

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>

D06\_a

# 核心服务Callback的 IBinder接口设计(a)

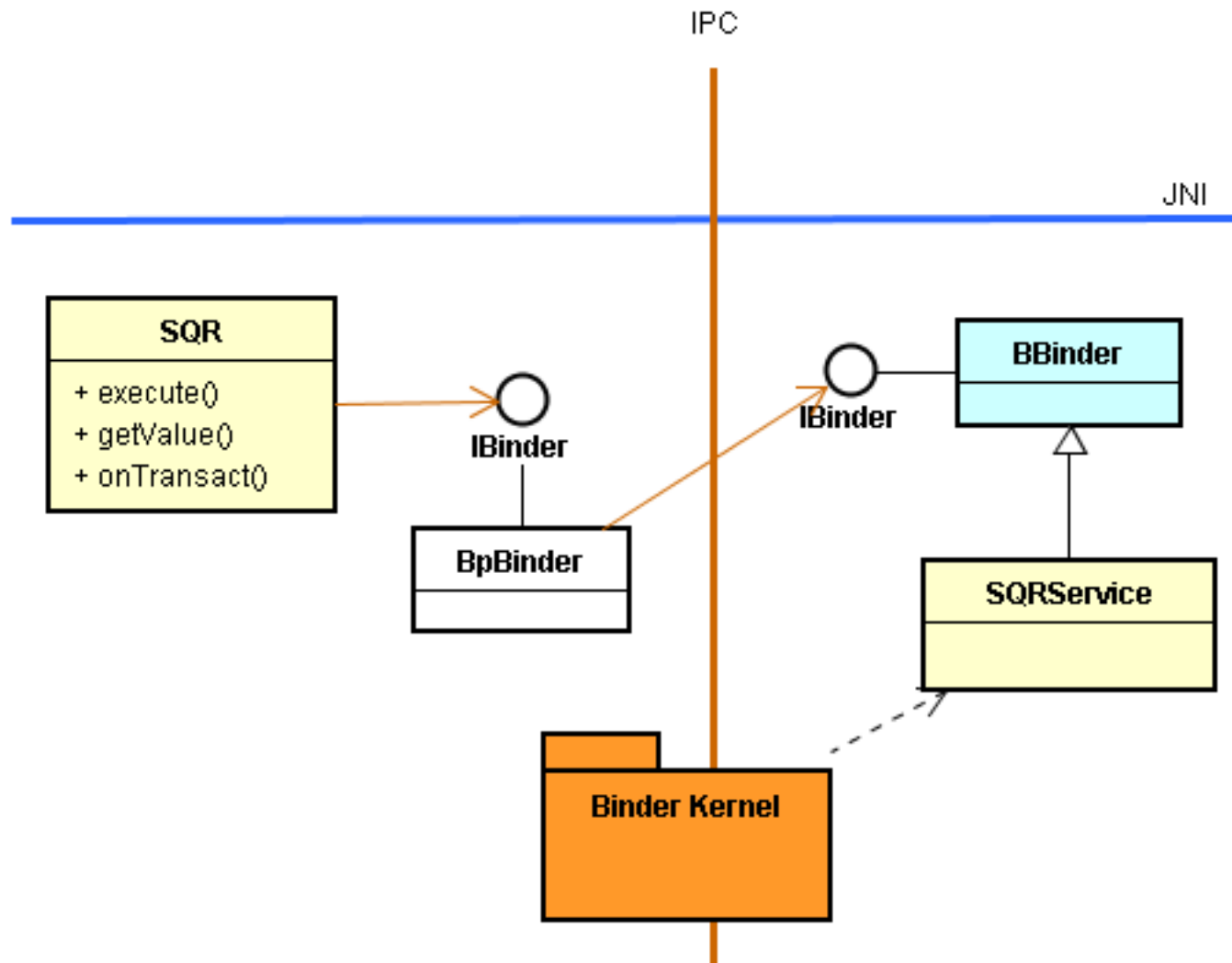
By 高煥堂

# 内容

1. 正向调用核心服务
2. 核心服务的远程回调(Callback)
3. Native Service范例代码实现
4. Client模块范例代码实现
5. Summary

