Running embedded operating systems on low-cost ARM-based platforms

Anthony Pellerin

apellerin@adeneo-embedded.com

Competitive Partner for Your Embedded Device Project

Adeneo Embedded

Embedding success

OUR TEAM

60+ engineering team - experts in system integration on complex embedded devices

Engineering offices in Europe and USA

OUR PARENT COMPANY

Subsidiary of Adetel Group
a 500+ employees company involved in complete embedded systems development

OUR EXPERTISE

Solutions for various operating systems - Windows Embedded, Windows Mobile, Embedded Linux and Android **- using various architectures and platforms from our industrial partners**

Automotive

Medical

Industrial and Handheld Devices

Consumer

Home/Building Automation

Windows® Embedded Gold Partner Excellence Awards 2007•2008•2009•2010•2011

freescale Alliance Member

Texas Instruments Design Network Platinum Member

Windows phone

Microsoft® Silverlight

Qt

Linaro

Firmware Expertise
References at a glance

Adeneo Embedded

Embedding success

Windows Embedded Compact

Windows Embedded CE

Fleet Management
BSP Customization

Handheld Terminal
Wifi Driver Integration

Home Energy Management
BSP/Driver Consulting

Car Diagnostic Rugged Device
Camera and Wifi Integration

Interactive Kiosk
BSP Development
Application Development

In-Vehicle Infotainment
Complete BSP Adaptation
CAN Stack Adaptation

Solar Powered Parking
Complete BSP Adaptation
Power Mgt Optimization

GIS/Survey Handheld
BSP optimization
Power Mgt, Graphical stack

Medical Injection System
BSP Customization
Communication Stacks

Mobile and Embedded Apps
References at a glance

Adeneo Embedded

Embedding success

Microsoft Silverlight

Partner

Windows Embedded
Gold Partner

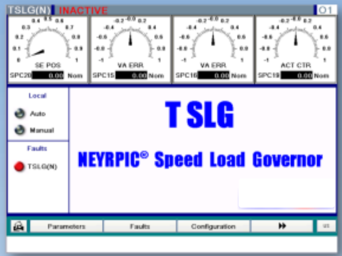
Windows phone

Qt

Fleet Management
UI and App dev
Navigation integration

Smartphone Applications
Windows Phone and Android
UI Design, Application dev

Multimedia Digital Player
UI and multimedia demo SW



Industrial Supervision
UI and Data management App
Industrial bus communications

Home Energy Management
UI and Data management App
Zigbee/Wifi communications

Consumer Application
Complete UI and SW

Metering Maintenance
Software architecture
UI design

RUNNING EMBEDDED OPERATING SYSTEMS ON LOW-COST ARM-BASED PLATFORMS

Agenda

- Embedded devices specificities
- Overview of Linux Embedded
- Overview of Windows Embedded Compact
- Overview of Android
- Comparison: Pros and cons
- How-to get started with embedded OSes on low-cost platforms
  - Linux embedded on the PandaBoard
  - Android 2.3 on the i.MX53 QSB
  - Windows Embedded Compact 7 on the BeagleBoard
- Questions

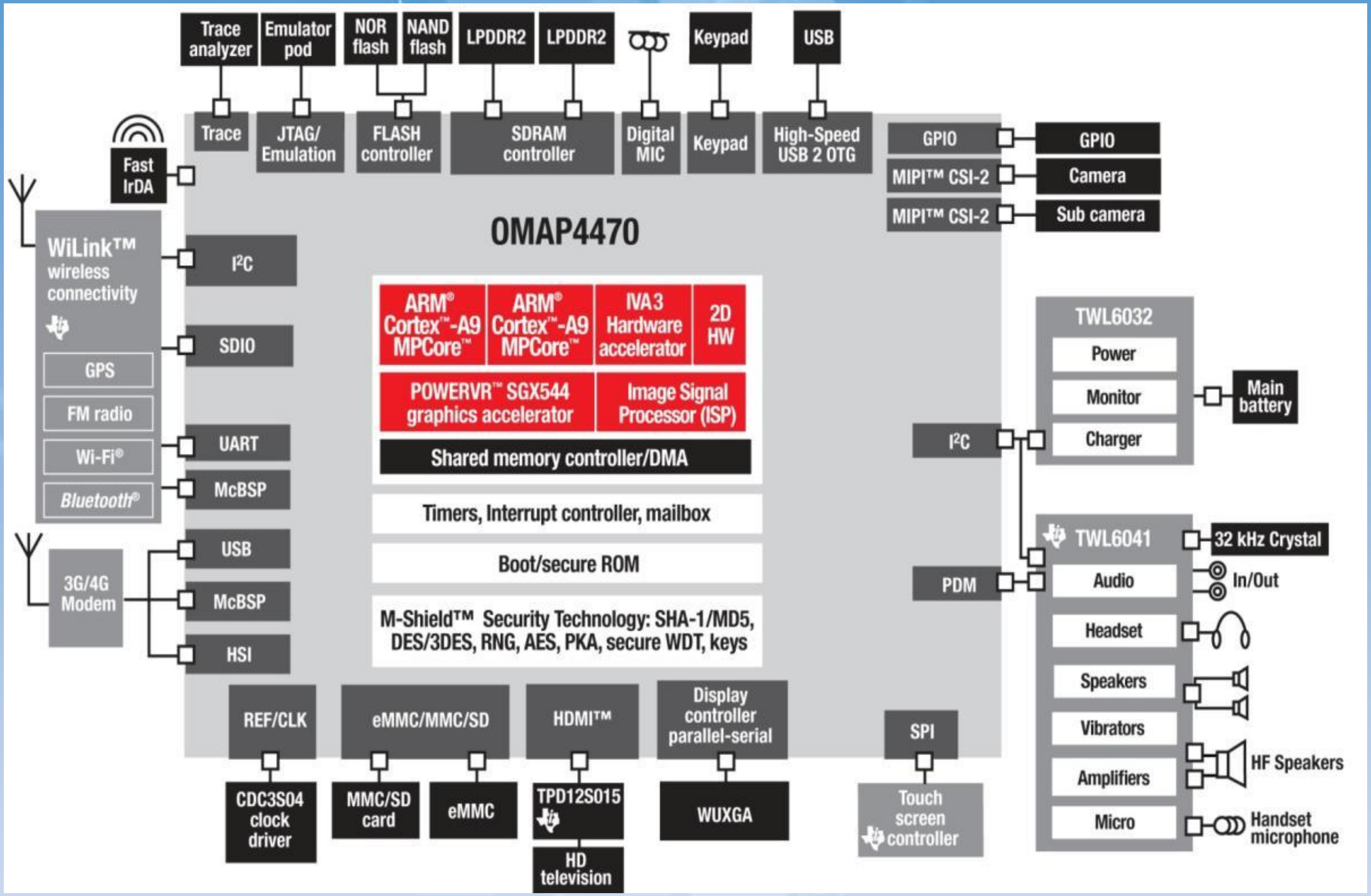
EMBEDDED DEVICES SPECIFICITIES

Embedded devices specificities

- Typical recent high-end embedded configuration
  - 1 GHz processor (may be dual-core)
  - 512 MB DDR-RAM
  - 32GB storage memory
  - Connectivity: LAN 10/100, Wifi, Bluetooth
- ARM cores are the market leaders
  - IP designed and licensed by ARM
  - Silicon vendors use ARM license to build their own ARM-based SOCs
- Each SOC has different peripherals from one SV to another
  - No standard boot program (BIOS, EFI on x86)
  - No standard internal buses
  - No standard power/clock management

Embedded devices specificities

Embedding success



Trace analyzer
Emulator pod
NOR flash
NAND flash
LPDDR2
LPDDR2
Keypad
USB

Trace
JTAG/ Emulation
FLASH controller
SDRAM controller
Digital MIC
Keypad
High-Speed USB 2 OTG

Fast IrDA

GPIO
MIPI™ CSI-2
MIPI™ CSI-2
GPIO
Camera
Sub camera

OMAP4470

WiLink™ wireless connectivity
GPS
FM radio
Wi-Fi®
Bluetooth®

I²C
SDIO
UART
McBSP

3G/4G Modem
USB
McBSP
HSI

ARM® Cortex™-A9 MPCore™
ARM® Cortex™-A9 MPCore™
IVA 3 Hardware accelerator
2D HW
POWERVR™ SGX544 graphics accelerator
Image Signal Processor (ISP)
Shared memory controller/DMA
Timers, Interrupt controller, mailbox
Boot/secure ROM
M-Shield™ Security Technology: SHA-1/MD5, DES/3DES, RNG, AES, PKA, secure WDT, keys

TWL6032
Power
Monitor
Charger
Main battery
I²C

TWL6041
32 kHz Crystal
PDM
Audio
In/Out
Headset
Speakers
Vibrators
Amplifiers
HF Speakers
Micro
Handset microphone

REF/CLK
eMMC/MMC/SD
HDMI™
Display controller parallel-serial
SPI

CDC3S04 clock driver
MMC/SD card
eMMC
TPD12S015
HD television
WUXGA
Touch screen controller

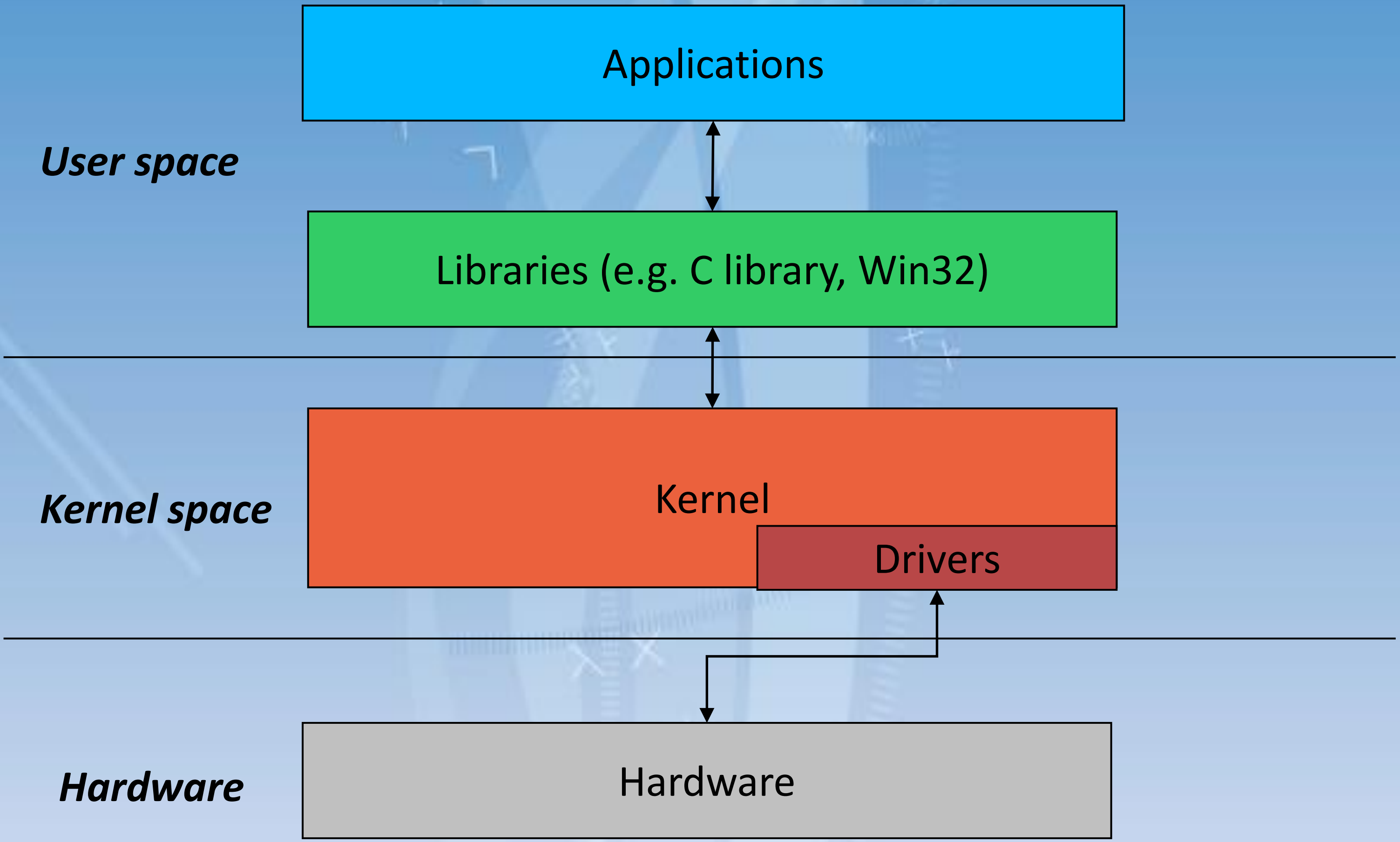
Embedded devices specificities

- X86-based platforms come in various configurations, bringing a generic set of features that may be slightly customized to address specific targets
- Unlike x86-based boards, ARM-based platforms are meant to be highly customizable to reduce costs and/or power consumption
  - The CPU is just the central unit of an embedded platform, external peripherals being tailored for the final product needs
  - Software/drivers must be written/adapted to run on a specific embedded device
    - ➢ *Each board needs a dedicated **B**oard **S**upport **P**ackage to allow a specific operating system to run on the platform*

Board Support Package

- A Board Support Package contains the low-level software needed by one OS to run on a specific target
- A BSP contains:
  - A bootloader:
    - *Performs first initialization of the CPU, memory and peripherals*
    - *Download/fetch the OS image from a media (storage, network,…)*
    - *Launch the OS image*
  - Drivers and low-level software
    - *Low-level support code for the kernel*
    - *Drivers to expose each peripheral to the kernel/applications*
  - Configuration files

Role of an OS kernel



OVERVIEW OF LINUX EMBEDDED

Linux key features

- Portability
  - Supported architectures (see *arch directory in the Linux sources)*: *alpha, arm, m68k, x86, mips, powerpc, sparc...*
- Scalability
  - Used on small embedded devices to super-computers
- Security
  - The code is constantly being reviewed by the community
- Reusability
  - Many drivers and platforms are part of the mainline. No need to reinvent them!
  - Well-defined coding standards
- Community support
  - Easy to find support and documentation

