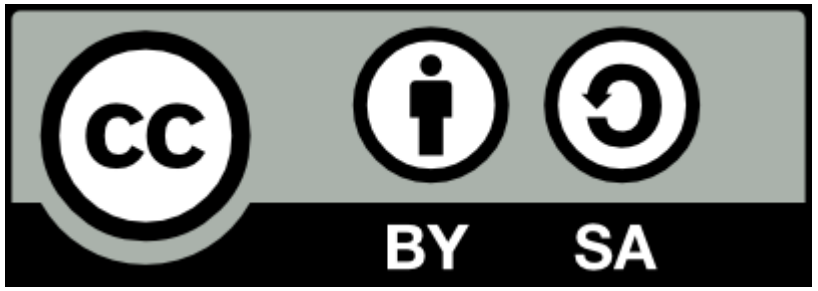
Headless Android

Android Builders Summit 2012

Karim Yaghmour
@karimyaghmour

karim.yaghmour@opersys.com



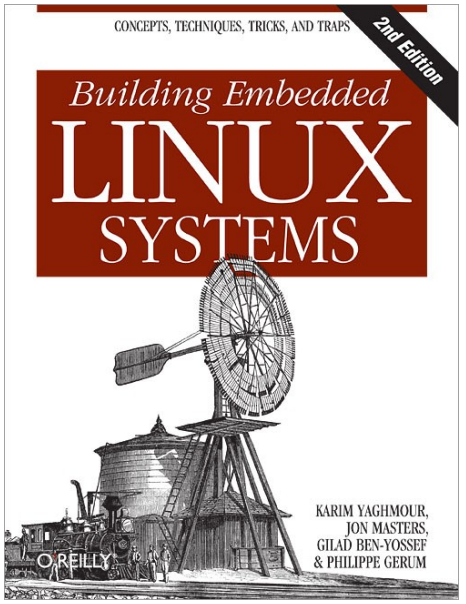
These slides created by: Karim Yaghmour

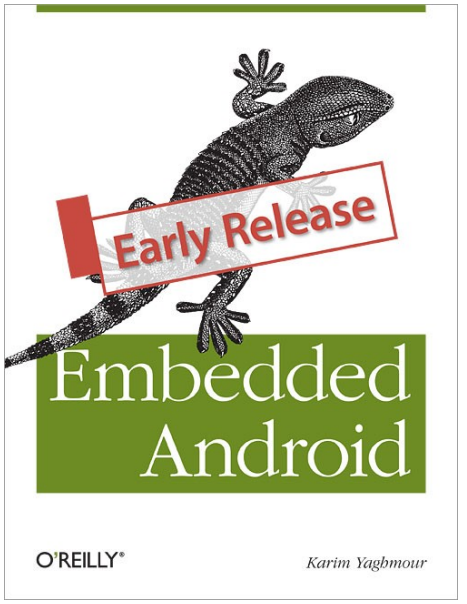
Originals at: www.opersys.com/community/docs

