

# Android Platform Debugging and Development

Android Builders Summit 2014

**Karim Yaghmour**

@karimyaghmour

karim.yaghmour@opersys.com





These slides are made available to you under a Creative Commons Share-Alike 3.0 license. The full terms of this license are here:  
<https://creativecommons.org/licenses/by-sa/3.0/>

Attribution requirements and misc., PLEASE READ:

- This slide must remain as-is in this specific location (slide #2), everything else you are free to change; including the logo :-)
- Use of figures in other documents must feature the below “Originals at” URL immediately under that figure and the below copyright notice where appropriate.
- You are free to fill in the “Delivered and/or customized by” space on the right as you see fit.
- You are FORBIDDEN from using the default “About” slide as-is or any of its contents.
- You are FORBIDDEN from using any content provided by 3<sup>rd</sup> parties without the EXPLICIT consent from those parties.

(C) Copyright 2013-2014, Opersys inc.

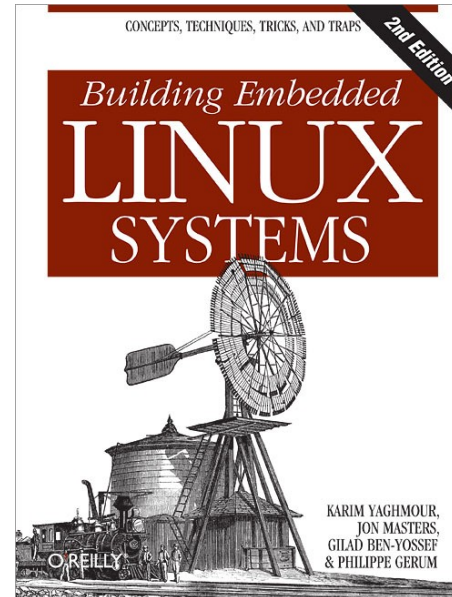
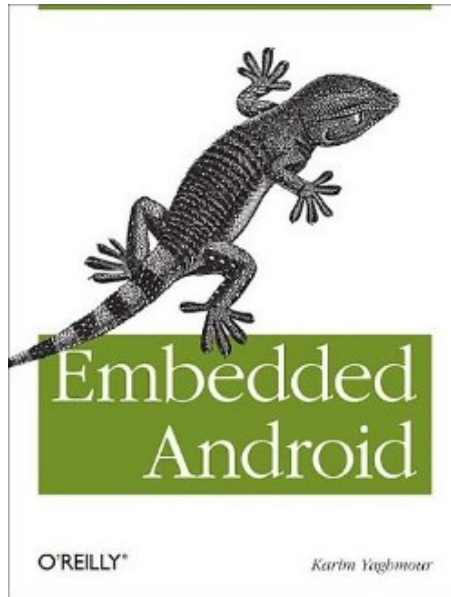
These slides created by: Karim Yaghmour

Originals at: [www.opersys.com/community/docs](http://www.opersys.com/community/docs)

Delivered and/or customized by

# About

- Author of:



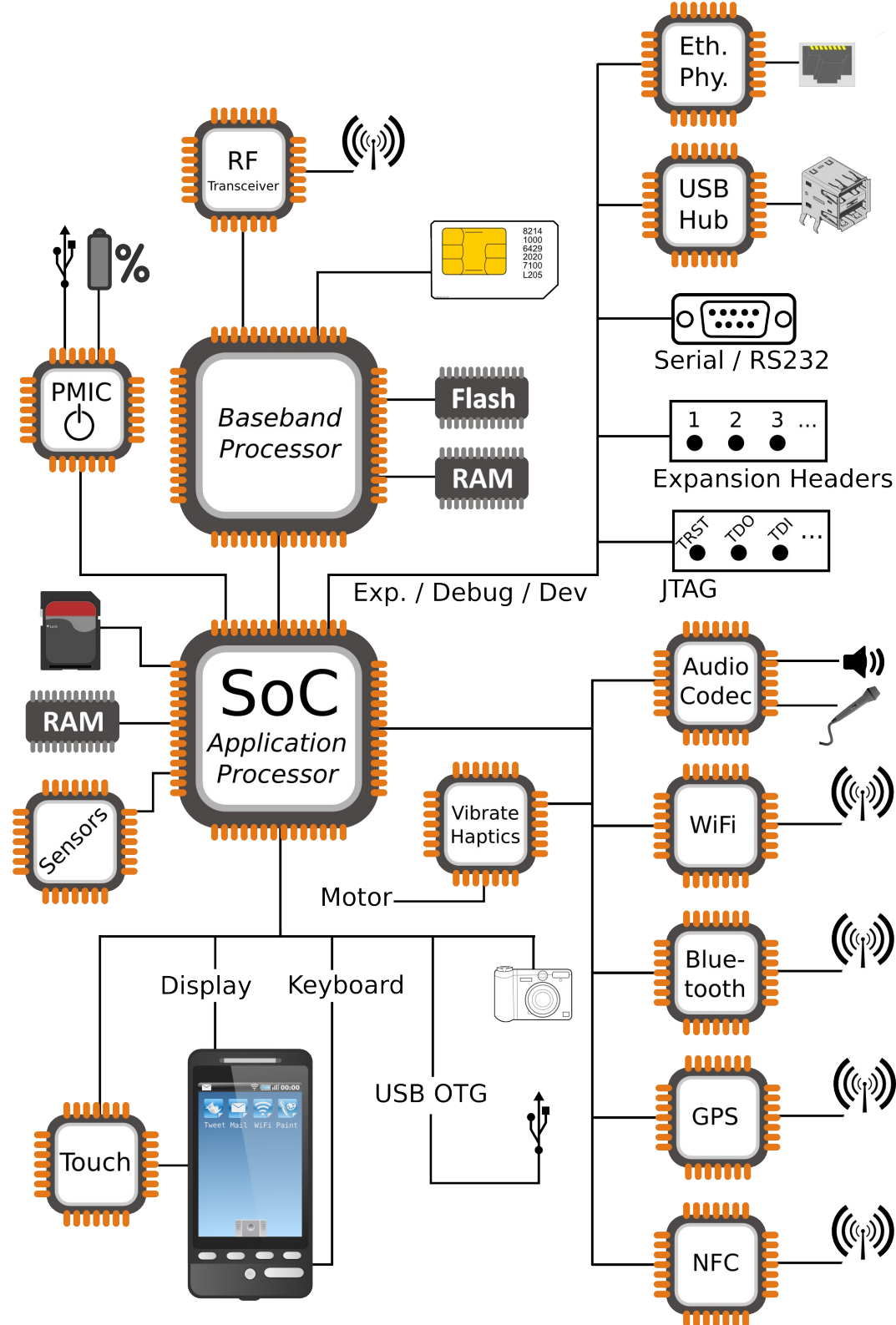
- Introduced Linux Trace Toolkit in 1999
- Originated Adeos and relayfs (kernel/relay.c)
- Training, Custom Dev, Consulting, ...

