connect your device to application

0xdroid – community-developed Android distribution by 0xlab

Jim Huang（黃敬群）, 0xlab
OSDC.tw – Apr 25, 2010

connect your device to application

0xdroid – ***community***-developed Android distribution by 0xlab

社群

connect your device to application

0xdroid – community-developed Android ***distribution*** by 0xlab

發布

connect your device to application

0xdroid – community-developed ***Android*** distribution by 0xlab

最後才是 Android

connect your device to application

0xdroid –
Community-developed Android distribution by 0xlab

“0xlab” 與” 0xdroid" 開頭字母都是數字零 (0)

核心概念

- 在開放的硬體平台，搭建開放的軟體 (Distribution)
- 透過開放原始碼的力量，將成果累積 (Community)

connect your device to application

connect your device to application

0xdroid 不僅是個 Android 爲基礎的專案，還是累積創新的社群平台

0xdroid(引用 COSCUP 2009 的議程簡報〈 How Android Differs from GNU/Linux? And How can we FIX it? 〉)

- 快速集中工作成果，提供可用的版本
- 專為懶人設計 (installer)
- 更加透明的開發 (issue tracking)
- 工作成果要能被重複使用 (patch based)

- http://gitorious.org/0xdroid

作爲創新的準備 — Distribution

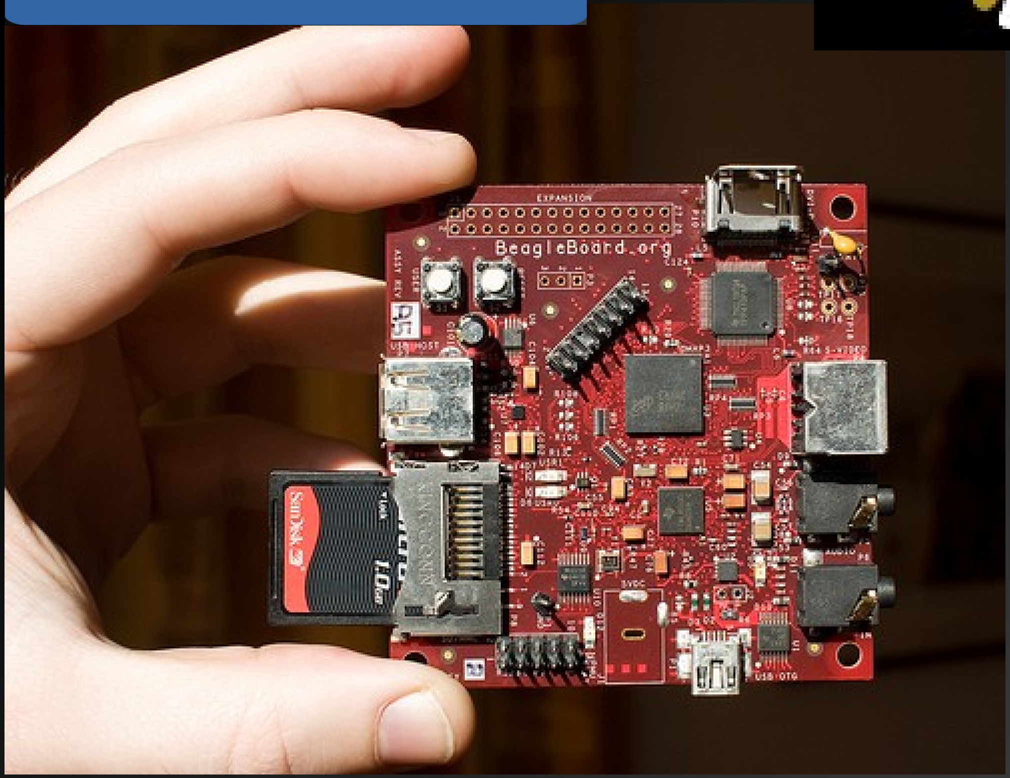
- 選定開放的硬體平台
  - Beagleboard
- 在 Android 官方原始碼發行的基礎上，充分支援開放硬體

作爲創新的準備 — Community

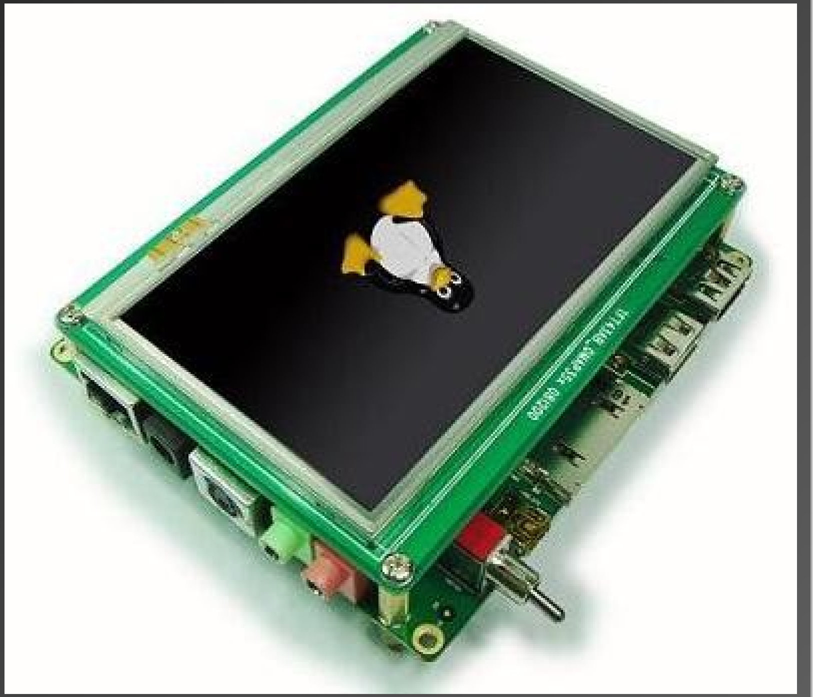
- 除了維護 0xlab 的開放原始碼專案外，與其他專案保持正面互動 (source code-level)
  - Android, Android-x86, Rowboat, CyanogenMod, OESF, ODROID, ...

Go 0xdroid!

Beagleboard



DevKit8000



Go 0xdroid!

Beagleboard

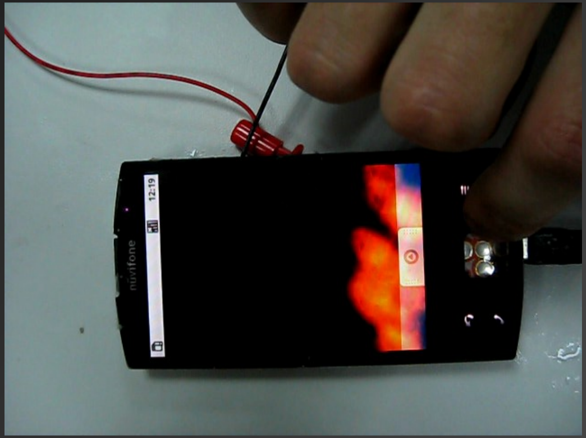
DevKit8000

Demo Video:

http://www.youtube.com/watch?v=OGpYk1p1UPI

在其他平台共享成果

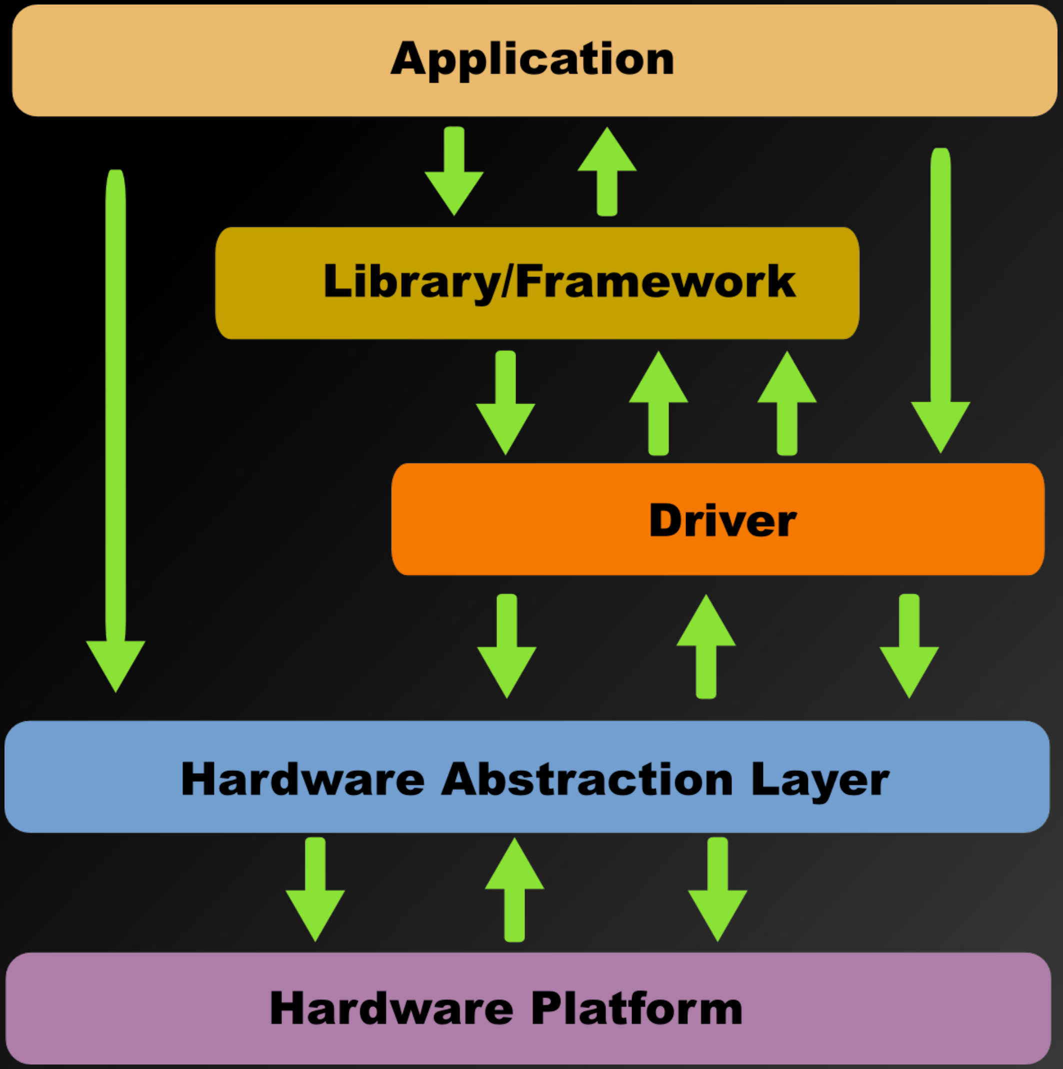
ASUS EeePC



技術只是基礎，唯有開放與合作，才能讓（嵌入式系統的）軟體層次提昇

- 以 Android 作爲切入點，保持開放共享、協同合作的態度，讓硬體的應用增添更多可能性
- 打破軟體應用的藩籬

不僅只是移植或增添硬體支援



- HAL 將硬體抽象化，使軟體工程師不必花太多心思去考慮程式將在何種硬體上執行

或是剔除原有系統的瑕疵

Lucky!

We encountered the "bug" in Android accidently

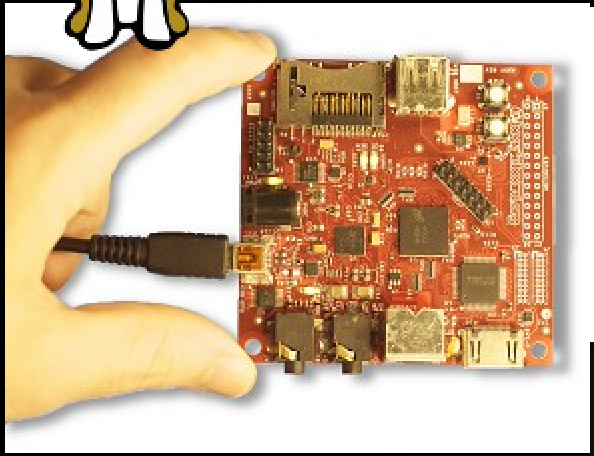
更重要的是，知識累積與開放原始碼

- 0xlab 成員的背景
  - 一群台灣的工程人員，熱衷於開放原始碼與消費性電子產品研發，附加骨子裡的的叛逆情愫
- 0xlab 成員過去的貢獻
  - Mesa/3D, FreeType, GNU GCC, Xorg/FreeDesktop, Linux Kernel, Openmoko（第一個開放原始碼的手機平台）, OpenEmbedded, LXDE, Debian GNU/Linux, FreeBSD, New Chewing（新酷音輸入法）, OpenVanilla（開放香草輸入法框架）, Kaffe, SCIM, PCManX, PCManFM, Qt Extended/Qtopia, Opkg, FFmpeg/MPlayer, OpenOCD, ...

Working Model

- 0xlab delivers the advantages of open source software and development
  - 快速引入新技術，連帶社群的大量測試與回饋
  - 建立品質控管的機制
  - 與其他開放原始碼專案合作： CyanogenMod, Android-x86, ODROID, OESF, …
  - Cooperation with Business Partners/Customers upon the refined Android codebase

Working Model from 0xdroid



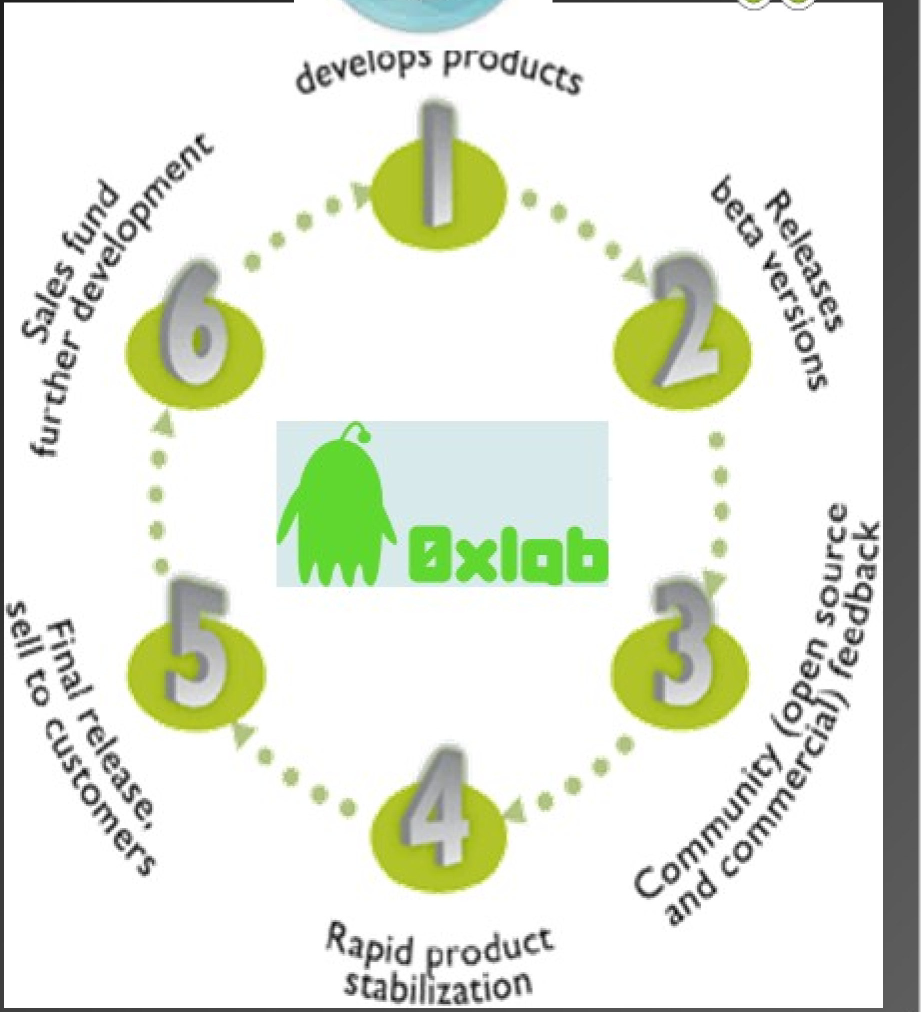
Rowboat

(TI OMAP3)

CyanogenMod

(Qualcomm 7k/8k)

Android-x86

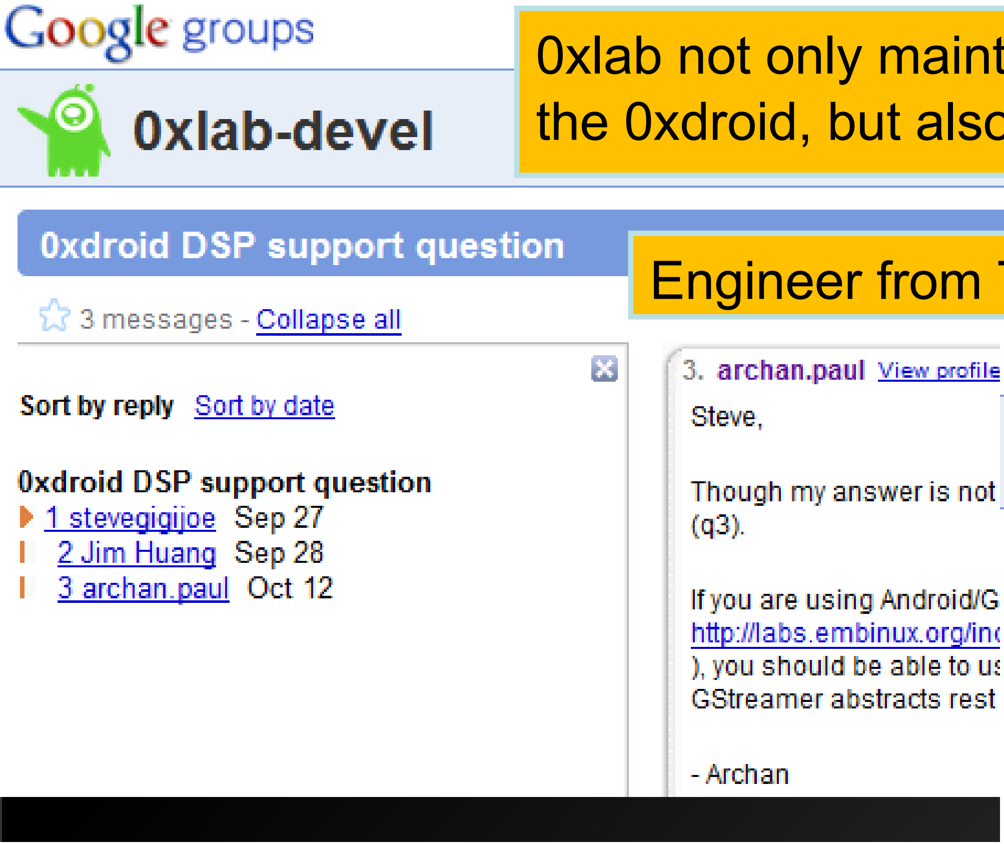

develops products
1
Releases beta versions
2
Community (open source and commercial) feedback
3
Rapid product stabilization
4
Final release, sell to customers
5
Sales fund further development
6
0xlab

Case Study: 0xdroid & android-x86

- 原本 0xdroid 與 android-x86 專案各自維護一套 software cursor 實做
  - 交叉對照、相互貢獻後，現在共用一致的程式碼

- 0xlab 在 2009 年中，根基於 Mesa/3D ，發展了世界上第一個（也是唯一的）開放原始碼的 libhgl (Hardware OpenGL|ES Acceleration for Android) ，立即被 Android-x86 專案採納，獲得廣泛測試
- 其他：圖形處理效能 , Dalvik VM, 3G modem, ...

0xlab not only maintains a full open source Android distribution, the 0xdroid, but also established a community with opened mind.