Delivered and/or customized by

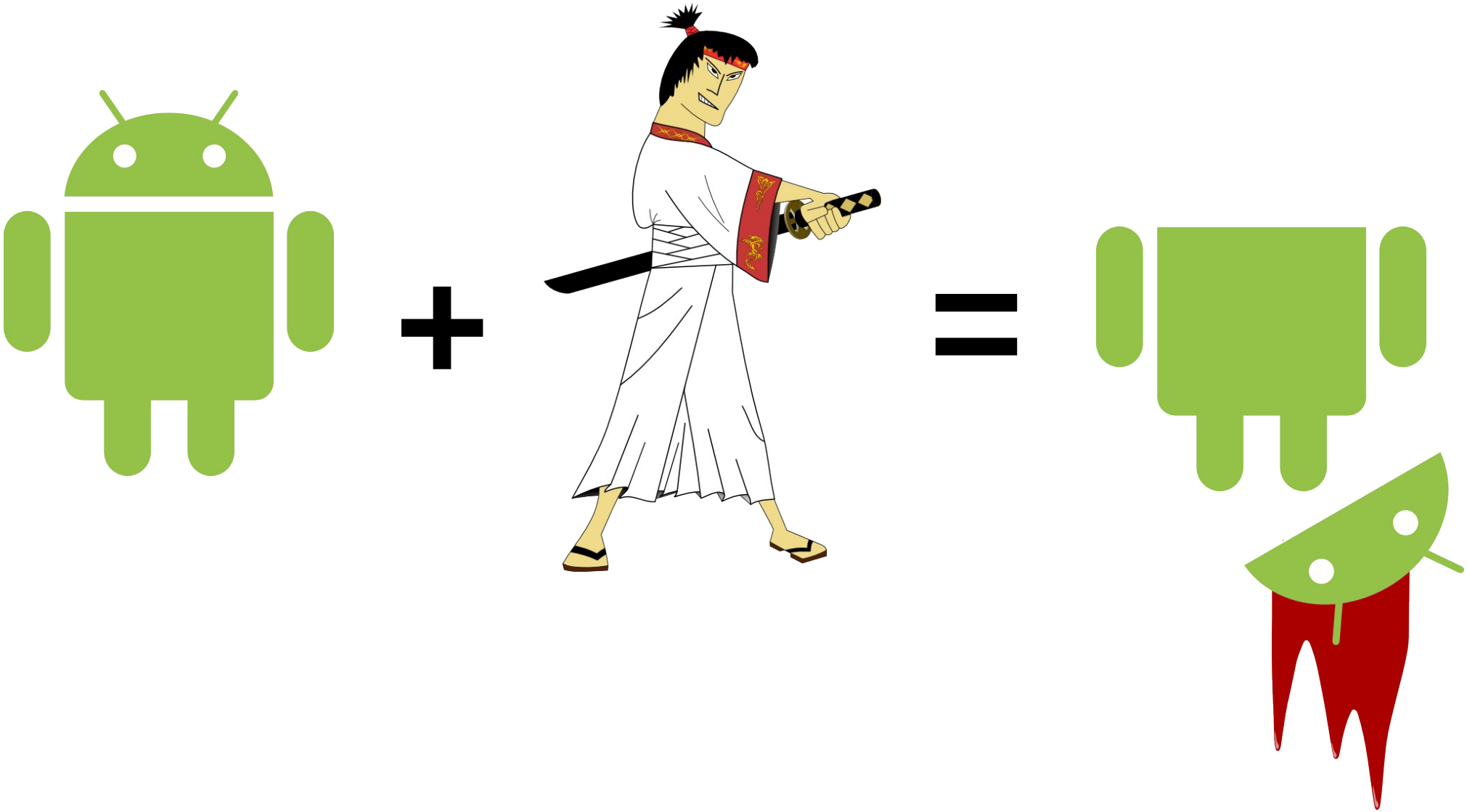
About

- Author of:





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



“And then GNU came back for revenge ...” -- Tarantino's sequel

Agenda

- Why?
- What is it exactly?
- How to get there?
- What's in there anyway?
- Demo

1. Why?

- Took me a long time to wrap my *head* around
  - “Why don't you just use embedded Linux?”

- What's “Embedded Linux” anyway?

