

Android Things:

Internals and cute embedded nonsense hacks

Embedded Linux Conference 2017

Karim Yaghmour

+karimyaghmour, @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 3rd parties without the EXPLICIT consent from those parties.

(C) Copyright 2017, Opersys inc.

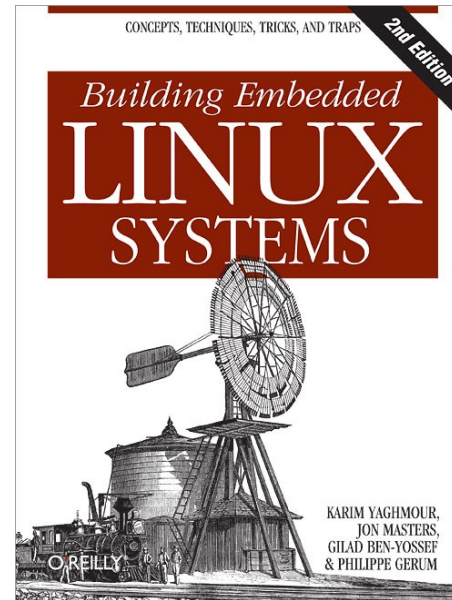
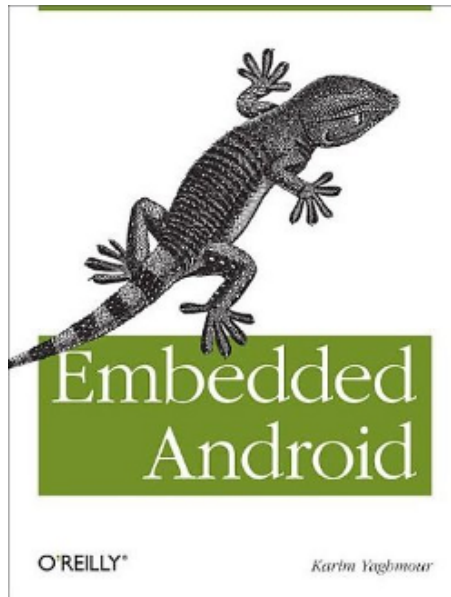
These slides created by: Karim Yaghmour

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, ...

Agenda

1. A bit of history
2. Legacy Architectures
3. The Brillo/Weave Intermezzo
4. Now back to your regular programming
5. Hardware
6. “Things” Architecture
7. Images
8. User-Space
9. Services / Daemons
10. APIs
11. Apps
12. What if I told you ... ?

1. A bit of history

- Embedded Linux
- Android
- Headless Android
- Brillo
- Android Things

1.1. 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

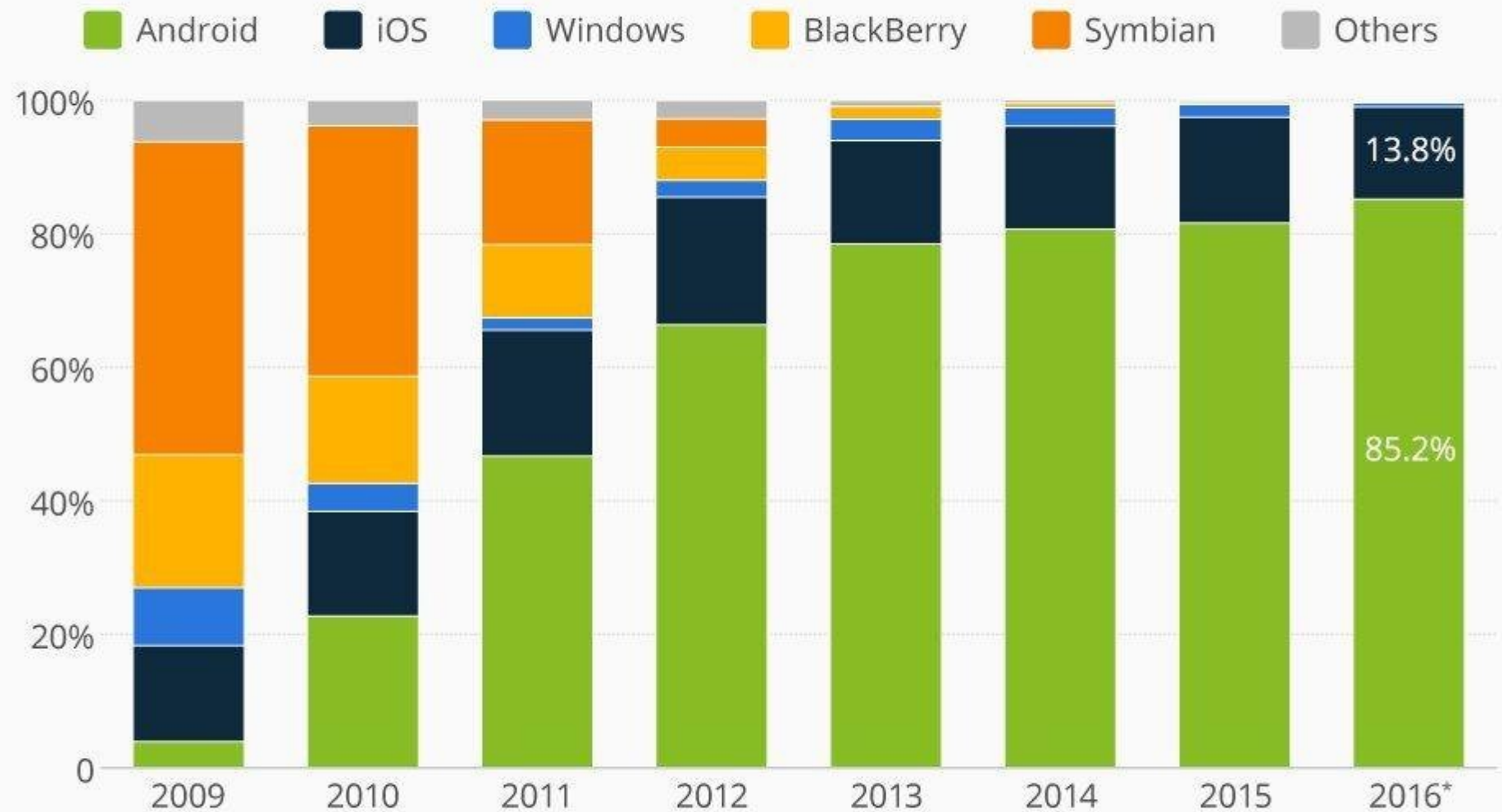
1.2. Android

- ... apart from its ubiquitous UX ...
- Fully-integrated IDE: Android Studio
- SDK/NDK
- ADB
- Fastboot
- Published, well-known, and very rich APIs
- A large and growing developer community
- And still we can use “embedded Linux” components:
 - GNU toolchain, BusyBox, u- boot, glibc, ...

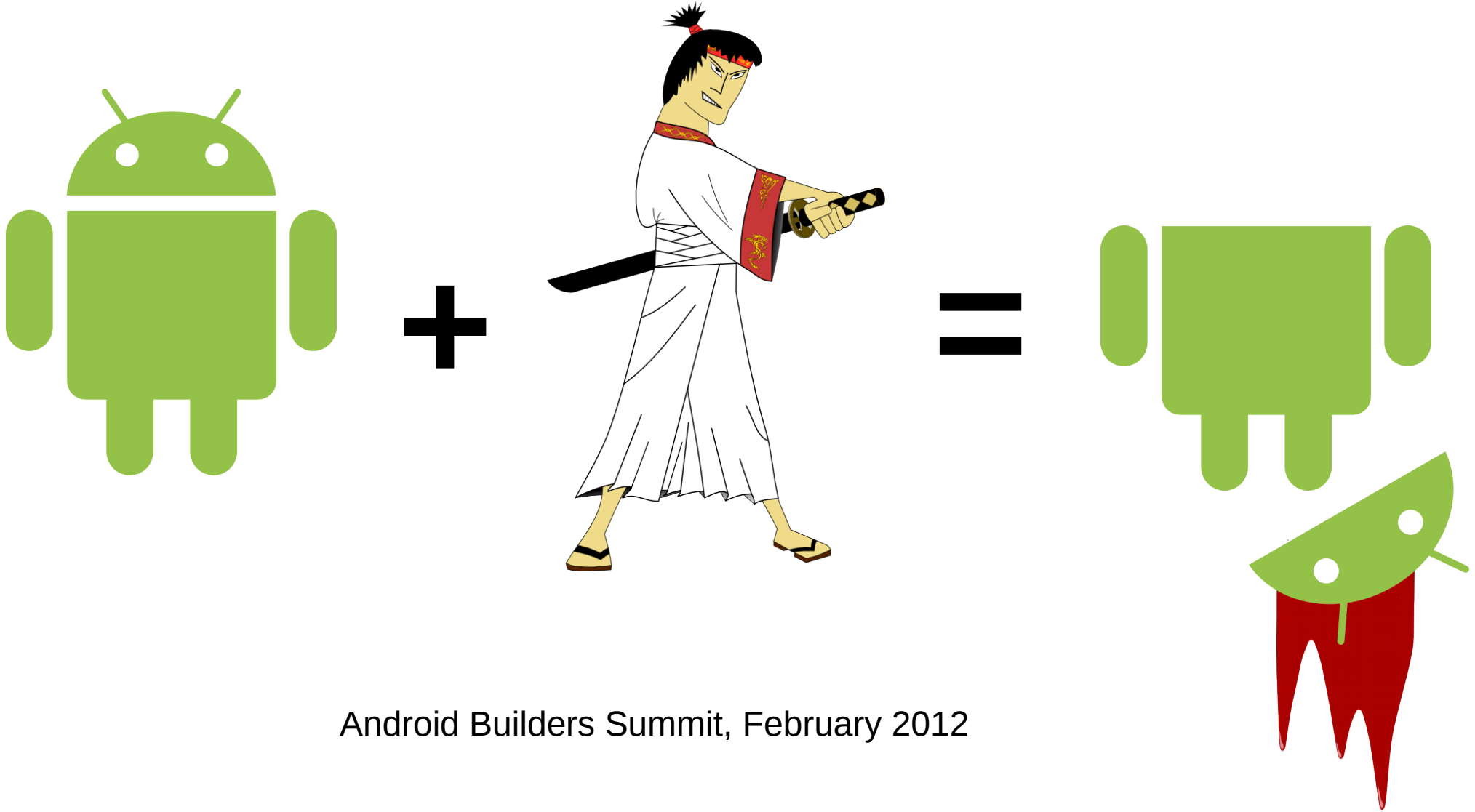
An actual standardized dev. env. across all product lines

The Smartphone Platform War Is Over

Worldwide smartphone operating system market share (based on unit sales)



1.3. Headless Android



Android Builders Summit, February 2012

A few months later ...