Google groups

0xlab-devel

0xdroid DSP support question

3 messages - Collapse all

Sort by reply   Sort by date

0xdroid DSP support question

1 stevegigijoe	Sep 27
2 Jim Huang	Sep 28
3 archan.paul	Oct 12

3. archan.paul View profile

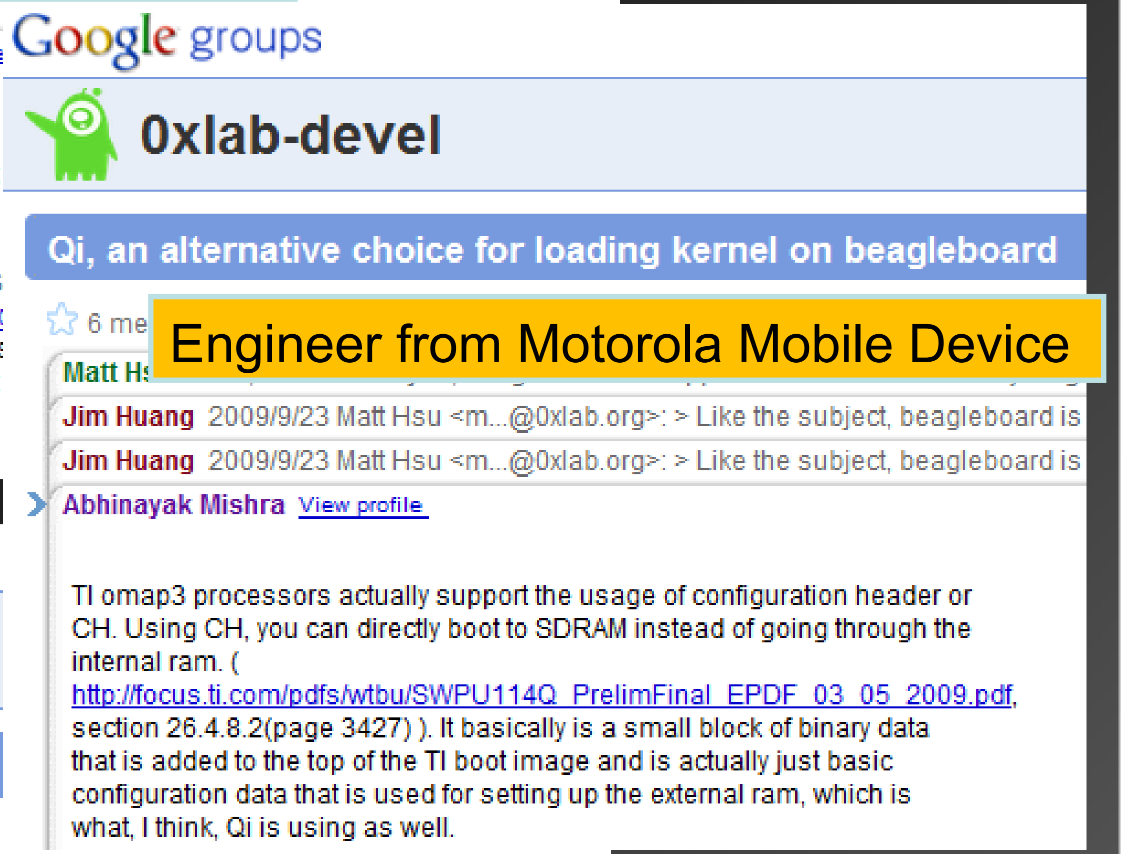
Steve,

Though my answer is not
(q3).

If you are using Android/G
http://labs.embinux.org/inc
), you should be able to us
GStreamer abstracts rest

- Archan

Engineer from TI/embinux



Google groups

0xlab-devel

Qi, an alternative choice for loading kernel on beagleboard

6 me

Matt Hs

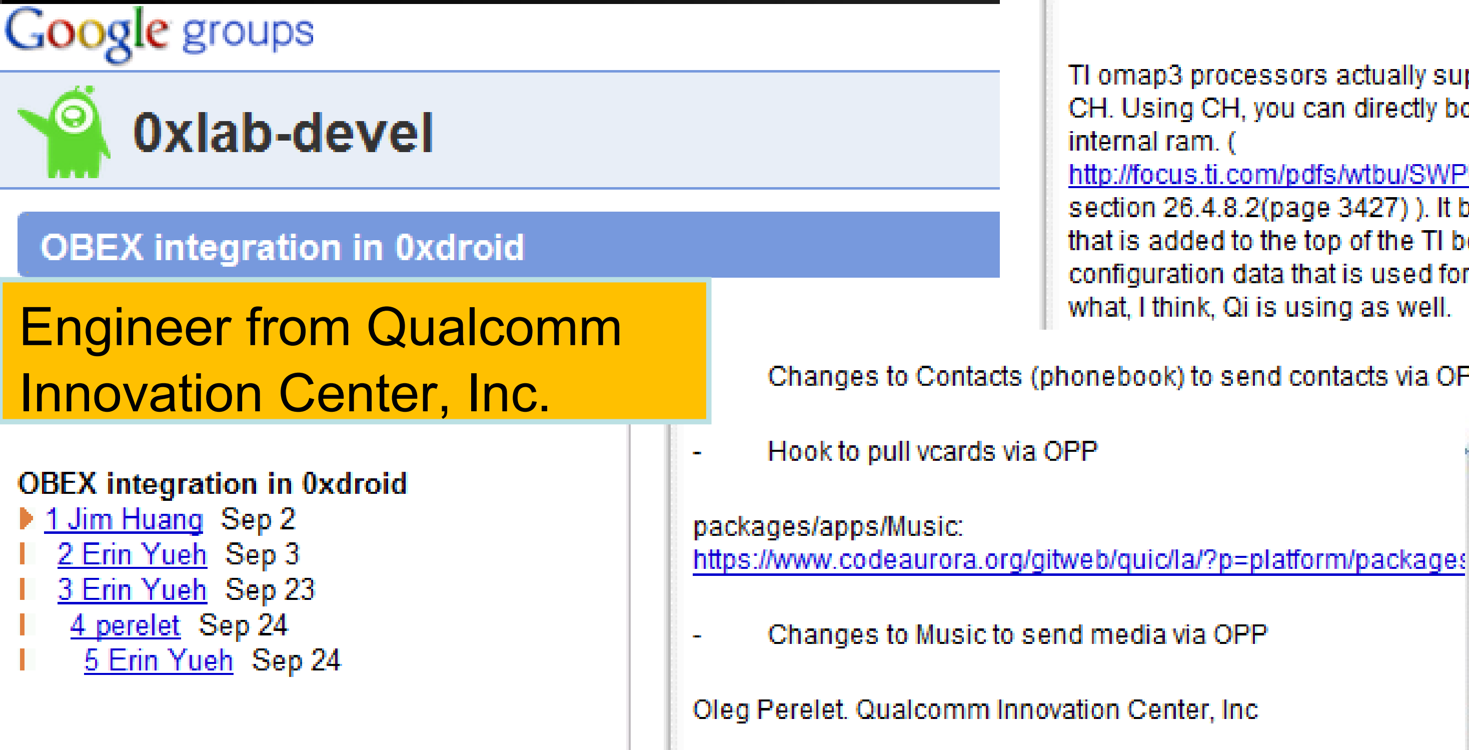
Jim Huang 2009/9/23 Matt Hsu <m...@0xlab.org>: > Like the subject, beagleboard is

Jim Huang 2009/9/23 Matt Hsu <m...@0xlab.org>: > Like the subject, beagleboard is

Abhinayak Mishra View profile

TI omap3 processors actually support the usage of configuration header or CH. Using CH, you can directly boot to SDRAM instead of going through the internal ram. (http://focus.ti.com/pdfs/wtbu/SWPU114Q_PrelimFinal_EPDF_03_05_2009.pdf, section 26.4.8.2(page 3427)). It basically is a small block of binary data that is added to the top of the TI boot image and is actually just basic configuration data that is used for setting up the external ram, which is what, I think, Qi is using as well.

Engineer from Motorola Mobile Device



Google groups

0xlab-devel

OBEX integration in 0xdroid

Engineer from Qualcomm Innovation Center, Inc.

OBEX integration in 0xdroid

1 Jim Huang	Sep 2
2 Erin Yueh	Sep 3
3 Erin Yueh	Sep 23
4 perelet	Sep 24
5 Erin Yueh	Sep 24

Changes to Contacts (phonebook) to send contacts via OPP

- Hook to pull vcards via OPP

packages/apps/Music:
https://www.codeaurora.org/gitweb/quic/la/?p=platform/packages

- Changes to Music to send media via OPP

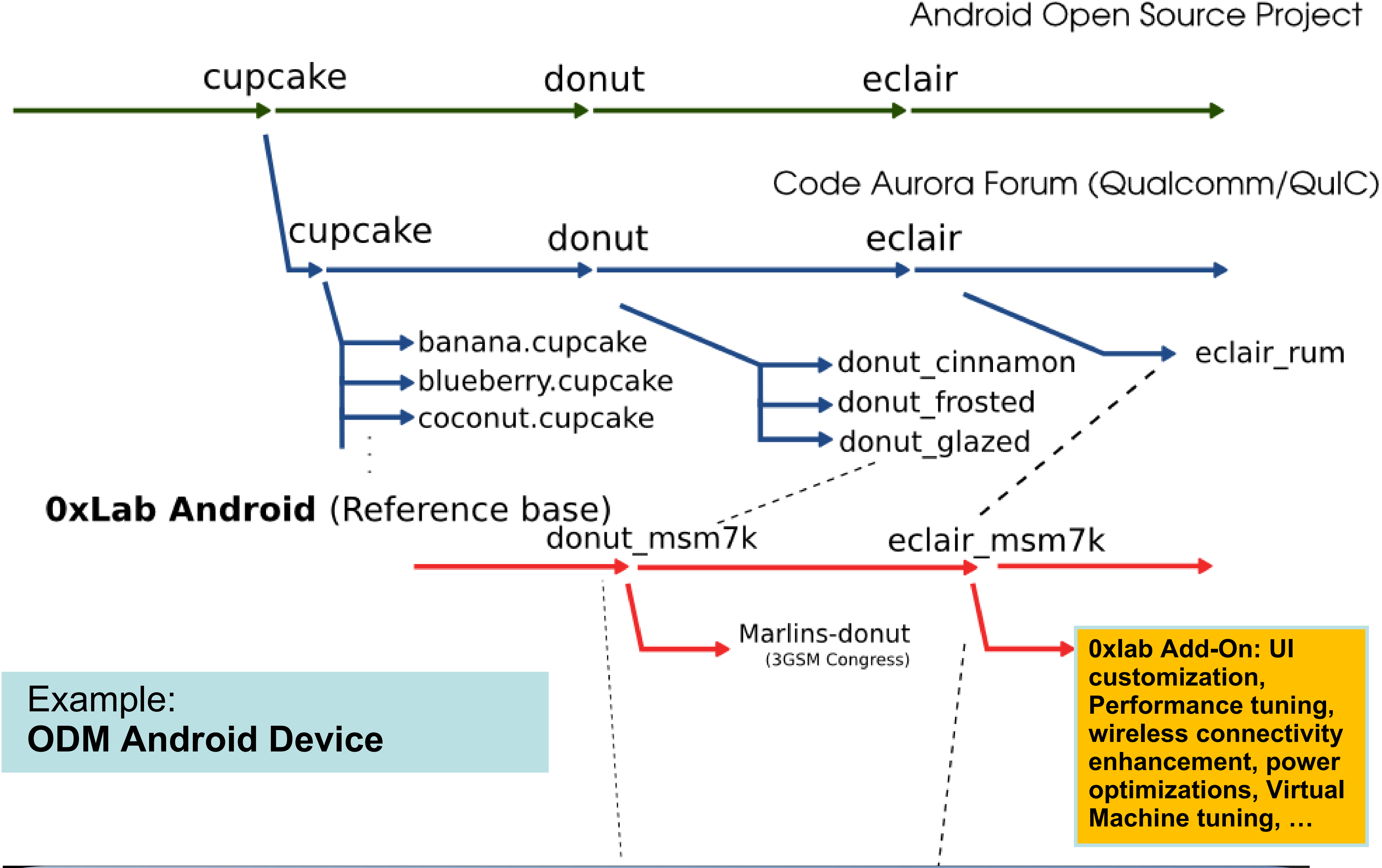
Oleg Perelet. Qualcomm Innovation Center, Inc

Craig Newell 檢視個人資料 翻譯為中文 (繁體)

寄件人： Craig Newell <cra...@vmware.com>
日期： Tue, 17 Nov 2009 06:17:35 -0800 (PST)
當地時間： 2009年11月17日(星期二) 下午10時17分
主旨： Re: [PATCH] Enable Android TLS on ARMv7 tar

回覆作者 | 轉寄 | 列印 | 個別訊息 | 顯示原始檔 | 刪除 | 回報此訊

Engineer from VMWare



Example:
ODM Android Device

0xlab Add-On: UI customization, Performance tuning, wireless connectivity enhancement, power optimizations, Virtual Machine tuning, …

與 0xlab 建立商業合作關係也是可行的

作爲一個開放原始碼專案與商業合作夥伴，我們在意整體的品質、標準支援度，及軟體客制化能力

- Device Enablement
- Platform Customizations and Verifications
- Visual Differential

Technical Impacts

- (Software) Graphics performance in Eclair is much slower than Donut. (measured 15%~43% drop)
  - Even worse, most pieces of Android frameworks expect good 3D/OpenGL|ES hardware. Google engineers don’t care about software implementation.
- Compatibility
- Quality of Android Frameworks & HAL
- New Android Launcher (Launcher2) only work under resolution 800x480.
  - Users won’t find something if the target resolution is smaller.

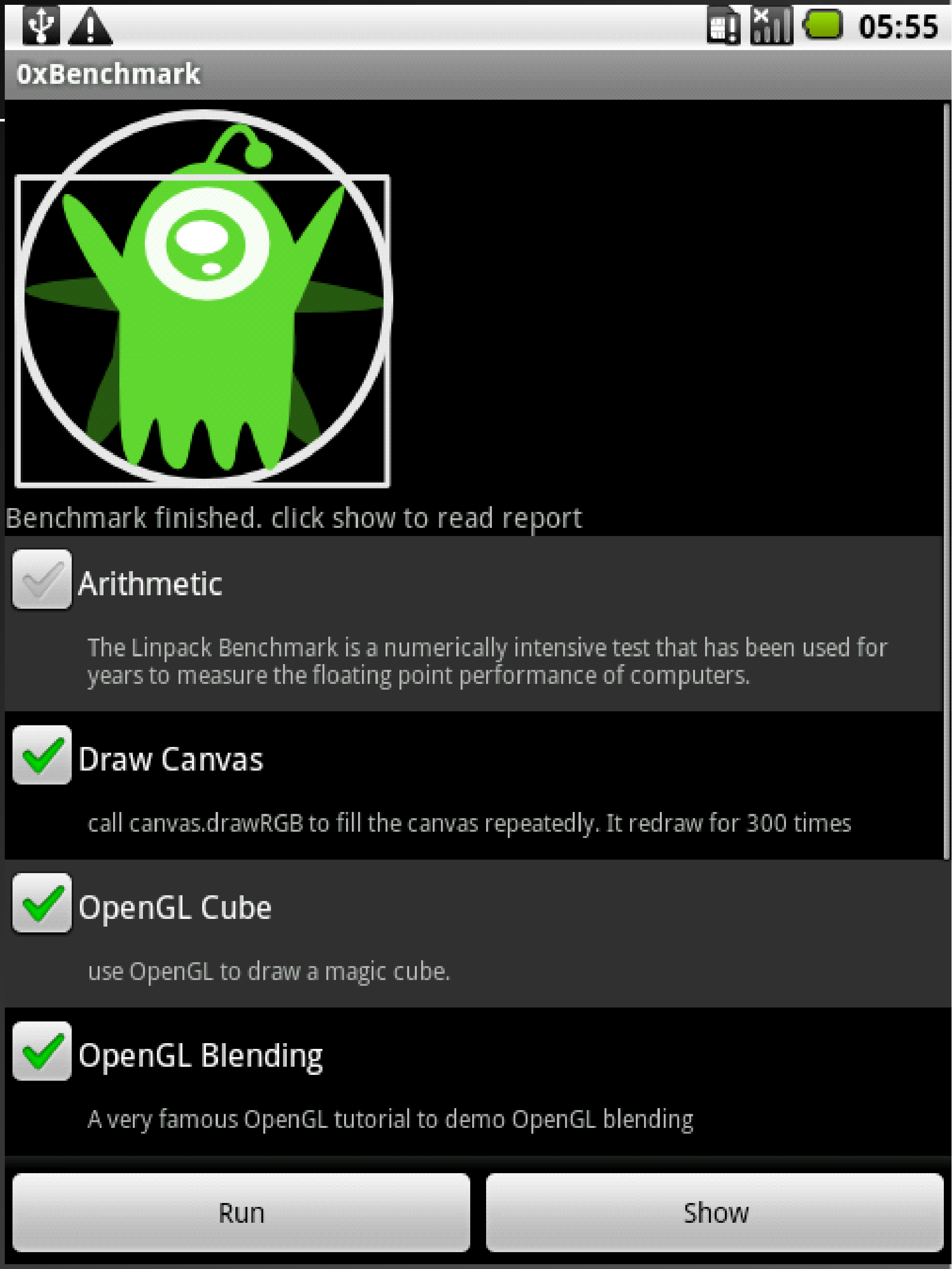
0xlab’s Approaches

- Profile the whole Android and perform aggressive optimizations dedicated to SoC
  - Eliminate the overhead between Java framework and native libraries
  - Implement ARMv6/ARMv7 optimized routines, SoC specific accelerations, Android Eclair framework tweaks
  - Avoid starvation of system resource
- Introduced Automated Testing Framework
  - Integrated Android CTS
  - Comprehensive benchmark suite
- Launcher/UI customizations

Comprehensive Benchmarking

- 0xlab develops a set of system utilities for Android to perform comprehensive system benchmarking
  - Dalvik VM performance
  - OpenGL|ES performance
  - Android Graphics framework performance
  - I/O performance
  - Connectivity performance
  - Micro-benchmark: stanard C library, system call, latency, Java invocation, ...

Consequently, 0xlab can control the system software quality in the comprehensive ways.



0xBenchmark

Benchmark finished. click show to read report

☐ Arithmetic

The Linpack Benchmark is a numerically intensive test that has been used for years to measure the floating point performance of computers.

☑ Draw Canvas

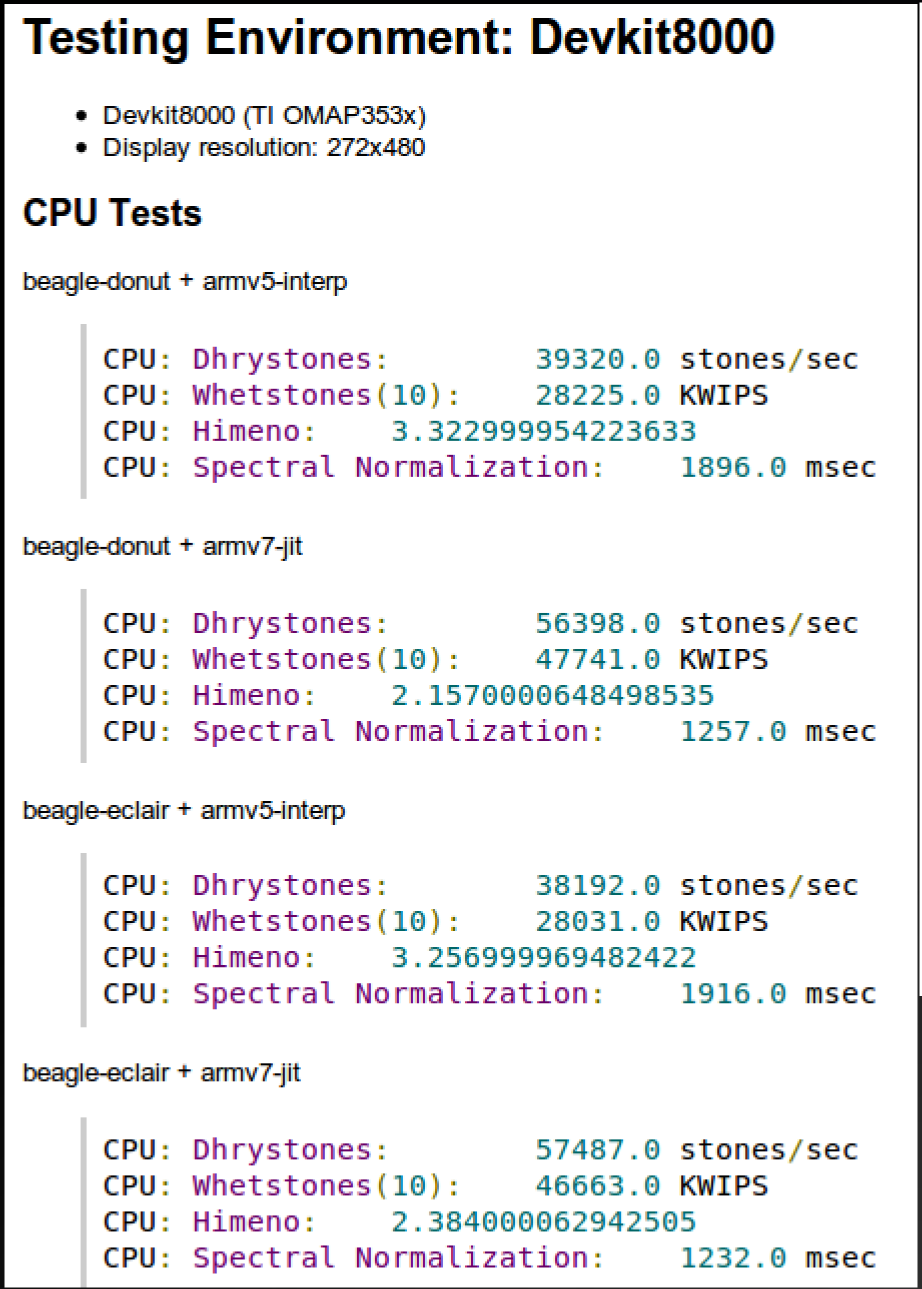
call canvas.drawRGB to fill the canvas repeatedly. It redraw for 300 times

☑ OpenGL Cube

use OpenGL to draw a magic cube.

