

MICROOH 麦可网

# Android-从程序员到架构师之路

出品人：Sundy

讲师：高焕堂（台湾）

<http://www.microoh.com>

D01\_b

# 认识Android核心服务(b)

By 高煥堂

### 3、区分两种核心服务：

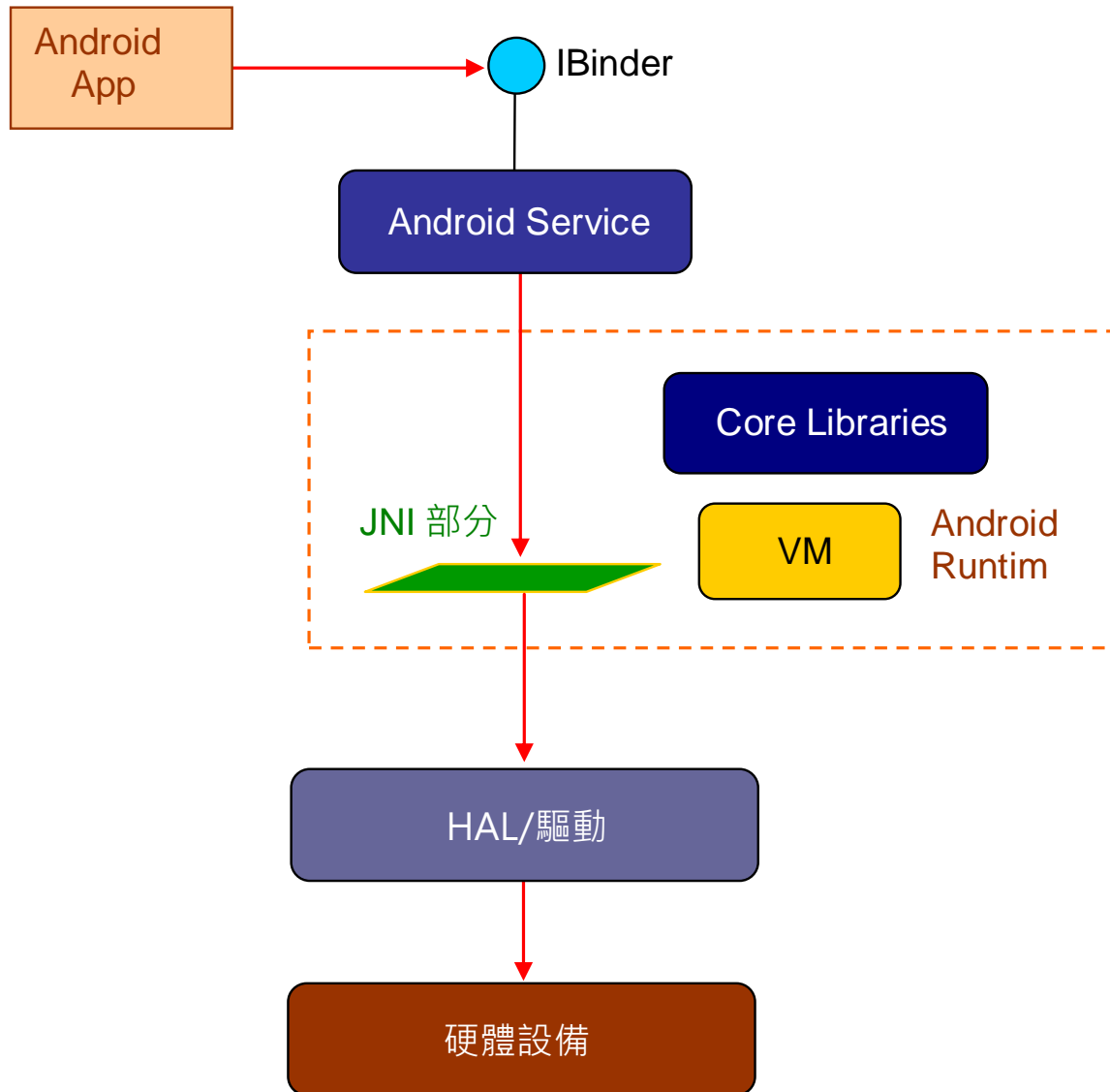
Android Service与Native Service

- 核心服务可以用Java撰写；也可以用C++撰写。
- 以Java撰写的核心服务通称为**Android Service**(例如AudioService和SensorService等)；
- 以C++撰写的核心服务则通称为**Native Service** (例如MediaPlayerService和CameraService等)。



