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_d

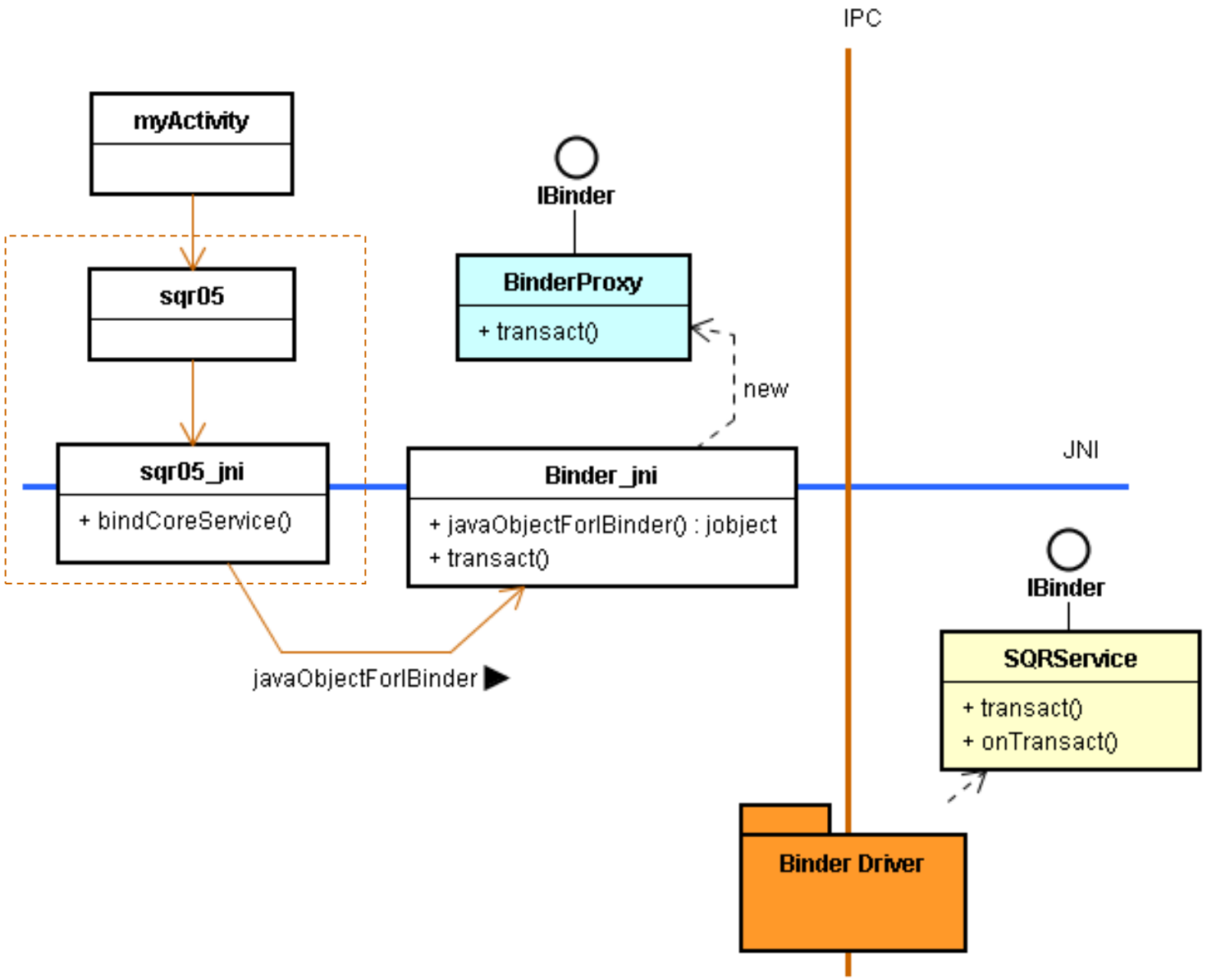
撰写你的第一个核心服务(d)

By 高煥堂

方案(二)的实现

- C层拥有**控制点**的必备表现是：
  1. **从C创建Java对象**
  2. **从C调用Java层函数**

- **步骤4.1:** 此时，myActivity必须透过JNI Native函数去绑定核心服务，然后由JNI Native函数在Java层诞生一个BpBinder对象的分身：即BinderProxy对象。如下图：