☑ OpenGL Blending

A very famous OpenGL tutorial to demo OpenGL blending



Testing Environment: Devkit8000

- Devkit8000 (TI OMAP353x)
- Display resolution: 272x480

CPU Tests

beagle-donut + armv5-interp

```
CPU: Dhrystones:         39320.0 stones/sec
CPU: Whetstones(10):     28225.0 KWIPS
CPU: Himeno:     3.322999954223633
CPU: Spectral Normalization:     1896.0 msec
```

beagle-donut + armv7-jit

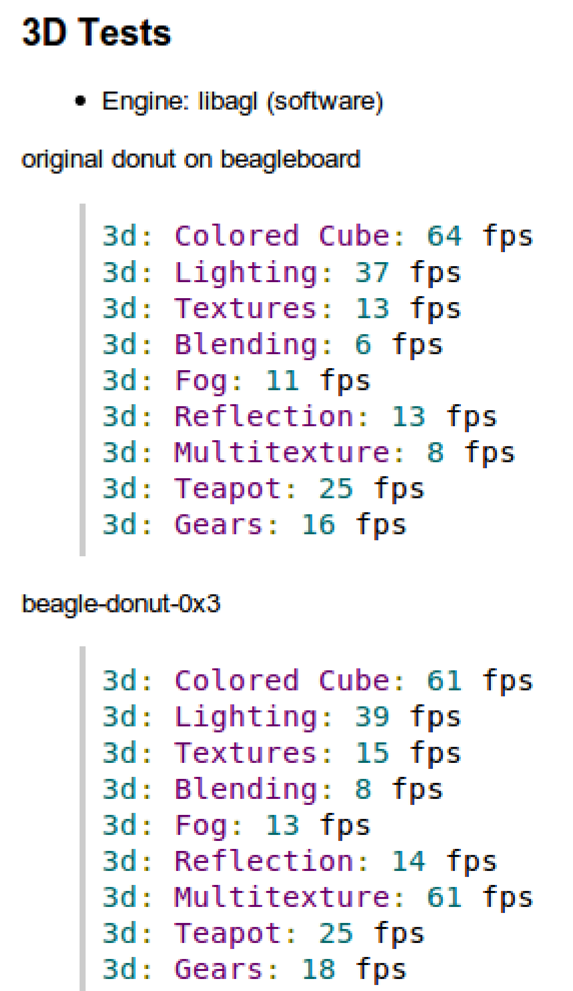
```
CPU: Dhrystones:         56398.0 stones/sec
CPU: Whetstones(10):     47741.0 KWIPS
CPU: Himeno:     2.1570000648498535
CPU: Spectral Normalization:     1257.0 msec
```

beagle-eclair + armv5-interp

```
CPU: Dhrystones:         38192.0 stones/sec
CPU: Whetstones(10):     28031.0 KWIPS
CPU: Himeno:     3.256999969482422
CPU: Spectral Normalization:     1916.0 msec
```

beagle-eclair + armv7-jit

```
CPU: Dhrystones:         57487.0 stones/sec
CPU: Whetstones(10):     46663.0 KWIPS
CPU: Himeno:     2.384000062942505
CPU: Spectral Normalization:     1232.0 msec
```



3D Tests

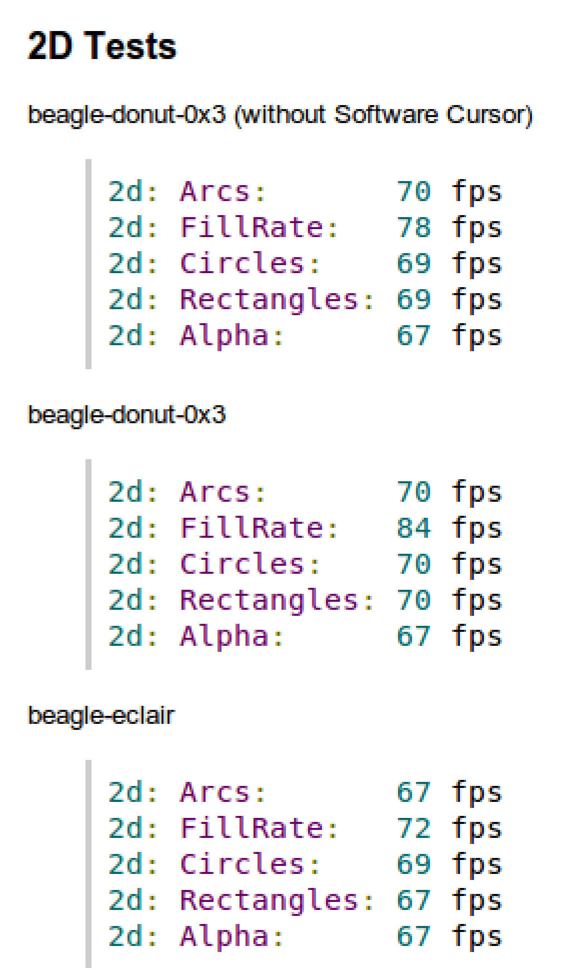
- Engine: libagl (software)

original donut on beagleboard

```
3d: Colored Cube: 64 fps
3d: Lighting: 37 fps
3d: Textures: 13 fps
3d: Blending: 6 fps
3d: Fog: 11 fps
3d: Reflection: 13 fps
3d: Multitexture: 8 fps
3d: Teapot: 25 fps
3d: Gears: 16 fps
```

beagle-donut-0x3

```
3d: Colored Cube: 61 fps
3d: Lighting: 39 fps
3d: Textures: 15 fps
3d: Blending: 8 fps
3d: Fog: 13 fps
3d: Reflection: 14 fps
3d: Multitexture: 61 fps
3d: Teapot: 25 fps
3d: Gears: 18 fps
```



2D Tests

beagle-donut-0x3 (without Software Cursor)

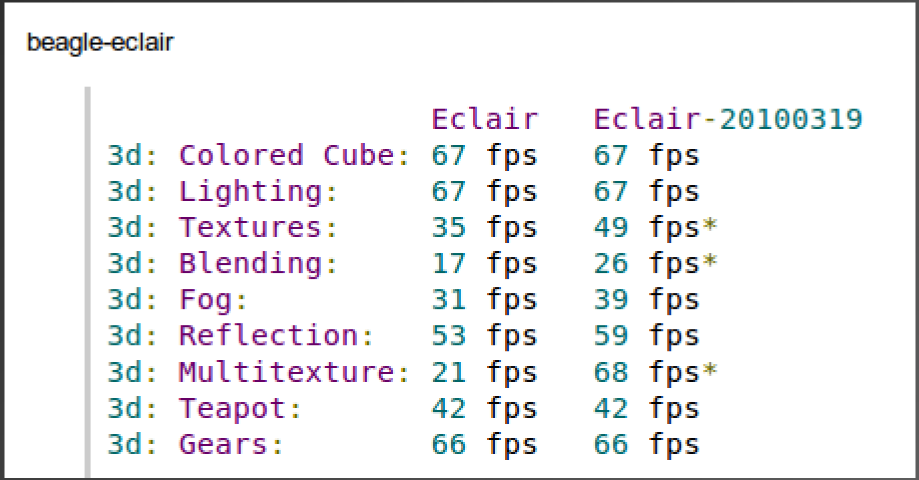
```
2d: Arcs:        70 fps
2d: FillRate:    78 fps
2d: Circles:     69 fps
2d: Rectangles:  69 fps
2d: Alpha:       67 fps
```

beagle-donut-0x3

```
2d: Arcs:        70 fps
2d: FillRate:    84 fps
2d: Circles:     70 fps
2d: Rectangles:  70 fps
2d: Alpha:       67 fps
```

beagle-eclair

```
2d: Arcs:        67 fps
2d: FillRate:    72 fps
2d: Circles:     69 fps
2d: Rectangles:  67 fps
2d: Alpha:       67 fps
```



beagle-eclair

```
                  Eclair    Eclair-20100319
3d: Colored Cube: 67 fps    67 fps
3d: Lighting:     67 fps    67 fps
3d: Textures:     35 fps    49 fps*
3d: Blending:     17 fps    26 fps*
3d: Fog:          31 fps    39 fps
3d: Reflection:   53 fps    59 fps
3d: Multitexture: 21 fps    68 fps*
3d: Teapot:       42 fps    42 fps
3d: Gears:        66 fps    66 fps
```

大膽假設，小心求證，社群驗證

SoC specific enablement (minimal efforts)

libopencorehw.so (OpenCore HW module)

http://gitorious.org/0xdroid/hardware_omap3_libopencorehw

liboverlay.so (Graphics overlays module)

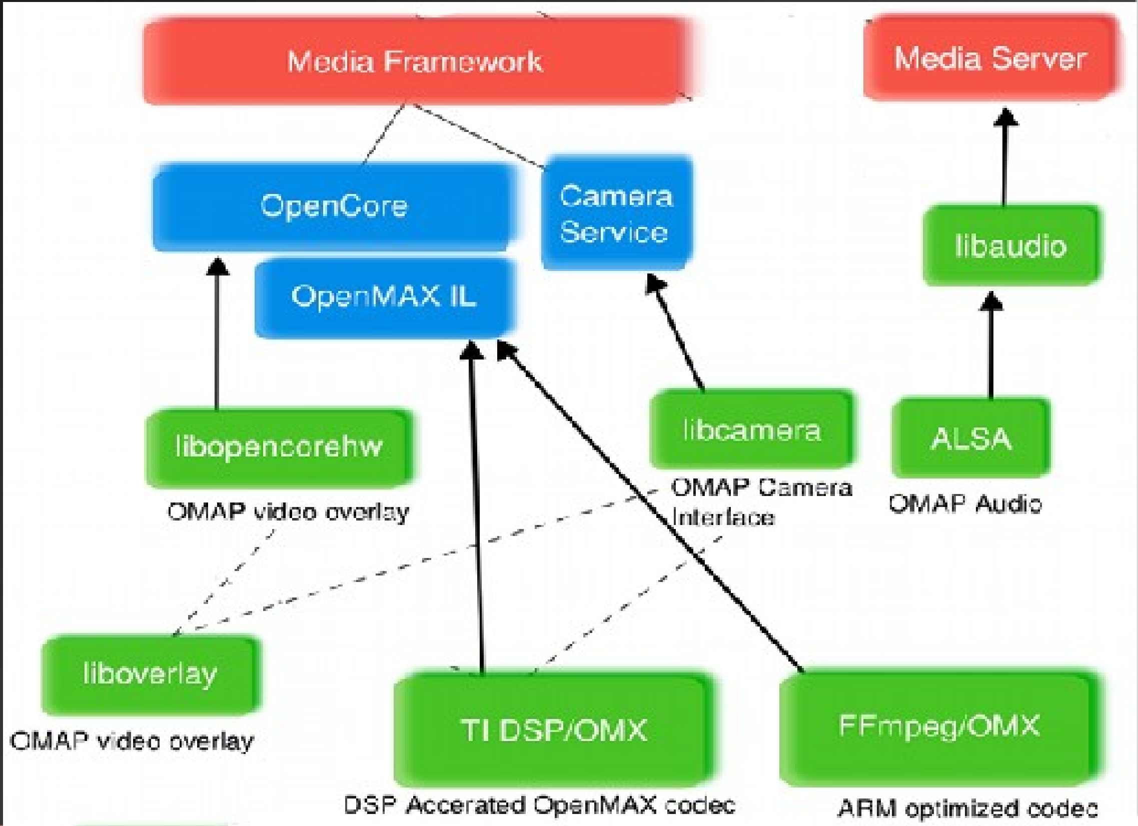
libcamera.so (Camera HAL)

http://gitorious.org/0xdroid/hardware_omap3_camera

libaudio.so (Audio HAL)

http://gitorious.org/0xdroid/hardware_alsa_sound

0xdroid provides the full source code of reference hardware acceleration modules for Android.



Case Study:

Performance Evaluation on Beagleboard

TI OMAP3 SoC powered

500 MHz / ARM Cortex A8

0xdroid – well-tuned Android for Beagleboard (TI OMAP 3530)

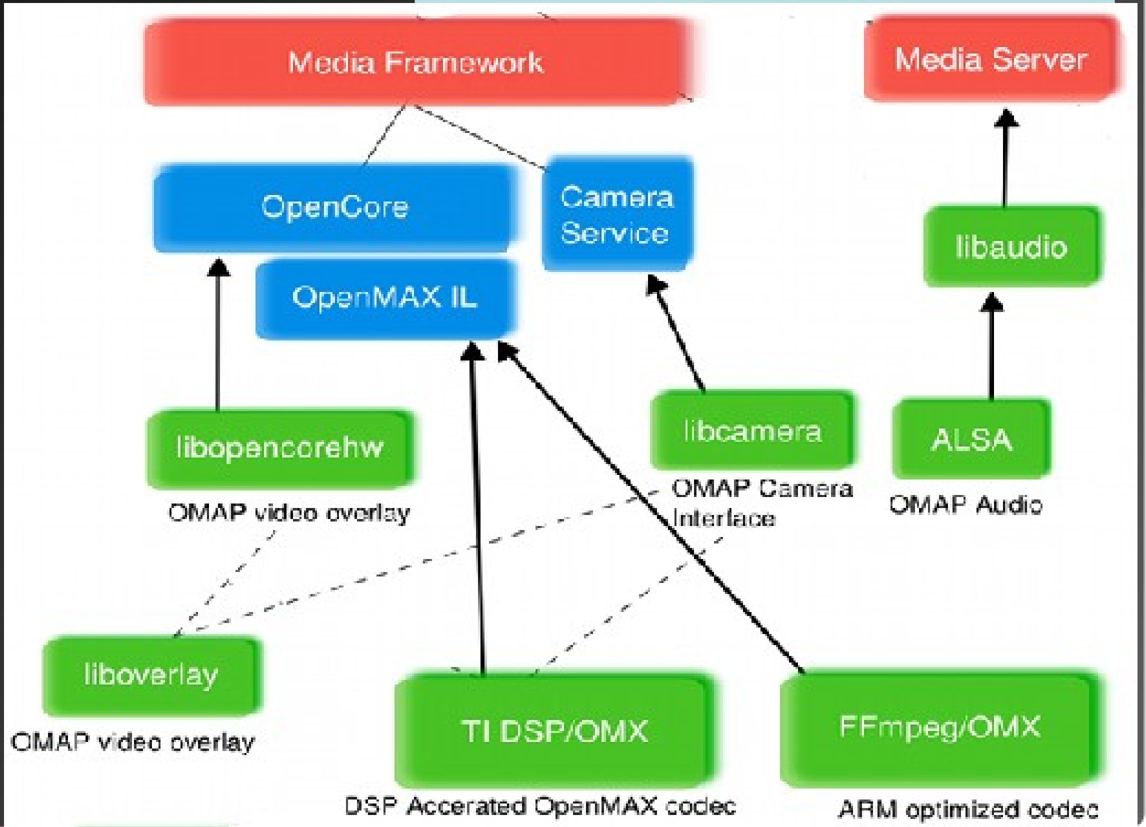
http://code.google.com/p/0xdroid/

Based on Android Eclair branch

beagle-eclair-0x4 (Apr 25, 2010)

0xdroid provides the full source code of reference hardware acceleration modules for Android.

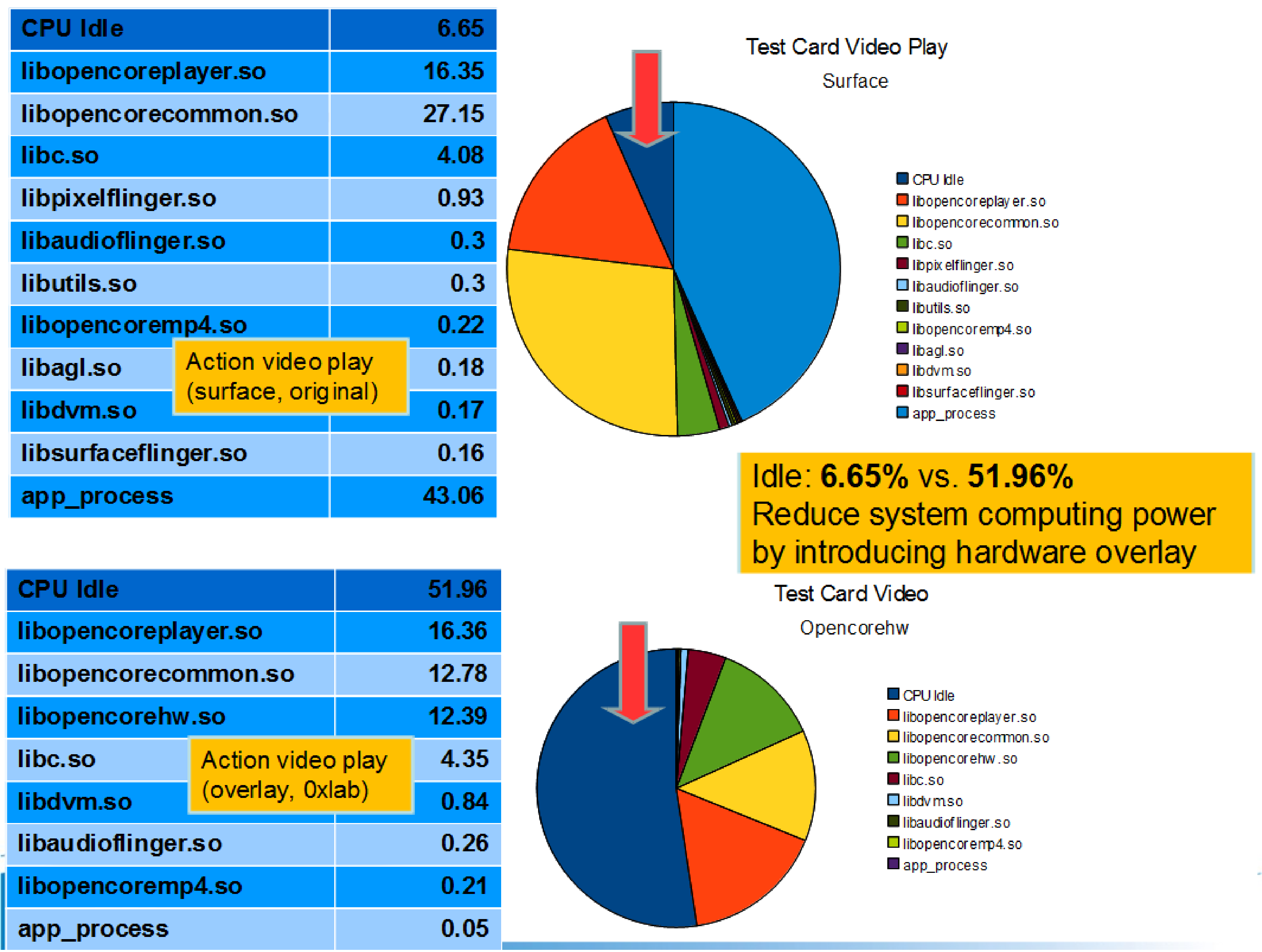
- **libopencorehw.so** (OpenCore HW module)
  http://gitorious.org/0xdroid/hardware_omap3_libopencorehw
- **liboverlay.so** (Graphics overlays module)
- **libcamera.so** (Camera HAL)
  http://gitorious.org/0xdroid/hardware_omap3_camera
- **libaudio.so** (Audio HAL)
  http://gitorious.org/0xdroid/hardware_alsa_sound



Evalutions scenario: Introduced libopencorehw.so

(measured by utility “oprofile”)

Video playback :: Test Card Video (480x360, 25fps, H.264)



CPU Idle	6.65
libopencoreplayer.so	16.35
libopencorecommon.so	27.15
libc.so	4.08
libpixelflinger.so	0.93
libaudioflinger.so	0.3
libutils.so	0.3
libopencoremp4.so	0.22
libagl.so	0.18
libdvm.so	0.17
libsurfaceflinger.so	0.16
app_process	43.06

Action video play
(surface, original)

Idle: **6.65%** vs. **51.96%**
Reduce system computing power by introducing hardware overlay

CPU Idle	51.96
libopencoreplayer.so	16.36
libopencorecommon.so	12.78
libopencorehw.so	12.39
libc.so	4.35
libdvm.so	0.84
libaudioflinger.so	0.26
libopencoremp4.so	0.21
app_process	0.05

Action video play
(overlay, 0xlab)