1.1. What's “Embedded Linux”?

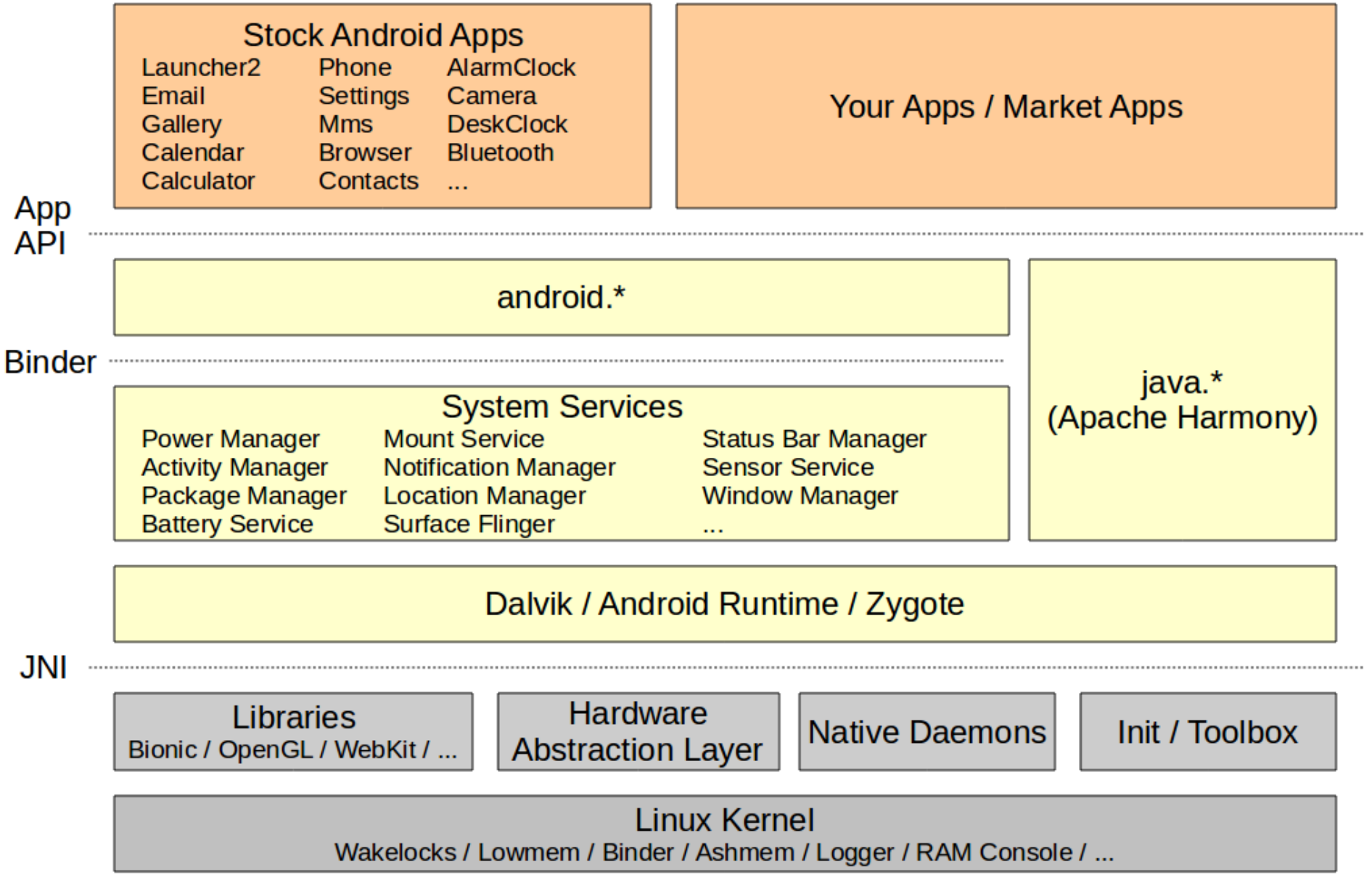
- A set of ad-hoc methods to package the Linux kernel with a (minimal) filesystem.
- FS content “to be determined” case-by-case
- APIs are specific to each device/build
- “Core software”:
  - BusyBox
  - U-Boot
  - GNU Toolchain
- Your flavor of:
  - glibc or uClibc or eglibc
  - yocto or buildroot or eldk or ltib or ptxdist or ...
- No serious UX framework

1.2. What does Android offer?

- ... apart from its increasingly well known UX ...
- Fully-integrated Eclipse IDE
- SDK/NDK
- ADB
- Fastboot
- Published, well-known, and very rich APIs
- A large and growing developer community
- And still we can use the usual suspects:
  - GNU toolchain, BusyBox, u- boot, glibc, ...