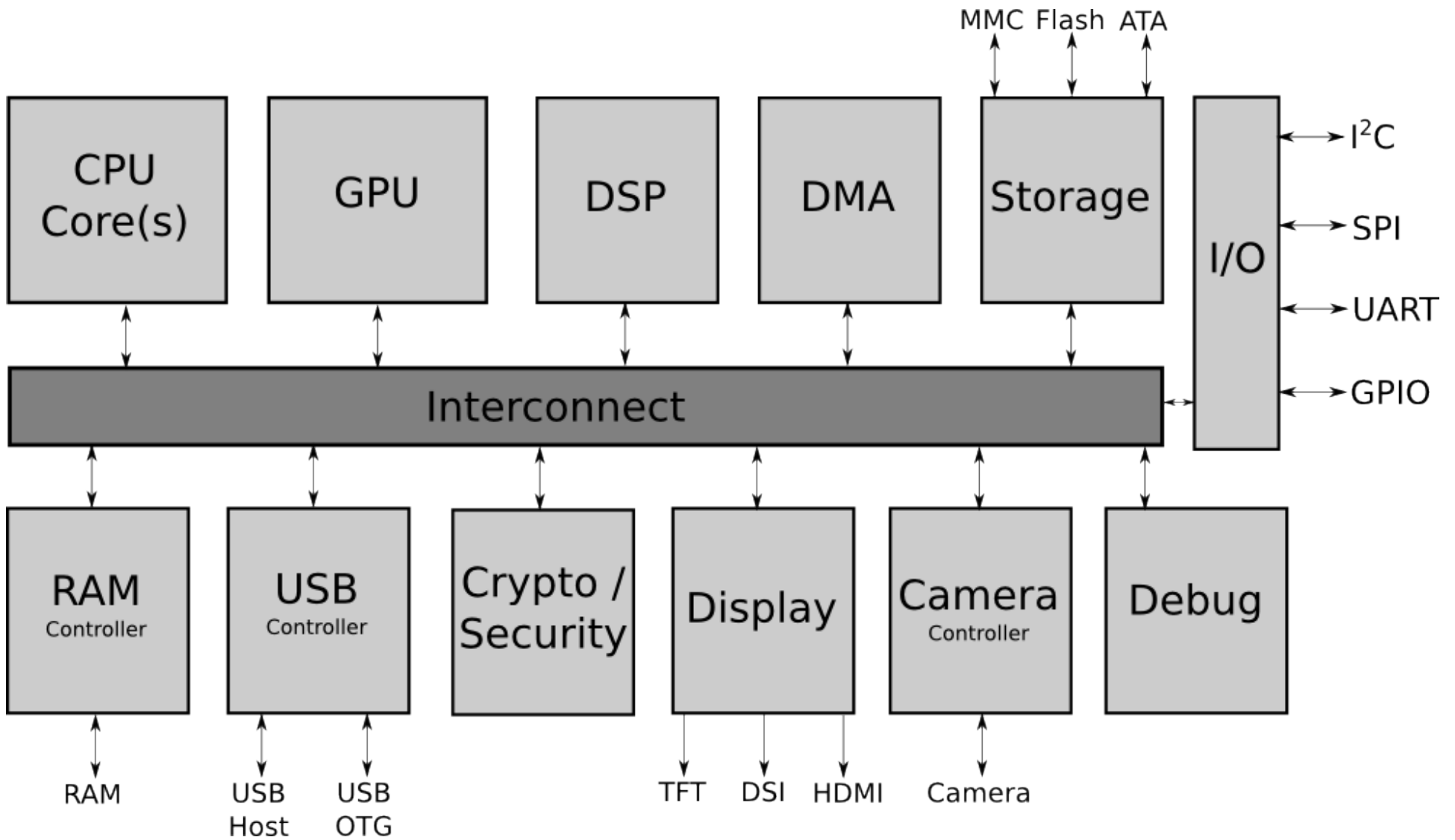
# Agenda

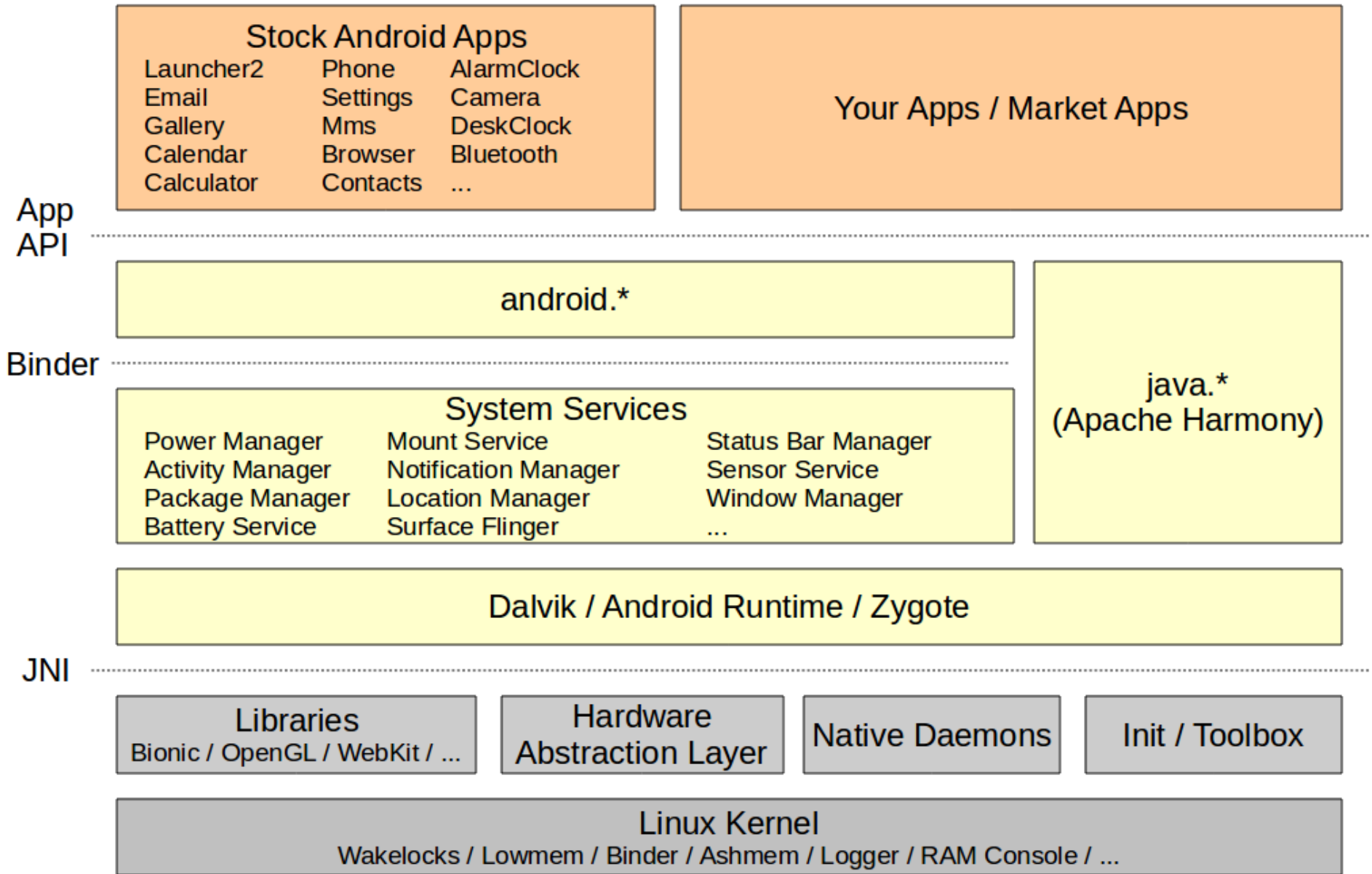
1. Architecture Basics
2. Development environment
3. Observing and monitoring
4. Interfacing with the framework
5. Working with the AOSP sources
6. Symbolic debugging
7. Detailed dynamic data collection
8. Benchmarking
9. Summing up

