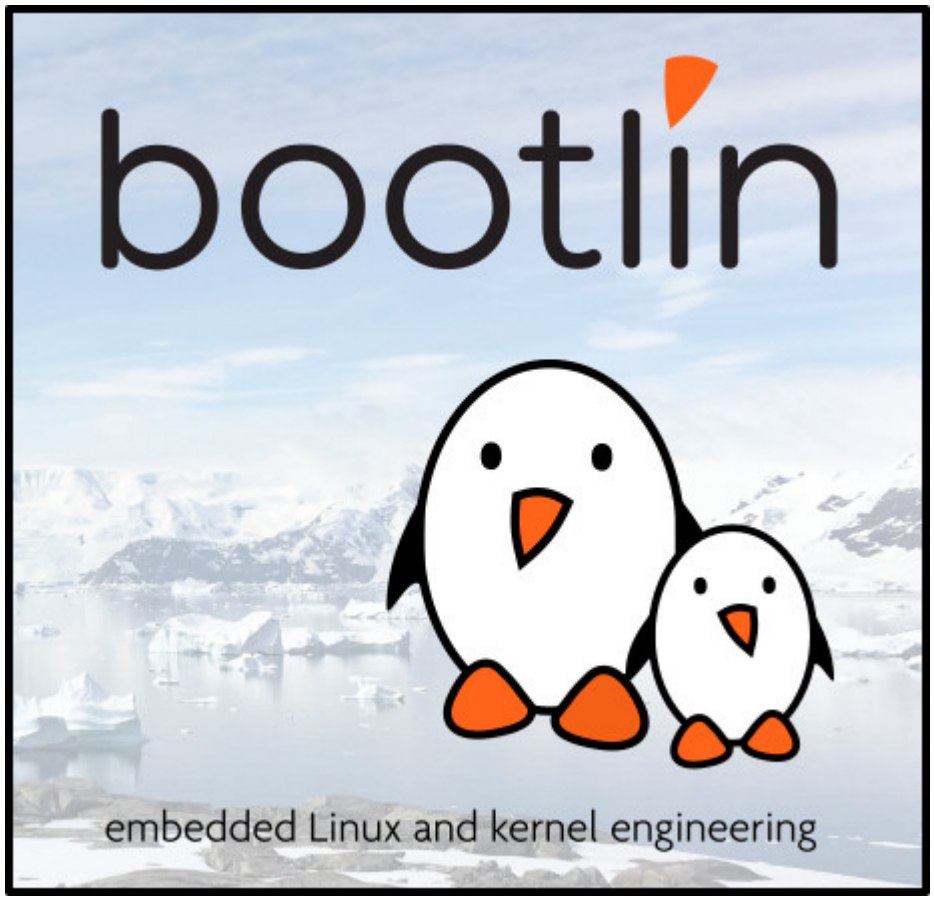
Choosing free software graphical libraries for embedded devices



Thomas Petazzoni
Bootlin
https://bootlin.com/

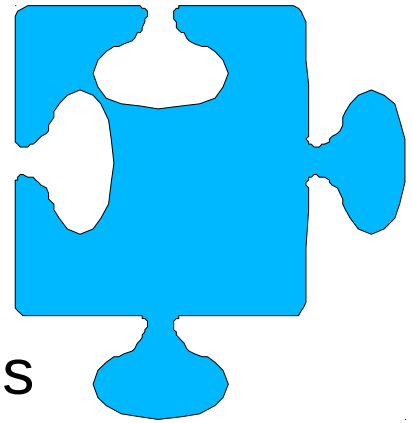
About this document

Documents updates can be found or described on
https://bootlin.com/pub/conferences/2008/elce/

Introduction

Graphical interfaces are a major component of many embedded systems

As usual, the free software and open source world offers a lot of choices to implement graphical interfaces



Such a diversity makes it difficult to find one's way through all the available solutions

This talk in the result of an investigation of the most popular graphical libraries, and describe them in terms of functionality, size, popularity, and ability to combine with other components, etc.

Some solutions or details might have been missed, don't hesitate to share your knowledge and experience during the talk

Building and testing

All of our tests have been made with Buildroot

A filesystem builder for embedded systems

http://buildroot.uclibc.org

Many fixes contributed to get all the graphical libraries to build properly

Size of libraries and components given in the talk correspond to the ARM, stripped version

Basic tests made in Qemu

Not very « flashy »

We will soon release target filesystems that'll allow anyone to quickly try and test the different solutions

« Low-level » solutions

DirectFB

- Low-level graphical library
  - Lines, rectangles, triangles drawing and filling
  - Blitting, flipping
  - Text drawing
  - Windows and transparency
  - Image loading and video display
- But also handles input event handling: mouse, keyboard, joystick, touchscreen, etc.
- Provides accelerated graphic operations on various hardware, more can be added in an easy way
- Integrated windowing infrastructure

DirectFB (2)