ro.config.headless



1.4. Brillo / Weave

- Initial Google platform for IoT*
- Based on Android
- Announced at Google I/O 2015
- Never officially released
- Remained in “developer preview” mode
- Needed to sign up for developer preview on <https://developers.google.com/brillo/>
- Sources available from android.googlesource.com
- “Apps” talk straight to HALs
- Note:

Material here based on sources, NOT on developer preview or any information thereof :P

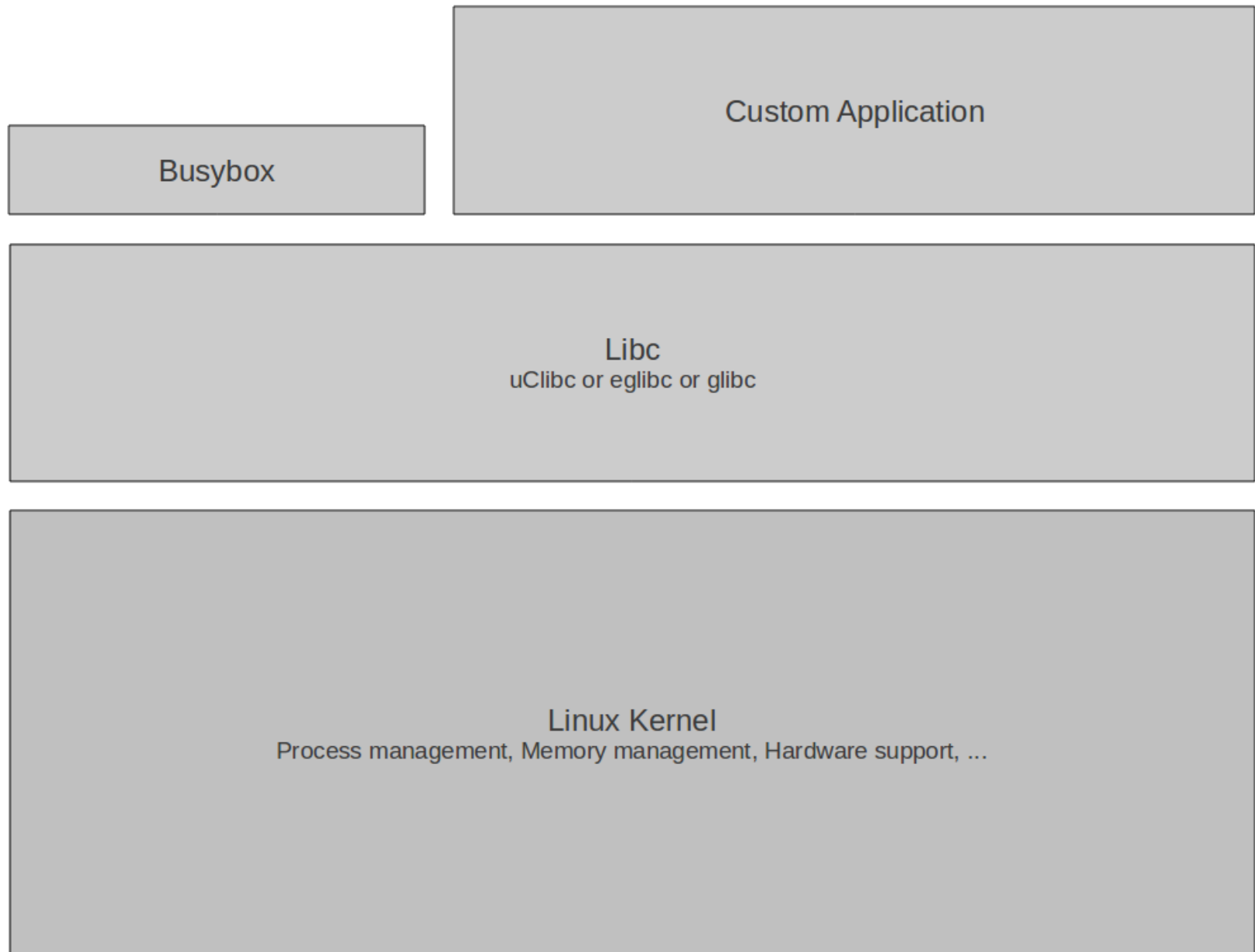
* New, hip way of talking about “Embedded Systems”, something that's been around for ~50+ years.

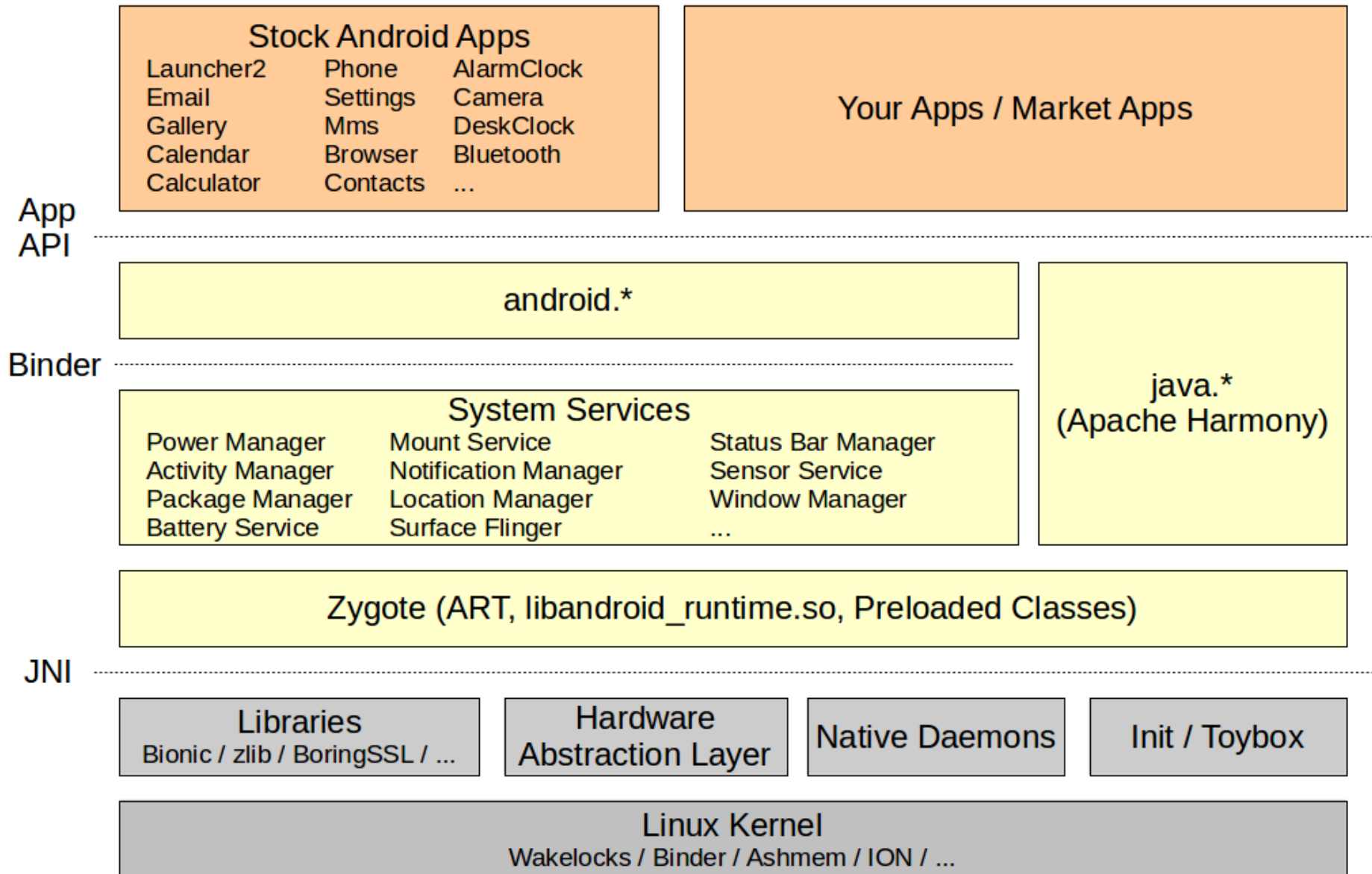
1.5. Android Things

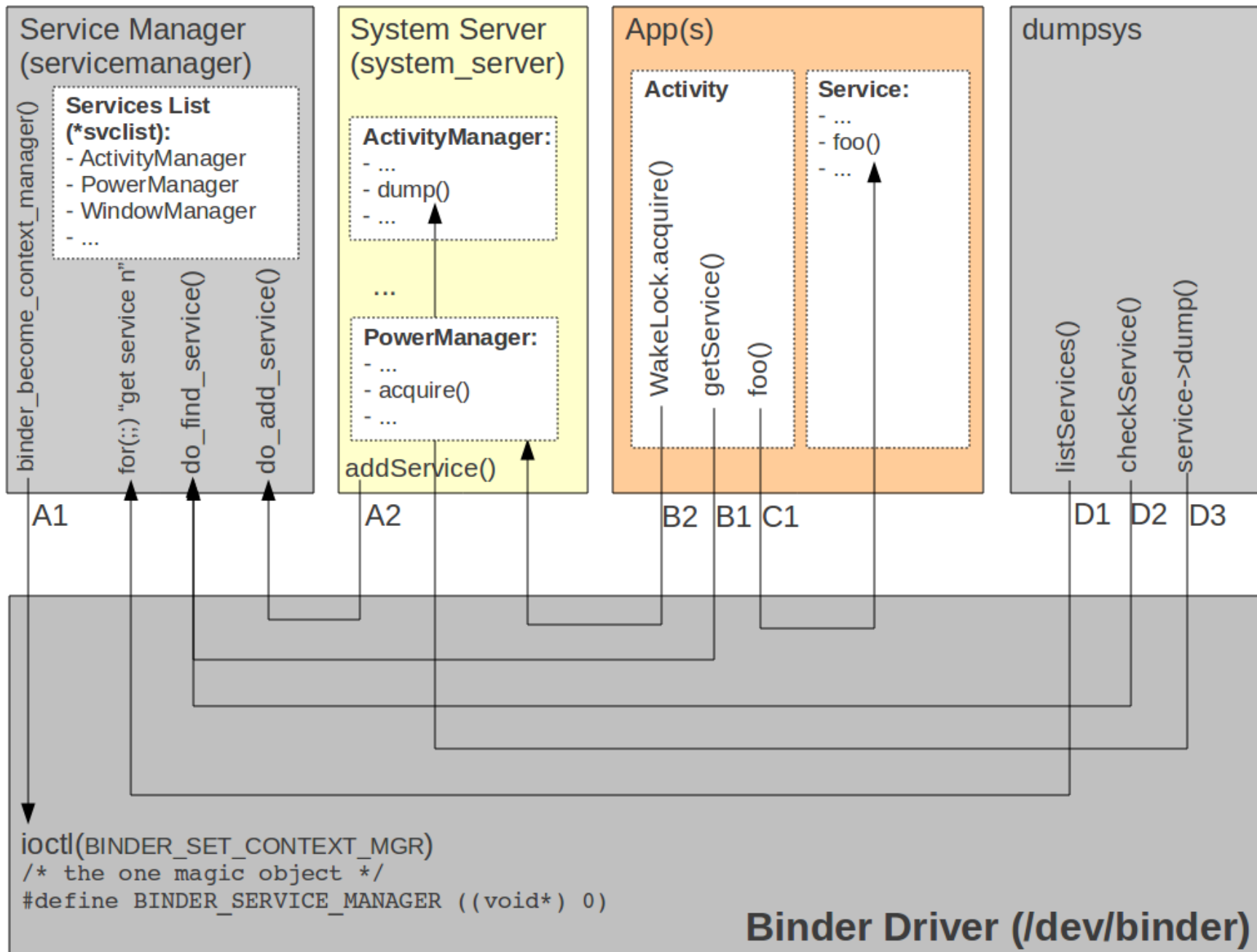
- “Reboot” of Google's Android IoT strategy
- Announced on Dec 13th, 2016
- Reinstates full Android environment
- Developers use existing Android dev tools
- C/C++ available through NDK
- Extended APIs for IoT
- Currently at Developer Preview 2
- Sources not yet available

