

Android Forensics

(Part 2)



Contents

- Smart Phone Market Share
- Introduction to Android Operating System
- Android Architecture
- Android Security
- Summary



Smartphone Market Share



- There are various devices which run on Android, but Android is widely used on Smart Phones
- Based on Worldwide Smartphone Shipment OS Market Share Forecast
- <https://www.idc.com/promo/smartphone-market-share/os> [Updated: 15 Dec 2020]

Year	2018	2019	2020	2021	2022	2023	2024
Android	85.1%	86.1%	84.8%	85.0%	85.3%	85.6%	85.7%
iOS	14.9%	13.9%	15.2%	15.0%	14.7%	14.4%	14.3%
Others	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
TOTAL	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

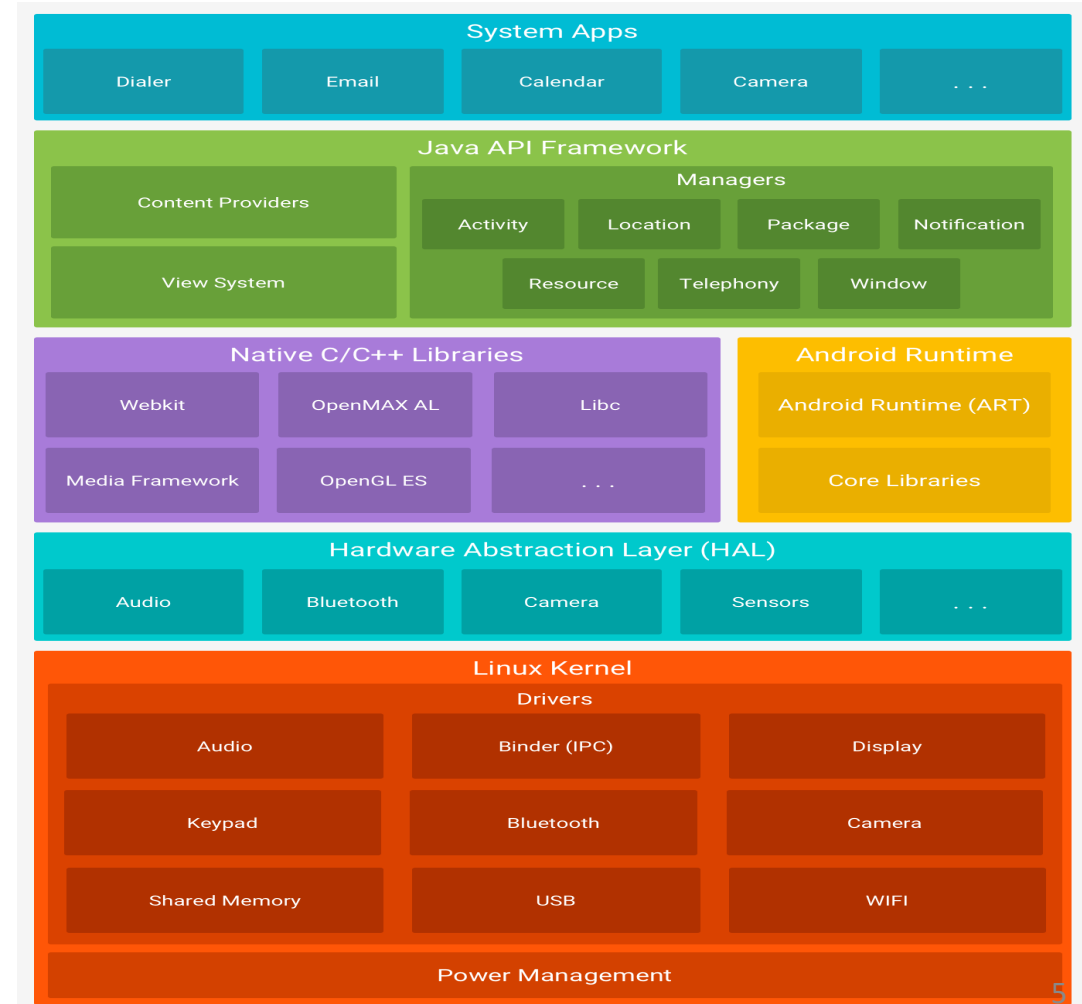
Introduction to Android Operating System