# Android Service

- Android Service又称为Java Service，是实现于应用框架层(Framework)里的。
- 这里所讲的Service是系统服务(System Service)，又称为Server；其与App开发上所讨论的Service不同。
- Android Service以Java撰写。

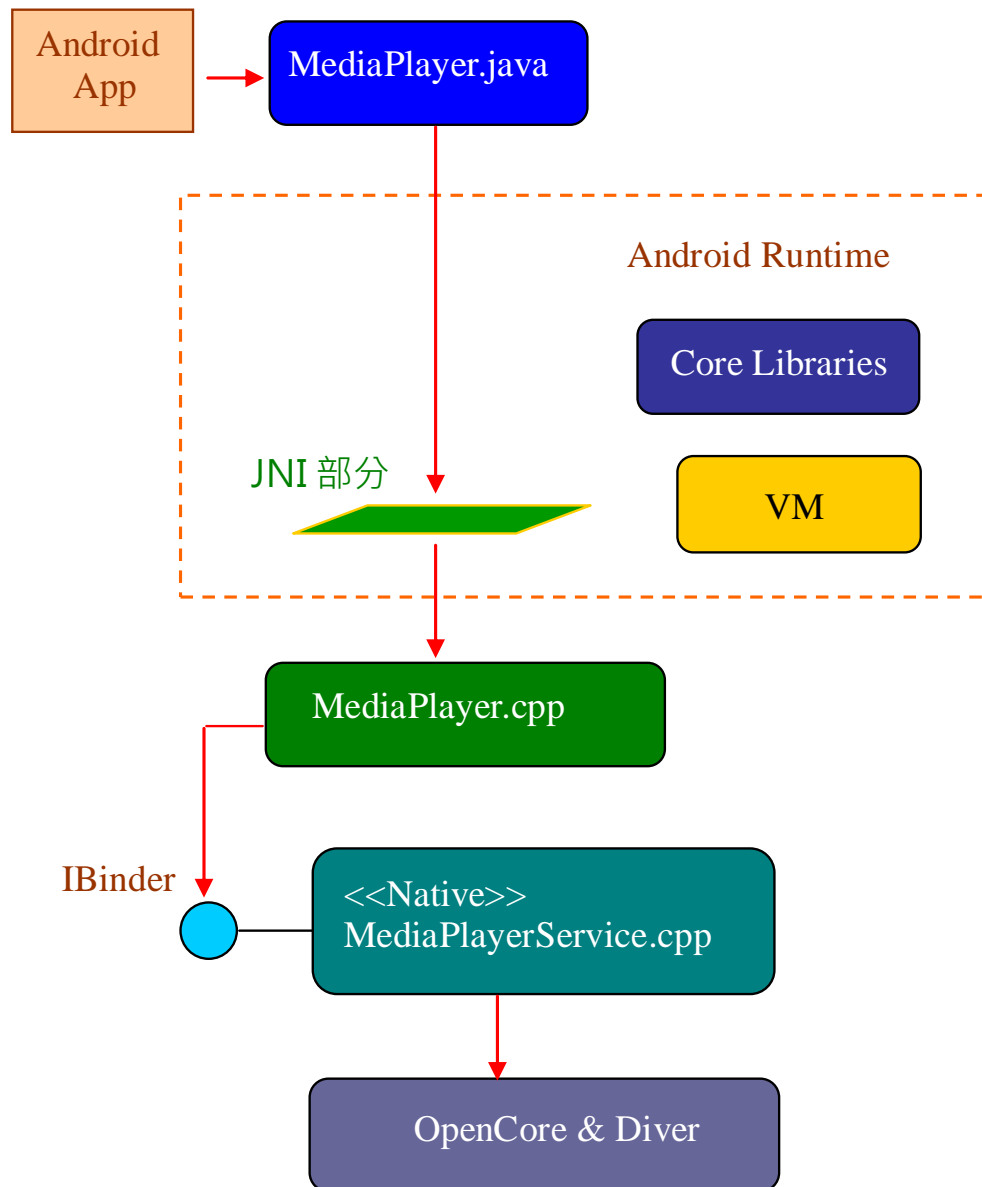