# 1. Architecture Basics

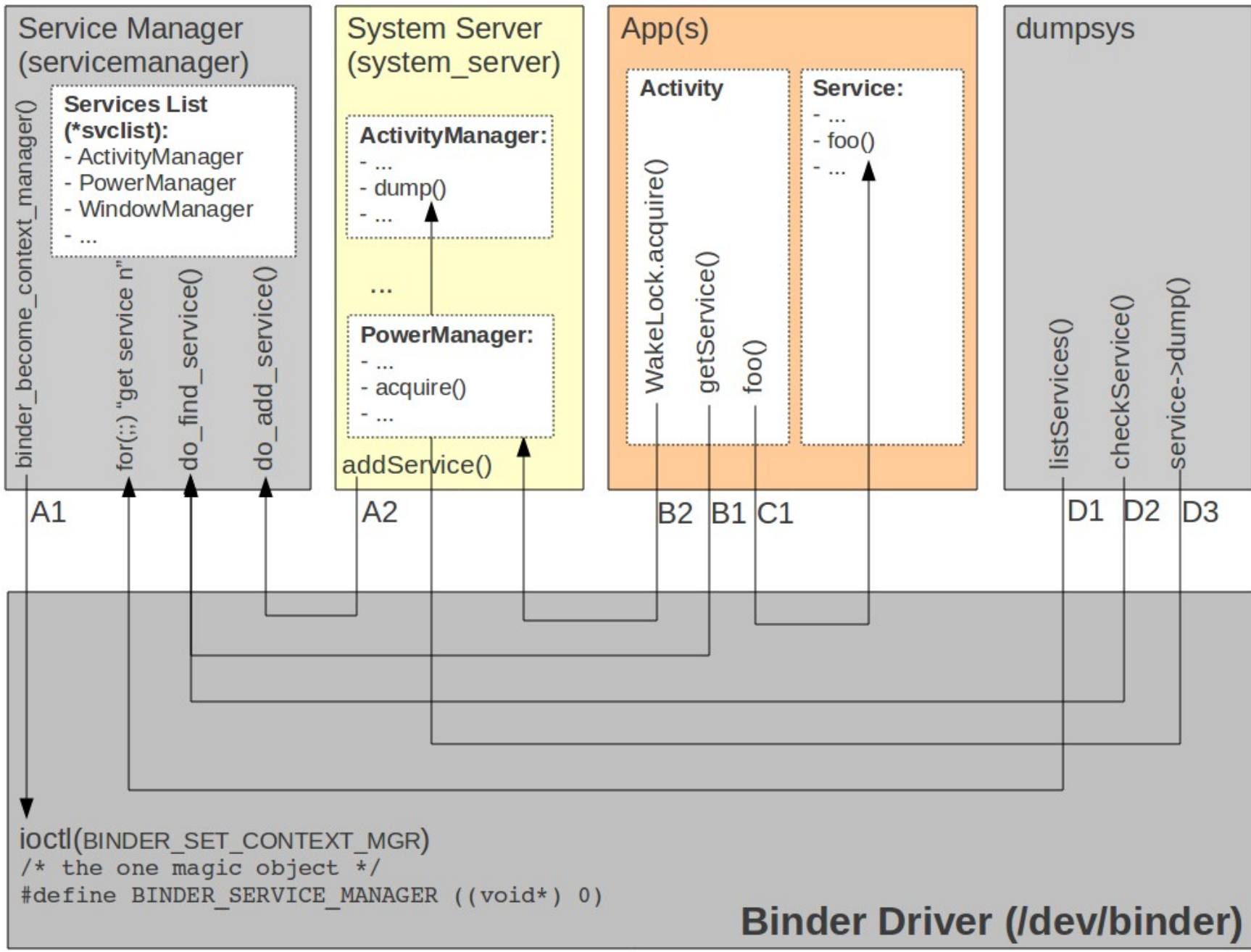
- Hardware used to run Android
- AOSP
- Binder
- System Services
- HAL











## System Services

### System Server

#### Java-built Services

Power Manager	Mount Service
Activity Manager	Notification Manager
Package Manager	Location Manager
Battery Service	Search Service
Window Manager	Wallpaper Service
Status Bar	Headset Observer
Clipboard Service	...

#### C-built Services

Sensor Service

#### Surface Flinger

#### Media Service

Audio Flinger  
Media Player Service  
Camera Service  
Audio Policy Service

##### Includes:

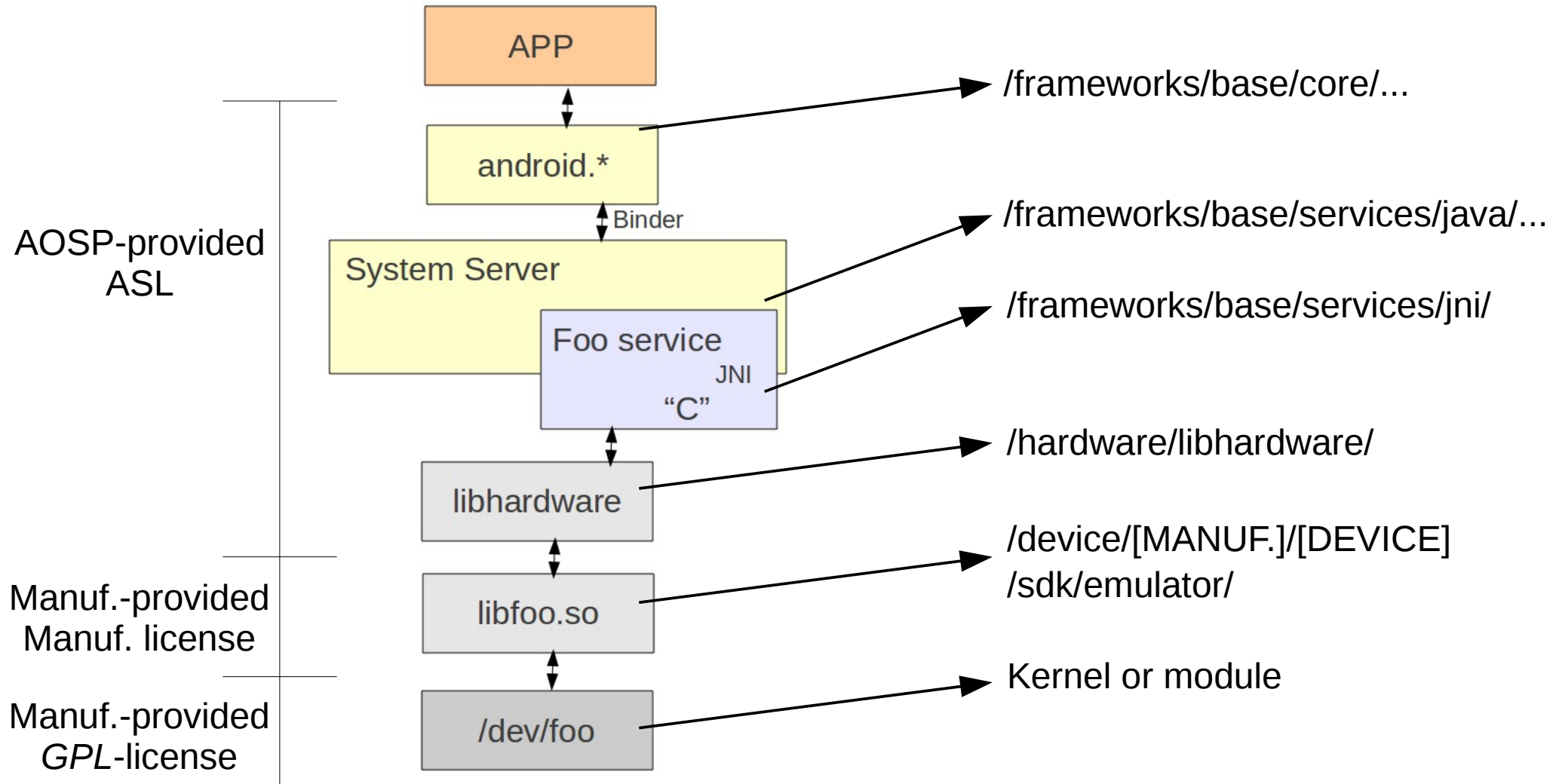
- StageFright
- Audio effects
- DRM framework

