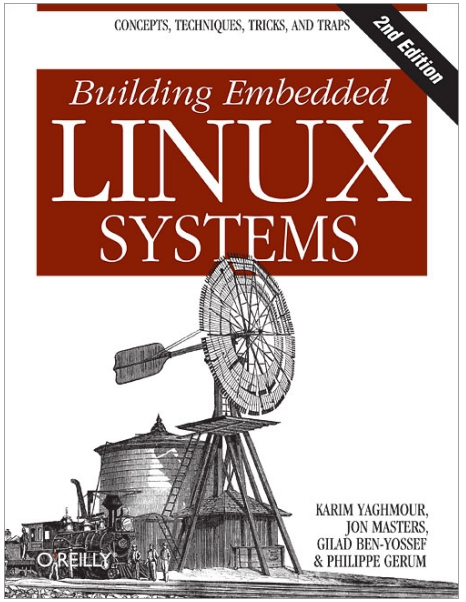
Android Internals

Android Builders Summit – April 13th 2011

Karim Yaghmour
karim.yaghmour@opersys.com
@karimyaghmour

About ...

- Author of:



- Introduced Linux Trace Toolkit in 1999
- Originated Adeos and relayfs (kernel/relay.c)

1. Android Concepts
2. Overall Architecture
3. System startup
4. Linux Kernel
5. Hardware Support
6. Native User-Space
7. Dalvik
8. JNI
9. System Server
10. Activity Manager
11. Binder
12. Stock AOSP Apps
13. Hacking

1. Android Concepts

- Components
- Intents
- Component lifecycle
- Manifest file
- Processes and threads
- Remote procedure calls

1.1. Components

- 1 App = N Components
- Apps can use components of other applications
- App processes are automagically started whenever any part is needed
- Ergo: N entry points, !1, and !main()
- Components:
  - Activities
  - Services
  - Broadcast Receivers
  - Content Providers

1.2. Intents

- Intent = asynchronous message w/ or w/o designated target
- Like a polymorphic Unix signal, but w/o required target
- Intents “payload” held in Intent Object
- Intent Filters specified in Manifest file

1.3. Component lifecycle

- System automagically starts/stops/kills processes:
  - Entire system behaviour predicated on low memory
- System triggers Lifecycle callbacks when relevant
- Ergo: Must manage Component Lifecycle
- Some Components are more complex to manage than others

1.4. Manifest file

- Informs system about app’s components
- XML format
- Always called AndroidManifest.xml
- Activity = <activity> ... static
- Service = <service> ... static
- Broadcast Receiver:
  - Static = <receiver>
  - Dynamic = Context.registerReceiver()
- Content Provider = <provider> ... static

1.5. Processes and threads

- Processes
  - Default: all callbacks to any app Component are issued to the main process thread
  - <activity>—<service>—<recipient>—<provider> have process attribute to override default
  - Do NOT perform blocking/long operations in main process thread:
    - Spawn threads instead
  - Process termination/restart is at system's discretion
  - Therefore:
    - Must manage Component Lifecycle
- Threads:
  - Create using the regular Java Thread Object
  - Android API provides thread helper classes:
    - Looper: for running a message loop with a thread
    - Handler: for processing messages
    - HandlerThread: for setting up a thread with a message loop