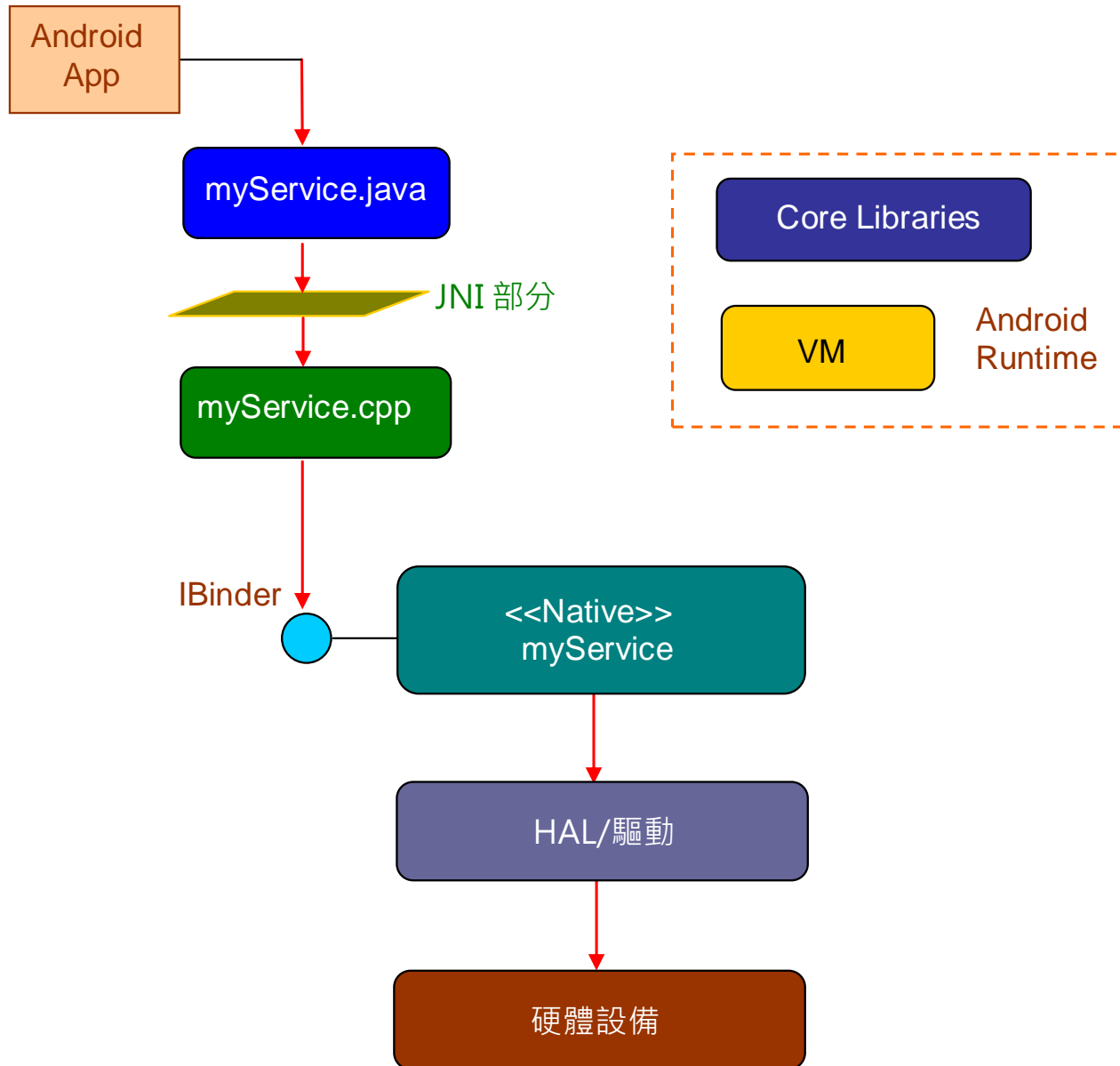
# Native Service

- Native Service则是实现于Runtime层里的Server。在系统服务开发上，我们有二个途径，一个是写成Android Service、其再透过JNI与HAL驱动沟通；另一个途径是，跳过Android Service，让App透过JNI直接与Native Service沟通。



EX.

