

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>

D02_b

撰写你的第一个核心服务(b)

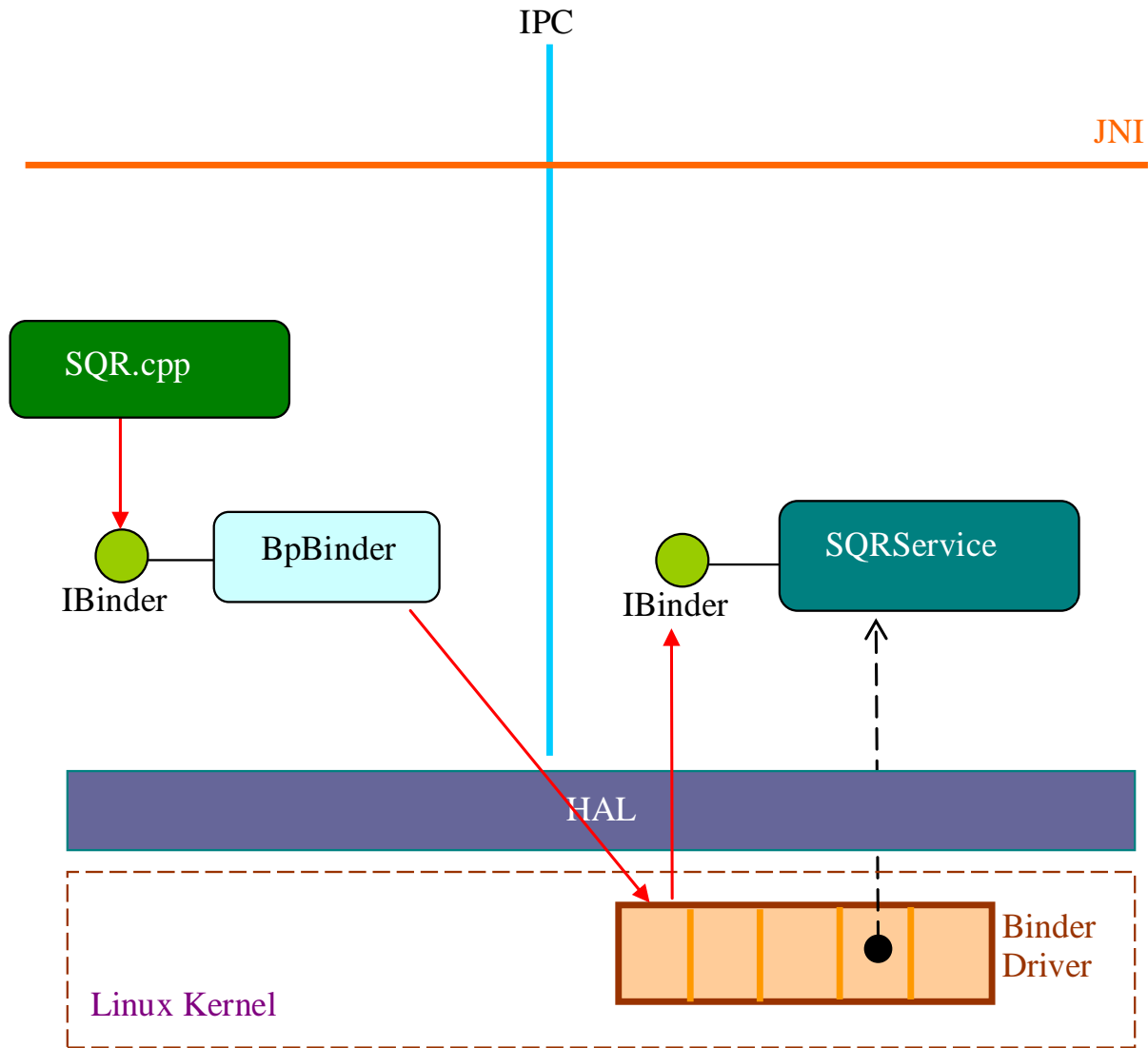
By 高煥堂

3、撰写SQR.CPP 來 使用SQRService

階段一：撰写一个本地层Client

以C++撰写一个SQR.CPP

- 在本节里，兹撰写一个SQR.CPP程序使用SQRService。
- **步骤2.1:**使用 C++ 开发一个名为SQR.cpp的C++层应用程序，它透过ServiceManager去绑定(Bind)了SQRService。
- 然后调用IBinder::transact()函数，进而调用SQRService核心服务的onTransact()去进行「平方」的服务。其架构图如下：



- 于是，使用C++撰写一个SQR类，如下：

```
// SQR.h
#ifndef ANDROID_MISOO_SQR_H
#define ANDROID_MISOO_SQR_H
namespace android {
class SQR {
    private:
        const void getSQRService();
    public:
        SQR();
        int execute(int n);
};
}; //namespace
#endif // ANDROID_MISOO_SQR_H
```

```
// SQR.cpp
#include <utils/IServiceManager.h>
#include <utils/IPCThreadState.h>
#include "SQR.h"
namespace android {
    sp<IBinder> m_ib;
    SQR::SQR(){
        getSQRService();
    }
    const void SQR::getSQRService(){
        sp<IServiceManager> sm = defaultServiceManager();
        m_ib = sm->getService(String16("misoo.sqr"));
        LOGE("SQR:getSQRService %p\n",sm.get());
        if (m_ib == 0)
            LOGW("SQRService not published, waiting...");
        return;
    }
}
```



```
int SQR::execute(int n) {  
    Parcel data, reply;  
    data.writeInt32(n);  
    LOGE("SQR::execute\n");  
    m_ib->transact(0, data, &reply);  
    int num = reply.readInt32();  
    return num;  
}  
}; //namespace BpBinder
```