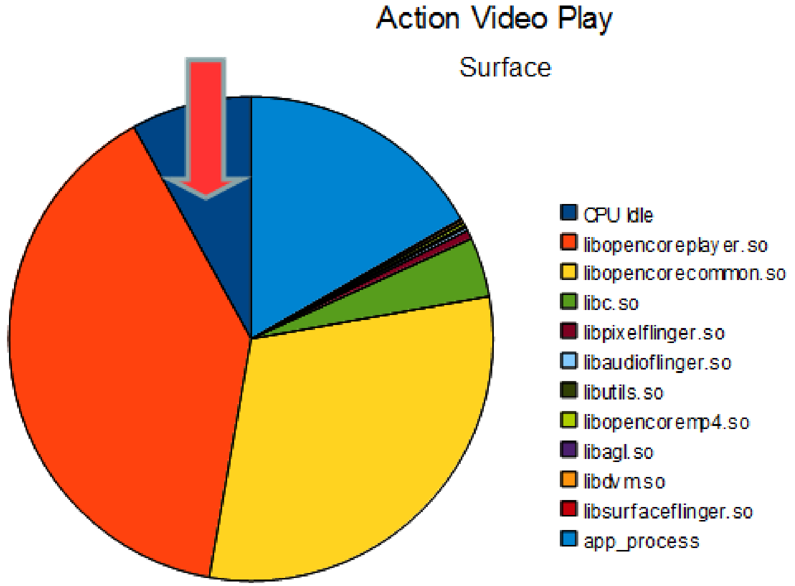
Evalutions scenario: Introduced libopencorehw.so

(measured by utility “oprofile”)

Video playback :: Action Video (480x360, 25fps, H.264)

CPU Idle	8.03
libopencoreplayer.so	39.05
libopencorecommon.so	30.47
libc.so	3.97
libpixelflinger.so	0.57
libaudioflinger.so	0.23
libutils.so	0.14
libopencoremp4.so	0.23
libagl.so	0.04
libdvm.so	0.19
libsurfaceflinger.so	0.06
app_process	16.68

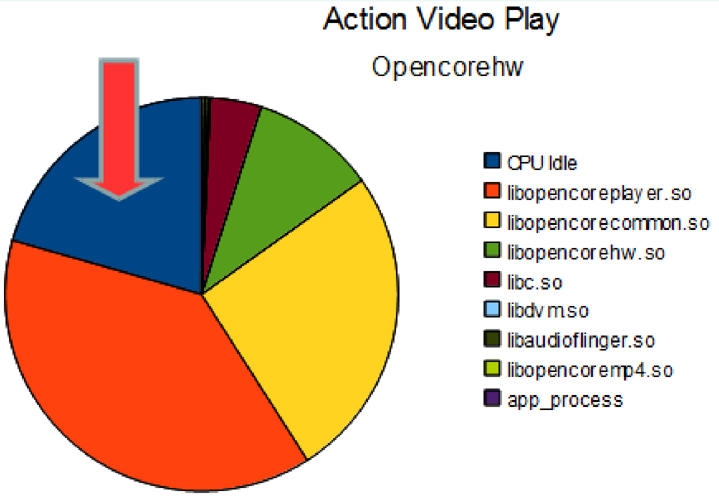
Action video play
(surface, original)



Idle: **8.03%** vs. **20.49%**
Even codec is quite busy, system computing power benefits from hardware overlays.

CPU Idle	20.49
libopencoreplayer.so	38.32
libopencorecommon.so	25.69
libopencorehw.so	10.2
libc.so	4.19
libdvm.so	0.17
libaudioflinger.so	0.28
libopencoremp4.so	0.23
app_process	0.03

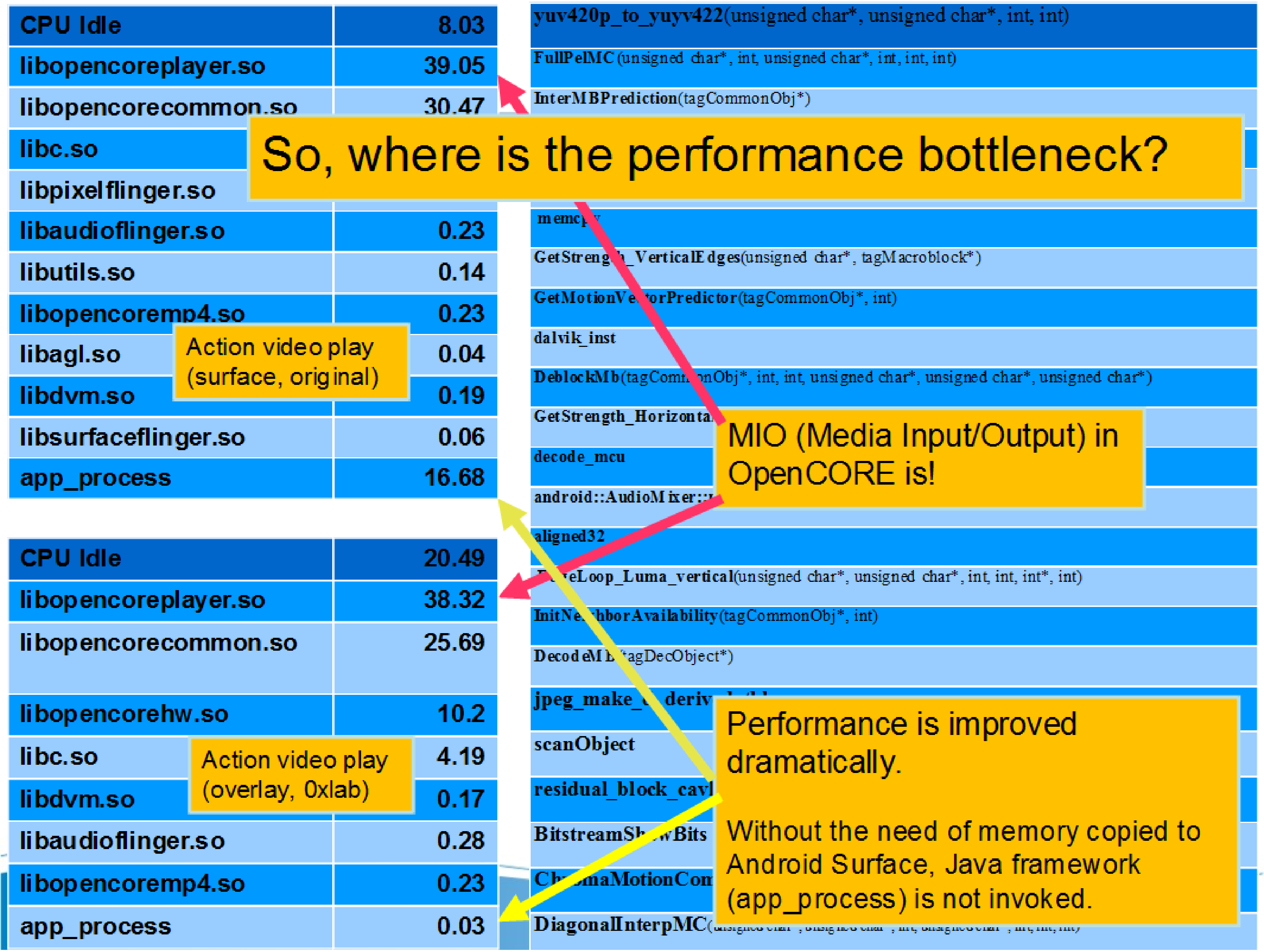
Action video play
(overlay, 0xlab)



Evalutions scenario: Introduced libopencorehw.so

(measured by utility "oprofile")

Video playback :: Action Video (480x360, 25fps, H.264)



CPU Idle	8.03
libopencoreplayer.so	39.05
libopencorecommon.so	30.47
libc.so	
libpixelflinger.so	
libaudioflinger.so	0.23
libutils.so	0.14
libopencoremp4.so	0.23
libagl.so	0.04
libdvm.so	0.19
libsurfaceflinger.so	0.06
app_process	16.68

Action video play (surface, original)

CPU Idle	20.49
libopencoreplayer.so	38.32
libopencorecommon.so	25.69
libopencorehw.so	10.2
libc.so	4.19
libdvm.so	0.17
libaudioflinger.so	0.28
libopencoremp4.so	0.23
app_process	0.03

Action video play (overlay, 0xlab)

yuv420p_to_yuyv422(unsigned char*, unsigned char*, int, int)

FullPelMC (unsigned char*, int, unsigned char*, int, int, int)

InterMBPrediction(tagCommonObj*)

memcp

GetStreng _VerticalEdges(unsigned char*, tagMacroblock*)

GetMotionVe torPredictor(tagCommonObj*, int)

dalvik_inst

DeblockMb(tagCom onObj*, int, int, unsigned char*, unsigned char*, unsigned char*)

GetStrength_Horizonta

decode_mcu

android::AudioMixer::

aligned32

eLoop_Luma_vertical(unsigned char*, unsigned char*, int, int, int*, int)

InitNe hborAvailability(tagCommonObj*, int)

DecodeM agDecObject*)

jpeg_make_ deriv

scanObject

residual_block_cavl

BitstreamSh wBits

Ch maMotionCom

DiagonalInterpMC(

So, where is the performance bottleneck?

MIO (Media Input/Output) in OpenCORE is!

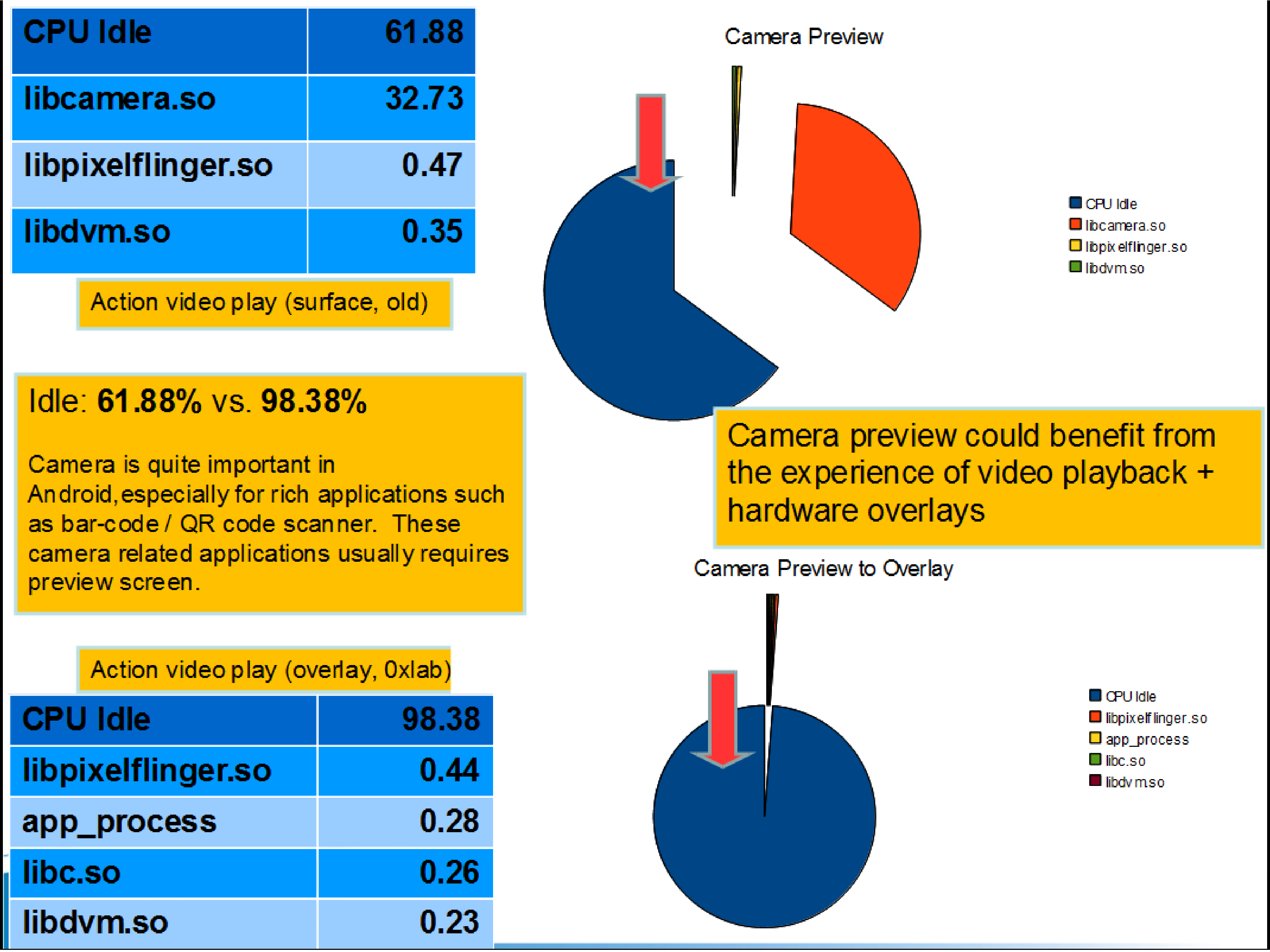
Performance is improved dramatically.

Without the need of memory copied to Android Surface, Java framework (app_process) is not invoked.

Evalutions scenario: Introduced libcamera.so

(measured by utility “oprofile”)

Camera preview (320x480)



CPU Idle	61.88
libcamera.so	32.73
libpixelflinger.so	0.47
libdvm.so	0.35

Action video play (surface, old)

Idle: **61.88%** vs. **98.38%**

Camera is quite important in Android, especially for rich applications such as bar-code / QR code scanner. These camera related applications usually requires preview screen.

Action video play (overlay, 0xlab)

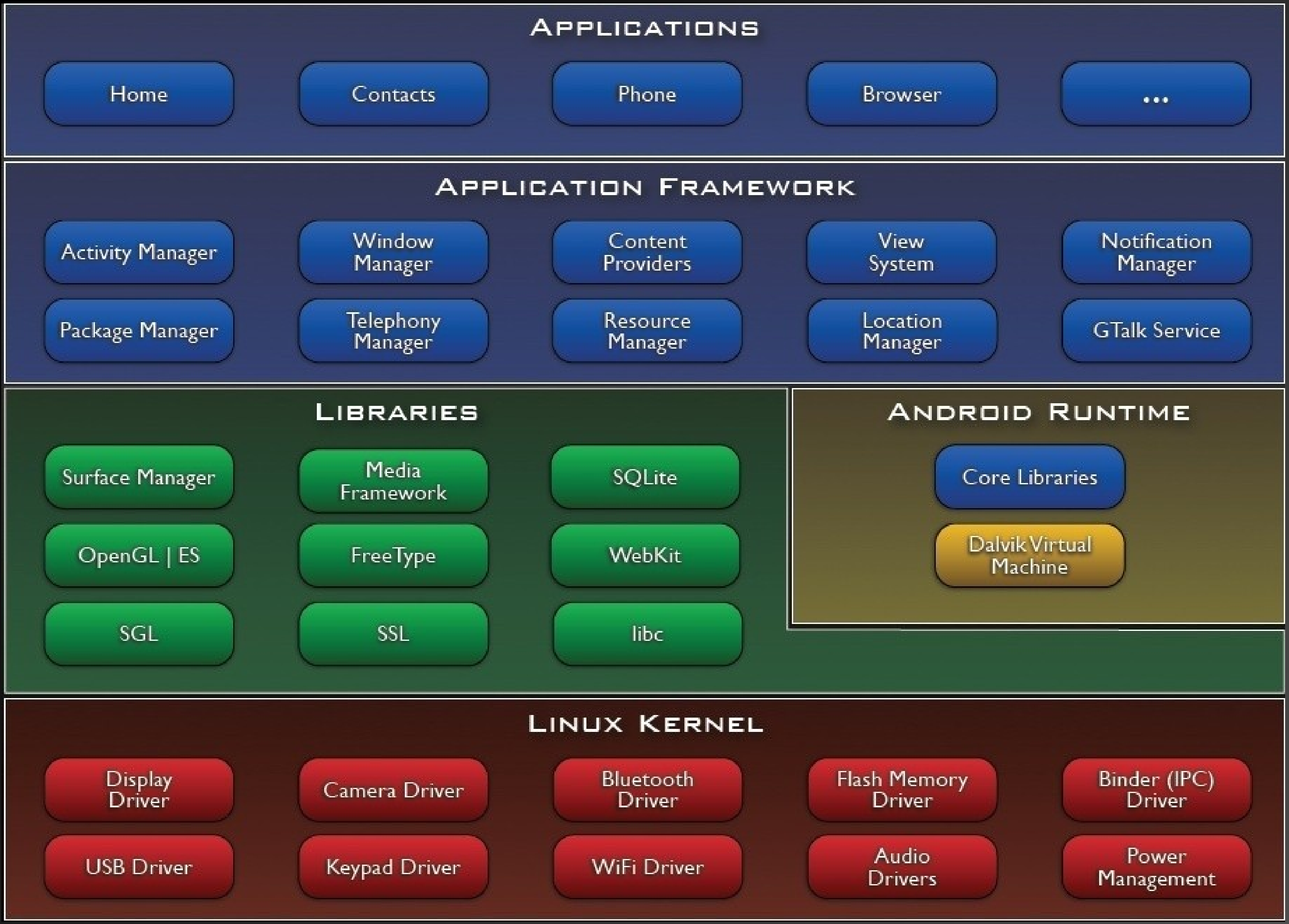
CPU Idle	98.38
libpixelflinger.so	0.44
app_process	0.28
libc.so	0.26
libdvm.so	0.23

Camera Preview

Camera preview could benefit from the experience of video playback + hardware overlays

Camera Preview to Overlay

Classical Android Architecture



APPLICATIONS

Home

Contacts

Phone

Browser

...

APPLICATION FRAMEWORK

Activity Manager

Window Manager

Content Providers

View System

Notification Manager

Package Manager

Telephony Manager

Resource Manager

Location Manager

GTalk Service

LIBRARIES

Surface Manager

Media Framework

SQLite

OpenGL | ES

FreeType

WebKit

SGL

SSL

libc

ANDROID RUNTIME

Core Libraries

Dalvik Virtual Machine

LINUX KERNEL

Display Driver

Camera Driver

Bluetooth Driver

Flash Memory Driver

Binder (IPC) Driver

USB Driver

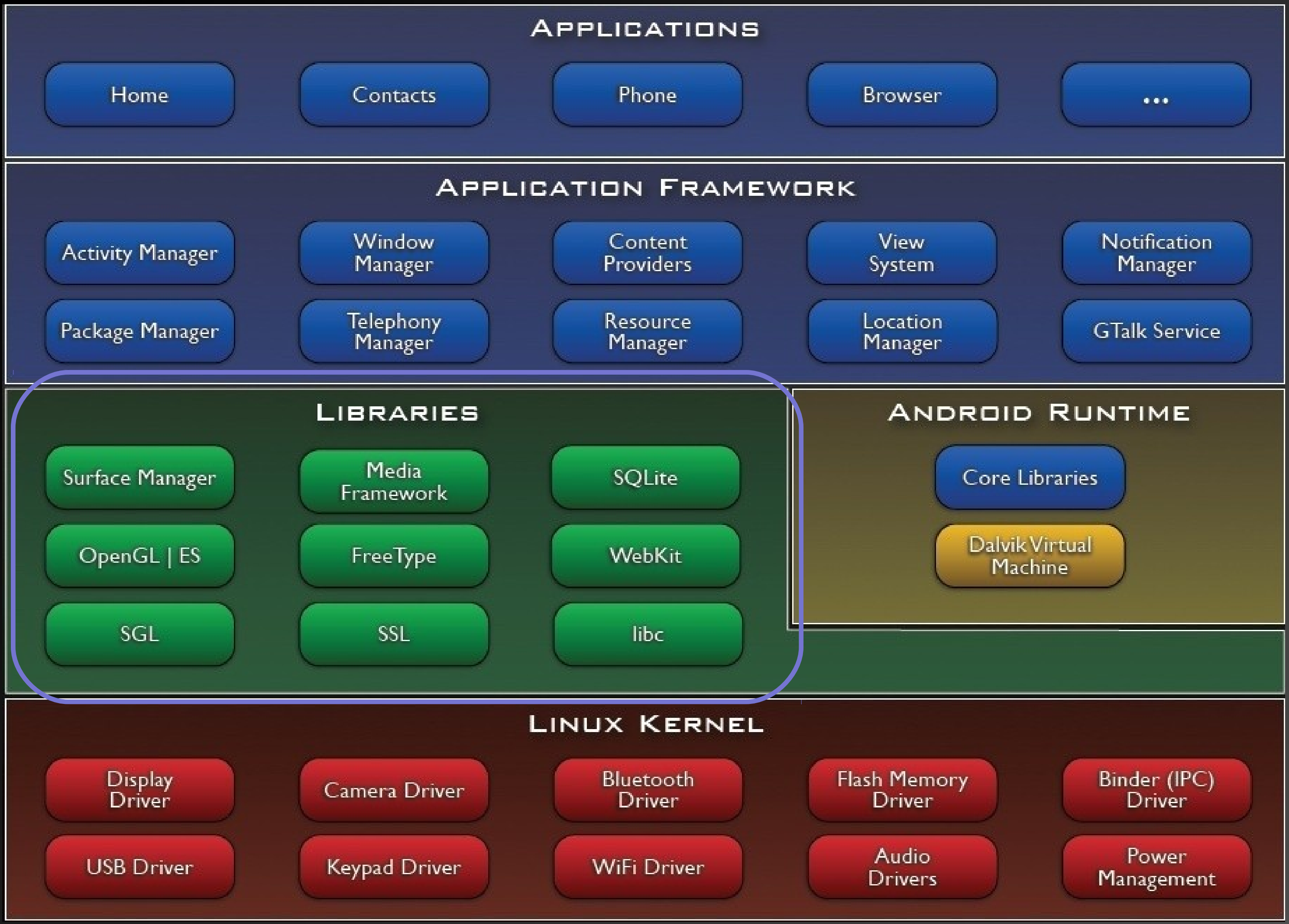
Keypad Driver

WiFi Driver

Audio Drivers

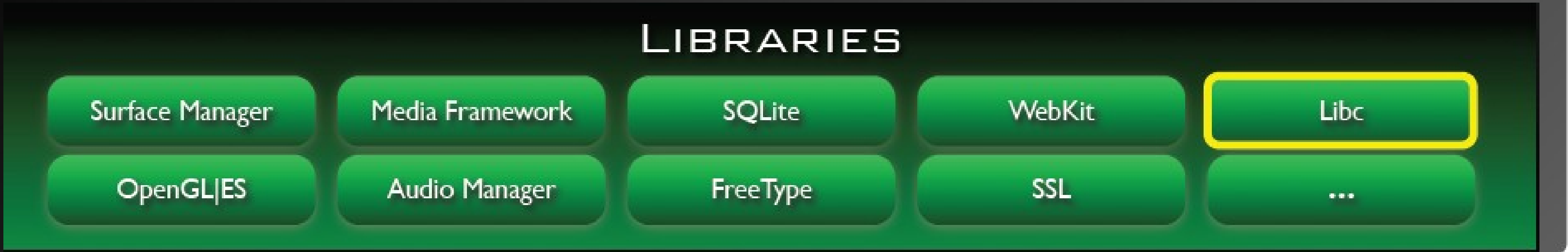
Power Management

Native Libraries



Bionic Libc

- Android C/C++ library
- 0xlab’s Optimizations
  - Memory operations: Use ARMv6 unaligned access to optimize usual cases
  - Atomic operations: Use ARMv6 ldrex/strex
  - Endian/Data Type conversion: Use ARMv6 fast endian primitives. Useful for TCP/IP (big endian ←→ little endian coverting)
  - ARMv7 SIMD/NEON optimization
  - Introduced ARM Thumb2 instructions to optimize string operations