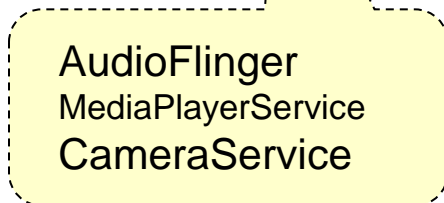
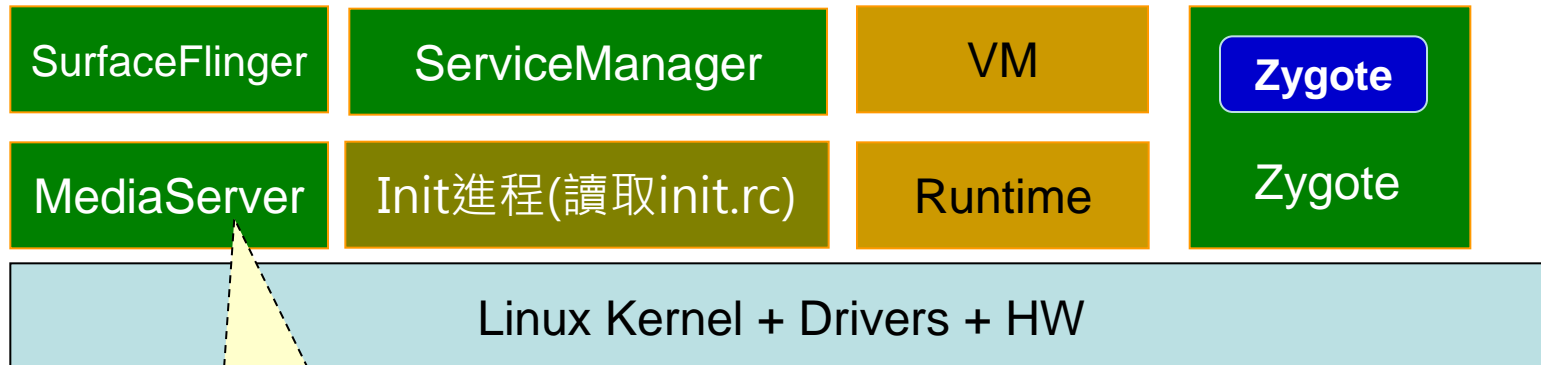
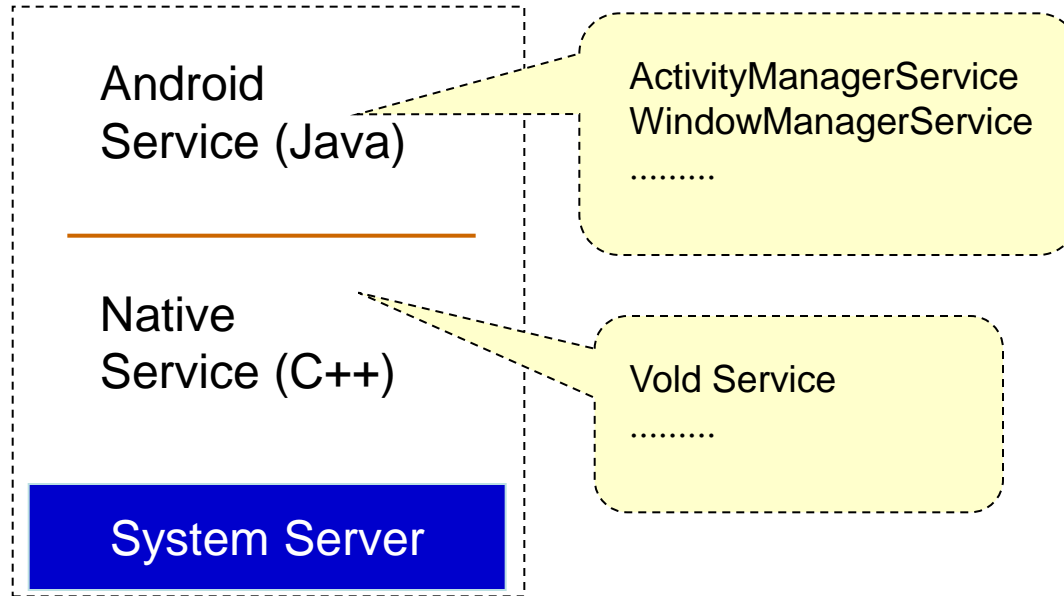