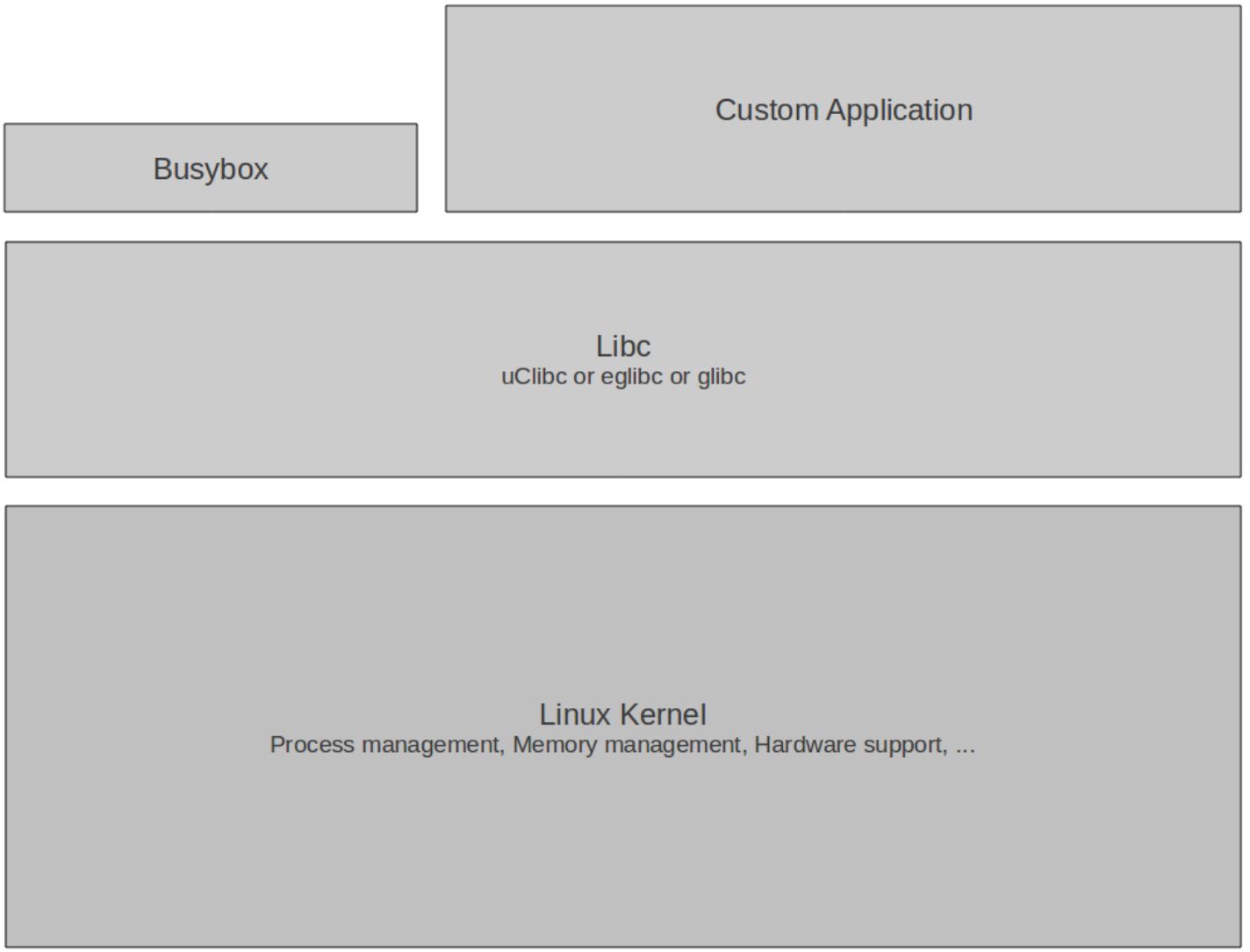
1.6. Remote procedure calls

- Apparently System V IPC is evil ...
- Android RPCs = Binder mechanism
- Binder is a low-level functionality, not used as-is
- Instead: must define interface using Interface Definition Language (IDL)
- IDL fed to aidl Tool to generate Java interface definitions

