

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）

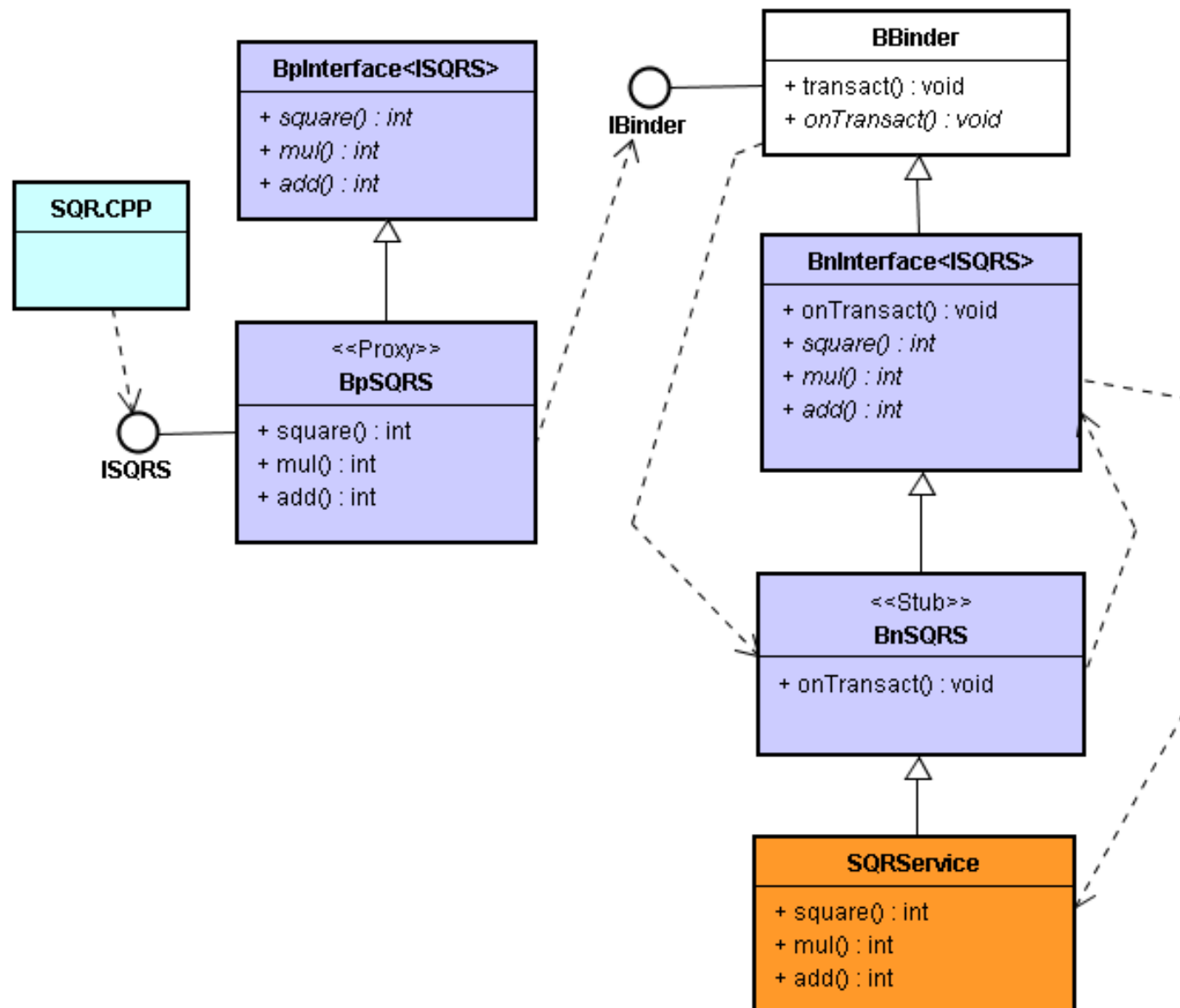
<http://www.microoh.com>

D04_b

亲自开发SQRService的 Proxy-Stub设计模式(b)

By 高煥堂

2、撰写 Proxy类代码



定义ISQRS接口

```
class ISQRS: public IInterface{  
public:  
    DECLARE_META_INTERFACE(SQRS);  
    virtual int square(const int& n) = 0;  
};
```

定义Proxy类

- 利用“ 模板+接口” ，产生 Proxy模板类：
BpInterface<ISQRS>类。
- 并衍生出Proxy类：BpSQRS类。

```
// ISQRS.h
#include <utils/RefBase.h>
#include <utils/Interface.h>
#include <utils/Parcel.h>
#ifndef ANDROID_MISOO_ISQRS_SERVICE_H
#define ANDROID_MISOO_ISQRS_SERVICE_H
namespace android {

class ISQRS: public IInterface{
public:
    DECLARE_META_INTERFACE(SQRS);
    virtual int square(const int& n) = 0;
};
```

```
//-----  
class BpSQRS: public BpInterface<ISQRS>{  
public:  
    BpSQRS(const sp<IBinder>& impl): BpInterface<ISQRS>(impl){}  
    virtual int square(const int& n);  
};  
}; // namespace android  
#endif
```