- Android is a mobile operating system based on a modified version of the Linux kernel
- Android is an operating system (OS) developed by the Open Handset Alliance (OHA)
- The OHA is the group that is in charge of the Android smartphone operating system
- It was created by Google and includes key members such as China Mobile, HTC, Intel, Motorola, NTT DoCoMo, Sprint, and T-Mobile USA
- The Alliance is a group of more than 50 mobile technology companies working together to develop a non-proprietary open standard platform called Android project



Android Architecture

- The basic architecture of Android, its core and Android OS builds are based on the Linux kernel



Android Architecture (Cont'd)

The Linux kernel

- All of the core functionalities of Android, such as process management, memory management, security, and networking are managed by the Linux kernel

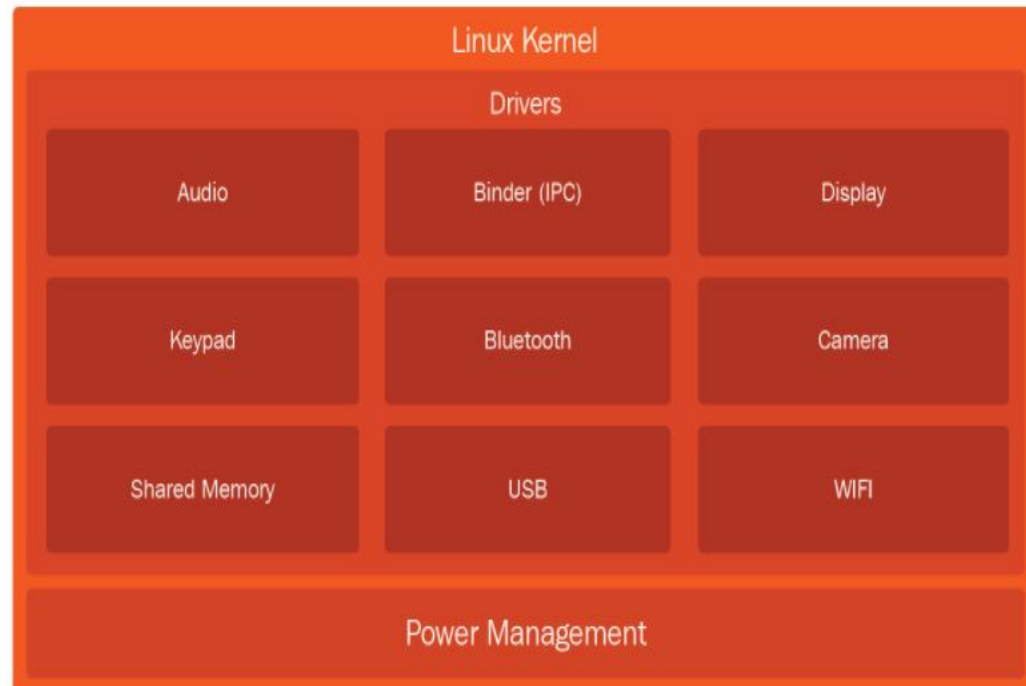
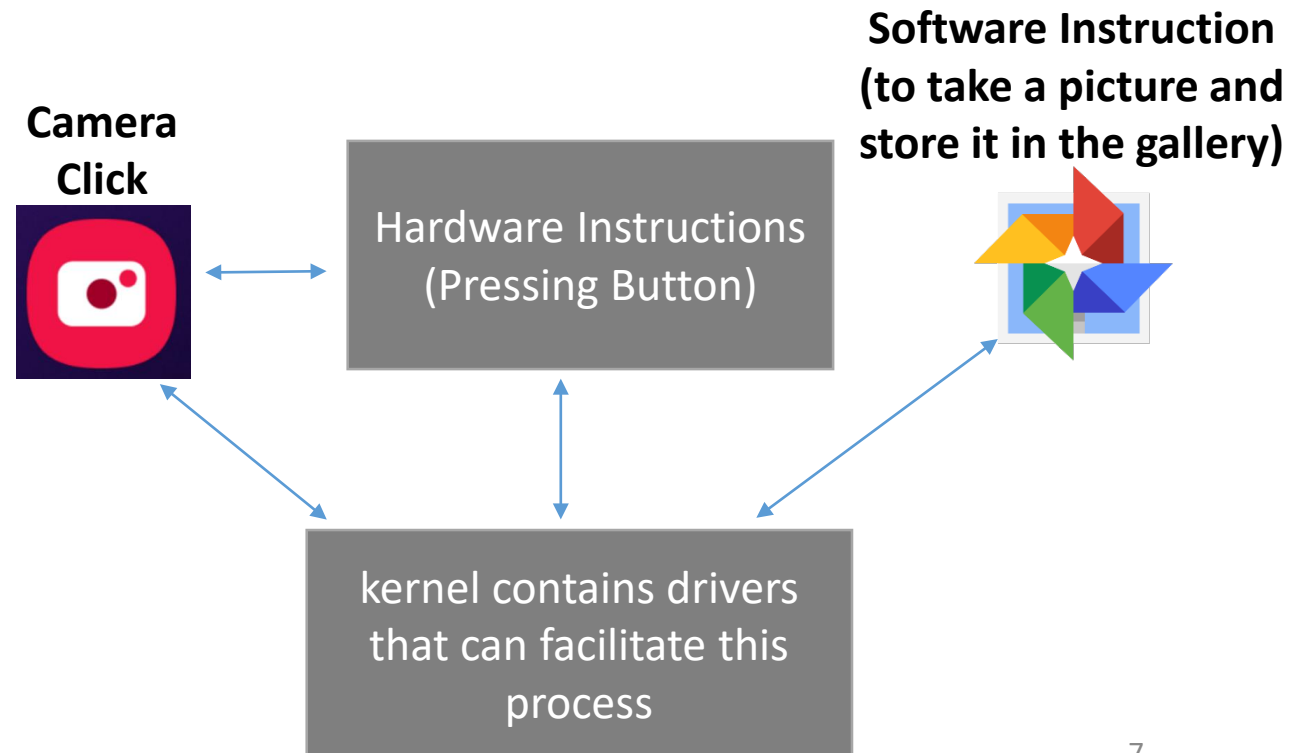


