



## Akademy 2008

### Programming for embedded Linux

Harald Fernengel

Professional Services Engineer

09 August 2008 | Somewhere in Belgium

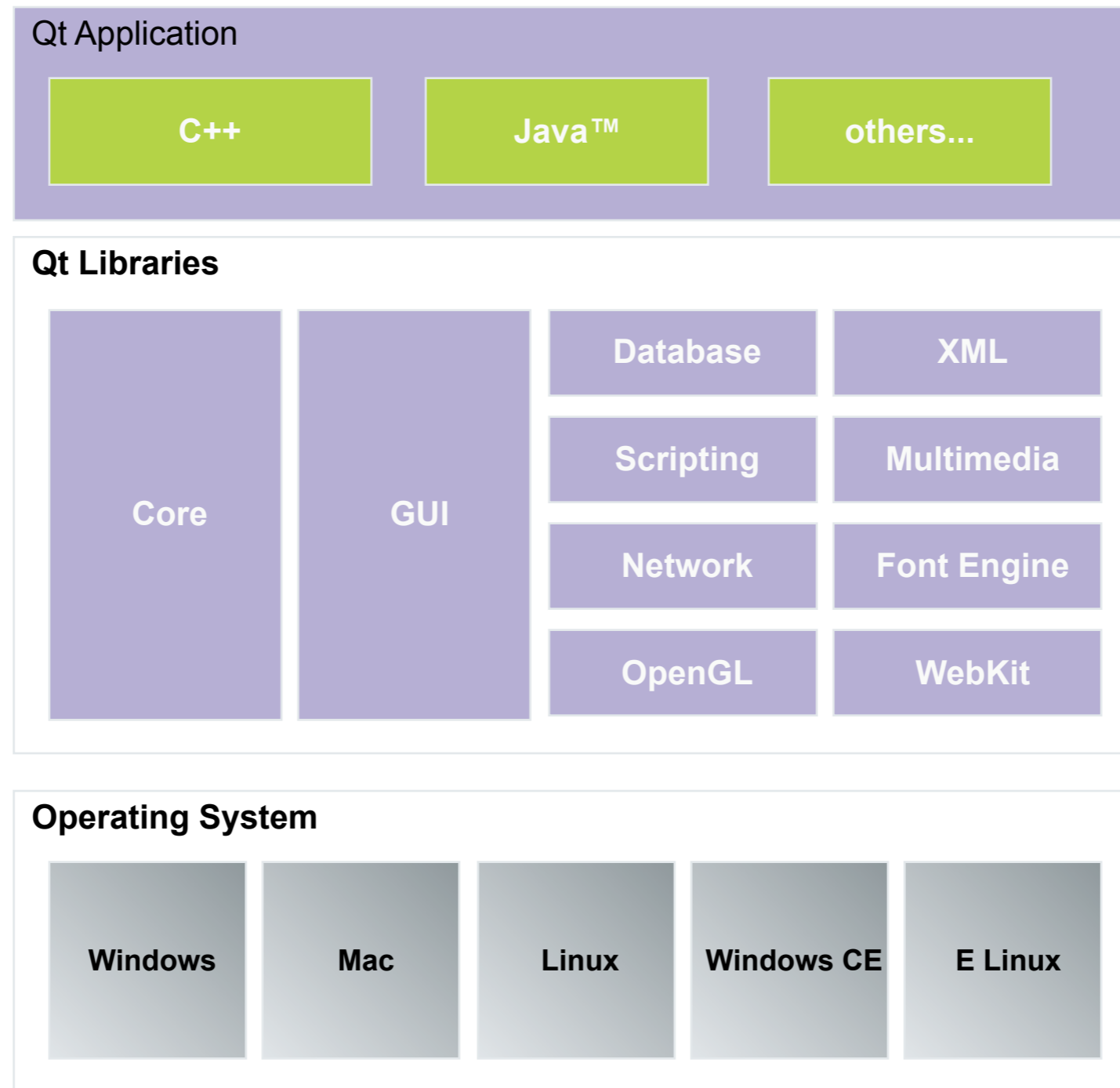


# Agenda

- 1 Qt for Embedded Overview
- 2 The Fun of Cross-Compiling



# Architecture overview



# Which OS is special from a UI perspective?

Qt

## Operating System

Windows

Mac

Linux

Windows CE

E Linux

# Something missing?

Qt

## Native UI

Win32 UI

Carbon /  
Cocoa

X11

CE libs

