

# **Графические Системы. Часть II**

## **Лекция № 8**

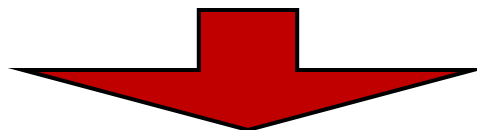
**Развитие и применение**

**основных концепций создания GUI,**

**сформулированных в стандарте X11**

# Развитие и применение основных концепций X Window System.

Итак, как показывает многолетняя практика развития концепций **X Window System**, этот стандарт доказал свою жизнеспособность и актуальность.



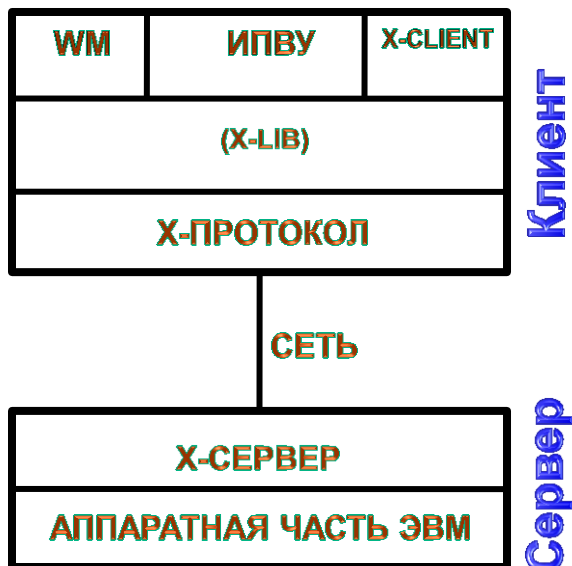
**X Window System** — оконная система, обеспечивающая стандартные инструменты и протоколы для построения графического интерфейса пользователя.

**X Window System** обеспечивает базовые функции графической среды: отрисовку и перемещение окон на экране, взаимодействие с мышью и клавиатурой.

**X Window System** не определяет деталей интерфейса пользователя — этим занимаются менеджеры окон, которых разработано множество.

В **X Window System** предусмотрена *сетевая прозрачность*: графические приложения могут выполняться на другой машине в сети, а их интерфейс при этом будет передаваться по сети и отображаться на локальной машине пользователя (в случае если это разрешено в настройках).

В контексте **X Window System** термины «клиент» и «сервер» имеют непривычное для многих пользователей значение: **«сервер»** означает локальный дисплей пользователя (*дисплейный сервер*), а **«клиент»** — программу, которая этот дисплей использует (она может выполняться на удалённом компьютере).



В настоящее время выпущена версия **X11R7.5**, базирующаяся на **X.Org Server** ("X.Org Foundation Open Source Public Implementation of X11") — свободной реализации сервера **X Window System** с открытым кодом. Текущая стабильная версия 1.16.2 является частью **X11R7.7**; выпущена в ноябре 2014.



# Развитие и применение основных концепций X Window System.



Клиент

Сервер

**X.Org Server** применяется во многих вариантах **Linux** и **BSD**, за исключением **NetBSD**, где используется **XFree86**. В **Solaris** на платформе **x86** также часто используется **X.Org Server**, однако в **Solaris SPARC** обычно используют **Xsun**, проприетарный **X-сервер** от **Sun Microsystems**.

В будущих версиях **OpenSolaris** для обеих платформ планируется перейти на **X.Org Server**. В **Microsoft Windows** сервер **X.Org** можно использовать в реализации **Cygwin/X**, **Xming** и др. В **Mac OS X**, начиная с версии 10.5 ("**Leopard**") включён **X-сервер** на основе **X.Org**.

**XFree86** — реализация сервера X Window System, графическая оконная система, используемая в операционных системах **UNIX**, **GNU/Linux**, и частично в **Mac OS X**. По сути, является воплощением **X Window System** в **open source** варианте.

**XFree86** в настоящее время практически не используется — большинство открытых операционных систем приняло **X.Org Server** вместо **XFree86**, а большинство разработчиков **XFree86** перешло к **X.Org Server**. Обычно, прикладные программы работают не непосредственно с **X Window** (в частности, **X.Org Server**), а с графическими средами, наиболее распространёнными на данный момент в сообществе **GNU/Linux** - **GNOME** и **KDE**. Последняя версия - **KDE SC 4.14.2** от 4 октября 2014 г.



# Развитие и применение основных концепций X Window System



Клиент

Сервер

Но наибольший интерес в плане развития идей человеко-машинного интерфейса представляют из себя разработки в области принципиально нового подхода к этой проблеме – а именно создание трехмерного **GUI**.

До сих пор разработчики графических операционных сред и наборов управляющих элементов для прикладных программистов **GUI** мыслили двумерно: графический интерфейс оформлялся в виде двухмерного Рабочего стола, на который выкладываются плоские же окна приложений. Соответственно, развитие оконного интерфейса за это время было связано, в основном, с той или иной стилевой проработкой пиктограмм, тем или иным цветовым решением базовой палитры или «выразительностью» обоев – задника корневых окон.

Ситуация стала меняться в начале XXI века. Речь идет о концептуальном проекте **Project Looking Glass** трехмерного пользовательского интерфейса рабочего стола, предложенного корпорацией **Sun Microsystems**.

Подробно с особенностями и наиболее интересными сторонами этого проекта вы сможете познакомиться самостоятельно с помощью тех материалов, которые выложены на сайте (раздел «Полезные ссылки»):

<http://www.osp.ru/os/2004/09/184573/>

[http://a0601.narod.ru/Progress\\_of\\_3d\\_interfaces.html](http://a0601.narod.ru/Progress_of_3d_interfaces.html)

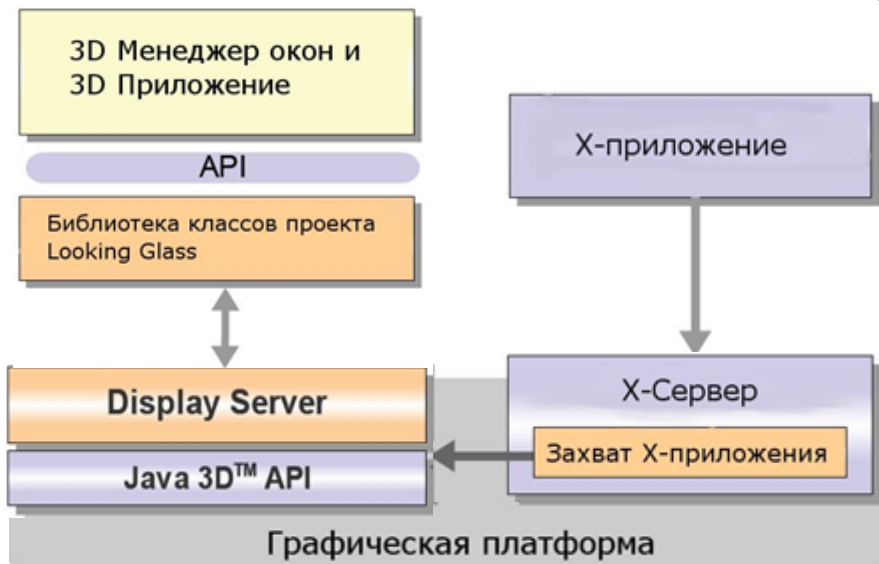
Рассмотрим некоторые характерные особенности развития трехмерных графических интерфейсов в плане развития идей и концепций **X Window System**.

# Развитие и применение основных концепций X Window System.



Для нас важно, что пионерские разработки в области создания средств для проектирования трехмерных **GUI** велись на основе типовой архитектуры **X Window System**.

Концептуальность проекта состояла в том, что он не являлся «плановым» развитием ни одной из массовых оконных сред (**CDE, KDE, GNOME**), применяемых в персональных компьютерах, и реализован заново на базе **Java**. Это принципиально отличало проект в плане переносимости - теперь он мог быть развернут на любой платформе.