IPC
myActivity
sqr05
sqr05_jni
+ bindCoreService()
IBinder
BinderProxy
+ transact()
new
Binder_jni
+ javaObjectForIBinder() : jobject
+ transact()
JNI
javaObjectForIBinder
IBinder
SQRService
+ transact()
+ onTransact()
Binder Driver

架构设计：sqr05的角色

- myActivity不宜含有native函数，于是委托sqr05来提供native函数，创建Java对象，然后回传接口给myActivity。

使用javaObjectForIBinder()函数

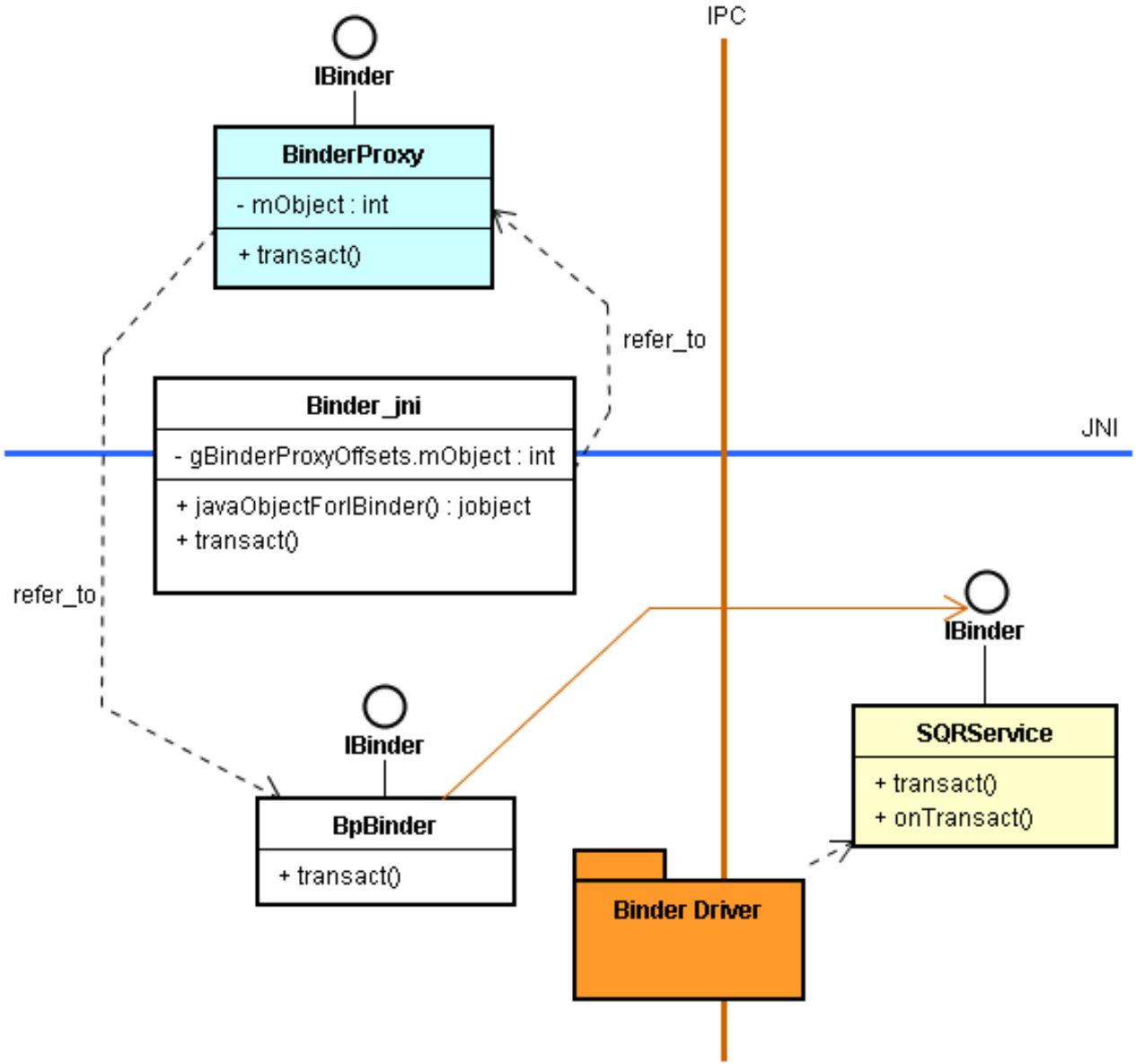
- 在Android里，提供了一个JNI Native 模块，内涵一个javaObjectForIBinder()函数，它能协助诞生Java层的BinderProxy对象，做为BpBinder对象的分身。

