EE Live!

Running Android atop a proper embedded Linux

Ryan Kuester

Independent Consultant

rkuester@insymbols.com

These slides at **http://insymbols.com/misc/android-atop-linux-ee2014.pdf**

Motivate Android atop Linux

Linux atop the Nexus 7

Android atop Linux

Integrating the two worlds

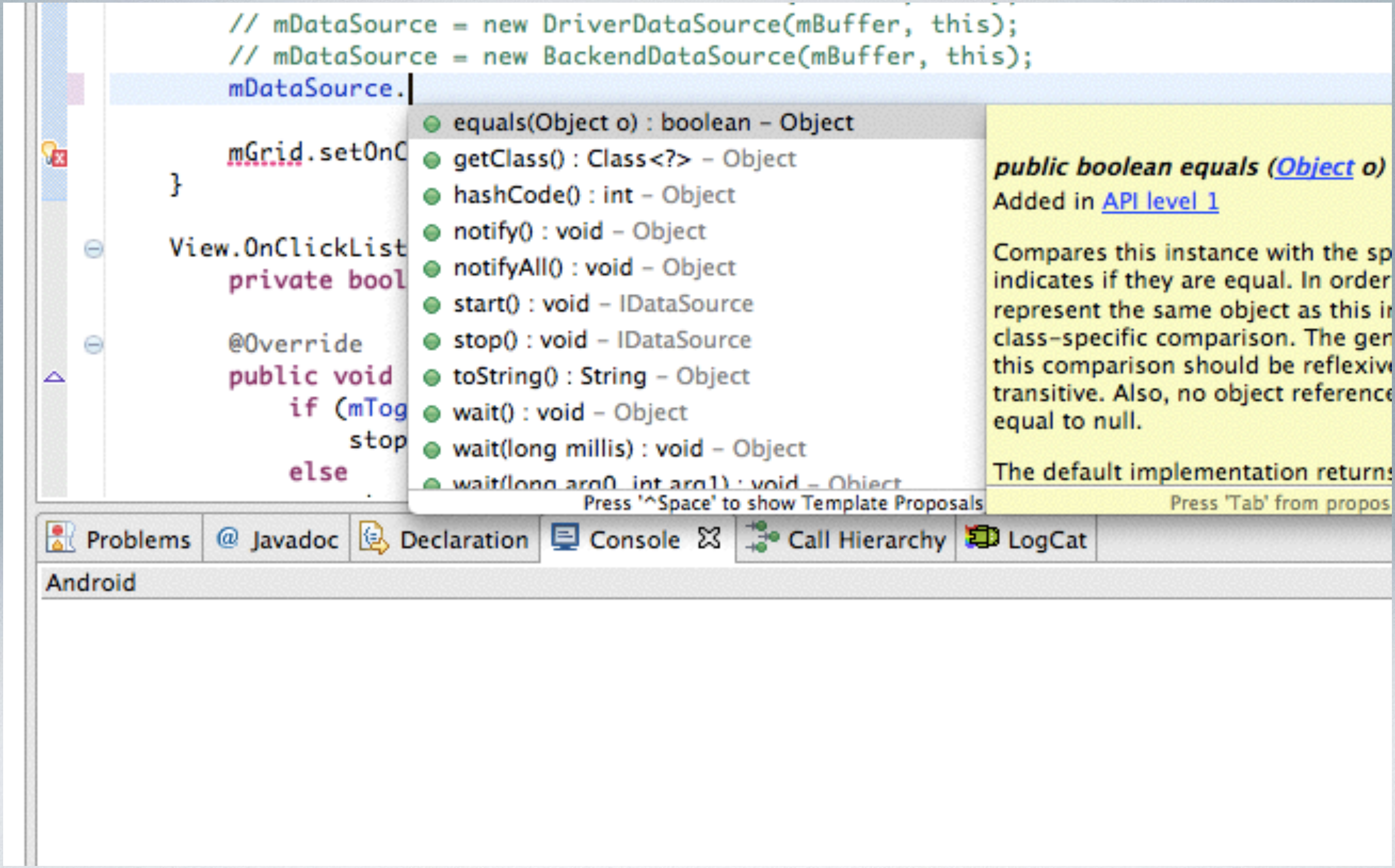
Contribute to discussion of Android's suitability for embedded work

Demonstrate one way of using Android in an embedded system

Motivation: Why Android?

Enormously **successful**

Productive and enjoyable
environment for writing applications
Java has great tools



```
        // mDataSource = new DriverDataSource(mBuffer, this);
        // mDataSource = new BackendDataSource(mBuffer, this);
        mDataSource.

        mGrid.setOnC
    }

    View.OnClickList
        private bool

        @Override
        public void
            if (mTog
                stop
            else
```

```
equals(Object o) : boolean – Object
getClass() : Class<?> – Object
hashCode() : int – Object
notify() : void – Object
notifyAll() : void – Object
start() : void – IDataSource
stop() : void – IDataSource
toString() : String – Object
wait() : void – Object
wait(long millis) : void – Object
wait(long arg0, int arg1) : void – Object
```

Press '^Space' to show Template Proposals

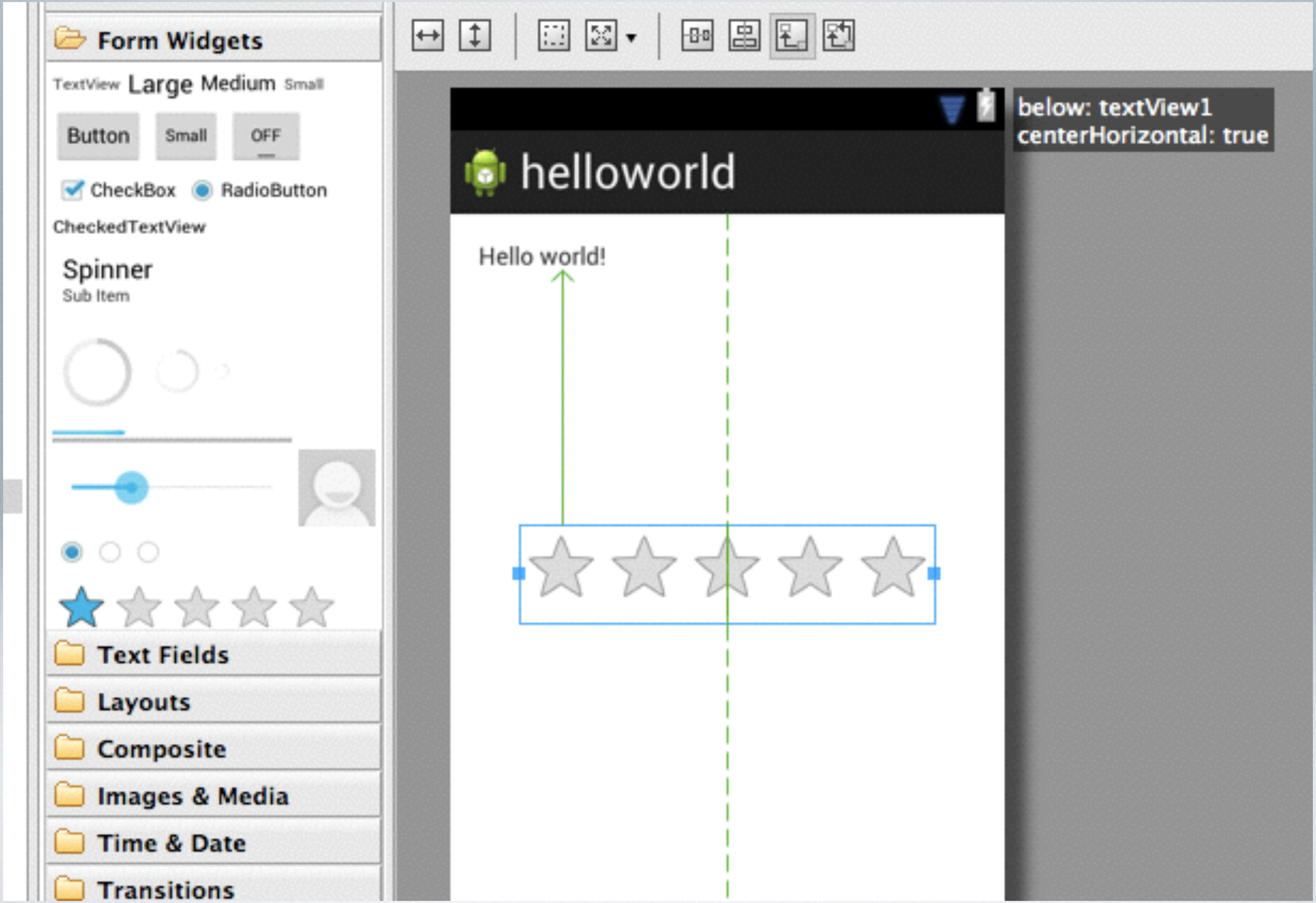
public boolean equals (Object o)

Added in API level 1

Compares this instance with the sp
indicates if they are equal. In order
represent the same object as this i
class-specific comparison. The ger
this comparison should be reflexive
transitive. Also, no object reference
equal to null.