An actual standardized dev. env. across all product lines

2. What is “Headless Android” exactly?



2.1. Possibilities

- No Java:
  - TinyAndroid:

```
$ BUILD_TINY_ANDROID=true make -j4
```

  - AOSP w/ custom products .mk file
- The full-blown stack **without**:
  - SurfaceFlinger
  - WindowManager
  - WallpaperService
  - InputMethodManager

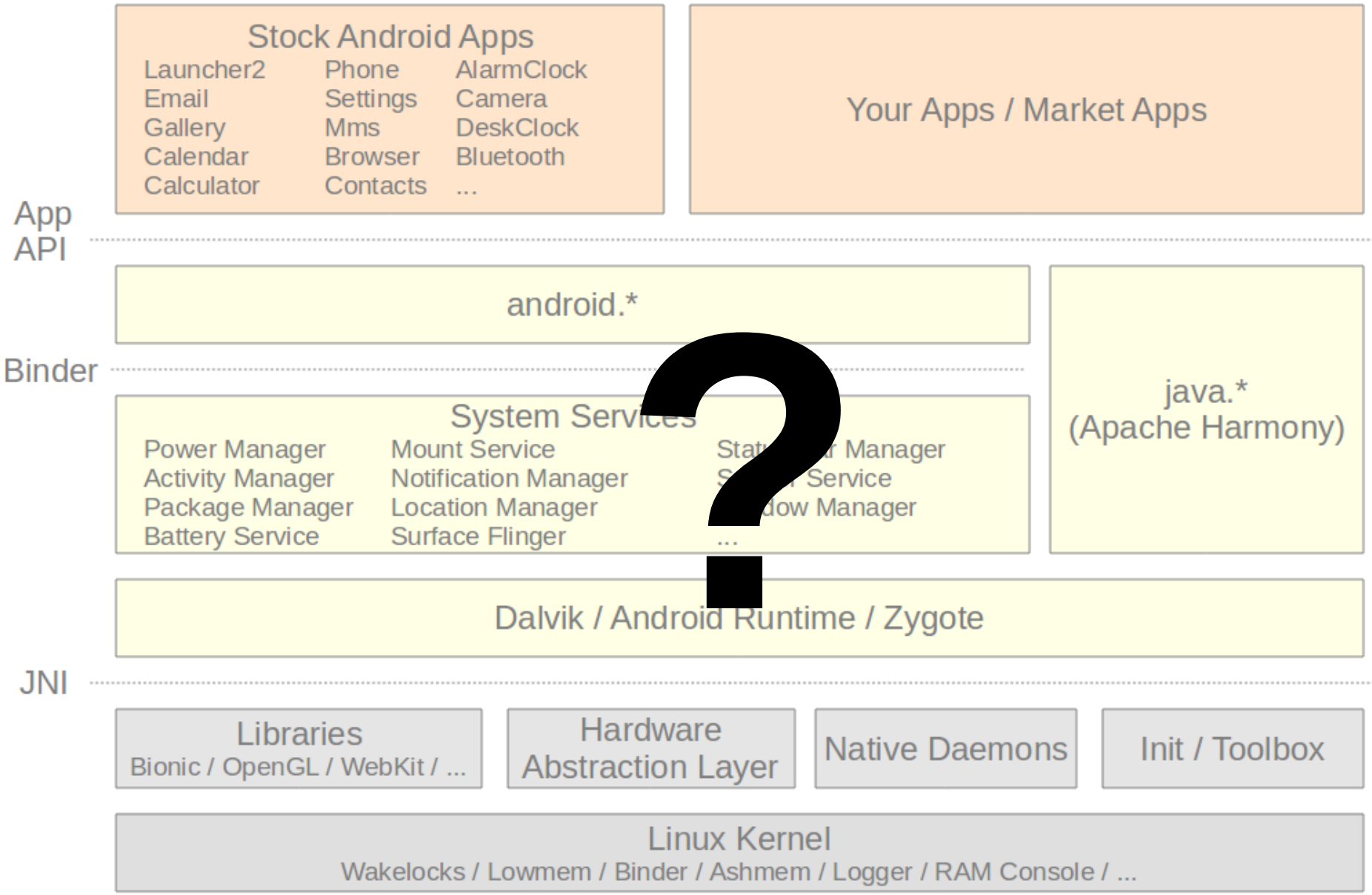
2.2. Tiny Android

- 3MB filesystem
- Minimal root fs
- init
- toolbox + shell
- adb
- bionic + utility libs
- No “system/framework/”
- No “system/app”