Windows

Mac

Linux

Windows CE

E Linux

## QWS to the rescue

Qt

### Native UI

Win32 UI

Carbon /  
Cocoa

X11

CE libs

QWS

Windows

Mac

Linux

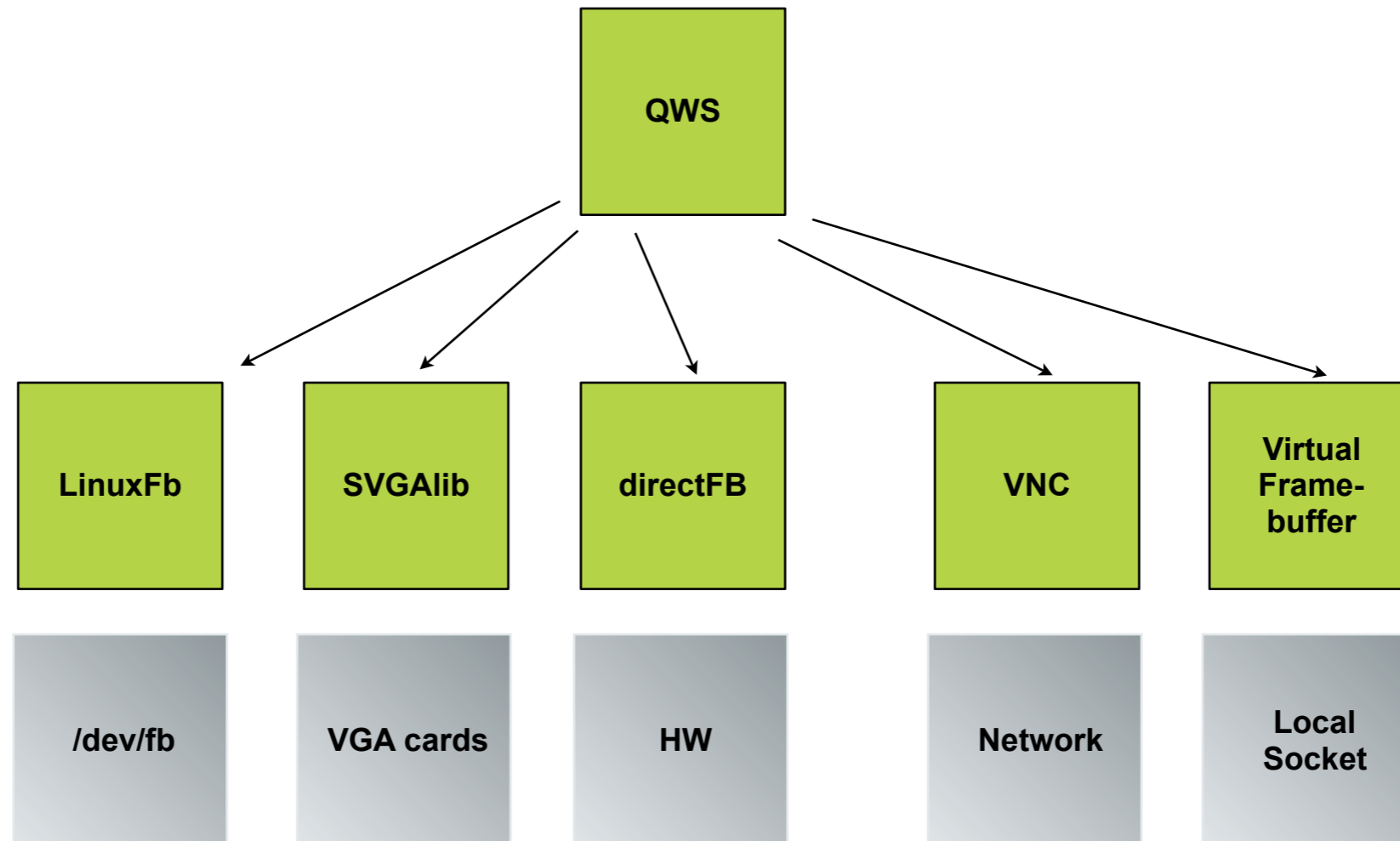
Windows CE

E Linux

## What is QWS?

- Qt Windowing System
  - Lightweight UI Server
    - window manager
    - plug-able font engines (qpf, freetype, Agfa Monotype, Arphic Font Engine, ...)
  - Connects Qt with low-level devices
    - Keyboard / Keypad
    - Mouse / Touchscreen
    - Display