#### Phone App

#### JNI

Native Methods for  
Java-built Services

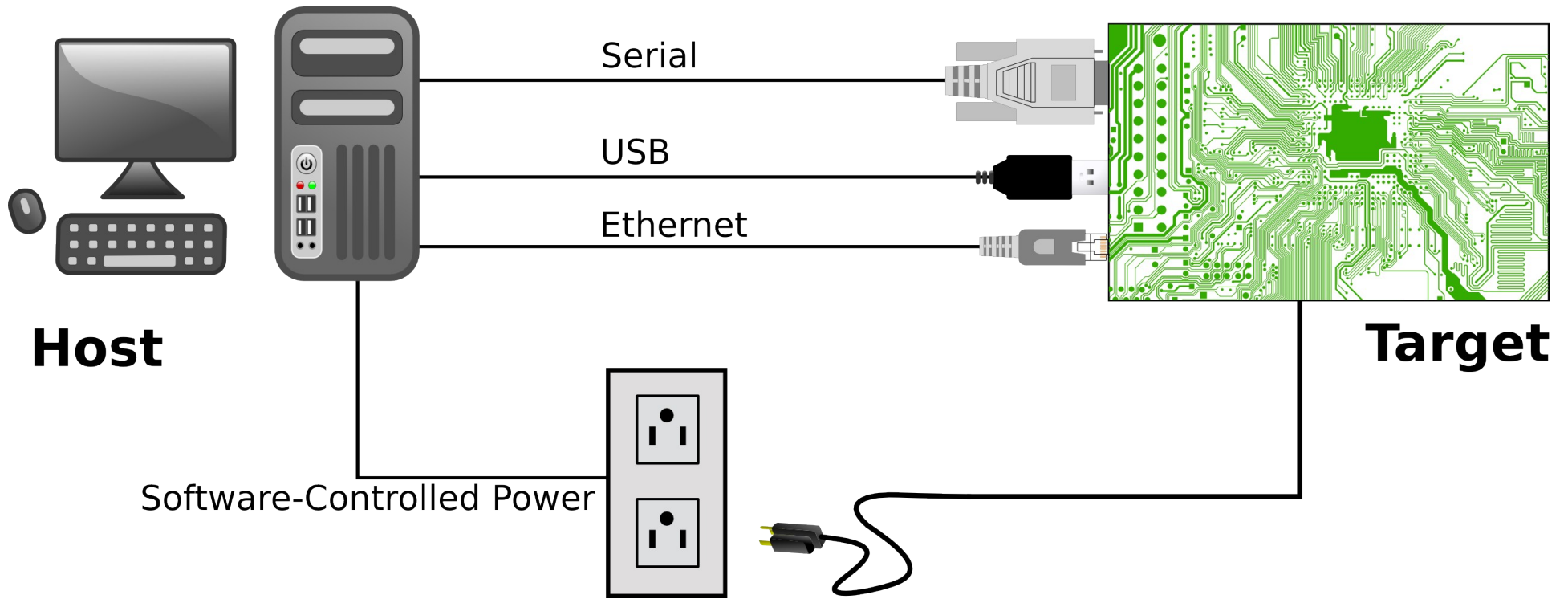
Hardware Abstraction Layer



# 2. Development Environment

- Host / Target setup
- IDE / Editor
- Eclipse setup

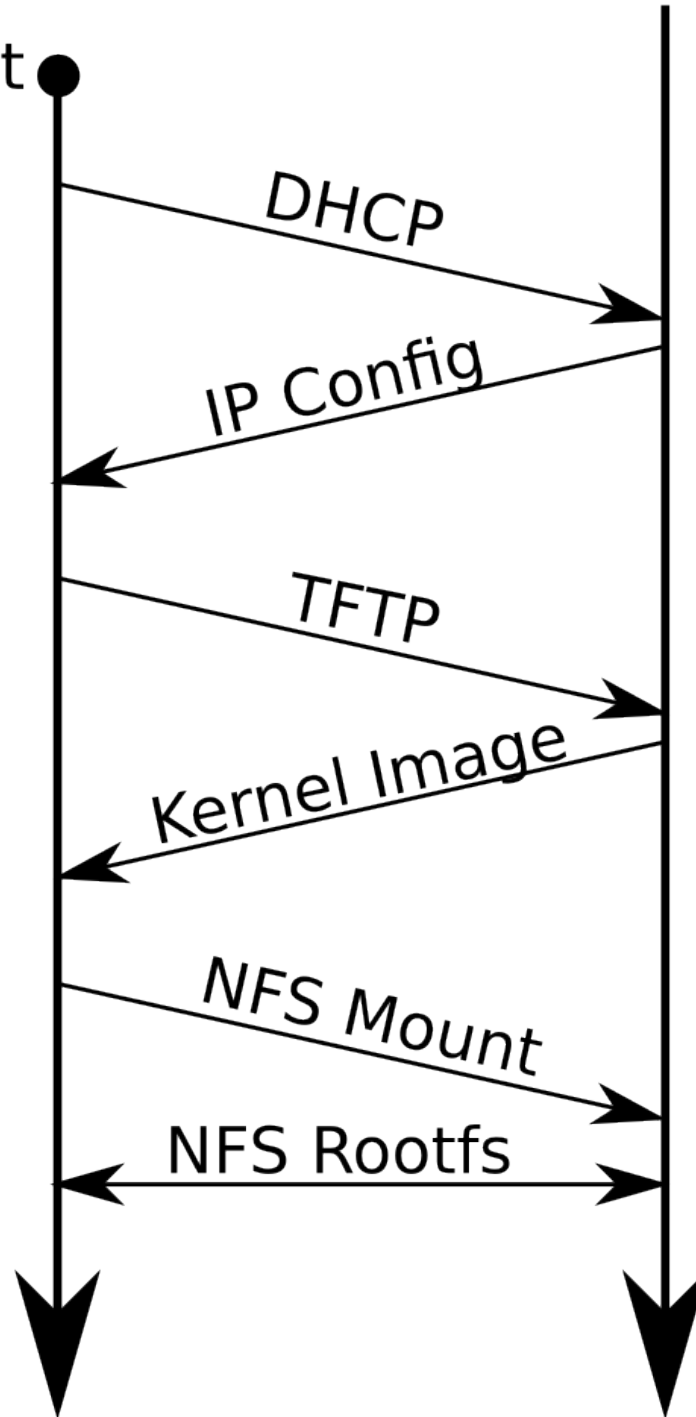
# 2.1. Host / Target setup



# Target

# Host

System Boot ●



## 2.2. IDE / Editor



# What if ... ?





## 2.3. Eclipse Setup

- Preparation
- Project importing
- AOSP fixups
- Browsing the sources

# 2.3.1. Preparation

- AOSP Basics:
  - Get AOSP ... from Google or otherwise
  - Extract if needed
  - Configure, build, etc.
- Eclipse / ADT:
  - Get ADT bundle from [developer.android.com](http://developer.android.com)
  - Extract
  - Start and update and if needed