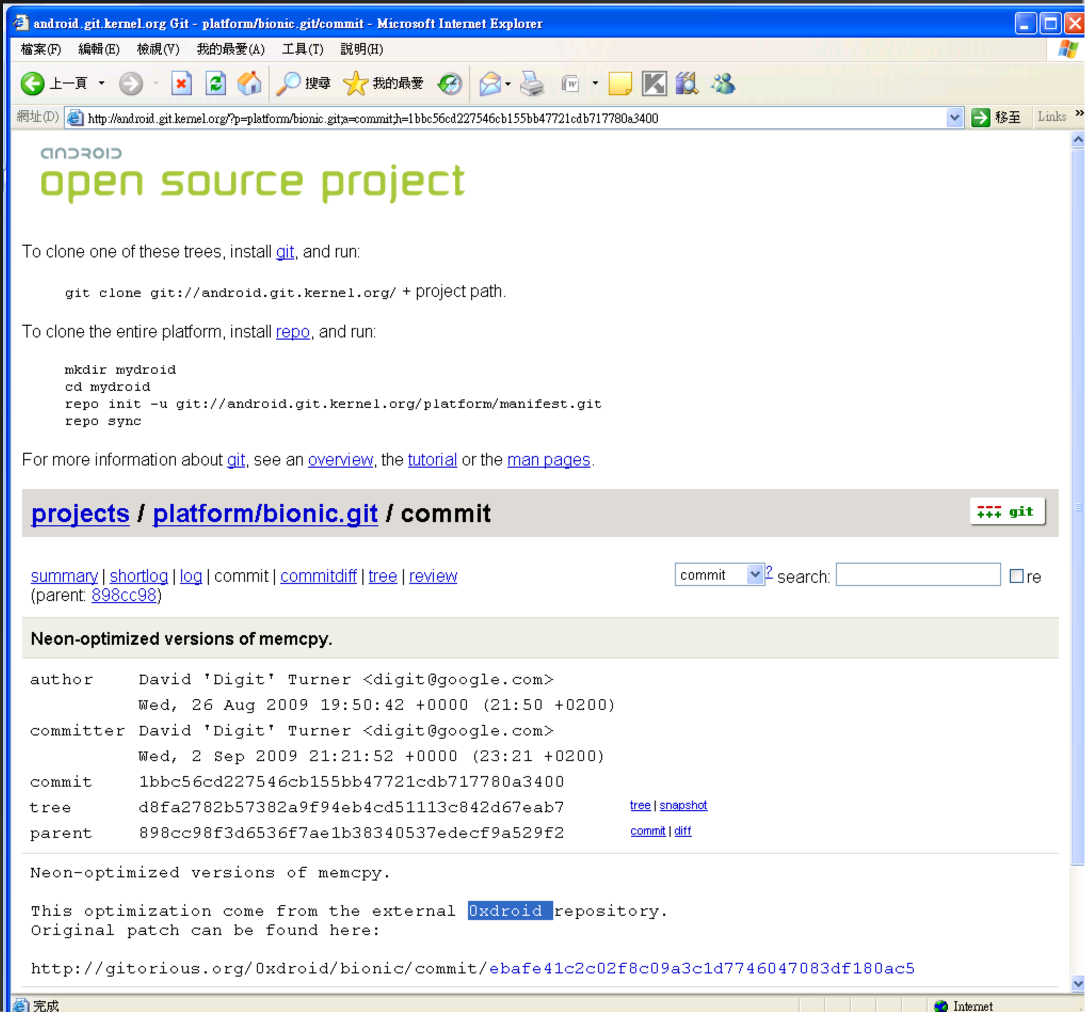
Memory Optimization to Utilize Advanced ARM Features

```
[[ very small data test ]]                  avg/peak
memcpy_neon :(24 bytes copy) = 230.9 MB/s / 551.4 MB/s
memcpy_armv5:(24 bytes copy) = 123.3 MB/s / 252.8 MB/s
memcpy_arm  :(24 bytes copy) = 170.2 MB/s / 226.6 MB/s

memcpy_neon :(31 bytes copy) = 314.9 MB/s / 712.5 MB/s
memcpy_armv5:(31 bytes copy) = 143.2 MB/s / 326.6 MB/s
memcpy_arm  :(31 bytes copy) = 197.0 MB/s / 272.1 MB/s
```

```
[[ L1 cached data ]]                             avg/peak
memcpy_neon :(4096 bytes copy) = 2132.7 MB/s / 2192.7 MB/s
memcpy_armv5:(4096 bytes copy) =  806.8 MB/s / 1289.2 MB/s
memcpy_arm  :(4096 bytes copy) =  830.5 MB/s / 1396.0 MB/s

memcpy_neon :(6144 bytes copy) = 2176.2 MB/s / 2216.7 MB/s
memcpy_armv5:(6144 bytes copy) =  820.0 MB/s / 1300.5 MB/s
memcpy_arm  :(6144 bytes copy) = 839.8 MB/s /  1411.7 MB/s
```

- memcpy_neon : 0xlab's otimized ARM NEON version
- memcpy_armv5 : donut/cupcake ARMv5 optimized
- memcpy_arm : LGPL ARMv5 optimized

Android engineer included the memcpy() improvements into Éclair codebase!


android.git.kernel.org Git - platform/bionic.git/commit - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
網址(D)
移至
Links
android
open source project
git
完成
Internet

http://android.git.kernel.org/?p=platform/bionic.git;a=commit;h=1bbc56cd227546cb155bb47721cdb717780a3400

To clone one of these trees, install git, and run:

```
git clone git://android.git.kernel.org/ + project path.
```

To clone the entire platform, install repo, and run:

```
mkdir mydroid
cd mydroid
repo init -u git://android.git.kernel.org/platform/manifest.git
repo sync
```

For more information about git, see an overview, the tutorial or the man pages.

projects / platform/bionic.git / commit

summary | shortlog | log | commit | commitdiff | tree | review
(parent: 898cc98)

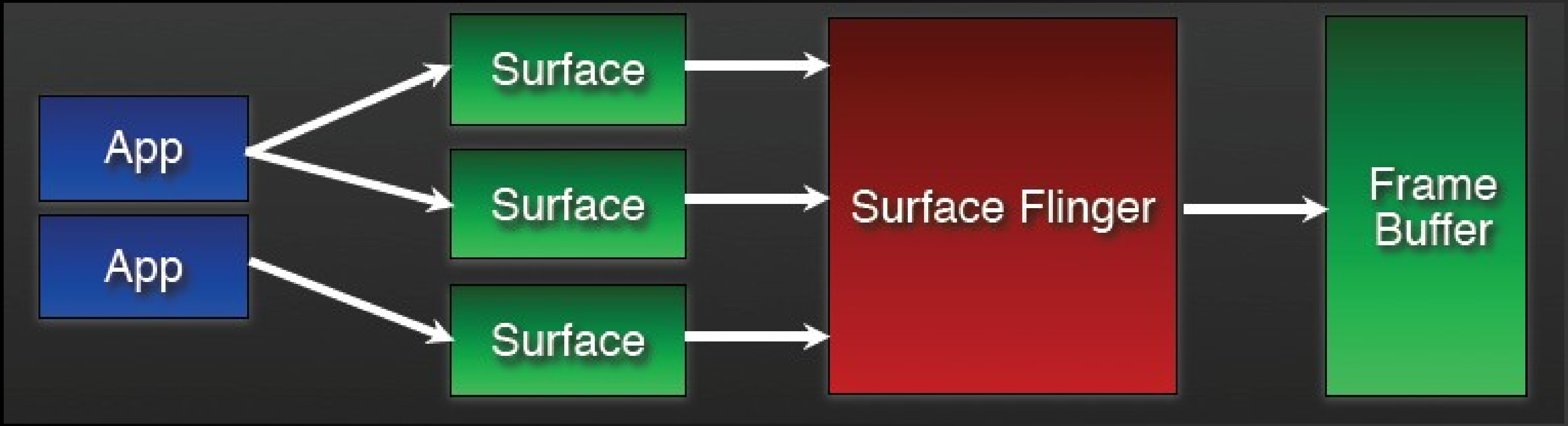
commit ? search: □re

Neon-optimized versions of memcpy.

```
author     David 'Digit' Turner <digit@google.com>
           Wed, 26 Aug 2009 19:50:42 +0000 (21:50 +0200)
committer  David 'Digit' Turner <digit@google.com>
           Wed, 2 Sep 2009 21:21:52 +0000 (23:21 +0200)
commit     1bbc56cd227546cb155bb47721cdb717780a3400
tree       d8fa2782b57382a9f94eb4cd51113c842d67eab7      tree | snapshot
parent     898cc98f3d6536f7ae1b38340537edecf9a529f2      commit | diff
```

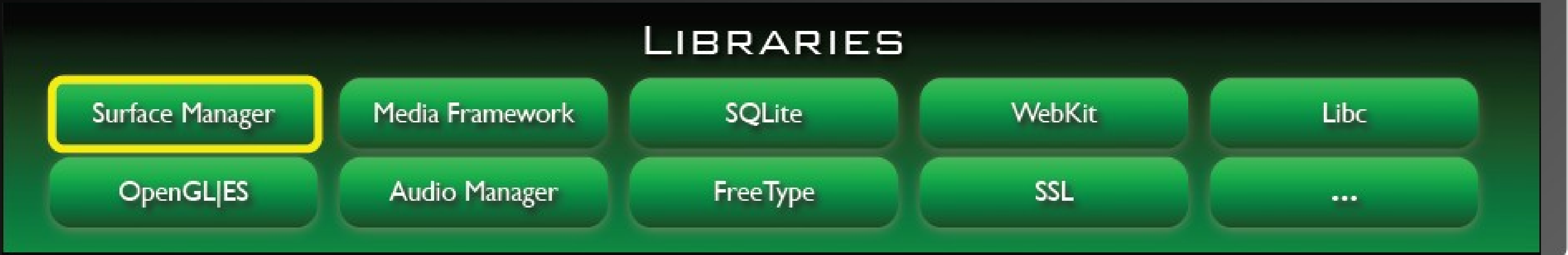
```
Neon-optimized versions of memcpy.

This optimization come from the external 0xdroid repository.
Original patch can be found here:

http://gitorious.org/0xdroid/bionic/commit/ebafe41c2c02f8c09a3c1d7746047083df180ac5
```

Native Server: Surface Flinger



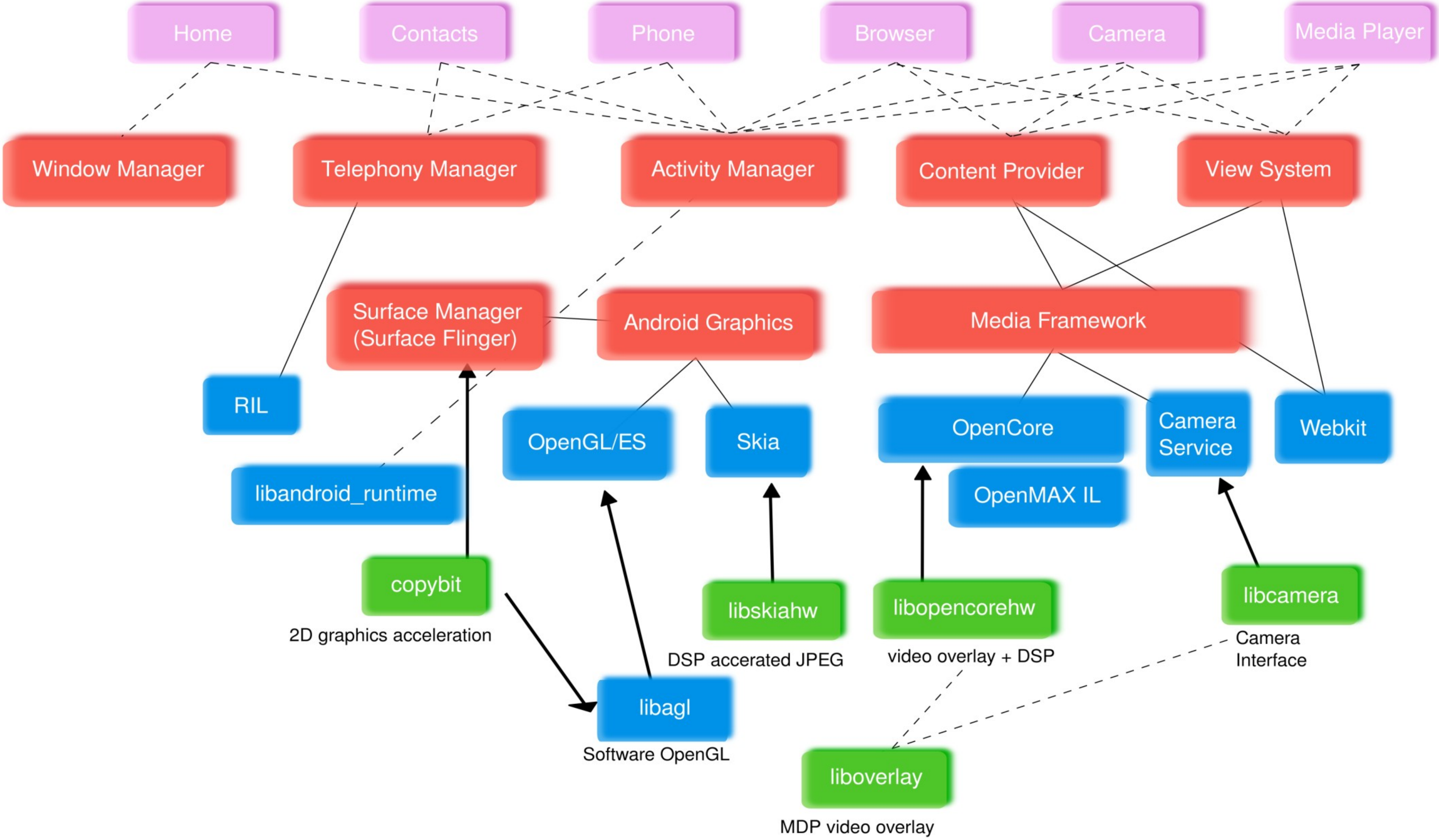
App
App
Surface
Surface
Surface
Surface Flinger
Frame Buffer

- Provides system-wide surface “composer”, handling all surface rendering to frame buffer device
- Combine 2D and 3D surfaces and surfaces from multiple applications
- 0xlab’s improvements
  - Eliminate the redundant computation for Surfaces (performance equals to Donut now)
  - Enable SoC 2D accelerations
  - Optimize PixelFlinger through ARMv6/ARMv7 optimized routines



LIBRARIES

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



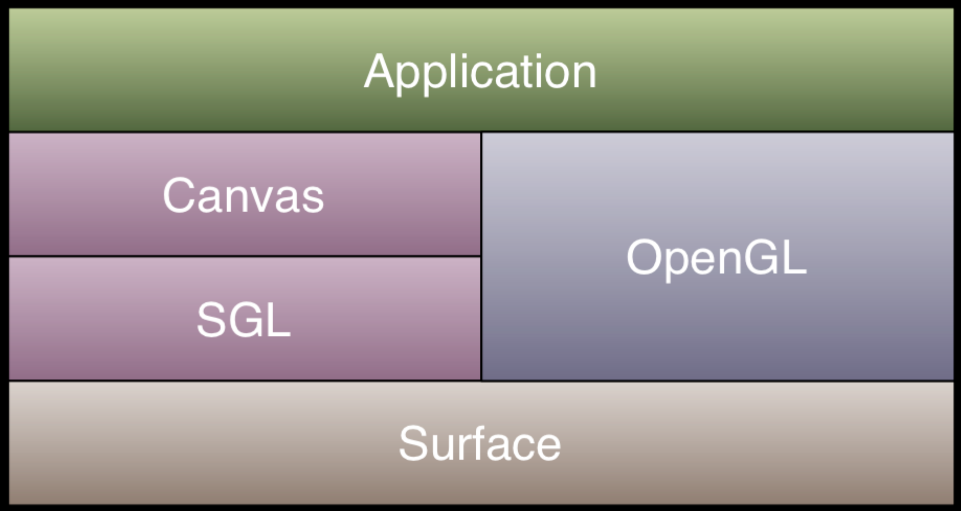
Home
Contacts
Phone
Browser
Camera
Media Player

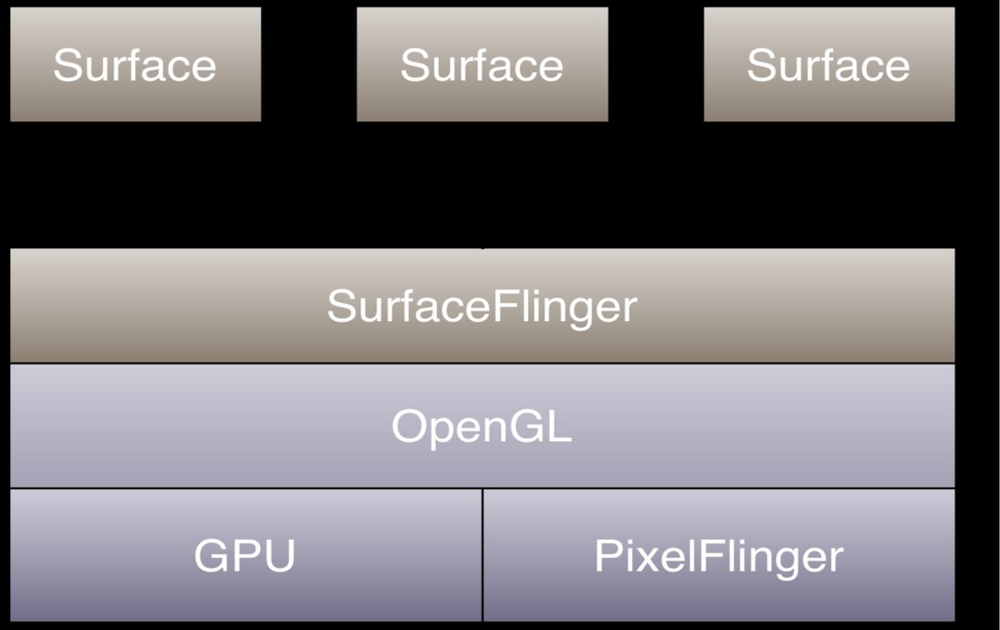
Window Manager
Telephony Manager
Activity Manager
Content Provider
View System

Surface Manager
(Surface Flinger)
Android Graphics
Media Framework

RIL
OpenGL/ES
Skia
OpenCore
Camera
Service
Webkit
libandroid_runtime
OpenMAX IL

copybit
2D graphics acceleration
libskiahw
DSP accerated JPEG
libopencorehw
video overlay + DSP
libcamera
Camera
Interface
libagl
Software OpenGL
liboverlay
MDP video overlay





NEON instructions

Advanced ARM SIMD

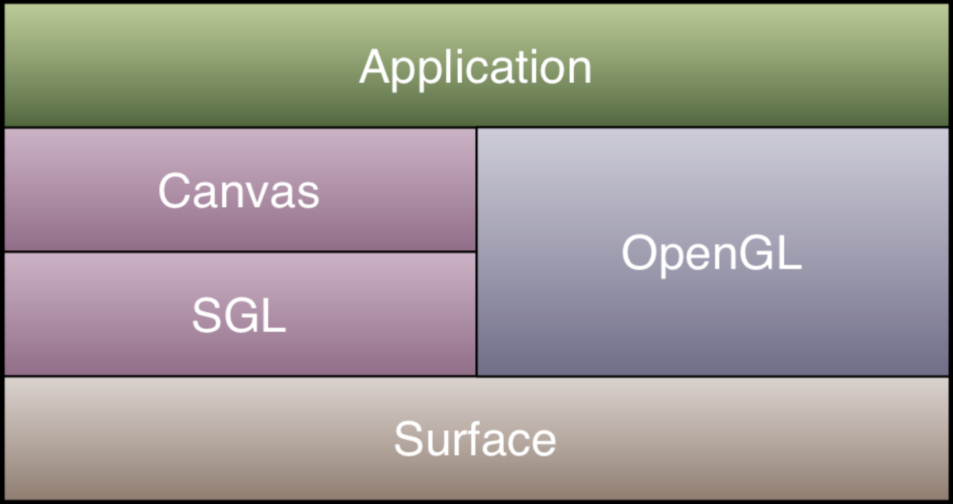
PixelFlinger JIT

optimized scanline_t32cb16

Reference benchmark on Beagleboard (TI OMAP353x) at 500 MHz

scanline_t32cb16_c memory bandwidth:	31.63 MB/s
scanline_t32cb16_neon memory bandwidth:	147.69 MB/s

It could dramatically improve boot animation performance.