2. Legacy Architectures

- Embedded Linux
- Android
- 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:

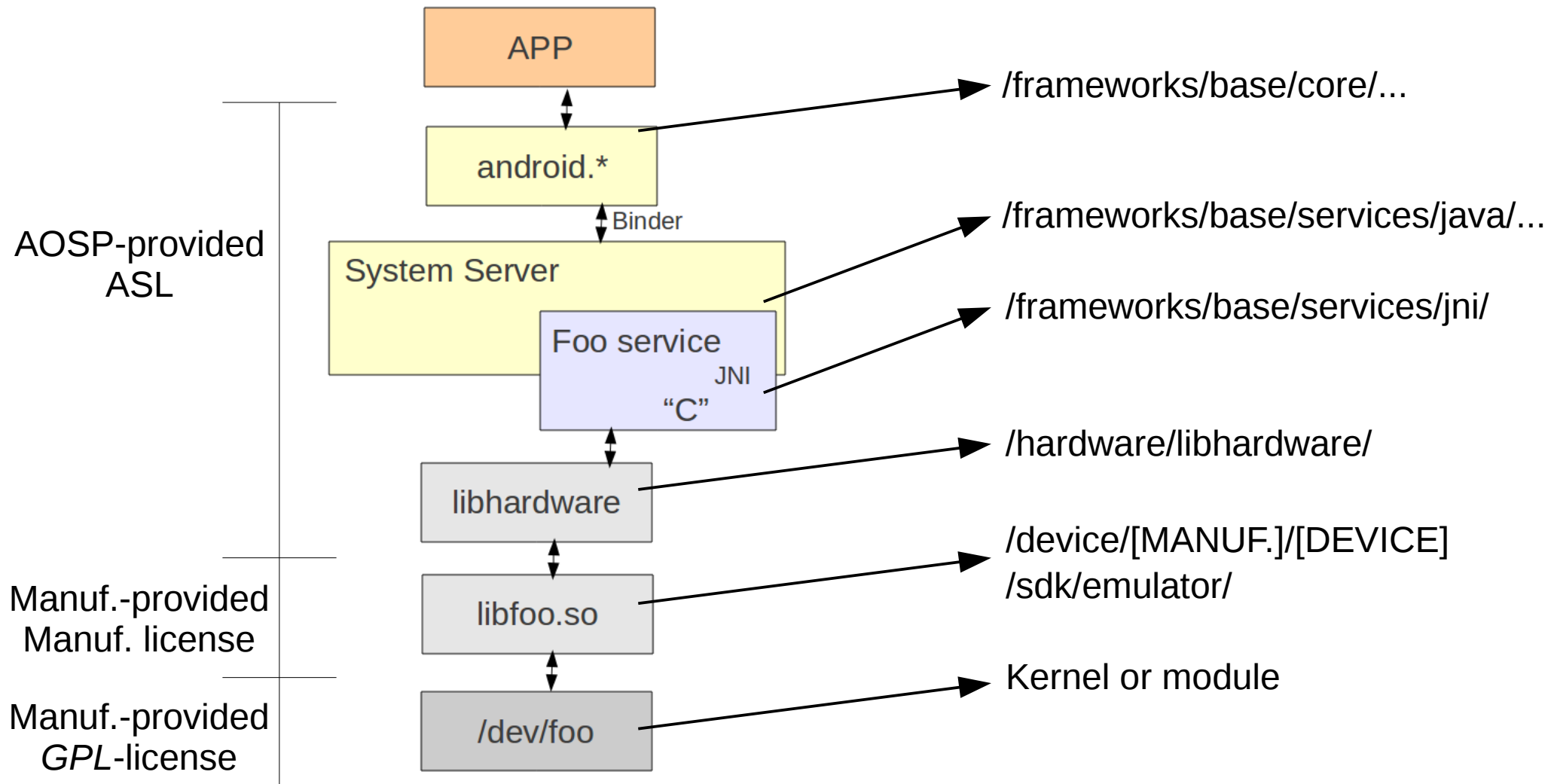
- StageFright
- Audio effects
- DRM framework