The default implementation returns

Press 'Tab' from propos



below: textView1
centerHorizontal: true

Application framework excellent, e.g., modularity of activities and intents

Users are familiar with Android GUIs

Framework services address common needs: **network, power, audio, management**

Biggest Weakness

Android couples the operating system choice to the GUI choice

Build System

Package Management

Library and program availability

Issues with Android as OS

Software availability with build system

Software compatibility with libc

Software compatibility with libc

see bionic/libc/CAVEATS

not complete POSIX,
only what Android needs, e.g.:

no C++ exceptions

no locales or wide chars

missing functions like getpwd()

Issues with Android as OS

Package management

Field upgrades

Build system

SCM scheme

Issues with Android as OS

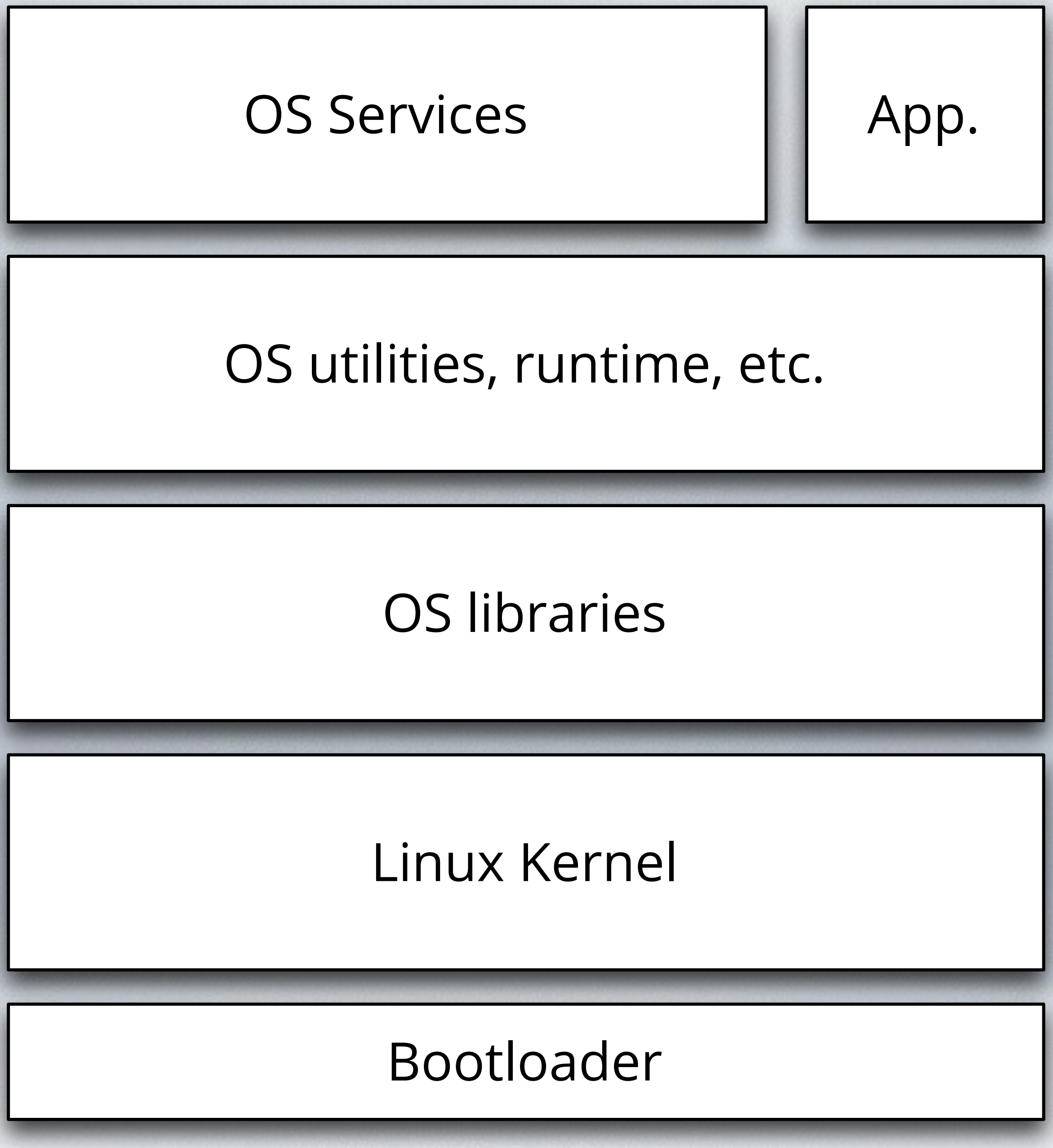
Limited and less-than-open community of platform developers

You need to partition your solution in any case

Running within Android framework is not suitable for some time-sensitive code

Can run code outside of framework, but not well-supported, and library challenges remain

Motivation: So what’s the problem?



OS Services

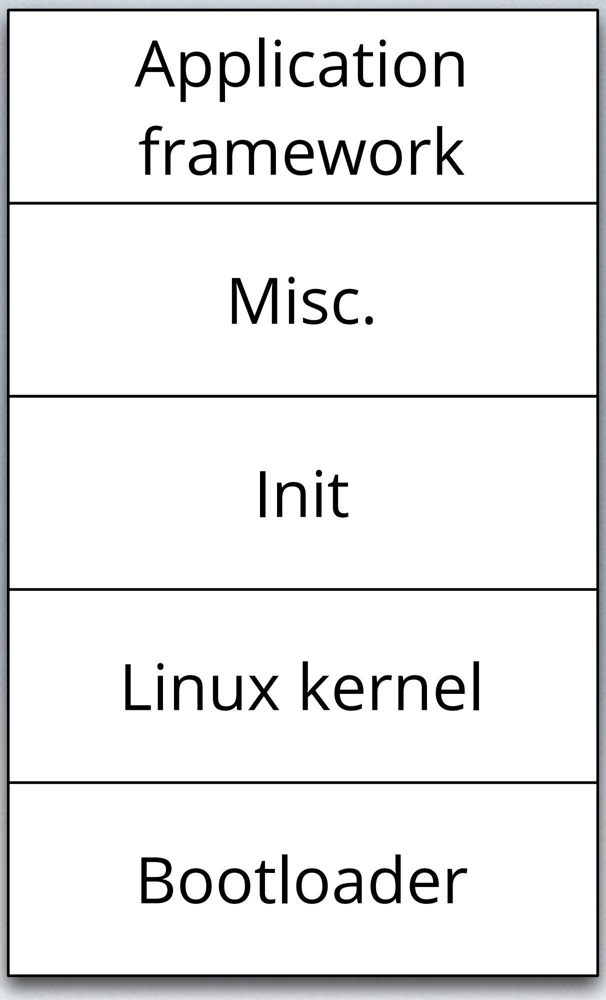
App.

OS utilities, runtime, etc.

OS libraries

Linux Kernel

Bootloader



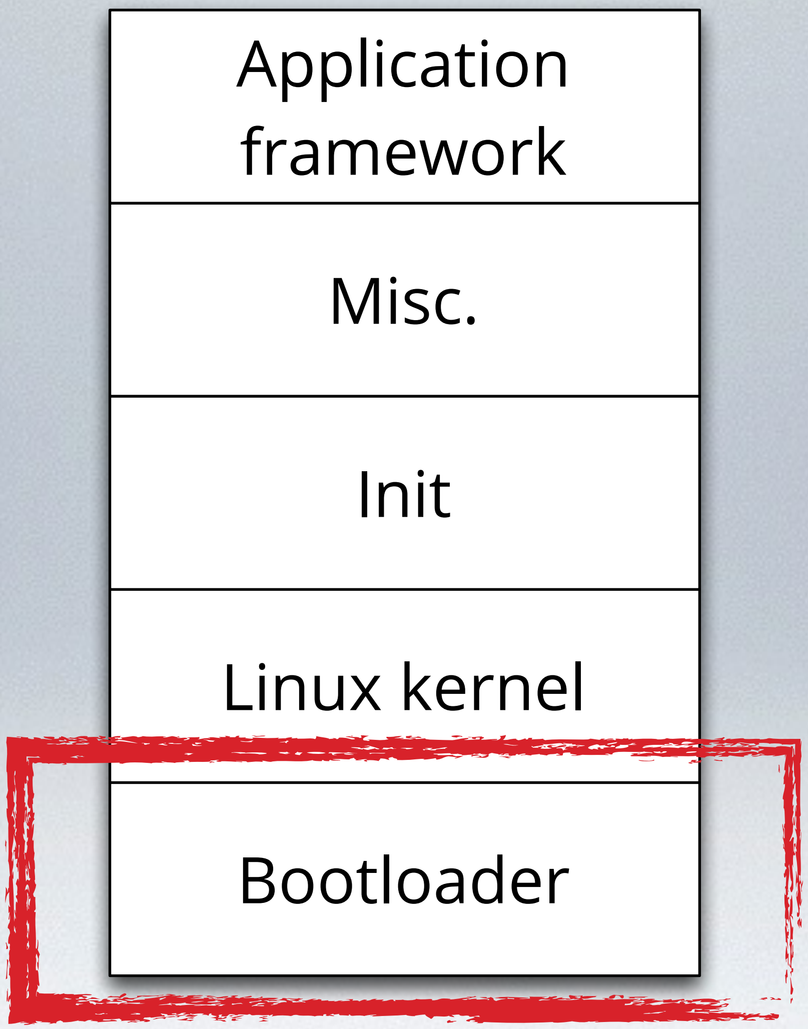
Application framework

Misc.

Init

Linux kernel

Bootloader



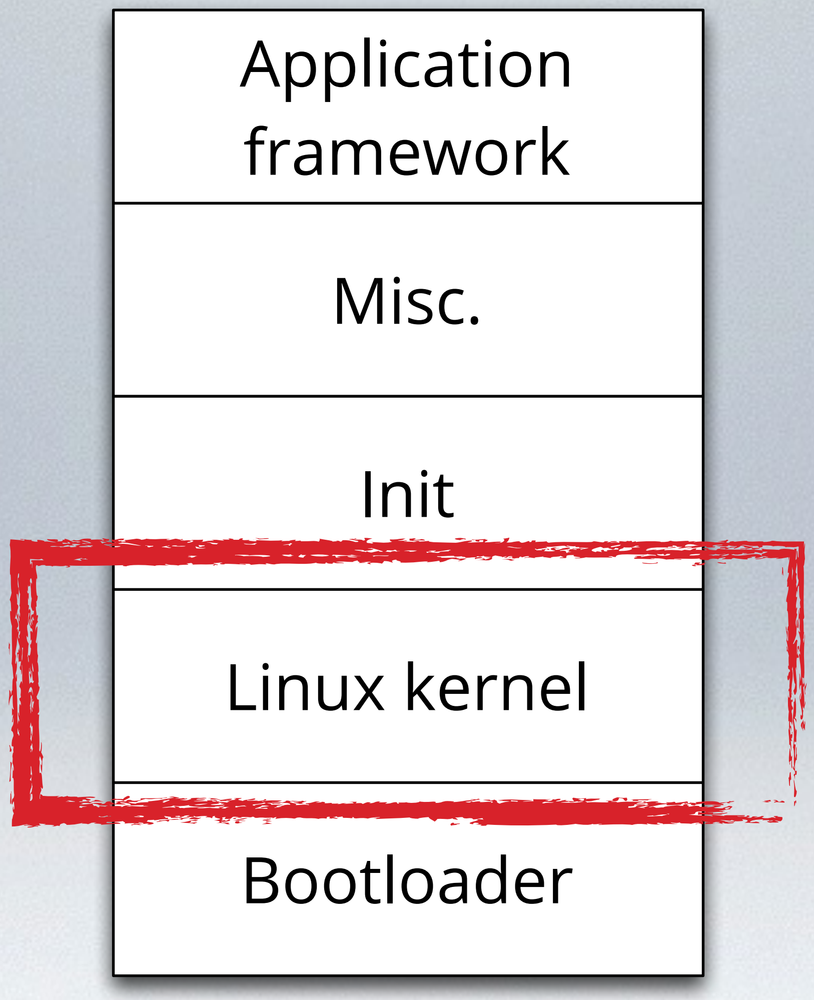
Application framework

Misc.