Single-application by default, but multiple applications can share the framebuffer thanks to « fusion »

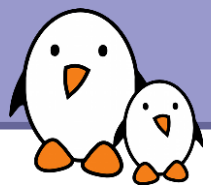
Development and community: very active

See also Denis Oliver Kropp talk, « Open Integration Layer - DirectFB 2.0 », tomorrow at 15:25

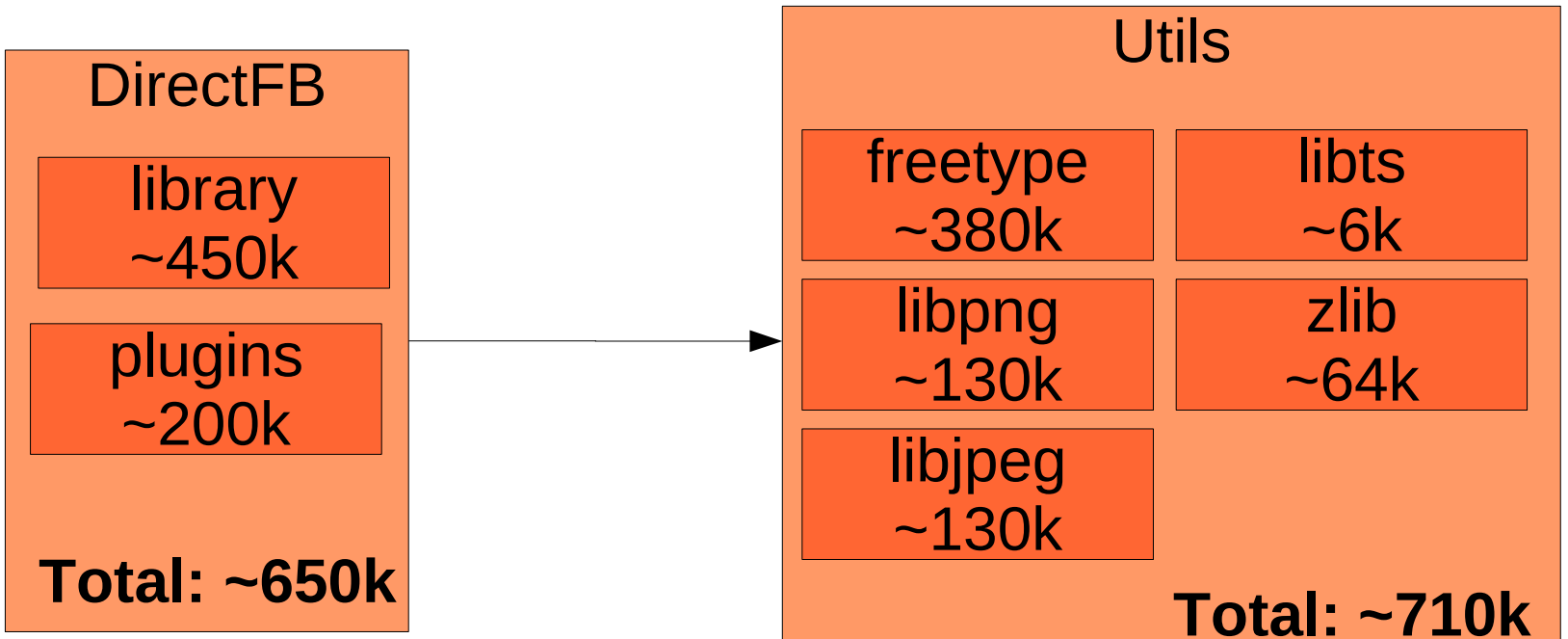
- Denis is the main developer of DirectFB