- 撰写JNI Native模块

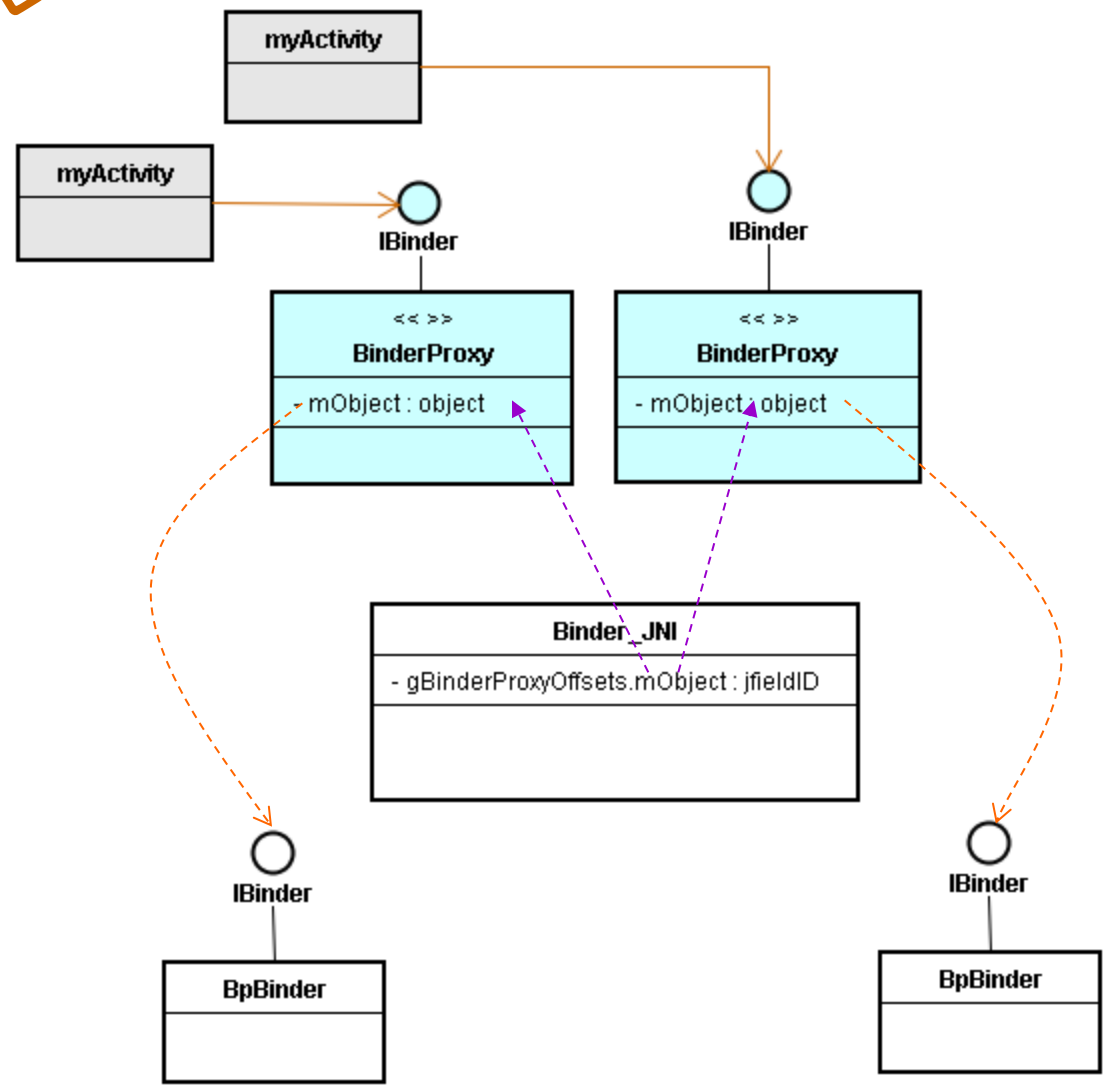
```
// sqr05.java
// .........
public class sqr05
{
        static
            { System.loadLibrary("SQRS05_jni"); }
        public native final IBinder bindCoreService();
}
```

```
/* com_misoo_service_sqr05.cpp */
//.....
sp<IBinder> m_ib;
JNIEXPORT jobject JNICALL
Java_com_misoo_service_sqr05_bindCoreService(JNIEnv *env, jobject
thiz){
    LOGE("bindCoreService");
    sp<IServiceManager> sm = defaultServiceManager();
    m_ib = sm->getService(String16("misoo.sqr"));
    LOGE("SM:getService %p\n",sm.get());
    if (m_ib == 0){
          LOGW("SQRService not published, waiting...");   return 0;
        }
    jobject jbi = javaObjectForIBinder(env, m_ib);
    if (jbi == 0)     {    LOGE("javaObjectForIBinder jbi = 0");   return 0; }
    return jbi;
}
```

