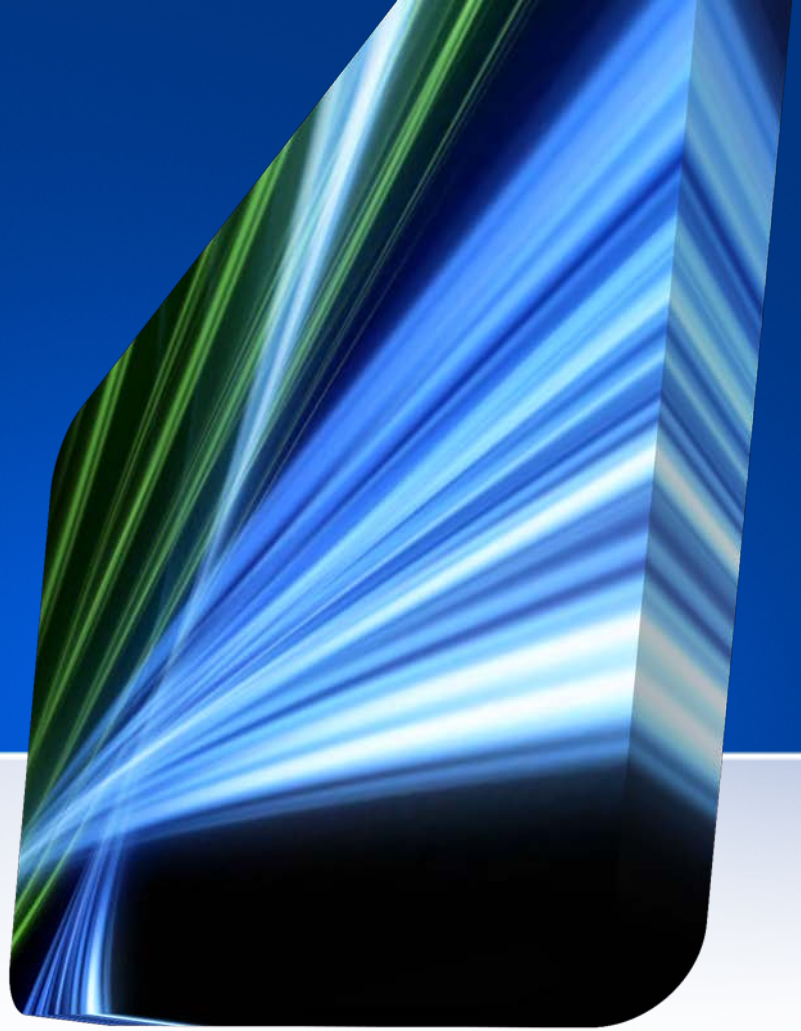
Android Security



Giovanni Russello

g.russello@auckland.ac.nz

The Smartphomania

- Total Smartphone sales 1Q13: 210 million units
- 43% increase from the same period in 2012
- Of these devices, 156 million units are Android phones (75%)!

Source Gartner
http://www.gartner.com/newsroom/id/2482816

Android Sales

- Android represents 75% of the total smartphone market
- Apple is just 18%
- The rest (Windows/RIM/Symbian) gets just crumbs

Google Android

- First Android handset released in 2008
- Open source
- Strict Sandboxing
- Java Dalvik VM
- Java Apps
- Lightweight code signing
- Permission Framework
- App Market (more 700K apps)

It is for free!!!

- Android is for free from Google
- You can get as well!
  - http://source.android.com/
- Vendors range from Samsung to small Chinese/Russian firms
  - Xunrui Communications: you can get one for $65

Fragmentation Problem

- Vendors customise the OS in their devices
  - Usually a lot of rubbish apps
  - The worst: Samsung apps also leak privileges
    - See: http://randomthoughts.greyhats.it/2013/03/owning-samsung-phones-for-fun-but-with.html
- However seldom a vendor does push any updates
  - Some devices can be 2 or 3 version behind
    - See: http://theunderstatement.com/post/11982112928/android-orphans-visualizing-a-sad-history-of-support

