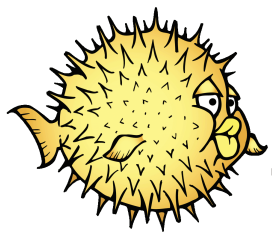
X.Org & BSD - Upcoming Plans

Matthieu Herrb

OpenBSD/X.Org



BSDCan, May 17 2008

http://www.laas.fr/~matthieu/talks/bsdcan2008.pdf

Agenda

Agenda

Introduction

- X is the underlying technology for all the ~~Linux~~ BSD on the desktop stuff.
- Often mis-understood by users (normal) and developers (less normal).

Goals of this talk:

- provide some information to *BSD developers on directions of the X window system
- attract some developers to X

Agenda

late 80’s - X11R5

- Mostly monochrome or 8 bit colour (indexed) frame-buffers
- Simple non-anti-aliased drawings
- Server side non-anti-aliased text rendering
- Athena widgets and Motif toolkits
- First steps at porting to the i386/PC architecture (X386, XFree86 2.1)

But relatively small and efficient for the epoch

90’s - X11R6

- Initial X11R6 release done by the X Consortium.
- No radical change in hw or core rendering model
- Lots of extensions and “peripheral” functionalities (Xprint, LBX, PEX, XIE, XKB, Xinput,..

XFree86 contributions :

- Modular X server architecture
- XAA: 2D hw acceleration
- DRI: 3D hw acceleration based on Mesa3D (OpenGL)

During this period, lots of code growth (and bloat) but few benefits for the end-users (focus on new hardware support).

Early 2000 - Xrender

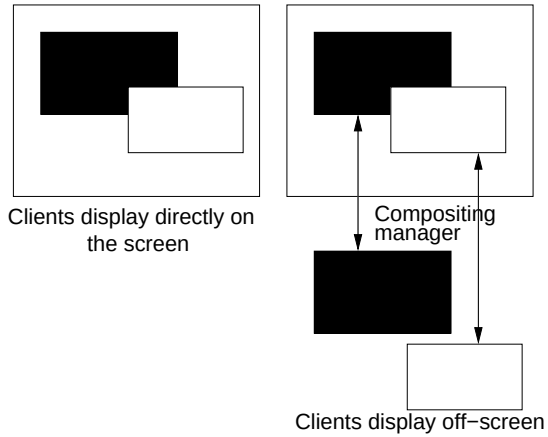
- Porter & Duff compositing inside a window
- fb + render code
- Font rendering in the client, with anti-aliasing
- KAA (EXA) new 2D hw acceleration framework
- Cairo new 2D drawing library

A political interlude

- Some developers unhappy with the XFree86 development model
- Put together the new X.Org foundation to resume X development
- XFree86 decides to change its license in a way that made it irrelevant in a couple of weeks...
- 1st visible change of the new X.Org foundation: change the build system to modular (autotools based).
- 2nd visible change : move repositories to **git**.