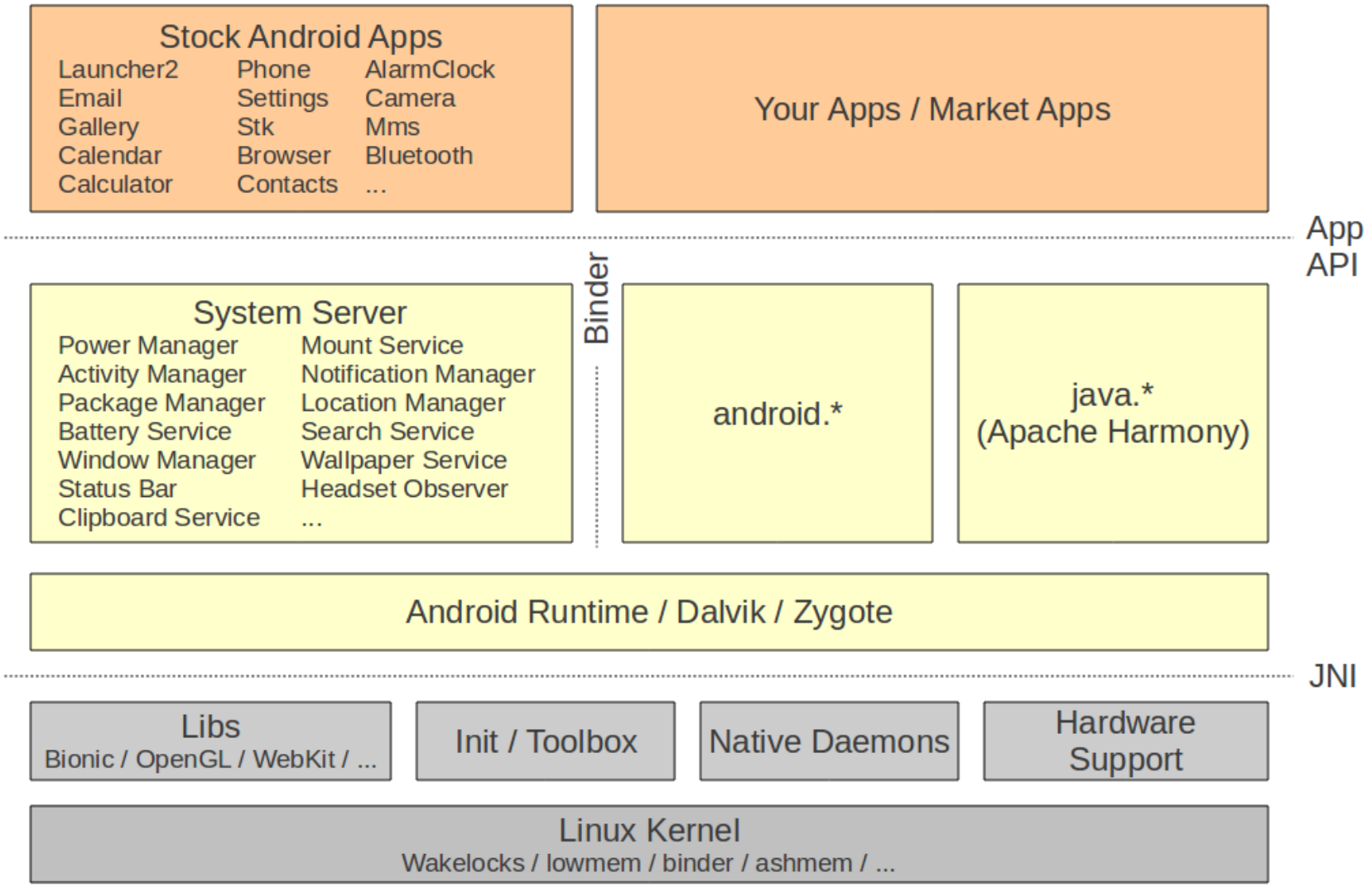
1.7. Development tools

- SDK:
  - android – manage AVDs and SDK components
  - apkbuilder – creating .apk packages
  - dx – converting .jar to .dex
  - adb – debug bridge
  - emulator – QEMU-based ARM emulator
  - ...
- Eclipse w/ ADT plugin
- NDK: GNU toolchain for native binaries

2.1. Overall Architecture - EL



2.2. Overall Architecture - Android



3. System Startup

- Bootloader
- Kernel
- Init
- Zygote
- System Server
- Activity Manager
- Launcher (Home)

3.1. Bootloader

- aosp/bootable/bootloader
  - Custom bootloader for Android
  - USB-based
  - Implements the “fastboot” protocol
  - Controlled via “fastboot” cli tool on host
- aosp/bootable/recovery
  - UI-based recovery boot program
  - Accessed through magic key sequence at boot
  - Usually manufacturer specific variant

- Flash layout:

```
0x000003860000-0x000003900000 : "misc"
0x000003900000-0x00003e00000 : "recovery"
0x00003e00000-0x000004300000 : "boot"
0x000004300000-0x00000c300000 : "system"
0x00000c300000-0x0000183c0000 : "userdata"
0x0000183c0000-0x00001dd20000 : "cache"
0x00001dd20000-0x00001df20000 : "kpanic"
0x00001df20000-0x00001df60000 : "dinfo"
0x00001df60000-0x00001dfc0000 : "setupdata"
0x00001dfc0000-0x00001e040000 : "splash1"
0x000000300000-0x000001680000 : "modem"
```

Kernel
/system
/data
/cache

From Acer Liquid-E

3.2. Kernel

- Early startup code is very hardware dependent
- Initializes environment for the running of C code
- Jumps to the architecture-independent start_kernel() function.
- Initializes high-level kernel subsystems
- Mounts root filesystem
- Starts the init process

3.3. Android Init

- Open, parses, and runs /init.rc:
  - Create mountpoints and mount filesystems
  - Set up filesystem permissions
  - Set OOM adjustments properties
  - Start daemons:
    - adbd
    - servicemanager (binder context manager)
    - vold
    - netd
    - rild
    - app_process -Xzygote (Zygote)
    - mediaserver
    - ...