По сути дела традиционный, **X-клиент** приложения взаимодействует с чуть измененным, по сравнению с традиционным вариантом, **X-сервером**. Последний дополнен серверным модулем, названным **X Client Capture**. Функциональность этого модуля сводится к тому, чтобы перехватить графическое представление традиционного **X-клиента** и передать его на интерпретацию расширенным средствам трансформации плоского изображения в объемное.

Этим преобразованием базовых **widget'ов** традиционных двухмерных **X-клиентов** занимаются модули **Looking Glass Display Server** и библиотеки примитивов трехмерного интерфейса.

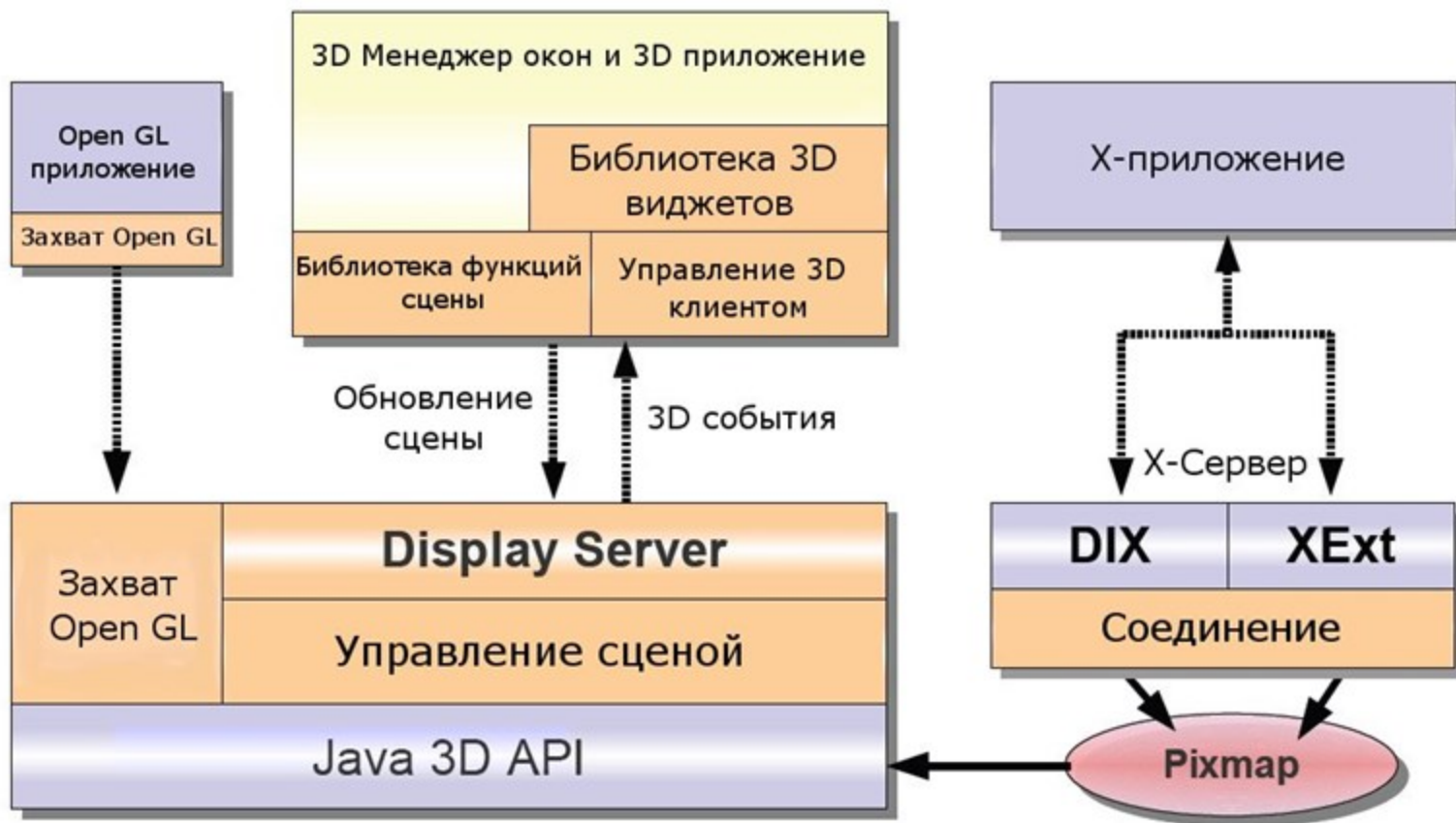
Выводом на экран результатов в виде объемных объектов занимается **3D Window Manager**, также написанный на **Java**, заменивший собой привычные «плоские» оконные менеджеры современных операционных систем. **Display Server** активно использует платформу **OpenGL** для трехмерных сцен.

На следующих двух слайдах эта архитектура конкретизируется.

# Развитие и применение основных концепций X Window System.



Подробная архитектура **Looking Glass**. Здесь более чётко отображена роль библиотеки **OpenGL** и событий, вызванных действиями среды.



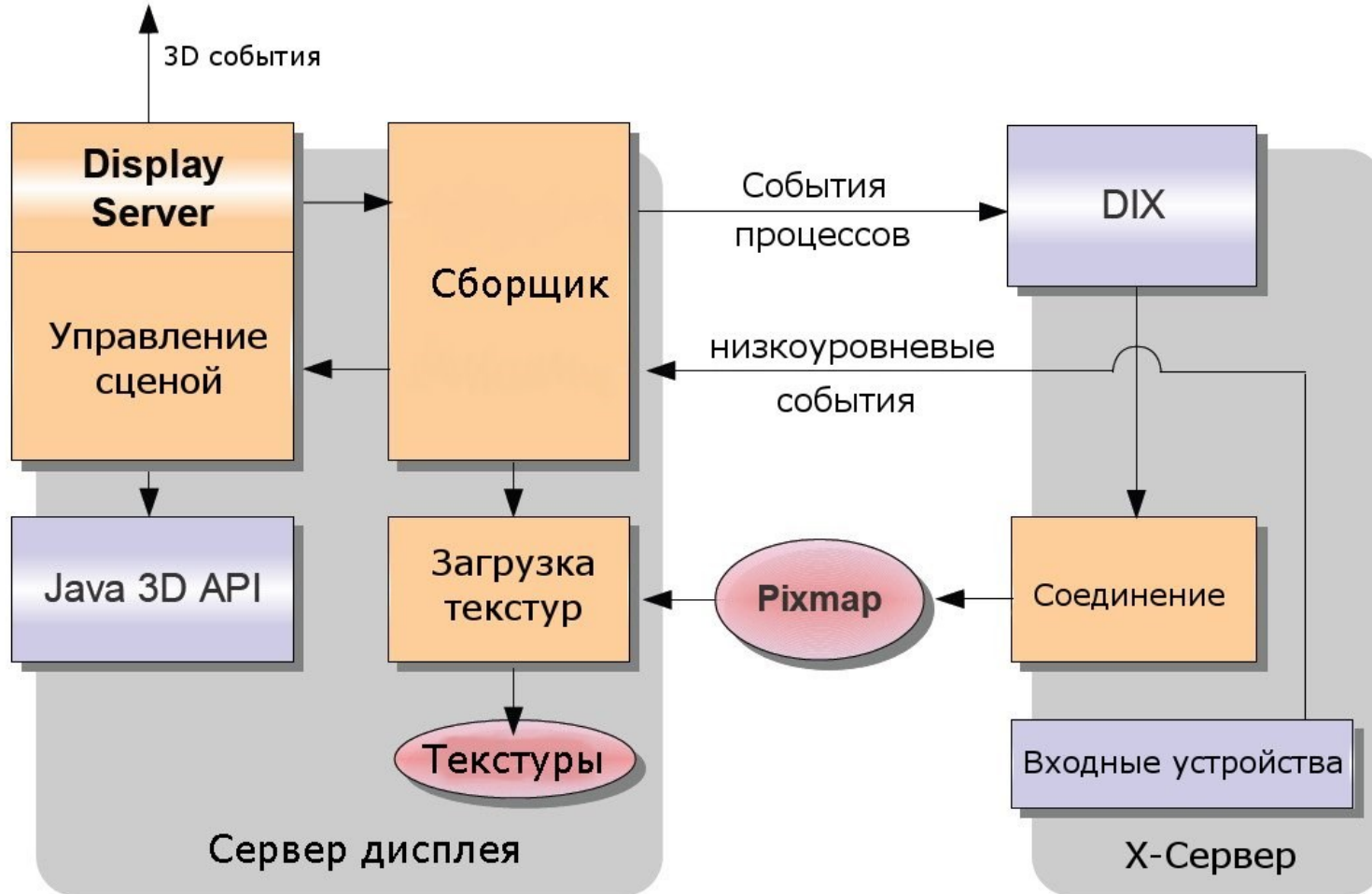
**Device-Independent Layer (DIX)** – Включает функции, не зависящие от графического оборудования.