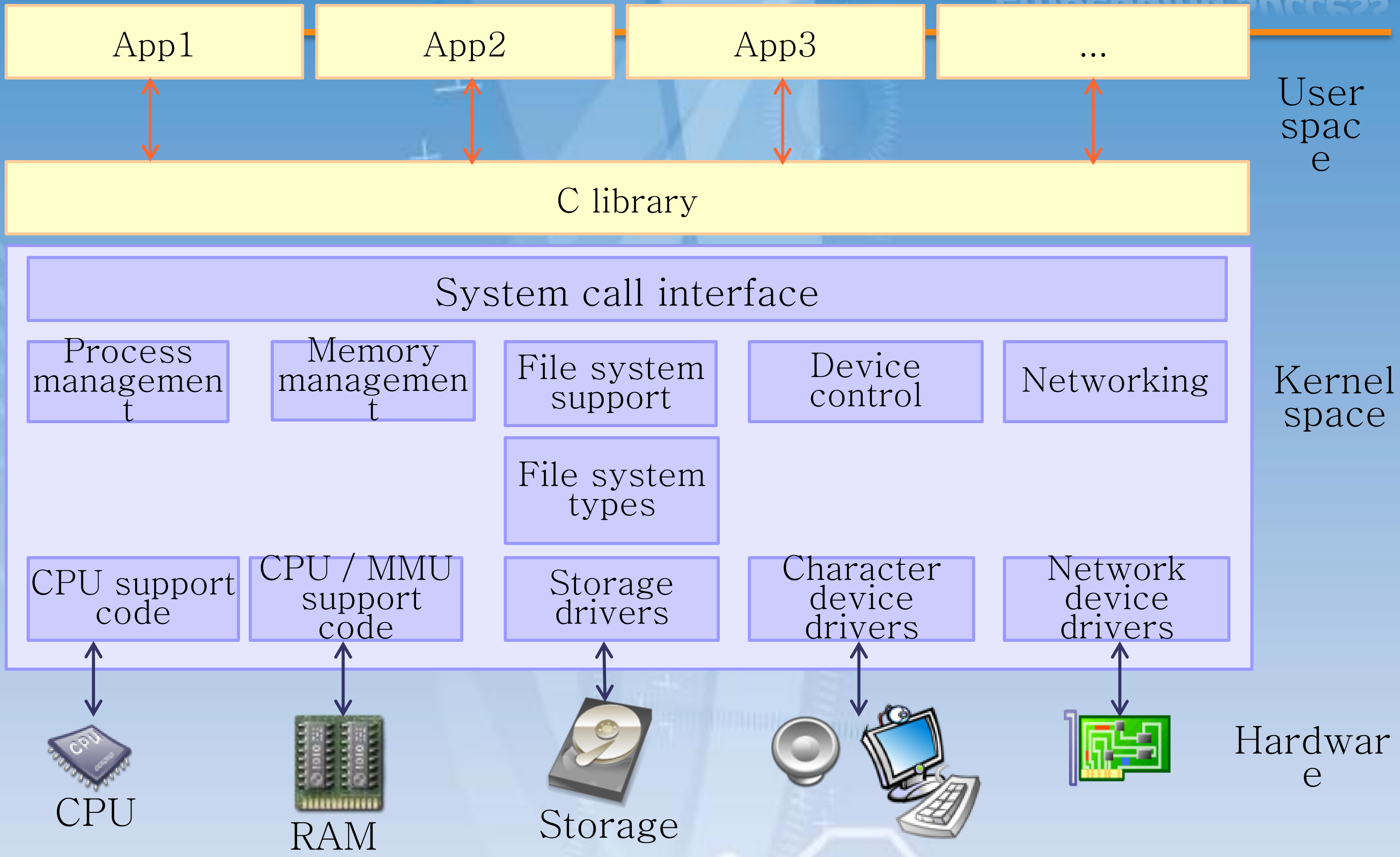
Linux development model

Embedding success

- Latest version is 3.6
  - About one release every 3 months
  - Stable branches are maintained by Greg Kroah Hartman
  - Longterm kernels: 2.6.32, 3.0
- Kernel sources available on http://kernel.org/
  - Can be downloaded as archives or using git
  - *"Mainline" or "Vanilla" kernel: contain the main, generic branch of development*
  - Released by Linus Torvalds after integrating the changes made by all other programmers
- Not all the Linux code is part of the *mainline*
  - Silicon Vendors typically manage their own tree

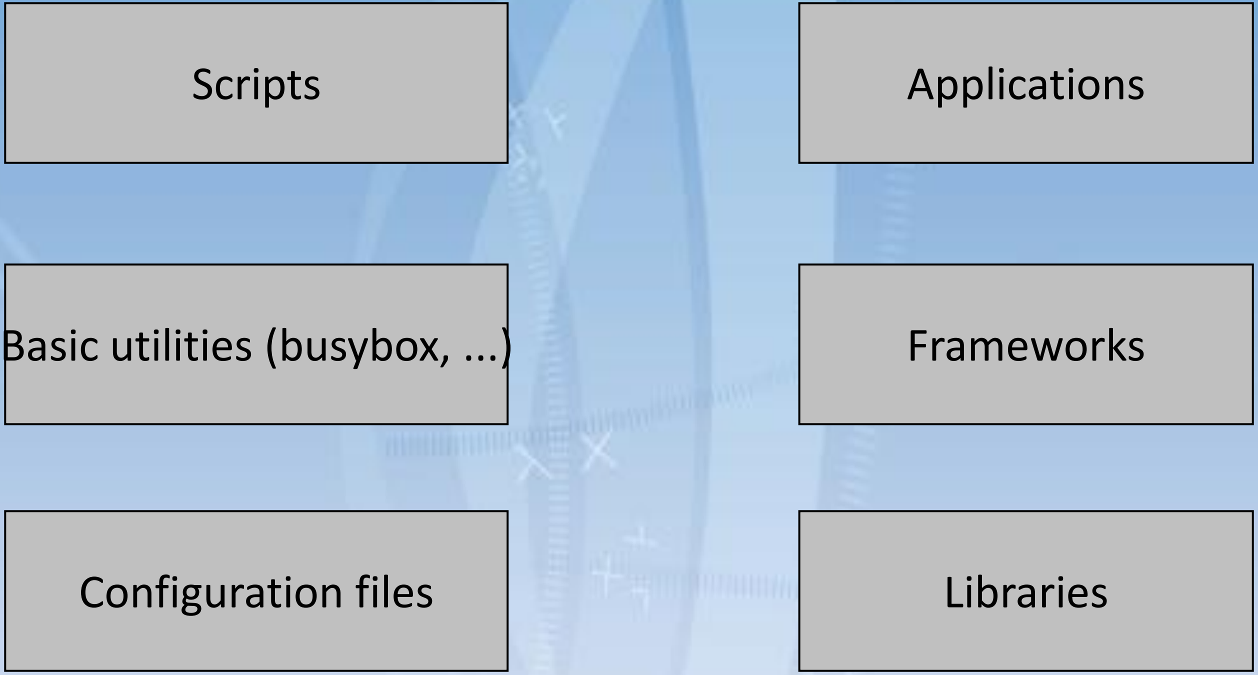
Linux architecture

Embedding success



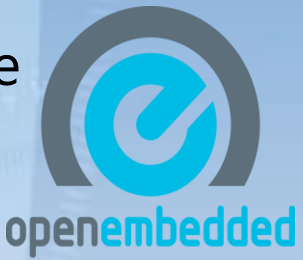
The Root File system

- The root file system is where all the files contained in the file hierarchy (including device nodes) are stored
- Many different components:



Embedded Linux generation tools

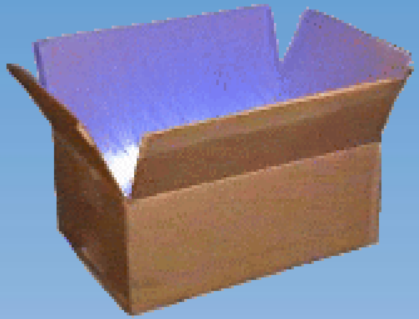
- A set of scripts that will generate a customize root file system for a specific target
  - Download/fetch a software package
  - Configure/compile the package
  - Install/deploy the resulting binaries into the ROOTFS
- Usually provided by Silicon Vendors containing the BSPs for supporting their own platforms
- Various open-source tools available
  - OpenEmbedded
  - LTIB
  - Buildroot



- Commercial distributions: Sell tools and support
  - MontaVista
  - Wind River

Busybox

- **BusyBox** combines tiny versions of many common UNIX utilities into a single small executable. e.g.
  - shell
  - coreutils (*cat, dd, head, tail...)*
  - process utilities (*ps, top...)*
  - ... and much more

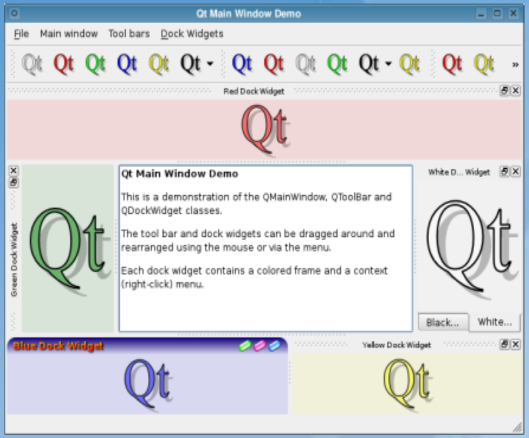


- Less features than the standard GNU implementations...
  - ... but often sufficient for embedded usage
- Features can be enabled or disabled at build-time

Network utilities

- **Dropbear**
  - SSH server and client
  - Small memory footprint (can be configured at build-time)
- **Busybox**
  - FTP/TFTP/DHCP/Telnet servers and clients
  - HTTP server
  - Basic network utilities (ping, ifconfig…)
- Most desktop/server projects can be ported easily to embedded devices (e.g. Apache)
- Alternative lightweight implementations often exist

Graphical interfaces: Qt

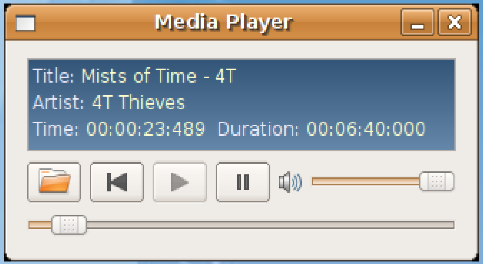


Qt Main Window Demo

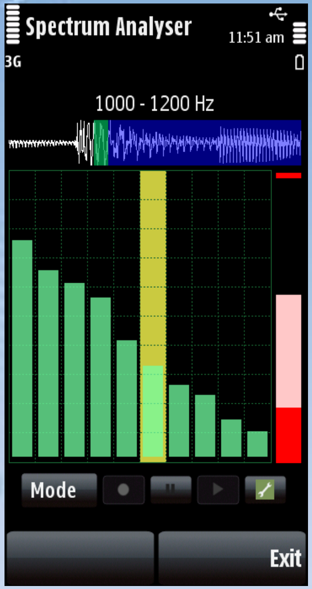
This is a demonstration of the QMainWindow, QToolBar and QDockWidget classes.

The tool bar and dock widgets can be dragged around and rearranged using the mouse or via the menu.

Each dock widget contains a colored frame and a context (right-click) menu.



Title: Mists of Time - 4T
Artist: 4T Thieves
Time: 00:00:23:489 Duration: 00:06:40:000

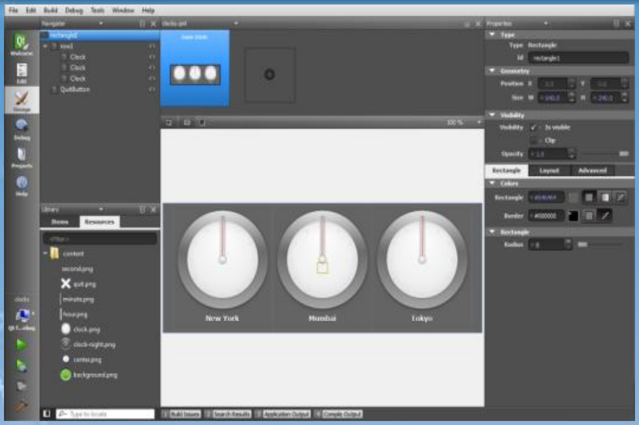


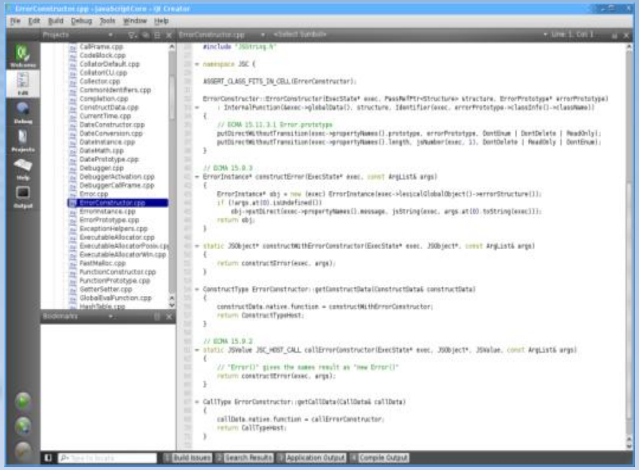
Graphical interfaces: Qt

- Cross-platform application and UI framework
  - Maintained by Nokia
- Written in C++
- Features:
  - GUI
  - XML parsing
  - Database access
  - File handling
  - Internationalization support
  - Graphics hardware acceleration
- Native performance
- Easy to prototype on PC

Graphical interfaces: Qt Creator

- Create interfaces with Qt Creator:
  - C++ and JavaScript code editor
  - Integrated UI designer
  - Project and build management tools
  - gdb and CDB debuggers
  - Support for version control
  - Simulator for mobile UIs
  - Support for desktop and mobile targets





```
#include "JSString.h"

namespace JSC {

ASSERT_CLASS_FITS_IN_CELL(ErrorConstructor);

ErrorConstructor::ErrorConstructor(ExecState* exec, PassRefPtr<Structure> structure, ErrorPrototype* errorPrototype)
    : InternalFunction(&exec->globalData(), structure, Identifier(exec, errorPrototype->classInfo()->className))
{
    // ECMA 15.11.3.1 Error.prototype
    putDirectWithoutTransition(exec->propertyNames().prototype, errorPrototype, DontEnum | DontDelete | ReadOnly);
    putDirectWithoutTransition(exec->propertyNames().length, jsNumber(exec, 1), DontDelete | ReadOnly | DontEnum);
}

// ECMA 15.9.3
ErrorInstance* constructError(ExecState* exec, const ArgList& args)
{
    ErrorInstance* obj = new (exec) ErrorInstance(exec->lexicalGlobalObject()->errorStructure());
    if (!args.at(0).isUndefined())
        obj->putDirect(exec->propertyNames().message, jsString(exec, args.at(0).toString(exec)));
    return obj;
}

static JSObject* constructWithErrorConstructor(ExecState* exec, JSObject*, const ArgList& args)
{
    return constructError(exec, args);
}

ConstructType ErrorConstructor::getConstructData(ConstructData& constructData)
{
    constructData.native.function = constructWithErrorConstructor;
    return ConstructTypeHost;
}

// ECMA 15.9.2
static JSValue JSC_HOST_CALL callErrorConstructor(ExecState* exec, JSObject*, JSValue, const ArgList& args)
{
    // "Error()" gives the sames result as "new Error()"
    return constructError(exec, args);
}

CallType ErrorConstructor::getCallData(CallData& callData)
{
    callData.native.function = callErrorConstructor;
    return CallTypeHost;
}
```

OVERVIEW OF WINDOWS EMBEDDED COMPACT

Windows Embedded Compact 7 introduction

- Windows Embedded Compact 7 integrates advanced technologies to rapidly build a wide range of innovative, small-footprint devices
- 32-bit, real-time, multitasking OS
  - Delivered as a granular set of components
  - Use Platform Builder tools to configure image
- Runs on x86, ARM, MIPS
- Reliable real time capabilities (256 Thread Priorities)
- Small Footprint (Kernel =~300K)
- Targeted for Low power devices
- Shared Source Code and Success Model