Phone App

JNI

Native Methods for
Java-built Services

Hardware Abstraction Layer

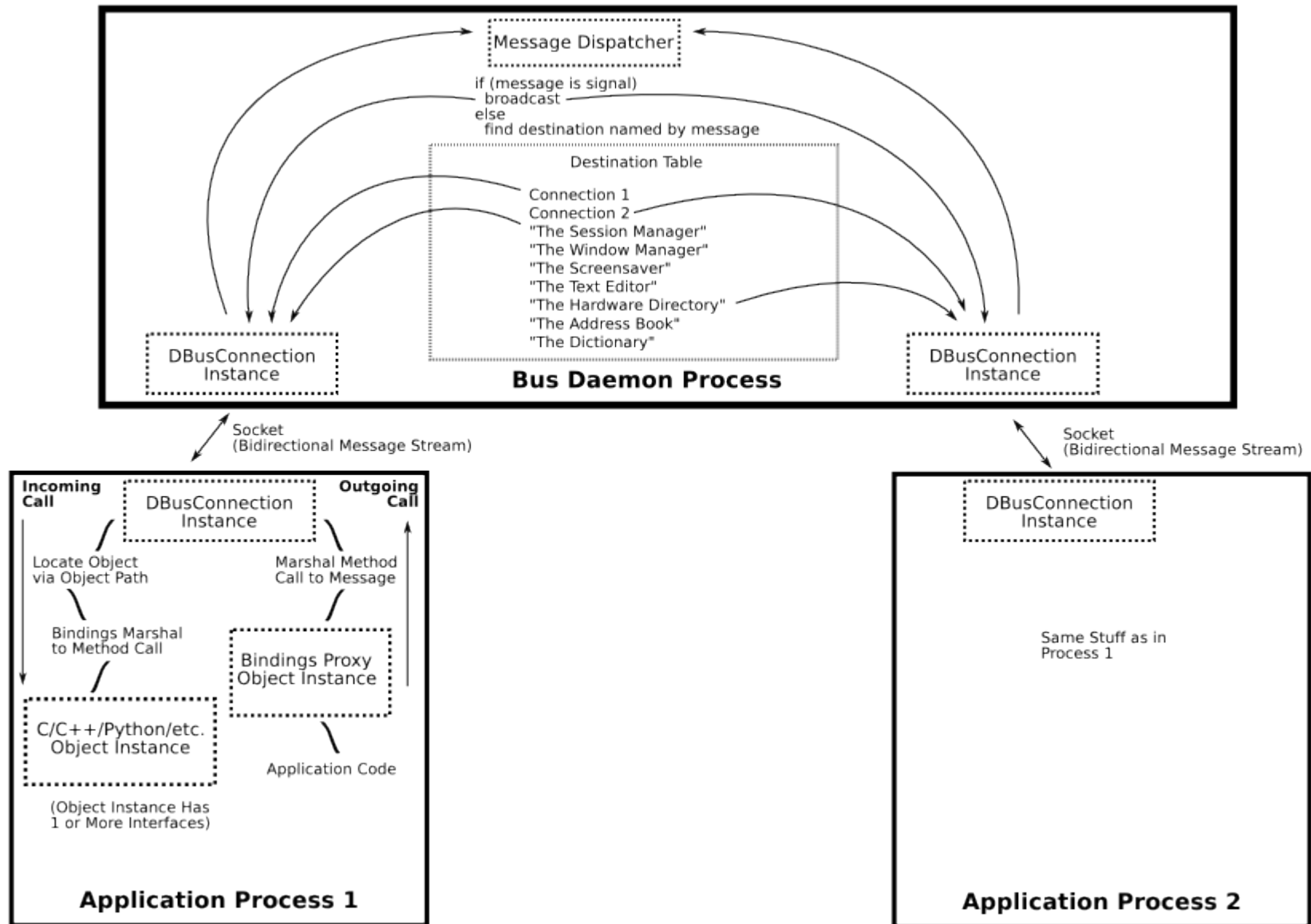


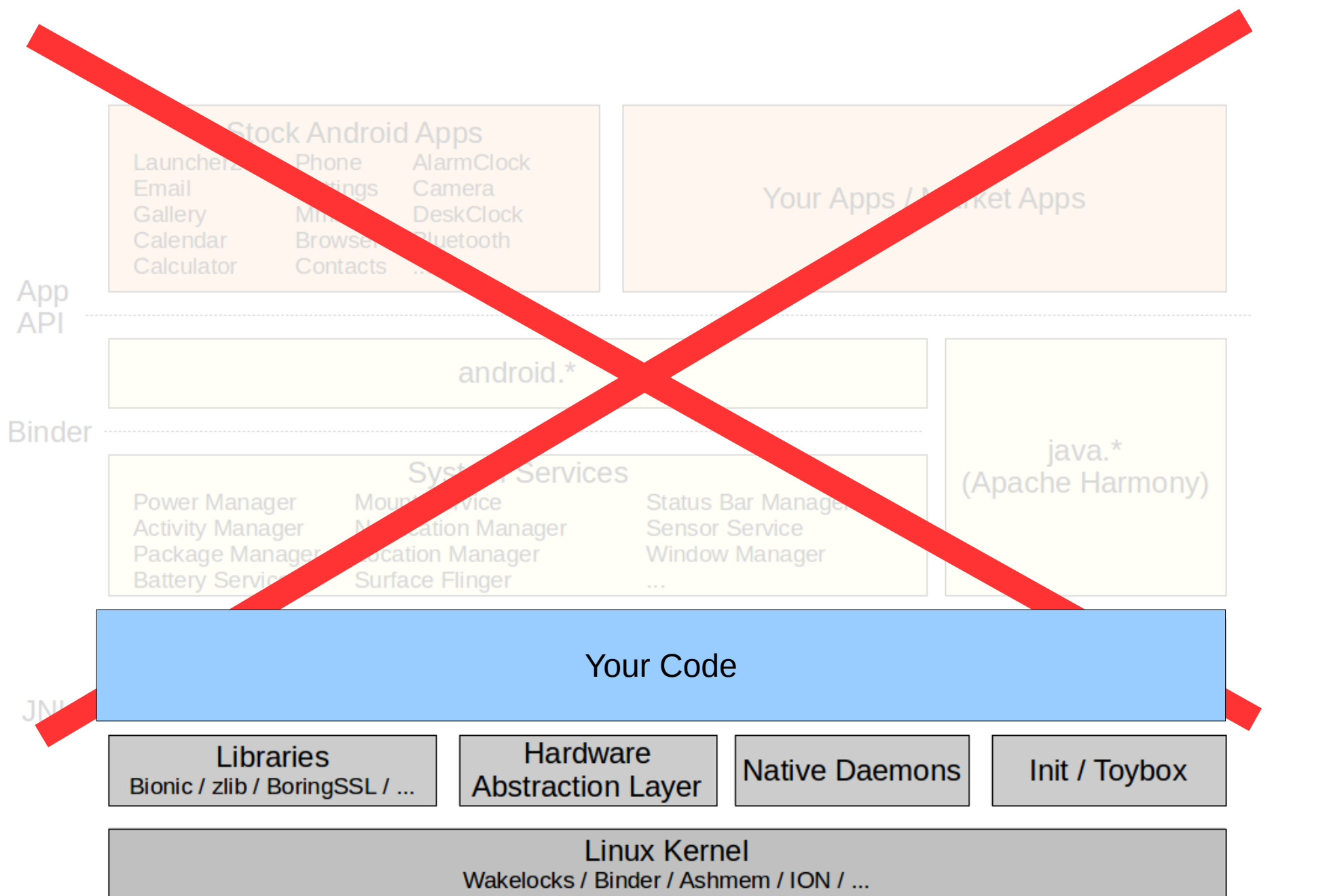
3. The Brillo/Weave Intermezzo

- Architecture
- Sources
- Images
- User-Space
- Services/Daemons
- Weave

3.1. Architecture

- DBus
- Brillo
- Weave







Weave

3.2. Sources

- Getting the sources
- Brillo top level
- Removed from Android top level
- Gone from external/
- New to external

3.2.1. Getting the sources

- Same project repo as Android:
android.googlesource.com
- Different manifest file
- Using “repo”:

```
$ repo init -u https://android.googlesource.com/brillo/manifest
$ repo sync
```

3.2.2. Brillo top level

bionic	C library replacement
bootable	Reference bootloader
build	Build system
device	Device-specific files and components
external	Copy of external projects used by AOSP
frameworks	Native system services and libraries
hardware	Hardware support libs
libnativehelper	JNI helpers
prebuilts	Prebuilt binaries
product	“Products” running on Brillo
system	Embedded Linux core
tools	Brillo Development Kit

3.2.3. Removed from Android top level

art
cts
dalvik
dvelopers
development
docs
frameworks/base and more
libcore
ndk
packages
pdk
sdk
system/vold and more
tools/external

3.2.4. Removed from external/

adt-infra	easymock	javassist	libusb-compat	oauth	v8
android-clat	eclipse-basebuilder	jcommander	libutf	objenesis	vboot_reference
androidplot	eclipse-windowbuilder	jdiff	libvncserver	okhttp	vixl
ant-glob	eigen	jetty	libvorbis	opencv	vogar
antlr	emma	jhead	libvterm	opencv3	vulkan-validation-layers
apache-commons-math	esd	jline	libxml2	owasp/sanitizer	webp
apache-harmony	eyes-free	jmdns	libyuv	parameter-framework	webRTC
apache-http	fdlibm	jsilver	littlemock	pdfium	xmlwriter
apache-xml	fio	jsmn	lld	piex	xmp_toolkit
blktrace	fonttools	jsoncpp	ltrace	ppp	zopfli
bouncycastle	freetype	jsr305	marisa-trie	proguard	zxing
caliper	fsck_msdos	jsr330	markdown	regex-re2	
cblas	gemmlowp	junit	mdnsresponder	replicaisland	
ceres-solver	giflib	junit-params	mesa3d	rmi4utils	
chromium-libpac	glide	kernel-headers	messageformat	roboelectric	
chromium-trace	google-fonts/carrois-gothic-sc	libavc	mmc-utils	roboto-fonts	
chromium-webview	google-fonts/coming-soon	libdivsufsort	mockftpserver	scrypt	
cmockery	google-fonts/cutive-mono	libdrm	mockito	seccomp-tests	
conscrypt	google-fonts/dancing-script	libedit	mockwebserver	sfntly	
crcalc	google-tv-pairing-protocol	libexif	mp4parser	skia	
dagger2	googletest	libhevc	mtpd	sl4a	
deqp	gptfdisk	libmojo	nanohttpd	slf4j	
dexmaker	guava	libmpeg2	nanopb-c	smali	
dlmalloc	guice	libmtsp	naver-fonts	snakeyaml	
dng_sdk	hamcrest	libnfc-nci	netcat	sqlite	
doclava	harfbuzz_ng	libnfc-nxp	netperf	srtsp	
donuts	hyphenation-patterns	libpcap	neven	svox	
drm_gralloc	ipsec-tools	libphonenumbers	nfacct	tagSoup	
drm_hwcomposer	jacoco	libpng	nist-pkits	testng	
droiddriver	jarjar	libusb	nist-sip	tcpdump	
dtc	jasqlite		noto-fonts	timezonepicker-support	

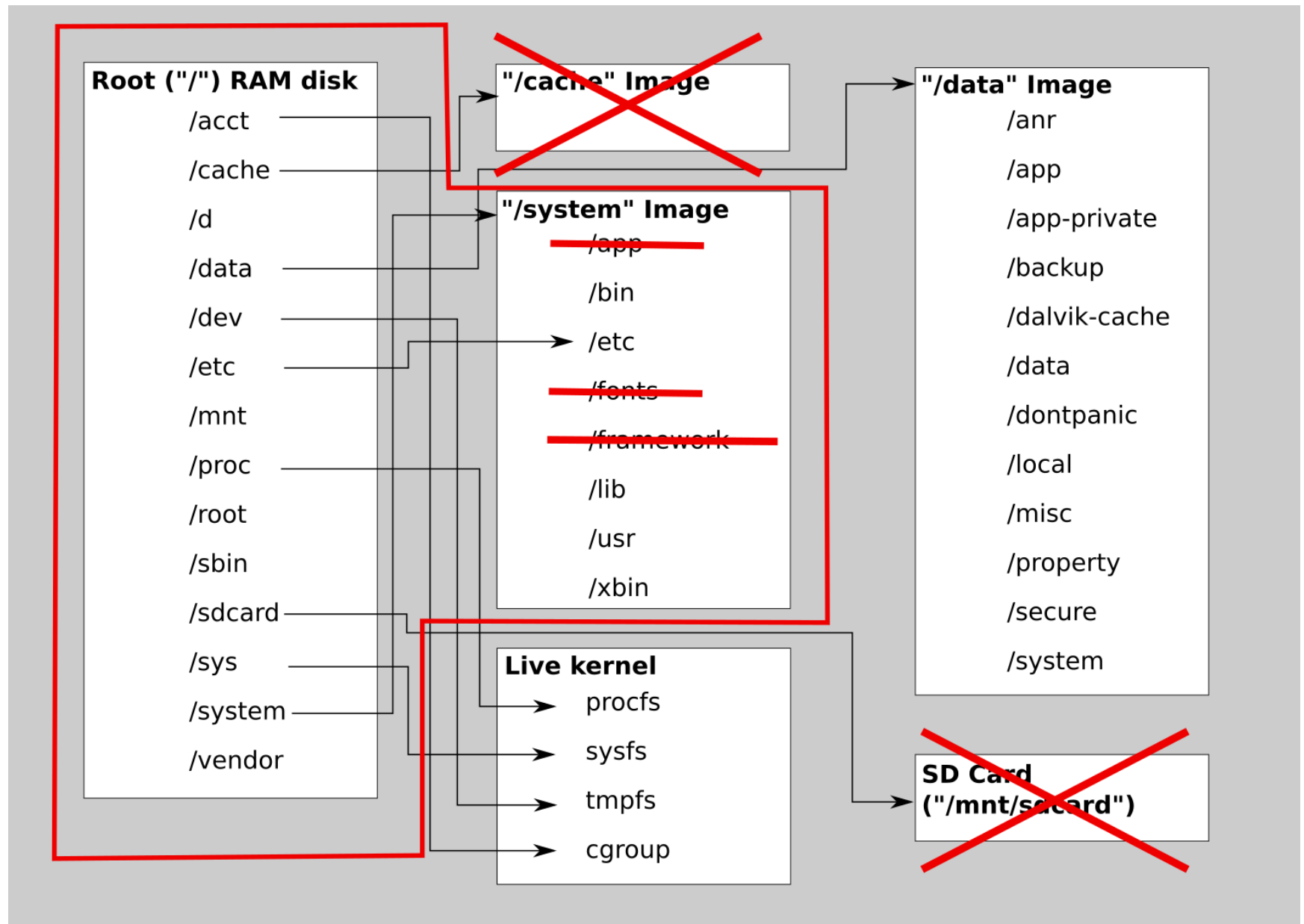
3.2.5. Added to external/

bvb	Brillo Verified Boot
chromite	Tools to build Chrome OS
gentoo	
libdivsufsort	lightweight suffix array construction algorithm library
lzop	LZO compression tool

3.3. Images

- AOSP x86 32-bit:
 - 5.1M out/target/product/generic_x86/cache.img
 - 1.3M out/target/product/generic_x86/ramdisk.img
 - **1.3G out/target/product/generic_x86/system.img**
 - 12M out/target/product/generic_x86/userdata.img
 - 551M out/target/product/generic_x86/userdata-qemu.img
- Brillo x86 64-bit:
 - 6.2M out/target/product/brilloemulator_x86_64/boot.img
 - 36K out/target/product/brilloemulator_x86_64/partition-table.img
 - **158M out/target/product/brilloemulator_x86_64/system.img**
 - 201M out/target/product/brilloemulator_x86_64/userdata.img
 - 551M out/target/product/brilloemulator_x86_64/userdata-qemu.img