**X Extensions library routines (Xext)** – расширения **X-сервера**.

# Развитие и применение основных концепций X Window System.



Здесь схематично представлен процесс обработки событий, поступающих от внешних устройств, а также генерация процессов, происходящих в системе, и реакция среды на эти события.



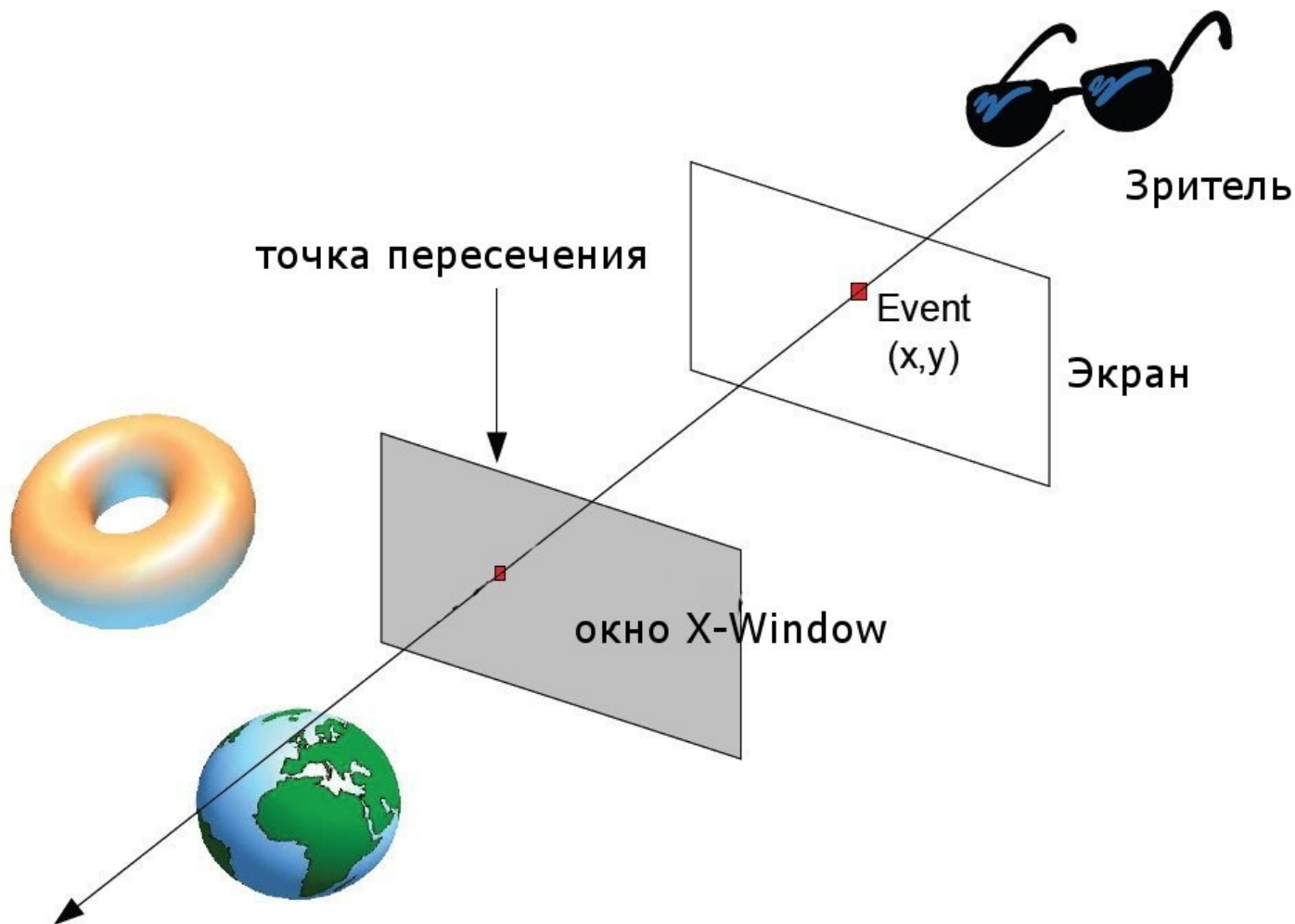
**Device-Independent Layer (DIX)** – Включает функции, не зависящие от графического оборудования.



# Развитие и применение основных концепций X Window System.



При указании пользователем в ту или иную область экрана при «прокрутке» псевдо-объемных обоев корневого окна, происходит событие (**user event**), которое необходимо однозначно интерпретировать прежде всего с точки зрения определения 3D-координат, на которые указал **User** в 2D-приложении. Схема подсчета координат приведена ниже:

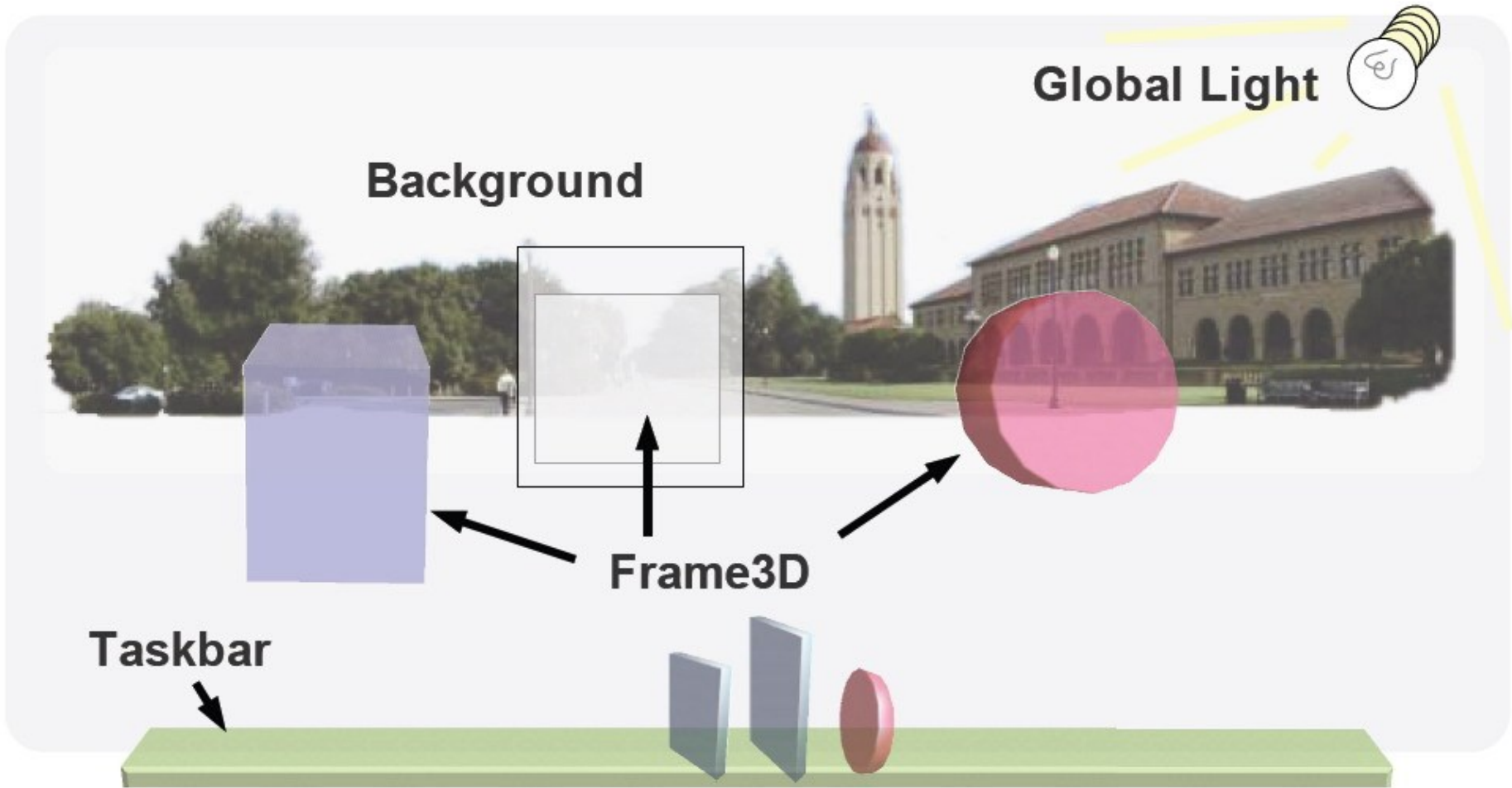




# Развитие и применение основных концепций X Window System



В проекте **Looking Glass** основе **Java3D** были дописаны классы **widget'ов** для отображения объёмных окон (**Frame3D**), включающий классы отображения кнопок, полей ввода (**Component3D**), «подоконников» – областей тумбнейлов и иконок (**Taskbar**) и т.д. Ниже представлен пример модели **3D сцены**, на которой управляющие элементы трехмерного интерфейса – **widget'ы** отображаются с помощью инстанций классов, реализованных в проекте **Looking Glass**.



# Развитие и применение основных концепций X Window System.

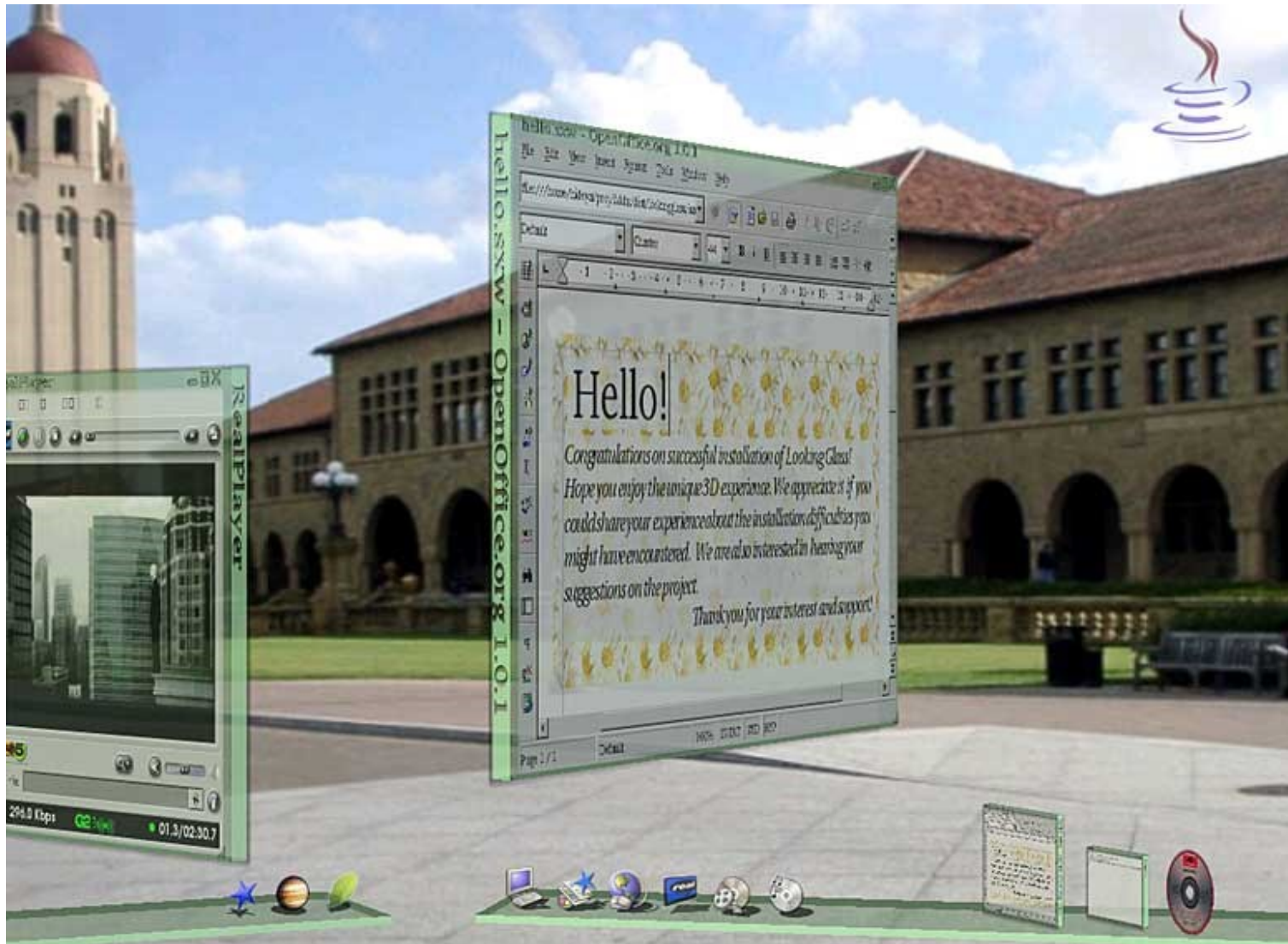
Так в чем же прелесть трехмерного общения, которую гарантировал **Looking Glass?**



В данном проекте все существующие «плоские» программы приобретают объёмное представление.