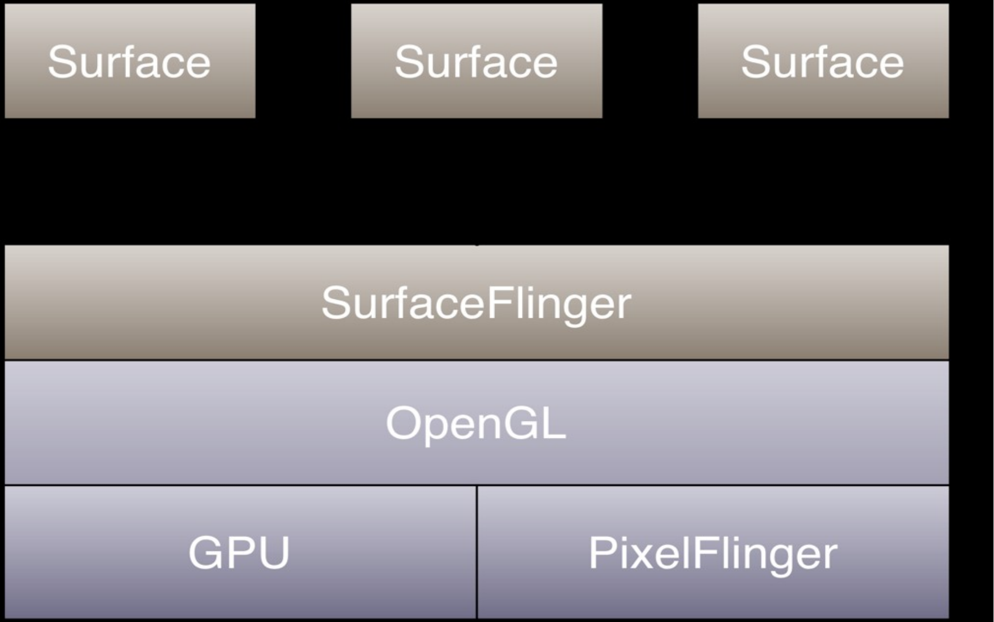
Application

Canvas

SGL

OpenGL

Surface



Surface

Surface

Surface

SurfaceFlinger

OpenGL

GPU

PixelFlinger

NEON instructions

Advanced ARM SIMD

PixelFlinger JIT

optimized t32cb16blend

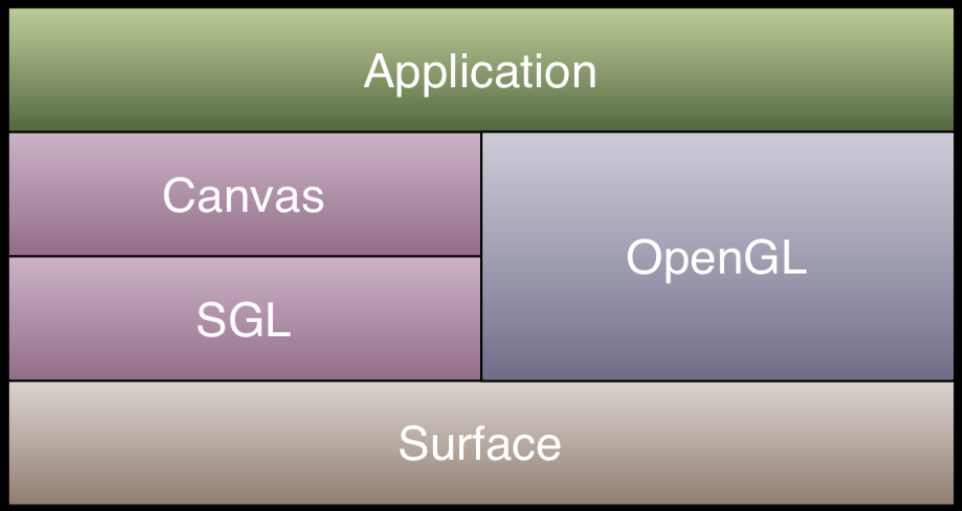
Reference benchmark on Beagleboard 500MHz:

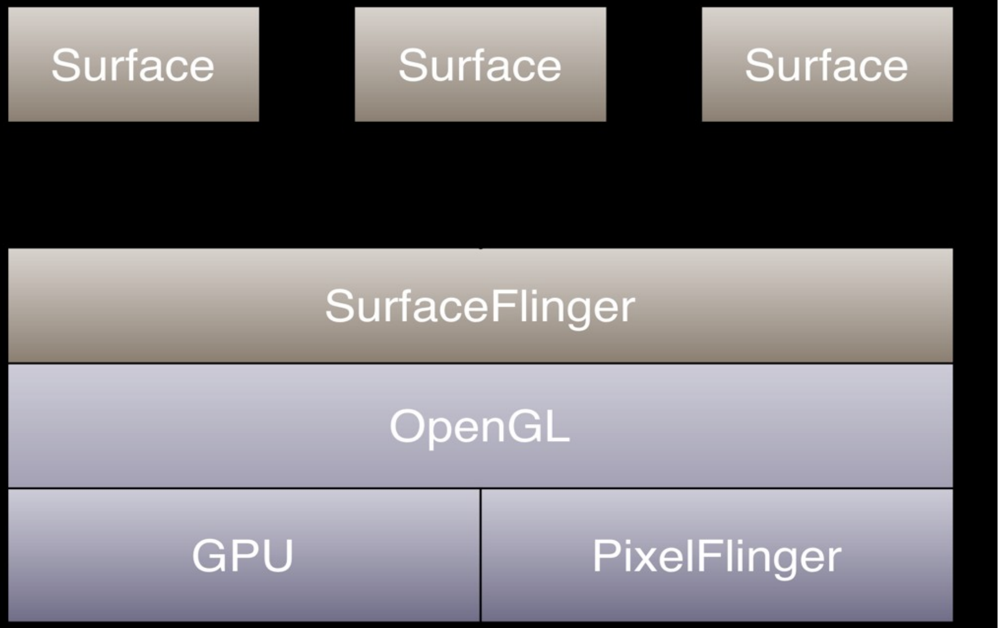
```
    scanline_t32cb16blend_c memory bandwidth:        12.81 MB/s
    scanline_t32cb16blend_arm memory bandwidth:      57.61 MB/s
    scanline_t32cb16blend_neon memory bandwidth: 128.66 MB/s
```

scanline_t32cb16blend_c: generic C implementation.

scanline_t32cb16blend_arm: ARMv5 optimized by Android.

scanline_t32cb16blend_neon: ARMv7 tweaked implementation.





UBFX instruction

Signed and Unsigned Bit Field Extract. Copies adjacent bits from one register into the least significant bits of a second register, and sign extends or zero extends to 32 bits.

PixelFlinger JIT

00000077:03515104_00000000_00000000

(Blends a single color into an RGB565 buffer.)

Before: 27 inst/pixel, After: 24 inst/pixel, Improvement: 12.5%

00000077:03545404_00000A01_00000000

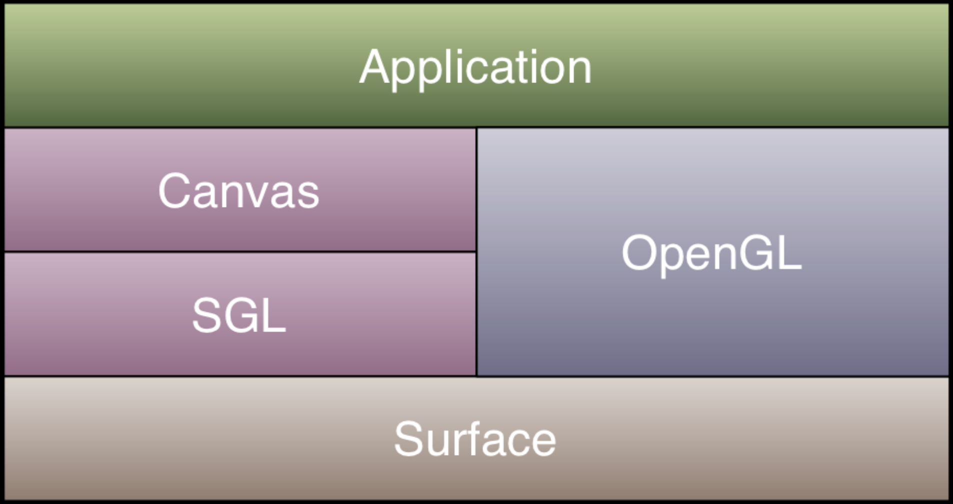
(Blends RGBA8888 texture into an RGB565 buffer using alpha.)

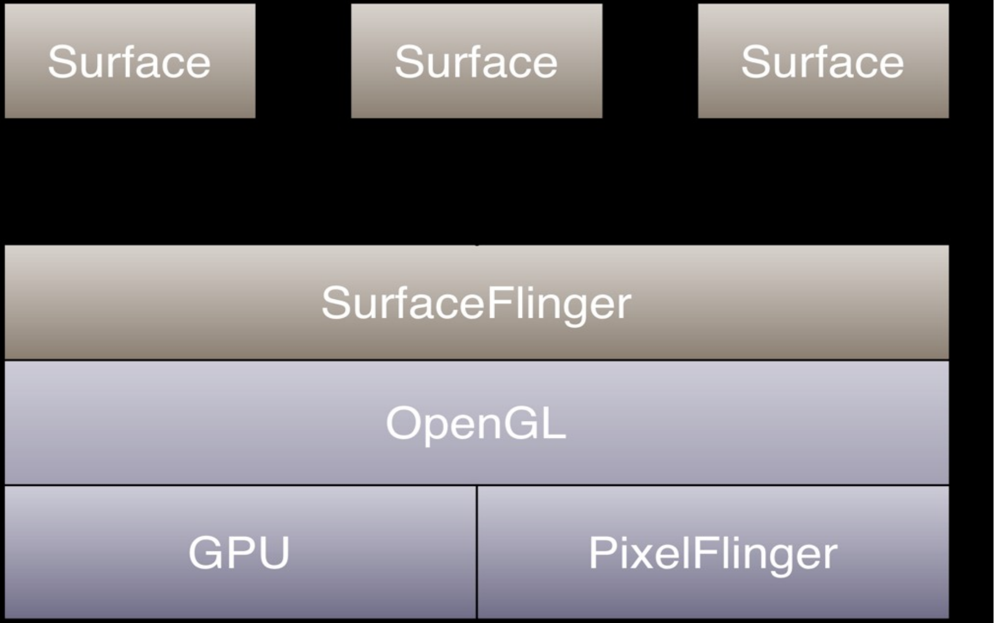
Before: 30 inst/pixel, After: 27 inst/pixel, Improvement: 11.1%

00000077:03545404_00000A04_00000000

(Blends RGB565 texture into an RGB565 buffer using alpha.)

Before: 29 inst/pixel, After: 27 inst/pixel, Improvement: 7.4%





UBXTB16 instruction

Introducing the UXTB16 instruction allows removal of some masking code, and is beneficial from a pipeline point of view - lots of UXTB16 followed by MUL sequences.

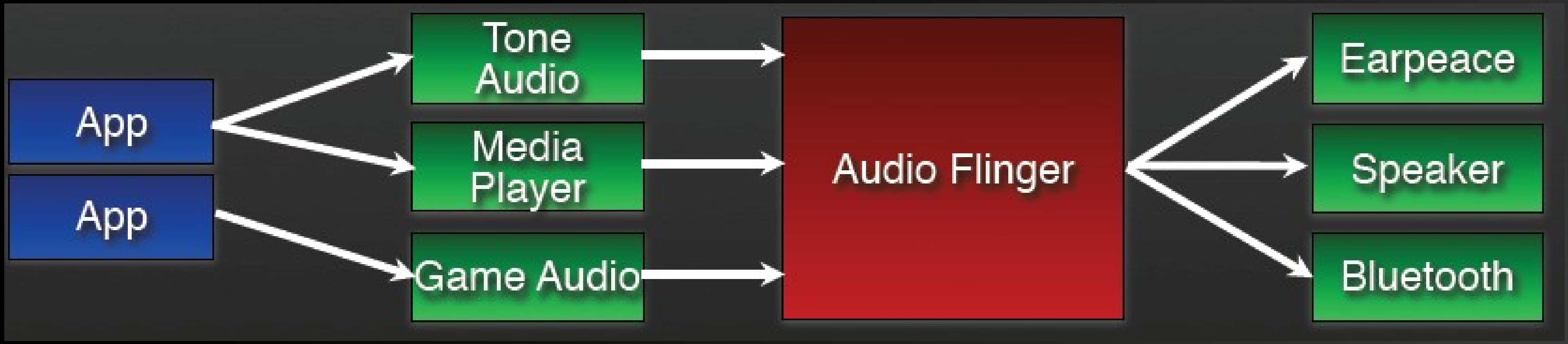
PixelFlinger JIT

Code has been scheduled for A8 pipeline, specifically aiming to allow multiplies to issue in pipeline 0, for efficient dual issue operation.

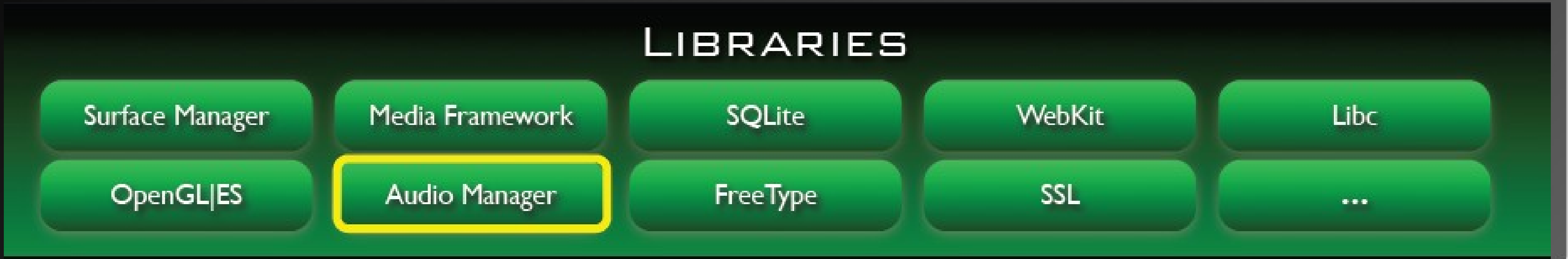
Testing on SpriteMethodTest (http://code.google.com/p/apps-for-android/) gives

8% improvement (12.7 vs. 13.7 fps.)

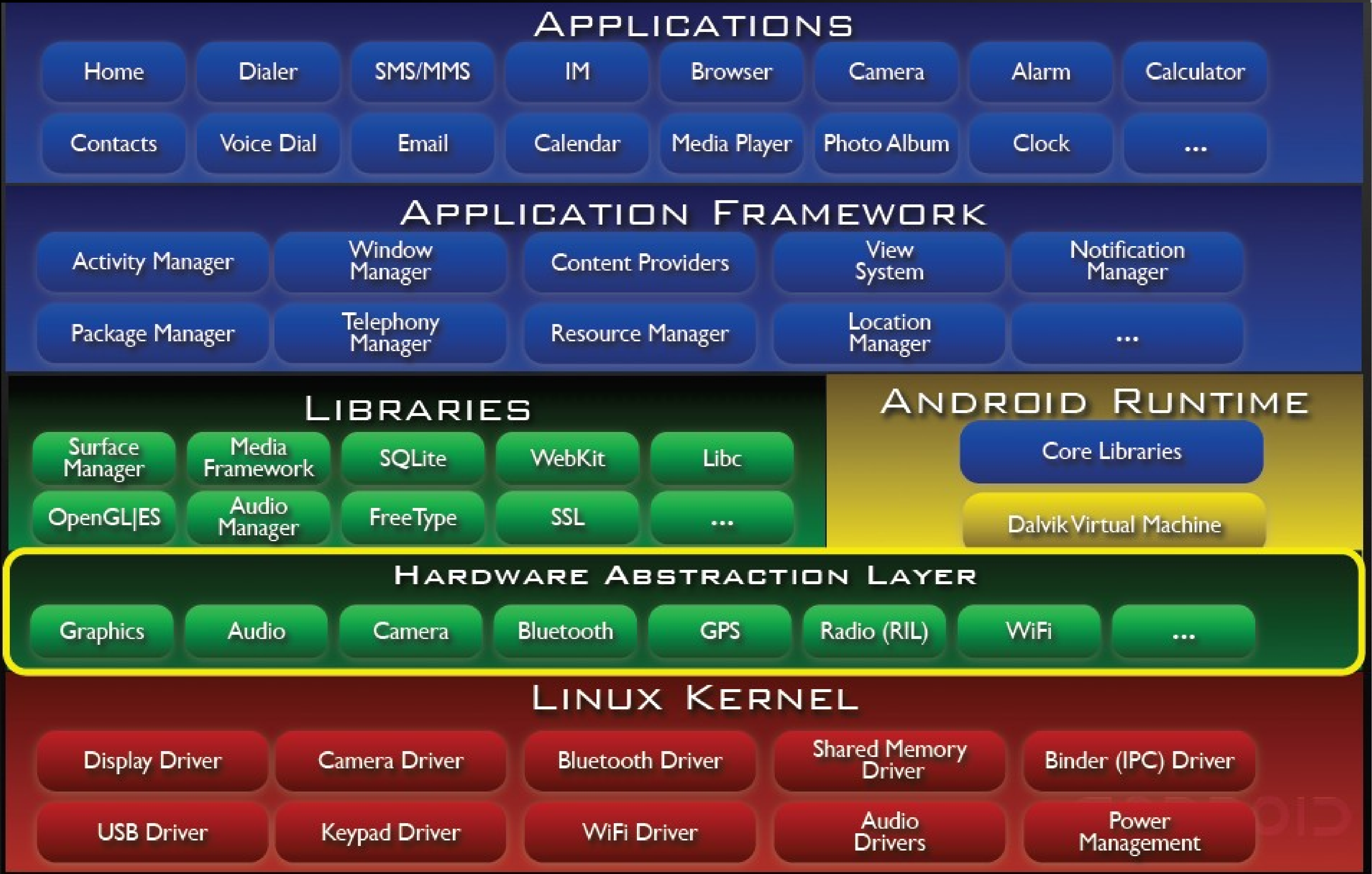
Native Server: Audio Flinger



- Manages all audio output devices
- Handles audio routing to various outputs
- 0xlab’s involvement
  - Extend the past experience about Android Audio processing and avoid unexpected / abnormal problems
  - Remove hard-coded Android implementations