- Set up basic classpath file:

```
[aosp]$ cp development/ide/eclipse/.classpath .
```

- Adjust eclipse.ini

- On my ADT bundle, it's:

- adt-bundle-linux-x86\_64-20130917/eclipse/eclipse.ini

- Change this:

- XX:MaxPermSize=256m

- Xms40m

- Xmx768m

- To this:

- XX:MaxPermSize=256m

- Xms128m

- Xmx1500m

## 2.3.2. Project importing

- Start Eclipse
- Create new "Java project"
  - Project name = your AOSP name
  - Deselect "Use default location"
  - Location = path to your AOSP
  - Click "Next"
  - Wait a little bit ...
  - Click "Finish"
  - Wait for it to build your project
  - ... it likely will fail ...

## 2.3.3. AOSP fixups

- Need to fix AOSP classpath file and sources
- Assuming 4.3 here
- Add this:

```
<classpathentry kind="src" path="frameworks/opt/timezonepicker/src"/>  
<classpathentry kind="src" path="frameworks/opt/colorpicker/src"/>  
<classpathentry kind="src" path="frameworks/opt/datetimetypepicker/src"/>  
<classpathentry kind="src"  
  path="frameworks/support/v8/renderscript/java/src"/>
```

- Remove this:

```
<classpathentry kind="src"  
  path="frameworks/support/renderscript/v8/java/src"/>
```

- Comment out a couple of things:

```
<!-- Redefines android.util.pools which confuses Eclipse
<classpathentry kind="src" path="packages/apps/Gallery2/src"/>
<classpathentry kind="src" path="packages/apps/Gallery2/src_pd"/>
<classpathentry kind="src"
path="packages/apps/Gallery2/gallerycommon/src"/>
-->
<!--
<classpathentry kind="src" path="packages/apps/Nfc/src"/>
<classpathentry kind="src" path="packages/apps/Nfc/nci/src"/>
-->
<!--
<classpathentry kind="src" path="frameworks/ex/carousel/java"/>
-->
```

- Manually build the following (cd to and "mm") -- or remove from .classpath:

```
packages/apps/Stk  
packages/screensavers/WebView  
development/samples/ApiDemos  
development/samples/HelloActivity  
development/samples/Home  
development/samples/LunarLander  
development/samples/NotePad  
development/samples/RSSReader  
development/samples/SkeletonApp  
development/samples/Snake
```

- Edit

packages/apps/Launcher/src/com/android/launcher2/DragLayer.java and modify:

```
private boolean isLayoutRtl() {
```

- to

```
public boolean isLayoutRtl() {
```

- **Now: right-click on project and select "Refresh"**
- It might still show "x" on some parts until it's done rebuilding the project



## 2.3.4. Browsing the sources

- Mouse-over object type to be taken to declaration
- Browse classes through “Outline”
- Browse Call Hierarchy
- View recently viewed files (Ctrl-e)
- Many other shortcuts, see:
  - <http://source.android.com/source/using-eclipse.html>
- Issues:
  - Can't compile with Eclipse ... still need “make”
  - For Java only

# 3. Observing and Monitoring

- Native
- Framework
- Overall
- Apps / Add-ons

# 3.1. Native

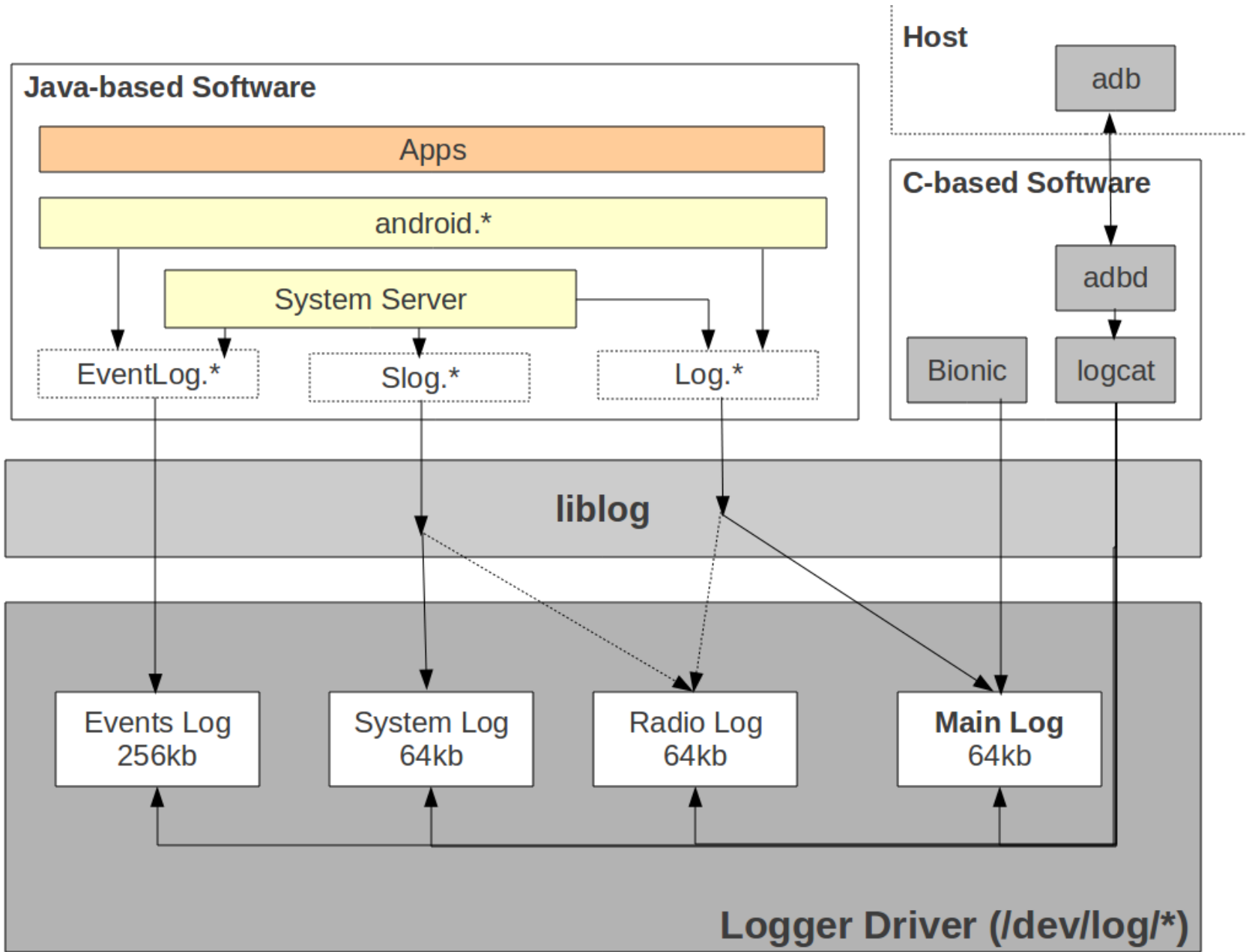
- schedtop
- librank
- procmem
- procrank
- showmap
- latencytop

## 3.2. Framework

- dumpsys
- service

# 3.3 Overall

- logcat
- dumpstate / bugreport
- watchprop / getprop



## 3.4. Apps / Add-ons

- Google Play:
  - Process Manager
  - Process Monitor
  - Task Manager
  - Process Tracker
  - ...