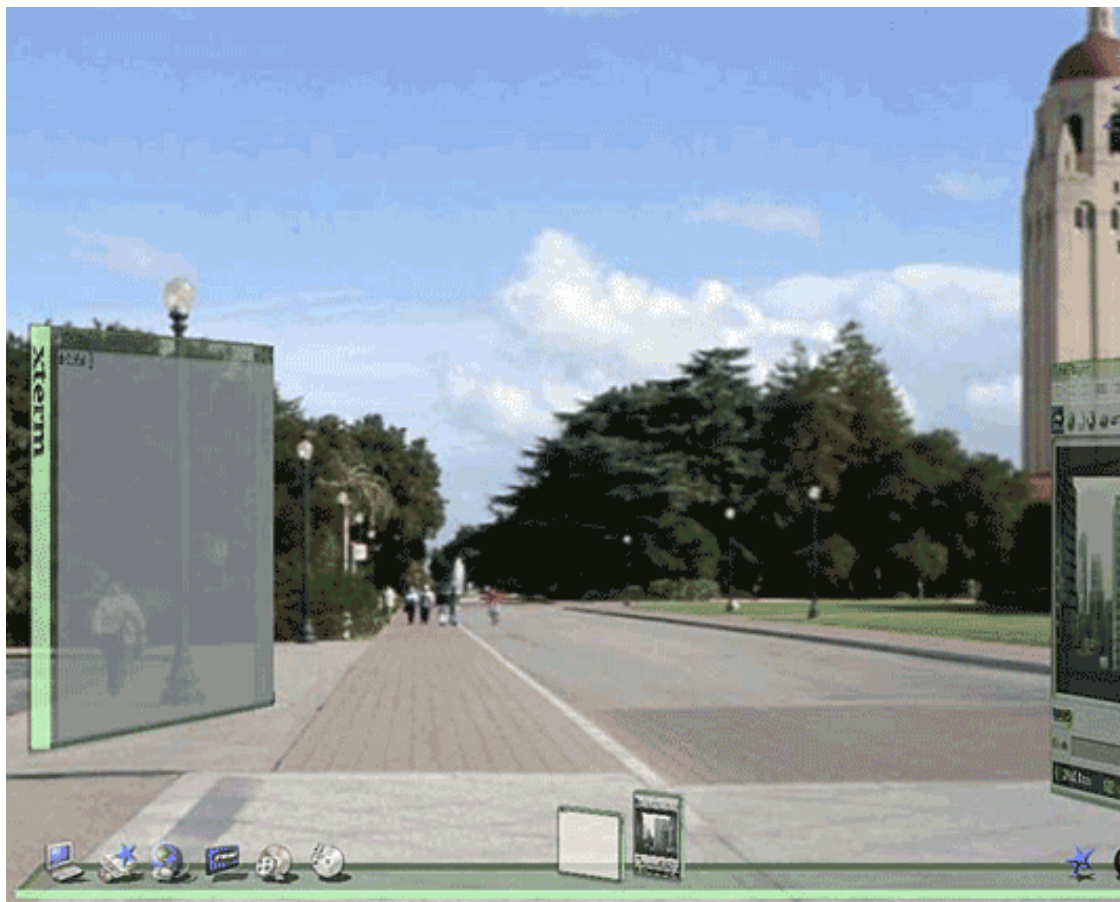


# Развитие и применение основных концепций X Window System

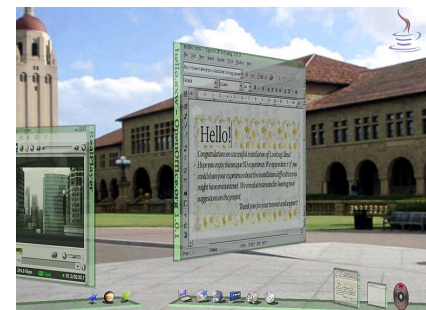


Рабочее пространство в **Looking Glass** - это именно пространство, составленное из нескольких подпространств. Каждое пространство имеет псевдообъемные обои, полупрозрачную «заваленную» наружу от пользователя панель задач и позволяет перемещаться вправо и влево относительно неподвижной в подпространстве панели задач.

# Развитие и применение основных концепций X Window System.



При перемещении внутри одного подпространства включается визуальный эффект «перспективы», а панель задач «передвигается» относительно обоев, имитируя эффект пространственной разнесенности панели и обоев, которые теперь уже превращаются в подоконник и вид-за-окном. Если пользователь «зайдет» слишком далеко, то будет перемещаться из одного подпространства в соседнее подпространство (из одного сектора пространства в соседний сектор).



В принципе, в традиционной оконной системе (например, **GNOME** или **CDE**) пользователь также имеет возможность переходить от одного рабочего стола к другому, но делается это дискретно. В **Looking Glass** это выполняется непрерывно, и возможны ситуации, когда пользователь может занять положение между двумя рабочими подпространствами, присутствуя одновременно в половинках обоих подпространств. Для окончательного вывода пользователя из двухмерного плена, «обои» обычно выполняются в виде стыкующихся частей круговой панорамы.



# Развитие и применение основных концепций X Window System.



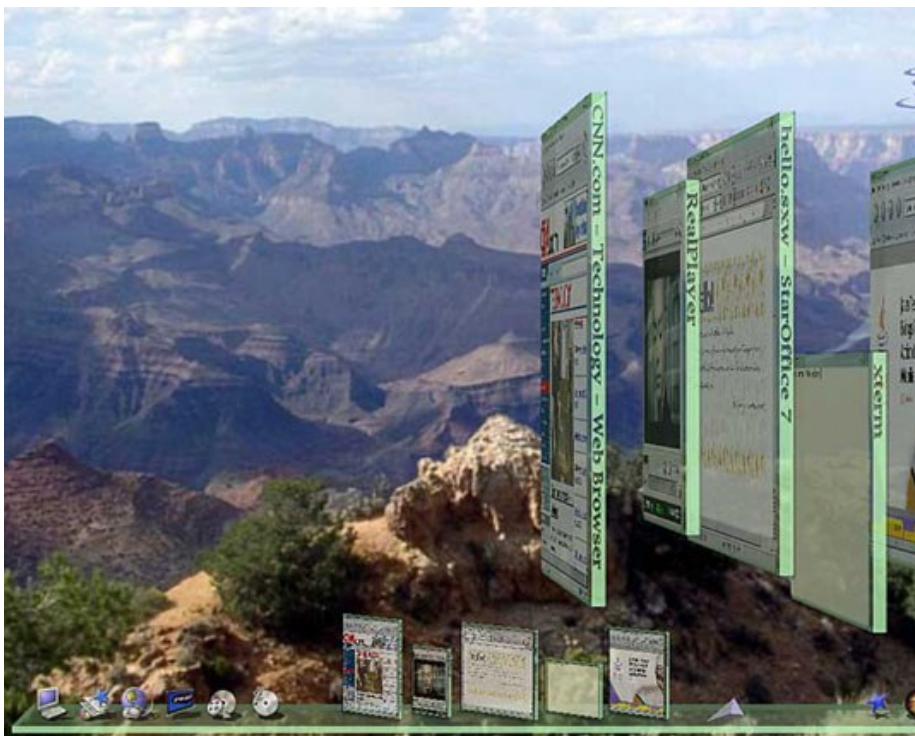
Каждое приложение, даже элементарный **X-терминал**, выводится на рабочее подпространство в объемном окне, которое можно, как и раньше, передвигать по подпространству и «схлопывать», но можно распахивать как «ставню» наружу.



При этом приложение, если оно активно, будет иметь возможность выводить в такое «полузакрытое» и занимающее мало места «окно» («ставню») любую графическую информацию: масштабирование и пространственную деформацию берут на себя библиотечные модули и оконный менеджер **Project Looking Glass**.

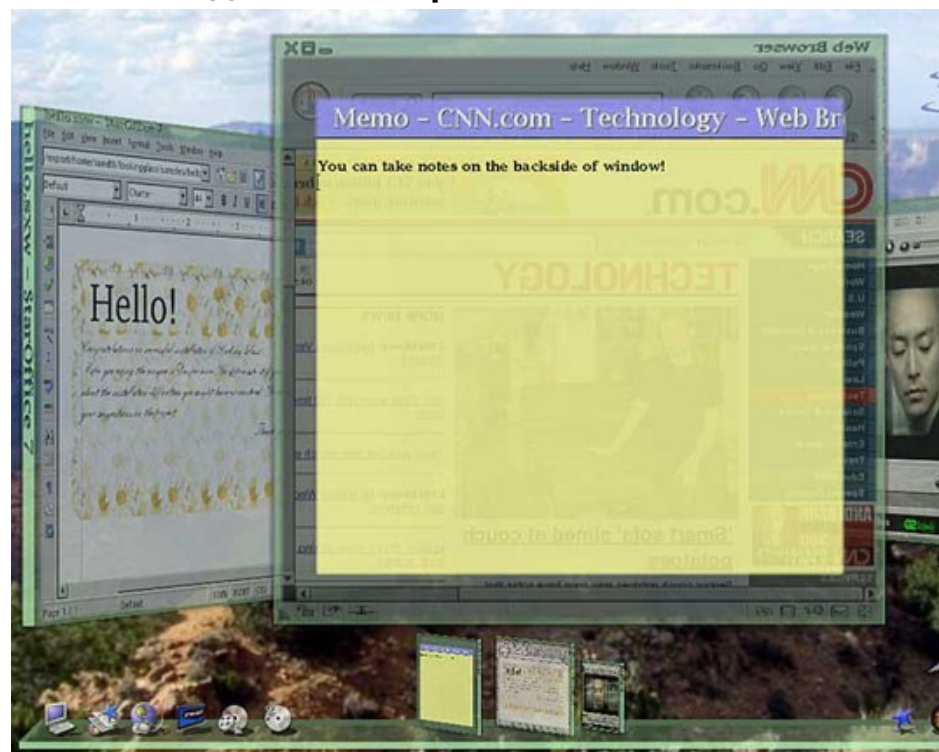
Для того чтобы пользователь знал, что за приложение он оставил работать на той или иной «ставне», торцы этих «ставен» несут на себе наименование приложения. А еще традиционная пиктограмма перестала быть пиктограммой — она обрела «внутреннее содержание», и теперь в ней «живет» приложение.

