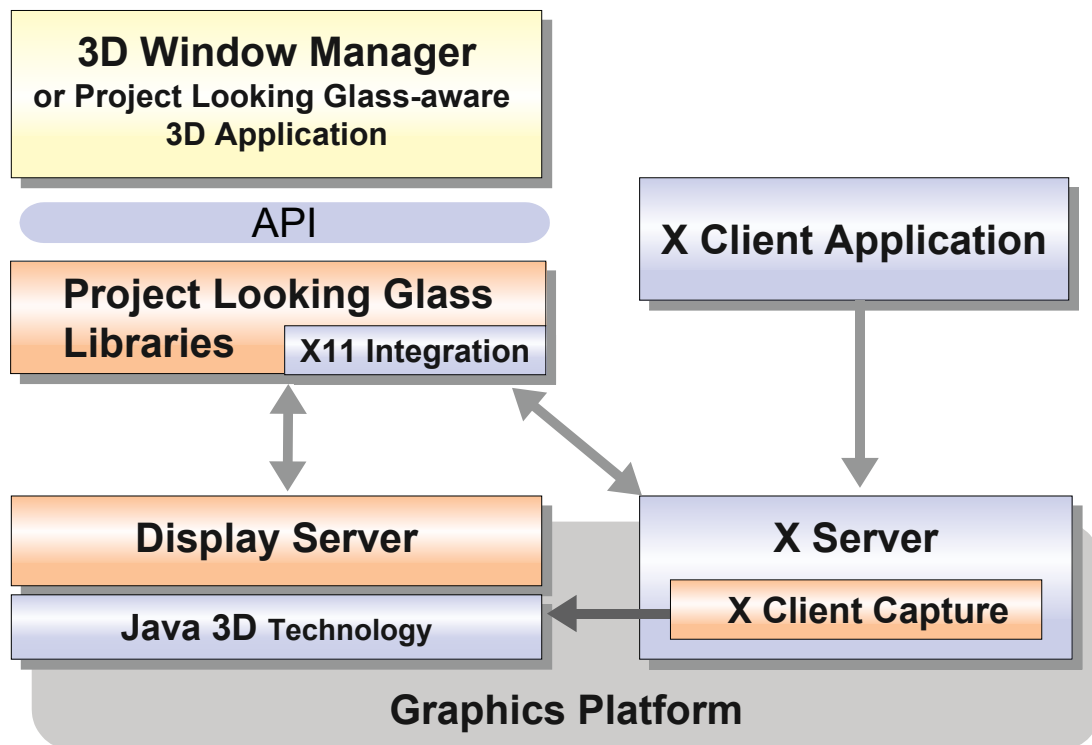
How All the Pieces Interact...

Example: Window Resize



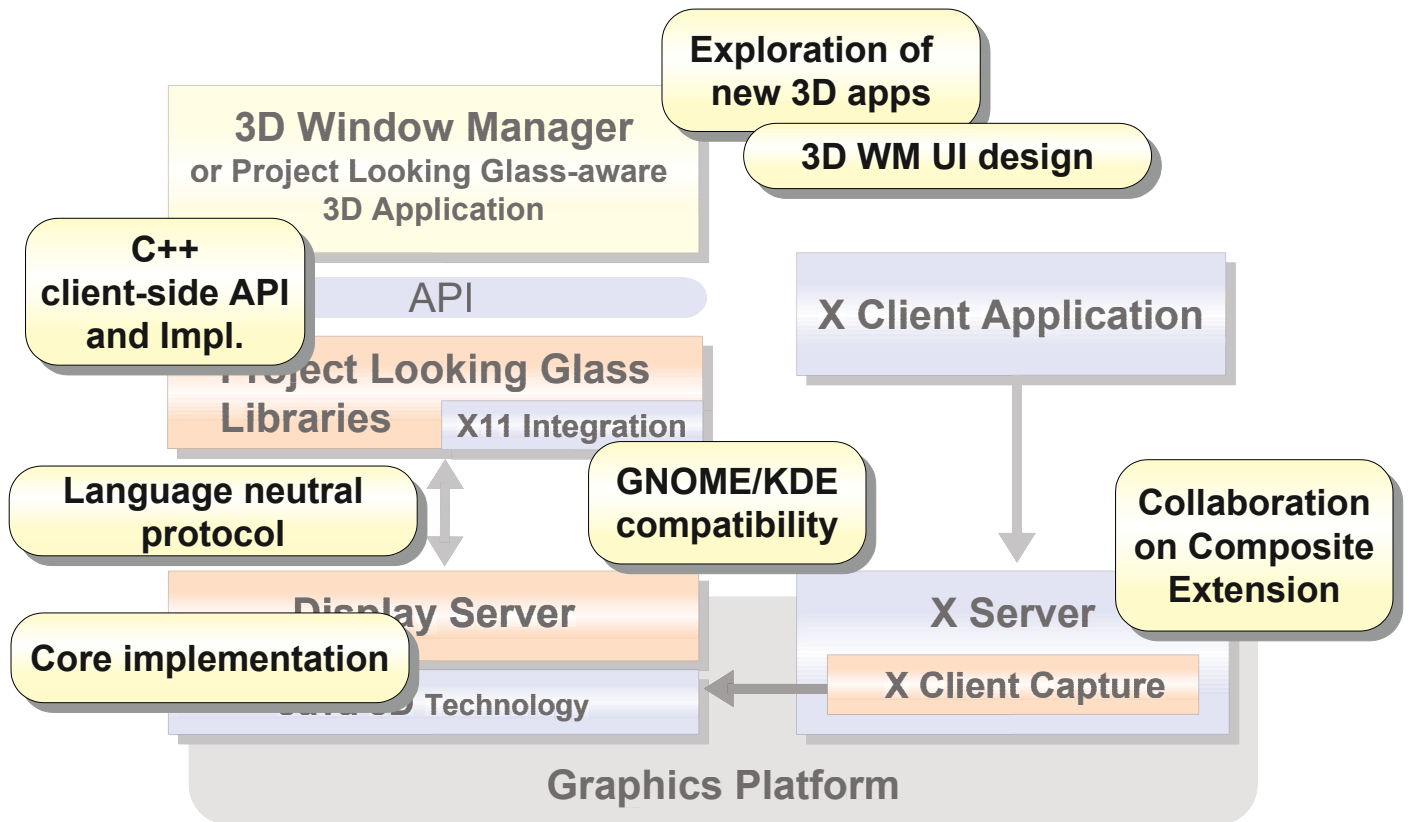
Moving Forward...