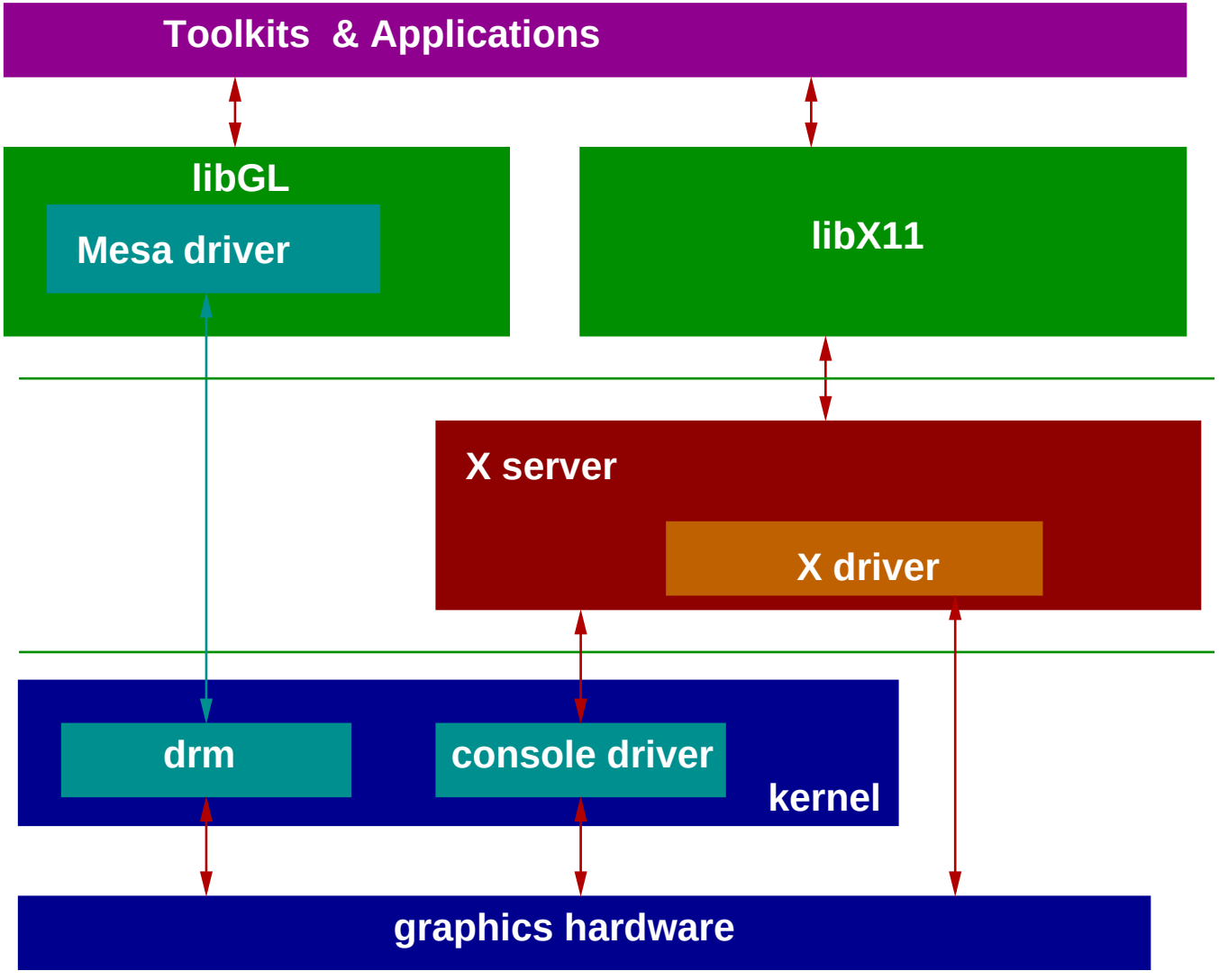
Mid 2000’s - Compositing desktop: Xgl & Compiz

- New desktop model: windows are rendered off-screen first and then ńcompositedż together on the desktop.
- Using 3D transforms between off-screen and the desktop, and hw-accelerated compositing based on OpenGL
  - Evolutions : AIGLX, removing the need for Xgl, better Xvideo support,...
  - XCB: new C bindings for the X11 protocol, based on an XML specification.