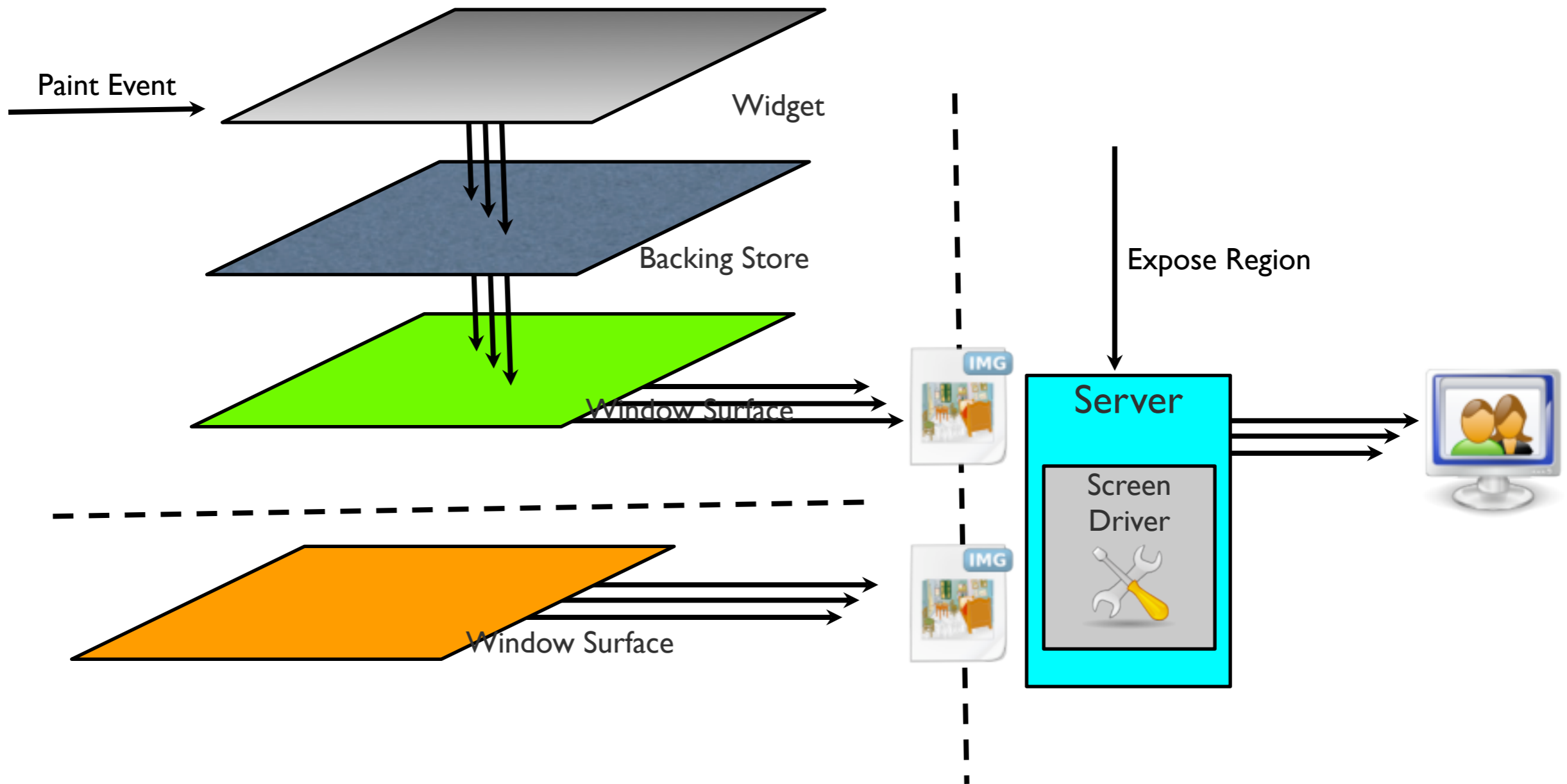
# QWS - Gfx Drivers



**QWS**

# Demo

# QWS - Painting



## Why (not) QWS?

- Lightweight implementation + legacy free
  - Multiple screens
  - Rotated screens
  - Top-level transparency & composition
  - Network transparency through VNC protocol
  - Both in-process or client-server mode
  
- Tailored for Qt - coexistence with other toolkits tricky
  - J2ME successfully integrated
- Smaller ecosystem



# Fine-tune library size with QConfig

The screenshot shows the Qt QConfig application window titled "QConfig - New File". The interface is divided into two main panes. The left pane displays a tree view of features, each with a checkbox and a corresponding class name. The right pane shows the details for the selected feature, "GRAPHICSVIEW".

Id	Name
<input checked="" type="checkbox"/>	TOOLBAR
<input checked="" type="checkbox"/>	QToolBar
<input checked="" type="checkbox"/>	MENUBAR
<input checked="" type="checkbox"/>	QMenuBar
<input checked="" type="checkbox"/>	WORKSPACE
<input checked="" type="checkbox"/>	QWorkSpace
<input checked="" type="checkbox"/>	PROGRESSBAR
<input checked="" type="checkbox"/>	QProgressBar
<input checked="" type="checkbox"/>	RESIZEHANDLER
<input checked="" type="checkbox"/>	Resize Handler
<input checked="" type="checkbox"/>	RUBBERBAND
<input checked="" type="checkbox"/>	QRubberBand
<input checked="" type="checkbox"/>	SPLITTER
<input checked="" type="checkbox"/>	QSplitter
<input checked="" type="checkbox"/>	SIGNALMAPPER
<input checked="" type="checkbox"/>	QSignalMapper
<input checked="" type="checkbox"/>	SIZEGRIP
<input checked="" type="checkbox"/>	QSizeGrip
<input checked="" type="checkbox"/>	SLIDER