Init

Linux kernel

Bootloader



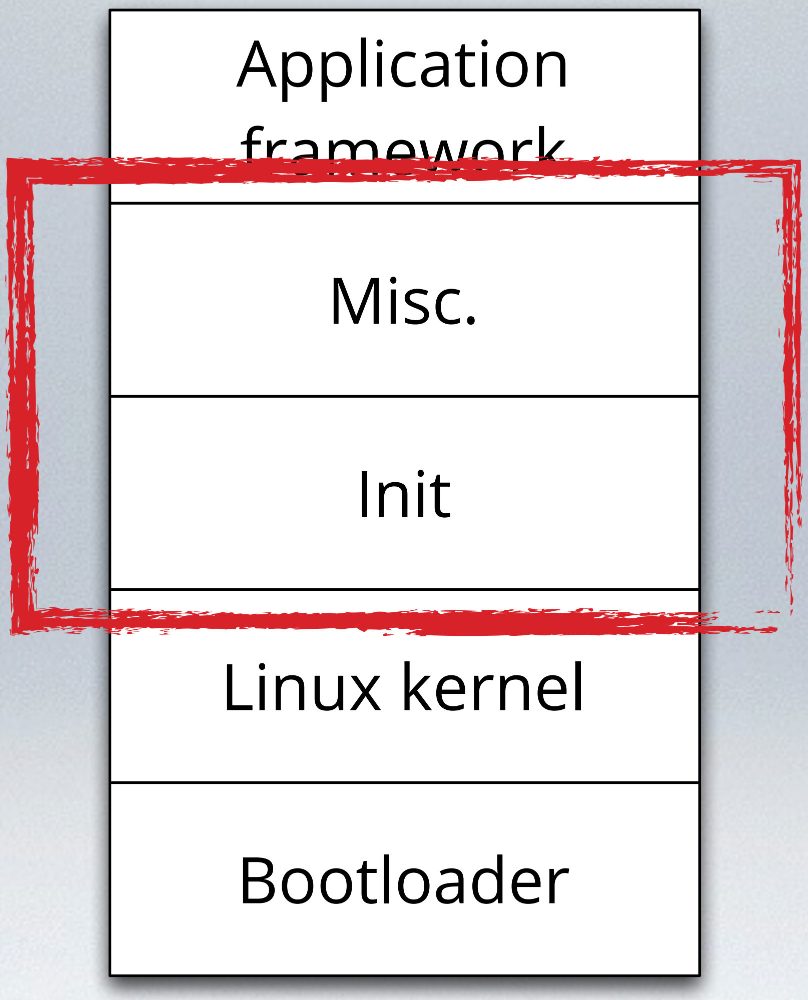
Application framework

Misc.

Init

Linux kernel

Bootloader



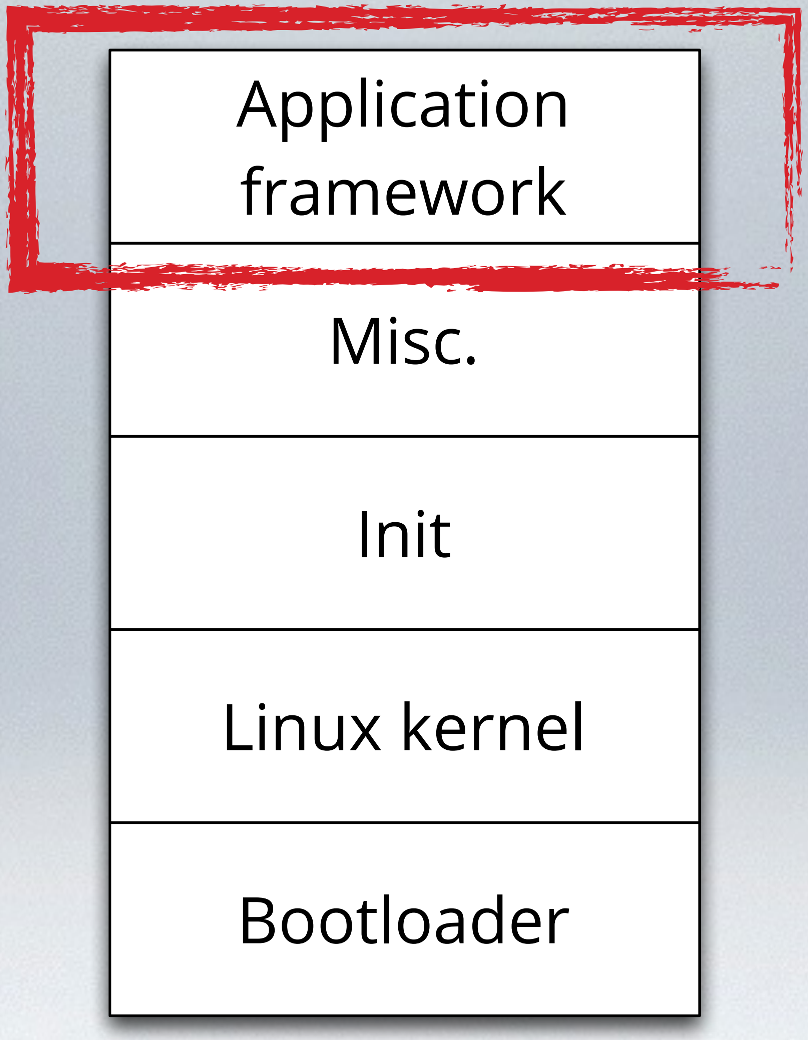
Application framework

Misc.

Init

Linux kernel

Bootloader



Application framework

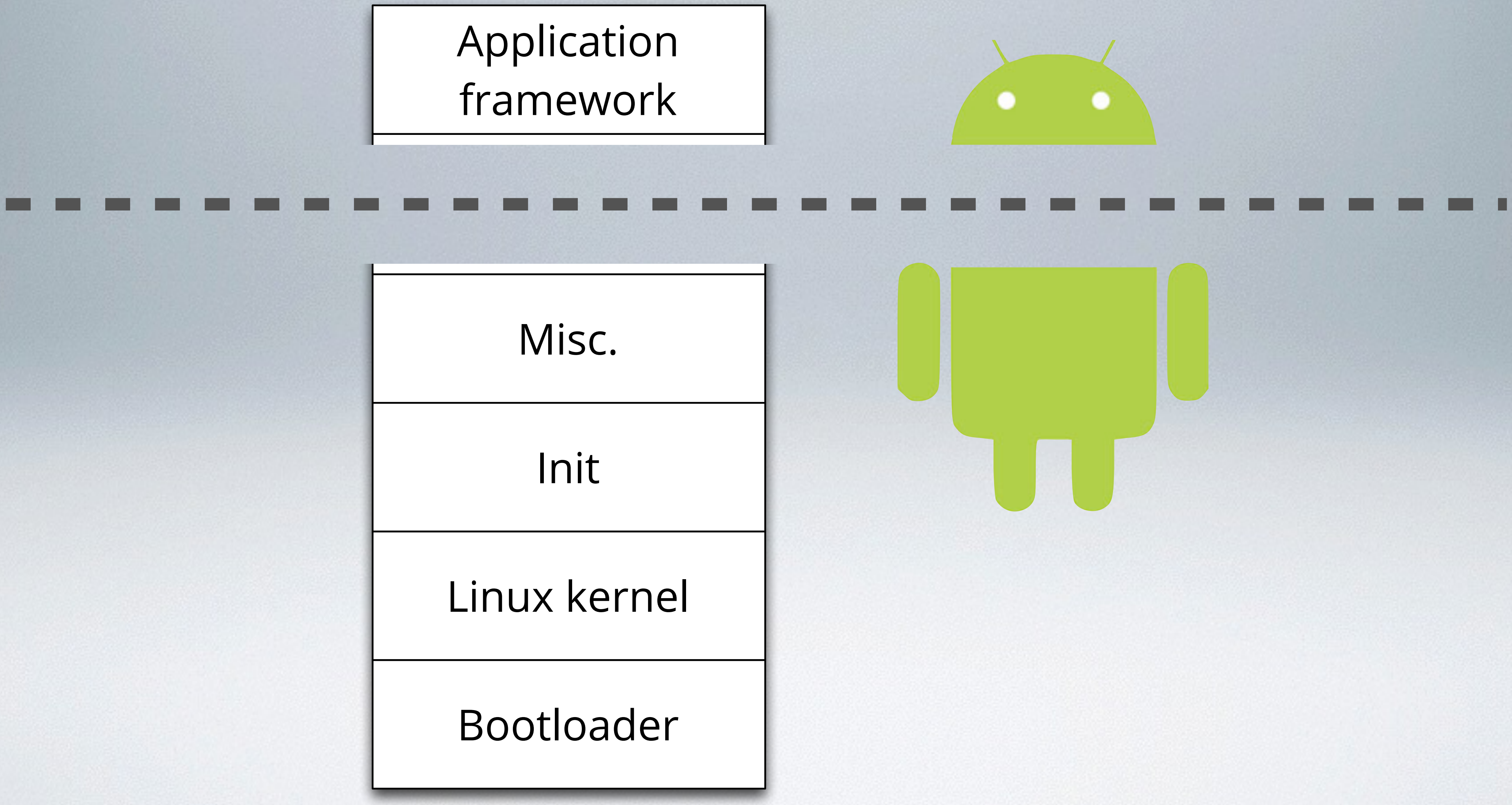
Misc.