# 1、正向调用核心服务

- 在前面，你已经亲自撰写了SQRService核心服务，而且撰写了SQR类别去绑定和调用该核心服务。如下图：



- 这是典型的正向调用途径。然而，在实务上，也经常需要从核心服务回调(Callback)到Client模块。

## 2、核心服务的远程回调 (Callback)

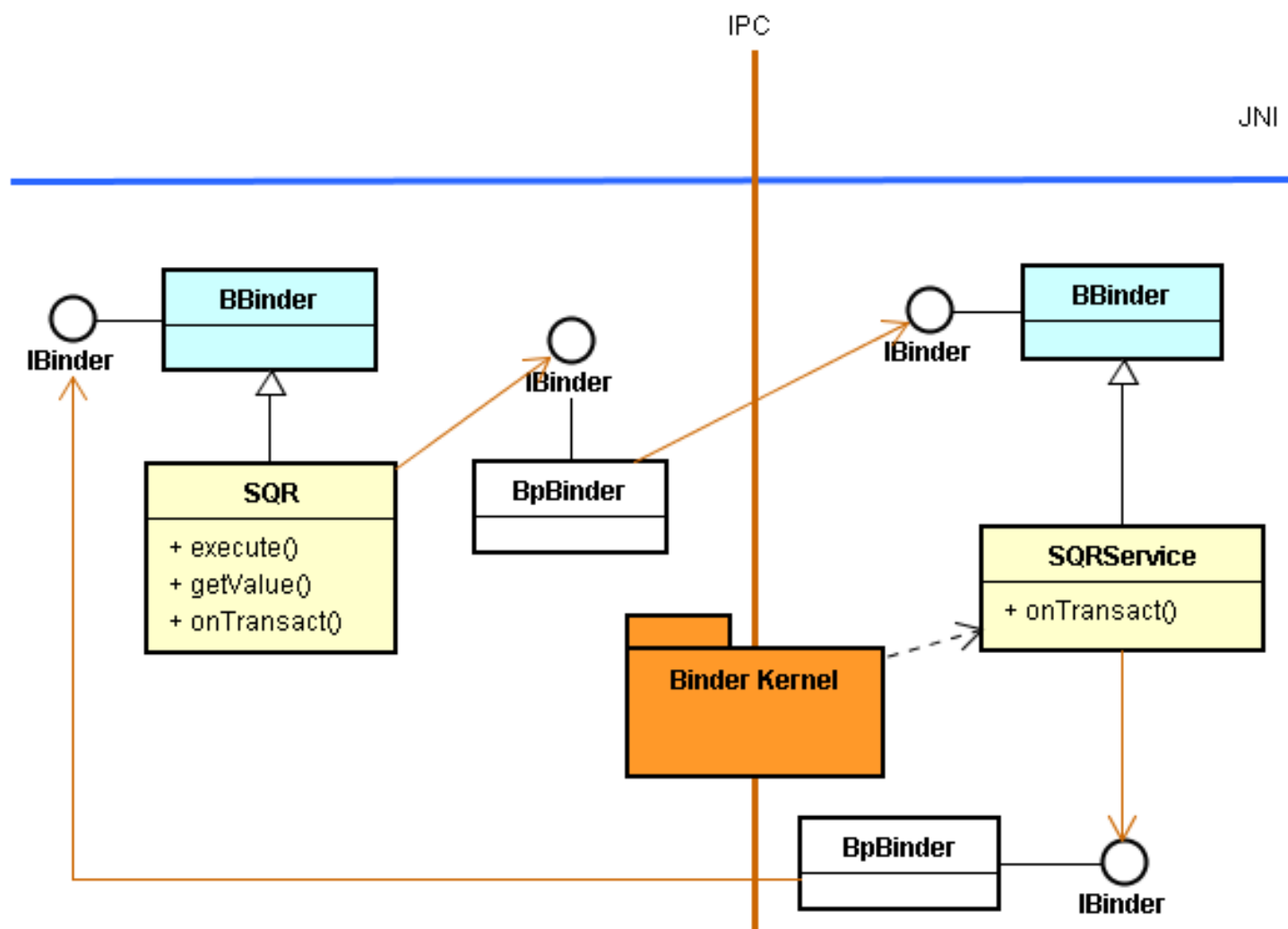


- 兹设想，如果SQRService负责播放一首歌曲；但很不幸地，播放到半途时，突然侦测到喇叭硬件错误(Hardware Error)了，此时SQRService必须回调Client模块，向用户提报错误讯息。

- 由于Client模块是在另一个进程里，SQRService是进行IPC回调到Client模块，此时也必须透过IBinder接口才行。于是，可以推论得知：

1. Client模块必须提供IBinder接口。
2. 实现方法就是：让Client模块继承BBinder基类。

- 如下图所示：



# 3、Native Service 范例代码实现

- 撰写SQRService核心服务

```
// SQRService.h
#ifndef ANDROID_MISOO_SQR_SERVICE_H
#define ANDROID_MISOO_SQR_SERVICE_H
#include <utils.h>
#include <utils/RefBase.h>
#include <utils/Interface.h>
#include <utils/Parcel.h>
namespace android {
class SQRService : public BBinder
{
    sp<IBinder> callback_ib;
public:
    static int    instantiate();
    virtual status_t onTransact(uint32_t, const Parcel&, Parcel*, uint32_t);
};
}; //namespace
#endif
```

```
// SQRService.cpp
#include "SQRService.h"
#include <utils/IServiceManager.h>
#include <utils/IPCThreadState.h>
namespace android {
int SQRService::instantiate()
{
    LOGE("SQRService instantiate");
    int r = defaultServiceManager()->addService(String16("misoo.sqr"),
                                                new SQRService());
    LOGE("SQRService r = %d\n", r);
    return r;
}
```



```
status_t SQRService::onTransact(uint32_t code, const Parcel& data,
                                Parcel* reply, uint32_t flags){
    switch(code) {
        case 0: {
            int x = data.readInt32();
            callback_ib = data.readStrongBinder();
            int k = x * x;
            Parcel data1, reply1;
            data1.writeInt32(k);
            callback_ib->transact(0, data1, &reply1);
            LOGE("SQRService::onTransact. %d\n", x);
            return NO_ERROR;
        }
        break;
        default:
            return BBinder::onTransact(code, data, reply, flags);
    }
}; //namespace
```

- 其中的指令：

```
callback_ib->transact(0, data1, &reply1);
```

就是远距回调到Client模块。





**~ Continued ~**