# Развитие и применение основных концепций X Window System.



Ставни и активные окна могут менять свою «непрозрачность», чтобы было видно, что там «за окном». Очень удобно совместное использование настроек тыла полупрозрачности окна — пользователь не теряет «из виду» и само приложение (пусть и в зеркальном отображении), и ход исполнения настроек.

В принципе, и икону и «Ставню» можно повернуть на любой угол относительно вертикальной оси, поэтому естественно, что «тыльная» часть такого трехмерного окна также не может оставаться не задействованной. Для «тыла» предусмотрено «наклеивание горчичников», на которые можно наносить любую текстовую информацию. Основное назначение тыла - настройка работы приложений путем вызова многочисленных меню и подменю настроек.





# Развитие и применение основных концепций X Window System.

Но, к сожалению, если сегодня выйти на страницу проекта Looking Glass, то на самом видном месте обнаружится следующий транспарант:

**NOTICE: THIS PROJECT IS INACTIVE  
(PRACTICALLY DEAD) since late 2006.**



После старта коммерческого проекта **Windows Aero** для **Microsoft Windows Vista**, в рамках которого было амбициозно заявлено, что все самые передовые наработки в области **3D-интерфейсов** будут в наилучшем виде реализованы в **ОС Microsoft Windows Vista**, и отказа компании **Sun** от использования **Looking Glass** в качестве поставляемого интерфейса в **ОС Solaris 10**,

дальнейшее развитие проекта на уровне некоммерческой инициативы потеряло практическую привлекательность.

Однако основные действующие лица команды разработчиков **Looking Glass** продолжили работу в области развития **3D – интерфейсов** в рамках другого проекта – **Wonderland**.

Как уже было сказано, проект был организован основными разработчиками проекта **Looking Glass** и вобрал в себя основные его идеи и инновации. Главное - аналогично **Looking Glass** среду и инструментарий **Wonderland** можно использовать для интеграции **X11-applications** в **3D-среду**, а также создания собственных **3D-applications**.



# Развитие и применение основных концепций X Window System.



Самая главная особенность разрабатываемого проекта заключается в том, что **Wonderland** – это, в первую очередь, не локальный, а многопользовательский интерфейс, предназначенный для поддержки сетевых коммуникаций. Иными словами, если на моей машине установлен интерфейс **Wonderland**, то я могу средствами этого интерфейса общаться с другими пользователями в виртуальном пространстве **Wonderland'a**, используя голос, видео, текст, изображения и приложения.

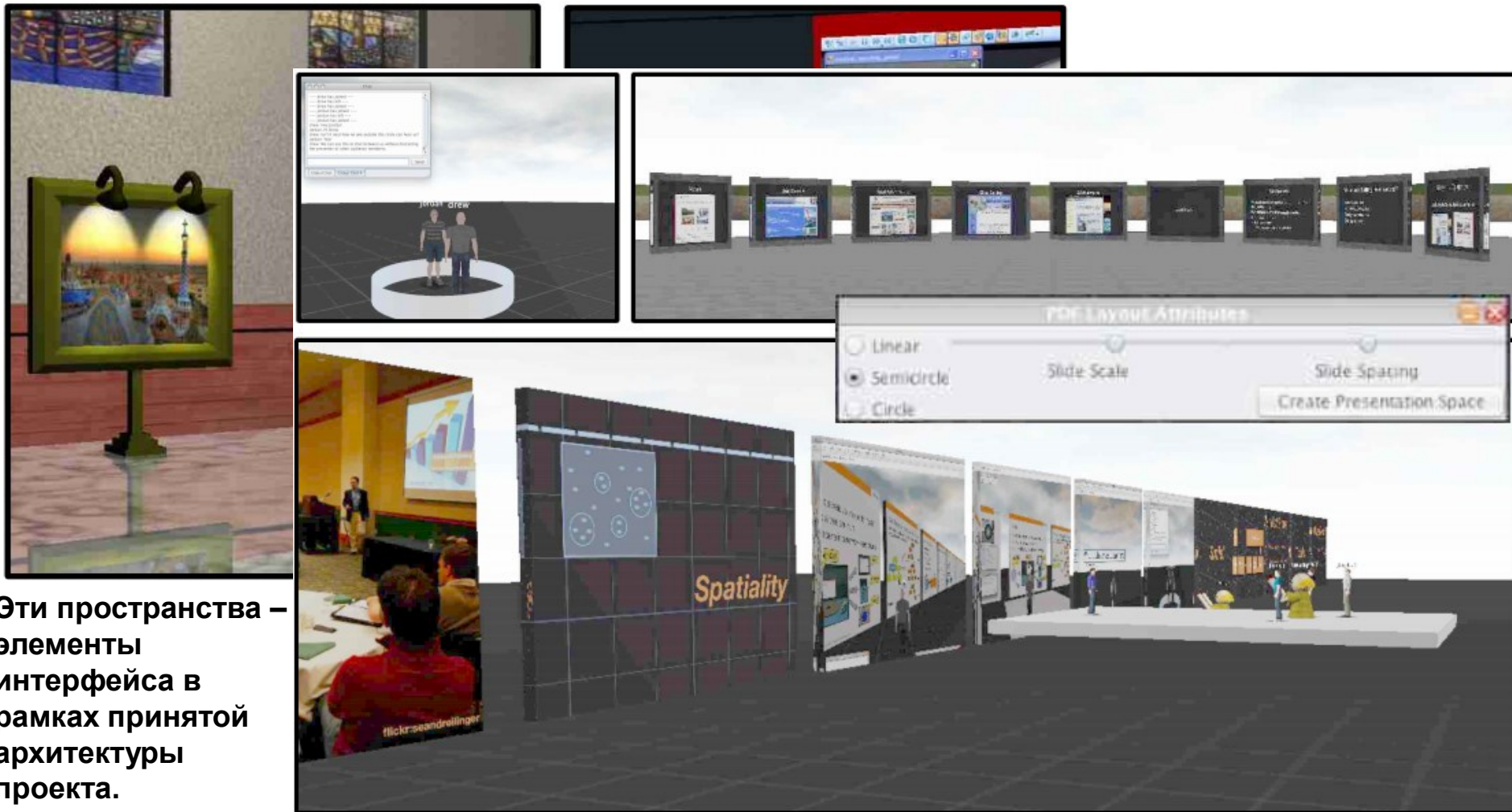
Забегая вперед, хочу продемонстрировать, каким комплектом программно-технического оборудования был обеспечен каждый преподаватель **MTI**, участвовавший в тестировании рабочей версии **Wonderland**.

Мы видим здесь «объемный звук» (**Immersive Audio**), как входной так и выходной, камеру с 180-градусным обзором, систему передачи широкополосного сигнала по телефонным линиям (**+VoIP**), дополнительное оборудование для подключения к сетевым устройствам. **MO** – это версия **Wonderland MPK20**.

# Развитие и применение основных концепций X Window System.

Сеть, со всеми необходимыми для пользователя сервисами, представляется в виде привычных для нормального человека пространств, «миров», например – музейный зал с картинами по стенам, зал заседаний, тренинг-центр с тренажерами, учебный класс, библиотека.

Один из элементов интерфейса – это **AVATAR**, «объемный курсор» - индикатор присутствия пользователя в системе, знакомый нам «клиент» и «сервер», вернее – сетевой «сервис» с индикатором его владельца.