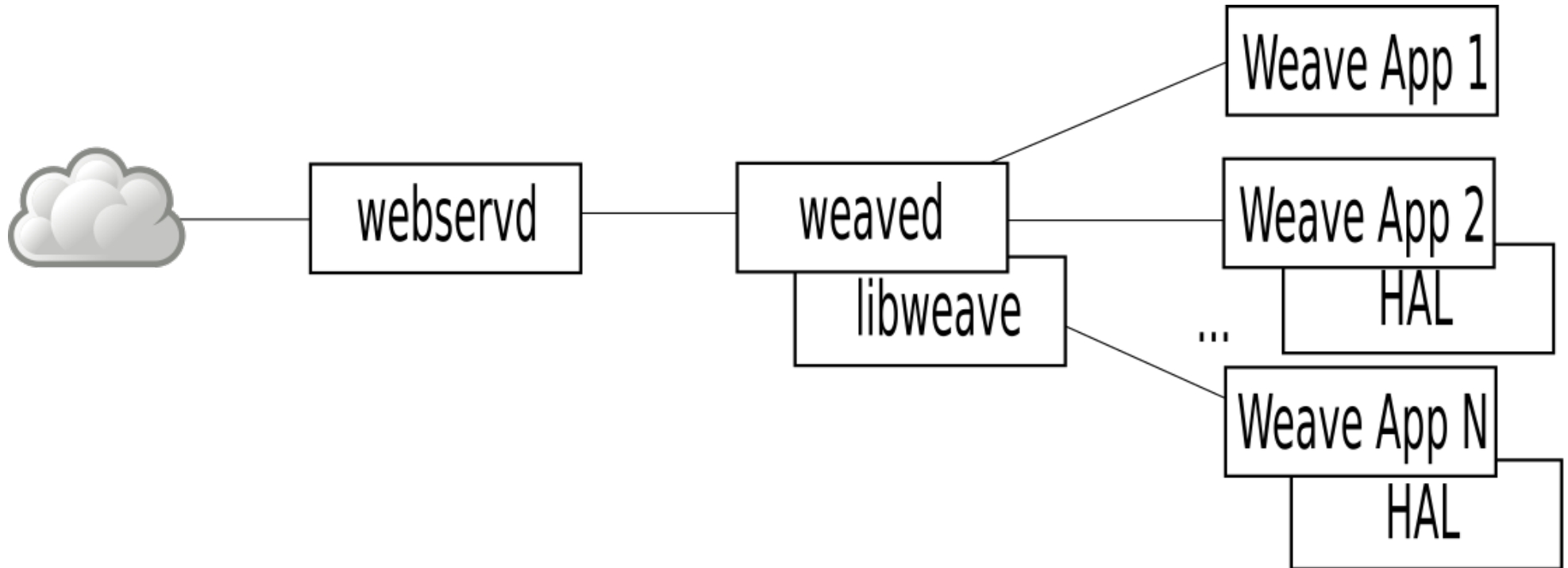
3.4. User Space



3.5. Services / Daemons

- dbus-daemon
- servicemanager
- avahi-daemon
- keystore
- nativeperms
- peripheralman
- sensorservice
- wpa_supplicant
- brilloaudioservice
- metrics_collector
- metriscsd
- perfprofd
- tlsdated
- tpm_managerd
- trunksd
- update_engine
- weaved
- webservd
- shill
- firewallld
- dhcpcd




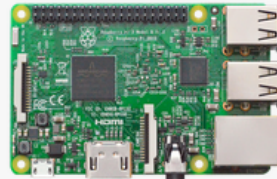
3.6. Weave



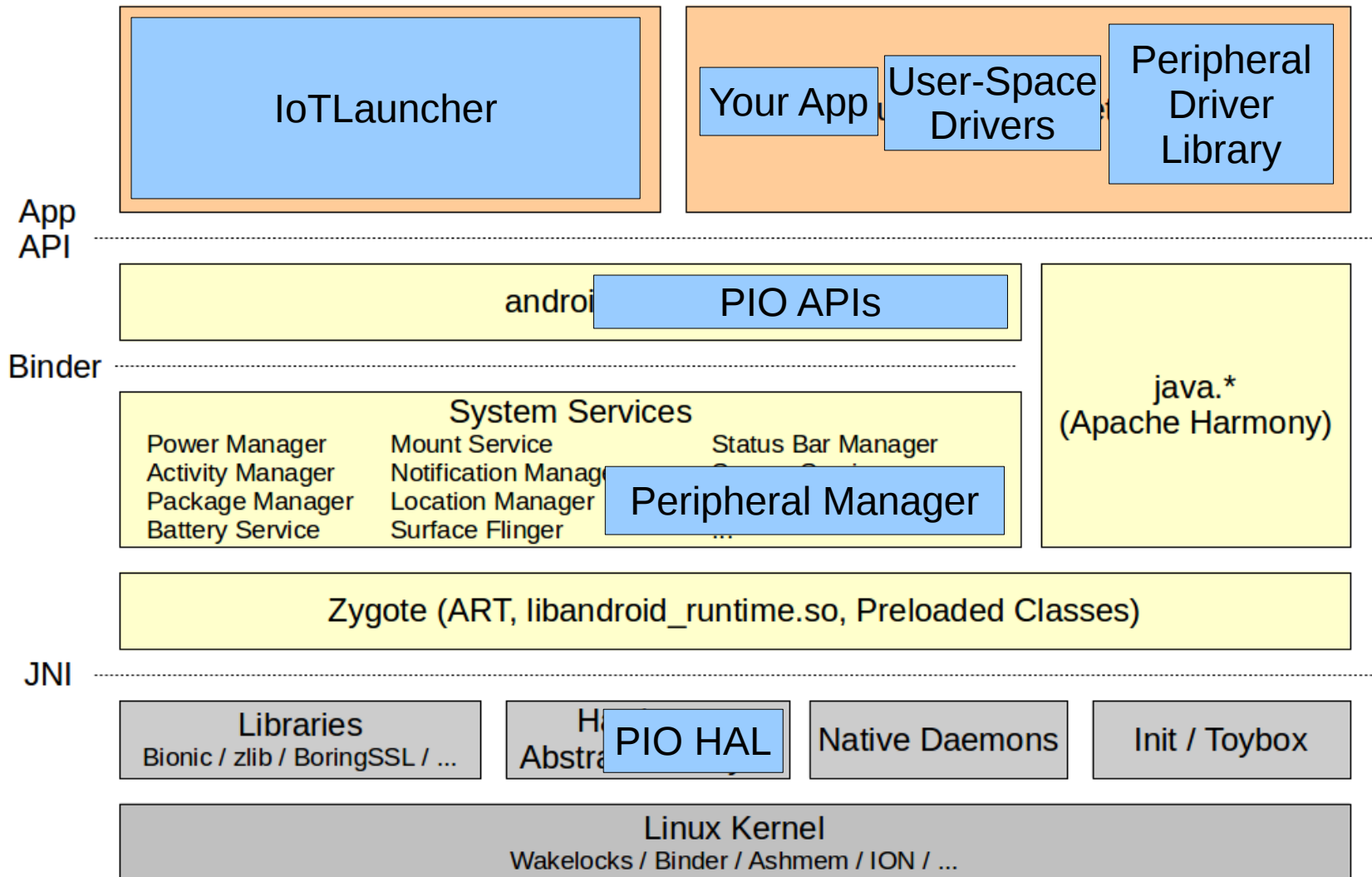
4. Now back to your regular programming

- Android Things does away with most of Brillo
- Revert architecture to original
- Weave seems gone from the FS, but site still on
- Go back to system services with HALs
- Extend Android API for IoT
- Use of Android Studio and co.
- Optional Display
- Use of traditional static permissions

5. Hardware

Platform	Intel® Edison	Intel® Joule	NXP Pico i.MX6UL	Raspberry Pi 3
	 <p>Learn More Where to buy Get Started</p>	 <p>Learn More Where to buy Get Started</p>	 <p>Learn More Where to buy Get Started</p>	 <p>Learn More Where to buy Get Started</p>
CPU & Memory	<ul style="list-style-type: none"> Intel® Atom™ 500MHz dual-core x86 1GB RAM 	<ul style="list-style-type: none"> Intel® Atom™ 1.5GHz/1.7GHz quad-core x86 3GB/4GB RAM 	<ul style="list-style-type: none"> NXP i.MX6Ultralite 500MHz ARM Cortex A7 512MB RAM 	<ul style="list-style-type: none"> Broadcom BCM2837 1.2GHz quad-core ARM Cortex A53 1GB RAM
Storage	4GB eMMC	8GB/16GB eMMC	4GB eMMC	MicroSD card slot
Display	No	HDMI	No	HDMI
Camera	No	CSI-2	No	CSI-2
Audio	USB 2.0	USB 2.0	3.5mm Analog	USB 2.0 3.5mm Analog Output
Networking	Wi-Fi 802.11n Bluetooth® 4.0	Wi-Fi 802.11ac Bluetooth® 4.2	10/100 Ethernet Wi-Fi 802.11n Bluetooth® 4.1	10/100/1000 Ethernet Wi-Fi 802.11n Bluetooth® 4.1
USB	1x USB 2.0 OTG	2x USB 2.0 Host 1x USB 3.0 OTG	1x USB 2.0 Host 1x USB 2.0 OTG	4x USB 2.0 Host

6. “Things” Architecture



7. Images

- Brillo x86 64-bit emulator:
 - 6.2M boot.img
 - 36K partition-table.img
 - **158M system.img**
 - 201M userdata.img
- Things x86 Edison:
 - 11M boot.img
 - 176M gapps.img
 - 4.6M oem.img
 - **420M system.img**
 - 376K u-boot-edison.img
 - 35M userdata.img
- Things Rpi3:
 - **350M root + system**
 - 150M gapps
- AOSP x86 32-bit:
 - **1.3G system.img**

8. User-Space

- Same FS layout as Android
- Mostly same processes
- In short, unlike Brillo, it's still very much Android

9. Services / Daemons

- /system/bin/peripheralman
- C++ System Service
- Responds to dumpsys:

```
dumpsys com.google.android.things.pio.IPeripheralManager
```

- Also was in Brillo
- Sources (6 months old):
 - <https://android.googlesource.com/platform/system/peripheralmanager/>
- New HAL:
 - peripheral_io.h
- As with other HALs:
 - /system/lib/hw/peripheral_io.<board_name>.so

10. APIs

com.google.android.things.pio

Classes

Gpio	Controls a GPIO pin.
GpioCallback	GPIO interrupt callback.
GpioDriver	Allow users to register GPIO expanders.
I2cDevice	Controls an I2C device.
PeripheralManagerService	Lists and opens peripherals.
PioDriverManager	Manager for PIO userspace drivers.
Pwm	Controls a PWM pin.
SpiDevice	Class used to control a SpiDevice.
UartDevice	Controls an UART device.
UartDeviceCallback	UART interrupt callback.

com.google.android.things.userdriver

Classes

GpsDriver	Driver to interface with user-connected GPS sensors.
InputDriver	Driver to interface with user-connected input devices.
InputDriver.Builder	Builder class for InputDriver objects.
UserDriverManager	Manager for all user drivers.
UserSensor	A user-defined sensor backed by a UserSensorDriver .
UserSensor.Builder	Builder to allow for flexible UserSensor construction.
UserSensorDriver	Driver to interface with a user-connected sensor.
UserSensorReading	A single user-sensor reading.