撰写BpSQRS类的实现代码

```
// ISQRS.cpp
#include "ISQRS.h"
namespace android {
enum {
    SQUARE = IBinder::FIRST_CALL_TRANSACTION,
};
int BpSQRS::square(const int& n) {
    Parcel data, reply;
    data.writeInt32(n);
    remote()->transact(SQUARE, data, &reply);
    int num = reply.readInt32();
    return num;
}
IMPLEMENT_META_INTERFACE(SQRS, "android.misoo.IAS");
}; // namespace android
```

3、撰写Client类代码

- 撰写一个SQR类来使用核心服务。此SQR类调用BpSQRS的ISQRS接口，再转而SQRService的通用性IBinder接口。

```
// SQR.h
#ifndef ANDROID_MISOO_SQR_H
#define ANDROID_MISOO_SQR_H
#include "../core_service/ISQRS.h"
namespace android {

class SQR{
public:
    int execute(int n);
private:
        static const sp<ISQRS>& getSQRService();
        static sp<ISQRS>    sSQRService;
};
}; // namespace android
#endif
```

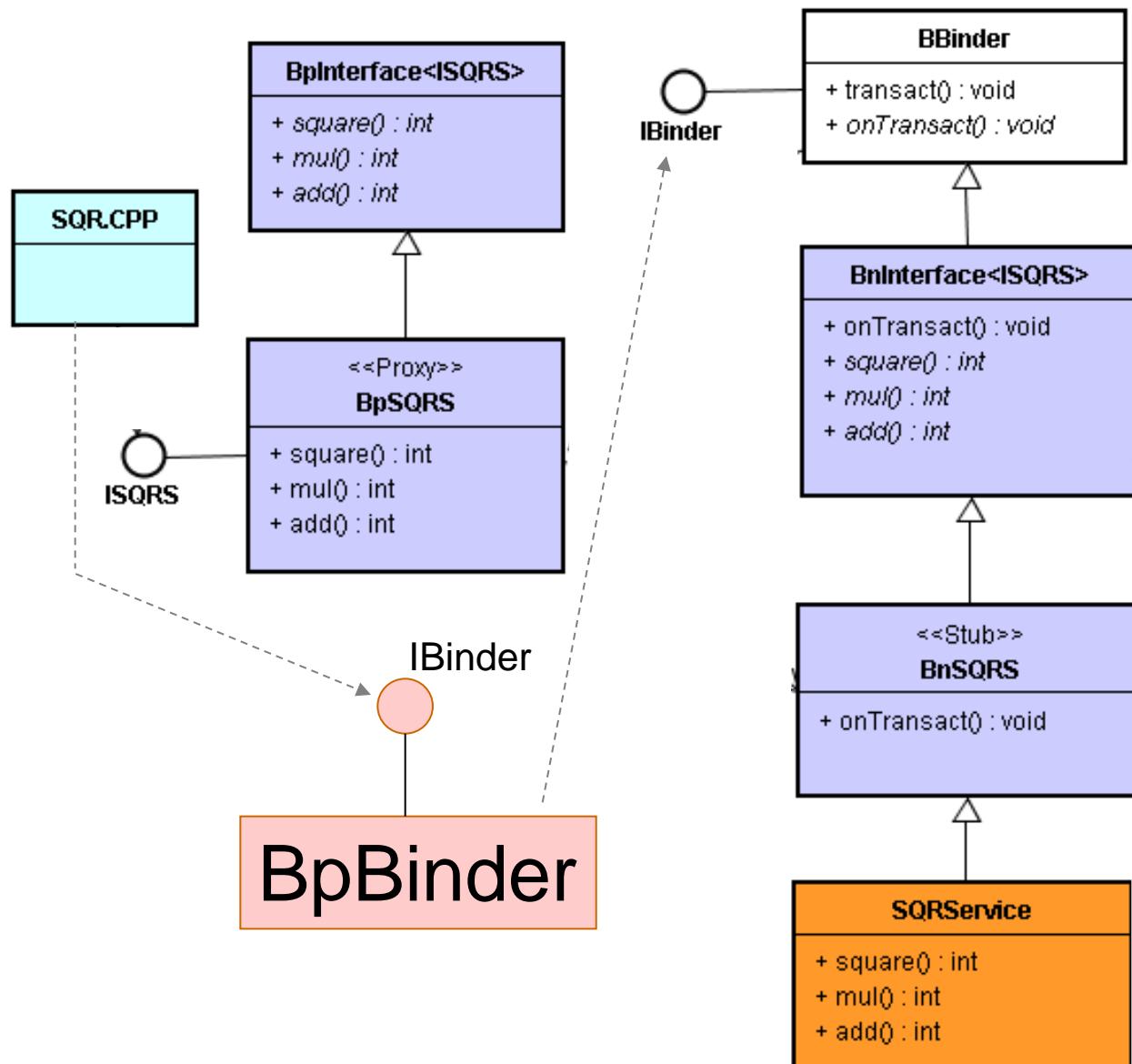
```

// SQR.cpp
#include <utils/IServiceManager.h>