License: LGPL 2.1

http://www.directfb.org



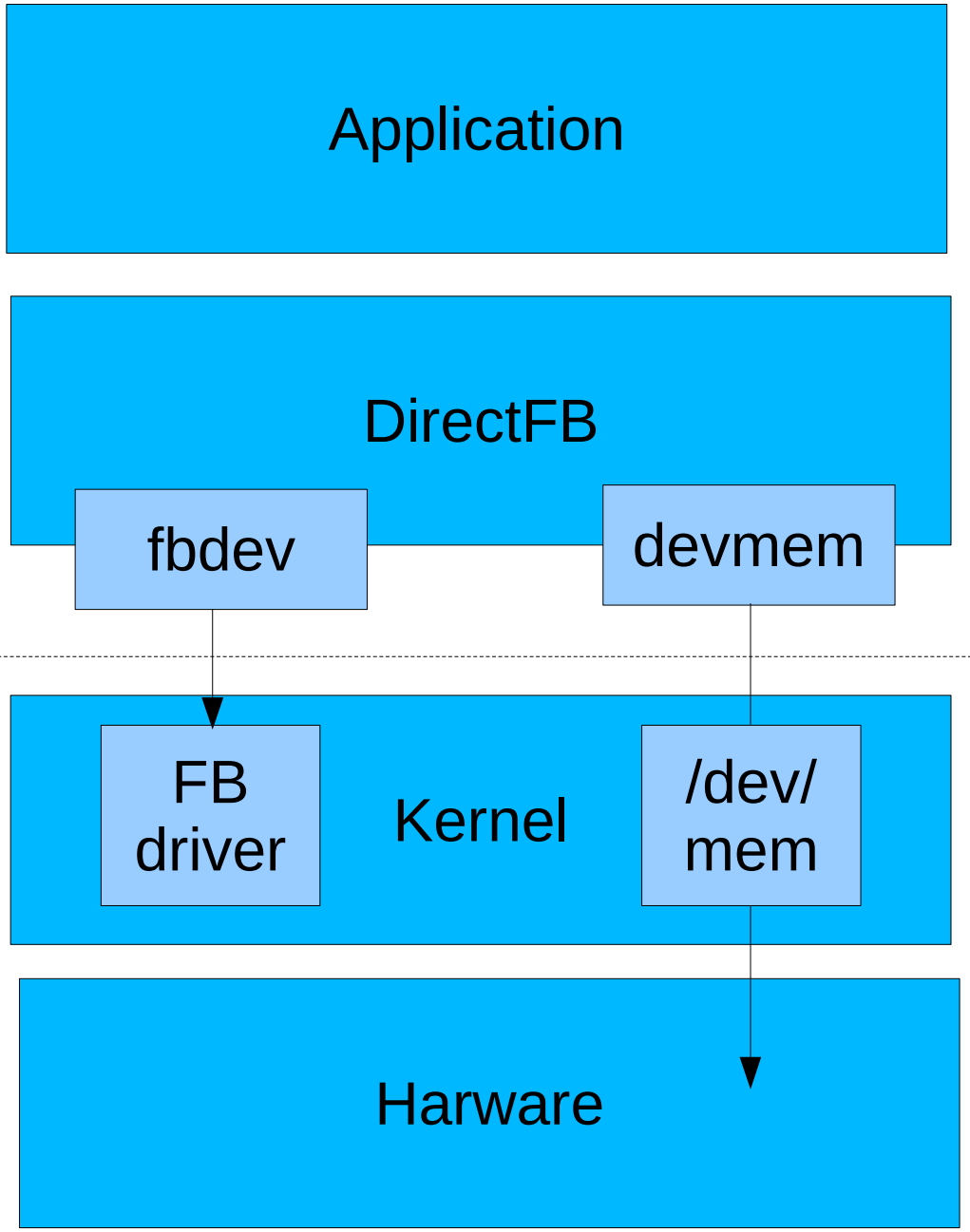
DirectFB : size and dependencies



Total: ~1.4m

Some of these dependencies are optional. This is a typical setup.

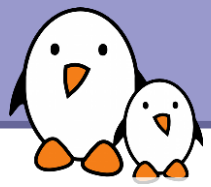
DirectFB : architecture



DirectFB : usage

- Multimedia applications
  - For example the Disko framework, for set-top box related applications
- « Simple » graphical applications
  - Industrial control
  - Device control with limited number of widgets
- Visualization applications
- As a lower layer for higher-level graphical libraries

DirectFB : usage



SDL

A library originally designed for game development

In addition to graphic display, also provides input event management, sound, CD-ROM audio, threads, timers, etc.

Can work on top of X11, the framebuffer or DirectFB

And DirectFB can work on top of SDL as well :-)

The API is roughly the same level as the one of DirectFB

Developed in C, C API, many bindings available

Actively maintained

DirectFB is probably more common in embedded systems, while SDL is more common for small desktop games

License: LGPL

http://www.libsdl.org/

X.org - KDrive

- Stand-alone simplified version of the X server, for embedded systems
  - Formerly know as Tiny-X
  - Kdrive is integrated in the official X.org server
- Works on top of the Linux frame buffer, thanks to the Xfbdev variant of the server
- Real X server
  - Fully supports the X11 protocol : drawing, input event handling, etc.
  - Allows to use any existing X11 application or library
- Actively developed and maintained
- X11 license

http://www.x.org

Kdrive : size and dependencies

X server

Xfbdev
~1.2m

Fonts

from a few kb
to several mb

X libraries

libxcb
~300k

libXfont
~380k

liblbxutil
~156k

Misc libs
~770k

libX11
~920k

Total: 2.5m

X toolkit (optional)