# 3.5. Process Explorer

The screenshot shows the Process Explorer application window. The top bar displays the current process delay (0.2s) and graph delay (0.1s). Below the process list, there are two small graphs showing CPU usage (0.00%) and memory usage (783740928.00). The main table lists various system processes, including Kernel, init, ueventd, healthd, servicemanager, vold, netd, debuggerd, rild, surfaceflinger, zygote, system\_server, and com.android.inputmethod.latin. Below the table, there is a filter section and a log of events. The log entry for PID 391 is highlighted in red, indicating an error: "NDC Command (437 bandwidth getetherstats) took too long (568ms)".

Name	PID	S	PRI	%CPU	%Mem	VSS	RSS	SHM	Time	Command line
Kernel	0			0.0%	0.0%	0 bytes	0 bytes	0 bytes	00.00	
init	1	S	20	0.0%	0.1%	640 Kb	496 Kb	276 Kb	00.04	/init
ueventd	31	S	20	0.0%	0.0%	580 Kb	304 Kb	144 Kb	00.00	/sbin/ueventd
healthd	42	S	20	0.0%	0.0%	1.39 Mb	140 Kb	108 Kb	00.00	/sbin/healthd
servicemanager	43	S	20	0.0%	0.0%	1.000 Kb	340 Kb	244 Kb	00.00	/system/bin/servicemanager
vold	44	S	20	0.0%	0.1%	4.55 Mb	1.09 Mb	792 Kb	00.00	/system/bin/vold
netd	46	S	20	0.0%	0.2%	9.54 Mb	1.24 Mb	856 Kb	00.06	/system/bin/netd
debuggerd	47	S	20	0.0%	0.3%	2.90 Mb	2.41 Mb	364 Kb	00.00	/system/bin/debuggerd
rild	48	S	20	0.0%	0.1%	5.37 Mb	860 Kb	628 Kb	00.02	/system/bin/rild
surfaceflinger	49	S	12	0.0%	0.3%	60.09 Mb	2.30 Mb	1.64 Mb	00.30	/system/bin/surfaceflinger
zygote	50	S	20	0.0%	5.2%	176.04 Mb	38.98 Mb	27.17 Mb	00.23	zygote
system_server	391	S	18	0.0%	6.2%	257.76 Mb	46.66 Mb	23.75 Mb	04.37	system_server
com.android.inputmethod.latin	520	S	20	0.0%	3.2%	194.35 Mb	23.94 Mb	10.20 Mb	00.02	com.android.inputmethod.latin

Filter: Clear Color [E] [W] [I] [D] [V]

Time	PID	Message
20:10:33.433	391	default: setPolicyDataEnable(enabled=true)
20:12:59.493	391	default: setPolicyDataEnable(enabled=true)
20:13:41.343	545	GC_FOR_ALLOC freed 511K, 18% free 3285K/3968K, paused 94ms, total 95ms
20:15:26.233	391	default: setPolicyDataEnable(enabled=true)
20:15:37.423	391	NDC Command (437 bandwidth getetherstats) took too long (568ms)
20:15:37.623	391	default: setPolicyDataEnable(enabled=true)
20:15:45.733	391	default: setPolicyDataEnable(enabled=true)



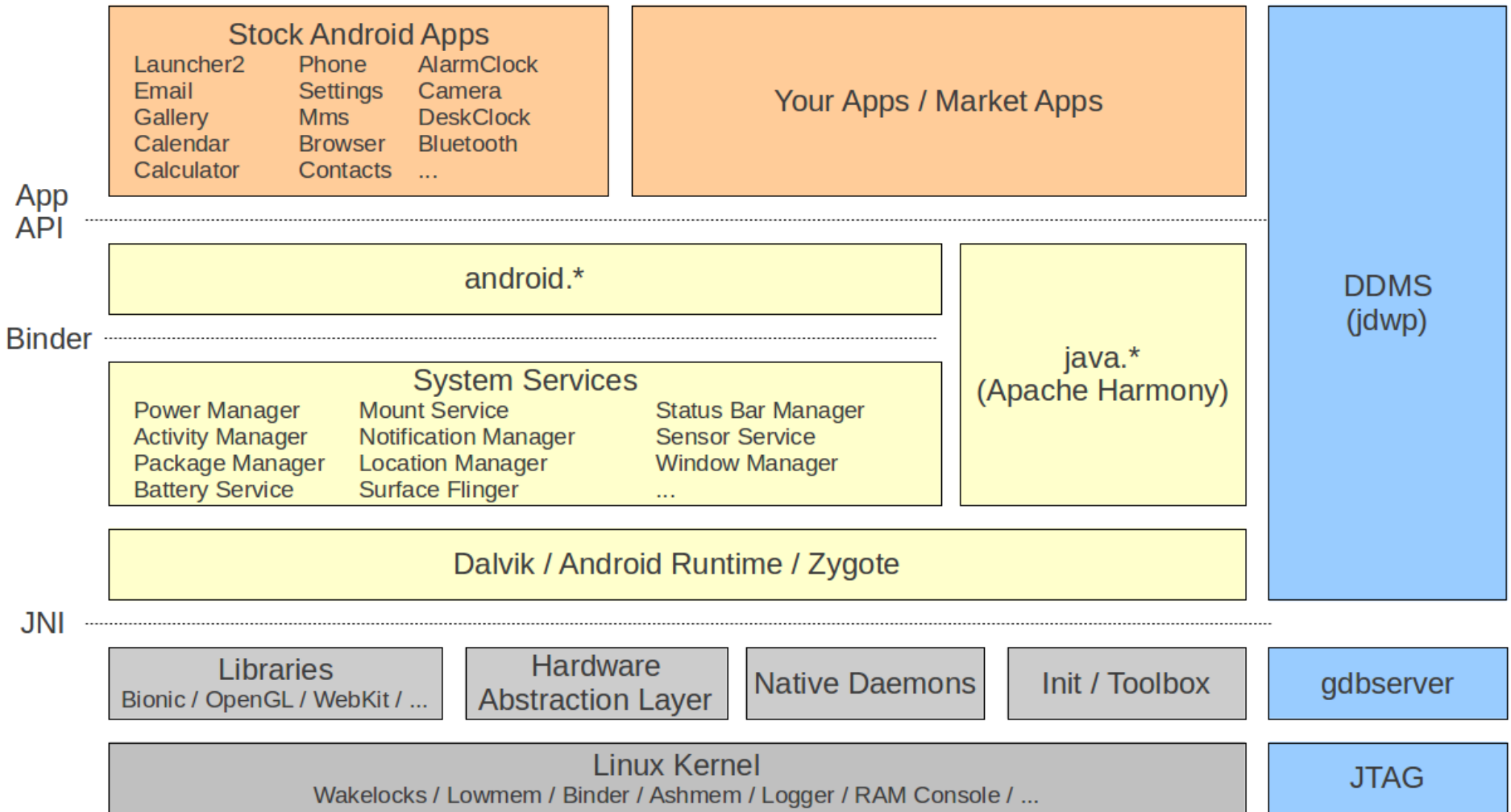
# 4. Interfacing With the Framework

- start / stop
- service call
- am
- pm
- wm
- svc
- monkey
- setprop

# 5. Working with the AOSP Sources

- You really need to check build/envsetup.sh
- Some tricks:
  - godir
  - croot
  - mm
  - m
  - jgrep
  - cgrep
  - resgrep
- It takes time to wrap your head around the tree