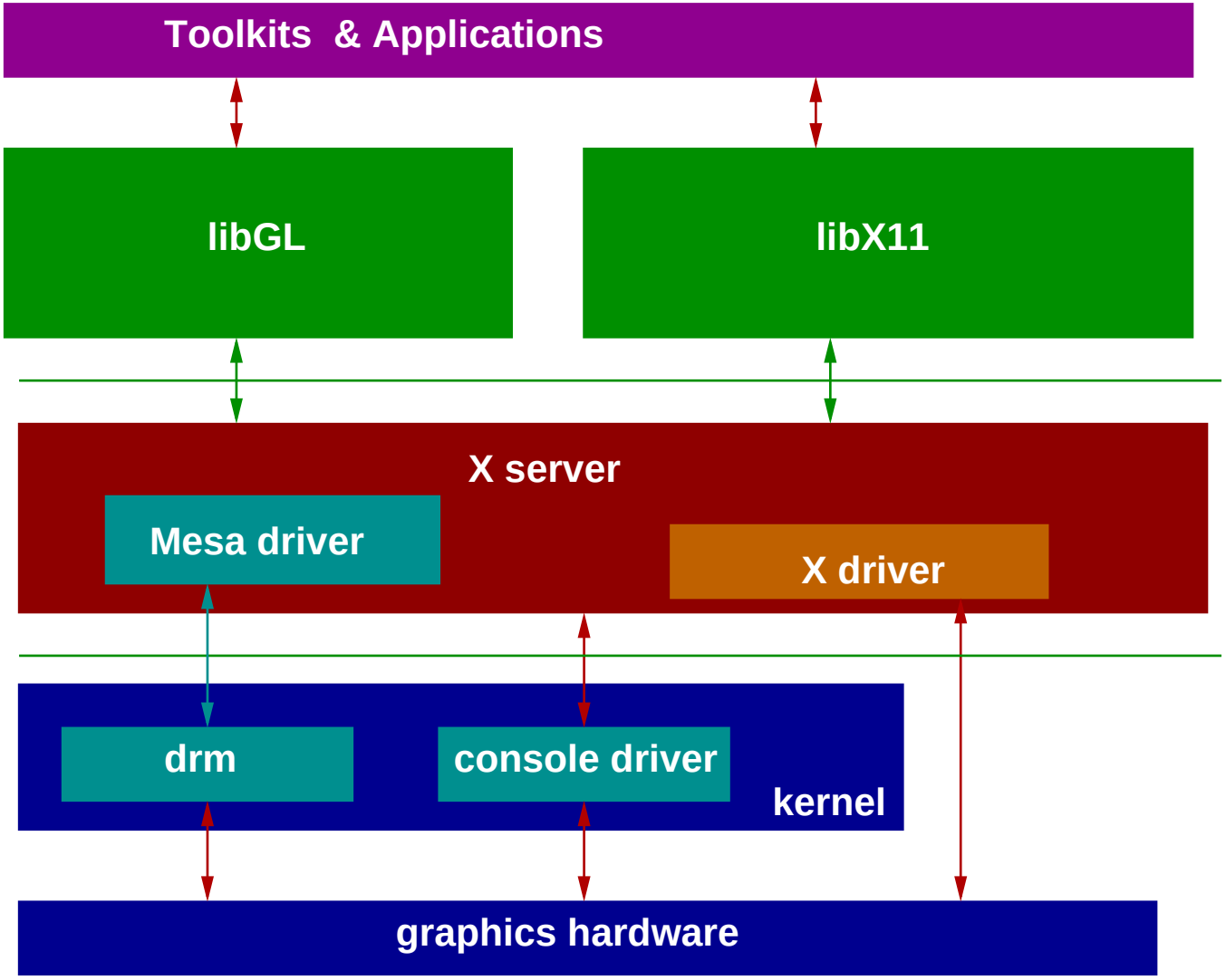


Agenda

X architecture - direct rendering



X architecture - indirect rendering



Last year: X.Org 7.3

New features in X.Org 7.3

- **XRandr 1.2**: dynamic management of screens
- new **pixman** library, to share many of the pixel-level compositing code between X server and other libs (Cairo) that need them.
- New intel driver 2.0, independent of the BIOS for mode-setting. Implements XRandr 1.2.
- Input hot-plug support and code clean-ups (breaks some existing drivers)
- Composite enabled by default
- XKB code cleanups
- Support for DTrace on Solaris
- Improvements in drivers and EXA code
- More auto-configuration capabilities

Coming soon: X.Org 7.4

- New **pciaccess** library for PCI bus access
- XACE replaces X Security extension
- Generalisation of XRandr 1.2 in drivers
- Radeonhd driver (based on newly available docs from AMD/ATI)

OpenBSD Status

- OpenBSD 4.3 (released May 1) includes X.Org 7.3.
- Work has started on porting DRI, based on NetBSD's port.
  - still beta, not enabled by default
  - almost all drivers work
- libpciaccess has been ported,
- We should be able to include X.Org 7.4 in OpenBSD 4.4.

Other BSDs

FreeBSD	7.0 includes Xorg 7.3
NetBSD	4.0 uses XFree86 4.6
pkgsrc	has xserver 1.3
DragonFly	1.12 uses pkgsrc

Some news from X.Org foundation

Current Board of Directors:

Eric Anholt	Egbert Eich	Matthieu Herrb	Adam Jackson
Bart Massey	Keith Packard	Daniel Stone	Carl Worth

On going work:

- XDC - Google campus, Mountain View Ca, april 2008
- XDS - Edinburgh Zoo, Scotland september 10-12, 2008
- New foundation membership agreement - boost membership
- New PR committee - better communication