11. Apps

- New Launcher: IoTLauncher.apk
 - Still responds to same intent as LauncherN.apk
 - Try: `aapt l -a IoTLauncher.apk`
 - Likely party firing `android.intent.category.IOT_LAUNCHER`
 - Try: `strings IoTLauncher.odex | grep IOT`
- Gone:
 - Home launcher (replaced by IoTLauncher)
 - Status bar
 - Settings (at least the content provider ... “ps” says the app is still running ;))
- Interesting:
 - Google services (some of them at least)
- Try:
 - `dumpsys gfxinfo`

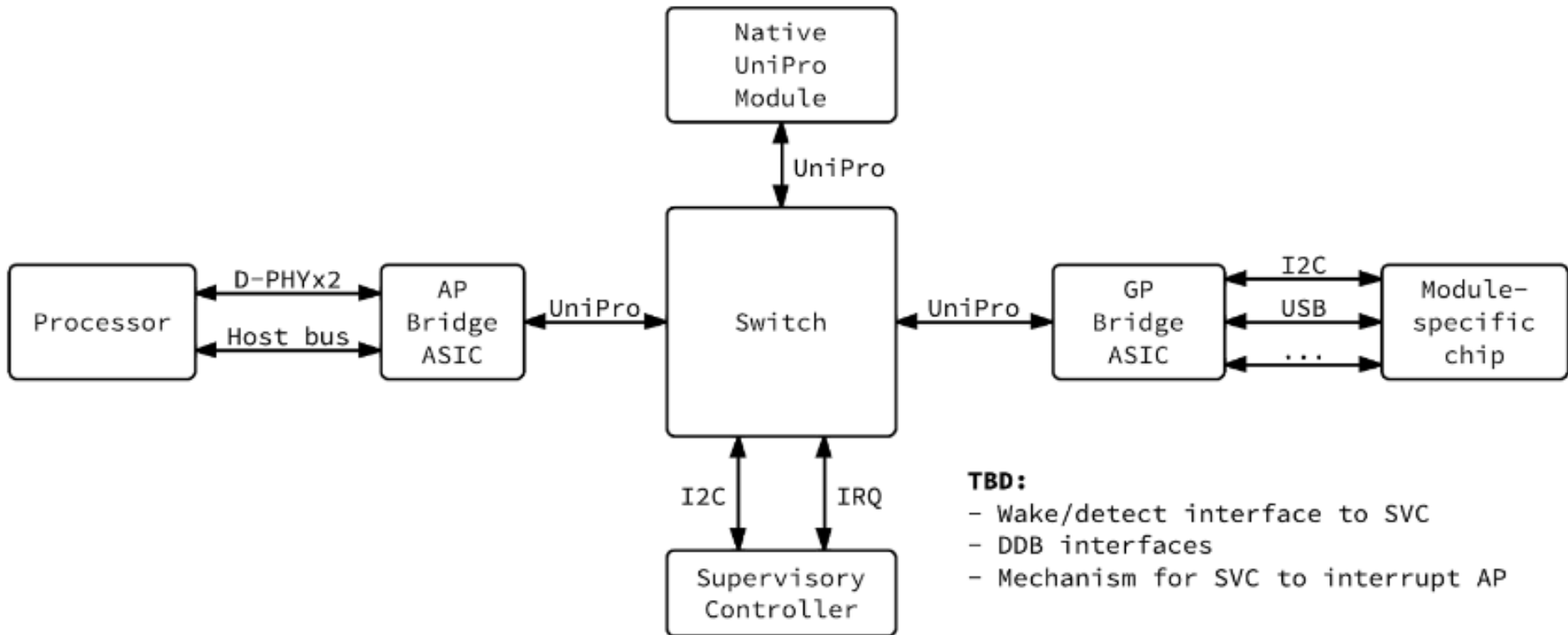
12. What if I told you ... ?

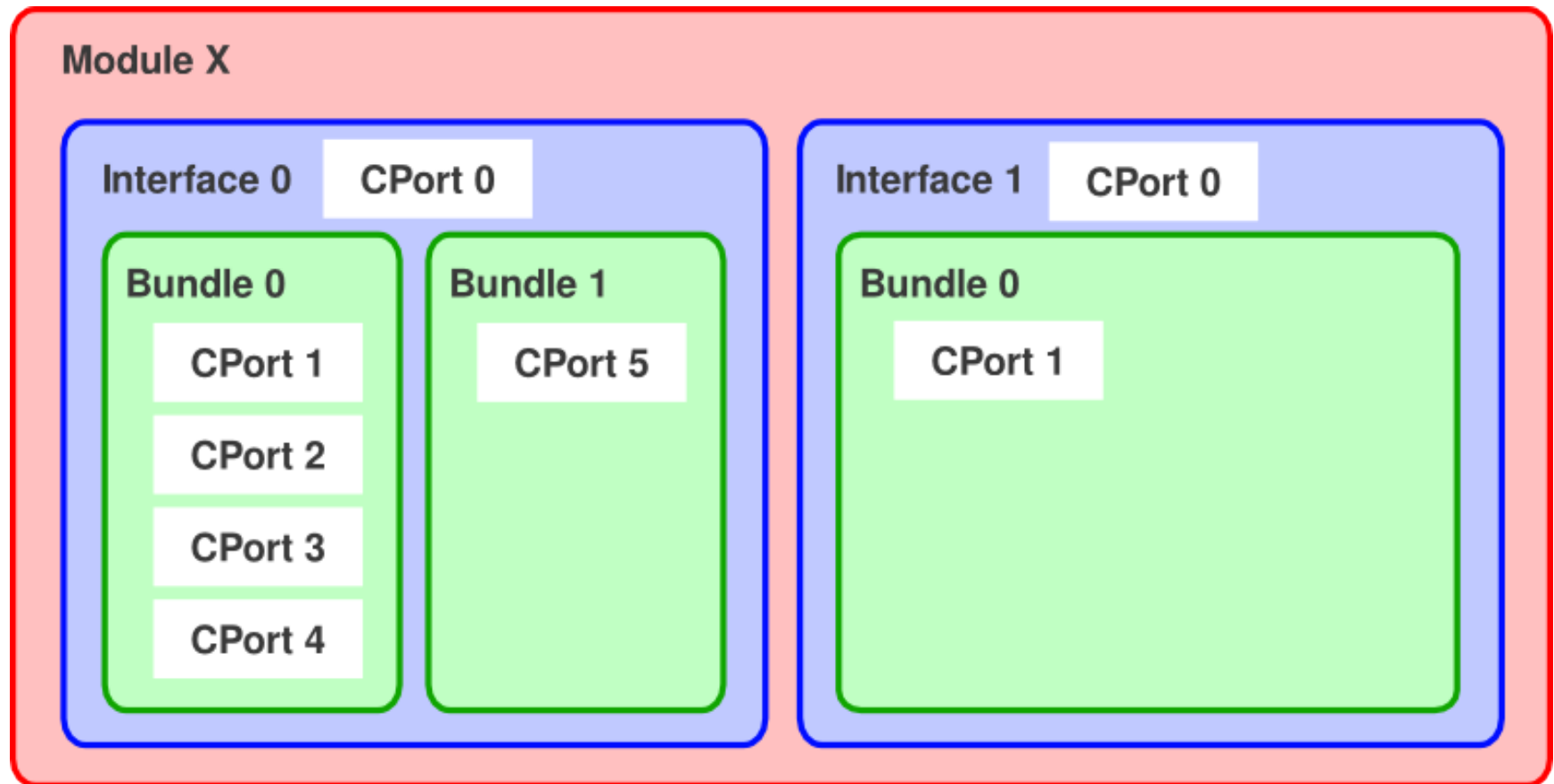
- What's this IoT thing ... ?
 - Consumer
 - Industrial/Commercial
- 256MB+ RAM for an embedded device, really?
- ...
- A long time ago in a galaxy far far away ...