Android Architecture (Cont'd)



The Linux kernel

- Example : What actually happens when you take a photo using the camera button on your mobile device?



Android Architecture (Cont'd)

Hardware abstraction level

- The hardware abstraction level (HAL) allows the higher level, Java API framework, to work with mobile device's hardware with help of standard interfaces
- This can be done due to multiple library modules, which provide interfaces for different types of hardware components, like Bluetooth or camera



Android Architecture (Cont'd)

Android Runtime

- Since Android 5.0 each application runs in its own process and with its own instance of the Android Runtime (ART)
- It allows multiple virtual machines to run on low-memory devices by executing DEX (Dalvik Executable) files
- Prior to version 5.0 Dalvik was Android Runtime, so applications developed for Dalvik should work when running with ART

Native C/C++ Libraries

- Many core Android system components and services, including those mentioned earlier, like HAL and ART, are built from native code, so they require native libraries written in C and C++

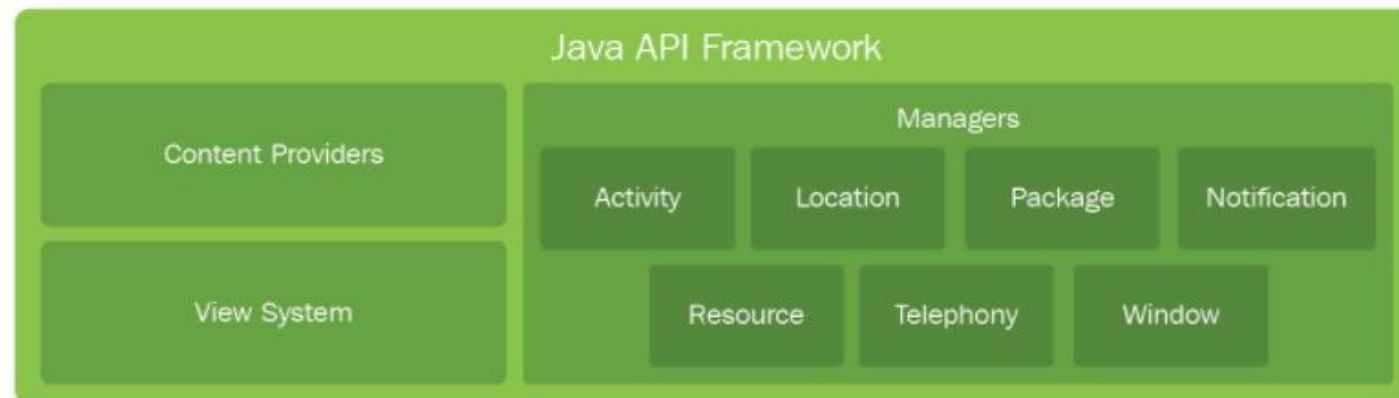


Android Architecture (Cont'd)



Java API Framework

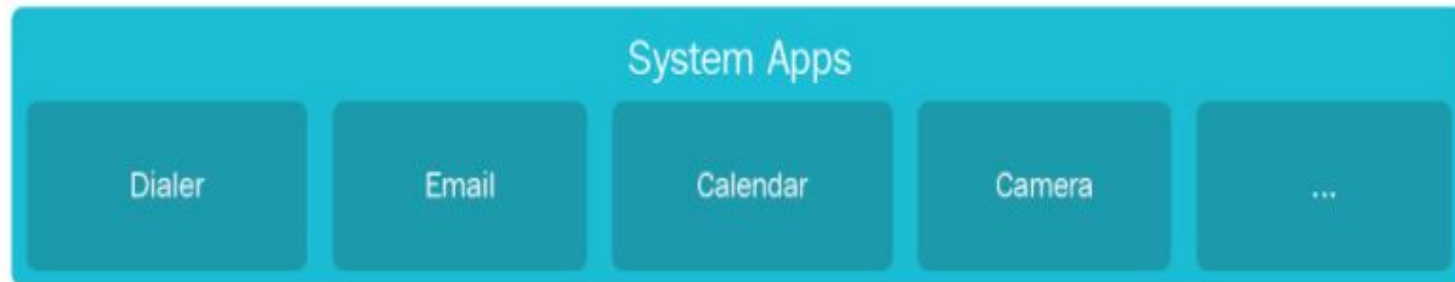
- Java API framework allows developers to create applications using modular system components and services as building blocks
 - **View System** allows to build application's user interface, and includes lists, grids, text boxes, buttons etc.
 - **Resource Manager** provides access to non-code components of an application, like localized strings, graphics and layout files
 - **Notification Manager** allows applications to display custom alerts
 - **Activity Manager** manages the lifecycle as applications, and their back stacks the order in which each activity is opened
 - **Content Providers** allows applications to access other applications data, and share their own



Android Architecture (Cont'd)

The Application Layer

- The topmost layer in the Android stack consists of applications called apps, which are programs that users directly interact with



Android Architecture (Cont'd)



The Application Layer

- There are two kinds of apps, System apps and User Installed apps
- System apps
 - These are applications that are pre-installed on the phone such as default browser, email client, and contacts are examples of system apps.
 - These generally cannot be uninstalled or changed by the user as they are read-only on production devices though some devices offer the ability to disable these applications
 - If a system application is disabled, the app and all of its data remain on the device on the system partition, the application icon is simply hidden from the user.
 - These applications can usually be found in the /system partition
 - Beginning in Android 4.4, apps installed in /system/priv-app/ are treated as privileged applications and are granted permissions with protection-level (signatureOrSystem) to privileged apps

Android Architecture (Cont'd)