X.Org - Google Summer of Code

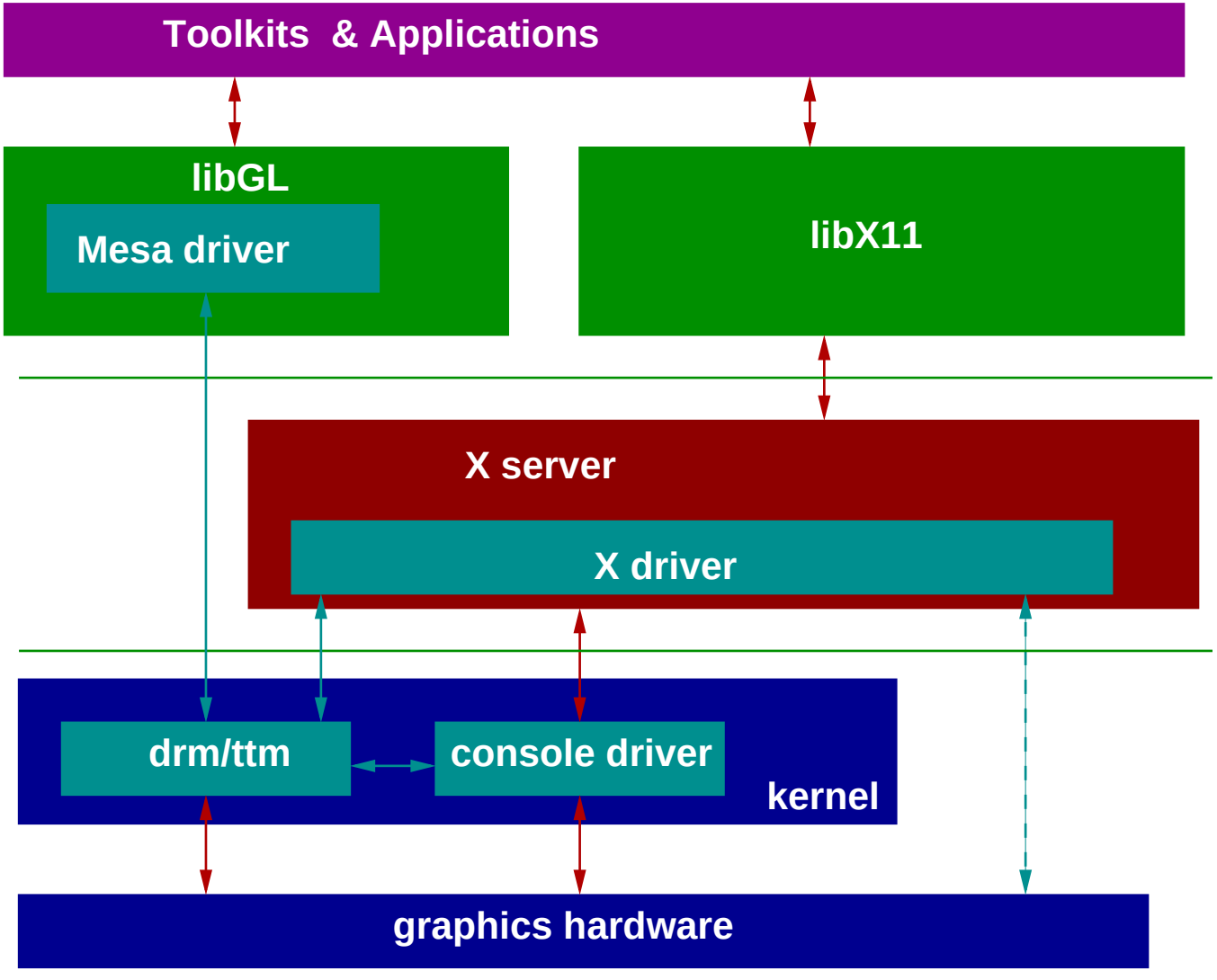
- Generic GPU-Accelerated Video Decoding
  Younes Manton Mentor: Stephane Marchesin
- Enhancing xcb-glx for wider usability
  Kristóf Ralovich Mentor: Bart Massey
- GUI Tool for assisted editing of the XKB configuration database
  Symeon Xenitellis Mentor: Sergey Udaltsov
- DRI/Mesa r300 Cleanups, Optimizations, and Features
  Erik Boettcher Mentor: Alexander Deucher
- Moving all the input code into a separate thread
  Tiago Vignatti Mentor: Daniel Stone

