ISSN (e): 2250 – 3005 || Volume, 08 || Issue,10|| October – 2018 ||
International Journal of Computational Engineering Research (IJCER)

Layered Mobile Device Architecture

[1]Shinto Kurian [2]Dr.K.Nirmala
K, Research Scholar(Reg.No:PhD/10/PTE/1/2017, Madras University),
Quaid-E-Millath College for Women, Chennai - 600 002, Tamilnadu,India.
Assoc. Professor,Dept. of Computer Science,
Quaid-E-Millath College for Women, Chennai - 600 002, Tamilnadu,India
Corresponding Author: Shinto Kurian

ABSTRACT

Mobile device structure is organised in a layered architecture from electronic components to application user interface. Based on various functionalities, the mobile devices are separated into multiple layers. Each layer has well defined boundaries and interacts with each other using certain protocols. The layered separation helps the devices to segregate the functionalities in stabilized and secured manner. Depending on manufacture, the components in each layer change. Most of the mobile devices are follow a standard architecture but the components and methodologies used in each layer have differences. The degree of smoothness between the layers directly proportionate with user friendliness of the mobile device.

KEYWORDS: Mobile Device, Operating system, Software, Hardware, BIOS, Firmware, User Interface.

Date of Submission: 15-12-2018 Date of acceptance: 31-12-2018

I. INTRODUCTION

Mobile devices are manufactured using multiple electronic or hardware components. This electronic components are interlinked through BIOS systems which is stored in a microchips or flash memory. Operating system is installing in physical storage and it is controlling the hardware components through BIOS systems. Many of the manufactures are developing their own themes called firmware and installing on top of operating system or integrate with operating system to cater their own identity. Various application used in mobile devices are installed in top of operating system. Each application has its own interfaces developed using various application architectures and it is interacting with device users.Mobile device can be categorize into multiple layers. Each layer takes care of different activities and interacts with each other. In high level the mobile device has six layers (Figure-1).

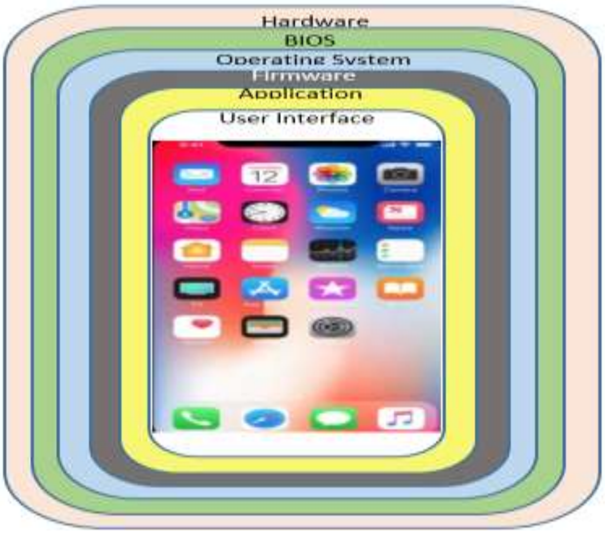


Figure-1

1. Hardware Layer
2. BIOS Layer
3. Operating system Layer
4. Firmware Layer
5. Application Layer
6. User Interface Layer

1 Hardware Layer: Base layer of a mobile device is hardware components. Main components are circuit board, processor, display, battery, SIM card slot, memory unit, USB/charging slot, Bluetooth, Wi-Fi, Speaker, Microphone, etc. This device supports multiple data connectivity methodologies. Many of the latest devices support GPS, media player, High resolution camera and various sensors. Few of the devices give biometric (Eg: Finger print sensor, eye retina sensor) related security options. Below mobile PCB diagram (Figure-2) shows high level hardware structure of the mobile device