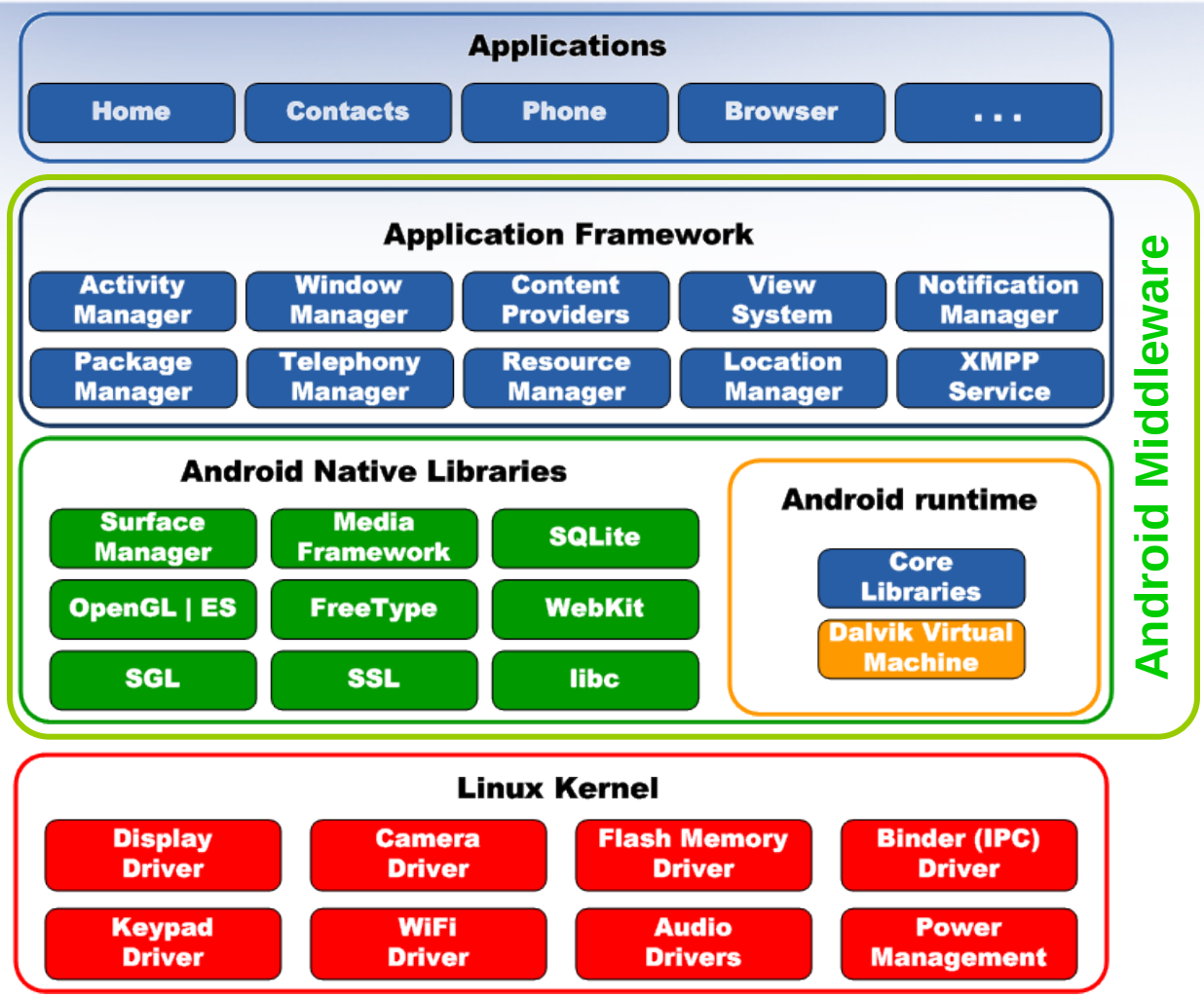
Heading for troubles

- The lack of support can lead to vulnerabilities
- Often vendors just ignore vulnerabilities on their software
- Apple does a much better job:
  - One single piece of hardware
  - One single software image

What is under the hood?

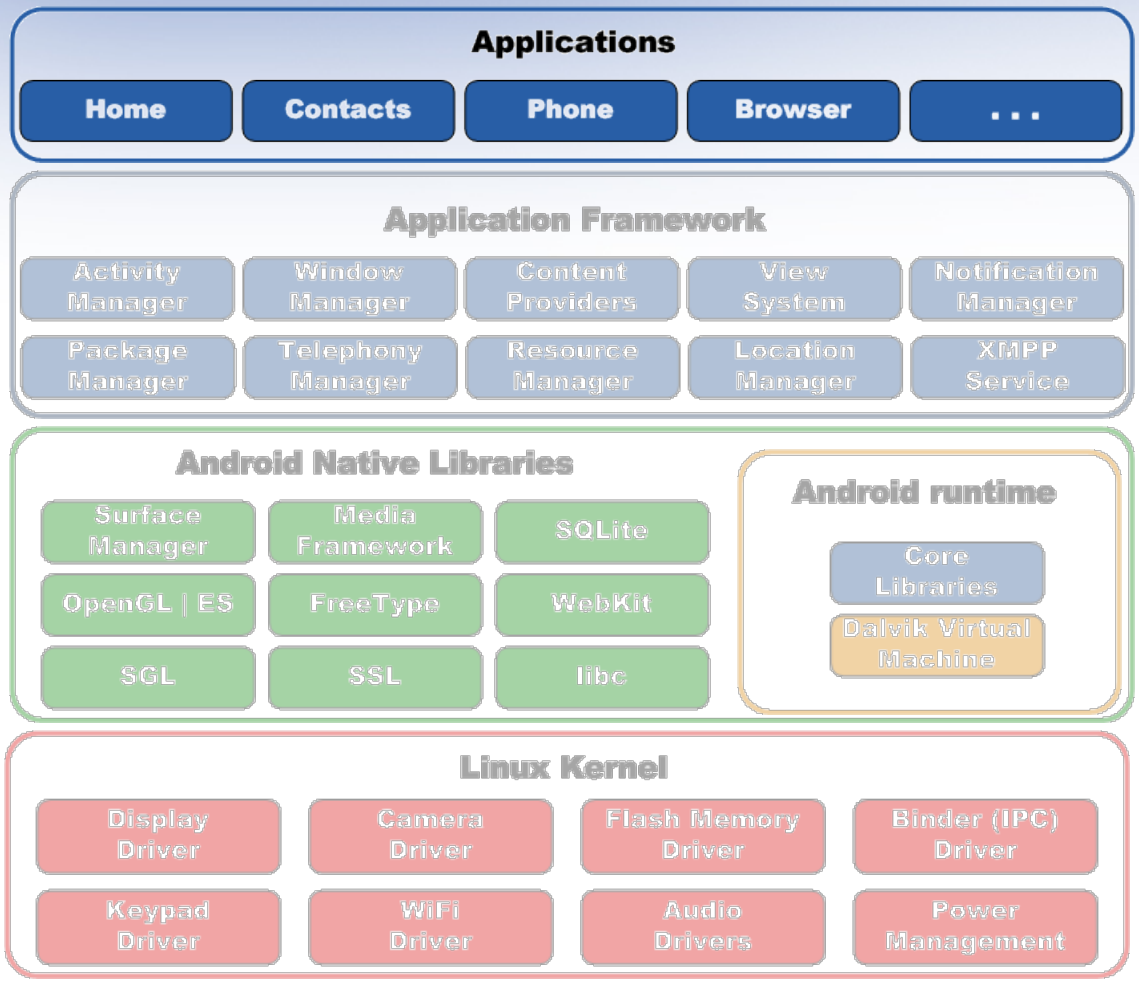
- Android is actually a middleware
- It sits between a Linux kernel and a set of API
- Android apps are mainly written in Java
  - Only Android apps can run on Android
- Through the Android API, apps can access all the device resources
  - It provides apps a rich set of information