Init

Linux kernel

Bootloader

Ideally we could just use the application framework



Application framework

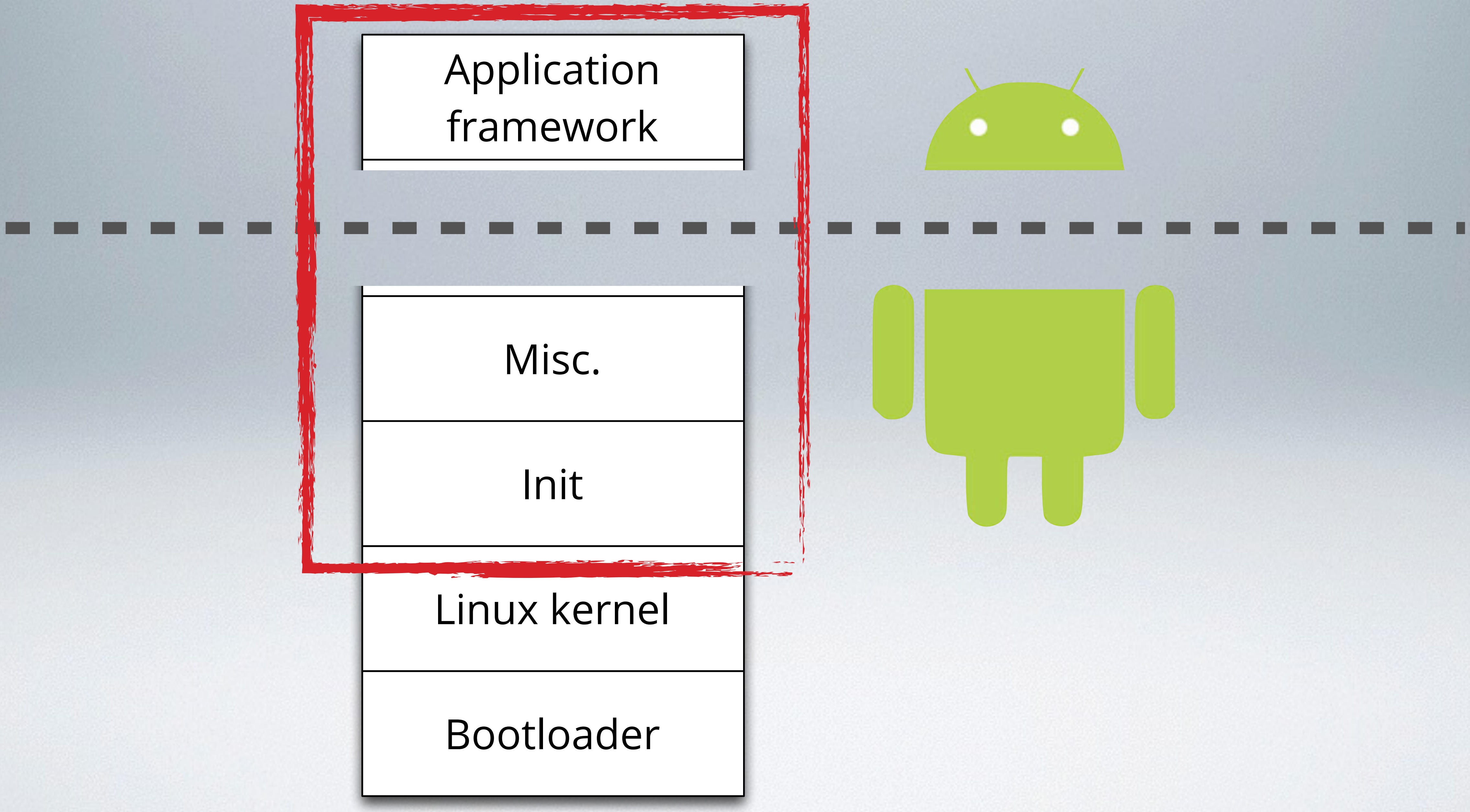
Misc.

Init

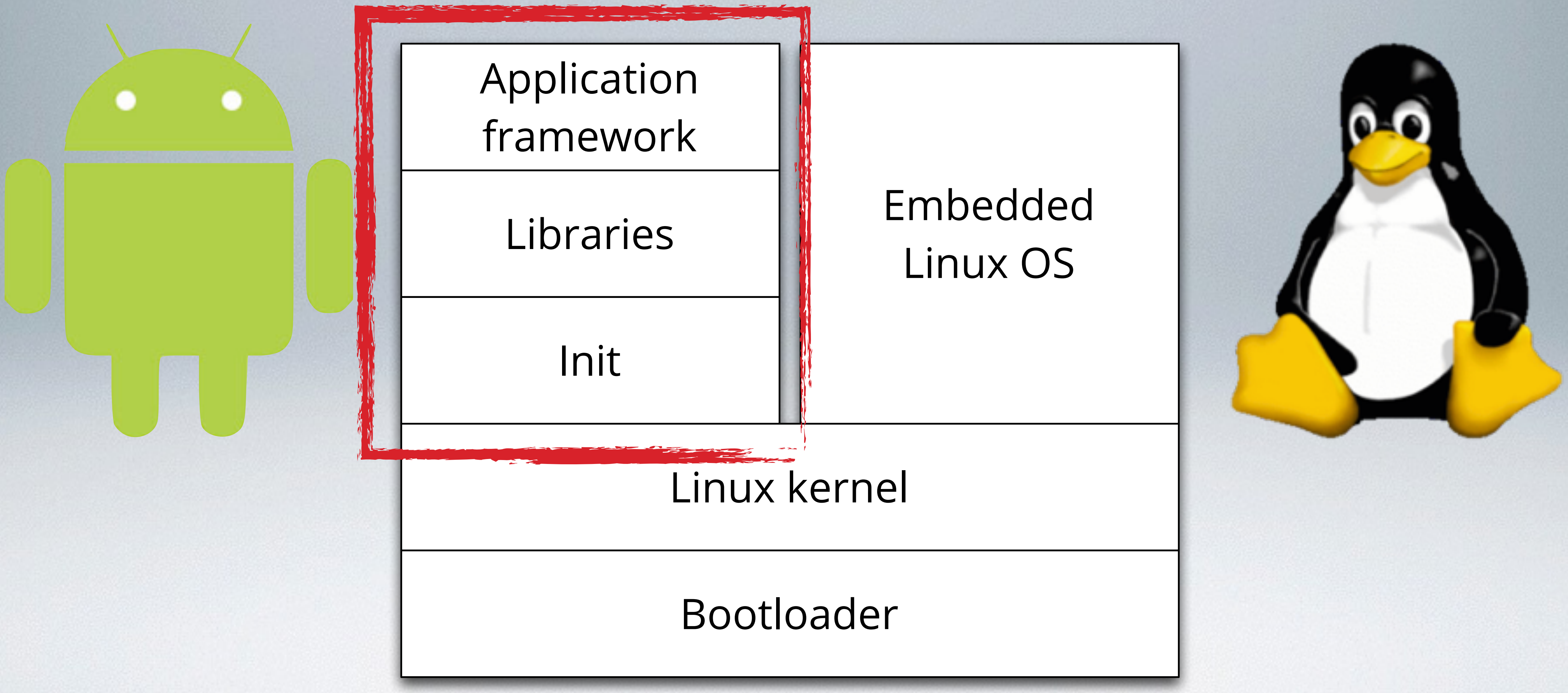
Linux kernel

Bootloader

but Android is architected and delivered as a **tightly-coupled** combination



forcing us to keep it intact and put other software alongside it



Application framework

Libraries

Init

Embedded Linux OS

Linux kernel

Bootloader

Why Linux, again?