Kernel

- Multiple architectures:
  - X86
  - ARM (ARMv5,ARMv6 and ARMv7)
  - MIPS
- Multi-core support
- Up to 3GB of addressable physical memory
- Address Space Location Randomization

Networking

- Web services on devices support
- DLNA support
- Internet Explorer Embedded
  - JScript engine from IE8
  - Improved panning and zooming capabilities
  - Anti-aliased font rendering
  - Support for Adobe Flash Player 10.1
- Remote Desktop Client
- Network Projector

User interface

- GDI and Windowing APIs
- Support for multi touch
- Multiple display support
- Silverlight For Windows Embedded
  - Support for Silverlight 3.0
  - Plug-in for Expression Blend 3
  - Visual tools integrated in Visual Studio 2008
  - Application wizard in Platform Builder

Microsoft® Silverlight™

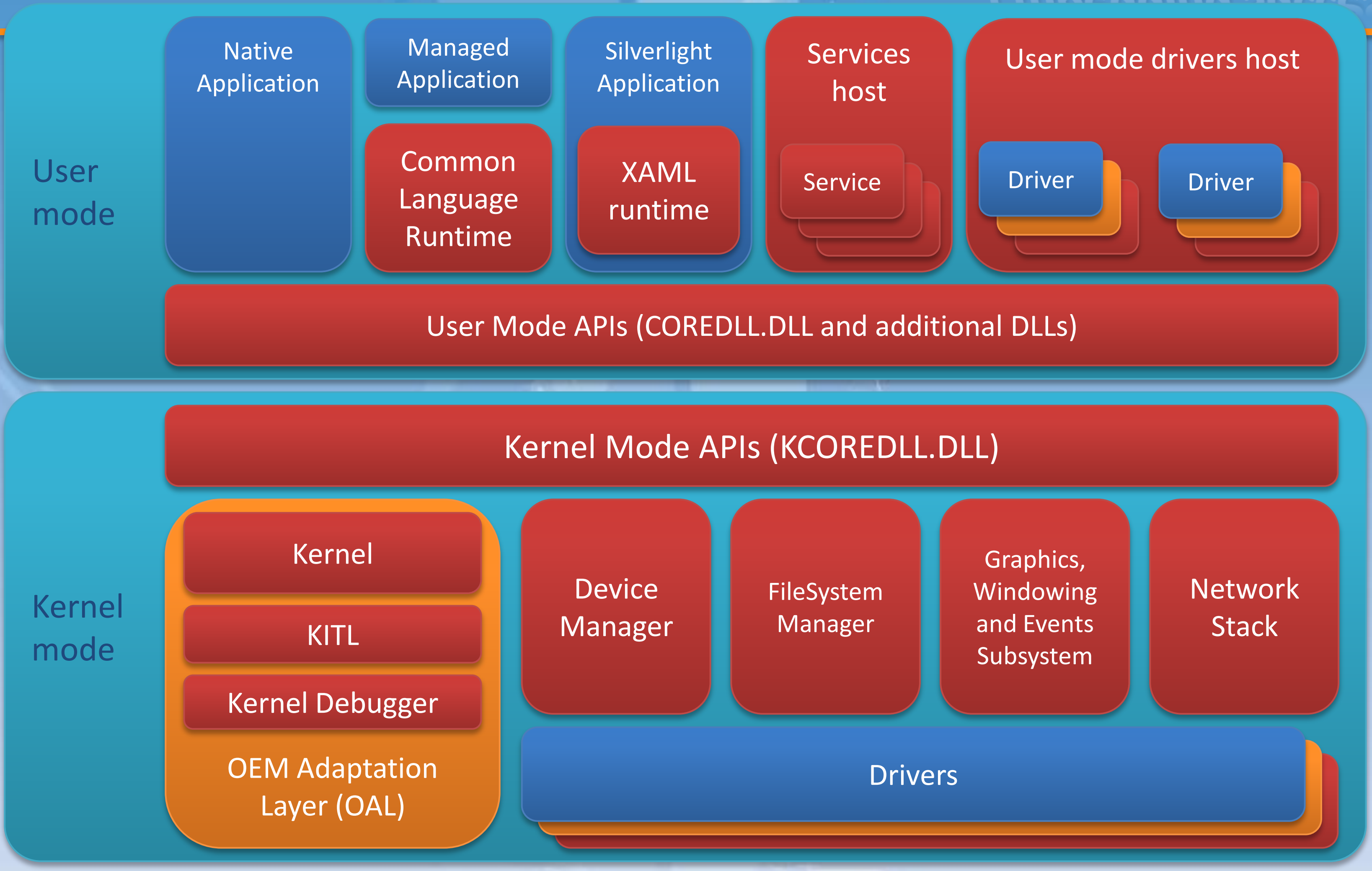
Debugging and Instrumentation

- Debugger integrated in the IDE
  - Infinite breakpoints (conditional, on data modification etc.), watches, Call stack, memory access, disassembly…
- Timeline view
  - Kernel Tracker
  - Performance Monitor
  - Power Monitor
- Profiler
- Remote tools
- Log and trace functions in the Operating System
- Post-mortem debugger support

Architecture

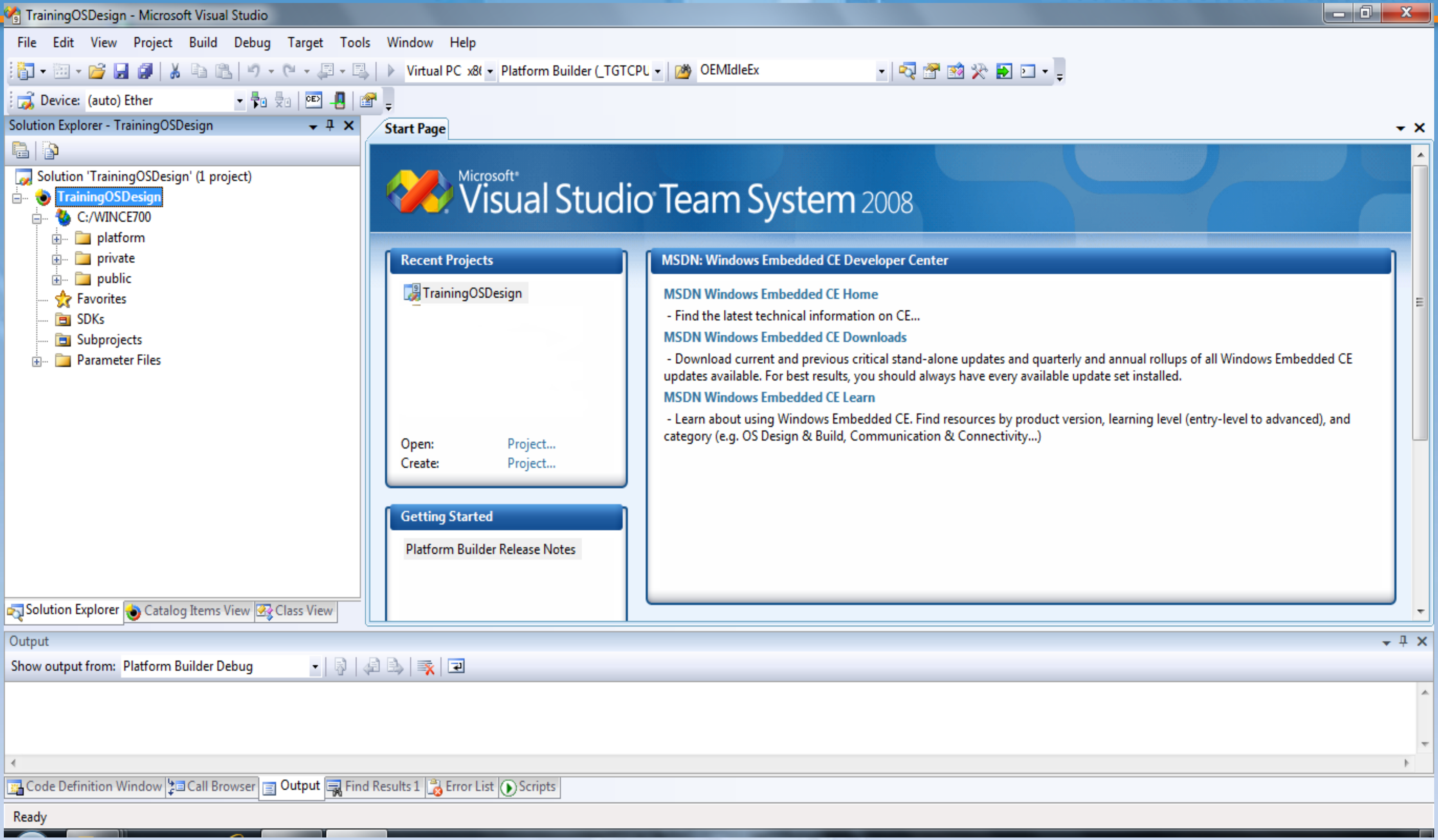
Adeneo Embedded

Embedding success



Development tools: Platform Builder

Embedding success



TrainingOSDesign - Microsoft Visual Studio
File Edit View Project Build Debug Target Tools Window Help
Virtual PC x8(Platform Builder (_TGTCPL OEMIdleEx
Device: (auto) Ether
Solution Explorer - TrainingOSDesign
Start Page
Microsoft Visual Studio Team System 2008
Solution Explorer Catalog Items View Class View
Code Definition Window Call Browser Output Find Results 1 Error List Scripts
Ready

Solution 'TrainingOSDesign' (1 project)
- TrainingOSDesign
  - C:/WINCE700
    - platform
    - private
    - public
  - Favorites
  - SDKs
  - Subprojects
  - Parameter Files

Recent Projects

TrainingOSDesign

Open: Project...
Create: Project...

Getting Started

Platform Builder Release Notes

MSDN: Windows Embedded CE Developer Center

MSDN Windows Embedded CE Home
- Find the latest technical information on CE...

MSDN Windows Embedded CE Downloads
- Download current and previous critical stand-alone updates and quarterly and annual rollups of all Windows Embedded CE updates available. For best results, you should always have every available update set installed.

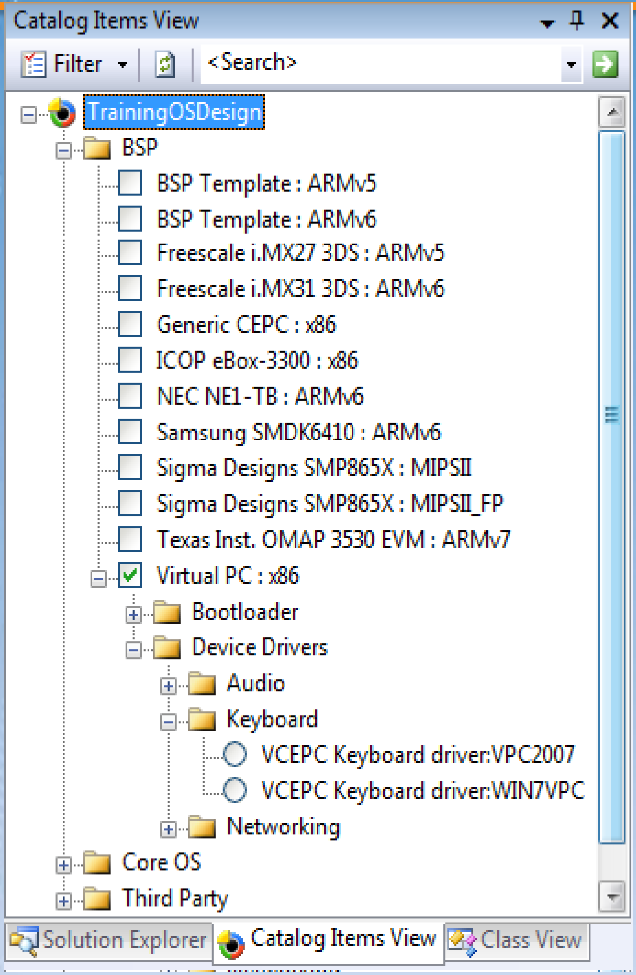
MSDN Windows Embedded CE Learn
- Learn about using Windows Embedded CE. Find resources by product version, learning level (entry-level to advanced), and category (e.g. OS Design & Build, Communication & Connectivity...)

Output

Show output from: Platform Builder Debug

The Catalog

- All the features of Windows Embedded Compact 7 are organized into components.
- A component is a set of related features that can be added to an OS Design.
- Components may depend on other components for their functions.
- The catalog provide an easy and extensible way to select components.



Catalog Items View

- TrainingOSDesign
  - BSP
    - □ BSP Template : ARMv5
    - □ BSP Template : ARMv6
    - □ Freescale i.MX27 3DS : ARMv5
    - □ Freescale i.MX31 3DS : ARMv6
    - □ Generic CEPC : x86
    - □ ICOP eBox-3300 : x86
    - □ NEC NE1-TB : ARMv6
    - □ Samsung SMDK6410 : ARMv6
    - □ Sigma Designs SMP865X : MIPSII
    - □ Sigma Designs SMP865X : MIPSII_FP
    - □ Texas Inst. OMAP 3530 EVM : ARMv7
    - ☑ Virtual PC : x86
      - Bootloader
      - Device Drivers
        - Audio
        - Keyboard
          - ○ VCEPC Keyboard driver:VPC2007
          - ○ VCEPC Keyboard driver:WIN7VPC
        - Networking
  - Core OS
  - Third Party

Application development

- Different technologies and tools can be used to develop applications on Windows Embedded Compact 7
- Some of those technologies require a runtime on the device and may require that some components are included in your OSDesign.
- A SDK can be generated to provide support for the specific set of features of your device.
- Many different application development technologies are available on Windows Embedded Compact:
  - Native APIs
  - MFC
  - ATL/WTL
  - Silverlight for Windows Embedded
  - .NET Compact Framework
  - Third party solutions (Java VMs, portable toolkits etc.)