Android View



Android is a set of programs for mobile devices that includes operating system, middleware and core applications

Applications



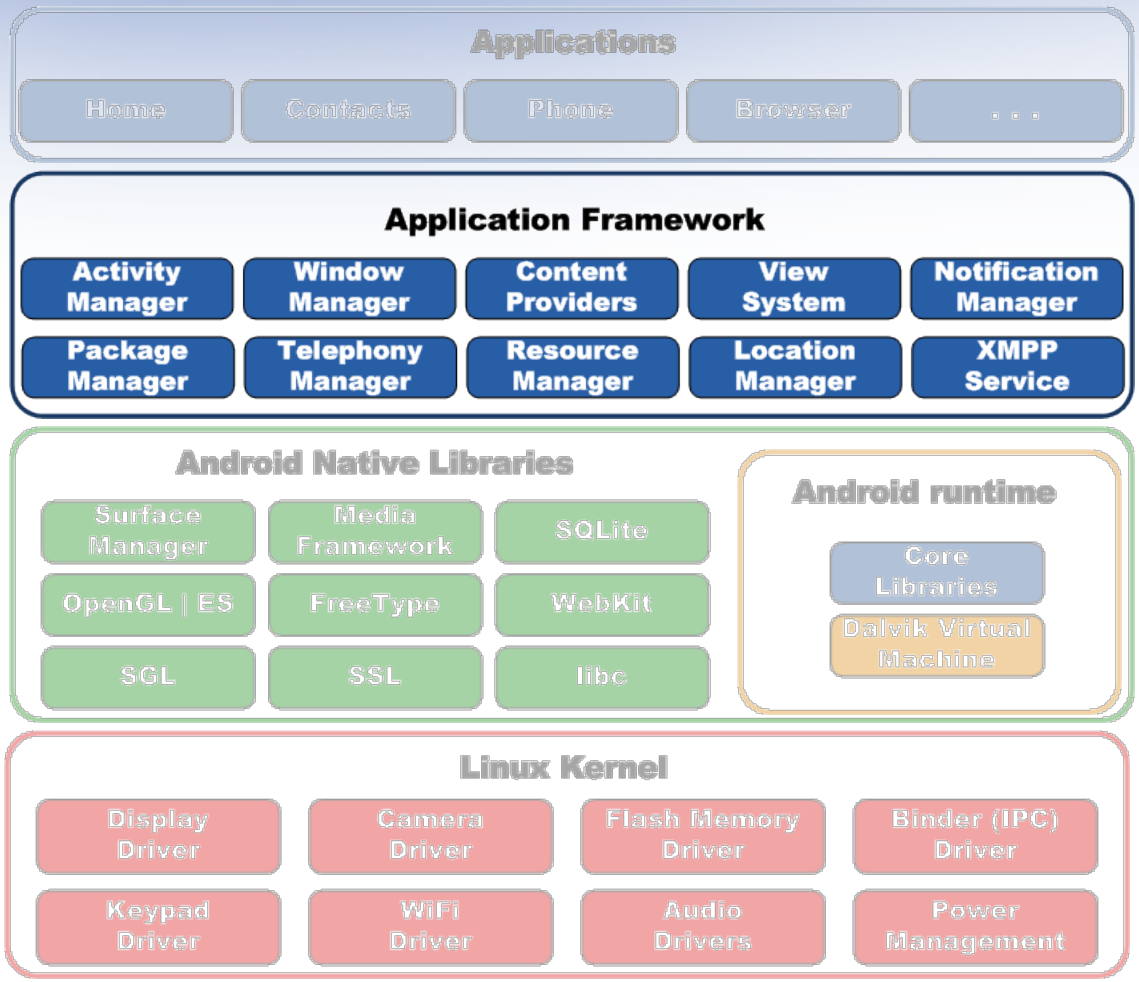
Core platform:

- Phone, Browser, Email…

Third-party:

- Apps written by third-party developers

Application Framework



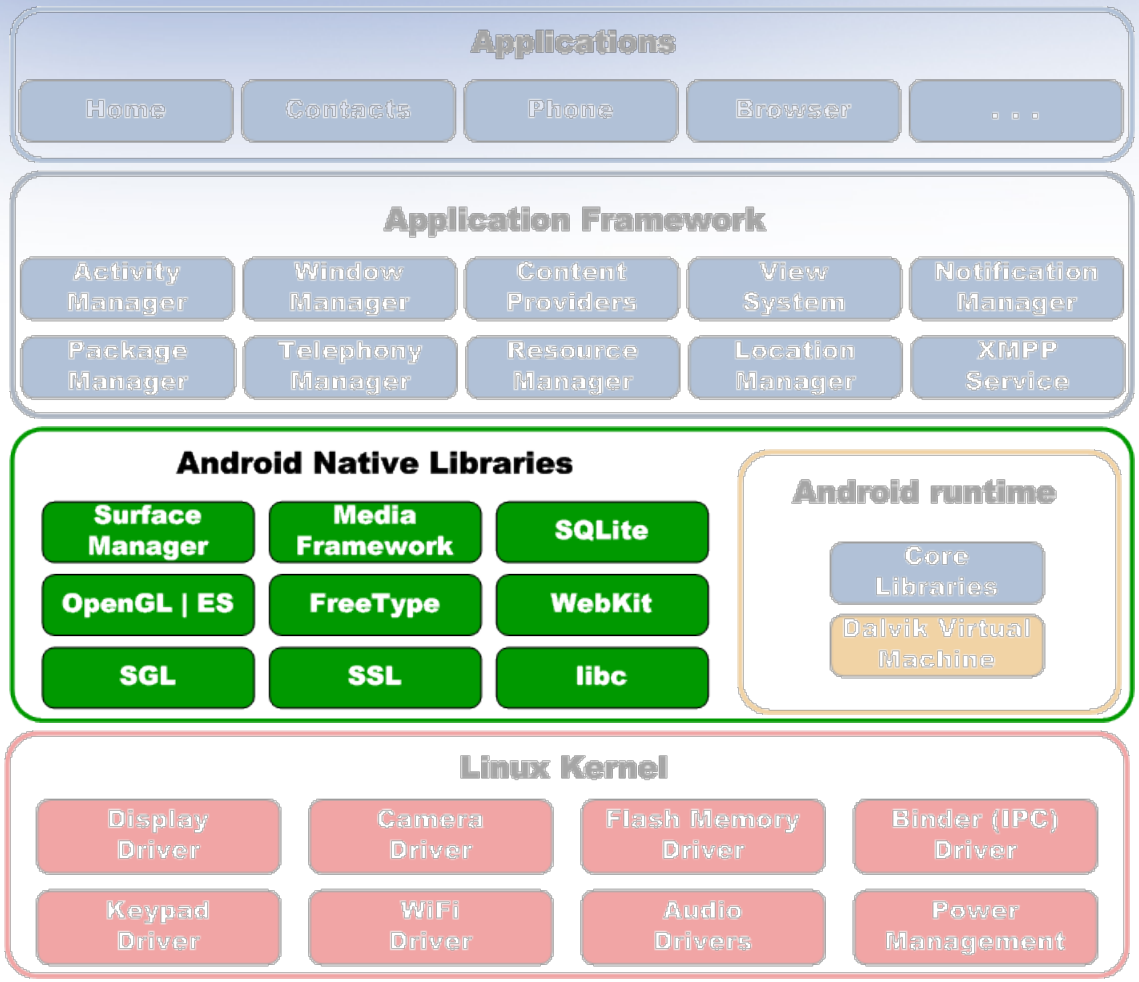
Core platform services:

- Activity, Package, Window and Content Providers

Hardware services:

- Telephony, Location, Bluetooth, WiFi, USB, and Sensor Services

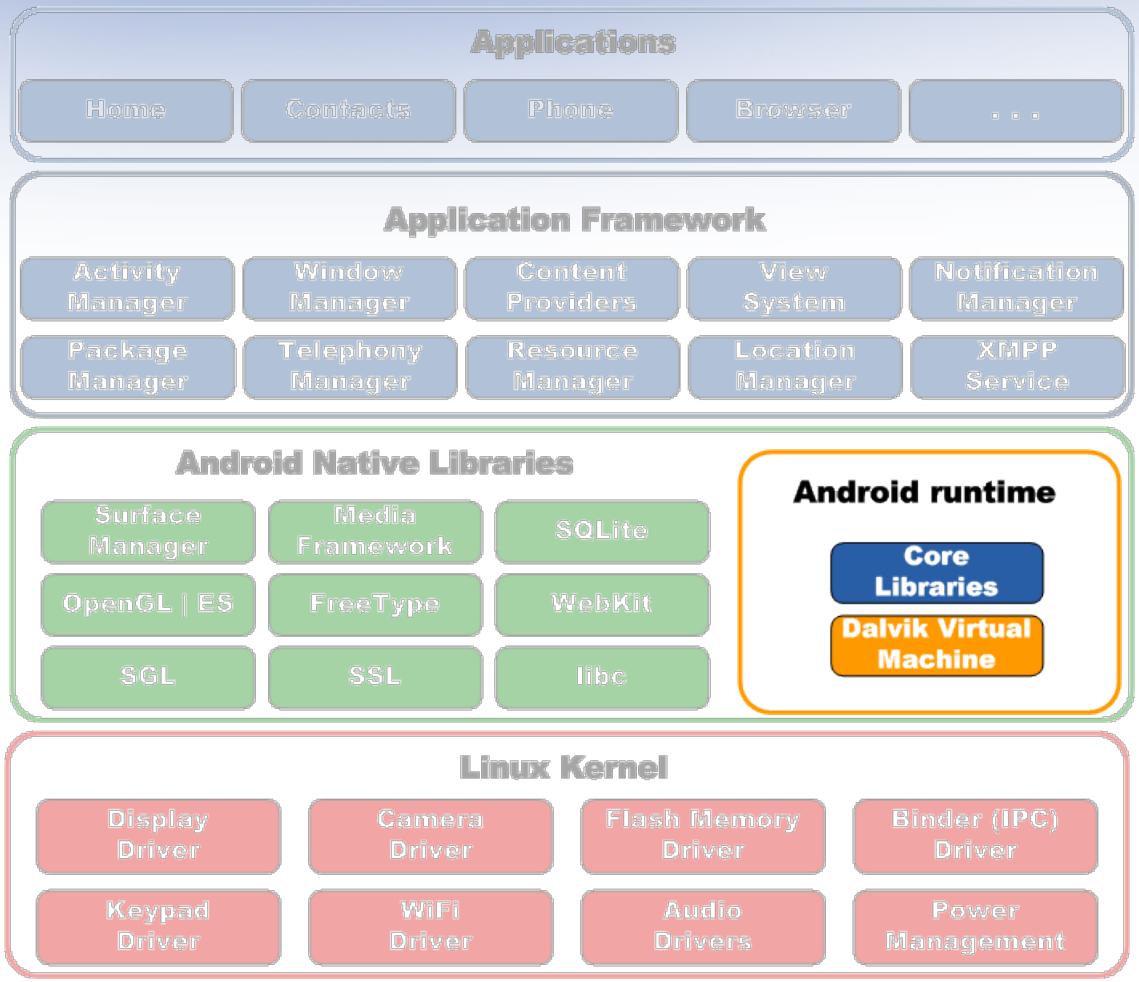
Android Native Libraries



Used for:

- Window management
- 2D and 3D graphics
- Media codecs
- Font rendering
- SSL
- The core of datastorage
- The core of web browser
- Bionic libc

