



## Design and optimization of Camera HAL layer based on Android

---

Yu Gan, Wei Hu and Yonghao Wang

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

September 6, 2019

# Design and optimization of Camera HAL layer based on Android

Yu Gan, Wei Hu, Yonghao Wang

<sup>1</sup> College of Computer Science and Technology, Wuhan University of Science and Technology, China

<sup>2</sup> Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan, China

<sup>3</sup> Digital Media Lab, Birmingham City University, UK

Rorchach2k@gmail.com

huwei@wust.edu.cn

yonghao.wang@bcu.ac.uk

**Abstract.** Nowadays, in the field of mobile and communication, with the rapid popularization and application of Internet and network, and the continuous expansion to the family field, the trend of integration of consumer electronics, computer and communication (3C) is becoming increasingly obvious, and embedded system naturally becomes a research hotspot. The embedded operating system Android has been widely used in mobile devices, such as smart phones, smart watches and so on. Our design is based on the Android platform to optimize and improve Camera HAL. This design is applied to the development of the Android system, the design of the Camera subsystem, the description of the Camera subsystem from the structure, function and data flow, and then proceed to structural optimization. This design uses TI AM335x as the hardware platform, based on Android system, designed and implemented the Android Camera hardware abstraction layer and Camera subsystem. At the end of the development, after the actual experimental test, the development of the system achieved the expected function and worked well.

**Keywords:** Android, Camera, HAL, Embedded System, System Architecture.

## 1 Introduction

The technology development in today's era is very rapid, especially the smart phone technology is rapidly emerging. Operating systems, including IOS and Android. They are attracting more and more attention from the market, and manufacturers around the world are beginning to enter the market. Due to the excellent interface and convenient environment of the Linux kernel-based operating system, more and more new players are investing in Android development in recent years. So in this article, we're gonna try to propose a method to optimize and improve Camera HAL based on the Android platform.

This paper first introduces the environment and background of this system design, explains the structure of the paper and introduces the key technologies used<sup>[1]</sup>. The article first introduces the Android system and some of the key technologies that need to be applied, and then introduces the overall design of the Camera subsystem, mainly for the improvement of the HAL architecture. Then, the specific process of Camera HAL development is described from the aspects of client, server and HAL layer. Finally, the entire process of design is summarized.

Android is the platform for this development. The core of its system is Linux, which is based on the mobile operating system of various services and UI. As former Android has a wide range of applications, Google has set up OHA (Open Handheld Alliance) to provide continuous updates. Its infrastructure is the Software Stack architecture, with the Linux kernel-based kernel layer as the lowest layer and the development language C<sup>[2]</sup>. The bottom layer provides a stable system interface to the upper layer. The middle layer is a system runtime layer, which uses C++-based function libraries and virtual machines. Further up is the Application layer. The basis for developing Android applications is the application framework layer.

HAL is proposed by the Android platform to protect the intellectual property of the manufacturer. It has the feature of bypassing the GPL development source of Linux, without having to publish all the code<sup>[3]</sup>. We can implement hardware control methods in the Android HAL, and only use the Linux driver to complete a simple data exchange, or directly map the hardware register space to the User Space. Android is based on the Apache license, which means that the hardware vendor only needs to provide binary code or library files<sup>[4]</sup>. So we said that Android is extremely open as an operating system, but it is not open source. But Android does not comply with the GPL, and the contradiction between GPL and IC design vendors still exists.

Android layers use JNI and HAL middleware technologies<sup>[5]</sup>. As mentioned before, the system is highly open and provides an unlicensed fee service. Many advantages make Android the world's most used mobile operating system. Google has cooperated extensively with many parties, established a standardized and shared mobile communication platform, and built an open ecosystem chain to provide better services to users around the world.

## 2 Related Work

### 2.1 Android system structure

Android divides the system architecture into four levels. The following describes the four layers in order from top to bottom.

Application layer (Application). The application layer contains a collection of applications<sup>[6]</sup>. The emails, calls, QQ, WeChat, maps, alarm clocks, etc. That we usually use are all listed, and the programs designed and written by the third-party developers downloaded from the app store using the Java language also run at this level<sup>[7]</sup>.

Application Framework layer (Application Framework). This layer is the basic framework of Android application development, which contains a lot of interfaces and

class objects, which are often applied in the upper layer application development. Designers can use the API framework provided by Google to provide a lot of convenience in designing the program architecture and logical structure, reducing a lot of tedious and repetitive projects.

System runtime library (Libraries). In order to implement the positioning function of Android, many components will need to be implemented by some C/C++ libraries, which are included in the system runtime layer.