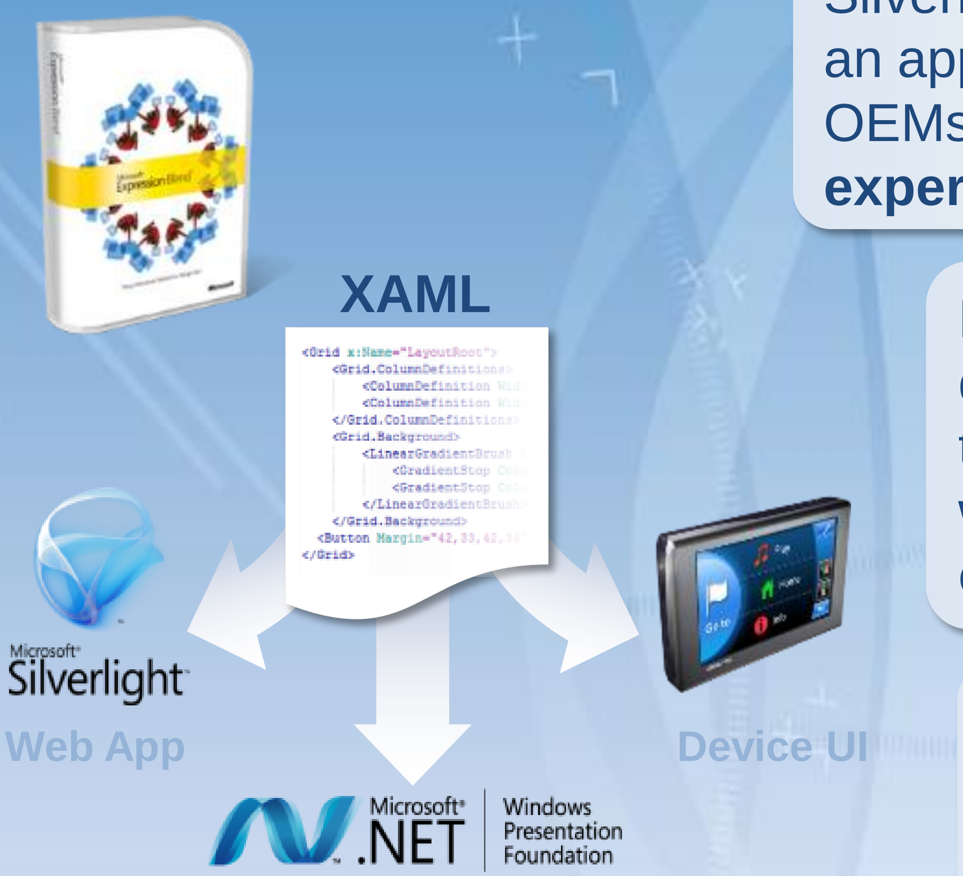
Native and managed applications

- **Native**
  - Are compiled in machine code and are platform-dependent
  - Access OS services using the Operating System APIs and can access all the OS functions.
  - Can provide real-time response times.

- **Managed**
  - Are compiled in Intermediate Language and are platform independent.
  - Access the OS service trough a class library and may require a native-code layer to access some functions.
  - Can’t be used for real-time

Silverlight for Windows Embedded

Expressio
n Blend



XAML

```
<Grid x:Name="LayoutRoot">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Wid
        <ColumnDefinition Wid
    </Grid.ColumnDefinitions>
    <Grid.Background>
        <LinearGradientBrush 
            <GradientStop Colo
            <GradientStop Colo
        </LinearGradientBrush>
    </Grid.Background>
  <Button Margin="42,33,42,34"
</Grid>
```

Web App

Device UI

Desktop App

Silverlight for Windows Embedded is an application framework that allows OEMs to **create innovative user experiences**

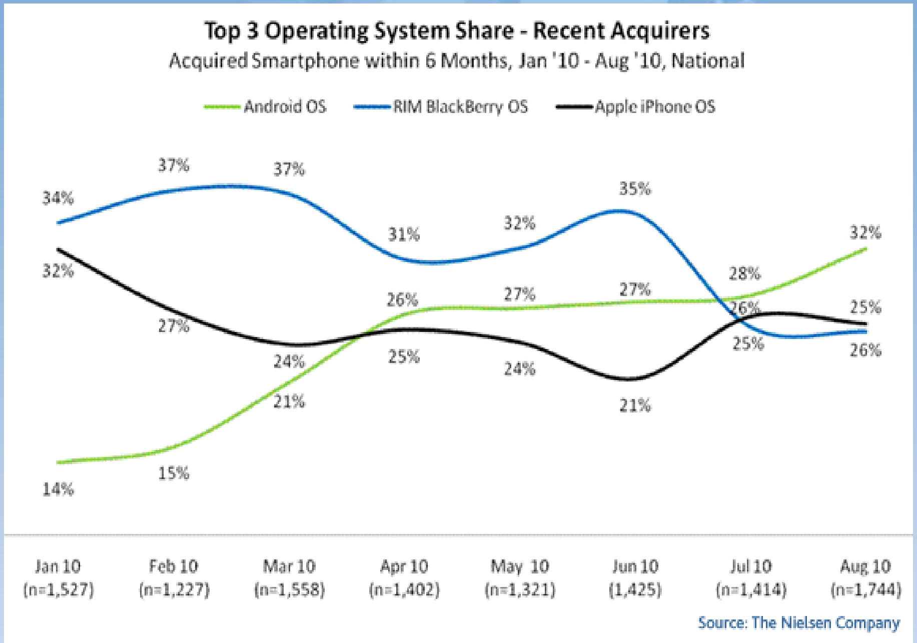
Empowers OEMs to easily differentiate their device using the Expression Design Suite while working in a **native C++** environment

OEMs can now leverage the **huge design community** of Web and desktop developers

OVERVIEW OF ANDROID

Introduction to Android

- Android is a Operating system created by Google for smart phones and tablets
- The first Android based phone was released in October of 2008
- Android based phone sales surpassed Apple and Blackberry in July of 2010



Top 3 Operating System Share - Recent Acquirers

Acquired Smartphone within 6 Months, Jan '10 - Aug '10, National

Source: The Nielsen Company

Android Phones

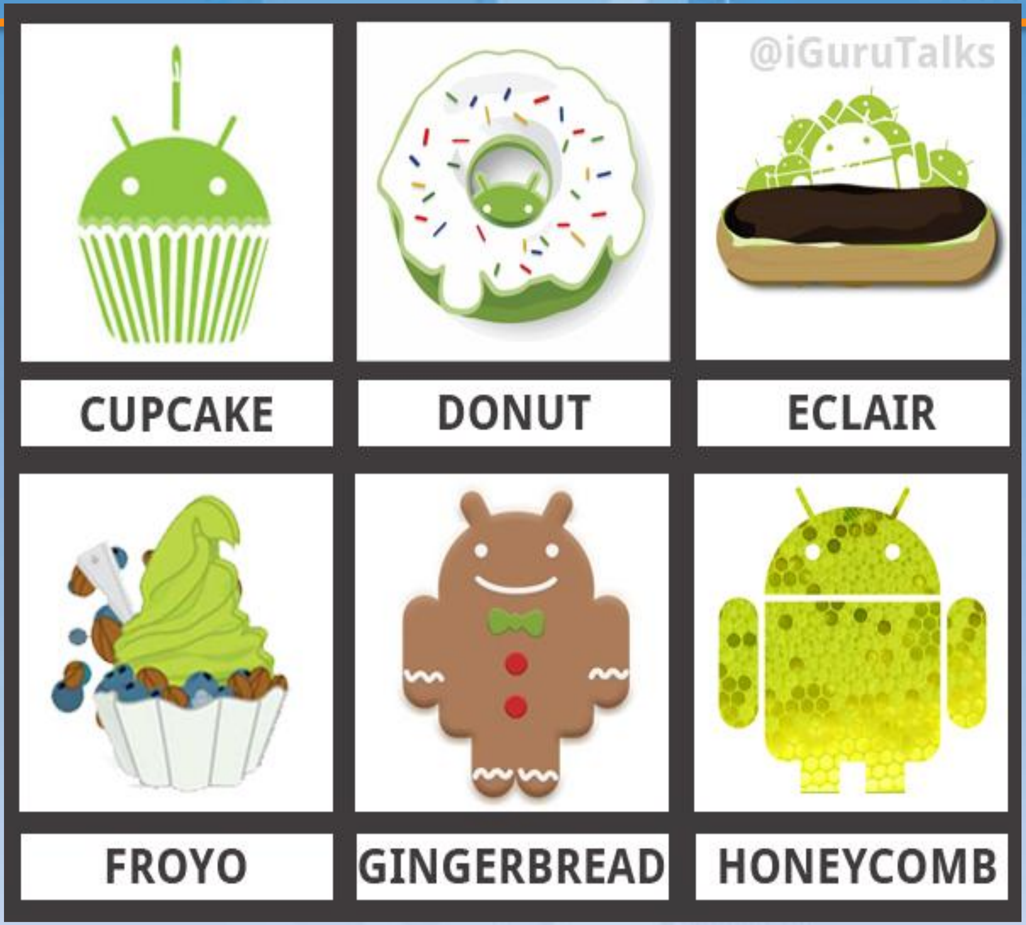
Embedding success

- Typical features:
  - 1 GHz ARM Cortex-A8
  - 16 GB Flash
  - 256 MB RAM
  - 3-axis accelerometer
  - GPS
  - Compass
  - Cameras
  - 800x480 LCD touch screens
  - Wifi, Bluetooth
  - USB

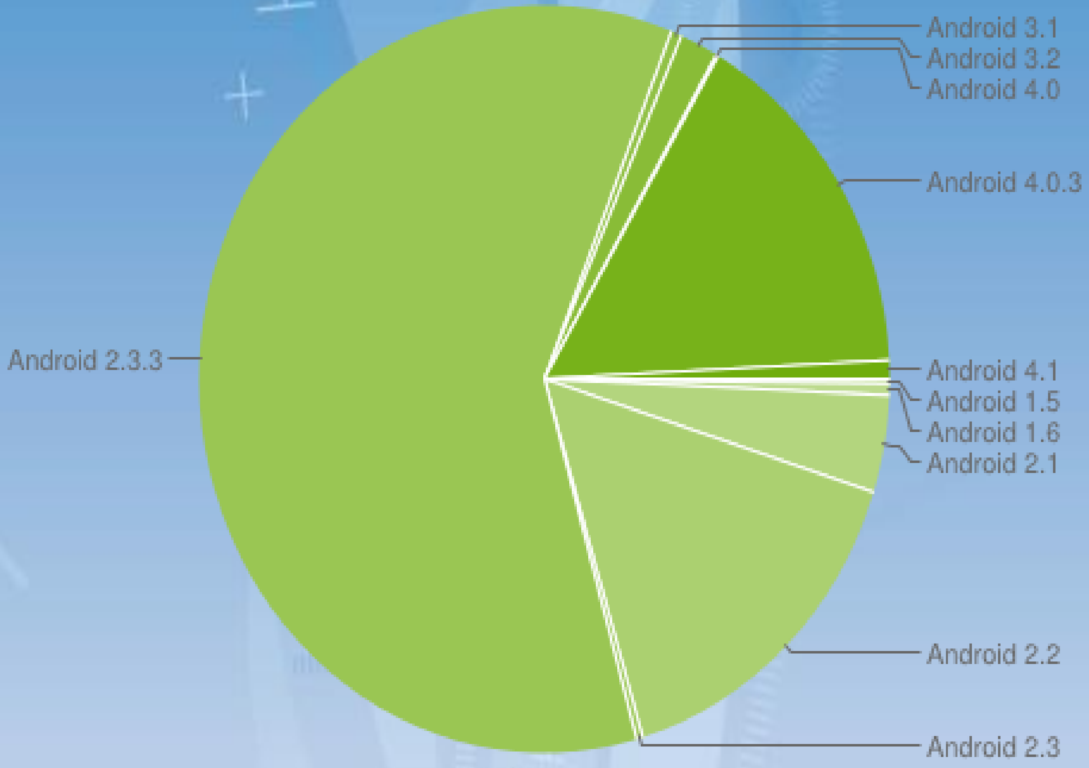
Android tablets

Android naming

Android naming



Usage share – September 2012



Android Ice Cream Sandwich (4.0)

- Android 4.0
  - API level 14-15
- Refined, evolved UI
  - Resizable widgets
  - Lock screen actions
- Voice Input
- Social API
- Unique version for tablet devices and mobile phones

Android Jelly Bean (4.1)

- Android 4.1
  - API level 16
- Project Butter
  - Improves responsiveness of the UI
- Widget resizing
- Predictive keyboard
- Google Now
- New camera application
- Android Beam enhancements (NFC)

Android is open-source

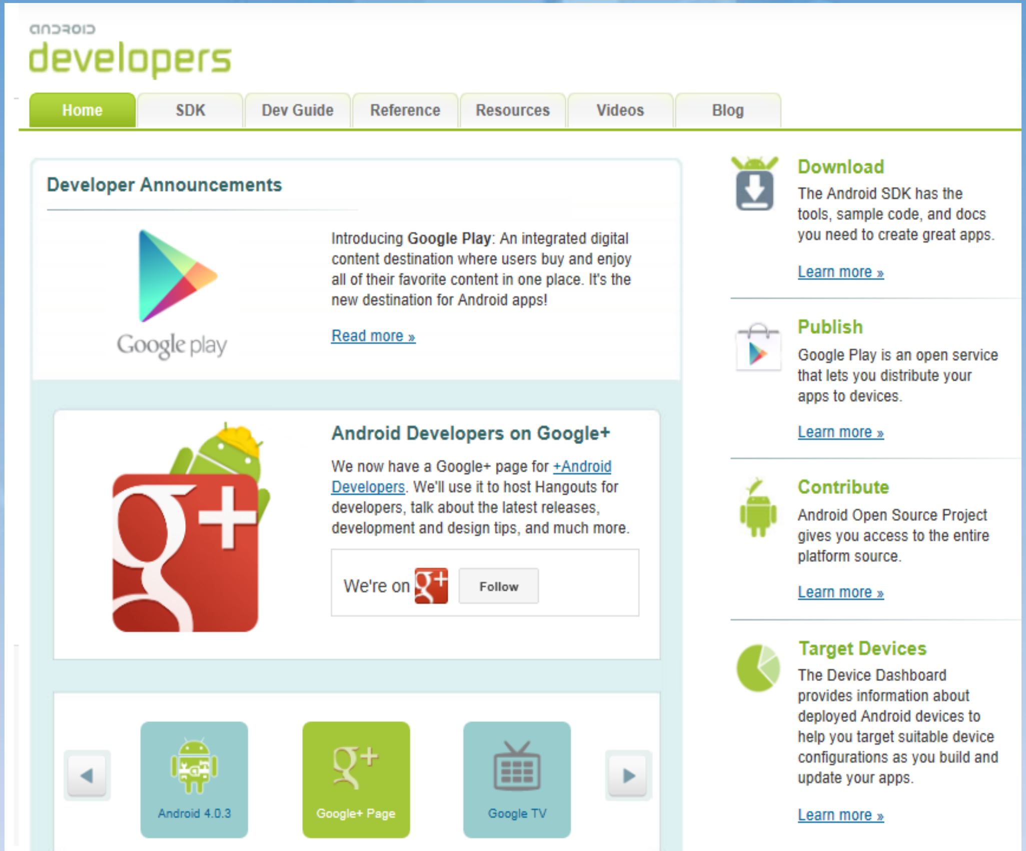
- The Android OS is open source, available for anyone to download
  - Most of the Android code is released under the Apache License
- Android runs on top of Linux, which of-course is also Open Source
- This makes Android based phones and devices completely open, and EASY to modify/Hack

developer.android.com

Adeneo Embedded

Embedding success

- For application developers (SDK)



Developer Announcements

Introducing **Google Play**: An integrated digital content destination where users buy and enjoy all of their favorite content in one place. It's the new destination for Android apps!

Read more »

Android Developers on Google+

We now have a Google+ page for +Android Developers. We'll use it to host Hangouts for developers, talk about the latest releases, development and design tips, and much more.

Download

The Android SDK has the tools, sample code, and docs you need to create great apps.

Learn more »

Publish

Google Play is an open service that lets you distribute your apps to devices.

Learn more »

Contribute

Android Open Source Project gives you access to the entire platform source.