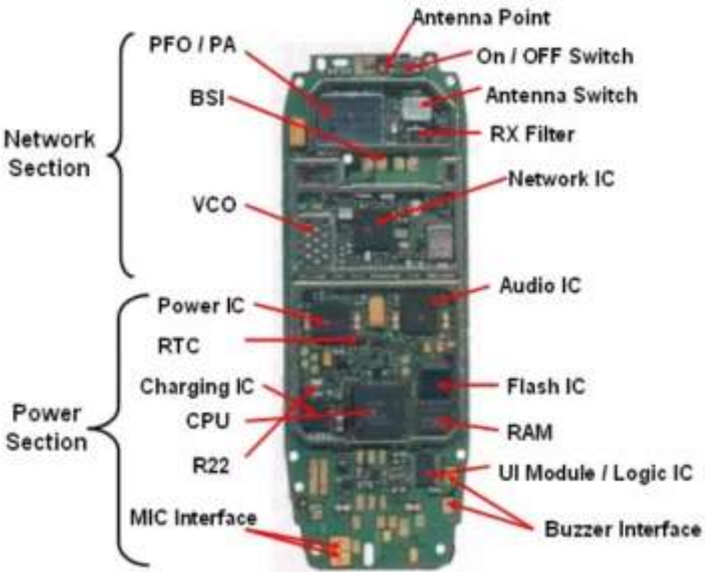
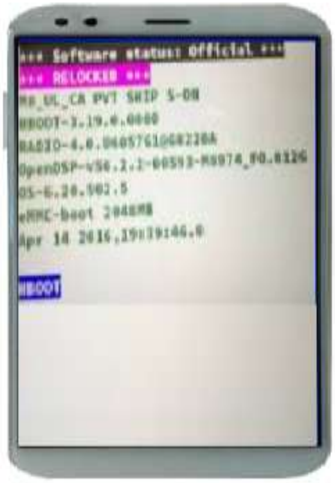


Figure-2

2 BIOS Layer: Basic Input Output System (BIOS) is a secured set of special code to interact closely with hardware components, it also called bootloader. Normally it resides on internal ROM.
BIOS code typically builds in low level language. Normally this code will w by the device manufacturer. It will setup the hardware components configuration and help to run the ROM level code before loading the operating system. When the device is switched on, the BIOS gets triggered and it activates the hardware components. We can see the bootloader messages in screen (Figure-3). BIOS trigger the Kernel of operating system through interrupts. Once operating system is up, BIOS is not mandatory.



```
*** Software status: Official ***
*** RELOCKED ***
M8_UL_CA PVT SHIP S-ON
HBOOT-3.19.0.0000
RADIO-4.0.0605761G68220A
OpenDSP-v56.2.2-00593-M8974_F0.0126
OS-6.20.502.5
eMMC-boot 2048MB
Apr 14 2016,19:19:46.0

HBOOT
```

Figure-3

3 Operating System Layer: This layer contains the Operating system kernel, system level component services and its configuration. When BIOS layer sends interrupts to mobile OS kernel, it loads the operating system in the device. Then operating system enables all the hardware components and make each one functional. By default the operating system provides multiples applications for various purposes. Normally the operating system upgrades are controlled by the Operating system owners or device manufactures. They provide options or references to upgrade the operating system. Operating system will provide the interface to access the hardware components. It also manages many of system level services. Different operating systems mange the hardware components in different ways. Android and iOS operating systems are used on more than 95% mobile devices. Figure-4 gives a high level overview of mobile OS market usage percentage.