- **User installed apps:**

- These are the applications that are downloaded and installed by the user from various distribution platforms such as Google Play
- Google Play is the official app store for the Android operating system, where users can browse and download the applications
- These apps are present under the /data partition



Android Security

- Android by default runs on Linux kernel
- Linux kernel provides Android security features
- The security features and offerings that are incorporated aim to achieve three (3) things:
 - To protect user data
 - To protect system resources
 - To make sure that one application cannot access the data of another application

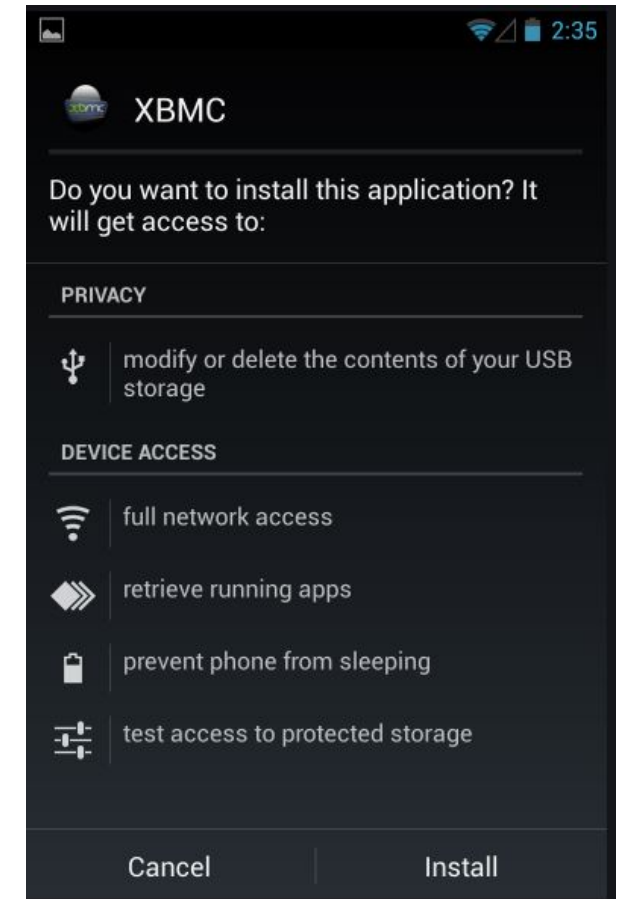




Android Security (Cont'd)

- **User-based permissions model**

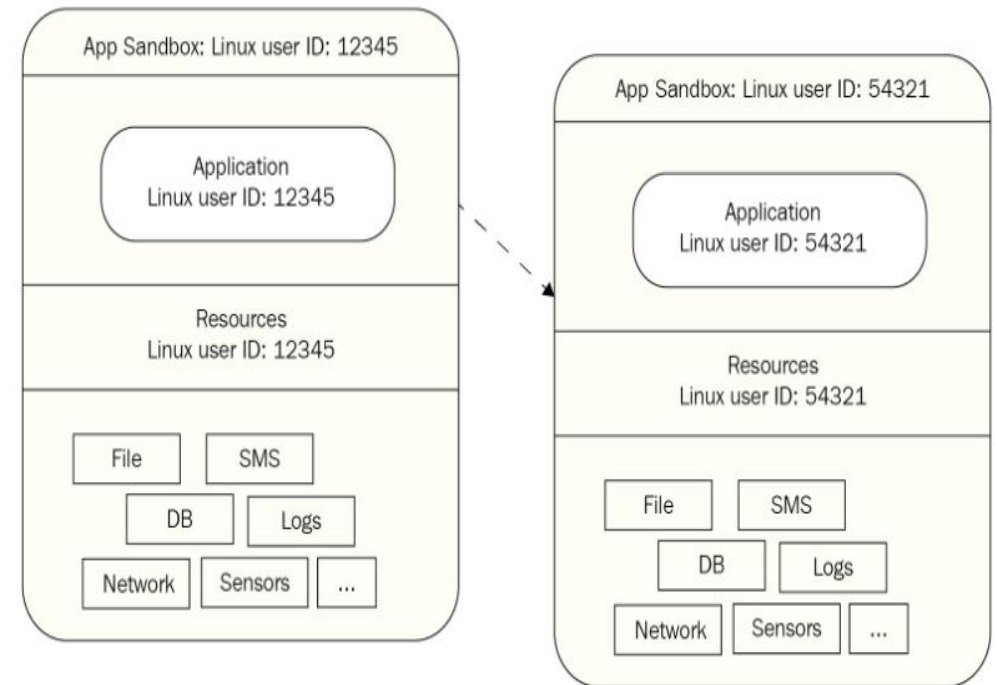
- Android OS has implemented permission model for individual apps
- Applications must declare a list of permissions they require in a file called **manifest.xml**
- When a user installs the application, Android presents permission list to the user so that they can view the list to allow or disallow the installation
- We cannot install an android application with few permissions. Either we have to accept all the permissions or decline to install it