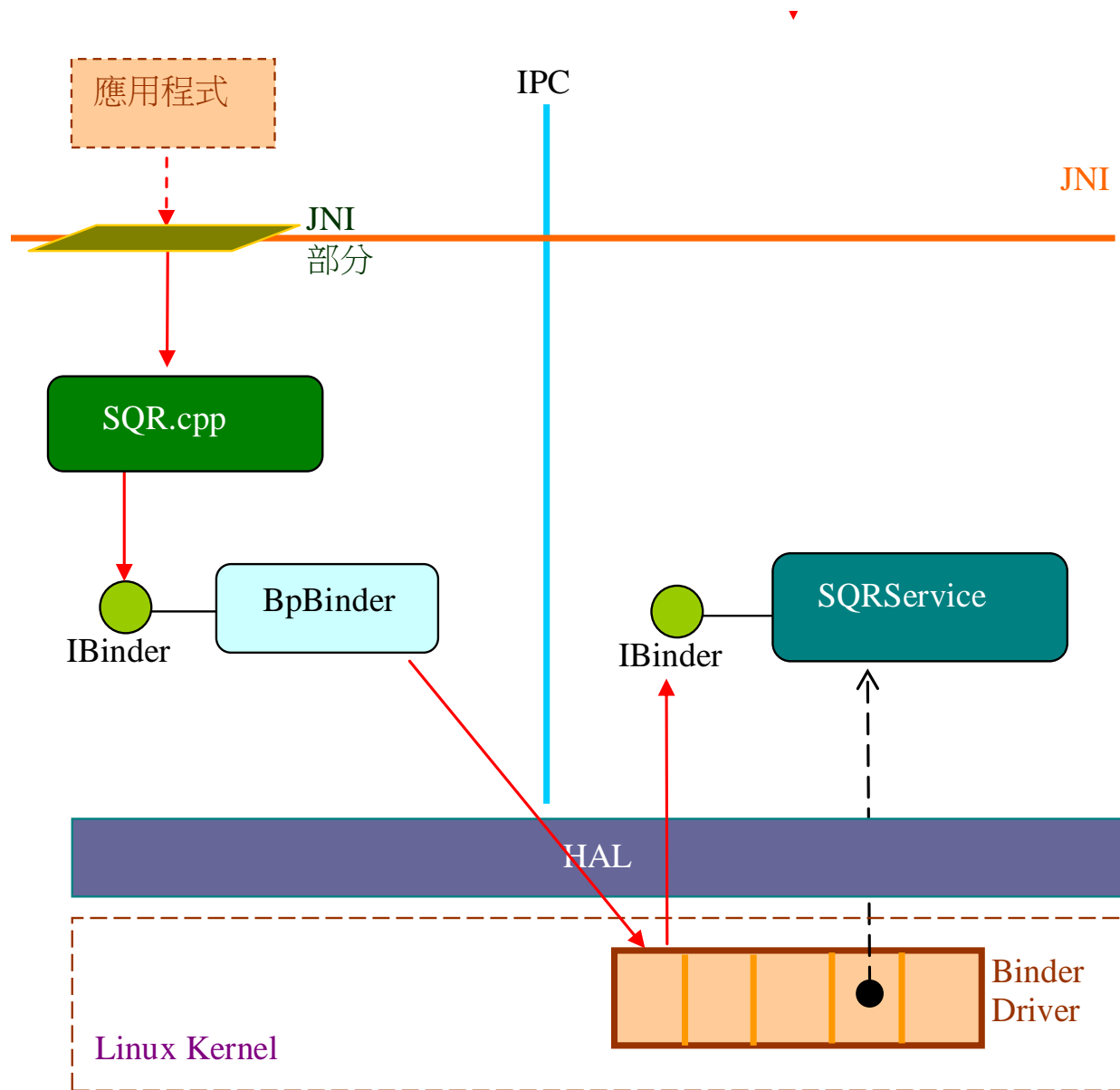
- 在构造式SQR()里调用getService來获得服务(亦即绑定服务)。
- 取得服务时 ServiceManager 传回 BpBinder 的IBinder 接口。
- 而execute()函数则实际调用IBinder接口的transact()函数，并转而调用onTransact()函数。