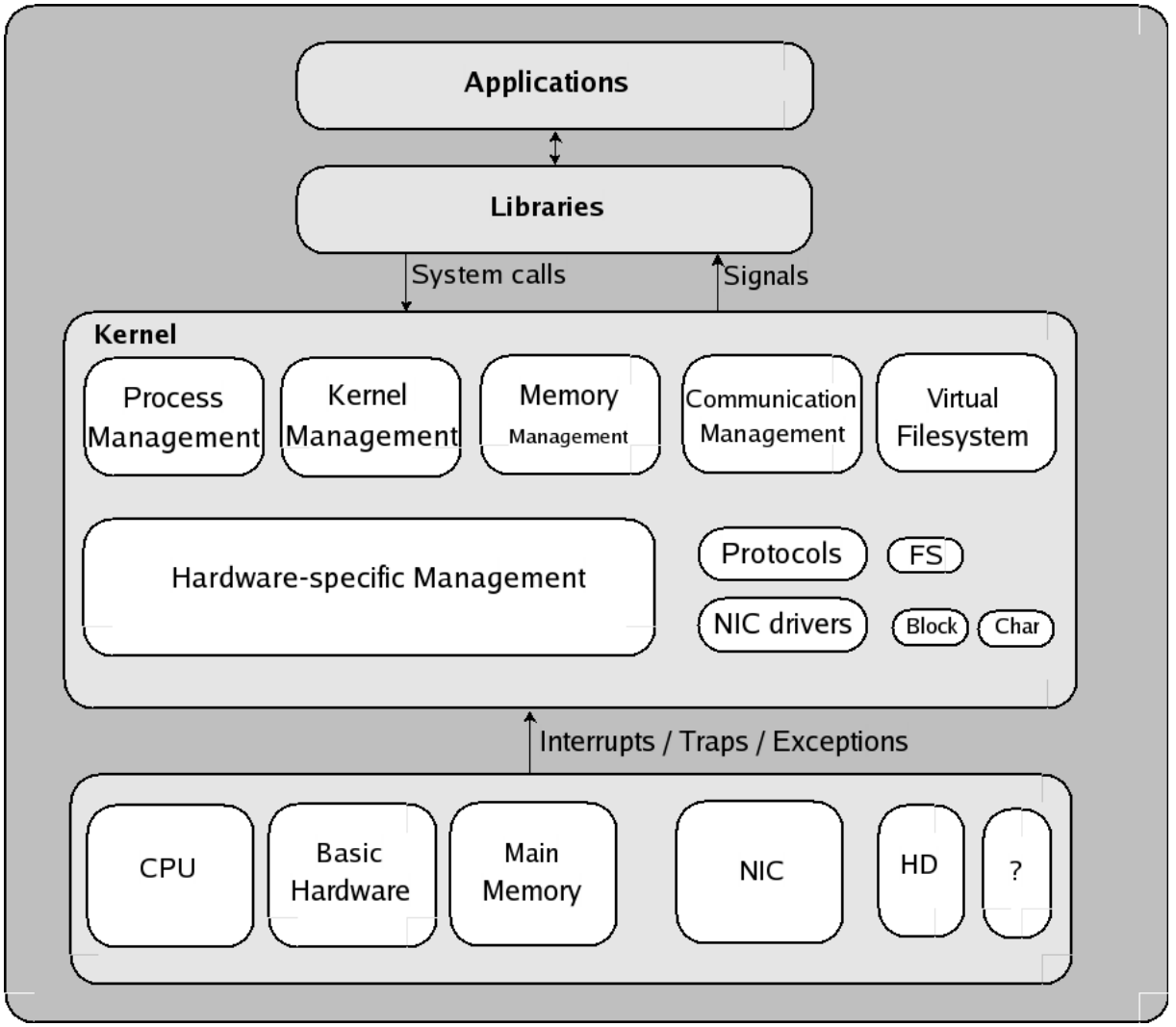
3.4. Zygote, etc.