Android Runtime



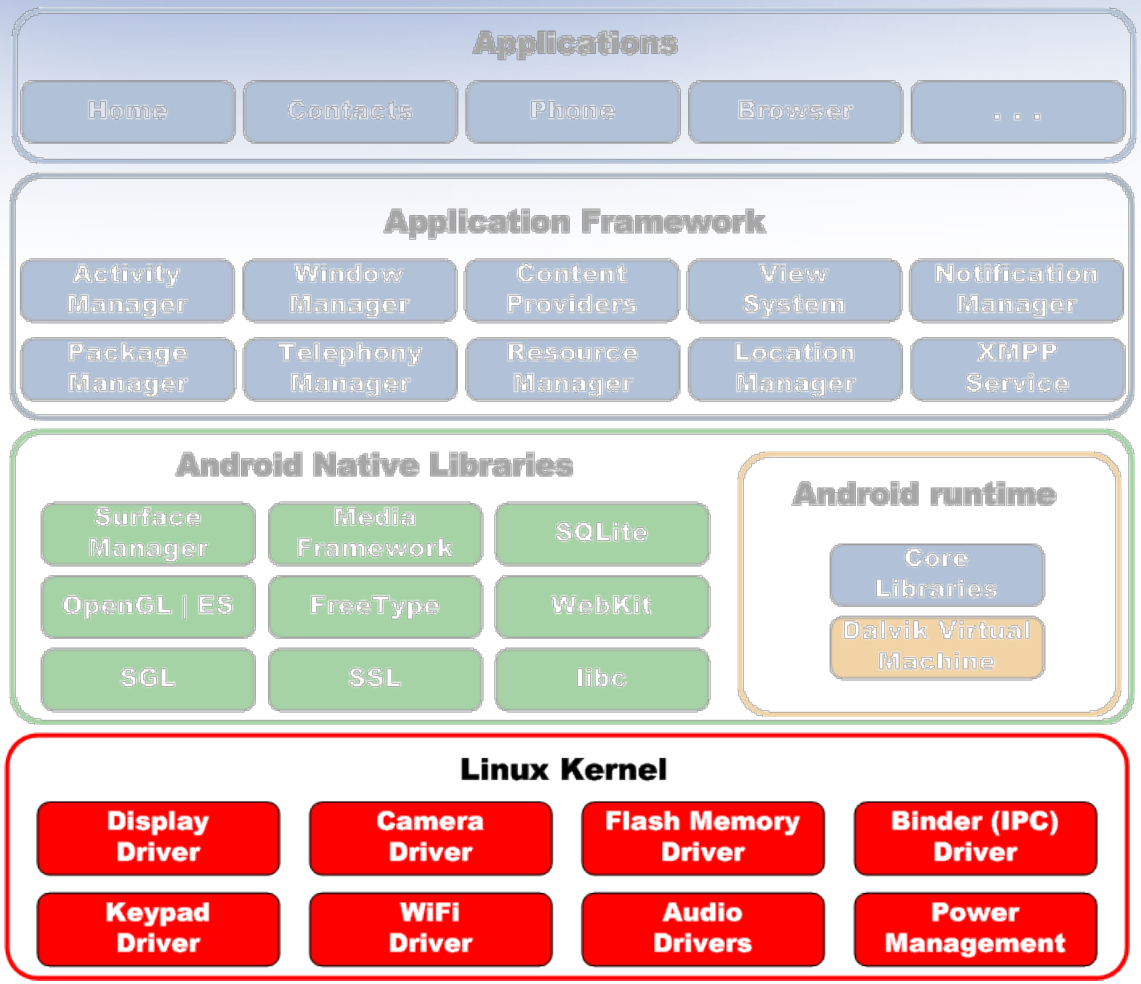
Core Libraries:

- Data structures, Utilities, File access, Network access, and Graphics

Dalvik VM:

- Provides application portability
- Supports multiple instances
- CPU and memory optimized to run on mobile devices

Linux Kernel



Linux features:

- Hardware abstraction layer
- Memory management
- Process management
- Security module
- Networking

Android enhancements:

- Power management
- Binder IPC
- Logger

Android Security Specification

- Android allows app developers to specify the security needs of their apps
- Each app comes with a Manifest file where the permissions listing the required permissions
- The user of the device has only two choices
  - Either install the app granting the whole set of permissions
  - Or not install the app
- **All-or-nothing model!**

Android Permission Levels

- Android provides a set of well-defined permissions
- *Normal Permissions* are assigned by default to apps
- *Dangerous Permissions* require user confirmation
- *Signature Permissions* are granted to apps signed by the same developer
- *System or Signature Permissions* are granted only to special apps installed in the data/system folder (i.e., apps signed by Google)

Permission example

- An app that wants to listen for incoming SMS has to declare in its manifest:

```
<uses-permission
android:name=android.permission.RECEIVE_SMS"/>
```

- The `RECEIVE_SMS` is consider a dangerous permission and the apps has to request it

Android Security Enforcement

- Android supports a security model that is enforced by two layers: Linux and Android middleware
- Linux enforces the DAC model
- Android middleware enforces a MAC model

Linux DAC in Android

- When an app is installed it gets a unique UID and GID
- Each app gets a home dir
  - `/data/data/<package_name>/`
- The UID and GID of the app get full access to its home dir and the files it contains
  - rwx,rwx,---