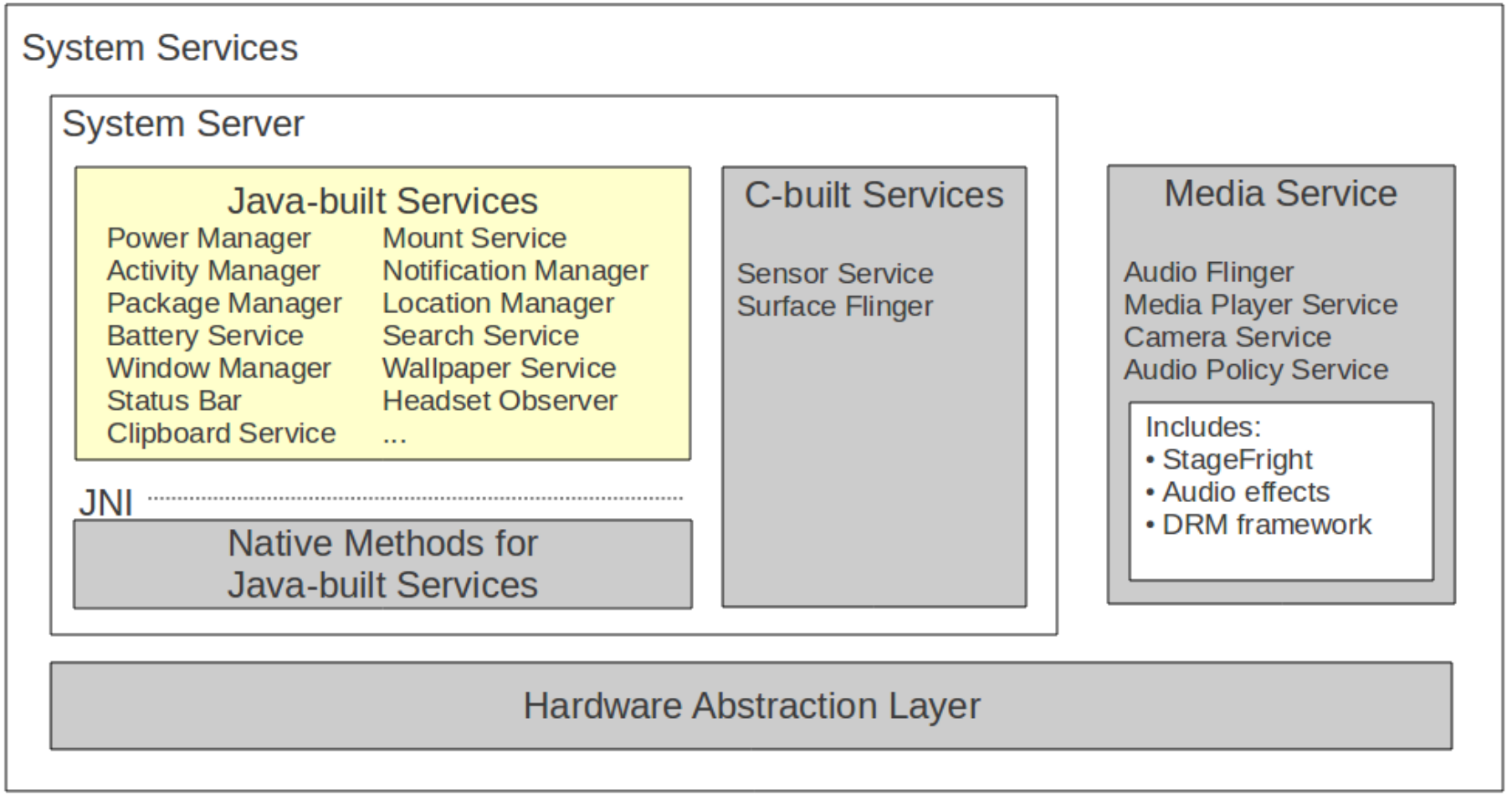
2.3. AOSP w/ custom product .mk

- Have a look at:
  - build/target/product/*.mk
- Create your own device under “/device” and have fun
- Disable zygote at startup
- Remove all apks
- ...