- Init:
  - app_process -Xzygote (Zygote)
- frameworks/base/cmds/app_process/app_main.cpp:
  - runtime.start(“com.android.internal.os.Zygote”, ...
- frameworks/base/core/jni/AndroidRuntime.cpp:
  - startVM()
  - Call Zygote's main()
- frameworks/base/core/java/com/android/internal/os/ZygoteInit.java:
  - ...

- preloadClasses()
- startSystemServer()
- ... magic ...
- Call SystemServer's run()

- frameworks/base/services/java/com/android/server/SystemServer.java:
  - Start **all** system services/managers
  - Start ActivityManager:
    - Send Intent.CATEGORY_HOME
    - Launcher2 kicks in

4. Linux Kernel



4.1. Androidisms

- Wakelocks
- lowmem handler
- Binder
- ashmem – Anonymous Shared Memory
- RAM console
- Logger
- ...

5. Hardware support

Bluetooth	BlueZ through D-BUS IPC (to avoid GPL contamination it seems)
GPS	Manufacturer-provided libgps.so
Wifi	wpa_supplicant
Display	Std framebuffer driver (/dev/fb0)
Keymaps and Keyboards	Std input event (/dev/event0)
Lights	Manufacturer-provided liblights.so
Backlight	
Keyboard	
Buttons	
Battery	
Notifications	
Attention	
Audio	Manufacturer-provided libaudio.so (could use ALSA underneath ... at least as illustrated in their porting guide)
Camera	Manufacturer-provided libcamera.so (could use V4L2 kernel driver underneath ... as illustrated in porting guide)
Power Management	“Wakelocks” kernel patch
Sensors	Manufacturer-provided libsensors.so
Accelerometer	
Magnetic Field	
Orientation	
Gyroscope	
Light	
Pressure	
Temperature	
Proximity	
Radio Layer Interface	Manufacturer-provided libril-<companyname>-<RIL version>.so

6. Native User-Space

- Mainly
  - /data => User data
  - /system => System components
- Also found:
  - /cache
  - /mnt
  - /sbin
  - Etc.

- Libs:

  Bionic, SQLite, SSL, OpenGL|ES,

  Non-Posix: limited Pthreads support, no SysV IPC
- Toolbox
- Daemons:

  servicemanager, vold, rild, netd, adbd, ...

7. Dalvik

- Sun-Java =

  Java language + JVM + JDK libs
- Android Java =

  Java language + Dalvik + Apache Harmony
- Target:
  - Slow CPU
  - Relatively low RAM
  - OS without swap space
  - Battery powered
- Now has JIT

7.1. Dalvik's .dex files

- JVM munches on “.class” files
- Dalvik munches on “.dex” files
- .dex file = .class files post-processed by “dx” utility
- Uncompressed .dex = 0.5 * Uncompressed .jar

8. JNI – Java Native Interface

- Call gate for other languages, such as C, C++
- Equivalent to .NET's pinvoke
- Usage: include and call native code from App
- Tools = NDK ... samples included
- Check out *“JNI Programmer's Guide and Specification”* - freely available PDF

9. System Server

Entropy Service
Power Manager
Activity Manager
Telephone Registry
Package Manager
Account Manager
Content Manager
System Content Providers
Battery Service
Lights Service
Vibrator Service
Alarm Manager
Init Watchdog
Sensor Service
Window Manager
Bluetooth Service

Device Policy
Status Bar
Clipboard Service
Input Method Service
NetStat Service
NetworkManagement Service
Connectivity Service
Throttle Service
Accessibility Manager
Mount Service
Notification Manager
Device Storage Monitor
Location Manager
Search Service
DropBox Service
Wallpaper Service

Audio Service
Headset Observer
Dock Observer
UI Mode Manager Service
Backup Service
AppWidget Service
Recognition Service
Status Bar Icons
DiskStats Service
ADB Settings Observer

9.1. Some stats

- frameworks/base/services/java/com/android/server:
  - 3.5 M
  - ~100 files
  - 85 kloc
- Activity manager:
  - 920K
  - 30+ files
  - 20 kloc

9.2. Observing with “logcat”

- Find the System Server's PID

  $ adb shell ps | grep system_server

  system 63 32 120160 35408 ffffffff afd0c738 S system_server

- Look for its output:

  $ adb logcat | grep “63)”

```
...
D/PowerManagerService(   63): bootCompleted
I/TelephonyRegistry(   63): notifyServiceState: 0 home Android Android 310260  UMTS CSS not supp...
I/TelephonyRegistry(   63): notifyDataConnection: state=0 isDataConnectivityPossible=false reason=null interfaceName=null
networkType=3
I/SearchManagerService(   63): Building list of searchable activities
I/WifiService(   63): WifiService trying to setNumAllowed to 11 with persist set to true
I/ActivityManager(   63): Config changed: { scale=1.0 imsi=310/260 loc=en_US touch=3 keys=2/1/2 nav=3/1 ...
I/TelephonyRegistry(   63): notifyMessageWaitingChanged: false
I/TelephonyRegistry(   63): notifyCallForwardingChanged: false
I/TelephonyRegistry(   63): notifyDataConnection: state=1 isDataConnectivityPossible=true reason=simL...
I/TelephonyRegistry(   63): notifyDataConnection: state=2 isDataConnectivityPossible=true reason=simL...
D/Tethering(   63): MasterInitialState.processMessage what=3
I/ActivityManager(   63): Start proc android.process.media for broadcast com.android.providers.downloads/.DownloadReceiver:
pid=223 uid=10002 gids={1015, 2001, 3003}
I/RecoverySystem(   63): No recovery log file
W/WindowManager(   63): App freeze timeout expired.
...
```

9.3. Snapshot with “dumpsys”

```
Currently running services:
  SurfaceFlinger
  accessibility
  account
  activity
  alarm
  appwidget
  audio
  backup
...
  wifi
  window
--------------------------------------------------------------------------------
DUMP OF SERVICE SurfaceFlinger:
+ Layer 0x396b90
      z=    21000, pos=(   0,   0), size=( 480, 800), needsBlending=1, needsDithering=1, invalidat ...
0]
      name=com.android.launcher/com.android.launcher2.Launcher
      client=0x391e48, identity=6
      [ head= 1, available= 2, queued= 0 ] reallocMask=00000000, inUse=-1, identity=6, status=0
      format= 1, [480x800:480] [480x800:480], freezeLock=0x0, dq-q-time=53756 us
...
```