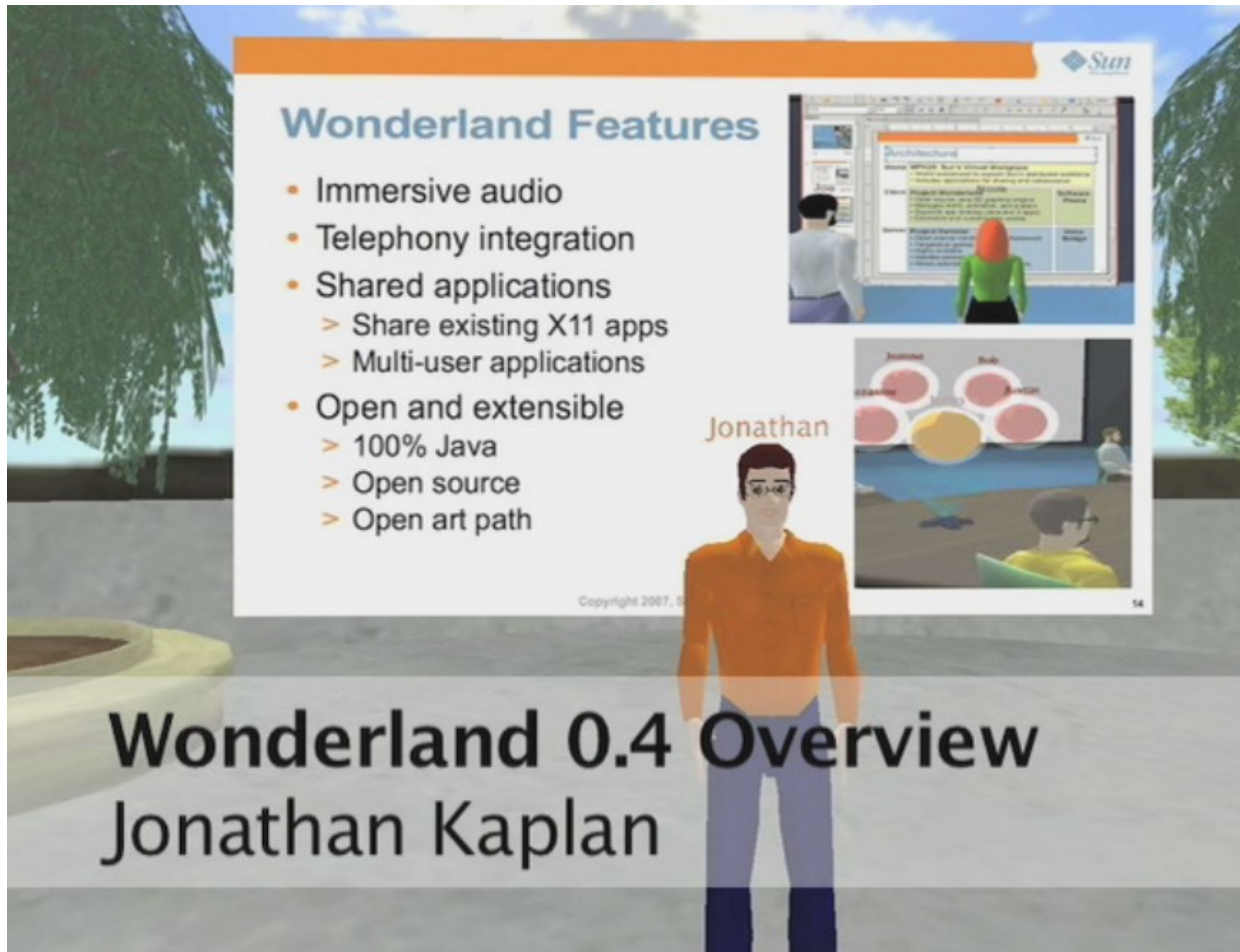
Эти пространства – элементы интерфейса в рамках принятой архитектуры проекта.

# Развитие и применение основных концепций X Window System.

Особенности проекта:

➤ Концепция – легкое средство для обеспечения интерактивного взаимодействия пользователей в рамках одного комьюнити с очевидным доступом ко всем сетевым сервисам.

➤ **X-Window System** – одна из неизменяемых основ этого проекта. Т.е. – если нужны плоские приложения и графический интерфейс для работы с ними, то используют **X11**.



The image shows a 3D-rendered presentation slide titled "Wonderland Features" with a Sun logo in the top right corner. The slide lists the following features:

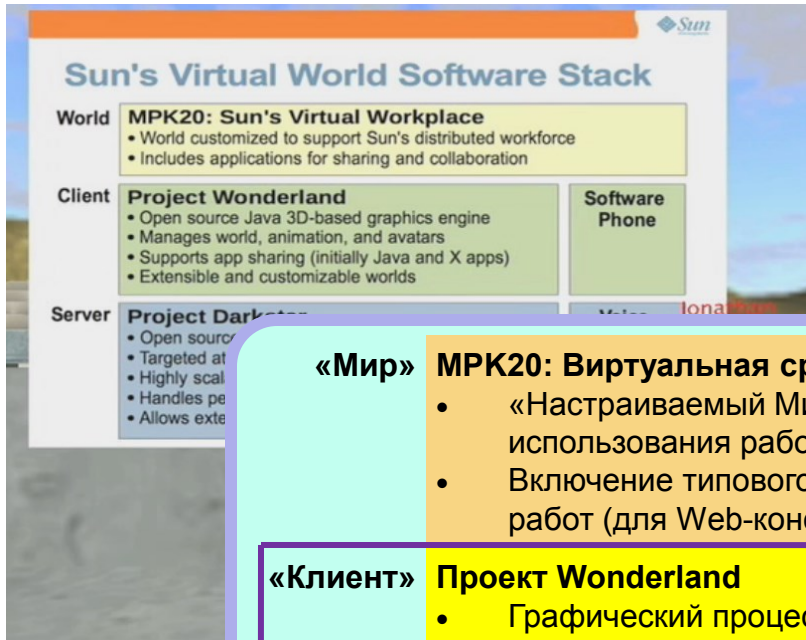
- Immersive audio
- Telephony integration
- Shared applications
  - > Share existing X11 apps
  - > Multi-user applications
- Open and extensible
  - > 100% Java
  - > Open source
  - > Open art path

Below the text, there are two small inset images: the top one shows two avatars (one white, one red) looking at a computer screen displaying a document; the bottom one shows a 3D character named "Jonathan" in an orange shirt standing in a virtual environment. At the bottom of the slide, the text "Wonderland 0.4 Overview Jonathan Kaplan" is visible. A copyright notice "Copyright 2007, 1" is also present.

➤ Открытая архитектура и 100%-ая реализация на **Java**, т.е. максимальное обеспечение переносимости, платформно-независимости и масштабируемости.



# Развитие и применение основных концепций X Window System.



## Архитектура проекта (**Software Stack**):

Собственно в рамках проекта разрабатывались выделенные на схеме функции:

### «Мир» **MPK20: Виртуальная среда рабочего места от Sun**

- «Настраиваемый Мир» для поддержки технологии от Sun для распределённого использования рабочей силы (User+необходимый ему сетевой ресурс)
- Включение типового набора приложений для совместной работы и распределения работ (для Web-конференций один, для тренингов – другой и т.д.)

### «Клиент» **Проект Wonderland**

- Графический процессор **Open source Java 3D-based** («движок»)
- Управляемое 3D-виртуальное пространство (мир), анимация и аватары
- Принципы и реализация распределения приложений (изначально Java и X приложения)
- Расширяемое и настраиваемое 3D-виртуальное пространство