Learn more »

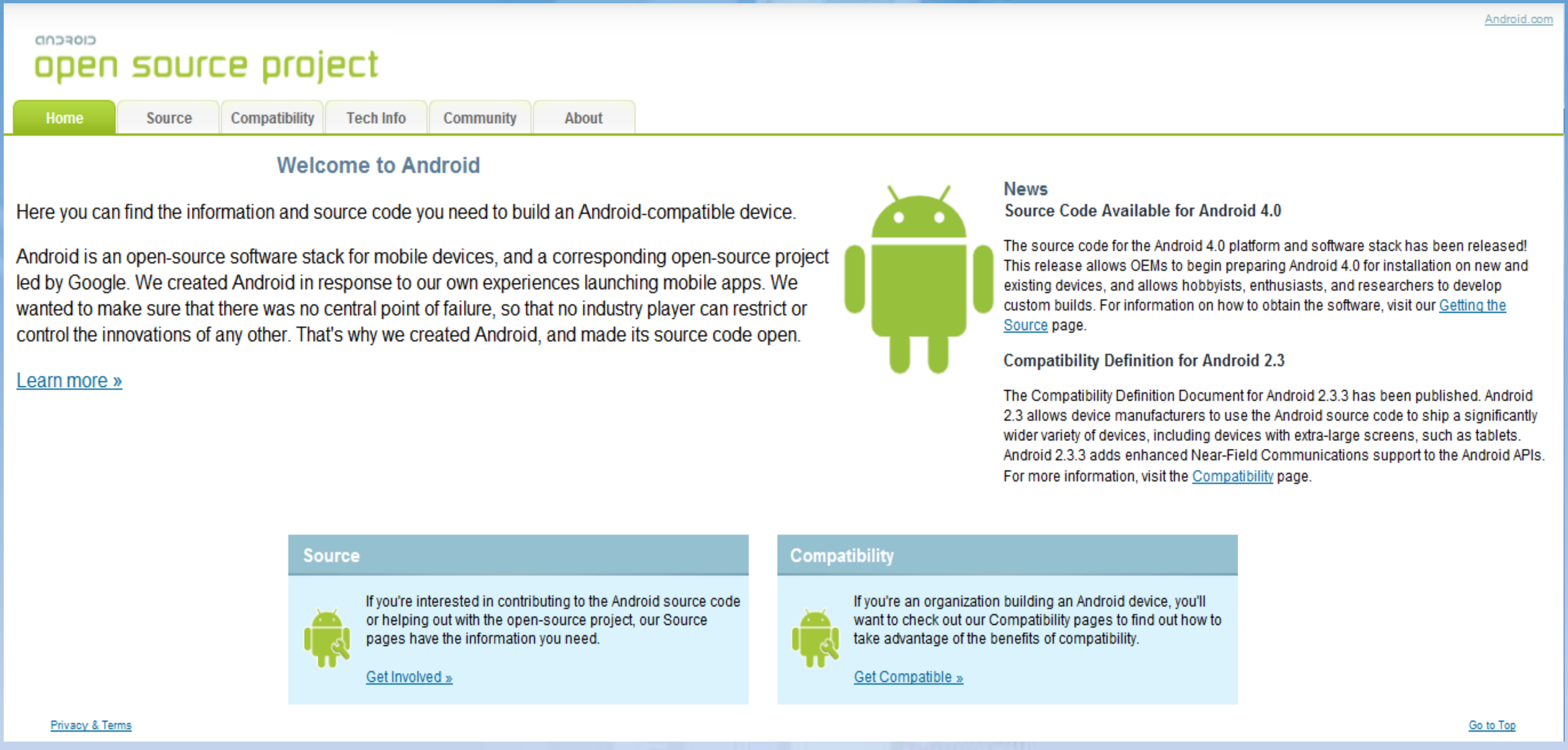
Target Devices

The Device Dashboard provides information about deployed Android devices to help you target suitable device configurations as you build and update your apps.

Learn more »

source.android.com

- For platform developers (BSP)



Welcome to Android

Here you can find the information and source code you need to build an Android-compatible device.

Android is an open-source software stack for mobile devices, and a corresponding open-source project led by Google. We created Android in response to our own experiences launching mobile apps. We wanted to make sure that there was no central point of failure, so that no industry player can restrict or control the innovations of any other. That's why we created Android, and made its source code open.

Learn more »

News

Source Code Available for Android 4.0

The source code for the Android 4.0 platform and software stack has been released! This release allows OEMs to begin preparing Android 4.0 for installation on new and existing devices, and allows hobbyists, enthusiasts, and researchers to develop custom builds. For information on how to obtain the software, visit our Getting the Source page.

Compatibility Definition for Android 2.3

The Compatibility Definition Document for Android 2.3.3 has been published. Android 2.3 allows device manufacturers to use the Android source code to ship a significantly wider variety of devices, including devices with extra-large screens, such as tablets. Android 2.3.3 adds enhanced Near-Field Communications support to the Android APIs. For more information, visit the Compatibility page.

Source

If you're interested in contributing to the Android source code or helping out with the open-source project, our Source pages have the information you need.

Get Involved »

Compatibility

If you're an organization building an Android device, you'll want to check out our Compatibility pages to find out how to take advantage of the benefits of compatibility.

Get Compatible »

Android key features

- **Connectivity**
  - Supports connectivity technologies including GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth, and Wi-Fi
- **Web browser**
  - Web browser available in Android is based on the open-source WebKit application framework
- **Media**
  - Supports the following audio/video/still media formats: H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, OGG Vorbis, WAV, JPEG, PNG, GIF, BMP
- **Hardware and graphics**
  - Can use video/still cameras, touch-screens, GPS, accelerometers, magnetometers, accelerated 2D bit blits (with hardware orientation, scaling, pixel format conversion) and accelerated 3D graphics

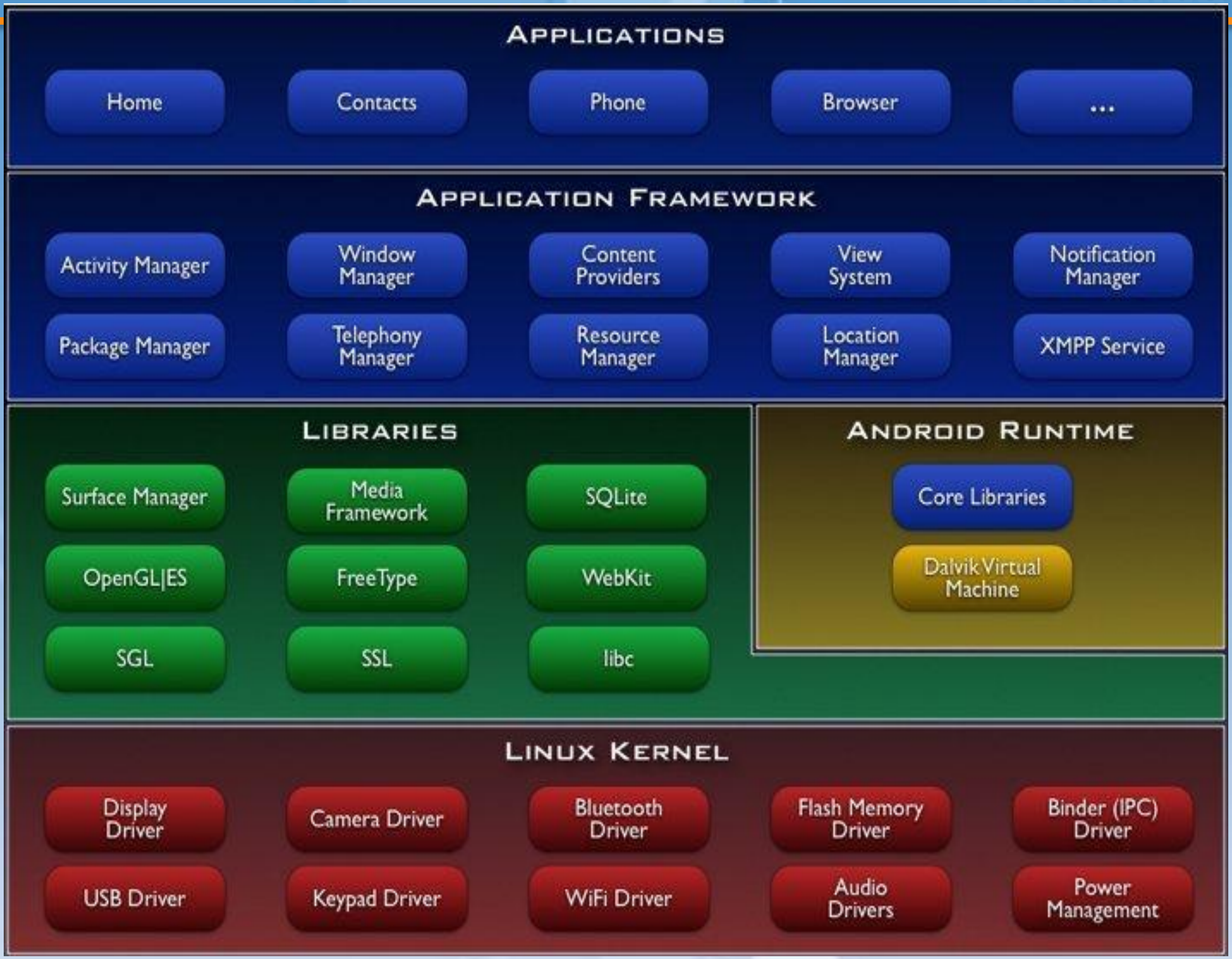
Android key features

- **Multi-touch**
  - Has native support for multi-touch which is available in newer handsets
- **Google Play (formerly “Android Marketplace“)**
  - Catalog of applications that can be downloaded and installed to target hardware over-the-air, without the use of a PC
- **Development environment**
  - Includes a device emulator, tools for debugging, memory and performance profiling, a plug-in for the Eclipse IDE

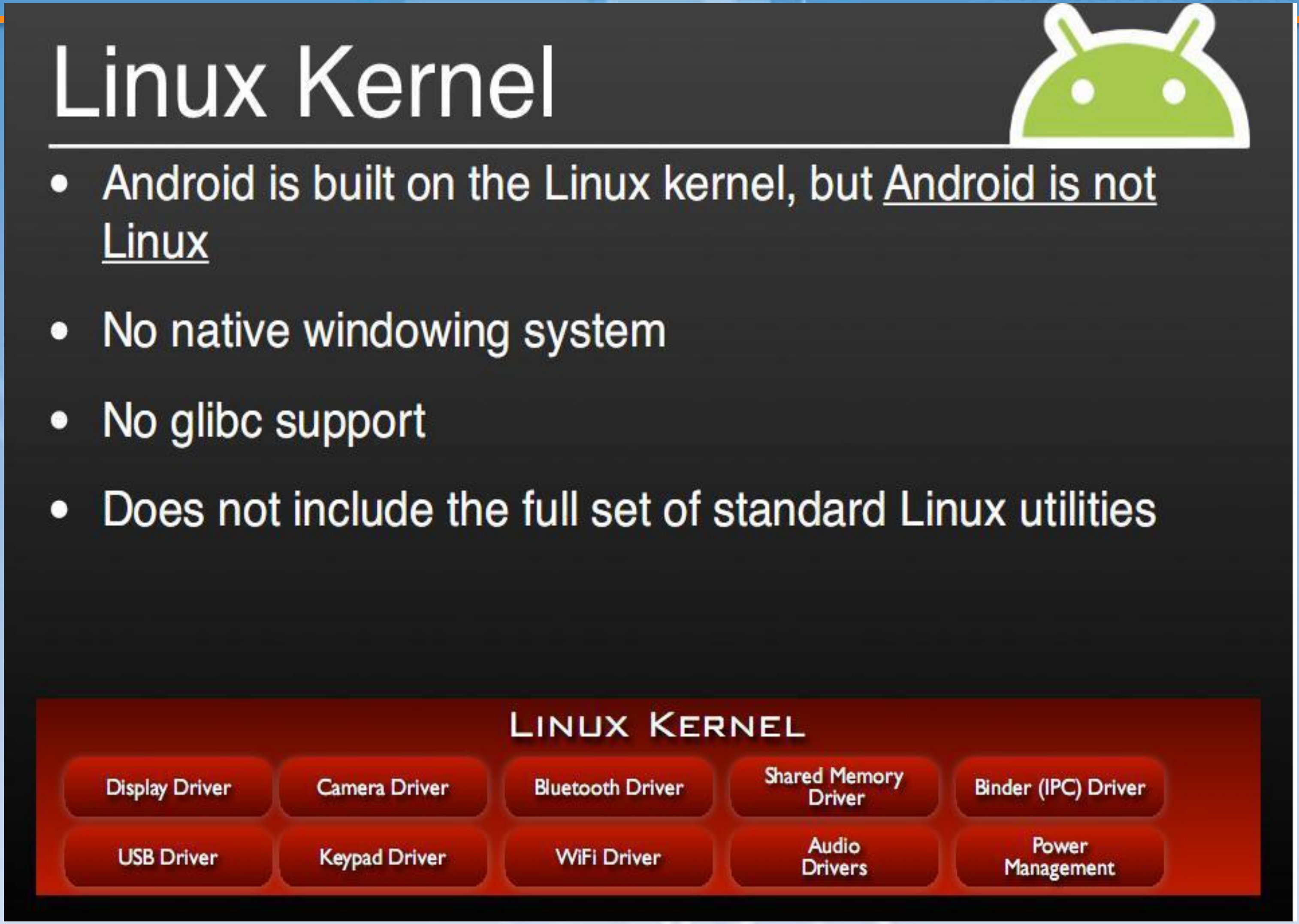
Android software stack

Adeneo Embedded

Embedding success



From “Android Anatomy and Physiology”