Linux kernel layer (Linux Kernel). Android is developed on the basis of the Linux kernel and has many software stacks<sup>[8]</sup>. The Linux kernel is between the hardware device and the software stack, and is still the core of the system itself.

## 2.2 Camera subsystem design

Camera as a subsystem of Android of course also conforms to such a framework structure. So to implement the new architecture, we only need to modify the framework layer and HAL layer on the original basis. The application layer implements page layout in XML language, mainly completing interface and interaction design. The main code is stored in the Camera.java file, which is eventually compiled into Camera.apk. We implement most of the functions and interfaces here. When the application layer accesses the local library, it is often necessary to use the java virtual machine to run java in the virtual machine. The framework/base/core/jni/ path is the corresponding code in this section. The key code provides a list of parameters for the application layer in the form of metadata for the data dictionary. The data will eventually be passed to the server. Camera belongs to the Android multimedia framework<sup>[9]</sup>. From the multimedia framework, the libui.so library is the interface library of the Android Binder mechanism in the Android multimedia framework, and the Camera as a subsystem also belongs to the multimedia framework. We use the libcameraservice.so shared library to implement the main features of Camera. But in addition to Camera, the video subsystem framework also calls libui.so, so Camera is not the only object.

## 3 Proposed Model

### 3.1 Android development kit

This design needs to be applied to some Android SDKs, which are available for free directly from the Android website. The SDKs of different versions of the device will be different. The platform has integrated the latest SDK into the tool. We can select and download it ourselves, or we can synchronize the SDK to an earlier version<sup>[10]</sup>. The SDK provides tools for developing on a variety of platforms including Windows, Linux, and Mac, and supports all platforms. Android originally supports C and C++, and the implementation of the SDK is encoded in Java, so to use the Android SDK you must use java to design. Google has announced that the Android virtual machine supports JNT technology, so all applications can call their own dynamic libraries through JNI. On the Android platform, it is possible to implement mixed

programming of C language, C++ and Java. But at the same time<sup>[11]</sup> Google has shown that the technology is still flawed, sometimes the development process is very complicated, or sometimes the code is very complicated. So they also provide the NDK tool (Native Development Kit), the local language development kit. This tool user uses C and C+ to implement certain features.

### **3.2 Android Binder mechanism**

Since Android is a completely open platform with a wide range of applications, security issues are a top priority<sup>[12]</sup>. For users, of course, all users do not want Android-based applications to steal their own private data, and the conventional process communication is not safe, and it is very easy to be hijacked and utilized by various Trojans. Trojans can easily push the program's recipient address to get a connection, which is almost impossible to block<sup>[13]</sup>. Binder is a new mechanism that emerged in this context. Since the communication structure adopts the C/S structure, the transmission process only needs one data copy, and the data sender can identify the ID of the user and the process, so the transmission efficiency and the security performance are all improved. The Binder mechanism uses the client/server communication mode. One of the processes provides some system services as a server<sup>[14]</sup>. Other processes act as clients, and send service requests to the server, and are uniformly scheduled by the service management unit to provide related system services. The Binder framework sets up four objects for Server, Client, ServiceManager and Binder. The server, client, and server management are all running on the User Space, and the Binder driver is running on the Kernel Space<sup>[15]</sup>. The relationship between the four objects can be compared to the Internet. The Server is the server, the Client is the client terminal, the ServiceManager is like the DNS, and the driver is the router.

### **3.3 Summary**

In this section, we focus on the key technologies used in this development process<sup>[16]</sup>. The development is based on the Android platform and applied to some official free development kits. The most critical part is the Android hardware abstraction layer and Binder-based communication. The C/S communication mechanism architecture built by way. It is not complicated by itself. The key to implementing this architecture is to distinguish the communication structure layer and the business logic layer of the application.

## **4 Experiments**

### **4.1 Install JDK**

JDK is a software development kit developed by Sun Microsystems for Java developers. In April 2014, Oracle acquired Sun, so the JDK is now officially available for download by Oracle. The Java Development Kit is the most frequently used toolkit

for developers from Java<sup>[17]</sup>.The toolkit includes basic components such as javac (compiler),java (interpreter),jar (packaging tool),javadoc (document generator),jdb (debug tool) and jre (Java runtime environment).Developers can use many convenient development interface interfaces through the toolkit.After the 8.0 version update,the operation efficiency is greatly improved.