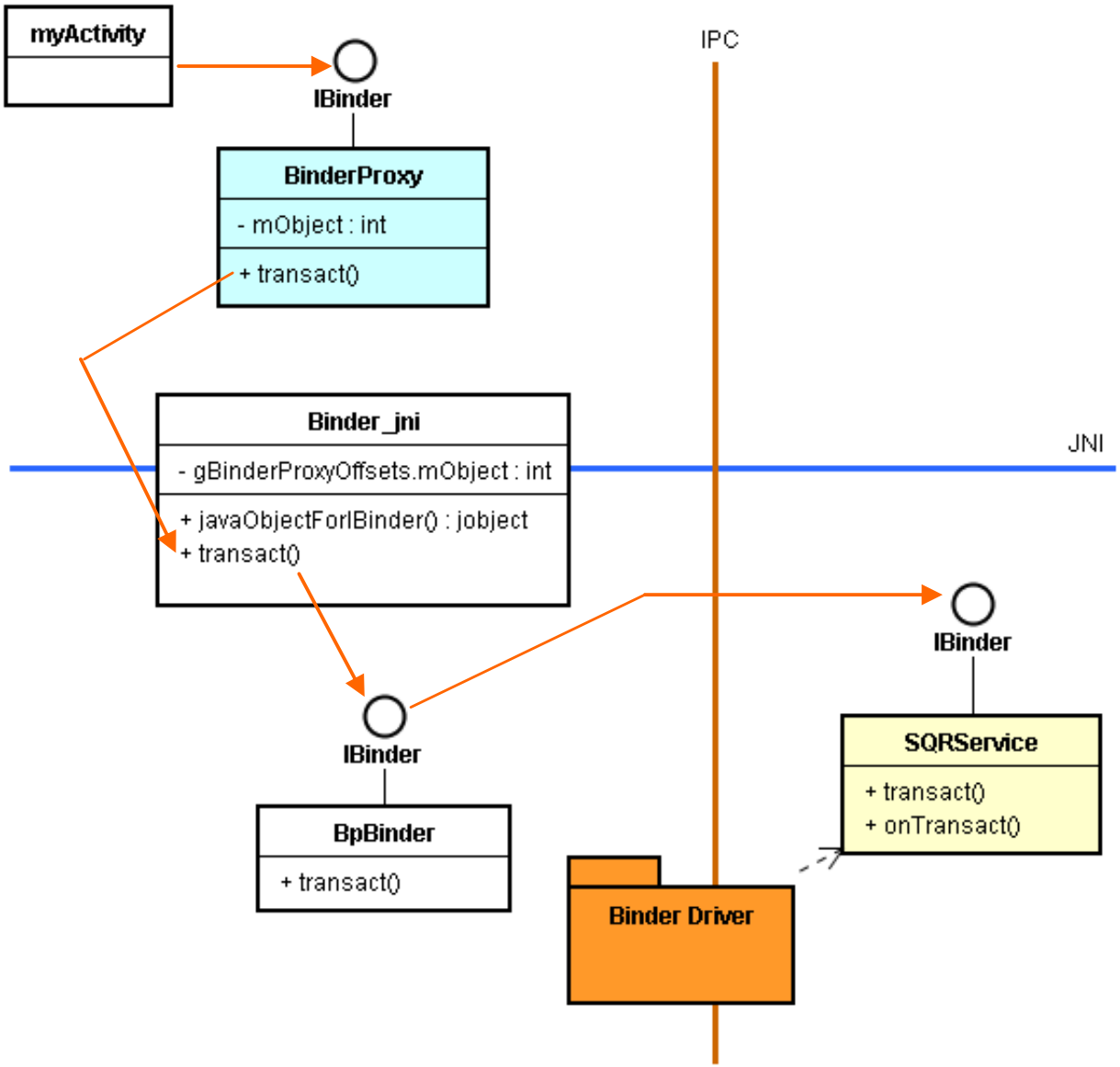
- 由javaObjectForIBinder()诞生Java层的BinderProxy对象。
- BinderProxy对象里的mObject属性指向BpBinder的IBinder接口。
- JNI Native模块里的gBinderProxyOffsets.mObject公用属性存有BinderProxy类别里的mObject属性的ID值(不是指针)。如下图：