3. How do we get the full stack?



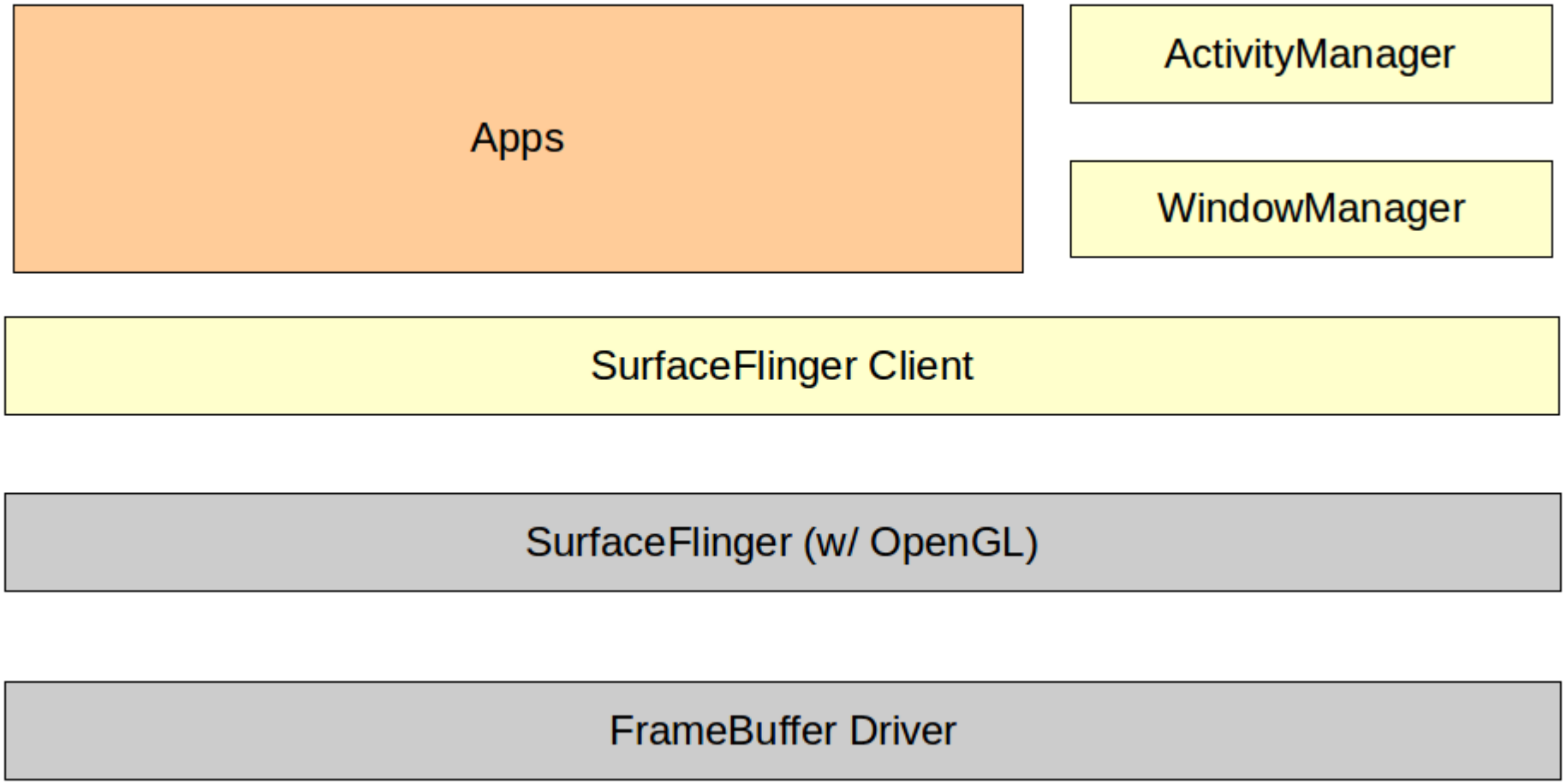
3.1. System Services



3.2. Challenges

- Integration
  - System Services are tightly coupled
  - House of cards
- Dependencies are deeply buried in internals
- There are ~100KLOC of System Services
- Framework expects all System Services
- Rendering/Display are central tenants of arch.

3.3. Android's display architecture



3.4. Attempt #1

- Target: Kill SurfaceFlinger and WindowManager
- Result: FAIL

3.5. Attempt #2

- Target: Kill link between SF and FB
- Result: FAIL

3.6. Attempt #3

- Target: Use VirtualFB
- Result: FAIL

3.7. Attempt #4

- Target: Kill SurfaceFlinger and WindowManager
- Result: SUCCESS!!!

3.8. How?

- Disable:
  - SurfaceFlinger
  - WindowManager
  - WallpaperService
  - InputMethodManager
  - SystemUI
- Don't let SurfaceFlinger Client try to open binder to SurfaceFlinger
- Feed bogus values back from SurfaceFlinger Client
- Disable qemud (emulator artefact)
- Tweak internals by disabling key calls:
  - In ActivityStack.java:
    - startHomeActivityLocked()
    - setAppStartingWindow()
  - wm.detectSafeMode()
  - wm.systemReady()
  - wm.reclaimSuraceMemoryLocked()

4. What's in there anyway?

- Most everything Android gives you:
  - **No UX**
  - Fully-integrated Eclipse IDE
  - SDK/NDK
  - ADB
  - Fastboot
  - Published, well-known, and very rich APIs