<input checked="" type="checkbox"/>	QSlider
<input checked="" type="checkbox"/>	DIAL
<input checked="" type="checkbox"/>	QDial
<input checked="" type="checkbox"/>	SCROLLBAR
<input checked="" type="checkbox"/>	QScrollBar
<input checked="" type="checkbox"/>	SCROLLAREA
<input checked="" type="checkbox"/>	QScrollArea
<input checked="" type="checkbox"/>	<b>GRAPHICSVIEW</b>
<input checked="" type="checkbox"/>	<b>QGraphicsView</b>
<input checked="" type="checkbox"/>	PRINTPREVIEW...
<input checked="" type="checkbox"/>	QPrintPreviewWidget
<input checked="" type="checkbox"/>	MDIAREA
<input checked="" type="checkbox"/>	QMdiArea
<input checked="" type="checkbox"/>	TEXTEDIT
<input checked="" type="checkbox"/>	QTextEdit
<input checked="" type="checkbox"/>	SYNTAXHIGHLI...
<input checked="" type="checkbox"/>	QSyntaxHighlighter
<input checked="" type="checkbox"/>	TEXTBROWSER
<input checked="" type="checkbox"/>	QTextBrowser
<input checked="" type="checkbox"/>	SPINWIDGET
<input checked="" type="checkbox"/>	Spin Widget
<input checked="" type="checkbox"/>	SPLASHSCREEN
<input checked="" type="checkbox"/>	Splash screen widget
<input checked="" type="checkbox"/>	STACKEDWIDGET
<input checked="" type="checkbox"/>	QStackedWidget
<input checked="" type="checkbox"/>	TABWIDGET
<input checked="" type="checkbox"/>	QTabWidget
<input checked="" type="checkbox"/>	STATUSBAR
<input checked="" type="checkbox"/>	QStatusBar
<input checked="" type="checkbox"/>	STATUSTIP
<input checked="" type="checkbox"/>	Status Tip
<input checked="" type="checkbox"/>	TABLEWIDGET
<input checked="" type="checkbox"/>	QTableWidget
<input checked="" type="checkbox"/>	TOOLBUTTON
<input checked="" type="checkbox"/>	QToolButton