IPC
IBinder
BinderProxy
- mObject : int
+ transact()
refer_to
Binder_jni
- gBinderProxyOffsets.mObject : int
+ javaObjectForIBinder() : jobject
+ transact()
JNI
refer_to
IBinder
SQRService
+ transact()
+ onTransact()
IBinder
BpBinder
+ transact()
Binder Driver

设计用意：<动态对动态>


myActivity
myActivity
IBinder
IBinder
<< >>
BinderProxy
- mObject : object
<< >>
BinderProxy
- mObject : object
Binder_JNI
- gBinderProxyOffsets.mObject : jfieldID
IBinder
IBinder
BpBinder
BpBinder

- 于是，在从myActivity类别里，就能透过BinderProxy对象而调用JNI Native模块，转而远距调用SQRService核心服务了。如下图：


myActivity
IBinder
IPC
BinderProxy
- mObject : int
+ transact()
Binder_jni
- gBinderProxyOffsets.mObject : int
+ javaObjectForIBinder() : jobject
+ transact()
JNI
IBinder
IBinder
SQRService
+ transact()
+ onTransact()
BpBinder
+ transact()
Binder Driver

- 撰写myActivity

```
// myActivity.java
// ……..ndroid.os.IBinder;
public class myActivity extends Activity implements OnClickListener {
      private Button btn, btn2;
      @Override
      //……
      }
 public void onClick(View v) {
      switch(v.getId()){
```

```
 case 101:
     sqr05 sqr = new sqr05();
     IBinder m_ib = sqr.bindCoreService();
     int code = 0;
     Parcel data = Parcel.obtain();
     data.writeInt(11);
     Parcel reply = Parcel.obtain();
      int flags = 0;
      try {
          m_ib.transact(code, data, reply, flags);
      } catch (Exception e) {  e.printStackTrace();     }
      setTitle("Value = " + String.valueOf(reply.readInt()));
      break;
 case 102:   finish();    break;
    }
}}
```

- myActivity执行到指令：

```
sqr05 sqr = new sqr05();
IBinder m_ib = sqr.bindCoreService();
```

- 就委托sqr05来调用Native的bindCoreService()函数：

```
Java_com_misoo_service_sqr05_bindCoreService(
        JNIEnv*, jobject)
```

- 这bindCoreService()函数内含指令：

```
sp<IServiceManager> sm = defaultServiceManager();
m_ib = sm->getService(String16("misoo.sqr"));
```