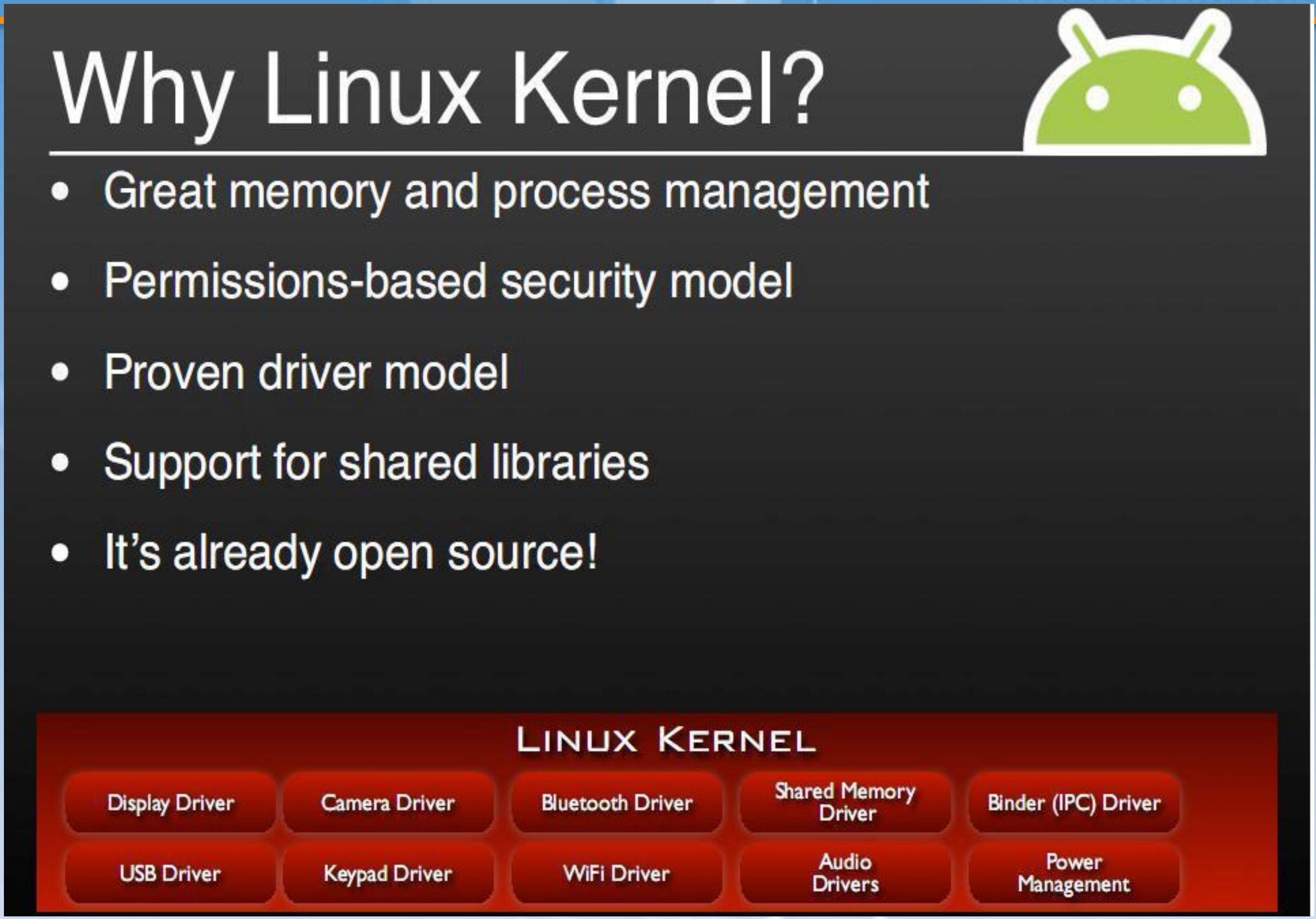
Linux Kernel

- Android is built on the Linux kernel, but Android is not Linux
- No native windowing system
- No glibc support
- Does not include the full set of standard Linux utilities

Linux Kernel

Display Driver
Camera Driver
Bluetooth Driver
Shared Memory Driver
Binder (IPC) Driver
USB Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

From “Android Anatomy and Physiology”



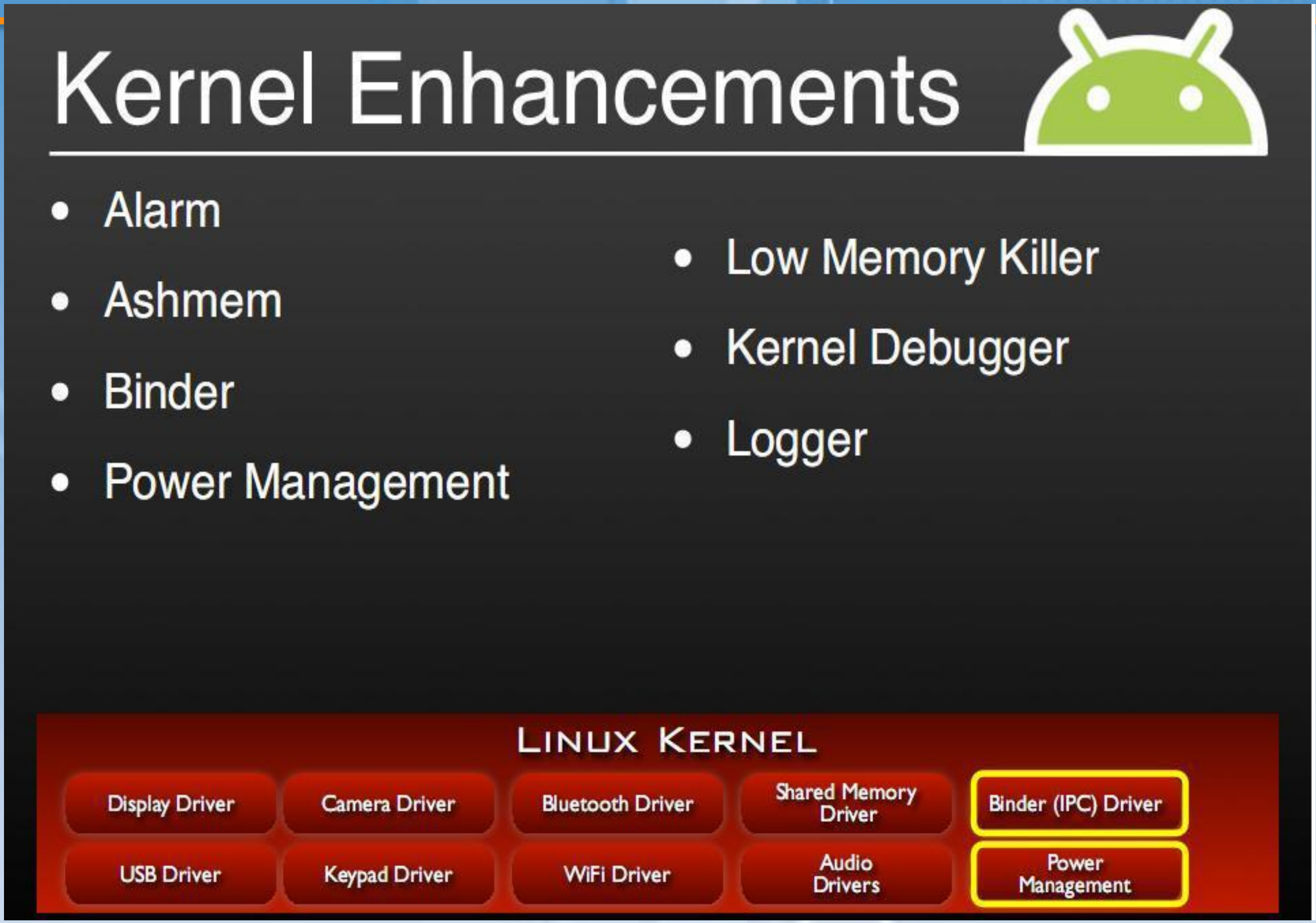
Why Linux Kernel?

- Great memory and process management
- Permissions-based security model
- Proven driver model
- Support for shared libraries
- It’s already open source!

LINUX KERNEL

Display Driver	Camera Driver	Bluetooth Driver	Shared Memory Driver	Binder (IPC) Driver
USB Driver	Keypad Driver	WiFi Driver	Audio Drivers	Power Management

From "Android Anatomy and Physiology"



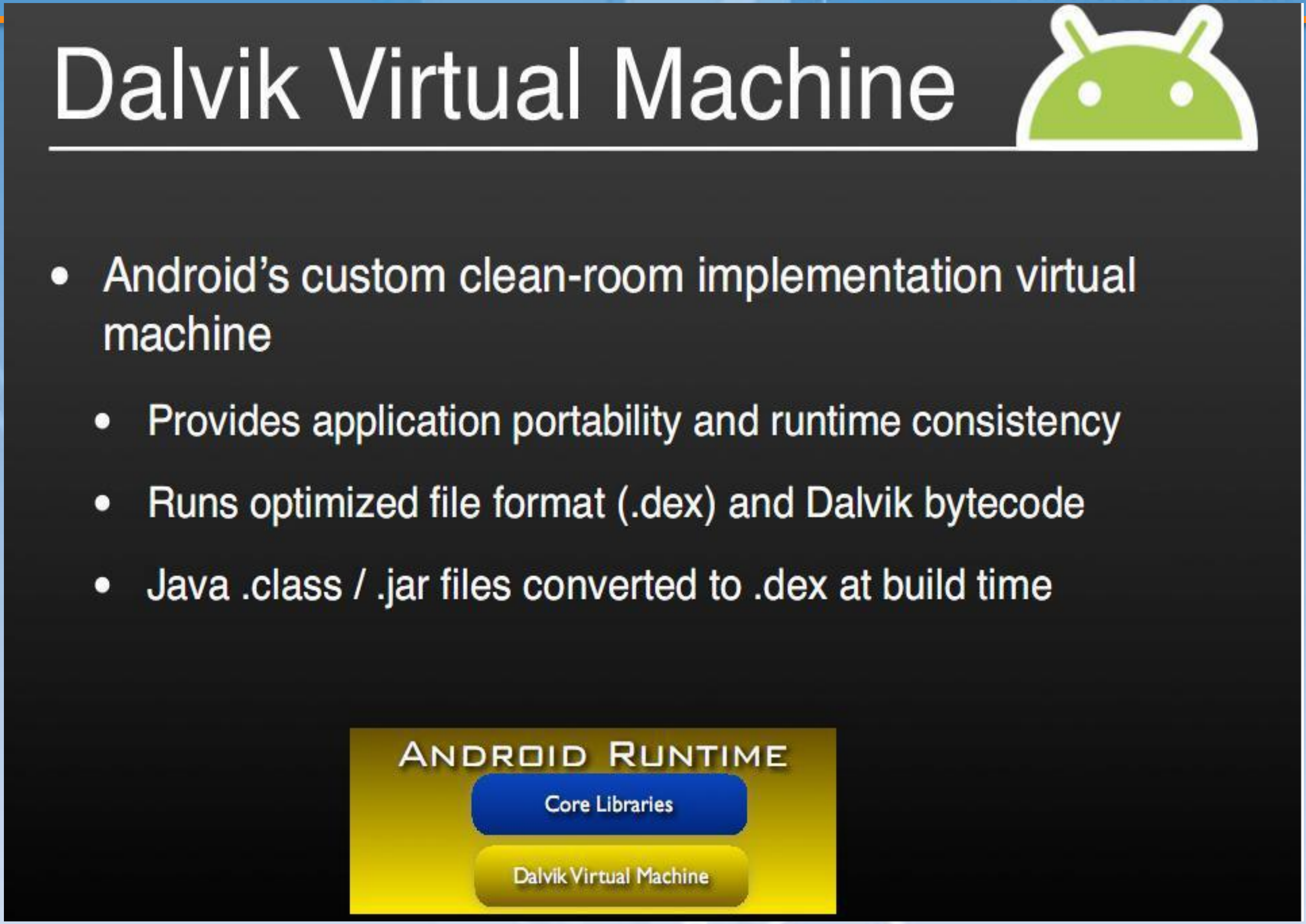
Kernel Enhancements

- Alarm
- Ashmem
- Binder
- Power Management
- Low Memory Killer
- Kernel Debugger
- Logger

LINUX KERNEL

Display Driver
Camera Driver
Bluetooth Driver
Shared Memory Driver
Binder (IPC) Driver
USB Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

From “Android Anatomy and Physiology”



Dalvik Virtual Machine

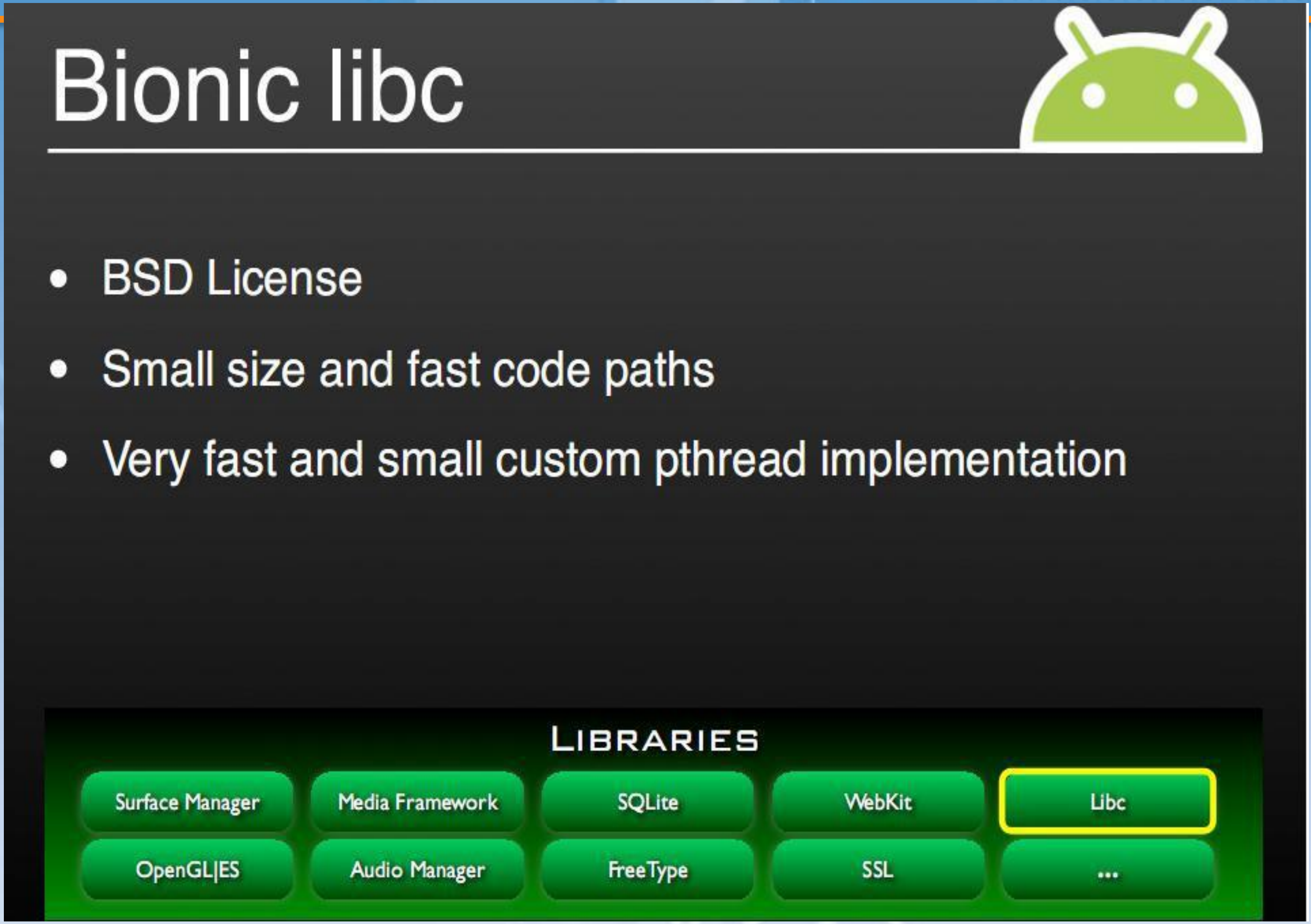
- Android’s custom clean-room implementation virtual machine
  - Provides application portability and runtime consistency
  - Runs optimized file format (.dex) and Dalvik bytecode
  - Java .class / .jar files converted to .dex at build time

ANDROID RUNTIME

Core Libraries

Dalvik Virtual Machine

From “Android Anatomy and Physiology”