4、撰写Java层Client 來使用核心服务

階段三：撰写一个Java层Client

以Java撰写一个sqr01.java

- Java层的应用程序可以透过JNI Native Code来调用上图里的SQR类别，进而取得SQRService的服务。其架构图如下：



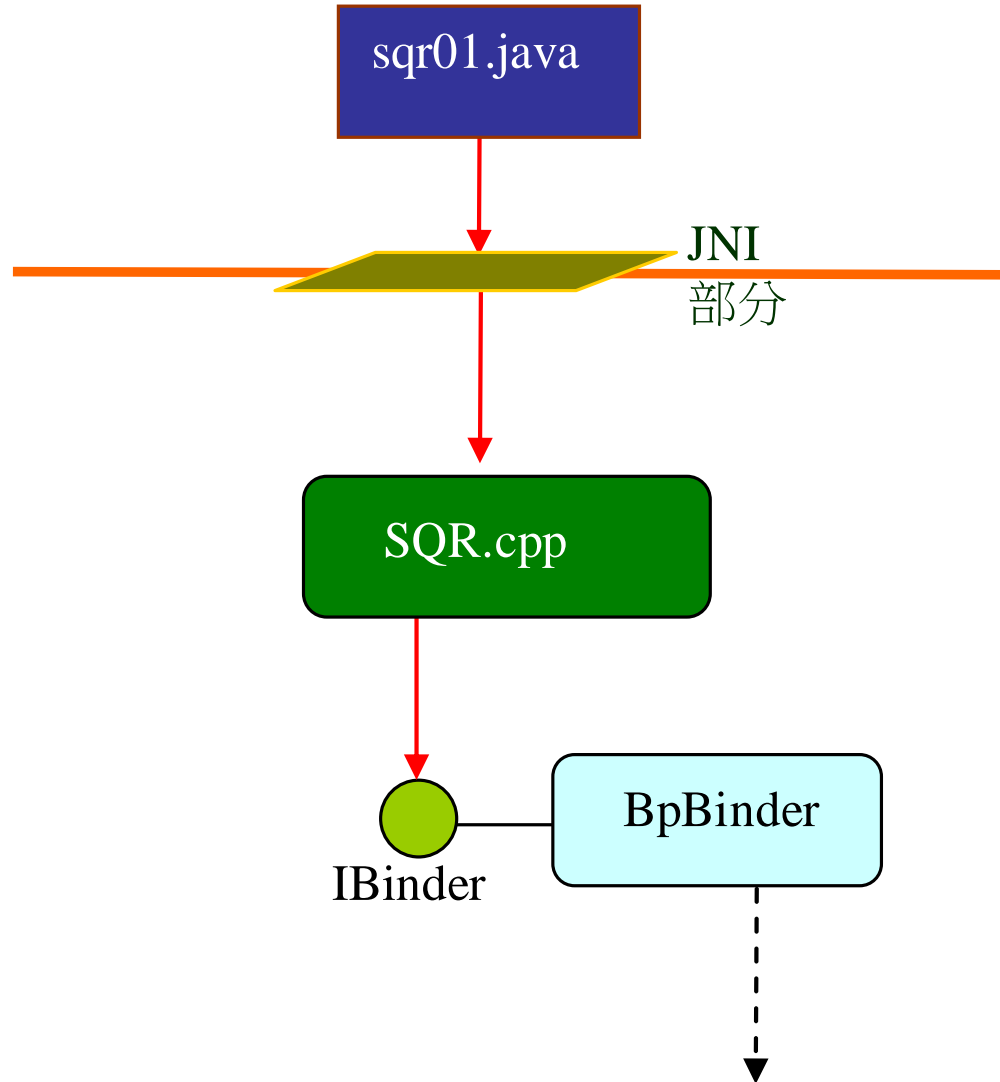
- **步骤3.1:** 使用C语言撰写JNI Native Code，来当作Java应用程序与SQR.cpp之间的桥梁，如下：

```
/* com_misoo_service_sqr01.cpp */
#include "com_misoo_service_sqr01.h"
#include <utils/Log.h>
#include <utils/IPCThreadState.h>
#include <utils/ProcessState.h>
#include "../core_service/SQRService.h"
#include "SQR.h"
using namespace android;
```

```
JNIEXPORT jint JNICALL
```

```
Java_com_misoo_service_sqr01_execute(JNIEnv *env, jobject thiz,
    jint x) {
    char s[] = "Java_com_misoo_service_sqr01_execute!!!";
    LOGE("%s X = %d\n", s, x);
    SQR* sqrObj = new SQR();
    int num = sqrObj->execute(x);
    return num;
}
```