The system version before 5.0 is compiled with JDK,and then compiled with open JDK7 or higher.The installation of the system version may be different.We use the 6.0 version as an example.Advanced Packaging Tools is a powerful package management mechanism,generally referred to as APT.We use APT for installation,which has the advantage of being simple and convenient and does not require manual configuration of the path.The tool will configure the environment variables at the time of installation<sup>[18]</sup>.The APT mechanism reduces the threshold for user operations, whether it is an installation upgrade or an uninstall, which greatly reduces the amount of user operations.

We first need to get the Android source code.First create a repository with repo init,initialize the android source code repository to download the latest code.

#### 4.2 Compile environment build and compile

(1) Environmental requirements:

Operating system: Officially recommended Linux Ubuntu10.04 (LTS, long term support version) 64bit;does not support compiling under Windows and Mac OS.

Memory and hard disk space size: 4G memory (the reason is to avoid "JVM run out of memory",while the larger the memory can speed up the compilation).

Disk space size: at least 25G or more storage space.

(2) Dependency package installation:

Refer to the official Android documentation,the following packages need to be installed to compile Android code under 64-bit ubuntu system.

Finally,the compilation section.We need to select a platform in the top level directory of the Android source code,and then select the required platform to compile with make.After the compilation is completed,the required image will be generated in the directory of the selected platform.The image is finally packaged into multiple image files such as ramdisk.img, system.img, and userdata.img.In addition to the root file system ramdisk,the other two images are mounted to their system and data directories after decompression.

#### 4.3 Detailed implementation of system functions

The application framework layer mainly provides an interface for the application layer.The application layer's call to the local library is implemented by JNI<sup>[19]</sup>.Here, you need to register the method you added and establish the connection through the function pointer.The camera in the application framework layer is equivalent to the client.The most basic architecture in Android is the C/S layer architecture.Most of the functions are provided in the form of system services.The client provides access to the server interface.The work is still done through the server,the camera is also used in

this architecture:the camera client process does not have any substantial camera data,but the server provides a complete service interface, the camera client can be easily obtained the storage address of the camera data can then be processed quickly.

The application layer is on the top,written by Android application developers,mainly designed the UI design and interaction of the camera,writing logic code.Next is the application framework layer,which provides a stable API for the application layer to ensure the most basic camera functionality<sup>[20]</sup>.Next is the JNI middleware,which can add a new camera function interface.The upper layer accesses the localization library and also needs to apply JNI technology.The Java method and the Native method are used to establish contact here.

Finally,we need to implement the camera client.We link the server here to implement the sending and receiving process.

The implementation of the Camera Client side is mainly to call the interface and method,and does not really implement the function.We implement the interface function implementation in the server side<sup>[21]</sup>.As mentioned above,the server side implements the RPC mechanism through the Binder mechanism to complete the camera subsystem architecture we designed.

The Camera HAL design mainly implements two control threads.

The picture thread,when the user uses the takePicture function,the system calls the camera thread,the program monitors the frame data in the buffer queue,and takes out the frame data from the buffer queue after the data arrives.We pass the data obtained by the camera to the Java layer,and the original data can be compressed into JPEG format and sent to the upper layer.

The bottom of the HAL layer exists as a daemon.Here,the core service of the camera is provided,the underlying camera driver is packaged,and the camera-related software stack is constructed.The C/S structure of the camera does not affect the upper layer,and the data processing and distribution are directly performed here to reduce the coupling.

## 5 Conclusions

The goal of this design is to develop Android source code level in Ubuntu system,which can be realized and verified by modifying Camera HAL.

The core of the design is the design of hardware abstraction layer,which is realized by modifying the Android platform.

The emphasis of development programming is as follows:

- (1) Improve and optimize the design of Android native HAL and put forward its own HAL model.
- (2) Write the corresponding Camera HAL code for the design of HAL.
- (3) Compile and debug the program.

Specific tasks can be divided into:

- 1.Research the existing Android HAL design and embedded device HAL design, arrange the main process of development and design.

2.Theory learning - learning Android system-level development tools and methods, familiar with the use of ubuntu.

3.In the Android development environment that has been built,the corresponding coding work is realized by combining the HAL model designed.

4.Testing Camera module in application layer,improving the design of HAL layer through testing.

5.Finally,after comprehensive adjustment,report and paper will be compiled according to the results of implementation.

There are five objectives: first,to accurately analyze the key technologies needed in the design of Android HAL.secondly,to develop and code Android at the source level in Ubuntu system; secondly, to test Camera HAL module through application software at the application level.finally,to get the analysis report and paper collation after getting the test results.