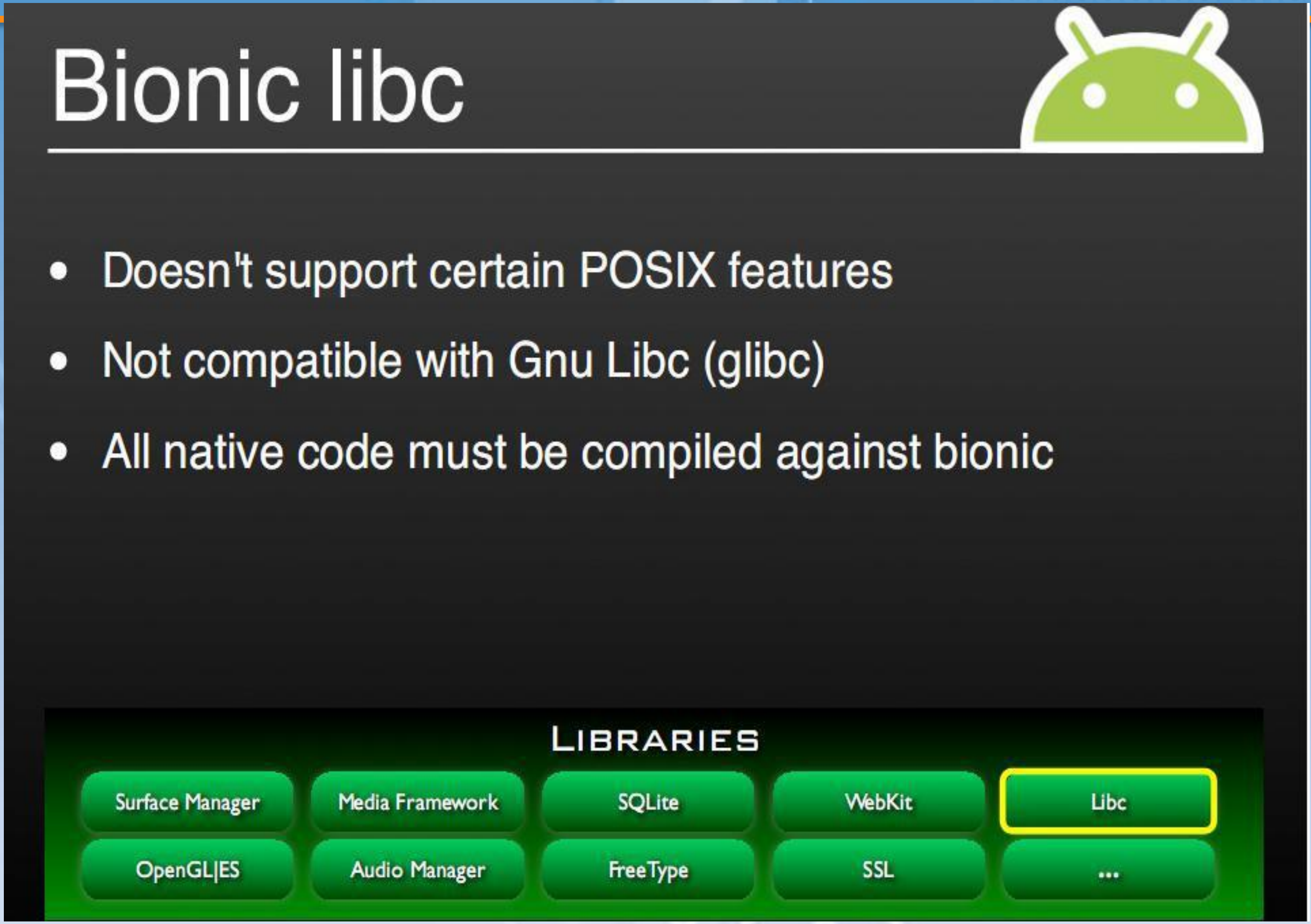
Bionic libc

- BSD License
- Small size and fast code paths
- Very fast and small custom pthread implementation

LIBRARIES

Surface Manager	Media Framework	SQLite	WebKit	Libc
OpenGL|ES	Audio Manager	FreeType	SSL	...

From “Android Anatomy and Physiology”



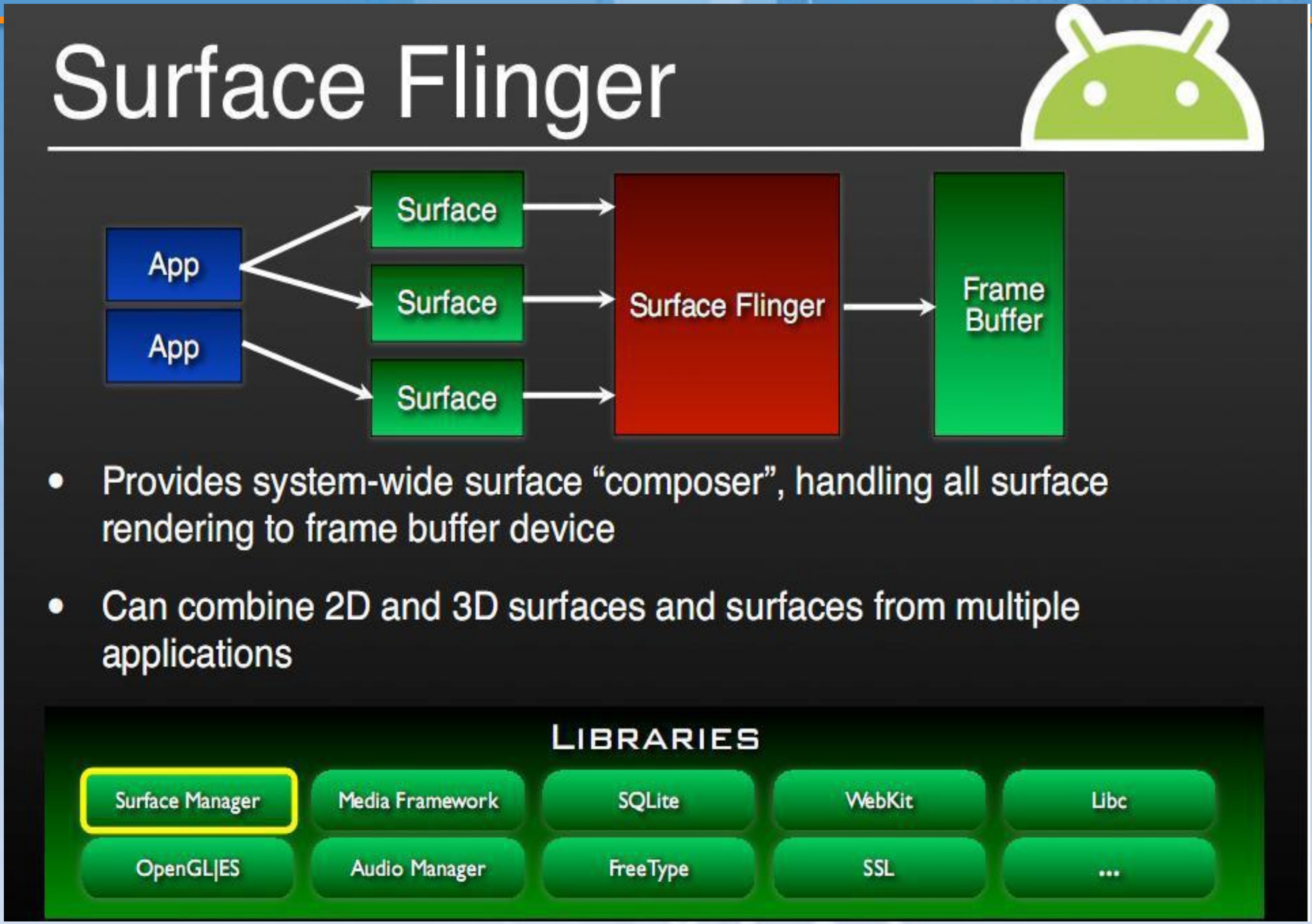
Bionic libc

- Doesn't support certain POSIX features
- Not compatible with Gnu Libc (glibc)
- All native code must be compiled against bionic

LIBRARIES

Surface Manager	Media Framework	SQLite	WebKit	Libc
OpenGL|ES	Audio Manager	FreeType	SSL	...

From “Android Anatomy and Physiology”



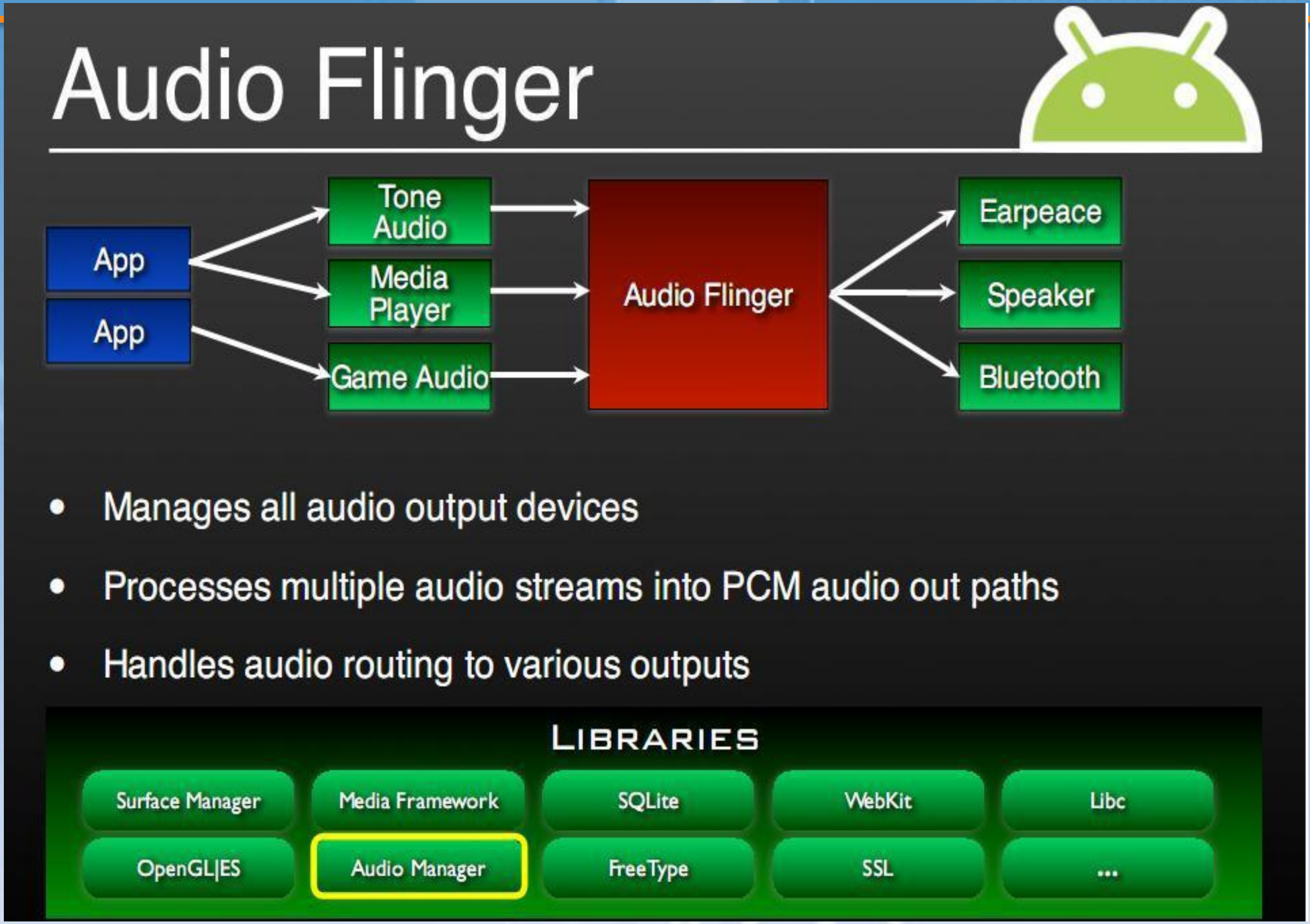
Surface Flinger

- Provides system-wide surface “composer”, handling all surface rendering to frame buffer device
- Can combine 2D and 3D surfaces and surfaces from multiple applications

LIBRARIES

Surface Manager	Media Framework	SQLite	WebKit	Libc
OpenGL|ES	Audio Manager	FreeType	SSL	...

From “Android Anatomy and Physiology”



Audio Flinger

- Manages all audio output devices
- Processes multiple audio streams into PCM audio out paths
- Handles audio routing to various outputs

LIBRARIES

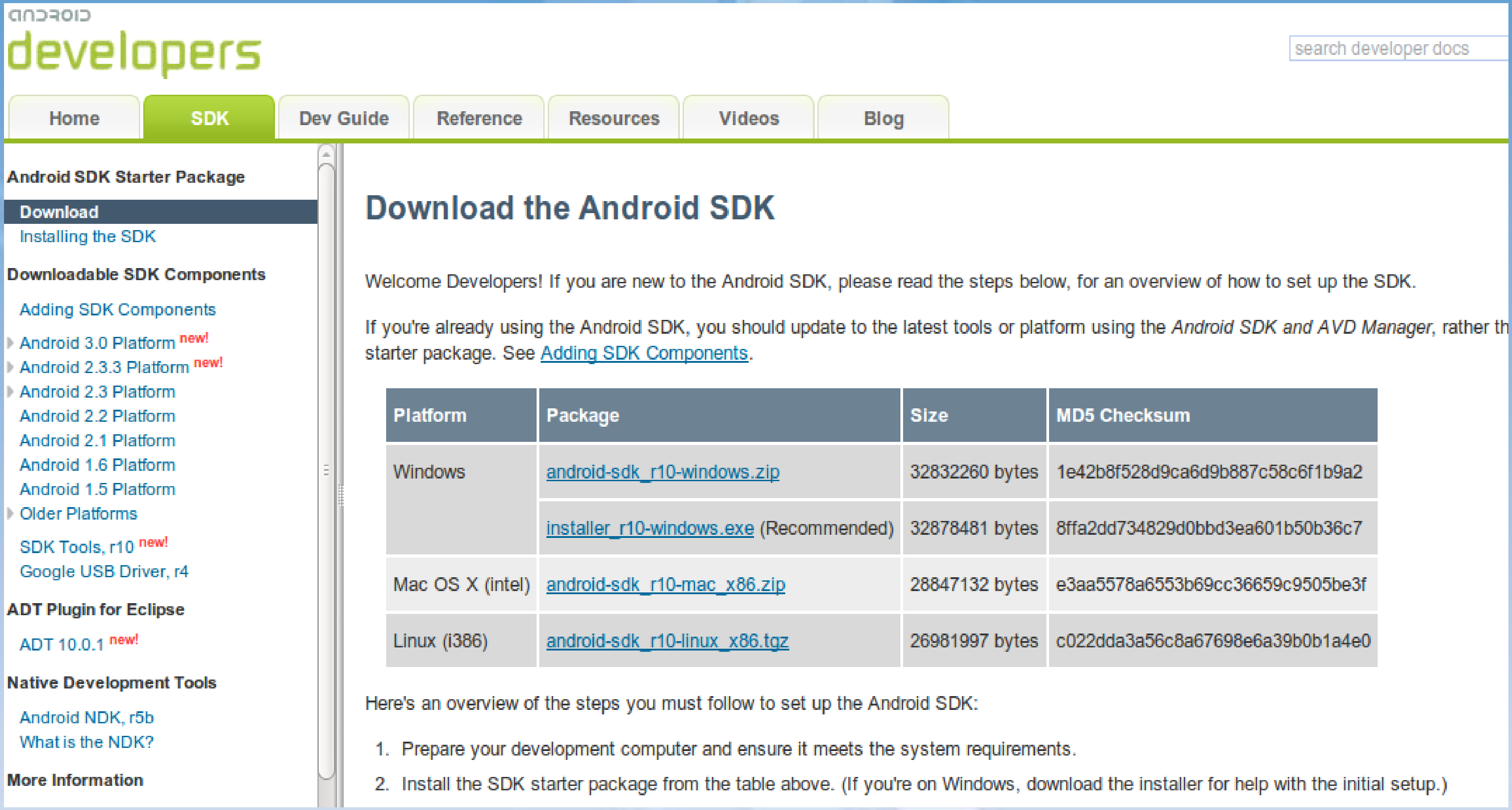
Surface Manager	Media Framework	SQLite	WebKit	Libc
OpenGL|ES	Audio Manager	FreeType	SSL	...