**GRAPHICSVIEW**

[QGraphicsView](#)

Supports the graphicsview classes.

**Section:**

- Widgets

**Requires:**

- [On] [SCROLLAREA](#)

**Required for:**

- [On] [PRINTPREVIEWWIDGET](#)
- [On] [GRAPHICSSVGITEM](#)

179 features loaded from /Users/harald/troll/qt-4.4/src/corelib/global/qfeatures.txt

# Agenda

- 1 Qt for Embedded Overview
- 2 The Fun of Cross-Compiling



## Building Qt embedded for your host

- `./configure -embedded -qvfb`
- `make`
- `sudo make install`

## Building Qt embedded for your target

- `./configure -embedded arm -xplatform qws/linux-arm-g++`
- `make`
- `sudo make install`

## Two tool chains

- Host tool chain
  - Tools for building Qt (qmake, moc, uic, rcc...)
- Target tool chain (cross-compiler)
  - Everything that runs on the target (libs, apps...)

## Cross Compiling

- **g++** main.cpp
- *a.out: ELF 32-bit LSB executable, Intel 80386*
  
- **arm-linux-g++** main.cpp
- *a.out: ELF 32-bit LSB executable, ARM*
  
- **ntomips-g++** main.cpp
- *a.out: ELF 32-bit MSB executable, MIPS*

## Cross Compiling

- Cross compiler can compile & link
  - ... but apps can't be executed (unless using scratchbox)
- Special care needed for configure tests

## Workarounds - configure tests

```
int main()  
{  
    printf("size of pointers: %d\n", sizeof(void*));  
}
```

## Workarounds - configure tests

```
int main()  
{  
    STATIC_ASSERT(sizeof(void *) == 4);  
}
```

## Workarounds - configure tests

```
// "MostSignificantByteFirst"
```

```
short msb_bigendian[] = { 0x0000, 0x4d6f, 0x7374, 0x5369,  
0x676e, 0x6966, 0x6963, 0x616e, 0x7442, 0x7974, 0x6546,  
0x6972, 0x7374, 0x0000 };
```

```
// "LeastSignificantByteFirst"
```

```
short lsb_littleendian[] = { 0x0000, 0x654c, 0x7361, 0x5374,  
0x6769, 0x696e, 0x6966, 0x6163, 0x746e, 0x7942, 0x6574,  
0x6946, 0x7372, 0x0074, 0x0000 };
```

**Thank You!**

And happy embedded hacking!