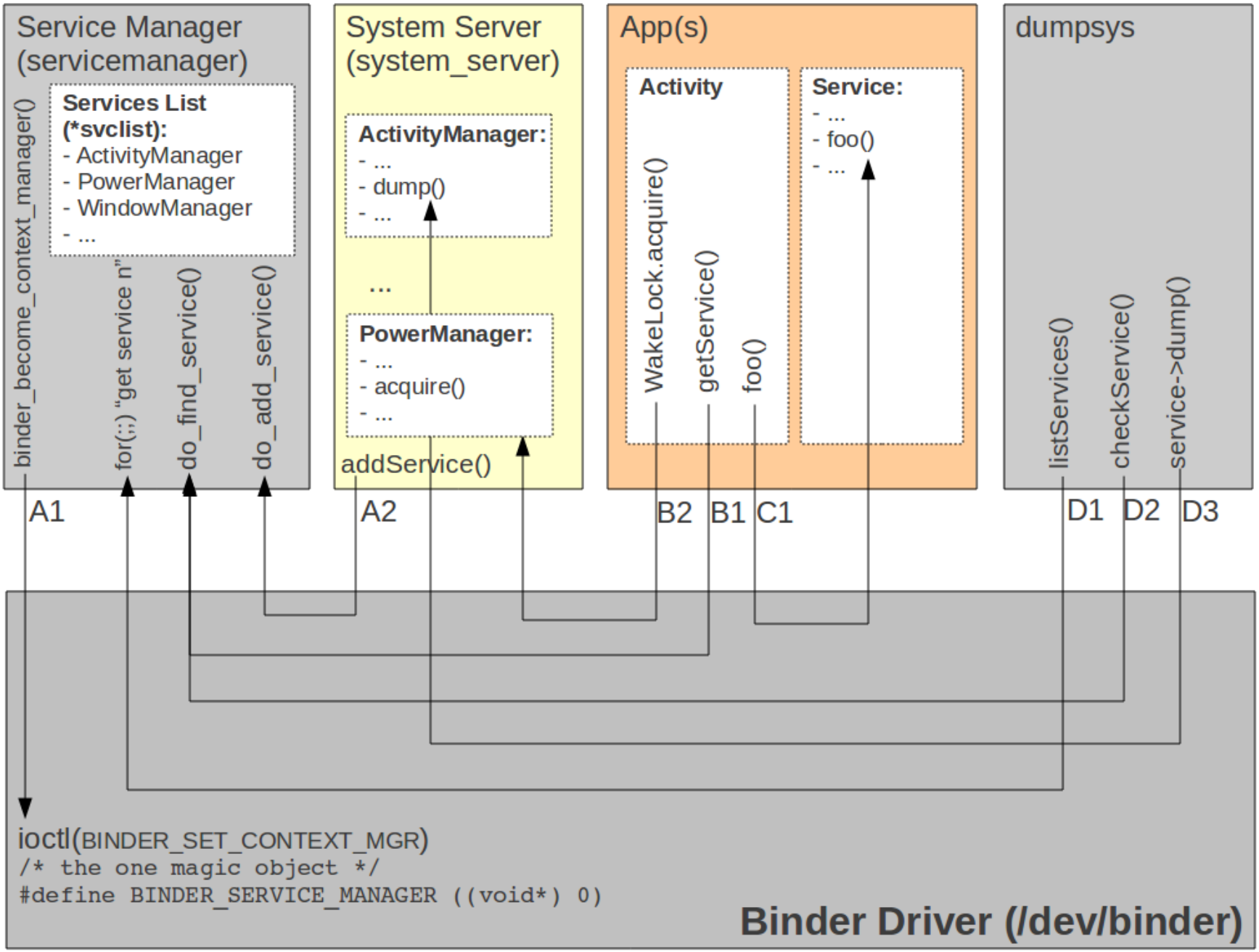
10. ActivityManager

- Start new Activities, Services
- Fetch Content Providers
- Intent broadcasting
- OOM adj. maintenance
- Application Not Responding
- Permissions
- Task management
- Lifecycle management

- Ex. starting new app from Launcher:
  - onClick(Launcher)
  - startActivity(Activity.java)
  - <*Binder*>
  - ActivityManagerService
  - startViaZygote(Process.java)
  - <*Socket*>
  - Zygote

11. Binder

- CORBA/COM-like IPC
- Data sent through “parcels” in “transactions”
- Kernel-supported mechanism
- /dev/binder
- Check /proc/binder/*
- android.* API connected to System Server through binder.


Service Manager
(servicemanager)
Services List
(*svclist):
- ActivityManager
- PowerManager
- WindowManager
- ...
binder_become_context_manager()
for(;;) "get service n"
do_find_service()
do_add_service()
System Server
(system_server)
ActivityManager:
- ...
- dump()
- ...
...
PowerManager:
- ...
- acquire()
- ...
addService()
App(s)
Activity
WakeLock.acquire()
getService()
foo()
Service:
- ...
- foo()
- ...
dumpsys
listServices()
checkService()
service->dump()
A1
A2
B2
B1
C1
D1
D2
D3
ioctl(BINDER_SET_CONTEXT_MGR)
/* the one magic object */
#define BINDER_SERVICE_MANAGER ((void*) 0)
Binder Driver (/dev/binder)

12. Stock AOSP Apps

/packages/apps

AccountsAndSettings
AlarmClock
Bluetooth
Browser
Calculator
Calendar
Camera
CertInstaller
Contacts
DeskClock
Email
Gallery
HTMLViewer
Launcher2
Mms
Music
PackageInstaller
Protips
Provision
QuickSearchBox
Settings
SoundRecorder
SpeechRecorder
Stk
VoiceDialer

/packages/providers

ApplicationProvider
CalendarProvider
ContactsProvider
DownloadProvider
DrmProvider
GoogleContactsProvider
MediaProvider
TelephonyProvider
UserDictionaryProvider

/packages/inputmethods

LatinIME
OpenWnn
PinyinIME

13. Hacking

- Source:
  - AOSP – source.android.com / android.git.kernel.org
  - Cyanogenmod – www.cyanogenmod.com
  - xdadevelopers – www.xda-developers.com
- Tools:
  - repo / git
  - fastboot
  - recovery
  - Kernel privilege escalation exploits -- “one-click root”
  - ...

Thank you ...

karim.yaghmour@opersys.com