#include <utils/IPCThreadState.h>
#include "SQR.h"
namespace android {
const sp<ISQRS>& SQR::getSQRService(){
    sp<IServiceManager> sm = defaultServiceManager();
    sp<IBinder> ib = sm->getService(String16("misoo.sqr"));
    LOGE("SQR::getSQRService\n");
    sp<ISQRS> sService = interface_cast<ISQRS>(ib);
    return sService;
}
int SQR::execute(int n){
    int k =0;
    const sp<ISQRS>& isqrs( getSQRService() );
    if (isqrs != 0) k = isqrs->square(n);
    return k;
}}; // namespace android

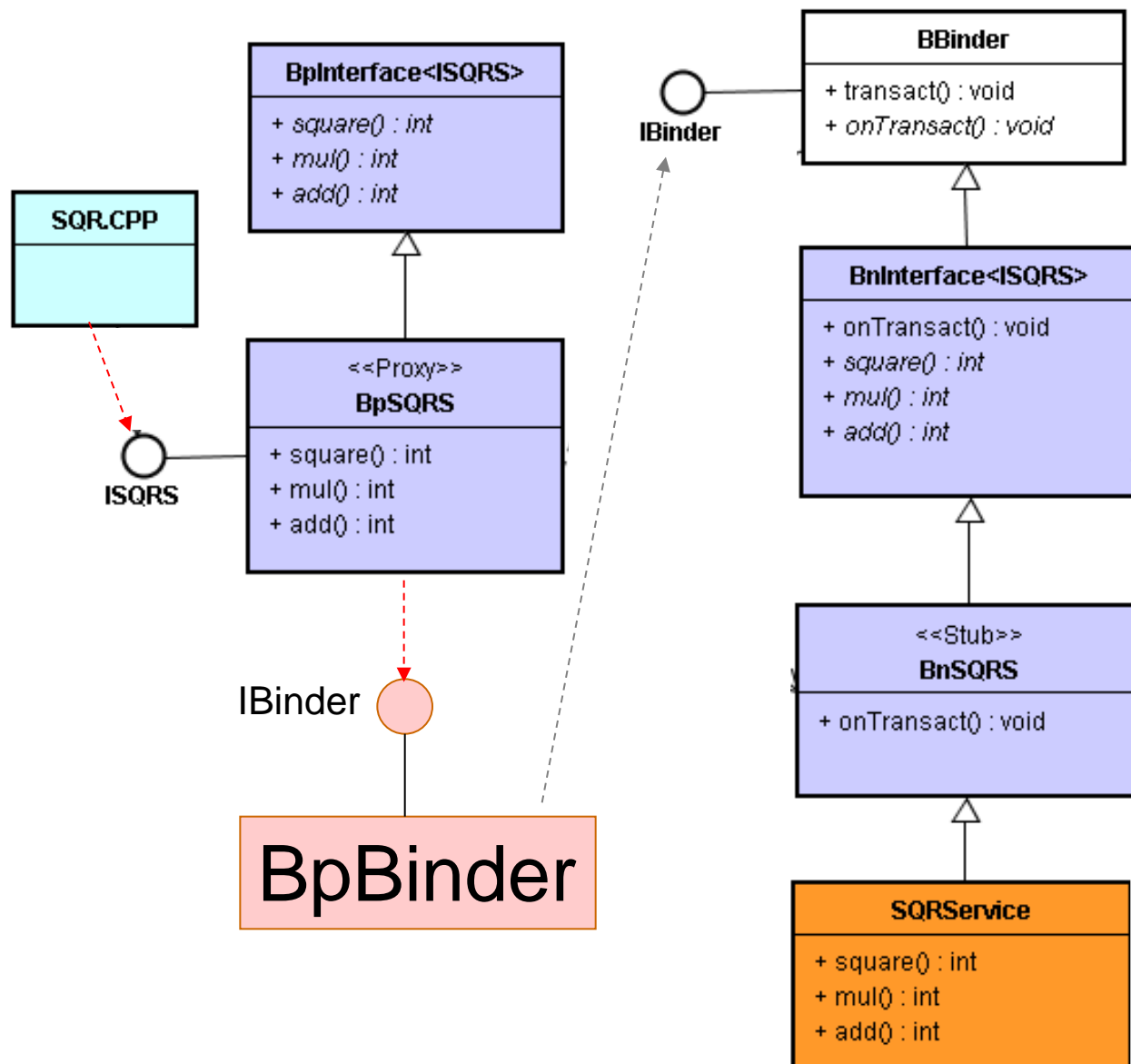
```

- 此Client模块执行时，首先由execute()函数调用getSQRService()函数。
- 这要求SM协助绑定SQRService核心服务。绑定了，SM就会在Client进程里诞生一个分身：BpBinder对象。接着，SM就将BpBinder的IBinder接口，回传给SQR类。

```
sp<IBinder> ib = sm-  
    >getService(String16("misoo.sqr"));
```

```
sp<ISQRS> sService =  
    interface_cast<ISQRS>(ib);
```



- 继续执行到指令：
`k = isqrs->square(n);`
- 直接使用**ISQR**接口了。





~ Continued ~