## 4、核心服务的特性

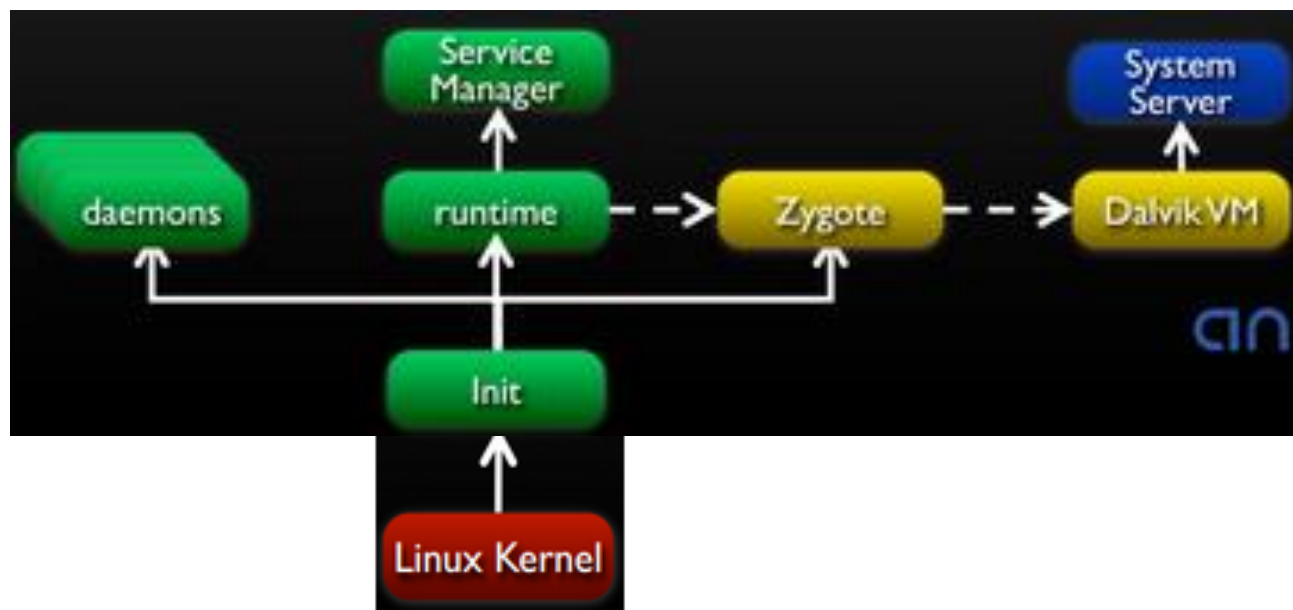
# 核心服务

- 其中，核心服务属于Android平台的系统服务模块，在开机过程，就会启动数十个核心服务；例如MediaPlayerService、AudioService、SensorService等。



- 关于核心服务的启动，就是在开机时，加载Linux kernel部分，进行Kernel-Space的初始化，然后加载硬件驱动程序，就启动了Linux系统。随后切换至User-Space，创建init进程，读取 init.rc文档，依据其内容的指示而先启动Native Service，再启动Android Service。就完成核心服务的启程序了。

- 由Linux内核启动用户空间的Init进程
  - \* 解析脚本文件：Init.rc
  - Init.rc是Android的初始化脚本
- 由Init进程创建ServiceManager和Zygote进程。
- 由Zygote创建(VM和)SystemServer进程。