libXaw6,7,8
~900k

libXt
~330k

Utils

dbus
lib: ~200k
bin: ~350k

libsysfs
~27k

libpng
~130k

expat
~120k

zlib
~64k

freetype
~380k

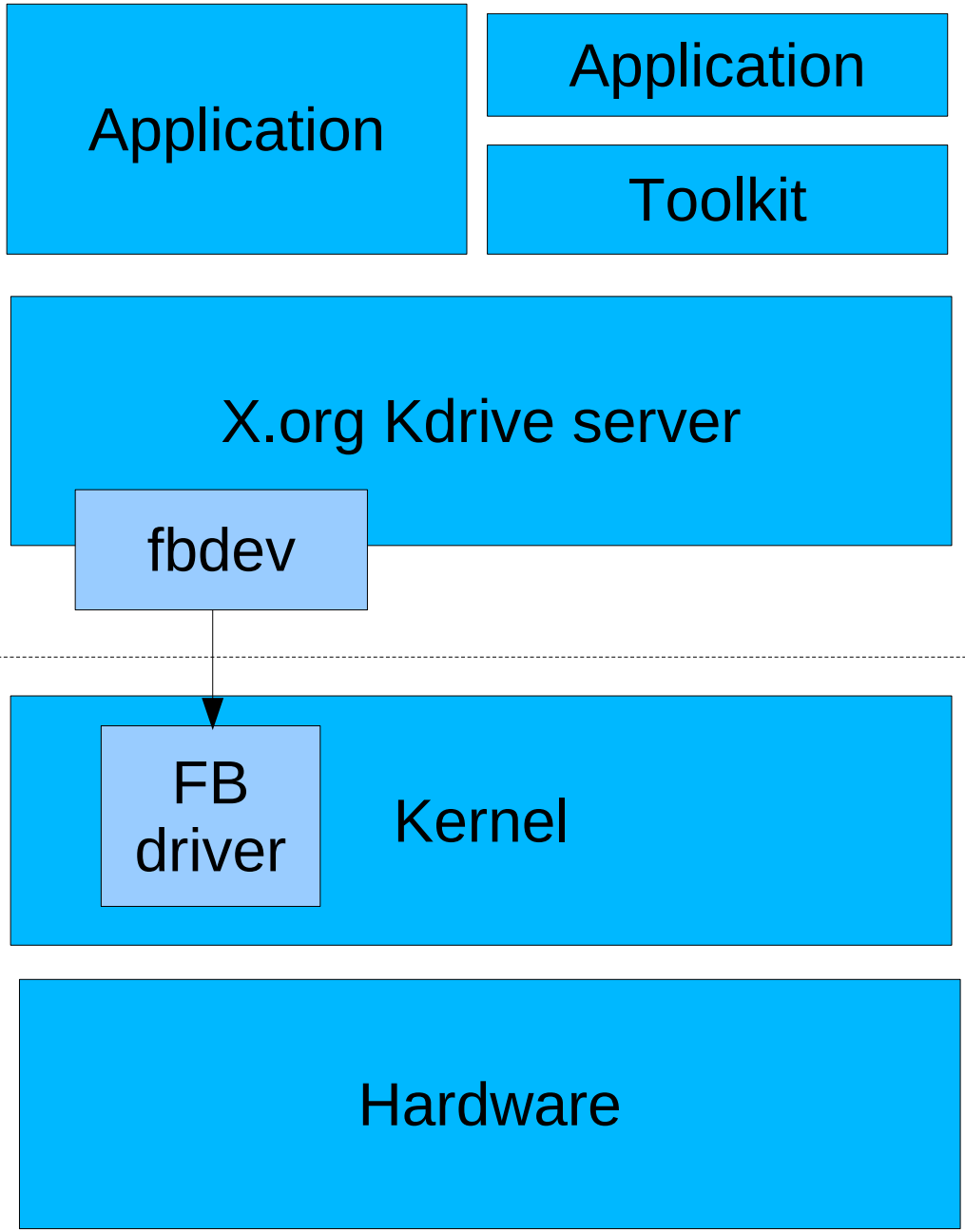
pixman
~130k

fontconfig
~165k

Total: 1.5m

Total, without X toolkit: 5.4m

Kdrive : architecture



Kdrive : usage

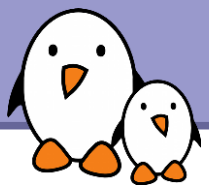
- Can be directly programmed using Xlib / XCB
  - Low-level graphic library
  - Probably doesn't make sense since DirectFB is a more lightweight solution for an API of roughly the same level (no widgets)
- Or, usually used with a toolkit on top of it
  - Gtk
  - Qt
  - Fltk
  - WxEmbedded