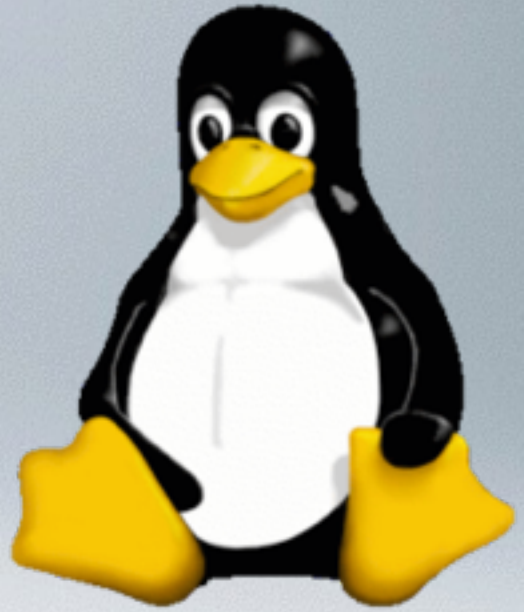
software availability

package management

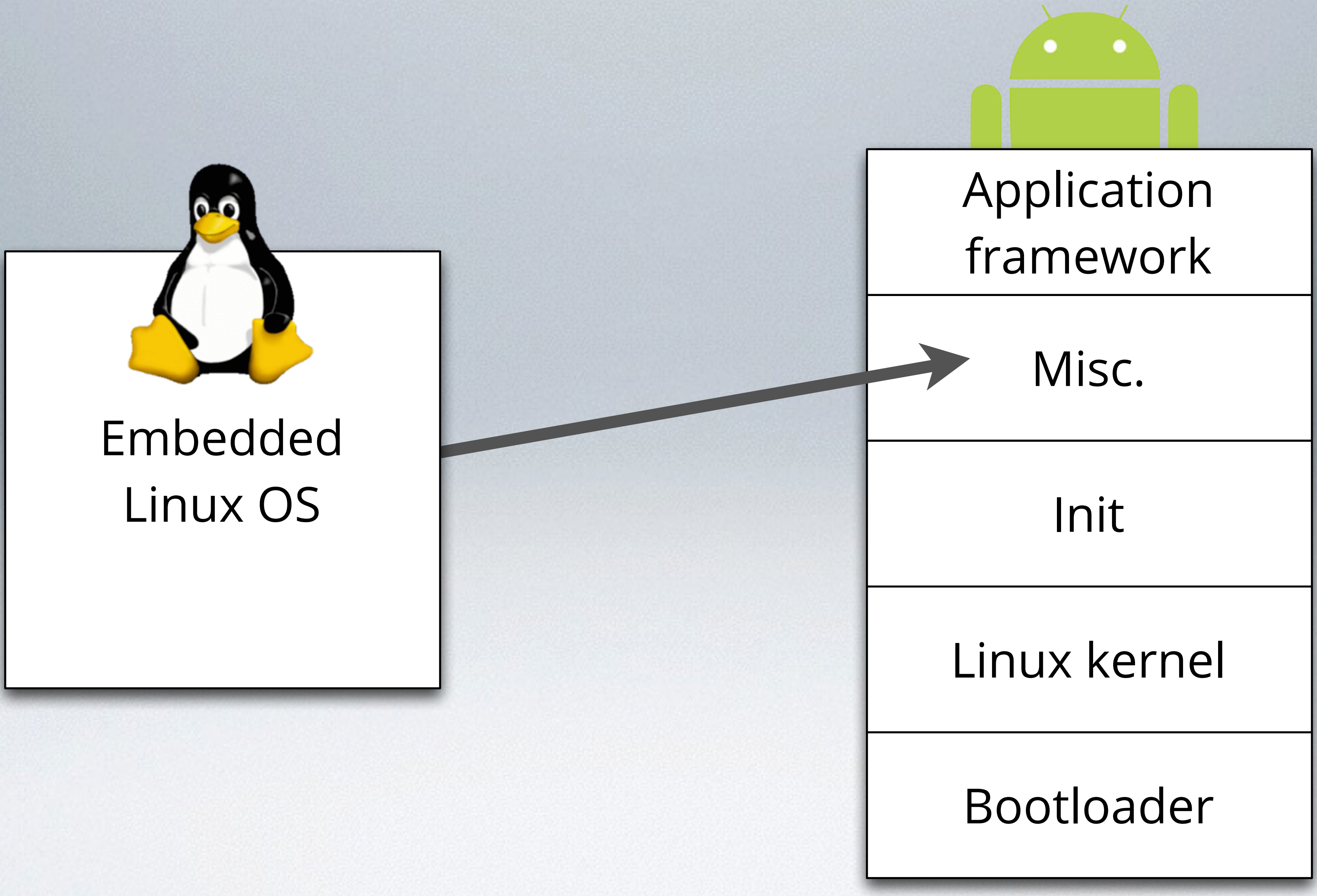
build system

tools

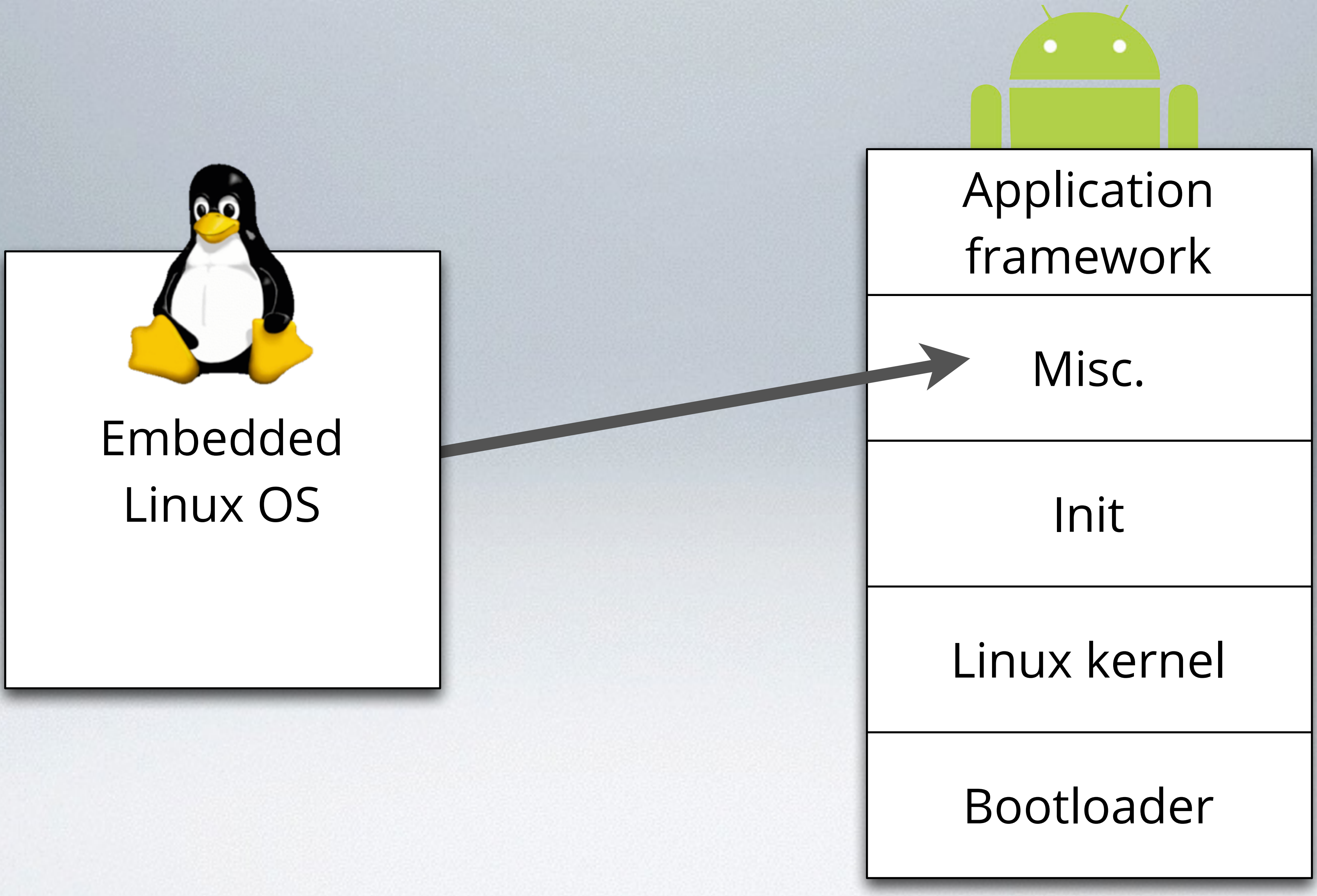
Embedded Linux OS



How? Some suggest injecting pre-built Embedded OS into Android build process



Backwards!—this leaves Android in charge of build process, package management, etc.



Embedded Linux OS

Application framework

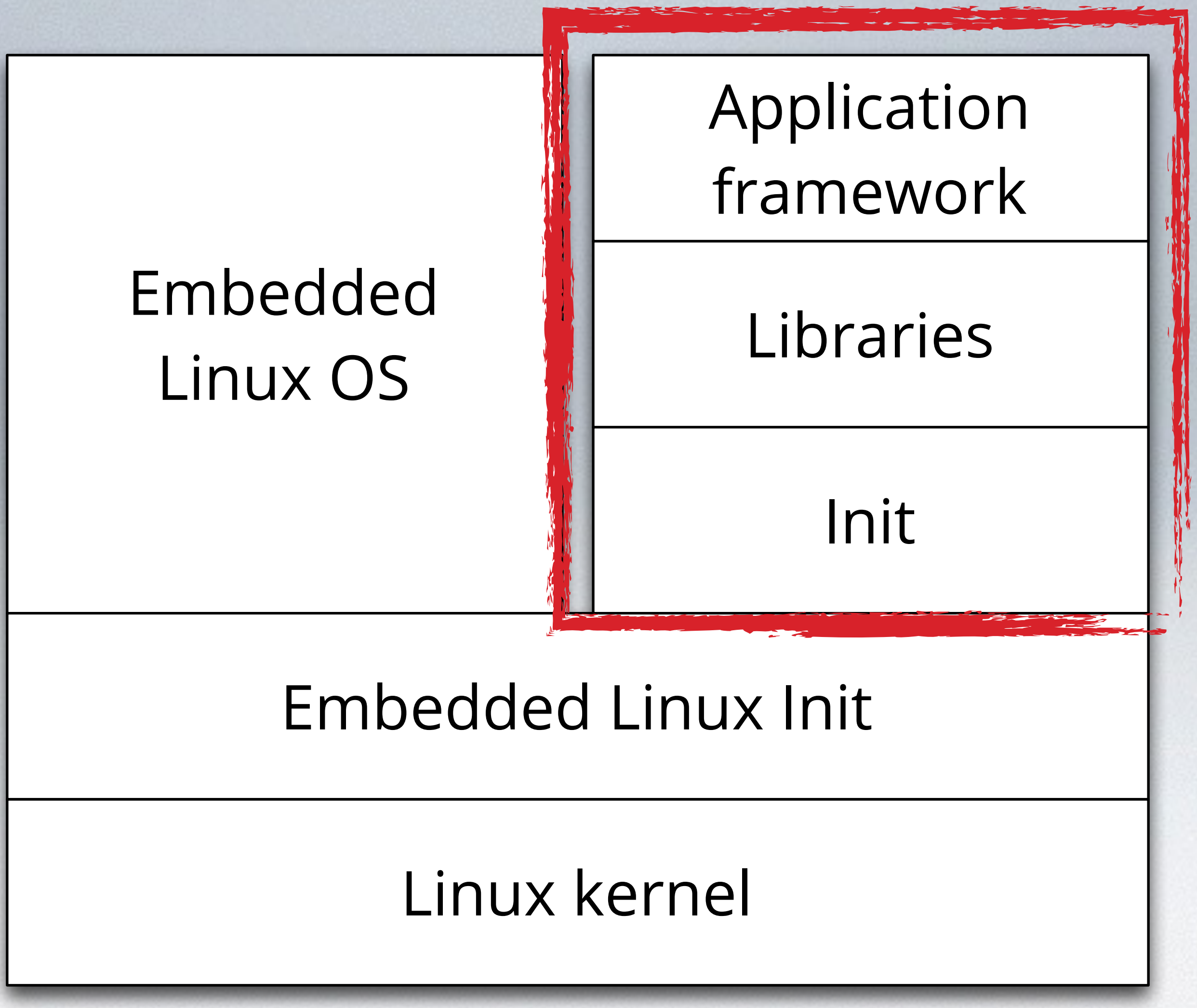
Misc.