# 6. Symbolic Debugging - basics



# 6.1. DDMS / Eclipse integration

- Start DDMS:
  - The one from the AOSP's command-line
  - Not the one from Eclipse (“connection refuse”)
- It takes some time to load -- a few **minutes**
- Each process has a separate host-side socket
- Select the process you want to debug:
  - It'll get port 8700

- Go to Eclipse:
  - Run->Debug Configurations->Remote Java Application
  - Connection Type: "Standard (Socket Attach)"
  - Host: localhost
  - Port: 8700

## 6.2. Starting debug w/ Eclipse

- Order is finicky:
  - Start your device or emulator
  - Start command-line DDMS before Eclipse otherwise you'll get this in logcat:

```
"I/jdwp ( 411): Ignoring second debugger -- accepting and dropping"
```
  - Start Eclipse
  - Eclipse will complain that there's already a DDMS running. Ignore that.

- ☑ • StatusBarManagerService [line: 123] - expa
- ☑ • StatusBarManagerService [line: 134] - colla

ddms



Could not open Selected VM debug port (8700). Make sure you do not have another instance of DDMS or of the eclipse plugin running. If it's being used by something else, choose a new port number in the preferences.

OK

e.java

outor

icked.

```
[View v) {
```

# 6.3. Debugging

- Select the process you want to debug in DDMS
- Go into Eclipse and click on the debug configuration you created earlier
- Check that the little green bug is beside your process in DDMS
- Again, things can look like they're freezing, this is "normal" for Eclipse ...
- Wait for Eclipse to show your Dalvik process in the "Debug" \*window\* in the "Debug" \*view\* -- all threads should show



Dalvik Debug Monitor

File Edit Actions Device

Name

<build> [emulator-5554]

- system\_process
- com.android.providers.calendar
- com.android.smspush
- com.android.inputmethod.latin
- com.android.phone
- com.android.musicfx
- com.android.launcher
- android.process.media
- com.android.systemui
- com.android.mms

Saved Filters + -

All messages (no filters)

### Dalvik Debug Monitor

File Edit Actions Device

Name	State	Memory	Private Memory
<build> [emulator-5554]	Online		<build> [4.3, debug]
system_process	275	8600 / 8700	
com.android.providers.calendar	609	8601	
com.android.smspush	444	8602	
com.android.inputmethod.latin	371	8603	
com.android.phone	396	8604	
com.android.musicfx	814	8605	
com.android.launcher	407	8606	
android.process.media	498	8607	
com.android.systemui	344	8608	
com.android.mms	672	8609	

## 6.4. Debugging multiple processes

- In the debug *\*view\** of eclipse, click on "Debug" for every time you change the process in DDMS
- Wait for that process' threads to load in the debug view
- Once threads are loaded, you can actually start debugging

# 6.5. gdbserver - target side

- AOSP already takes care of debug:
  - “-g” flag added to all native binaries
  - Unstripped binaries in out/target/product/.../symbols/...
- Attaching to running process

```
# gdbserver --attach localhost:2345 30
```
- Start app for debugging with gdbserver prepended

```
# gdbserver localhost:2345 service list
```
- Forward the port on the host:

```
$ adb forward tcp:2345 tcp:2345
```

# 6.6. gdb - host side

- Load file **\*\*FIRST\*\*** and then attach on host side

```
$ prebuilts/gcc/linux-x86/arm/arm-eabi-4.7/bin/arm-eabi-gdb
GNU gdb (GDB) 7.3.1-gg2
Copyright (C) 2011 Free Software Foundation, Inc.
...
(gdb) file out/target/product/generic/symbols/system/bin/service
(gdb) target remote localhost:2345
(gdb) b main
Cannot access memory at address 0x0
Breakpoint 1 at 0x2a00146c: file frameworks/native/cmds/service/service.cpp, line 59.
(gdb) cont
Continuing.
warning: Could not load shared library symbols for 11 libraries, e.g. /system/bin/linker.
...

Breakpoint 1, main (argc=2, argv=0xbe882b74) at frameworks/native/cmds/service/service.cpp:59
59 {
(gdb) n
60     sp<IServiceManager> sm = defaultServiceManager();
(gdb) n
59 {
(gdb) n
60     sp<IServiceManager> sm = defaultServiceManager();
(gdb) n
61     fflush(stdout);
```

# 6.7. JNI debugging

```
$ prebuilts/gcc/linux-x86/arm/arm-eabi-4.7/bin/arm-eabi-gdb
GNU gdb (GDB) 7.3.1-gg2
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=x86_64-linux-gnu --target=arm-linux-
android".
For bug reporting instructions, please see:
.
(gdb) target remote localhost:2345
Remote debugging using localhost:2345
0xb6f125cc in ?? ()
(gdb) set solib-search-path out/target/product/generic/symbols/system/lib/
warning: Could not load shared library symbols for 8 libraries, e.g.
/system/bin/linker.
Use the "info sharedlibrary" command to see the complete listing.
Do you need "set solib-search-path" or "set sysroot"?
```

```
(gdb) b com_android_server_power_PowerManagerService.cpp:171
Breakpoint 1 at 0xaa107604: file
frameworks/base/services/jni/com_android_server_power_PowerManagerService.
cpp, line 171.
(gdb) cont
Continuing.
^C
Program received signal SIGINT, Interrupt.
epoll_wait () at bionic/libc/arch-arm/syscalls/epoll_wait.S:10
10      mov      r7, ip
(gdb) b com_android_server_power_PowerManagerService.cpp:161
Breakpoint 2 at 0xaa1077ba: file
frameworks/base/services/jni/com_android_server_power_PowerManagerService.
cpp, line 161.
...
```

# 6.8. JTAG

- Requires hardware device
- Sometimes interfaces with gdb
- Not Android specific
- Some allow transparent kernel/user-space debug
- Don't know of any that go all the way up to Dalvik