Nano-X

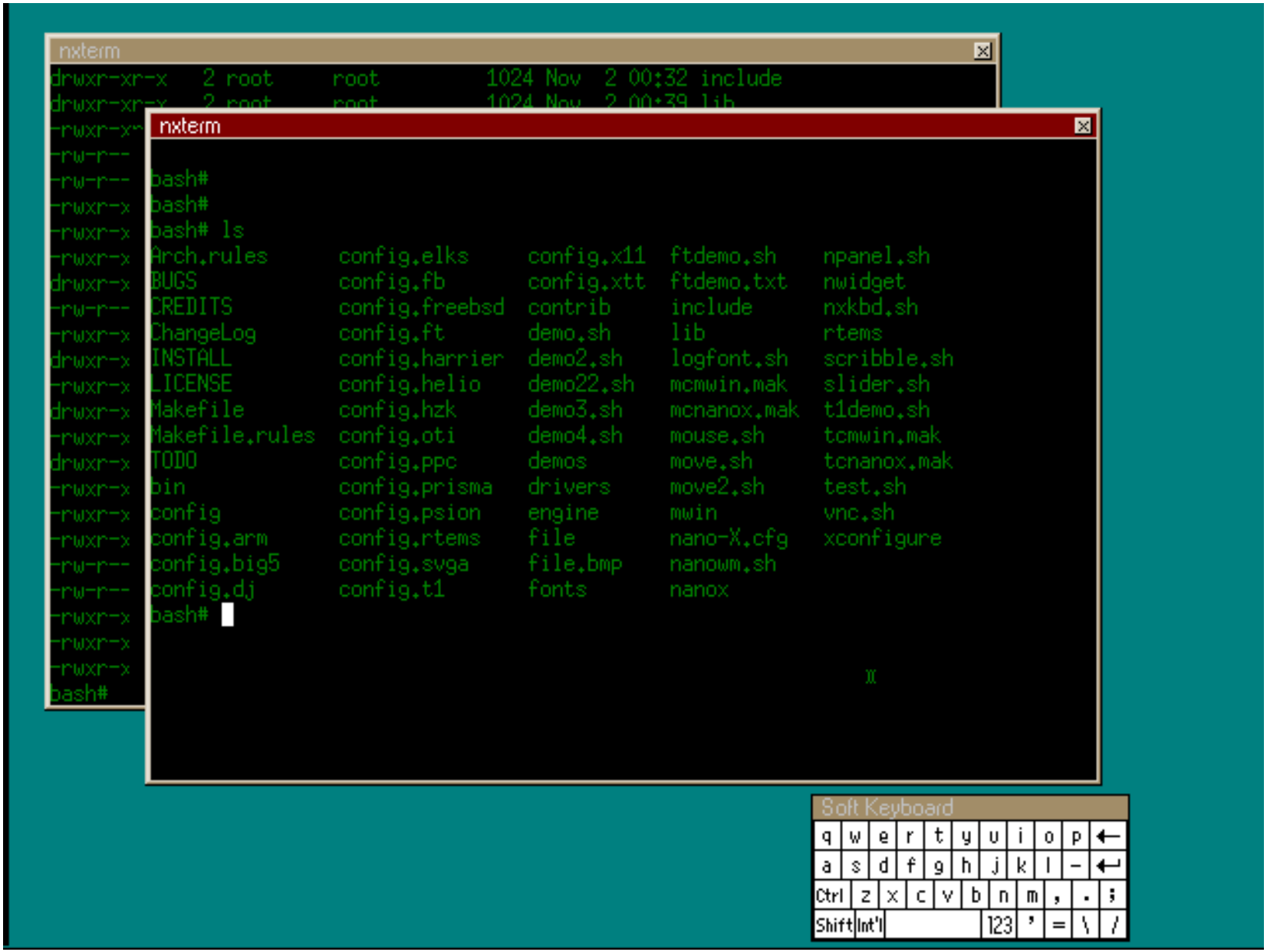
- An alternative graphic system
- Various graphic back-ends, including Linux framebuffer
- Provides two programming APIs
  - A Win32-like API
  - A Xlib-like API, but not compatible with Xlib
  - Requires specially designed widget toolkits
- Client/server model, with clients communicating with the server through an Unix socket. Optionally, a client can be directly linked with the server
- Lightweight

Nano-X

- Doesn't come with a widget library, but offers a window manager
- Relatively small user and developer base. No release since 2005, but some activity on the mailing-list, including the official maintainer.
  - Probably not lively enough to provide a long-term maintenance guarantee
- Licensed under the Mozilla Public License
- Webpage: http://www.microwindows.org/



Nano-X


nxterm
drwxr-xr-x 2 root root 1024 Nov 2 00:32 include
nxterm
bash#
bash#
bash# ls
Arch.rules config.elks config.x11 ftdemo.sh npanel.sh
BUGS config.fb config.xtt ftdemo.txt nwidget
CREDITS config.freebsd contrib include nxkbd.sh
ChangeLog config.ft demo.sh lib rtems
INSTALL config.harrier demo2.sh logfont.sh scribble.sh
LICENSE config.helio demo22.sh mcmwin.mak slider.sh
Makefile config.hzk demo3.sh mcnanox.mak t1demo.sh
Makefile.rules config.oti demo4.sh mouse.sh tcmwin.mak
TODO config.ppc demos move.sh tcnanox.mak
bin config.prisma drivers move2.sh test.sh
config config.psion engine mwin vnc.sh
config.arm config.rtems file nano-X.cfg xconfigure
config.big5 config.svga file.bmp nanowm.sh
config.dj config.t1 fonts nanox
bash#
Soft Keyboard


Web Media
CENTURY SOFTWARE.COM
National Semiconductor
Video
source
web
pgm guide
email
shopping
exit
DVD
--- no disc ---
other text information
Financial Zone
Newsmakers
Stock Market Update
[Close] - In a move that was widely anticipated, the Federal Reserve lowered short-term interest rates today, confirming the expectations that have provided the market with a nice boost in recent sessions...