Init

Linux kernel

Bootloader

Proposal—treat Android as package within the Embedded Linux OS



Embedded Linux OS

Application framework

Libraries

Init

Embedded Linux Init

Linux kernel

Demonstration of this concept applied to the Nexus 7

Linux atop the Nexus 7

Pragma**tux**

A community-developed Linux distribution for embedded systems

http://pragmatux.org

Pragma**tux**

Workstation environment

Target device operating system

Deployment management system

Developer ecosystem

Pragmatux

Utilizes tools, concepts from the Debian Project

Isn’t “Debian”, but has a similar look and feel

debian emdebian

Prag**ma**tux

package management and repository tools

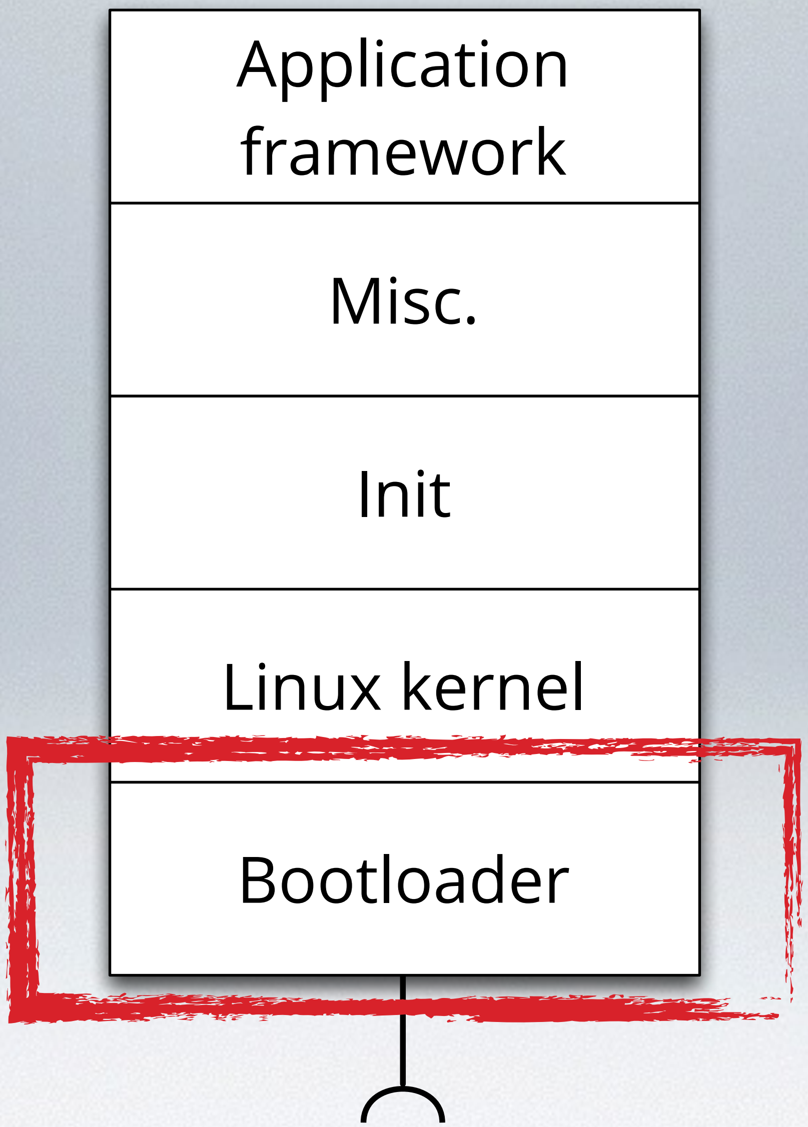
cross-toolchains

minimized packages

keeps binary compatibility

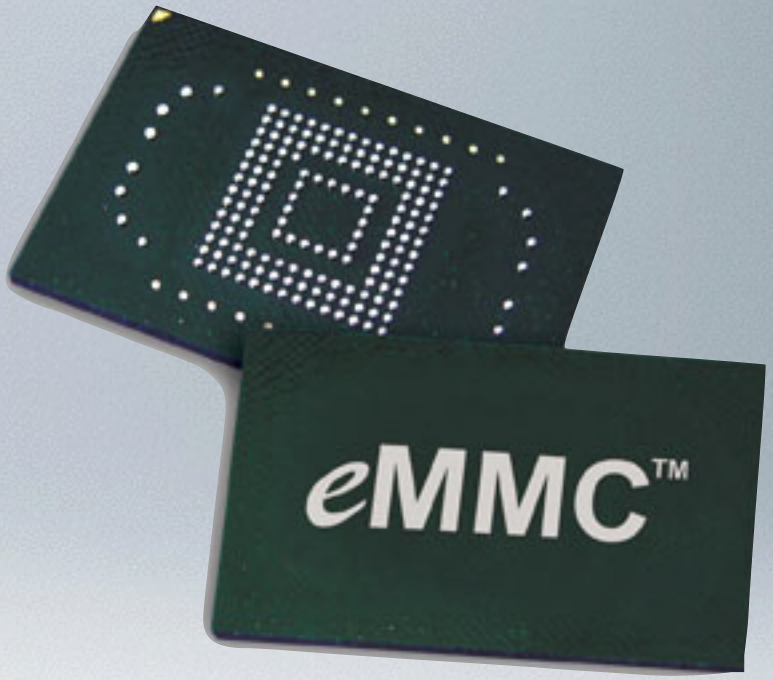
debian emdebian

How Android device storage is partitioned

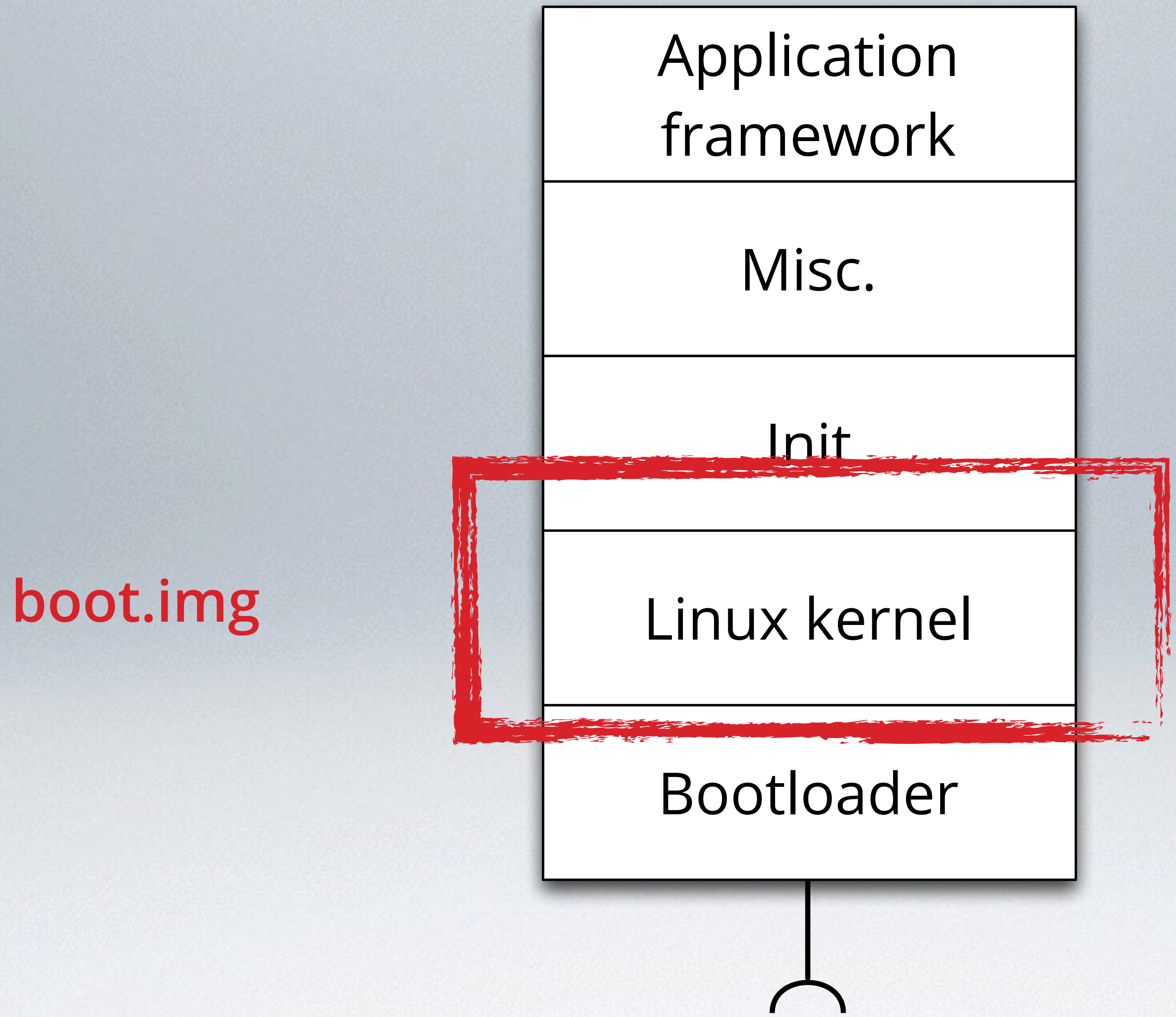


Normally don't touch this

```
# fastboot flash $IMAGE
```



How Android device storage is partitioned



Application framework

Misc.