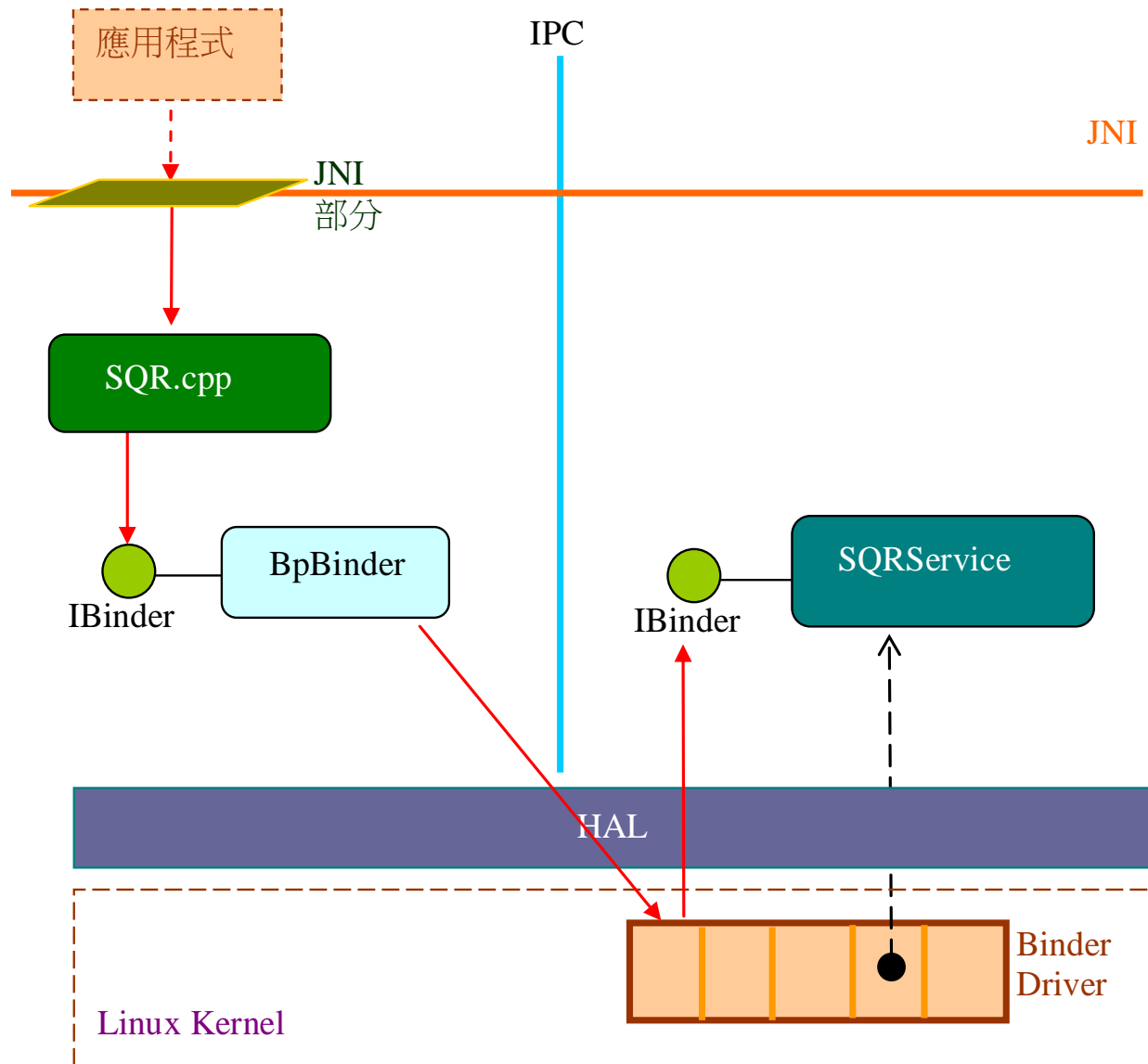

- 执行 Android.mk 档，产出 libSQRS01_jni.so 共享程序库，就可以加载手机或模拟器裡执行了。
- 执行时，Native 函數调用SQR，转而调用BpBinder，如下图所示。



- 设计Java层的应用类别来调用上述的JNI Native Code，就完成串接Java应用程序与硬件驱动的目标了。
- 于是，撰写一个与JNI Native Code相对应的Java类别：

```
// sqr01.java  
package com.misoo.service;  
public class sqr01  
{  
    static  
        { System.loadLibrary("libSQRS01_jni"); }  
    public static native int execute(int x);  
}
```

- 当此sqr01类别被加载到内存时，就会把libSQRS01_jni.so也加载进来，于是sqr01类别就与libSQRS01_jni.so结合起来了；而且sqr01.execute()函数就能调用到libSQRS01_jni.so里的JNI Native函数了。





~ Continued ~