Программная эмуляция телефонной связи

### «Сервер»

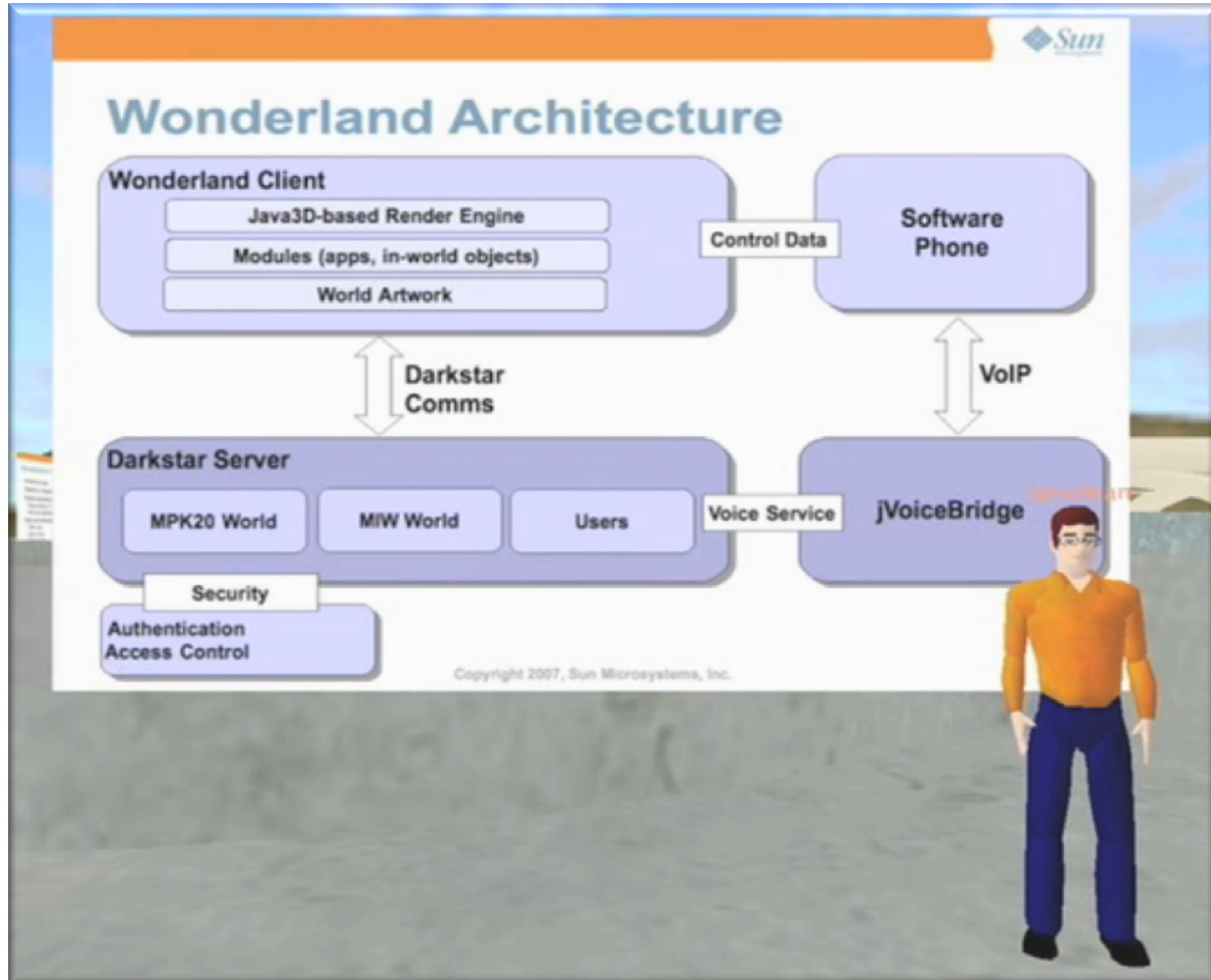
#### **Проект Darkstar** (готовая серверная игровая среда)

- Коммуникативные и прикладные инструменты
- Предназначение: игровая среда
- Наилучшая масштабируемость
- Управляемая надёжность
- Открытый набор расширений базовых сервисов

**jVoiceBridge** (специально дописанное в рамках проекта Java-приложение для работы с акустическими потоками)

# Развитие и применение основных концепций X Window System.

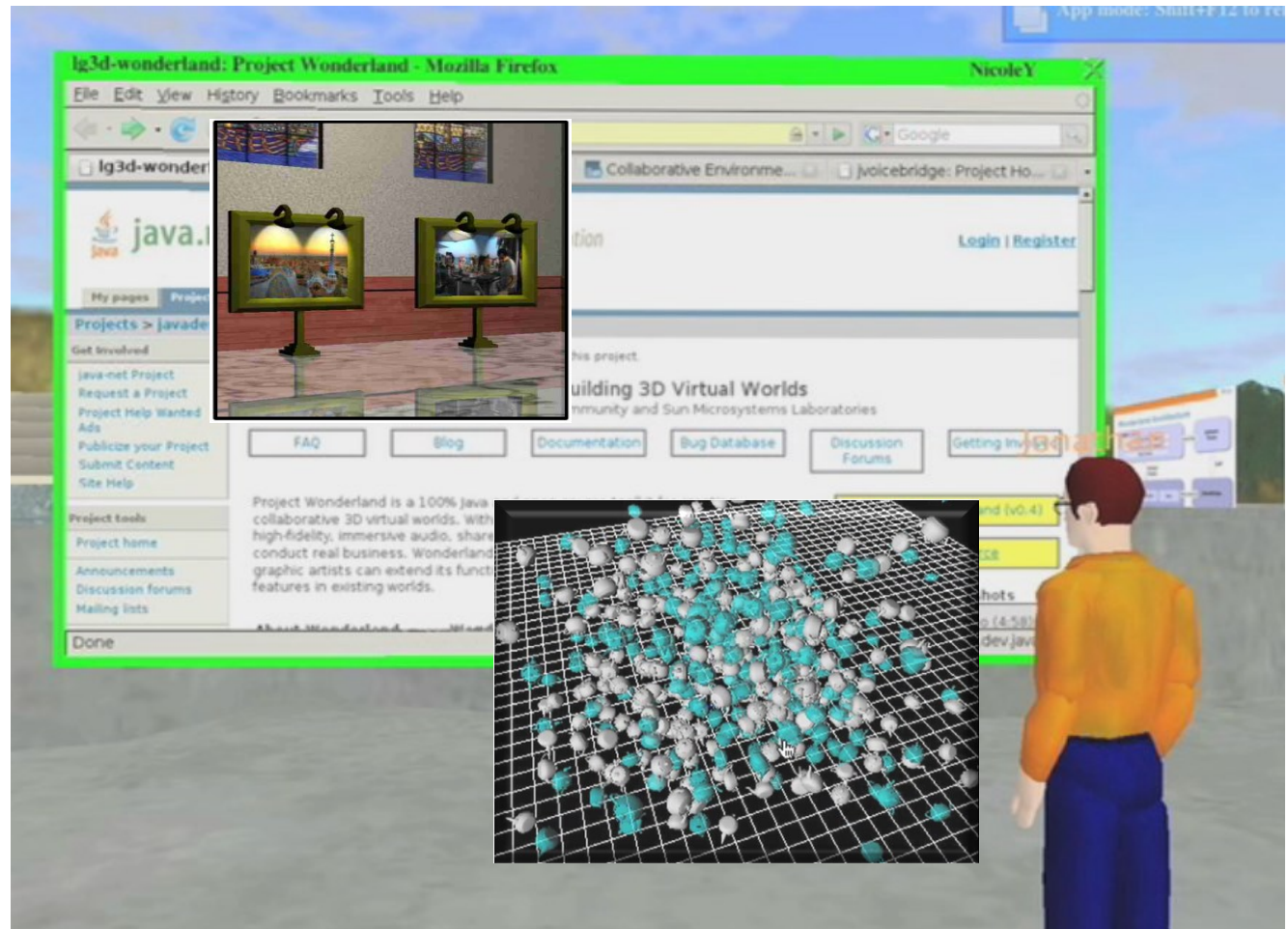
Архитектура проекта :



# Развитие и применение основных концепций X Window System.

Примеры: видеопрезентации по проекту вы можете найти по ссылке:  
[openwonderland.org](http://openwonderland.org)

А также в презентации [PDF\\_\\_Wonderland\\_in\\_the\\_Age\\_of\\_Immersive\\_Education.pdf](#)



# Развитие и применение основных концепций X Window System.



## **XGL**

- X-server архитектура, январь 2006
- Window manager
  - Beryl
  - Compiz (Novell)
- ICCCM Стандарт (Inter-Client Communication Conventions Manual)

**Среды рабочего стола GNOME, KDE – различные Linux-дистрибутивы (Mandriva, Debian, Ubuntu)**

**Для самостоятельного  
ознакомления!!**