« High-level » solutions

Gtk

The famous toolkit, providing widget-based high-level APIs to develop graphical applications

Standard API in C, but bindings exist for various languages: C++, Python, etc.

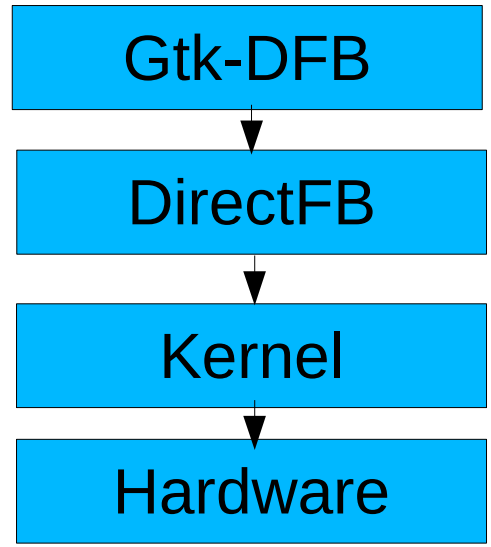
Two GDK back-ends

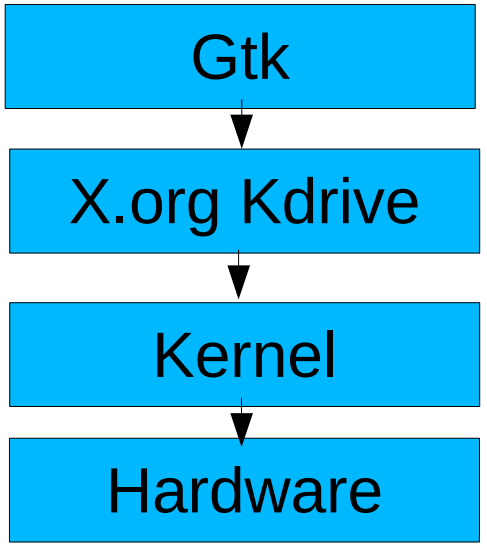
- The classical Xorg back-end
- The DirectFB back-end, which removes the need for an Xorg server

No windowing system, a lightweight window manager needed to run several applications. Possible solution: Matchbox.