# 7. Detailed Dynamic Data Collection

- Logging
- strace
- ftrace
- perf



# 7.1. Logging

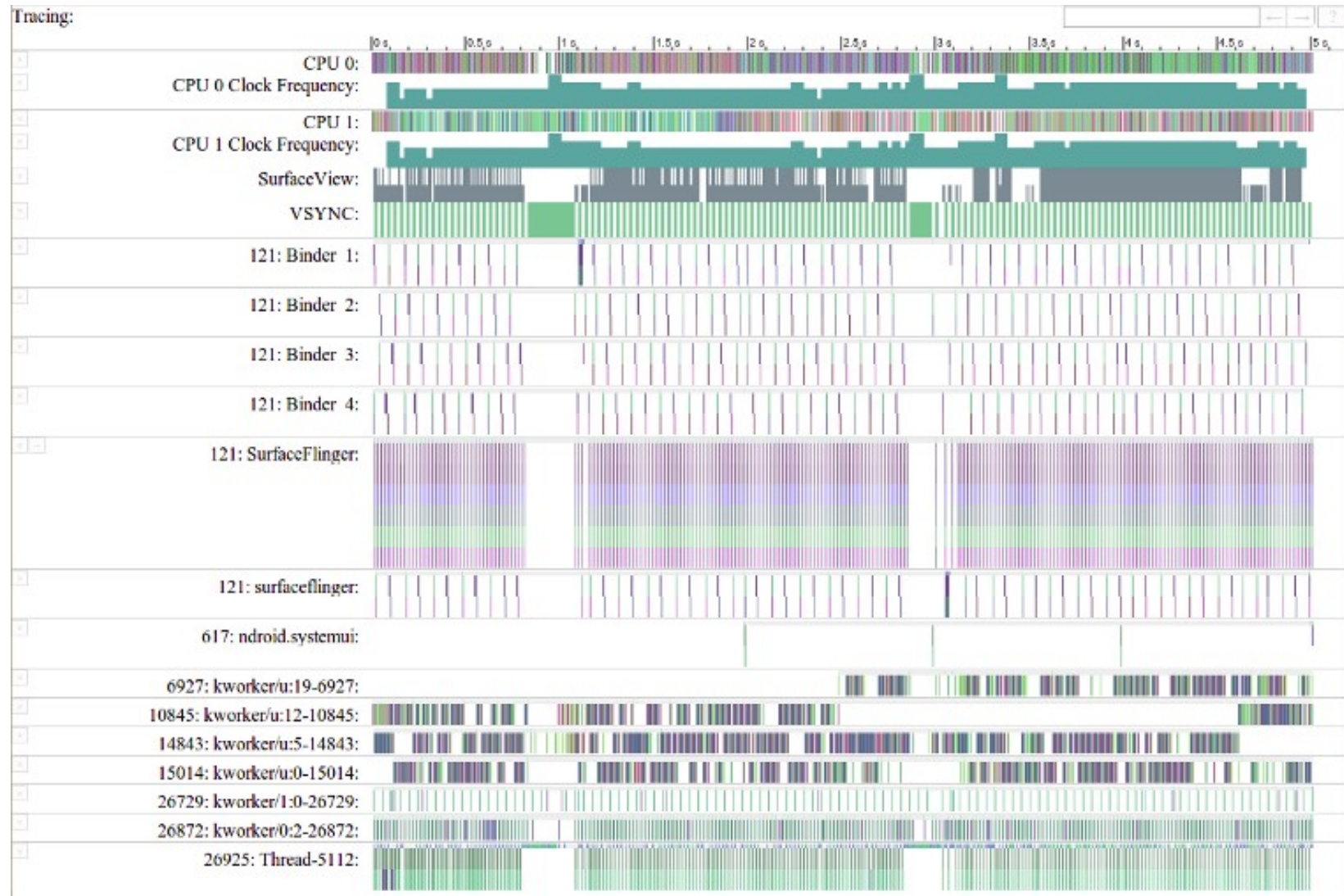
- logcat is the most rapid/consistent way to observe dynamic behavior.
- Trivial to add instrumentation points
- **It just works ...**

## 7.2. strace

- Same as Linux
- Use man page if need be

# 7.3. ftrace

- With 4.1, Google introduced systrace/atrace
- systrace is a Python script running on host side
- atrace is native Android binary
- systrace calls atrace via ADB
- atrace uses ftrace to capture kernel events
- Stack instrumented to feed events to ftrace
- Google's doc:
  - <https://developer.android.com/tools/help/systrace.html>
  - <https://developer.android.com/tools/debugging/systrace.html>



# ... trouble is ...

- Finicky -- notes from my attempts with 4.3:
  - I can't get it to work !\*!@#\$&!#\*\$!
  - Default goldfish kernel doesn't have ftrace
  - Able to build ftrace-enabled kernel for goldfish
  - Can trace that system ... so long as I **don't use** atrace/systrace ... WTF<sup>1</sup>?
- Not all Android kernels have ftrace enabled
- Generates HTML file that can only be read by Chrome ... **it doesn't work in Firefox**. NIH?

1: The AOSP sources define WTF as “What a Terrible Failure”. We trust they've done their research.

# ... still ...

- Have a look at these files:
  - `/external/chromium-trace/systrace.py`
  - `/frameworks/native/cmds/atrace`
  - `/frameworks/base/core/java/android/os/Trace.java`
  - `/erameworks/native/include/utils/Trace.h`
  - `/system/core/include/cutils/trace.h`
  - `/frameworks/native/libs/utils/Trace.cpp`
- Look for:
  - `ATRACE*` in c/cpp files
  - `Trace.traceBegin()/trace.traceEnd()` in Java files

**# atrace --help**

usage: atrace [options] [categories...]

options include:

- a appname enable app-level tracing for a comma separated list of cmdlines
- b N use a trace buffer size of N KB
- c trace into a circular buffer
- k fname,... trace the listed kernel functions
- n ignore signals
- s N sleep for N seconds before tracing [default 0]
- t N trace for N seconds [default 5]
- z compress the trace dump
- async\_start start circular trace and return immediatly
- async\_dump dump the current contents of circular trace buffer
- async\_stop stop tracing and dump the current contents of circular trace buffer
- list\_categories list the available tracing categories

```
# atrace --list_categories
    gfx - Graphics
    input - Input
    view - View System
webview - WebView
    wm - Window Manager
    am - Activity Manager
    audio - Audio
    video - Video
camera - Camera
    hal - Hardware Modules
    res - Resource Loading
dalvik - Dalvik VM
```



## 7.3. perf on Android on ARM



## 8. Benchmarking

**WARNING**

**"Whitelisting" benchmarking  
tools in your product  
will result in  
mainstream media coverage**

Oxbench  
AnTuTu  
Passmark  
Vellamo  
Geekbench2  
SunSpider  
GLBenchmakr  
Quadrant Standard Edition  
Linpack  
Neocore  
3DMark  
Epic Citadel  
Androbench  
CF-bench  
SD Tools

RL Benchmark: SQL  
Benchmark & Tunning  
A1 SD Bench  
Quick Benchmark Lite  
3DRating benchmark  
Smartbench 2011  
NenaMark  
Rightware Browsermark  
An3DBenchXL  
CaffeineMark  
NBench  
Methanol  
AndEBench  
SmartBench 2012  
RealPi

# 9. Summing Up



- Works relatively well:
  - logcat
  - Eclipse / DDMS
  - Framework tools
- Works ok:
  - gdb/gdbserver
  - native tools
  - ftrace
- Finicky:
  - systrace/atrace
  - perf

# 10. Loose ends

- debuggerd
- tombstones
- anr traces

Thank you ...

[karim.yaghmour@opersys.com](mailto:karim.yaghmour@opersys.com)