After three months'efforts,we have basically realized the preview and photography functions of Camera subsystem,and improved the hardware abstraction layer of Android.The driver uses a general V4L2 framework,which is universal and improves the running efficiency of Camera subsystem.In the aspect of HAL design,the realization of photo threads and preview threads is completed.An optimization scheme is proposed for HAL.Core Service is added to the lower layer of HAL to improve the efficiency of program and realize the integration of software and hardware. In the later stage,it is necessary to improve the functions and further optimize the structure of HAL in order to achieve efficiency and versatility.Find a balance between them.

There is no standard evaluation of efficiency and performance.In the process of further optimization of the framework,hot spot function analysis,time stamp,process analysis and other methods should be introduced to test the performance of the system,including photo delay,focusing delay and so on.Good test system and targeted optimization can make the designed Camera subsystem more robust and efficient.

## References

1. Bradley, A.P.: The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognit.* 30(7), 1145–1159 (1997).
2. Arora, A., Peddoju, S.K.: Minimizing network traffic features for android mobile malware detection. In: *Proceedings of the 18th International Conference on Distributed Computing and Networking, ICDCN '17*, pp. 32:1–32:10. ACM (2017).
3. Cimpanu, C.: CopyCat adware infects Zygote Android core process, bleepingcomputer.
4. Grace, M., Zhou, Y., Zhang, Q., Zou, S., Jiang, X.: Riskranker. In: *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, MobiSys '12* (2012).
5. Kapratwar, A., Troia, F.D., Stamp, M.: Static and dynamic analysis of android malware. In: *Proceedings of the 3rd International Conference on Information Systems Security and Privacy* (2017).



6. Liu, B., Nath, S., Govindan, R., Liu, J.: DECAF: Detecting and characterizing ad fraud in mobile apps. In: Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation, NSDI'14, pp. 57–70. USENIX Association (2014).
7. Moonsamy, V., Rong, J., Liu, S.: Mining permission patterns for contrasting clean and malicious Android applications. *Future Gener. Comput. Syst.* 36, 122–132 (2014).
8. Sharma, D.: Android malware detection using decision trees and network traffic. *Int. J. Comput. Sci. Inf. Technol.* 7(4), 1970–1974 (2016).
9. Stamp, M.: *Introduction to Machine Learning with Applications in Information Security*. Chapman and Hall/CRC, Boca Raton (2017).
10. Yan, L.-K., Yin, H.: DroidScope: Seamlessly reconstructing the OS and Dalvik semantic views for dynamic Android malware analysis. In: *USENIX Security Symposium*, pp. 569–584 (2012).
11. Zhang, L., Guan, Y.: Detecting click fraud in pay-per-click streams of online advertising networks. In: *The 28th International Conference on Distributed Computing Systems, ICDCS'08*, pp. 77–84. IEEE (2008).
12. Aafer Y, Du W, Yin H (2013) Droidapiminer: mining api-level features for robust malware detection in android. In: *Proceedings of international conference on security and privacy in communication systems*, pp 86 - 103.
13. Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K.: Drebin: Effective and explainable detection of android malware in your pocket. In: *21st Annual Network and Distributed System Security Symposium, NDSS 2014*. The Internet Society (2014).
14. Bradley, A.P.: The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognit.* 30(7), 1145 - 1159 (1997).
15. Crussell, J., Stevens, R., Chen, H.: Madfraud: investigating ad fraud in Android applications. In: *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 123 - 134. ACM (2014).
16. Daswani, N., Stoppelman, M.: The anatomy of Clickbot.A. In: *Proceedings of the First Conference on Hot Topics in Understanding Botnets, HotBots ' 07*, pp. 11 - 31. USENIX Association, Berkeley (2007).
17. Enck, W., Gilbert, P., Han, S., Tendulkar, V., Chun, B.-G., Cox, L.P., Jung, J., McDaniel, P., Sheth, A.N.: TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Trans. Comput. Syst.* 32(2), 5 (2014).
18. Miller, B., Pearce, P., Grier, C., Kreibich, C., Paxson, V.: What ' s clicking what? Techniques and innovations of today ' s clickbots. In: *Proceedings of the 8th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA'11*, pp. 164–183. Springer (2011).
19. Qiu, H., Noura, H. , Qiu, M.:A User-Centric Data Protection Method for Cloud Storage Based on Invertible DWT. *IEEE Transactions on Cloud Computing (Early Access)*,pp.1–1,2017.
20. Qiu, H. , Memmi, G.: Fast Selective Encryption Method for Bitmaps Based on GPU Acceleration. *IEEE International Symposium on Multimedia*. IEEE(2014).
21. Qiu, H. , Memmi, G., Kapusta, K., Lu, Z., Qiu,M.: All-Or-Nothing data protection for ubiquitous communication: Challenges and perspectives.*Information Sciences*, 434-445 (2019).