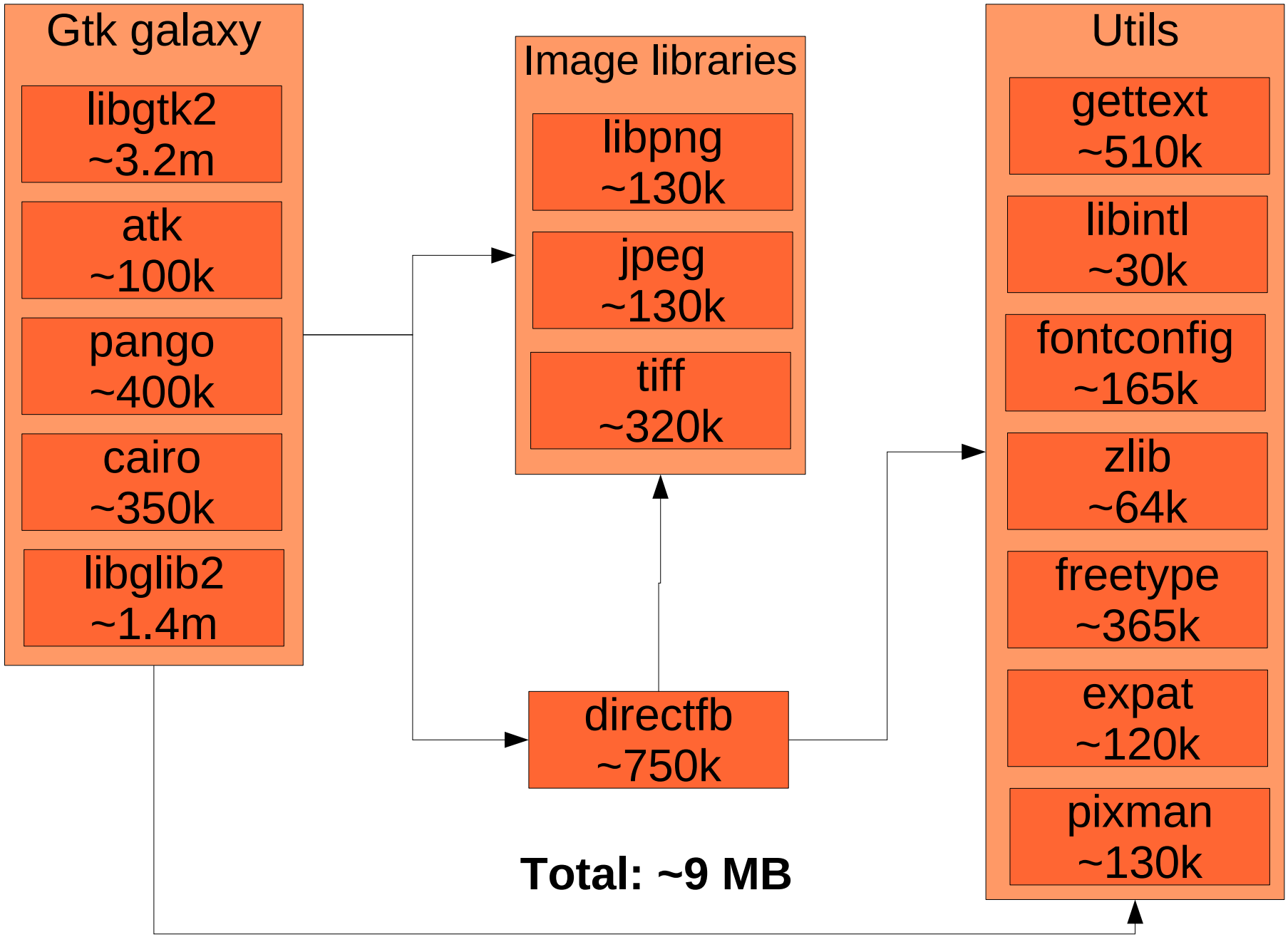
License: LGPL

http://www.gtk.org





Gtk-DFB : dependencies and size



Total: ~9 MB

Qt

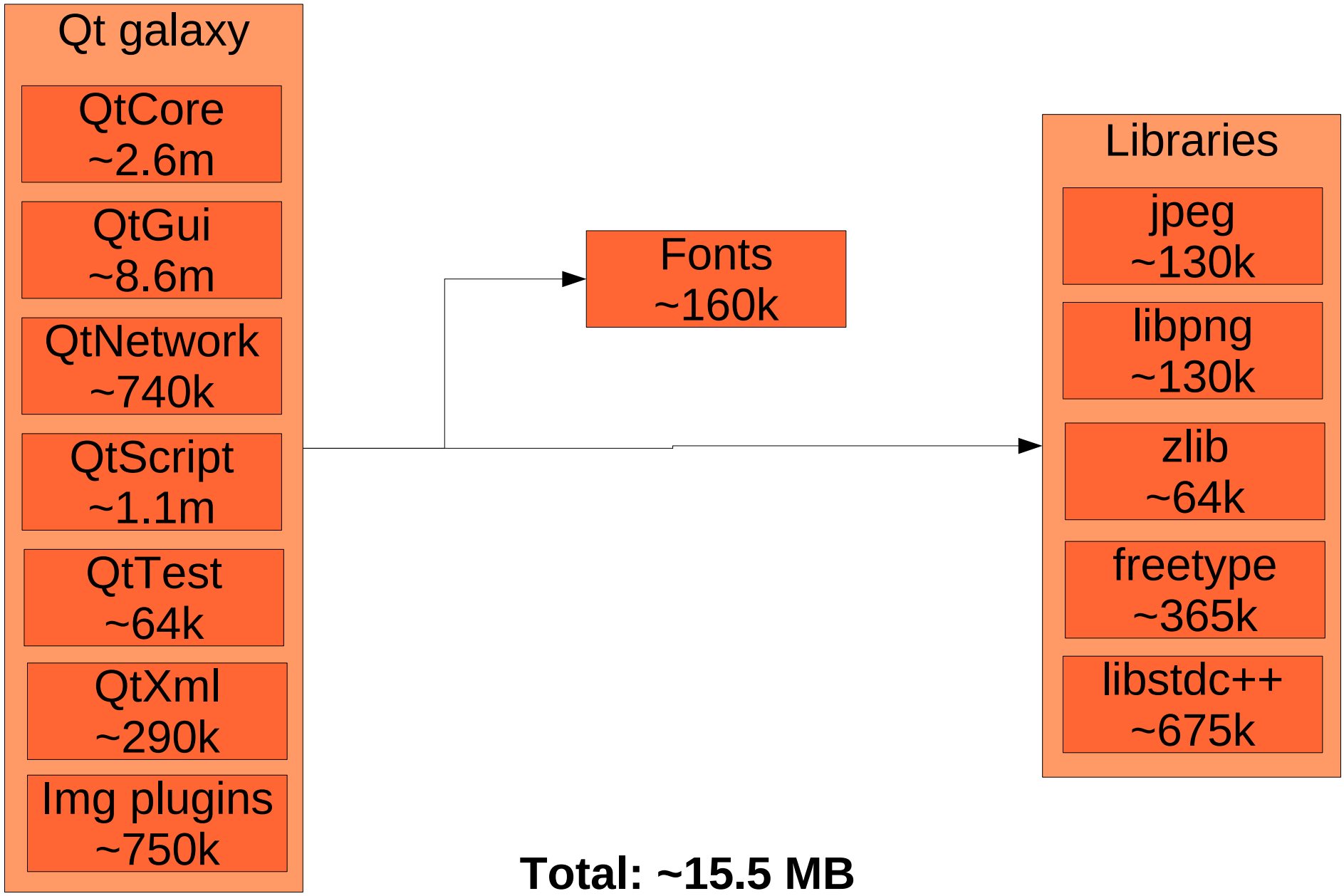
- The other famous toolkit, providing widget-based high-level APIs to develop graphical applications
  - « Qt for Embedded Linux », formerly known as Qtopia Core, is the version of Qt that runs on top of a frame buffer, on embedded devices. It includes a windowing system
  - « Qt Extended », formerly known as Qtopia, extends « Qt for Embedded Linux » with useful components on embedded devices : communication, contents, application-specific and user experience components.
- Implemented in C++
  - the C++ library is required on the target system
  - standard API in C++, but bindings are also available for other languages