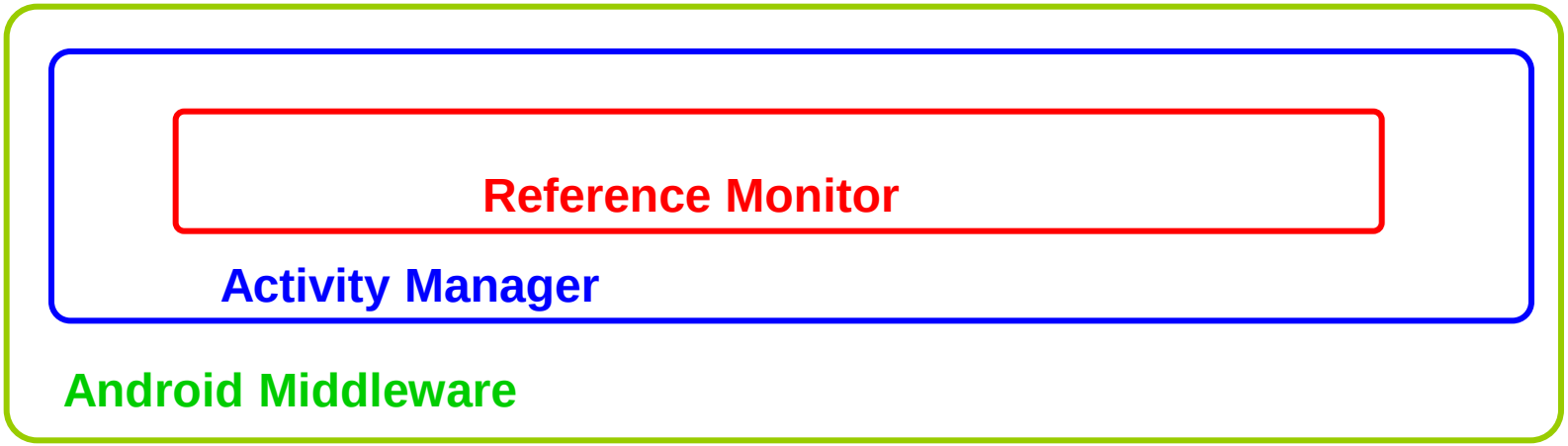
Linux Special Groups

- Linux also maintains special groups for the Internet, External Storage, and Bluetooth
- If an app asks for accessing Internet (and the user install it) it is assigned to the Internet Group

Android Middleware MAC

- The Android Middleware controls the way in which apps use the ICC mechanism
- Each protected feature that is reachable through the ICC mechanism is assigned a label
- When the app asks for a permission in its manifest the corresponding label is assigned to the app