# 核心服务的特性

- 核心服务通常在特定的进程里执行。
- 必须提供IBinder接口，让App进行跨进程的绑定(Bind)和呼叫。
- 因为共享，所以必须确保多线程安全(Thread-safe)
- 以C++类别或Java类别定义，诞生其对象，请ServiceManager(简称SM)将该对象参考值加入到Binder Driver里。

- App可请SM协助而远距绑定某核心服务，此时SM会传IBinder接口给App。
- App可透过IBinder::transact()函数来与核心服务互传讯息。

- 核心服务启动在先，应用程序启动在后。
- 核心服务能以C++撰写；也可以用Java撰写(如AudioService)。
- 核心服务与App服务不同，App服务不会加入到Binder Driver里。

## 5、简介几个 Android Service

- 在Android Service里，常见的有如  
ActivityManager、LocationManager等。  
兹针对其中的几个常见服务介绍如下：

- **Activity Manager(活动管理器)** :  
提供所有的应用程序的生命周期，以及Activity之间的互动之服务。
- **WindowManager(窗口管理器)** :  
提供由关频幕窗口之管理服务。
- **ResourceManager(资源管理器)** :  
管理程序使用的各项资源，例如：字符串、图片、UI布局(Layout)定义档。

- **LocationManager(位置管理器)**：  
提供位置服务(Location-based Service)。
- **TelephonyManager(电话管理器)**：  
提供有关手机通话的服务，例如关闭电话等。
- 其它服务。

## 6、简介几个 Native Service



- Native Service是以C++语言所撰写的核心服务。

- 在Native Service里，大家比较熟悉得有如ServiceManager、MediaPlayerService、Zygote、CameraService等等。
- 兹针对其中的几个常见服务介绍如下：
- **ServiceManager(服务管理器)**：  
协助登录与绑定系统服务。

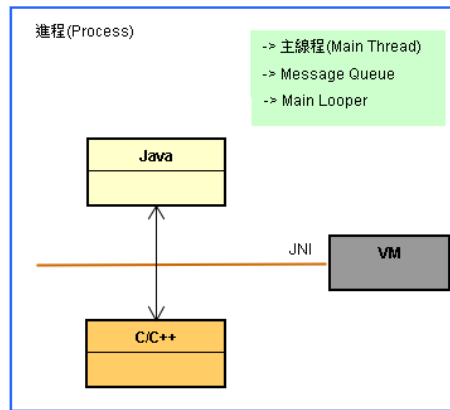
- ServiceManager是Android平台里的一个基本模块。在Android系统启动过程中，它就会被登录到Binder Kernel里，成为天字第一号的可提供远距服务之模块。

- 使用C/C++撰写Android的NativeService模块时，可以透过defaultServiceManager()函数来取得ServiceManager模块的IServiceManager接口之指针。例如，在CameraService.cpp程序里的instantiate()函数，其内容如下：

```
// CameraService.cpp
void CameraService::instantiate() {
    defaultServiceManager()->addService(
        String16("media.camera"), new CameraService());
}
```

- 它使用了defaultServiceManager()函数来取得ServiceManager模块，然后呼叫其IServiceManager接口里的addService()函数来将自己对象(即CameraService类别之对象)传给ServiceManager模块，请它登录到Binder Kernel里，成为一个可提供远距服务之模块(即CameraService模块)。

- **Zygote服务**：Android最内层有个Zygote服务。它是Android Java层的孵化器。Zygote就在幕后复制孵化(Fork)出一个子进程来给这些Java层的系统服务或应用程序。所以，Android的应用程序，以及上述的各项服务，也都是由zygote所繁殖出来的。Zygote服务的架构位置如下图：



System Server

SurfaceFlinger

ServiceManager

VM

Fork Service

MediaServer

Init進程(讀取init.rc)

Runtime

Zygote

Zygote

Linux Kernel + Drivers + HW



- **CameraService(摄像服务)** :  
其实作于libcameraservice.so共享库里，与底层Camera 硬件的 HAL驱动程序衔接，藉由Linux 的 videodev接口来撷取视讯。
- 其它服务。





**~ Continued ~**