RIP Project Ara

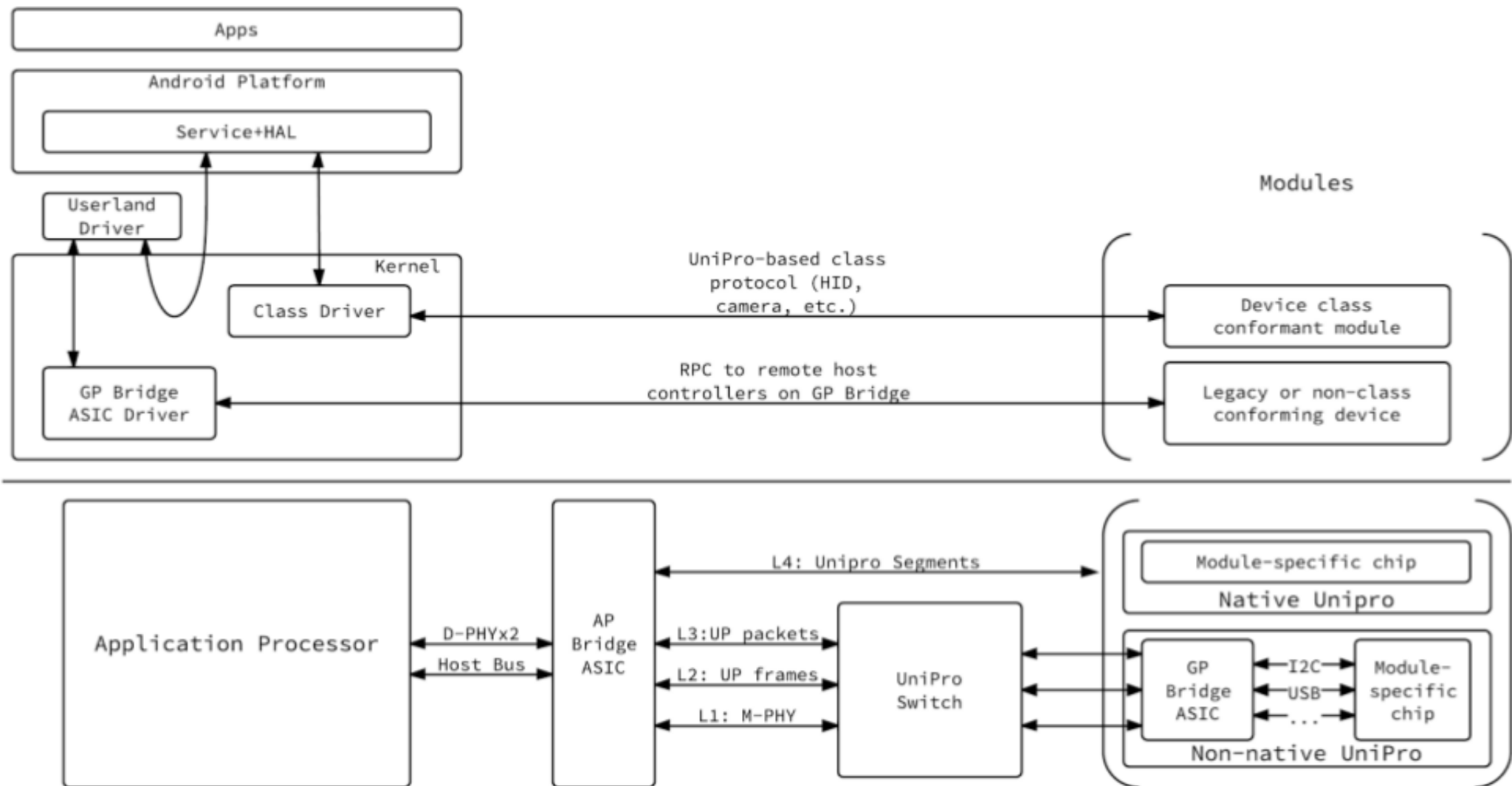


12.1. Hardware Architecture





12.2. Software Architecture

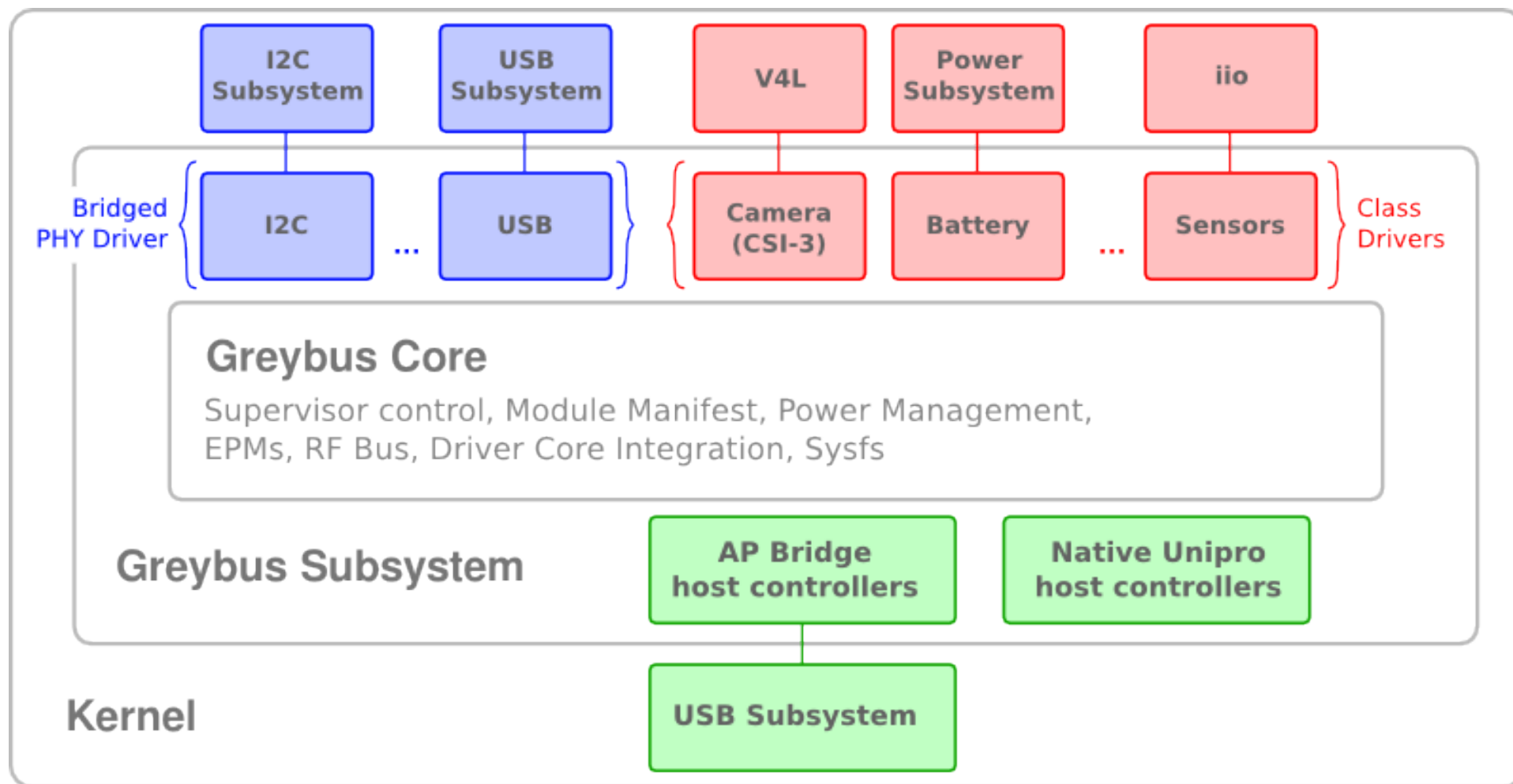


12.3. Greybus Device Classes

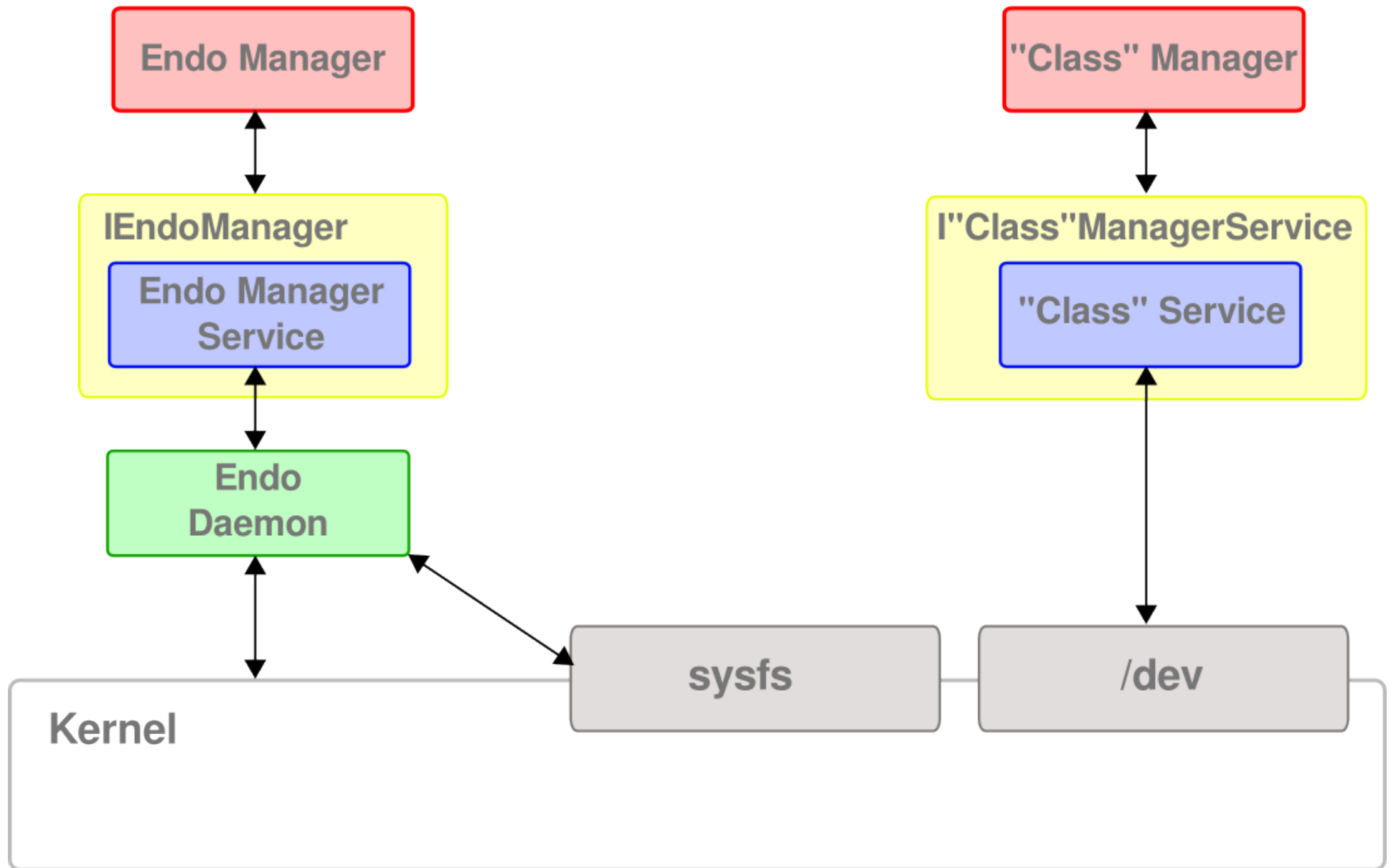
- Vibrator
- Battery
- Audio
- Baseband modem
- Bluetooth
- Camera
- Consumer IR
- Display
- GPS
- Keymaster
- Lights
- NFC
- Sensors
- Wifi