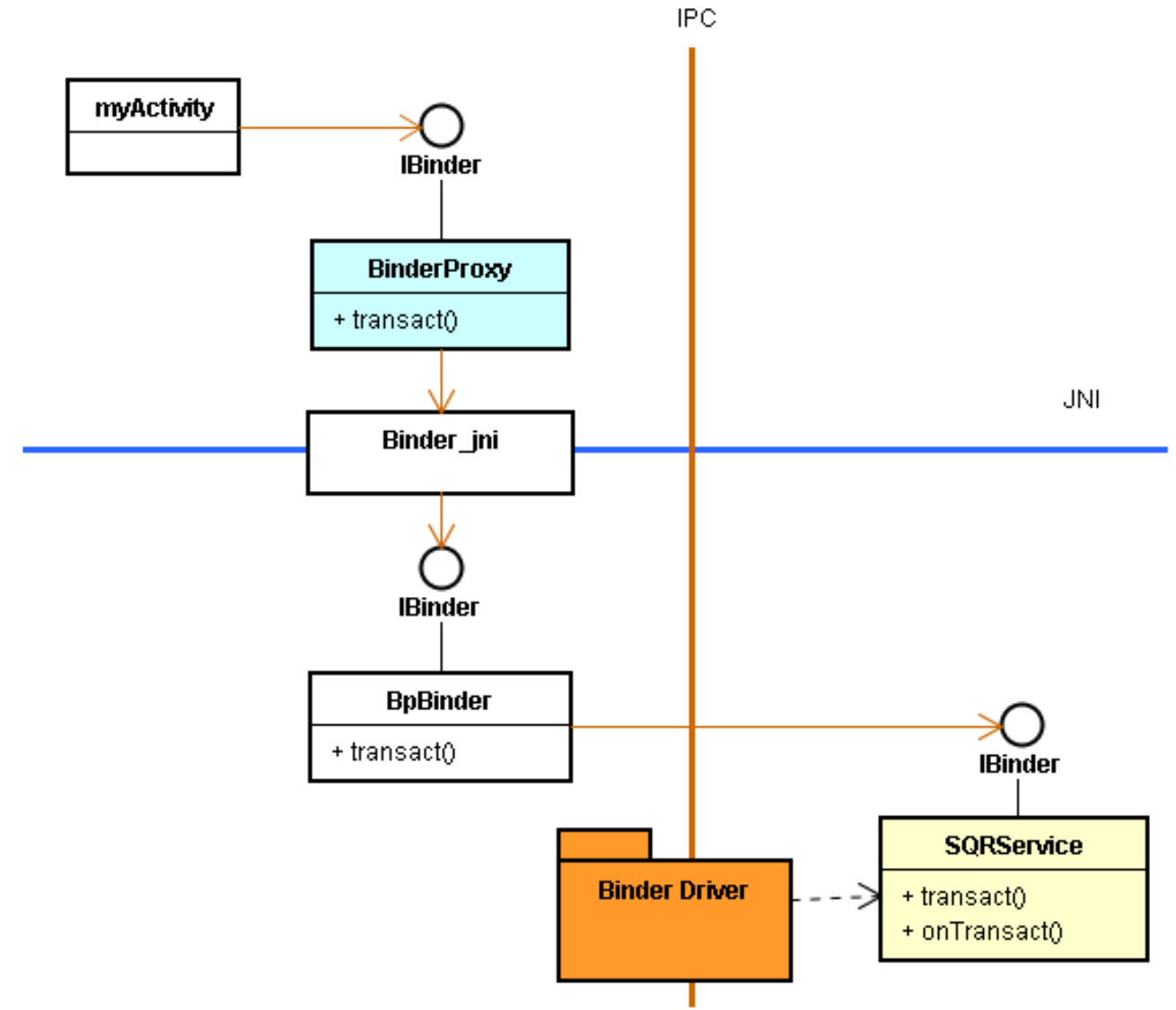
- 于是，bindCoreService()函数绑定了SQRService核心服务。

- 再执行指令：

  jobject jbi = javaObjectForIBinder(env, m_ib);

- 这bindCoreService()函数就诞生了Java层的BinderProxy对象。
- 最后执行到指令： *return jbi;*

- 这bindCoreService()函数就将BinderProxy对象的IBinder接口回传给myActivity类别。
- 于是，myActivity类别可以透过此IBinder接口而调用BinderProxy的transact()函数，转而调用SQRService核心服务的onTransact()函数。
- 实现了下图的结构了


IPC
myActivity
IBinder
BinderProxy
+ transact()
JNI
Binder_jni
IBinder
BpBinder
+ transact()
IBinder
SQRService
+ transact()
+ onTransact()
Binder Driver

~ Continued ~