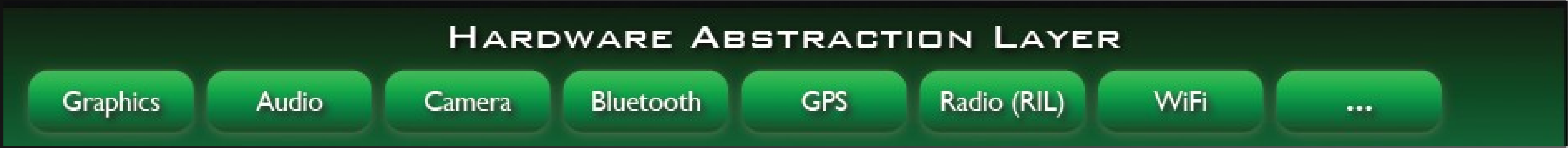


HAL (Hardware Abstraction Libraries)

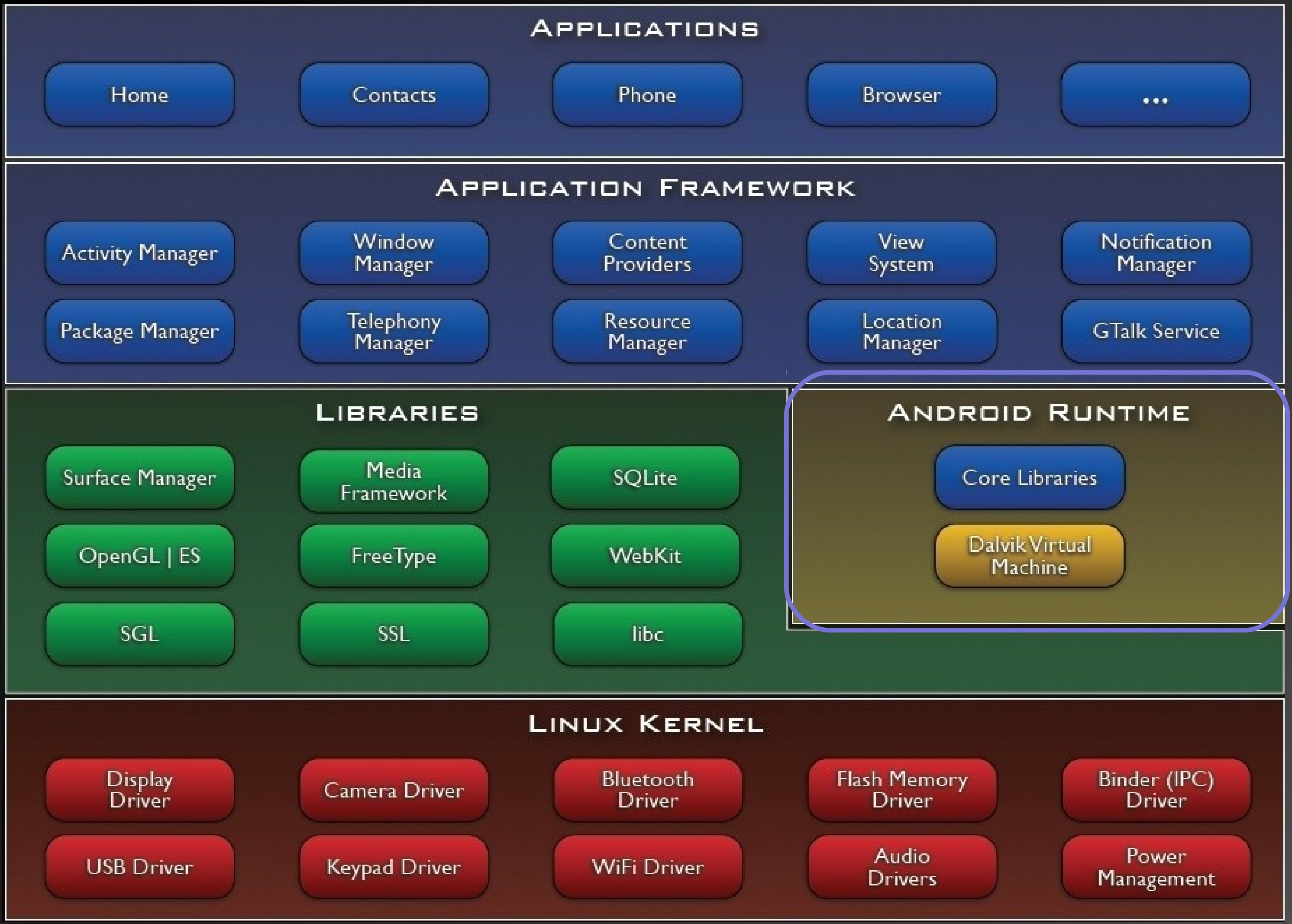


HAL (Hardware Abstraction Libraries)

- User space C/C++ library layer
- Defines the interface that Android requires hardware “drivers” to implement
- Separates the Android platform logic from the hardware interface
- 0xlab’s involvement
  - Ensure the software quality about WiFi / Bluetooth / FM including API level
  - Extra peripherals enablement on Beagleboard: Camera, Motion Sensor, GSM modem/3G data card (full source)
  - Properly handle GPL issues



Android Runtime



Android Runtime

Dalvik Virtual Machine is the core of Android Java Framework

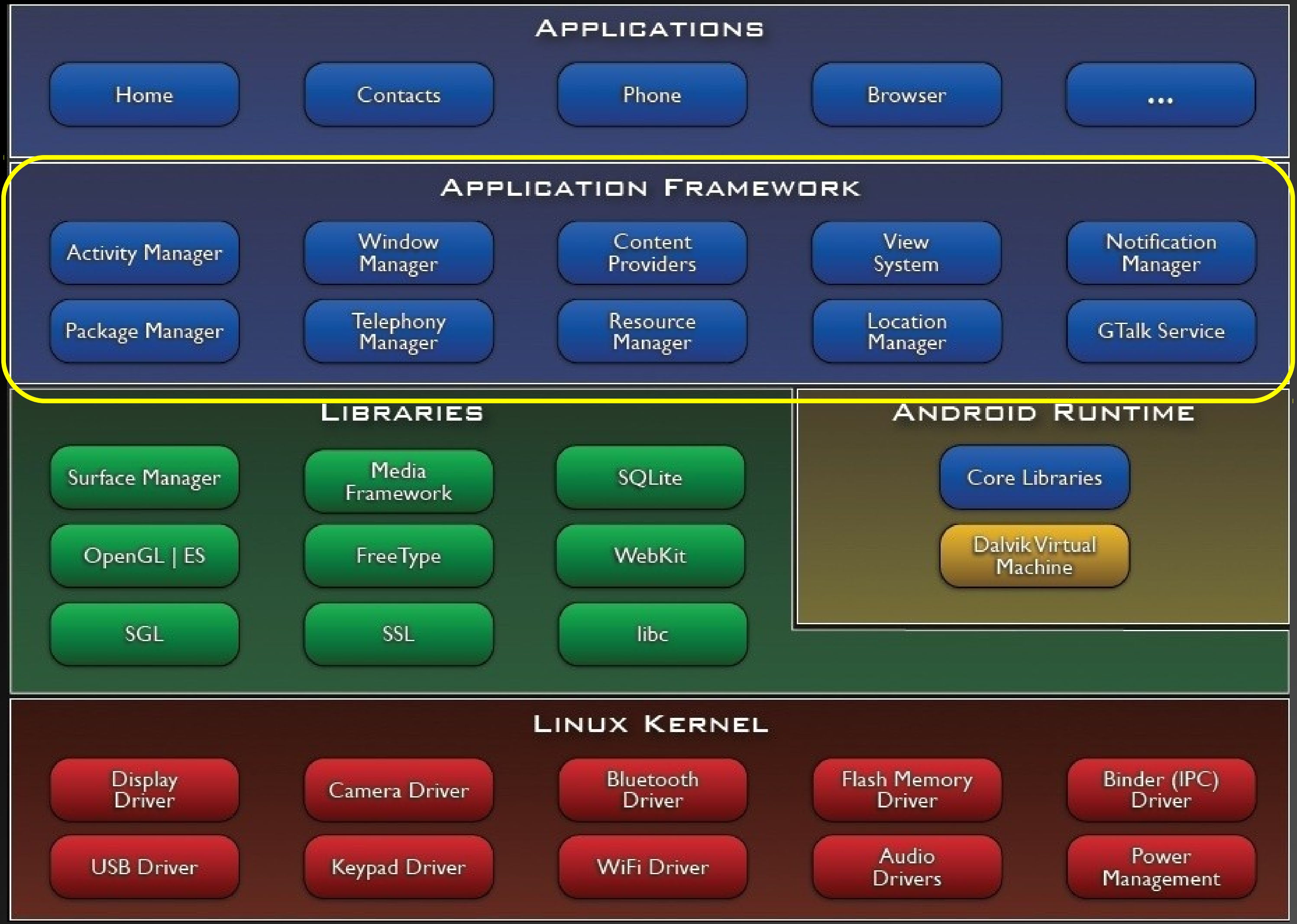
- DDMS (Dalvik Debug Monitor Server) could expose the systyem information

0xlab’s Enhancements

- Enable Just-In-Time compiler to improve Java execution (expectation: 2x speedup)
- System stability and security
- CTS specific code introspection

ANDROID RUNTIME
Core Libraries
Dalvik Virtual Machine

Application Framework



Application Framework

Activity manager
- Manage the life cycle of applications

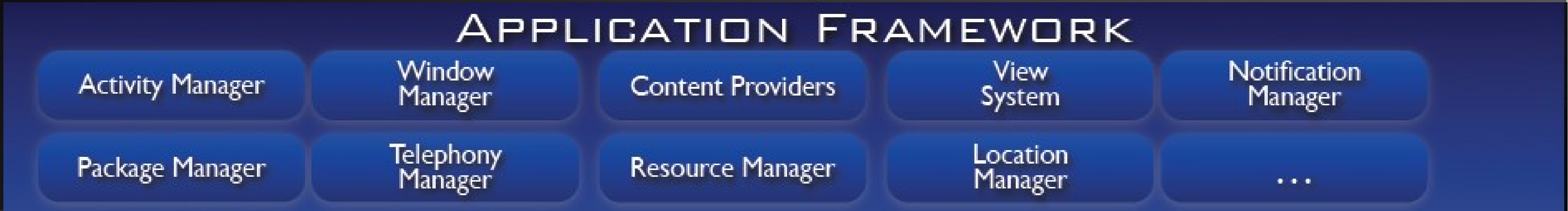
Content Provider
- Share data between applications

Resource Manager
- Manager non-code resource

Notification Manager
- Display custom alerts in the status bar

Views System
- A rich and extensible set, which can construct UI

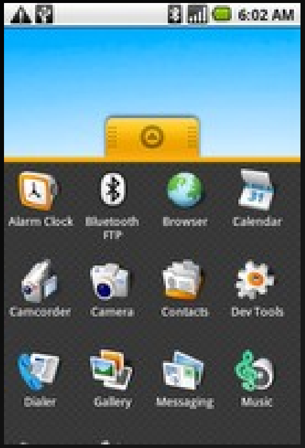
- 0xlab's Involvement
  - Comply with CTS
  - Eliminate race condition, system server crash, memory usage, etc.
  - Properly backport the fixes from Froyo branch

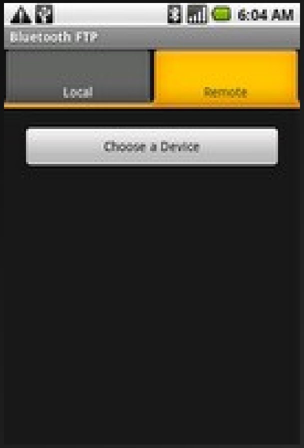

APPLICATION FRAMEWORK

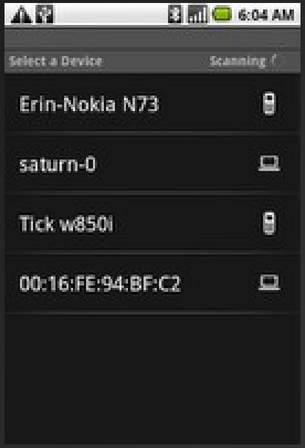
Activity Manager
Window Manager
Content Providers
View System
Notification Manager
Package Manager
Telephony Manager
Resource Manager
Location Manager
...

Bluetooth

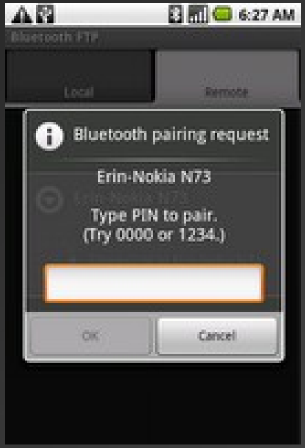
- OBEX: OPP, FTP profiles with UI support
  - Check http://i-miss-erin.blogspot.com/2009/10/android-bluetooth-ui-application-from.html
  - Used by CyanogenMod, a famous community build used on HTC Android devices.



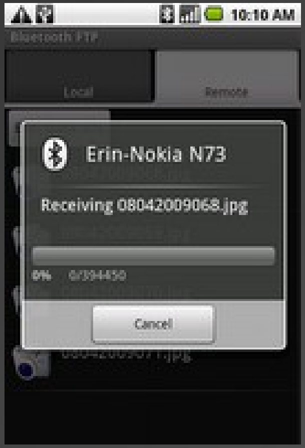




Erin-Nokia N73
saturn-0
Tick w850i
00:16:FE:94:BF:C2



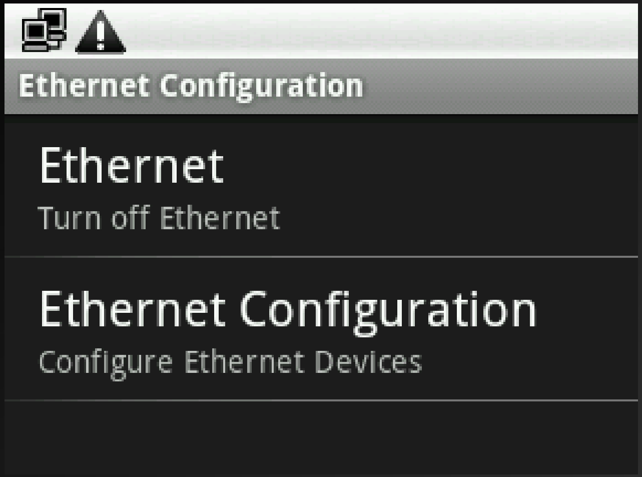
Bluetooth pairing request
Erin-Nokia N73
Type PIN to pair.
(Try 0000 or 1234.)



Erin-Nokia N73
Receiving 08042009068.jpg
0% 0/394450

Ethernet manager

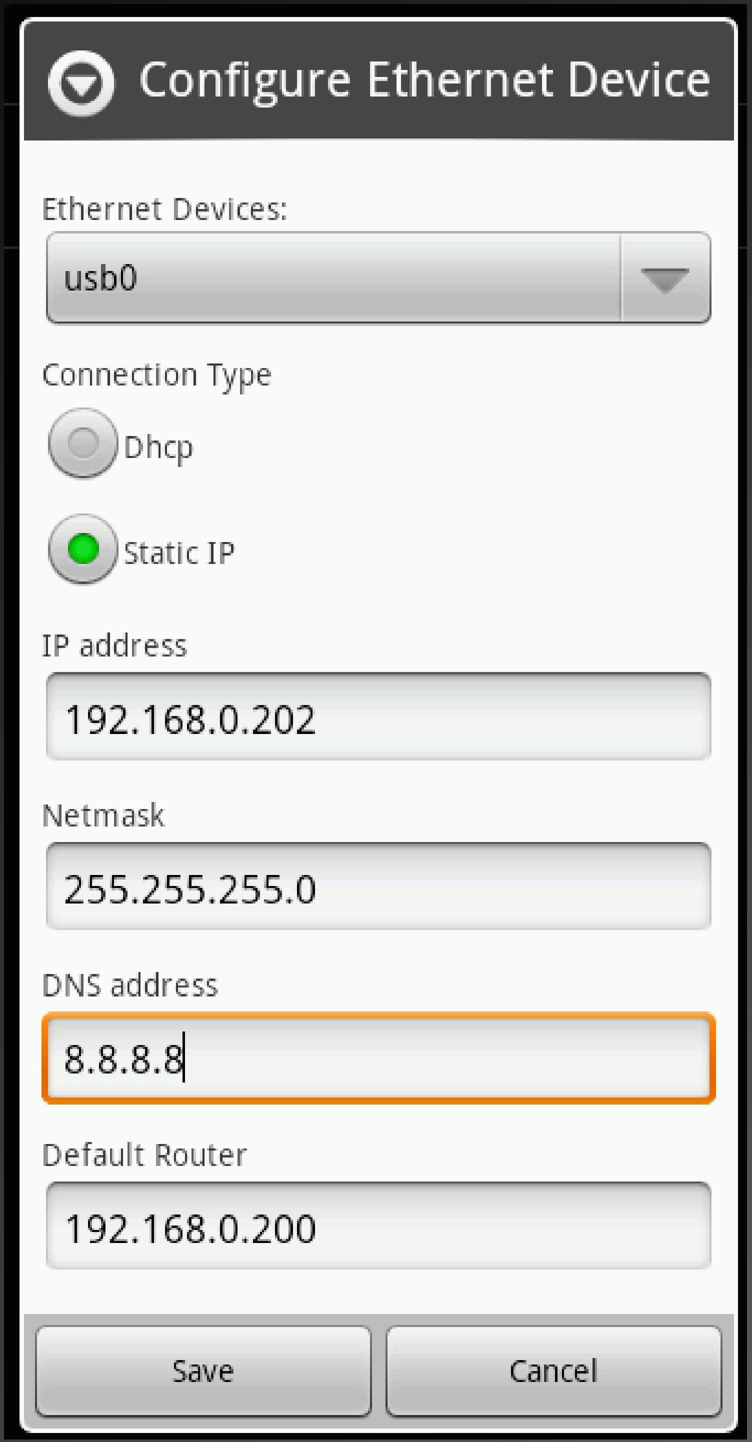
Supports configure and display Ethernet connection as well.