12.4. Bridged PHY Connection Protocols

- USB
- GPIO
- SPI
- UART
- PWM
- I2C
- SDIO



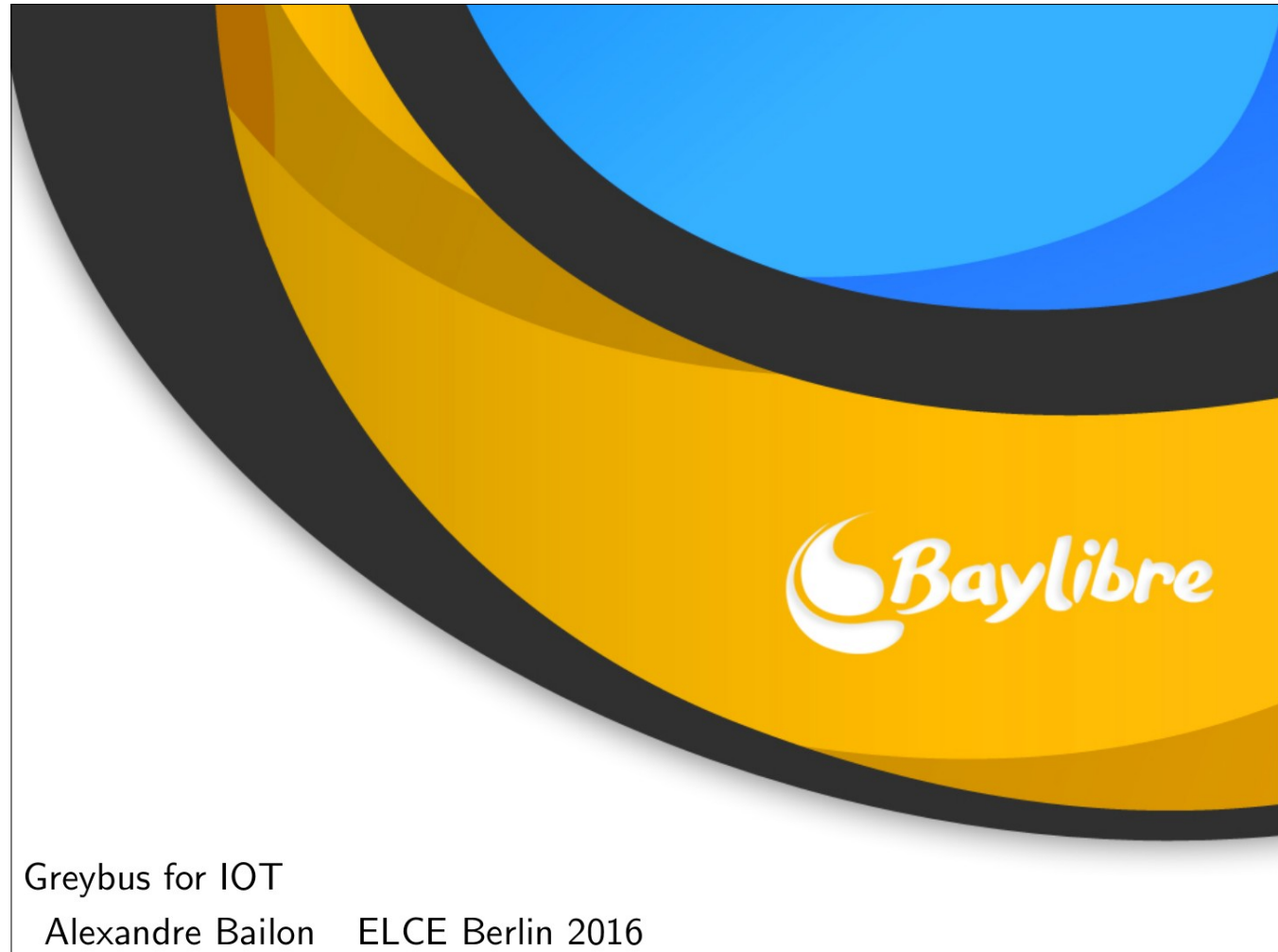
Framework

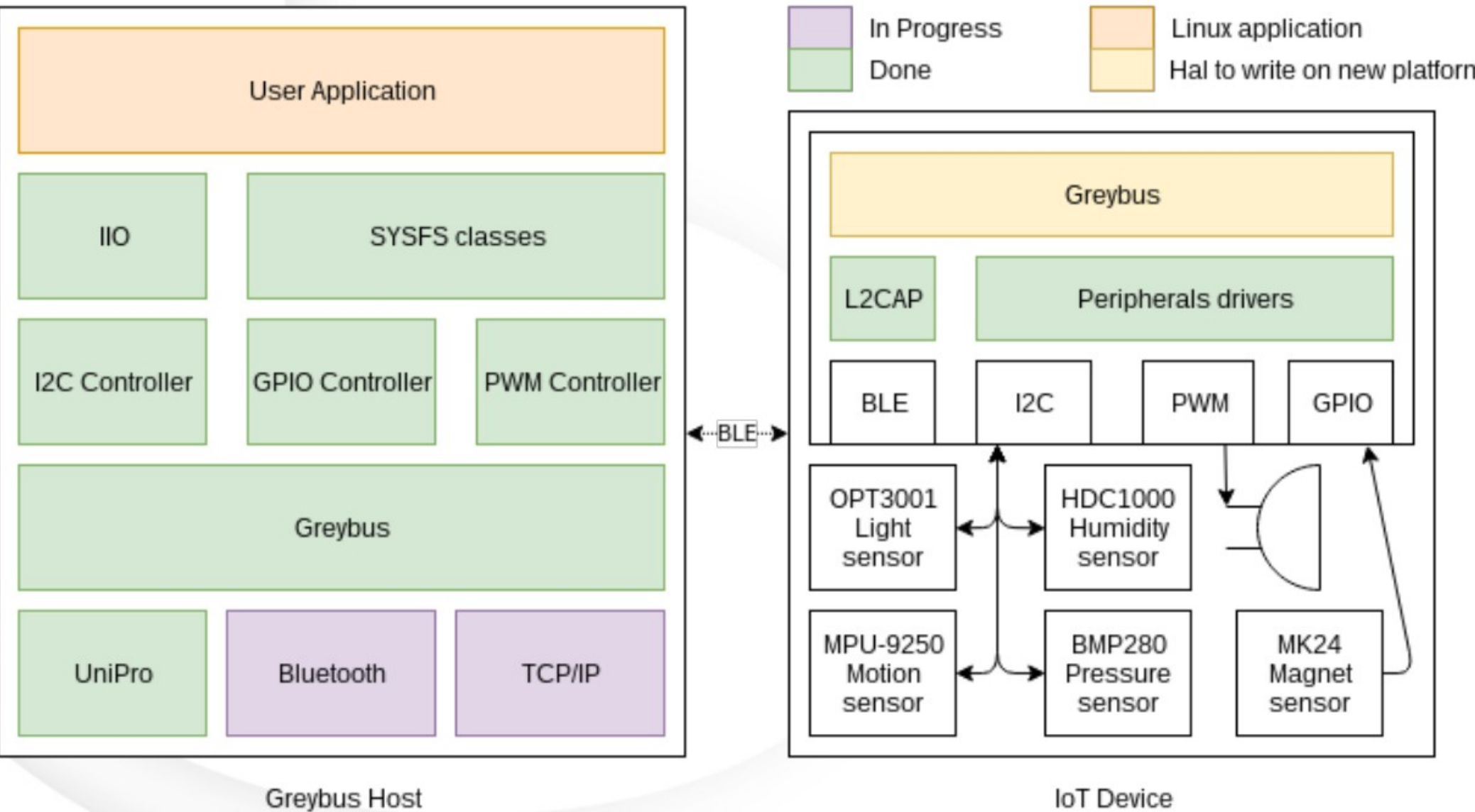


12.5. Greybus for IOT



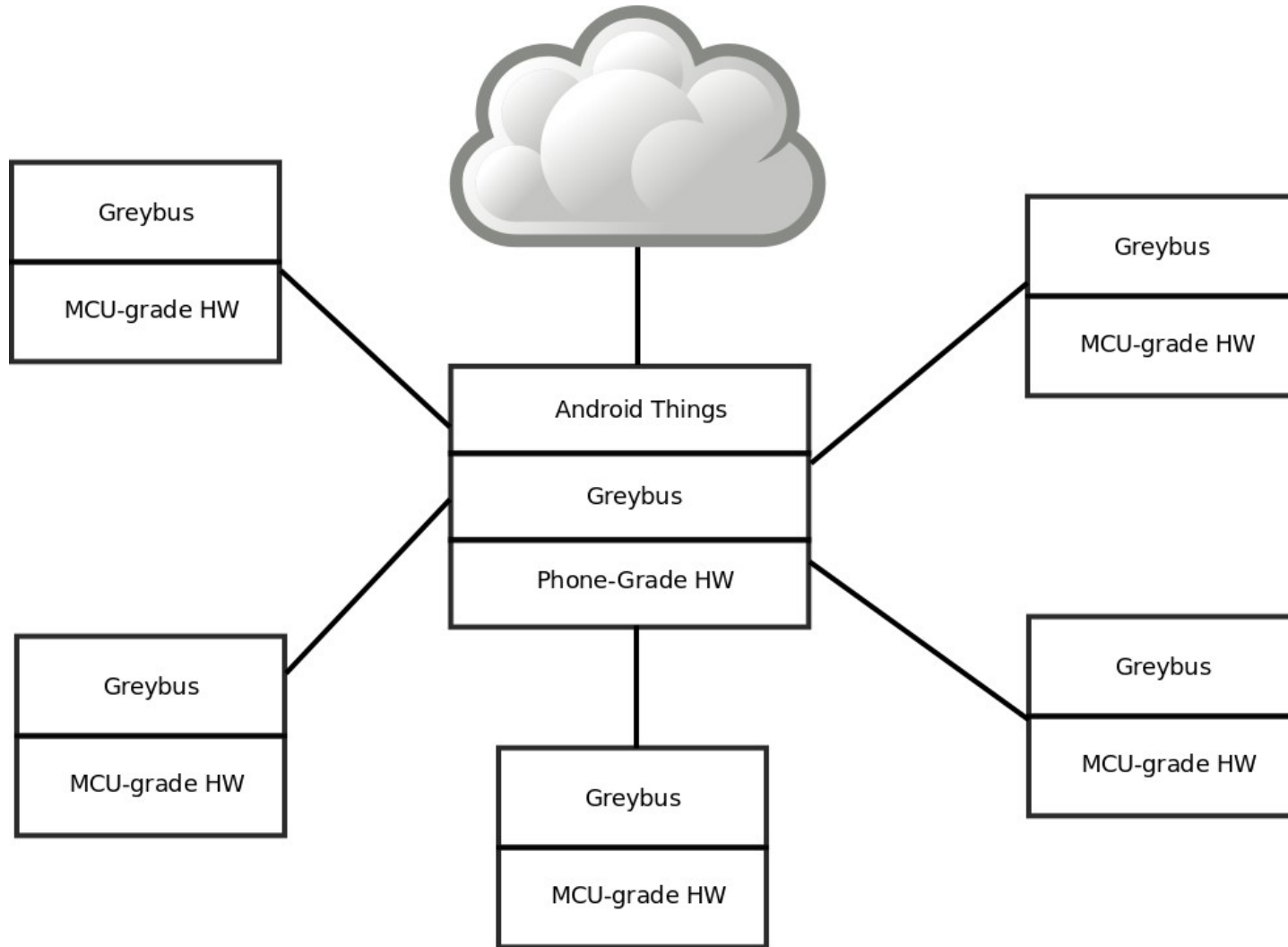
Alexandre Baillon
Baylibre





From Alexandre's ELCE 2016 slides

12.6. Can I haz an Android for real-world IOT ?



Thank you ...

karim.yaghmour@opersys.com