Let's collaborate!



Moving Forward...

Let's collaborate!

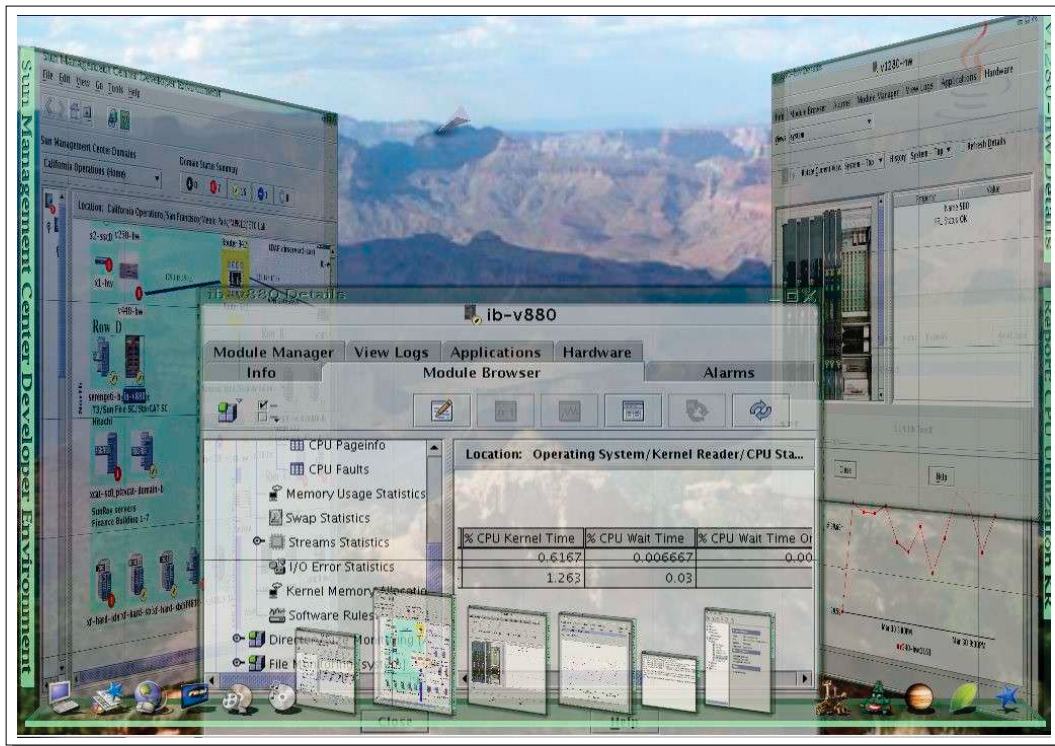


Moving Forward...

Let's collaborate!

- Graphics platform (short term)
 - Composite extension
 - Direct Render-to-Texture (both X11 and OpenGL)
 - Interprocess shared textures
 - Event Redirection and Xevie
- Experiments around 3D (long term)
 - 3D WM, apps, Widget Set
 - GNOME/KDE to leverage Project Looking Glass for 3D enhancements
 - Language independence support (C++ binding, UNO integration?)
- Suggestions...?

By the way...



How would this look on today's 2D desktop?

By the way...



Would you believe this is exactly the same software?

Q&A

Project Looking Glass:

3D Desktop Exploration...

Deron.Johnson@sun.com

Hideya.Kawahara@sun.com

