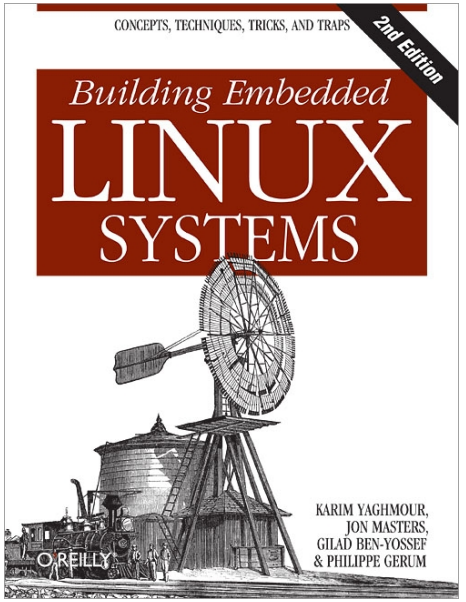
Understanding the Android System Server

AnDevCon – March 9th 2011

Karim Yaghmour / @karimyaghmour

About ...

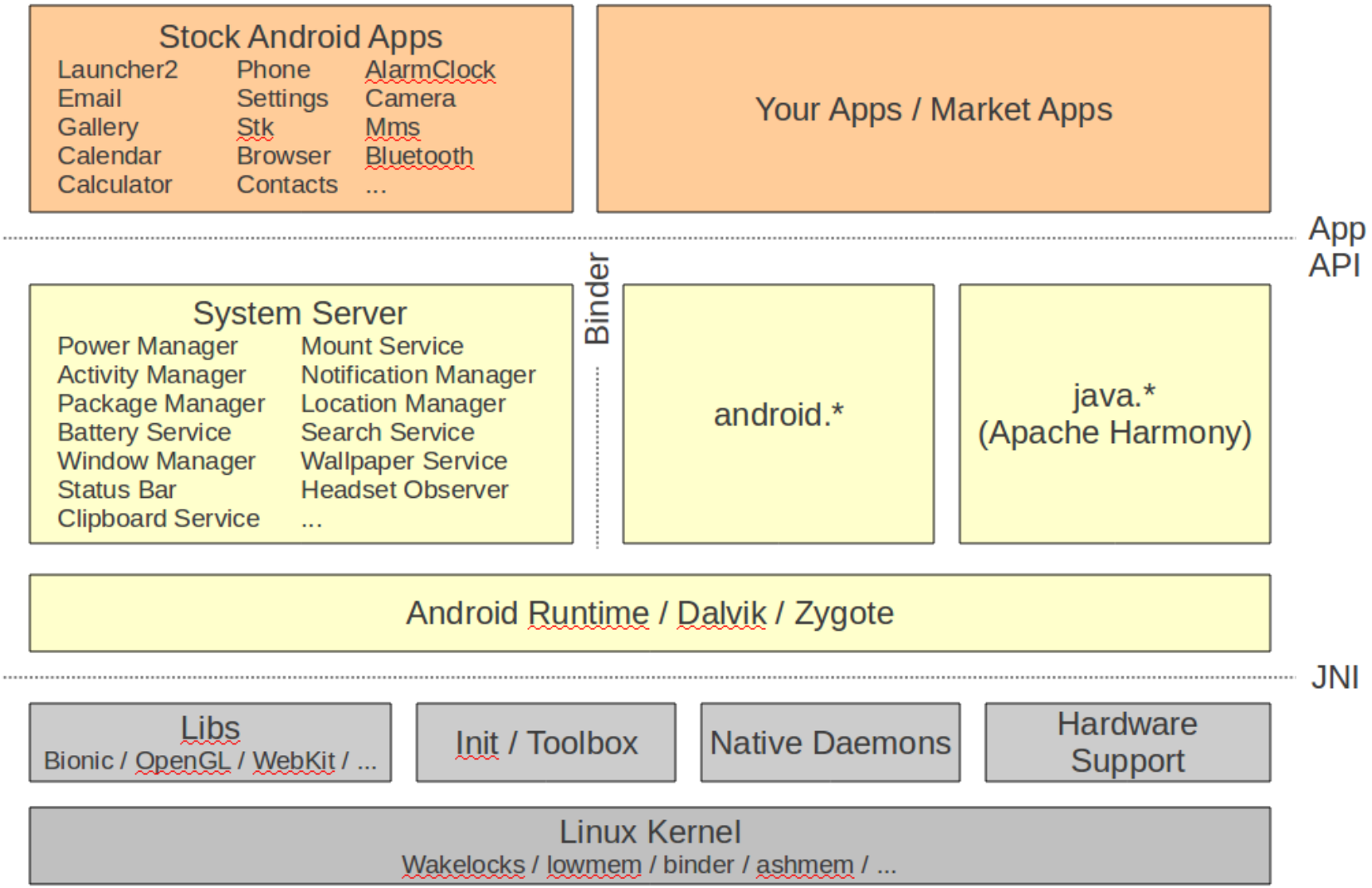
- Author of:



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

1. Architecture recap
2. Bootup sequence
3. Services run by the System Server
4. Observing the System Server in action
5. Binder
6. Calling on System Services
7. Inside a few System Services
8. Creating your own System Service

1. Architecture recap



2. Bootup sequence

- 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

3. Services run by the 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

3.1. Some stats

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

4. Observing the System Server in action

- 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.
...
```

5. Binder

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

6. Calling on System Services

- Use getSystemServer
- Ex: NotificationManager Object reference:

```
String ns = Context.NOTIFICATION_SERVICE;

NotificationManager mNotificationManager = (NotificationManager)
getSystemServService(ns);
```

- Prepare your content
- Call on the object:

```
mNotificationManager.notify(HELLO_ID, notification);
```

7. Inside a few System Services

- Get the AOSP ... repo, etc.
- Tricks:
  - Import into Eclipse and collapse methods
  - Use reverse-engineering tools:
    - Imagix
    - Rationale
    - Lattix
    - Scitools
    - ...
- Be patient, this isn't documented anywhere ...

7.1. 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

7.2. Package Manager

- 10 kloc
- 450 K
- Installation / removal
- Permissions
- Intent resolution (also IntentResolver.java)
- Called by Activity Manager

7.3. Window Manager

- Main thread
- Window manipulation
- Wallpaper handling
- Orientation
- Focus
- Layering
- Input event management

7.4. Notification Manager

- Toasts
- Notifications
- Sound playback (see NotificationPlayer.java)

7.5. Power Manager

- Wakelocks
- Sleep
- Brightness
- Lock

7.6. Network Management Service

- Talks to “netd” /system/netd
- Interface configuration
- Tethering
- DNS

7.7. Mount Service

- Mount / Unmount
- Format
- USB mass storage
- OBB

7.8. Location Manager

- Manage location providers
- getBestProvider()
- Proximity alerts
- Last known location

7.9. Status Bar Manager

- Expand / collapse
- Icon visibility
- Reveal callbacks
- Callbacks for notification manager

7.10. Backup Manager

- Enable / disable
- Transport management
- backupNow()
- ...

8. Creating your own System Service

- Have new funky hardware?
- Add your code to: frameworks/base/services/java/com/android/server/
- Have the SystemServer.java init your service
- Expose through:
  - frameworks/base/core/java/android/os/[server].aidl
- Call on native “driver” code through JNI
- Create an app that calls on service
- May need to generate custom SDK ...

Thank you ...

karim.yaghmour@opersys.com

www.opersys.com