Qt

- Works either on top of
  - Framebuffer
  - X11
  - DirectFB backend integrated in version 4.4, which allows to take advantage of the acceleration provided by DirectFB drivers
- Qt is more than just a graphical toolkit, it also offers a complete development framework: data structures, threads, network, databases, XML, etc.
- Very well documented
- GPL license, commercial licenses available for proprietary applications

http://trolltech.com/products

Qt : size and dependencies



Total: ~15.5 MB

Qt

FLTK

Another high-level toolkit, providing widget-level graphic programming, currently in version 2.x

Written in C++, main programming API in C++

Designed with smallness in mind

- Designed to reduce size when statically linked, small memory consumption for each widget
- Core of the library, as included in an statically compiled hello world program: 82K. An example application which uses all widgets: 352K

Works on top of Xlib

Actively maintained, licensed under the LGPL

A port of FLTK 1.x to DirectFB was made, available from DirectFB's git repository, but only maintained by a single person

http://www.fltk.org

WxEmbedded

- Originally, WxWidgets is a library abstracting existing toolkits (Win32 native toolkit, Gtk2, etc.), allowing the creation of portable applications
  - Depends on an underlying toolkit
- An internal toolkit, WxUniversal has been implemented. It allows to write WxWidgets applications without the need for an underlying toolkit.
- Works on top of various graphic back-ends
  - X11
  - DirectFB
  - Nano-X Xlib-like and Win32-like APIs
- Written in C++, C++ API, C++ library required

WxEmbedded

- Size
  - 2.5 Mb of shared library size
  - 1 Mb for a simple statically-linked example, but the size doesn't increase much for more complicated examples
- While WxWidgets is active and well-maintained, the amount of attention given to WxUniversal and WxEmbedded is much smaller
  - Some widgets are not implemented, some graphic backends don't receive a lot of attention
- Licensed under the LGPL, with an exception giving more freedom on distribution of derived works in binary form
- WxEmbedded: http://www.wxwidgets.org/docs/embedded.htm
- WxUniversal: http://www.wxwidgets.org/about/wxuniv.htm

LiTE

LiTE is a Toolbox Engine

Designed exclusively for DirectFB

Two layers

- Lite: low-level plumbing to ease the implementation of widgets
- Leck: implementation of the widgets

Only very simple widgets available

Limited user base

Very small: only 43k + 36k in addition to a typical DirectFB installation.

LGPL 2.1

http://www.directfb.org/wiki/index.php/LiTE:About

Other untested solutions

- MiniGUI
  - Another toolkit, with several graphical backends including a frame buffer backend
  - The GPL version is limited in functionalities, commercial versions available
  - http://www.minigui.org/
- Enlightenment foundation libraries
  - See Gustavo Sverzut Barbieri talk, « Rich GUI Without Pain » today at 14:45
  - http://www.enlightenment.org/p.php?p=about/efl

Conclusion

Low-level components: DirectFB, X.org, Nano-X

High-level components: Gtk, Qt, Fltk, WxEmbedded, EFL, miniGUI

There is a large number of solutions to develop graphical interfaces with free software libraries

However, the size of the user and developer community and its vitality is one of the most important criteria when choosing a solution

For that reason, DirectFB, X.org, Gtk and Qt seem to be most interesting solutions today

Questions ?

What solutions are you using for your graphical interfaces ?

What were your decision criteria ?

Are you satisfied with the chosen solution ? The existing ones ?

?