Ethernet Configuration

Ethernet
Turn off Ethernet

Ethernet Configuration
Configure Ethernet Devices



Configure Ethernet Device

Ethernet Devices: usb0

Connection Type
○ Dhcp
● Static IP

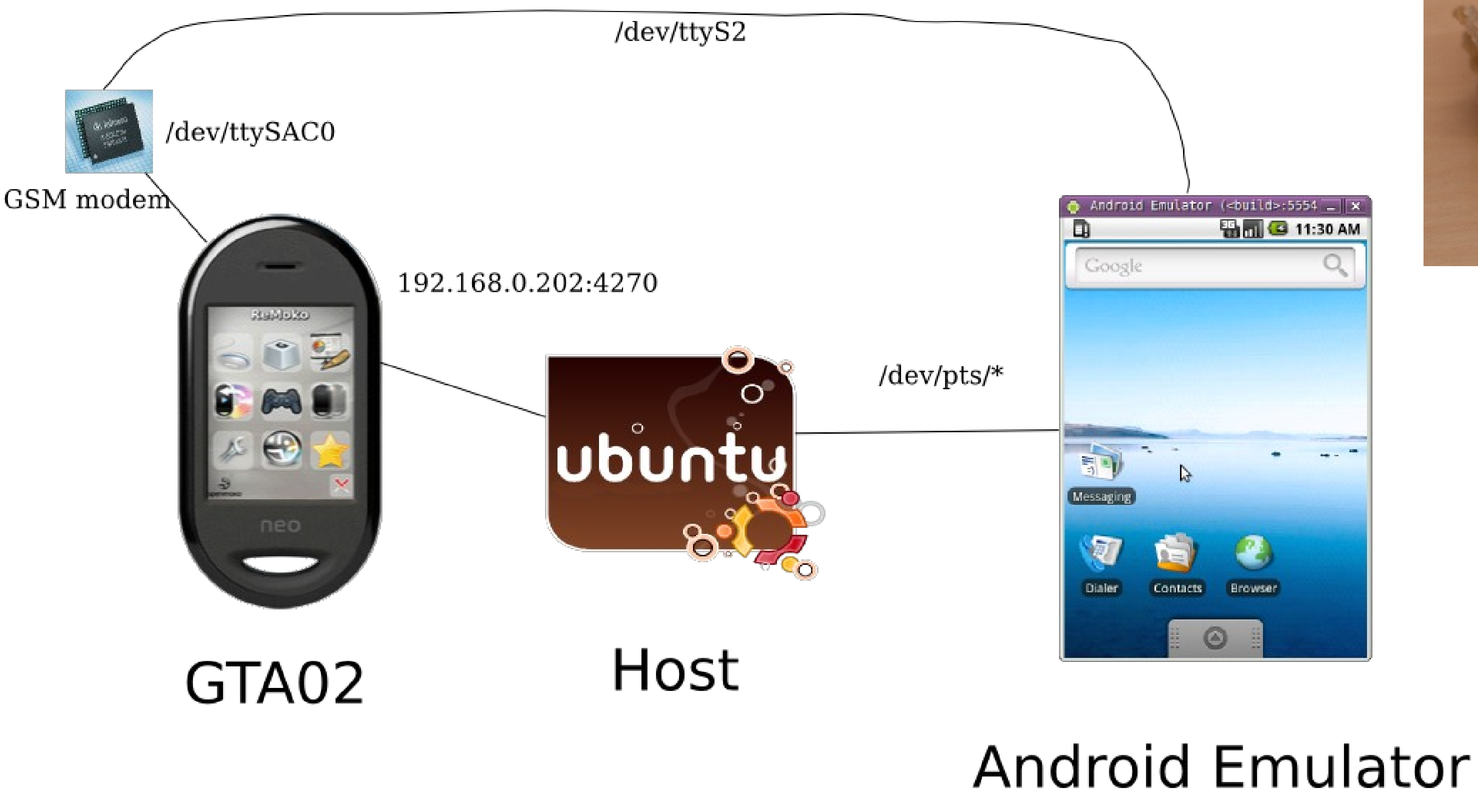
IP address: 192.168.0.202

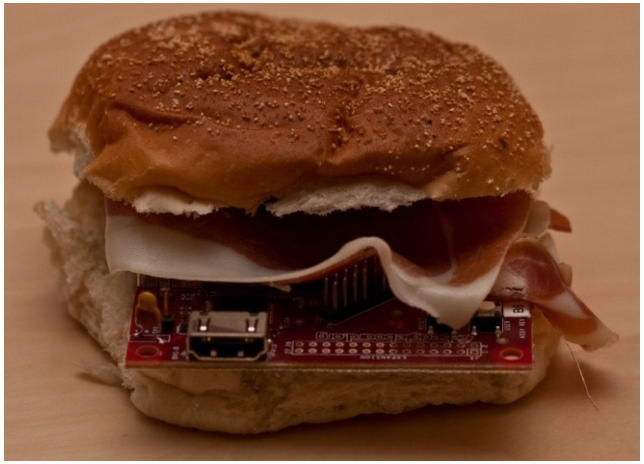
Netmask: 255.255.255.0

DNS address: 8.8.8.8

Default Router: 192.168.0.200

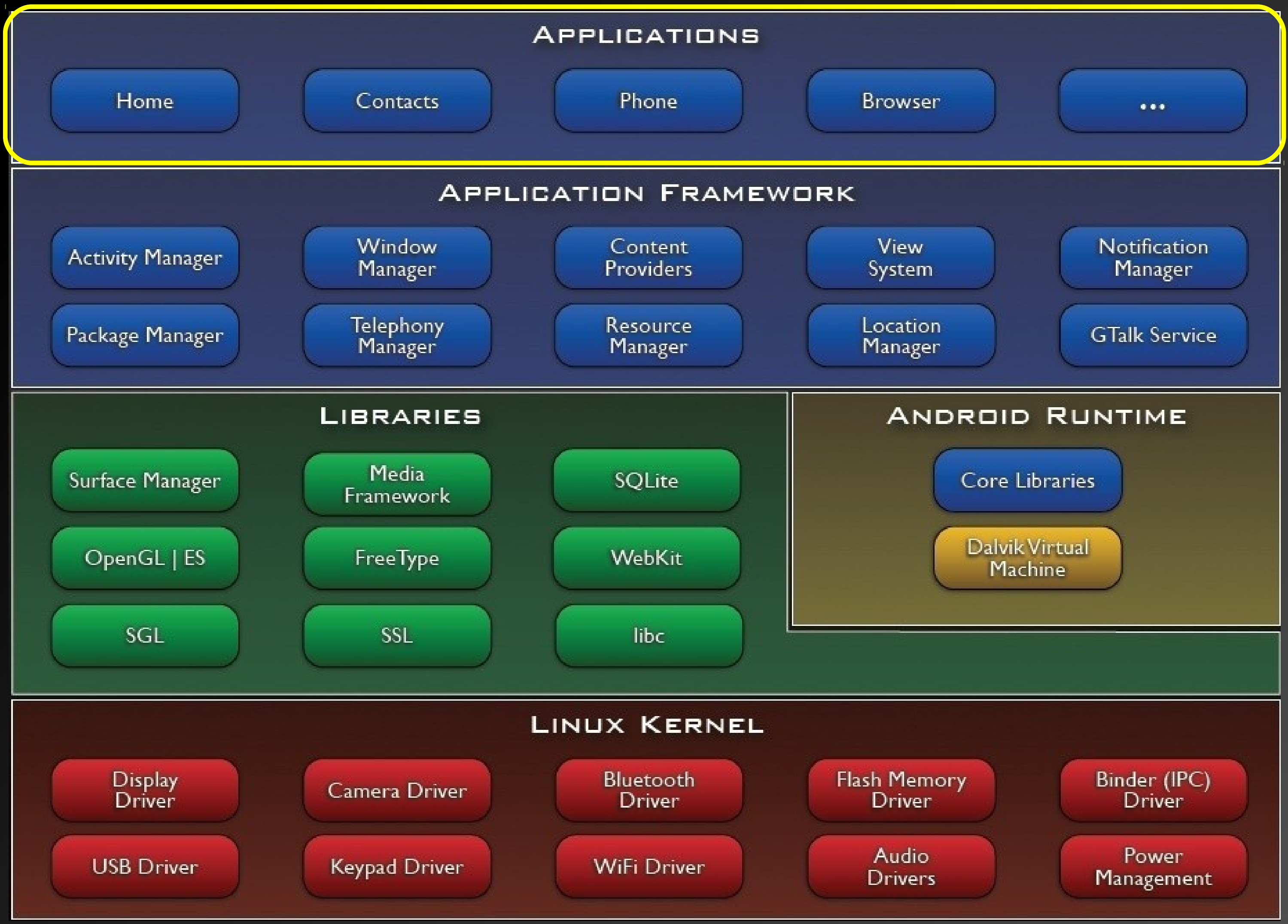
Even able to make phone call through external GSM modem





- External modem.

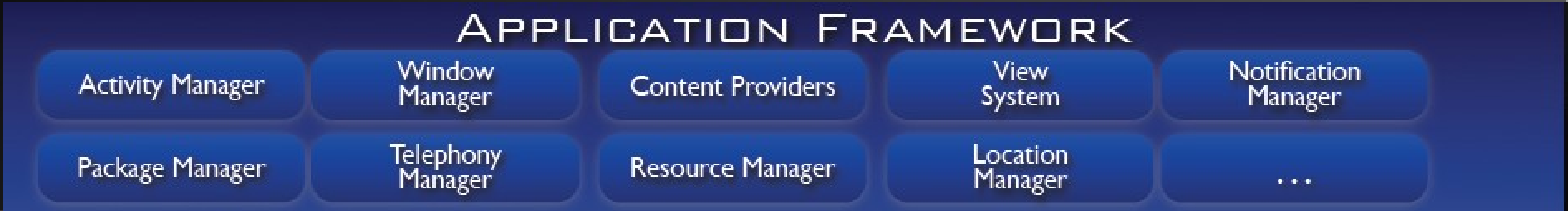
Applications



Applications

Launcher2, Camera, Album, Contacts, Email, Messaging, Music, Phone, Alarm, etc.

- 0xlab’s Improvements
  - Extensive and revised Launcher2 (re-)implementation
    - VGA/HVGA Display support
    - Visual effects smoothly on OMAP3 and Qualcomm 7K (even no GPU) platforms
    - Pretty straightforward visual customizations
    - Rapid development with art designer
  - Automated Testing Framework accelerates the UI component verifications



APPLICATION FRAMEWORK

Activity Manager

Window Manager

Content Providers

View System

Notification Manager

Package Manager

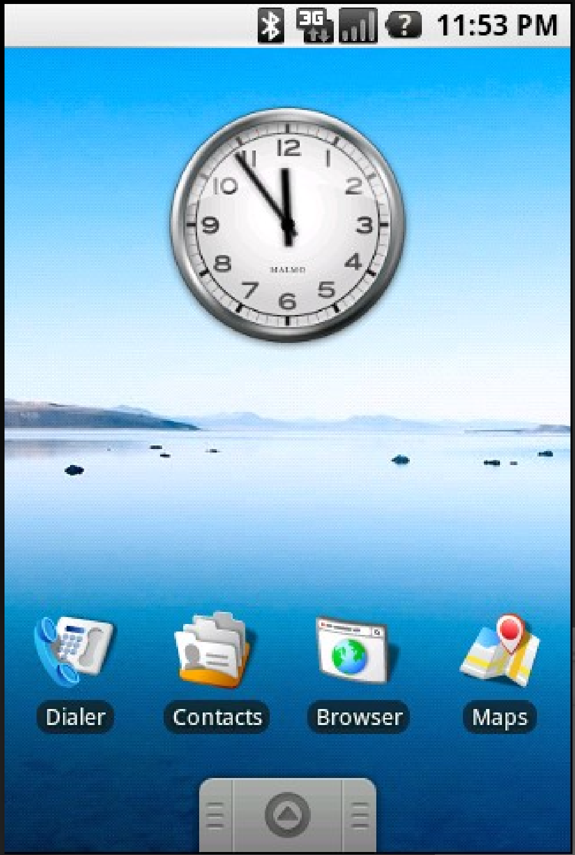
Telephony Manager

Resource Manager

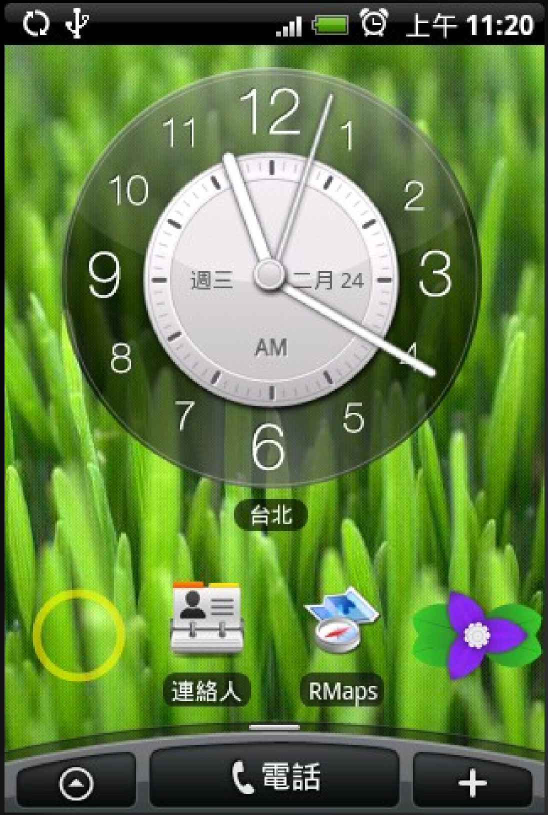
Location Manager

...

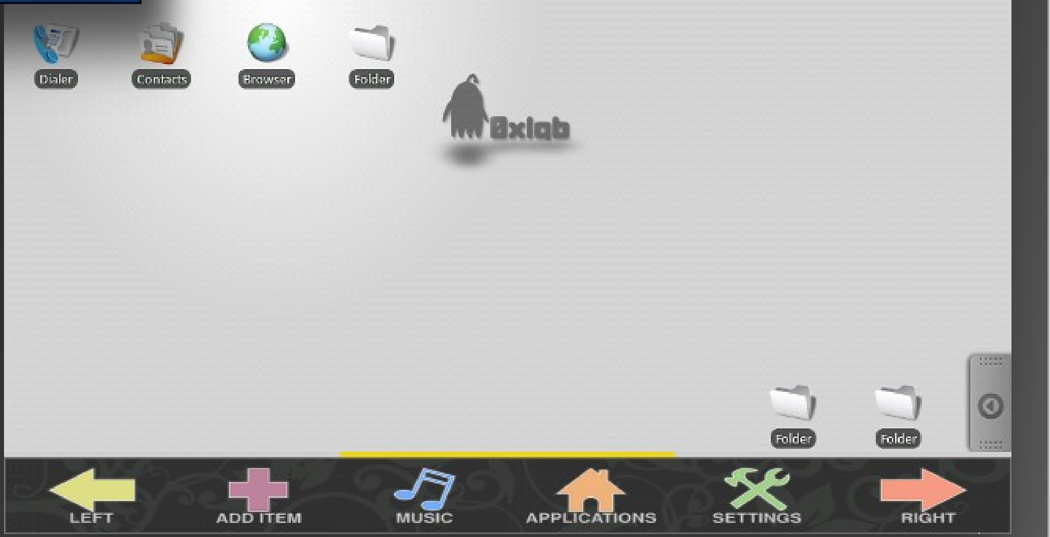
Android Official



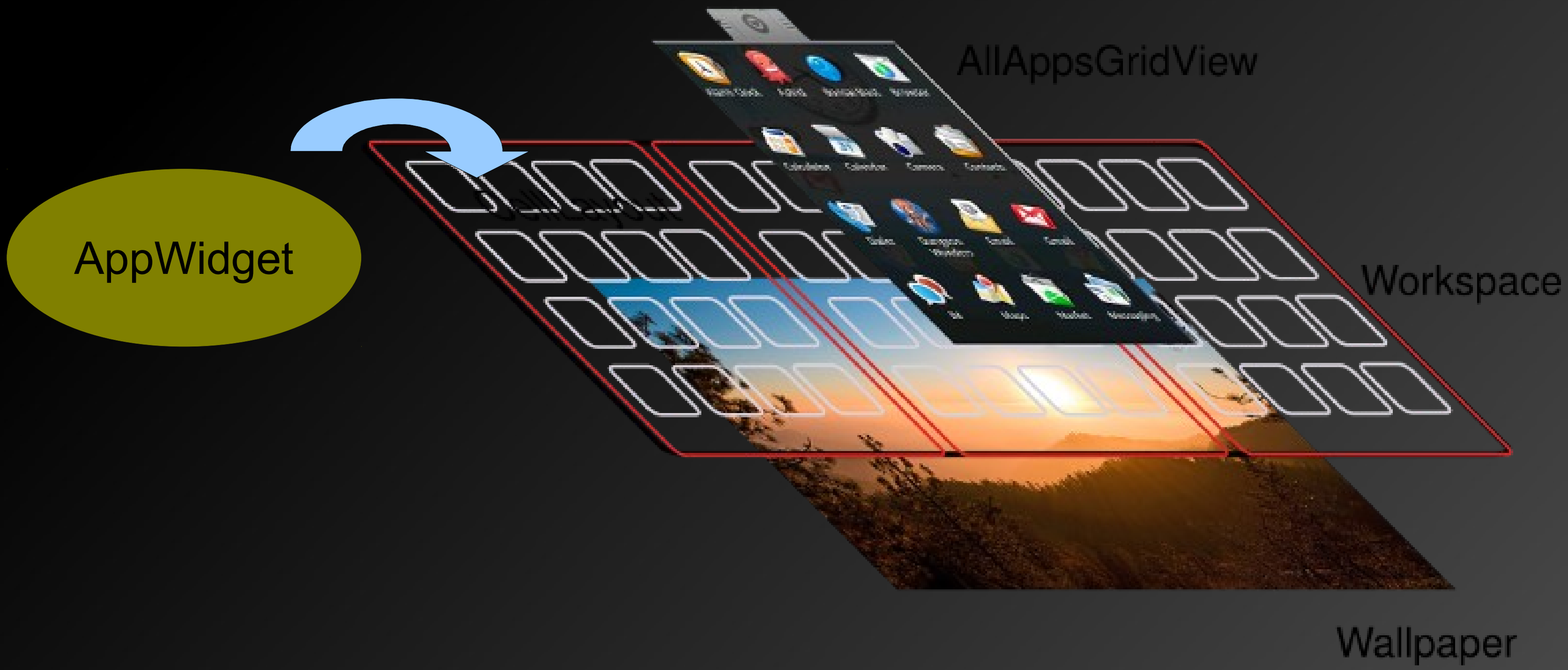
HTC-Sense UI



0xdroid



Disassemble Launcher



AllAppsGridView

AppWidget

CellLayout

Workspace

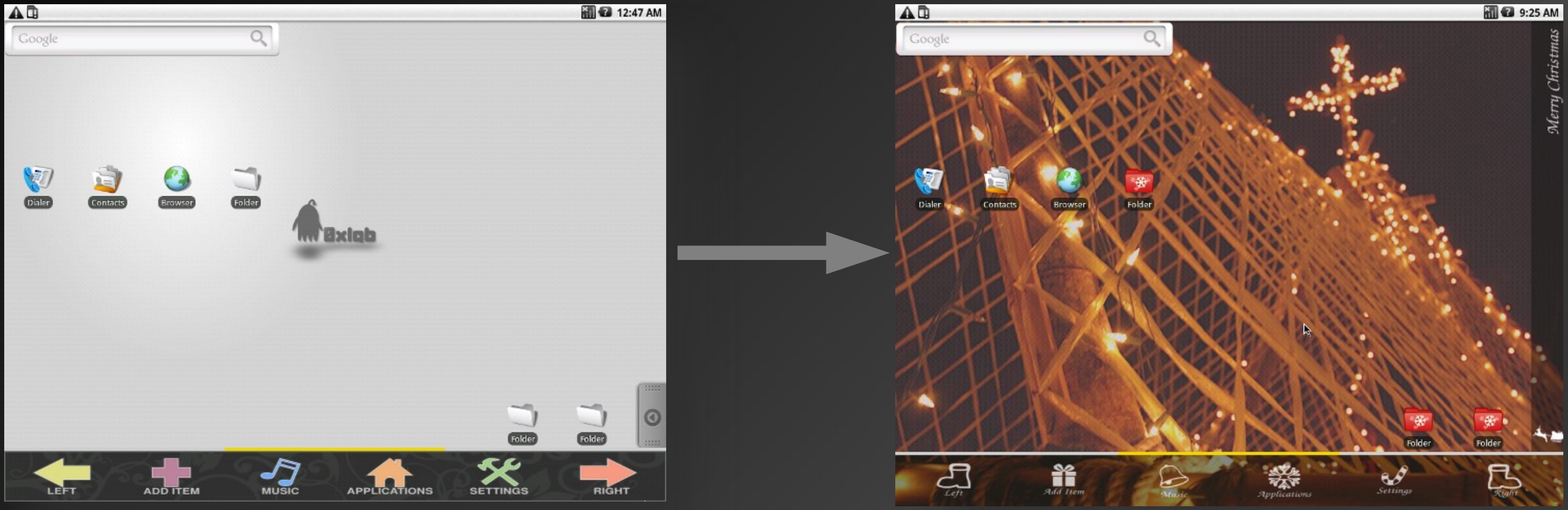
Wallpaper

Some UI changes by 0xLab

- BottomBar

Source code: **http://gitorious.org/0xdroid/packages_apps_launcher**

- PositionBar
- Visible Hint
- ThemeSelector
  - http://code.google.com/p/0xdroid/wiki/LauncherTheme



企盼您的協助

- 0xdroid + Beagleboard 是個理想的開放軟硬體平台，適合作研究實驗或教學應用（如交通大學）
- 廣泛的測試與回饋
- 提供新的應用（概念或實做）

思考：借力使力 – 如何讓 Android 善用社群已有的豐富寶藏，放入社群裡的優質套件

- 改進硬體抽象層，降低移植的複雜度
- 將成果分享

聯繫 0xlab/0xdroid 開發團隊

- 0xdroid Roadmap: http://code.google.com/p/0xdroid/wiki/Roadmap
- Source repository: http://gitorious.org/0xdroid
- Wiki: http://code.google.com/p/0xdroid/w/list
- Demo videos: http://www.youtube.com/channel/0xlab
- Mailing-list:
  - General discussion: http://groups.google.com/group/0xlab-discuss
  - Technical / Development: http://groups.google.com/group/0xlab-devel
- IRC channel (FreeNode): #0xlab

Reference

- 0xlab website
  http://0xlab.org/
- 0xdroid project
  http://code.google.com/p/0xdroid/
- CyanogenMod
  http://www.cyanogenmod.com/
- Android-x86
  http://www.android-x86.org/
- Open Embedded Software Foundation (OESF)
  http://www.oesf.jp/

AzureWave
AzureWave Technologies, Inc.

Google

Texas Instruments

ODROID

BRONCHO
life work enjoy

Thank you

Wake up your device quickly and customize it with personal style

http://0xlab.org