Applicative development

- Android development is done in Java (using the SDK), C, and C++ (using the NDK)
- The tools are provided for Windows, Linux, and Mac OSX hosts
- The tools include:
  - The Eclipse IDE (not mandatory for Android development)
  - An emulator to test your code on your computer
  - Complete debugger, to test your code directly on your Android device
- The tools are free and well supported

Downloading the Android SDK

- http://developer.android.com/sdk



Download the Android SDK

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SDK.

If you're already using the Android SDK, you should update to the latest tools or platform using the *Android SDK and AVD Manager*, rather th starter package. See Adding SDK Components.

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r10-windows.zip	32832260 bytes	1e42b8f528d9ca6d9b887c58c6f1b9a2
	installer_r10-windows.exe (Recommended)	32878481 bytes	8ffa2dd734829d0bbd3ea601b50b36c7
Mac OS X (intel)	android-sdk_r10-mac_x86.zip	28847132 bytes	e3aa5578a6553b69cc36659c9505be3f
Linux (i386)	android-sdk_r10-linux_x86.tgz	26981997 bytes	c022dda3a56c8a67698e6a39b0b1a4e0

Here's an overview of the steps you must follow to set up the Android SDK:

1. Prepare your development computer and ensure it meets the system requirements.
2. Install the SDK starter package from the table above. (If you're on Windows, download the installer for help with the initial setup.)

COMPARISON

Linux pros and cons

- **Pros**
  - Free software and open-source license
  - Very strong online community and popularity
  - UNIX based and POSIX compliant
  - Variety of supporting tools and libraries
- **Cons**
  - No standard development tools
  - Support is generally expensive
  - Learning curve may be important for people not familiar with Linux
  - Variety of supporting tools and libraries

WEC pros and cons

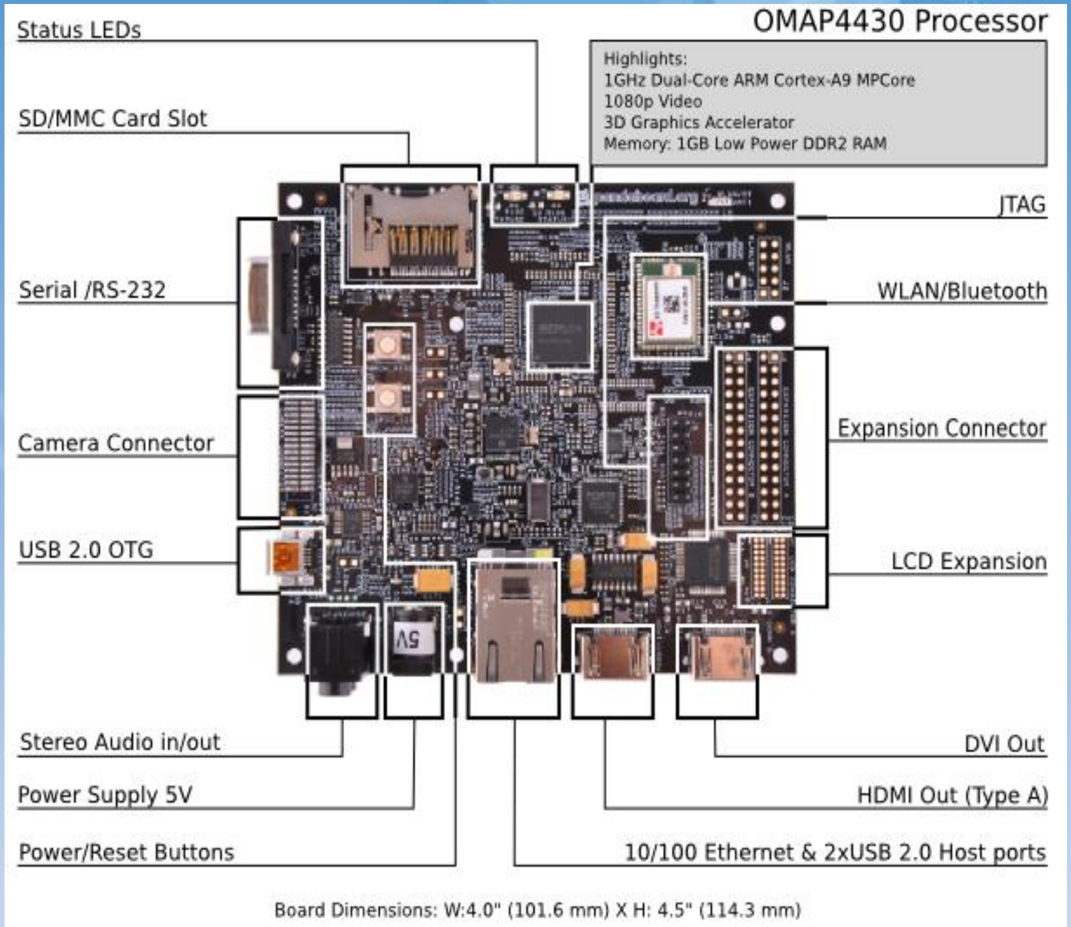
- **Pros**
  - Well-known development suite (Visual Studio)
  - .NET and Win32 APIs consistent with Windows desktop version
  - Monthly bug fixes and updates provided by Microsoft
  - Free online support from third-parties and Silicon Vendors
- **Cons**
  - Pricy license and royalty fees
  - Proprietary license and closed source parts
  - Not compliant with POSIX standards
  - Suffers from the “Microsoft” and “Windows” reputation

Android pros and cons

- **Pros**
  - Full-featured operating system for building smartphones and tablets
  - Open-source license and full source code availability
  - Free integrated development suite for developing applications
  - JAVA language for applicative development, leveraging the large JAVA developers community
- **Cons**
  - Written for smartphones and tablets, may require heavy customization to target any other kind of embedded devices
  - Not compliant with POSIX standards
  - Not designed to achieve critical real-time tasks
  - Fast development pace, making it hard for industrial products to follow up on new versions

HOW-TO GET STARTED WITH EMBEDDED OSES ON LOW-COST PLATFORMS

Linux Embedded on the PandaBoard



OMAP4430 Processor

Highlights:
1GHz Dual-Core ARM Cortex-A9 MPCore
1080p Video
3D Graphics Accelerator
Memory: 1GB Low Power DDR2 RAM

Status LEDs

SD/MMC Card Slot

JTAG

Serial /RS-232

WLAN/Bluetooth

Expansion Connector

Camera Connector

USB 2.0 OTG

LCD Expansion

Stereo Audio in/out

DVI Out

Power Supply 5V

HDMI Out (Type A)

Power/Reset Buttons

10/100 Ethernet & 2xUSB 2.0 Host ports

Board Dimensions: W:4.0" (101.6 mm) X H: 4.5" (114.3 mm)

PandaBoard characteristics

- Texas Instruments OMAP4 processor
  - Dual-core at 1GHz
  - ARM Cortex A9 architecture
  - PowerVR SGX540 core for OpenGL ES 2.0 acceleration
  - Full HD (1080p) video hardware encoder/decoder
- 1 GB DDR2-RAM
- Onboard SD/MMC slot
- Display: HDMI, DVI, LCD
- Audio: HDMI and Jack 3.5mm
- Wireless: Wifi and Bluetooth module
- Onboard 10/100 Ethernet
- USB: Host port and OTG

Available for less than **200$**

http://www.pandaboard.org/

Running Linux on the PandaBoard

- Ready-to-use Ubuntu root file systems
  - http://linaro.org
    - ➢ *Scripts to generate prebuilt images for various platforms/OSes*
  - http://omappedia.org/wiki/OMAP_Ubuntu_Main
    - ➢ *Prebuilt images provided by OmapPedia for OMAP based platforms*
- Ready-to-use minimal Linux file system
  - http://www.omappedia.com/wiki/Minimal-FS
    - ➢ *Small root file systems for fast minimal Linux environment bringup*
- Angstrom distribution for full customization of the root file system
  - http://www.openembedded.org
    - ➢ *Full distribution allowing complete system customization*
    - ➢ *Very long to compile the first time, requires ramping-up on the environment*

Android 2.3 on the i.MX53 QSB

i.MX53 Quick Start Board characteristics

- Freescale i.MX53 processor
  - Single core at 1 GHz
  - ARM Cortex A8 architecture
  - Onchip GPU for OpenGL ES 2.0
  - Onchip VPU for video encoding/decoding
- 1 GB DDR3-RAM
- Onboard SD/MMC slot
- Display: VGA, HDMI, LVDS, LCD
- Audio: Jack 3.5mm
- SATA connector
- Onboard 10/100 Ethernet
- USB: Host port and OTG port

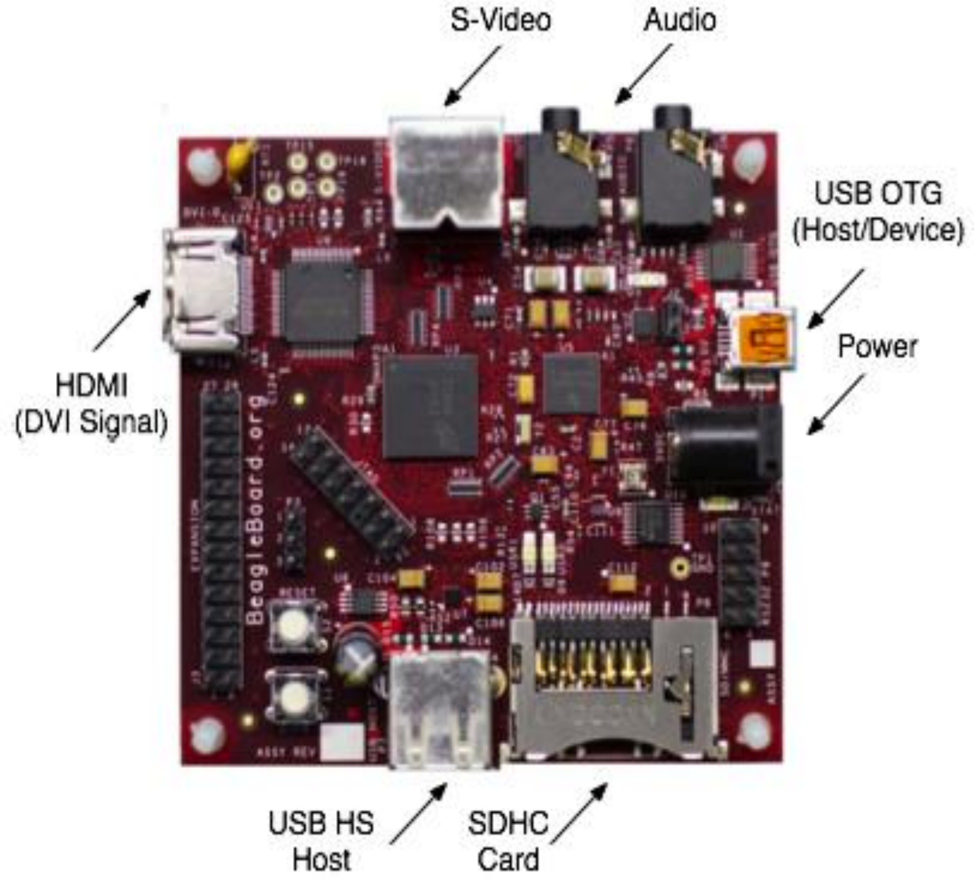
Available for **149$**

http://www.freescale.com

Running Android on the i.MX53 QSB

- Ready-to-use Android root file system
  - http://linaro.org
    - *Scripts to generate prebuilt images for various platforms/OSes*
- Ready-to-use Android demo for the i.MX53 QSB
  - http://www.adeneo-embedded.com
    - *Fast bring-up for application development*
- Full source BSP and patches for customization of Android
  - http://www.adeneo-embedded.com
    - *Full source code access for Android 2.3 (Gingerbread)*
    - *All features of the i.MX53 QSB are supported in the BSP*
    - *Free support community: http://www.imxcommunity.org*

WEC7 on the BeagleBoard



Windows Embedded Compact 7

BeagleBoard xM characteristics

- Texas Instruments DM3730 processor
  - Single core at 1GHz
  - ARM Cortex A8 architecture
  - PoweVR SGX core for OpenGL ES 2.0 acceleration
- 512 MB LPDDR-RAM
- Onboard SD/MMC slot
- Display: DVI, LCD
- Audio: Jack 3.5mm
- Onboard 10/100 Ethernet
- USB: Host port and OTG port

Available for **149$**

http://beagleboard.org

Running WEC7 on the BeagleBoard

Embedding success

- Download ready-to-use prebuilt image
  - http://www.adeneo-embedded.com
- Download Visual Studio 2008 professional 90-days trial and service packs
  - http://www.microsoft.com/download/
- Download Windows Embedded Compact 7 180-days trial
  - http://www.microsoft.com/download/
- Download full source BSP for the i.MX53 QSB (350€ or 20-days evaluation)
  - http://www.adeneo-embedded.com

Sabrelite board characteristics

- Freescale iMX6
  - Quad core at 1GHz
  - ARM Cortex A9 architecture
  - Onchip GPU for OpenGL ES 2.0
  - Onchip VPU for video encoding/decoding
- 1 GB DDR3-RAM
- Onboard SD/MMC slot
- Display: HDMI, LVDS
- Audio: Jack 3.5mm
- SATA
- PCIe
- Onboard 10/100 Ethernet
- USB: Host port and OTG port

Available for **199$**

http://boundarydevices.com

Running OS on the Sabrelite

Embedding success

WEC7

- Download ready-to-use prebuilt image
  - http://www.adeneo-embedded.com
- Download Visual Studio 2008 professional 90-days trial and service packs
  - http://www.microsoft.com/download/
- Download Windows Embedded Compact 7 180-days trial
  - http://www.microsoft.com/download/
- Download full source BSP for the QSB (4000€ or 20-days evaluation)
  - http://www.adeneo-embedded.com

Linux/Android

- Download BSP and ready-to-use prebuilt image
  - http://www.freescale.com

Conclusion

- ARM-based embedded platforms generally require more work to bring-up operating systems than x86
- Various embedded operating systems are available, each of them targeting different products and goals
- Software for embedded platforms tends to be more and more accessible to usual desktop developers
- Various full-featured ARM-based platforms are affordable for hobbyists to play with
- Anyone can start developing for these embedded OSes starting from free online resources

Questions

apellerin@adeneo-embedded.com