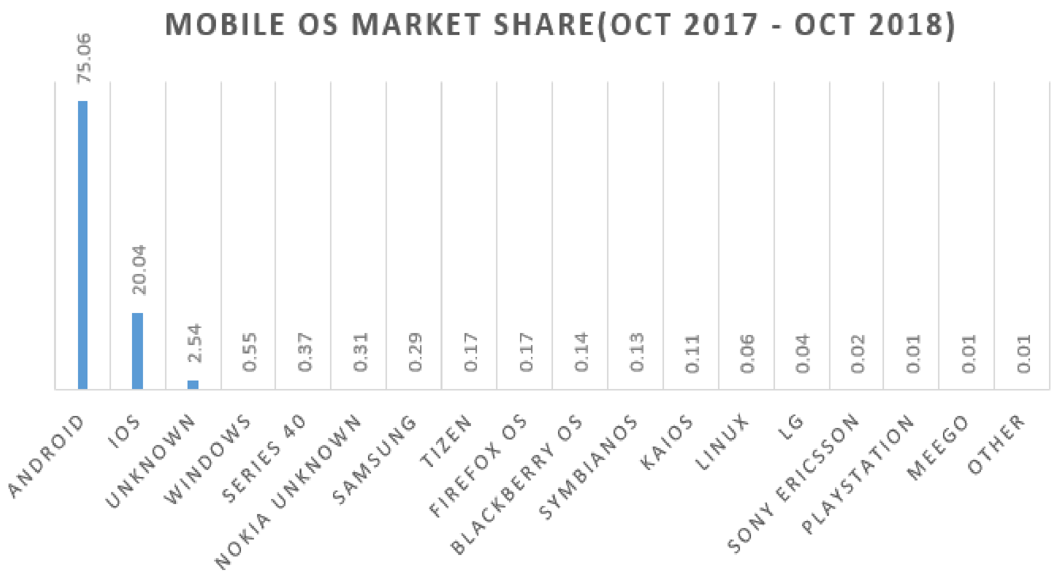


Figure-4

4 Firmware layer: Firmware is a customized operating system. Many of the manufactures are customising an existing operating system and using as their own operating system. The customization mainly targets to bring some specific features as well as to provide separate look and feel. In Some scenarios the firmware is working as an operating system, few other cases the firmware placed on top of exiting operating system. The base operating system OS upgrades normally cannot apply in this customised version. The firmware owners has to release separate upgrades. Android OS has many firmware, below (Table-1) are few popular firmware.

Firmware	Owner /Manufacturer
MIUI	Xiaomi Tech
YunOS	Alibaba
CarbonROM	Team Carbon
CyanogenMod	Cyanogen Inc
IndusOS	Team Indus OS
LeWaOS	Lewa Technology
LineageOS	LineageOS community
OxygenOS	OnePlus creators
SmartisanOS	Smartisan Technology

Table-1

5 Application Layer: Mobile applications are running on top of operating system. Each application development process might change based on operating system but its working structure is almost same. Applications are triggering the operating system level commands to execute the designed functionality. Presently many of application development tools and interfaces are available to develop applications specific to each operating system. Also some tools are available to develop the applications in some common manner and can convert that to different operating system specific. Normally applications manage many functionalities like receives the input data, process data based on requested functionality, store the information, implement the security features, trigger the operating system level commands, etc. Below diagram (Figure-5) give an overview of internal structure of an application