Agenda

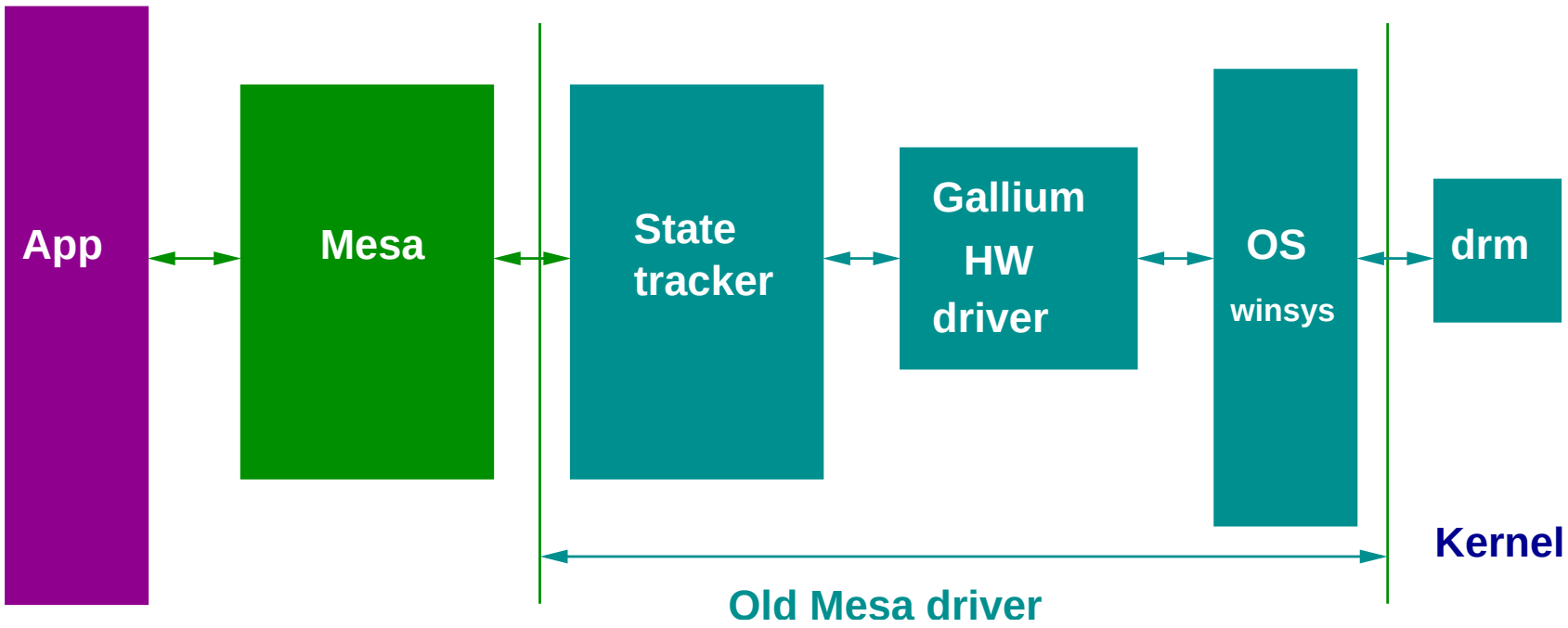
Work in progress

- X server internal API cleanups
- XRandR 1.3: support for several separate cards
- TTM (or GEM ?) - new memory management code in DRM, uses more Gart features, enables the kernel instead of the userland X server to manage video memory.
- DRI 2 + Gallium: new architecture for Mesa3D drivers. Closer to the hardware.
- MPX - multi-pointer X
- Input transforms for composite
- In-kernel mode-settings

X architecture - DRI2



Gallium



- make drm simpler / closer to modern 3D HW
- hw independent state trackers takes care of OpenGL

MPX

Multi **P**ointer **X**

- Virtual pointers - cursors
  - attached to zero or more physical devices
  - provide the events to the applications
- Virtual keyboards - focus
  - attached to zero or more physical devices
  - provides the events to the applications

To be merged in master next week.

Kernel-side mode-settings

Mode-setting: programming the graphics card into a given graphical mode (1280x1024 at 72Hz with 32 bits per pixel for example).

- requires access to the hardware registers of the card
- currently done by the driver code in the X server itself:
  - kernel doesn't know about the state of the video card, can't restore a text mode in case of a crash
  - duplication of work/code with graphical text modes ("framebuffer" mode)
  - security issues: privilege escalation to kernel privilege
- goal is to :
  - integrate mode settings in drm
  - let XRandR control all configuration
- some code exists for the intel and radeon drivers.
- Dave Airlie has an accelerated X server running without any privilege

*BSD TODO

- Input hot-plug HAL/DBus ?
- Porting more DRMs (nouveau, ...)
- TTM/Gem
- Framework for in-kernel mode settings
- Promote the MIT/BSD license

NetBSD/OpenBSD Legacy architectures

Problems:

- 8 bits or less displays
- slow CPUs and limited RAM
- gcc 2.95/a.out/no shared libs (OpenBSD)

X.Org evolution is slowly but firmly stopping to support those arches. (Modern embedded systems have 16bit or better)

Solutions:

- port kdrive to the BSD console drivers (partly done - to be debugged)
  - but kdrive needs gcc3 or better
- a new simple X server with only semi-legacy technologies (no Composite, no OpenGL, no fancy Xinput)?

Agenda

Conclusion

- X development is going well
- Again exploring the leading edges of user interfaces
- Challenge for *BSD to participate and not only follow

Some open issues:

- Colour management
- Hardware documentation for Open-Source drivers (nVidia,...)

Questions ?