  - A large and growing developer community
  - And still we can use the usual suspects:
    - GNU toolchain, BusyBox, u- boot, glibc, ...
- Caveat -- “Activity” no longer works
- You have:
  - Services
  - ContentProviders
  - BroadcastReceivers

4.1. Usage recommendations

- Use “am” to start your components
- Mark your apks as “persistent”
- ...

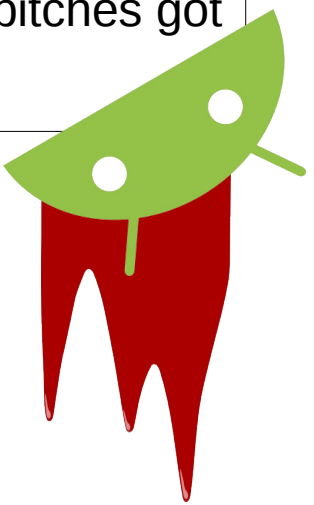
5. Demo

6. Housekeeping

- Work in progress / proof-of-concept
- Do NOT ship a product with this just yet
- Will merge that into cyborgstack
- I haven't bothered removing any of the standard apps: Browser, Email, Launcher2, ...
- Lingering references to WindowManager in ActivityManager
- ... and likely tons I overlooked or didn't test ...

And then he said ...

“As your leader, I encourage you from time to time, and always in a respectful manner, to question my logic. If you're unconvinced that a particular plan of action I've decided is the wisest, tell me so, but allow me to convince you and I promise you right here and now, no subject will ever be taboo. Except, of course, the subject that was just under discussion. The price you pay for bringing up either my GNU or Linux heritage as a negative is... I collect your fucking head. Just like this fucker here. Now, if any of you sons of bitches got anything else to say, now's the fucking time!”



Thank you ...

karim.yaghmour@opersys.com

TM