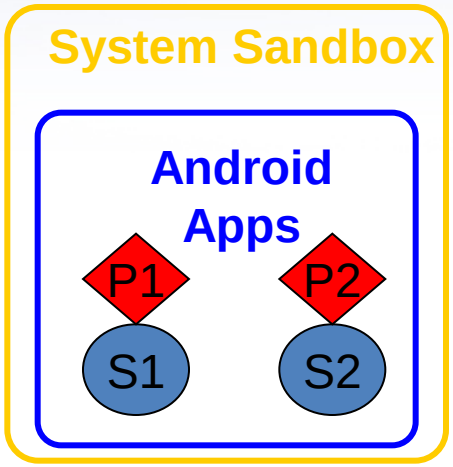
Android MAC Model

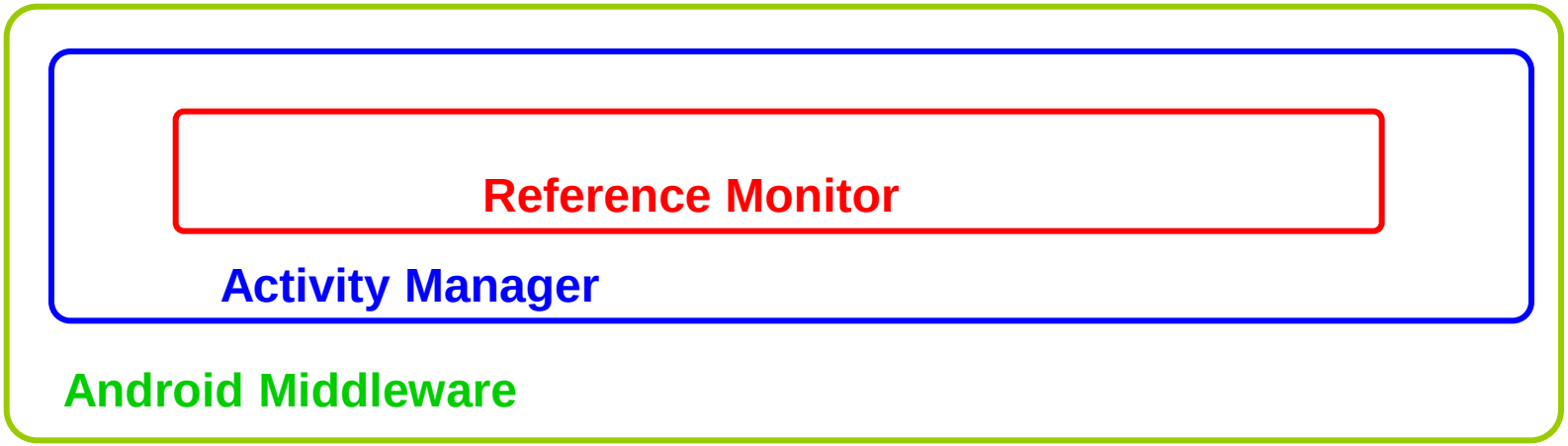


Protection Domain

S1 = Location Service

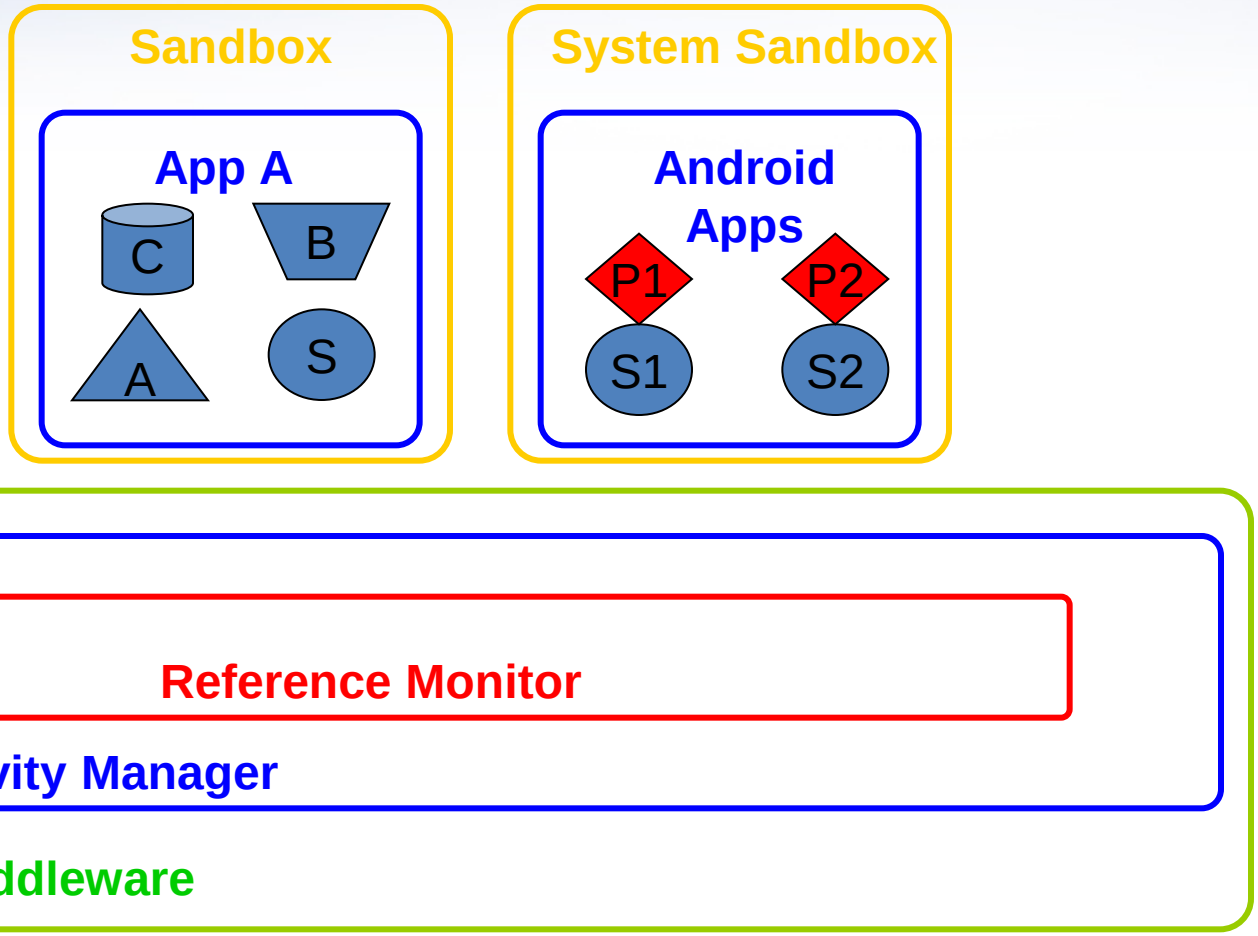
P1 = LOCATION_PERMISSION



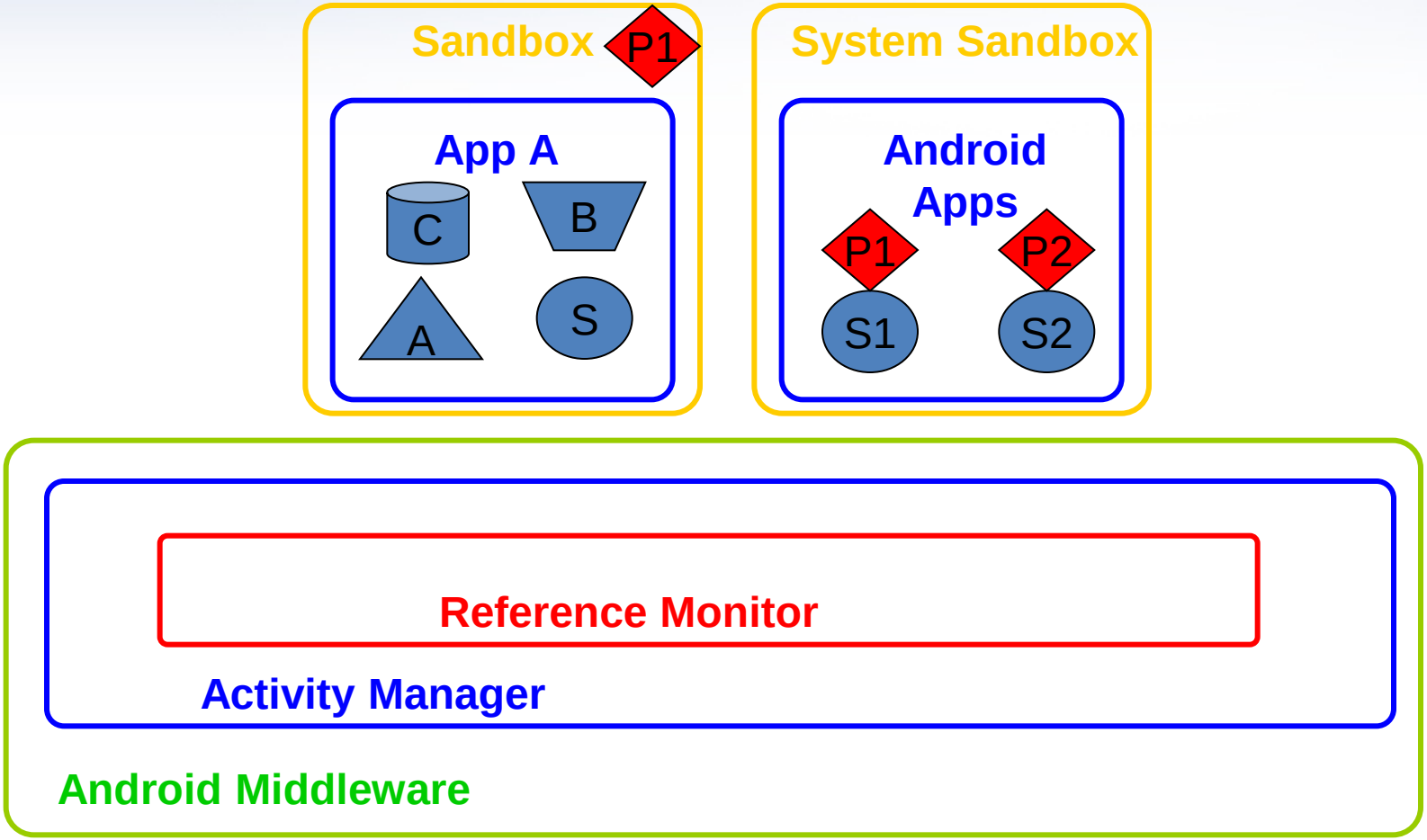


Assignment of Permissions

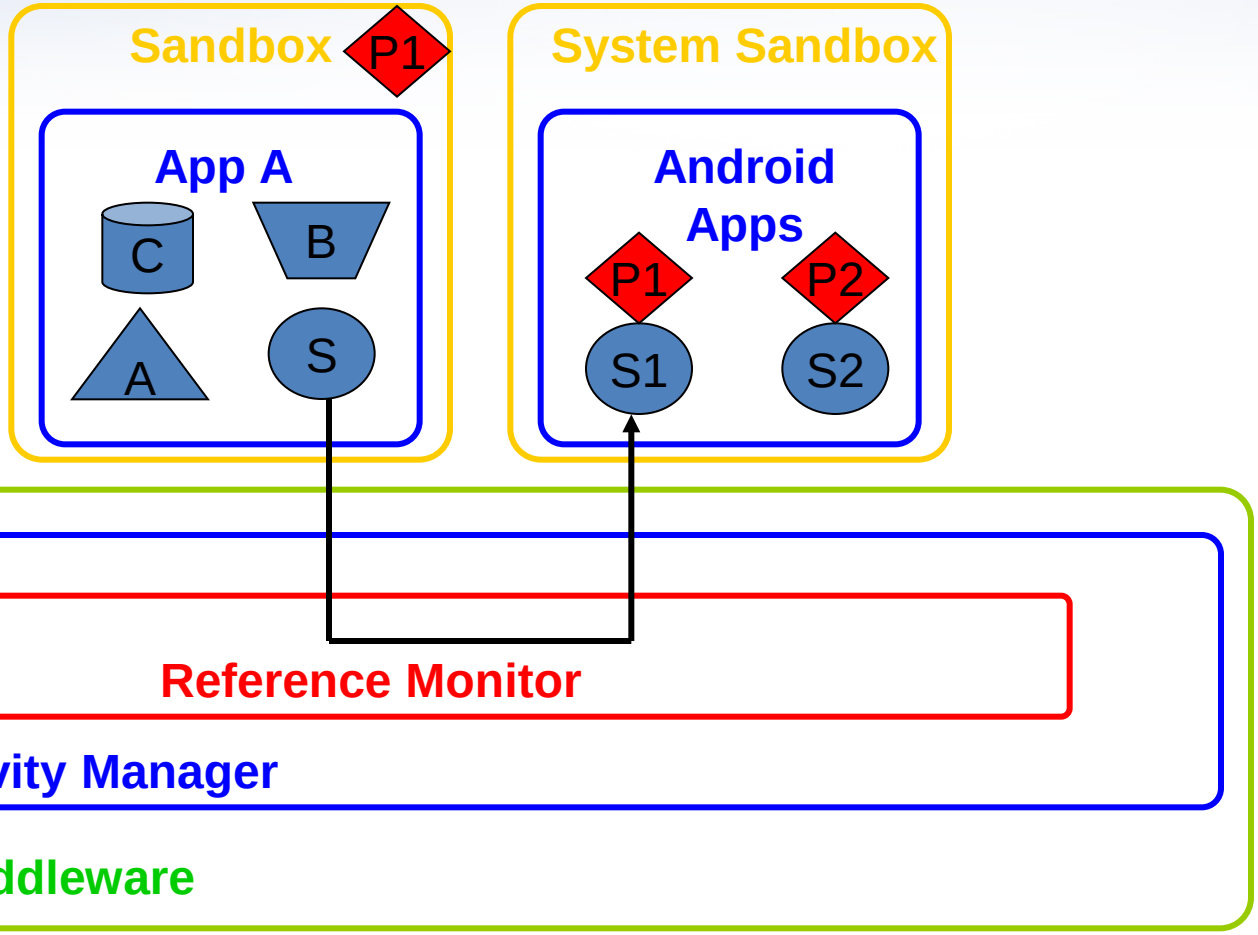
Install Time: Uses Permission = P1?



Using the Permission



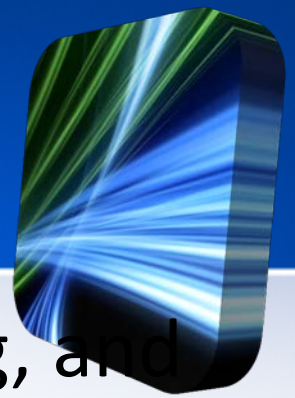
Reference Monitor



Security Confinement

- Once the labels are assigned neither the app nor the user can change them
- Apps cannot delegate their permissions
- However, components can expose interfaces to other apps
- This makes difficult in standard Android to control information flow (can lead to severe attacks)

Resources



- Read: [1] William Enck, Machigar Ongtang, and Patrick McDaniel. **Understanding Android Security**, *IEEE Security and Privacy Magazine*, 7(1):50--57, January/February, 2009.