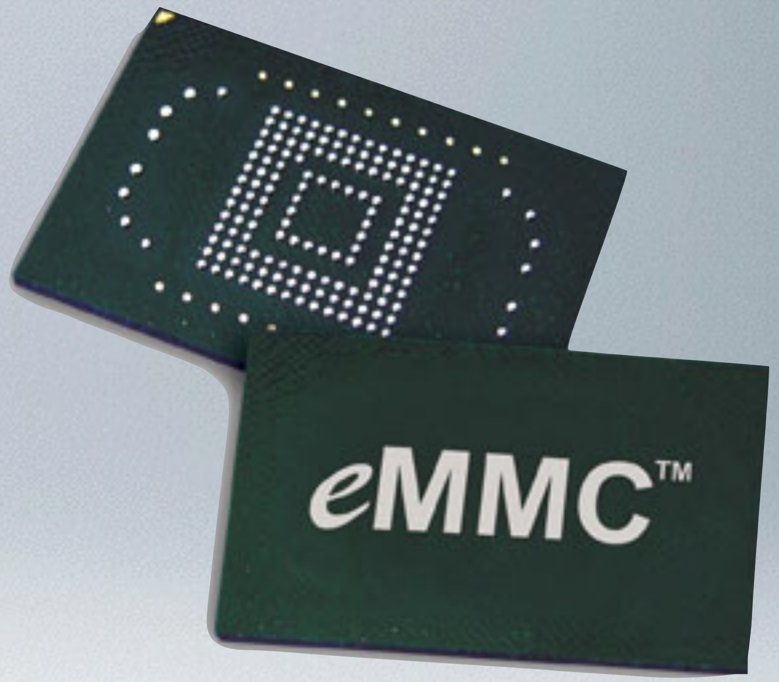
Init

Linux kernel

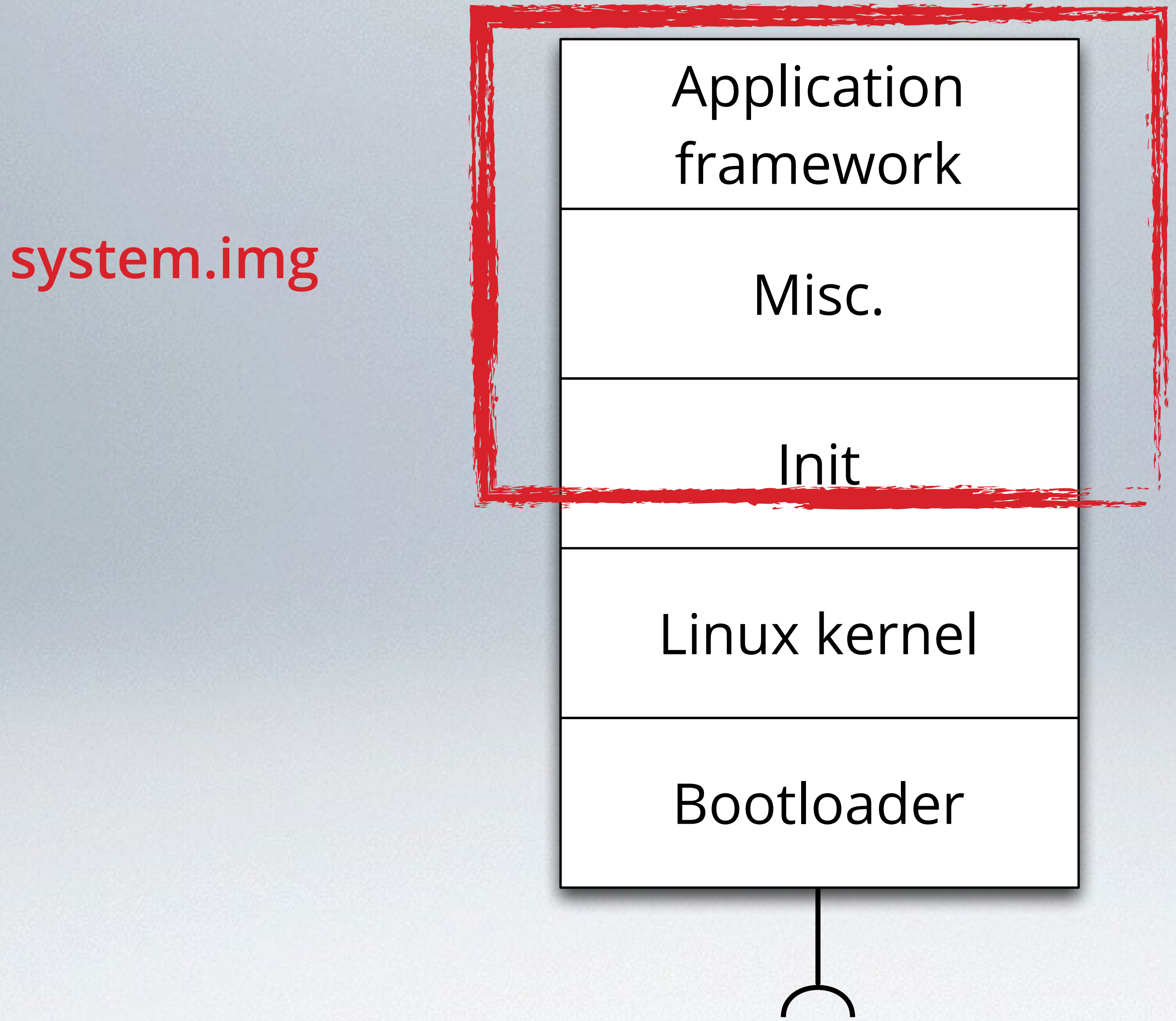
Bootloader

boot.img

```
# fastboot flash $IMAGE
```



How Android device storage is partitioned



system.img

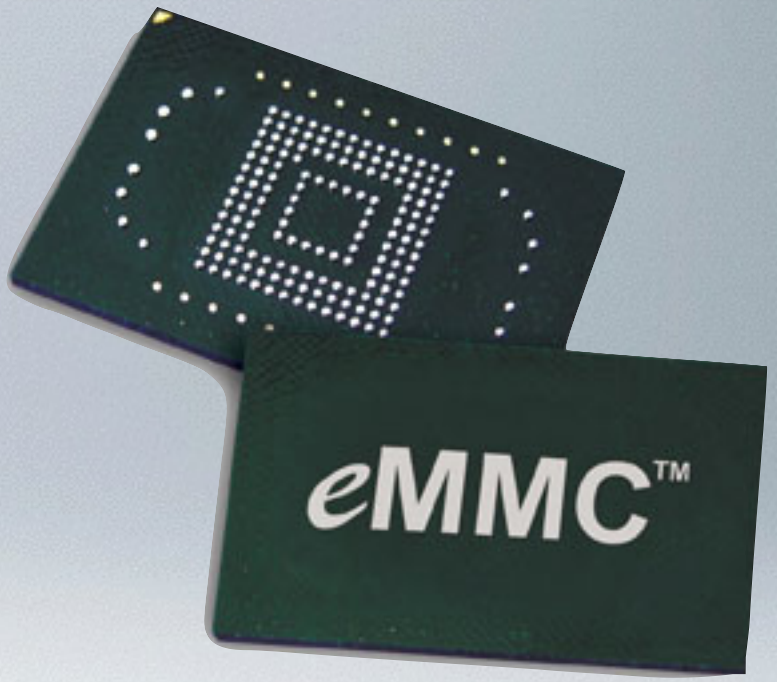
Application framework

Misc.

Init

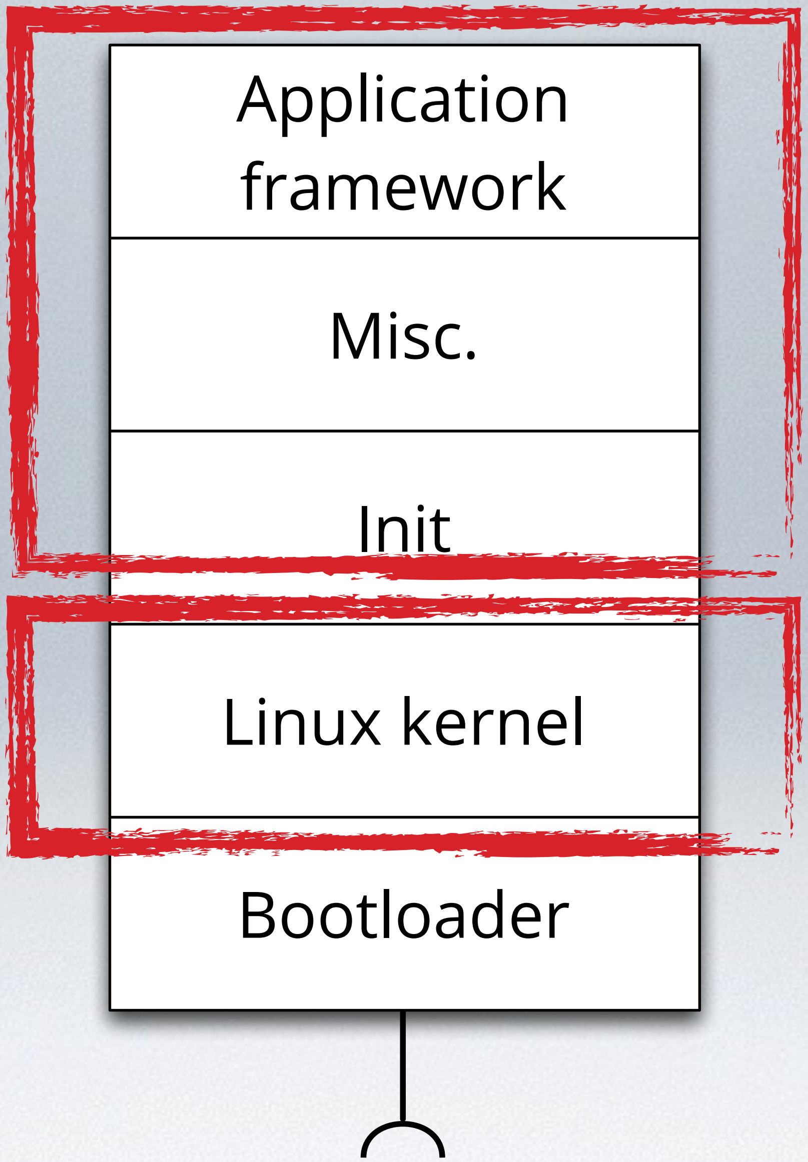
Linux kernel

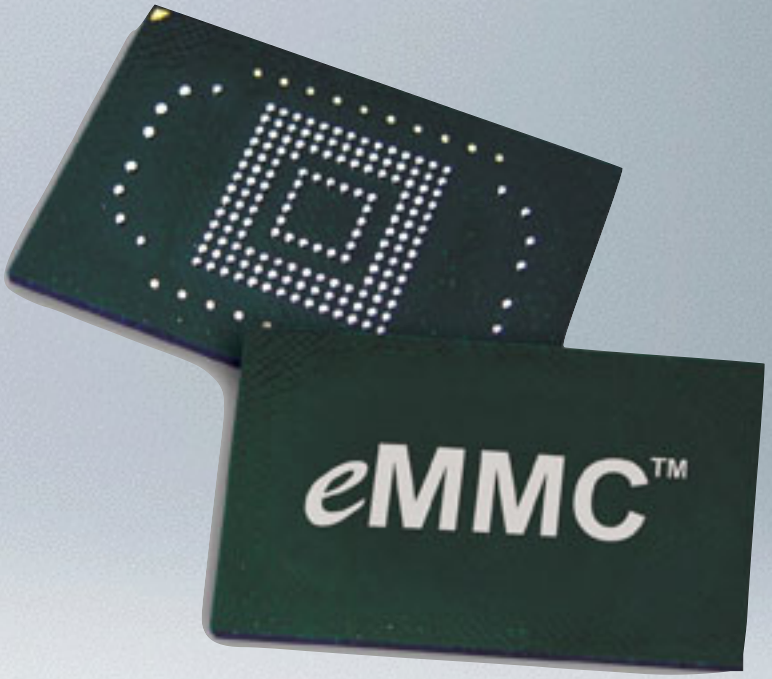
Bootloader



```
# fastboot flash $IMAGE
```