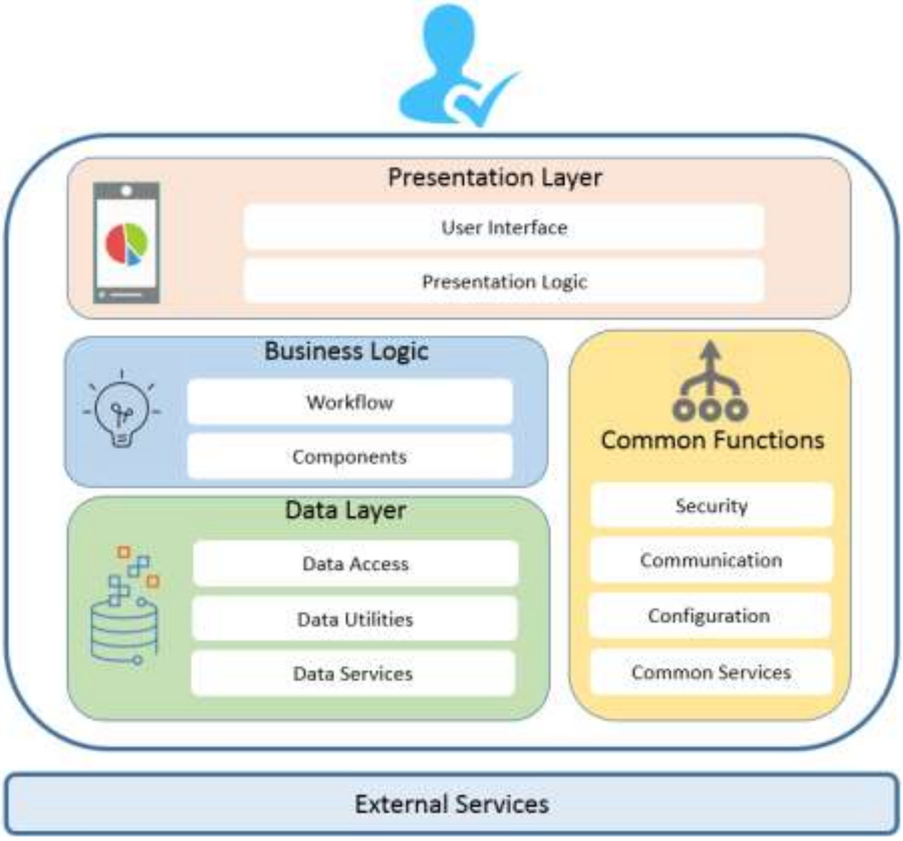


Figure-5

Majorly three type of mobile applications (Figure-6) are available

Figure-6

5.1 Native Apps: Applications are developed specific to each operating system. Example: iOS apps, Android apps. The application developed for Android OS can't install on iOS mobile device. This type of application give very well in performance than any other type of apps. Normally this kind of applications are installed from respective operating system's application store.

5.2 Hybrid Apps: This type of applications are developed using particular app development tools. Phonegap, Xamarin, Cordova, Ironic are some familiar examples of this tools. This type of application provides key features of Native apps as well as Hybrid apps. This type of apps work like native app and it have the many capabilities of web apps. Main advantage of this type of applications are less cost and development time.

5.3 Web Apps: This is a client server type application and running through any browser. Mostly this type of applications are developed in HTML5, CSS3 and JavaScript coding standards and business functionalities are running server based services. Key advantage of this type of applications is platform independency

6 User Interface Layer: This is most attractive and sophisticated part of an application. Using this interfaces, mobile device users are accessing and operating the applications. In a normal mobile device usage, consumers are interacting with this layer to provide inputs to the mobile applications. The operating system owners or application owners are design the user interfaces in particular fashion. When the users accessing any of the application the particular interface is working in similar fashion. Below are two types of user interfaces

6.1 Platform User Interface: This type of UI are designed and developed by operating system owners. When use this interface in any application its look & feel and working structure would be same. Example: Date and time selection in iOS applications, the rolling date and time feature will applicable in all the applications.

6.2 Custom User Interface: This type of UIs are developed as part of particular applications or common UI component development. Many open source UI components are available in market to use for application development, specifically for Android OS. Example: GBoard (google key board)

II. CONCLUSION

Many of mobile devices manufactures follow the layered architecture. Few of manufactures are using tightly coupled layered architecture while designing their devices. Which helps to enable high level security and integrity. This type of devices cross verifies the components between the layers. For instance operating system verifies the connected hardware is genuine before start the communication. Few other manufactures design their devices in loosely coupled layered architecture. In these type of devices, each layer operates partially independent to certain extend. In Such scenario few of the layers controlled by hardware manufacture and others controlled by software (Operating system, Application, UI Component) manufacture. The devices require high level security and integrity uses tightly coupled layered architecture other devices uses loosely coupled layered architecture.

REFERENCES

[1]. ‘Wikipedia’ https://en.wikipedia.org/wiki/Mobile_operating_system

[2]. ‘Parts of a Mobile’ http://www.mobilecellphonerepairing.com/parts-of-a-mobile-cell-phone-and-their-function-big-parts.html

[3]. ‘Mobile OS Statistics’ http://gs.statcounter.com/os-market-share/mobile/worldwide/

[4]. ‘Mobile App Architecture’ https://www.peerbits.com/blog/all-about-app-architecture-for-efficient-mobile-app-development.html

Shinto Kurian"Layered Mobile Device Architecture "International Journal of Computational Engineering Research (IJCER), vol. 08, no. 10, 2018, pp 38-42