Android Security (Cont'd)

- **Application sandboxing**

- By default, Android make use of Linux user-based protection model
- Every android application is assigned is a UID and is run as a different process
- By default, an application cannot access data of other application
- So, if an application tries to do something malicious, it can do within its context and permission it has been assigned by the OS



Android Security (Cont'd)

- **SELinux in Android**

- From Android 4.3, SELinux (Security Enhanced Linux) is supported by Android
- SE android uses MAC (Mandatory Access Control) which ensures Application runs in an isolated environment
- With SELinux Android, if a user installs the malicious app, malware cannot access the OS and corrupt the device

- **Application Signing**

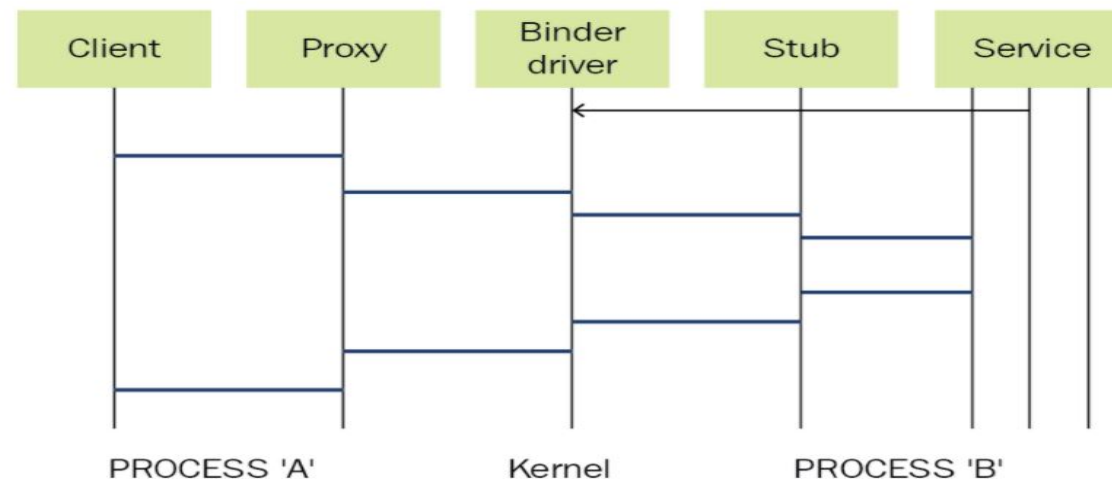
- All android apps are signed by the developer/owner to determine the author of the app
- The key store and private key which is used during signing need to be protected for pushing a new update to the application



Android Security (Cont'd)

- **Secure interposes communication**

- As discussed, sandboxing is obtained between the processes using UID assigned to different processes
- The Binder framework in Android provides the capabilities required to organize all types of communication between various processes
- All communications between the processes using the Binder framework occur through the Linux kernel driver, **/dev/binder**



Summary

- Understanding Android architecture and security model is crucial to having a proper understanding of Android forensics
- The inherent security features in Android OS such as application sandboxing and permission models safeguard the device from various threats and pose as an obstacle for forensic experts during an investigation