New kernel, new system image (now in userdata partition)





```
# fastboot flash $IMAGE
```

Demo: Building and loading images

Nexus 7 **kernel** source:

https://github.com/pragmatux/linux-n7-2013

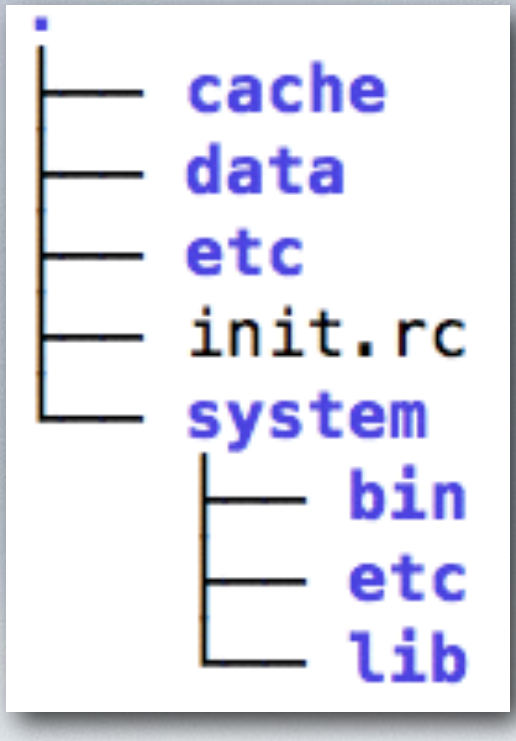
Originates from AOSP’s Qualcomm kernel:
https://android.googlesource.com/kernel/msm.git

Android package source:
https://github.com/rkuester/android-n7-2013-wifi-aosp

Originates with the binary output of an
AOSP build for the Nexus 7, see:
http://source.android.com/source/building.html

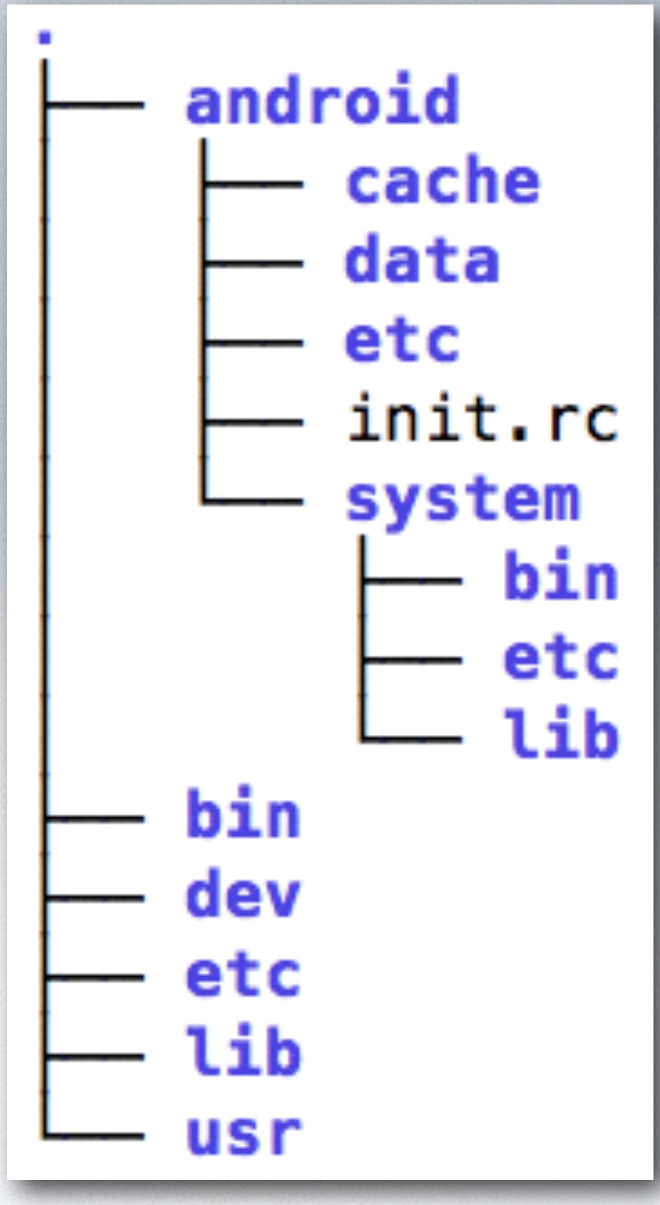
Android filesystem in chroot

Android



```
.
├── cache
├── data
├── etc
├── init.rc
└── system
    ├── bin
    ├── etc
    └── lib
```

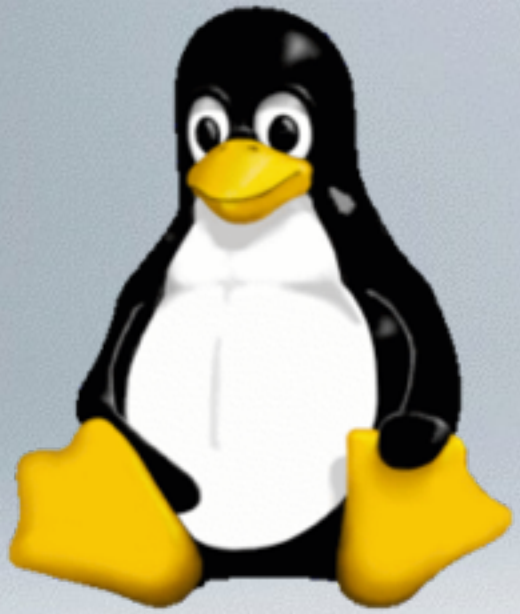
Embedded OS

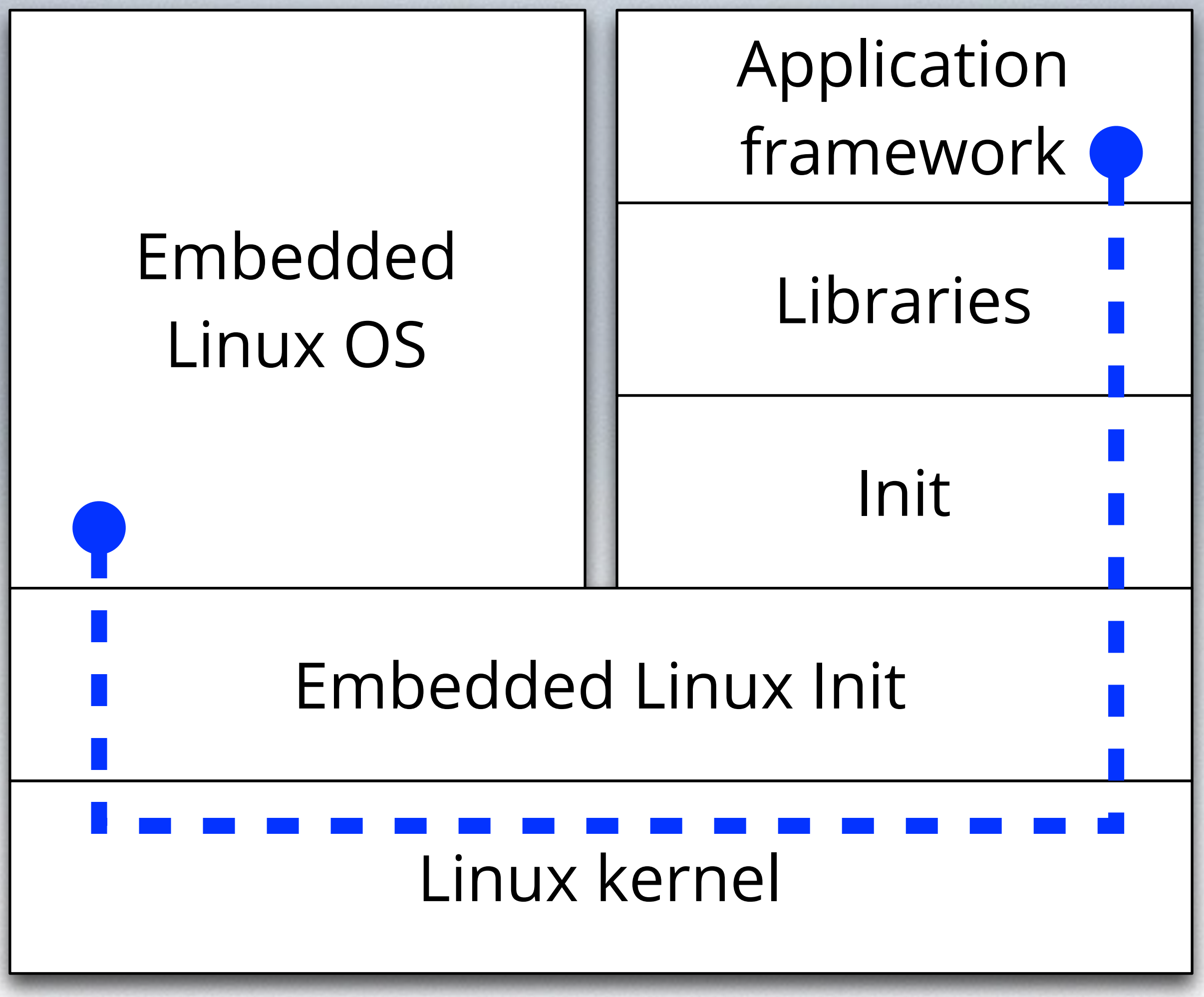


```
.
├── android
│   ├── cache
│   ├── data
│   ├── etc
│   ├── init.rc
│   └── system
│       ├── bin
│       ├── etc
│       └── lib
├── bin
├── dev
├── etc
├── lib
└── usr
```

Demo: Integrating and Android app and a Linux daemon

Communication via kernel still works as always





Embedded Linux OS

Application framework

Libraries

Init

Embedded Linux Init

Linux kernel

EE Live!

Running Android atop a proper embedded Linux

Ryan Kuester
Independent Consultant
rkuester@insymbols.com

These slides at **http://insymbols.com/misc/android-atop-linux-ee2014.pdf**