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

D05_b

Android Service的Proxy-Stub设计模式(b)

By 高煥堂

3、实践Proxy-Stub模式

基于AIDL架构

- 基于AIDL架构，可以包装这个IBinder接口，来提供更好用的接口(如ISampeService)。例如，定义接口如下：

服务开发者定义接口

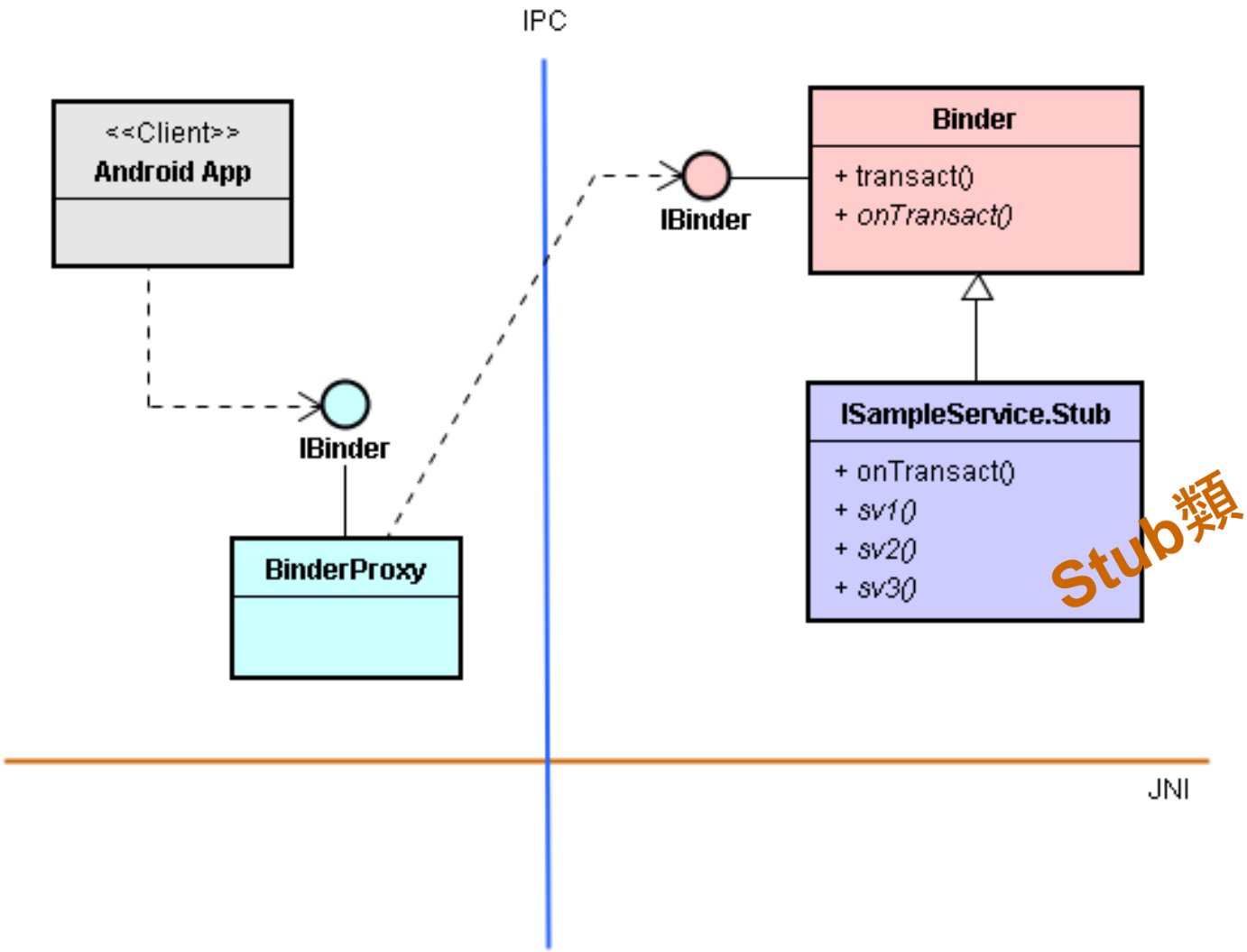
```
/*
* aidl file : frameworks/base/core/java/android/os/ISampleService.aidl
* This file contains definitions of functions which are exposed by service
*/
package android.os;
interface ISampleService {
        void sv1();
        void sv2();
        void sv3();
}
```

- 此时，会使用aidl.exe工具产生ISampleService.Stub类。
- 它一方面继承了Binder框架基类，得到IBinder接口。
- 同时。也继承了ISampleService接口所定义的sv1(), sv2()和sv3()函数。其定义如下：

AIDL工具生成Stub类和Proxy类

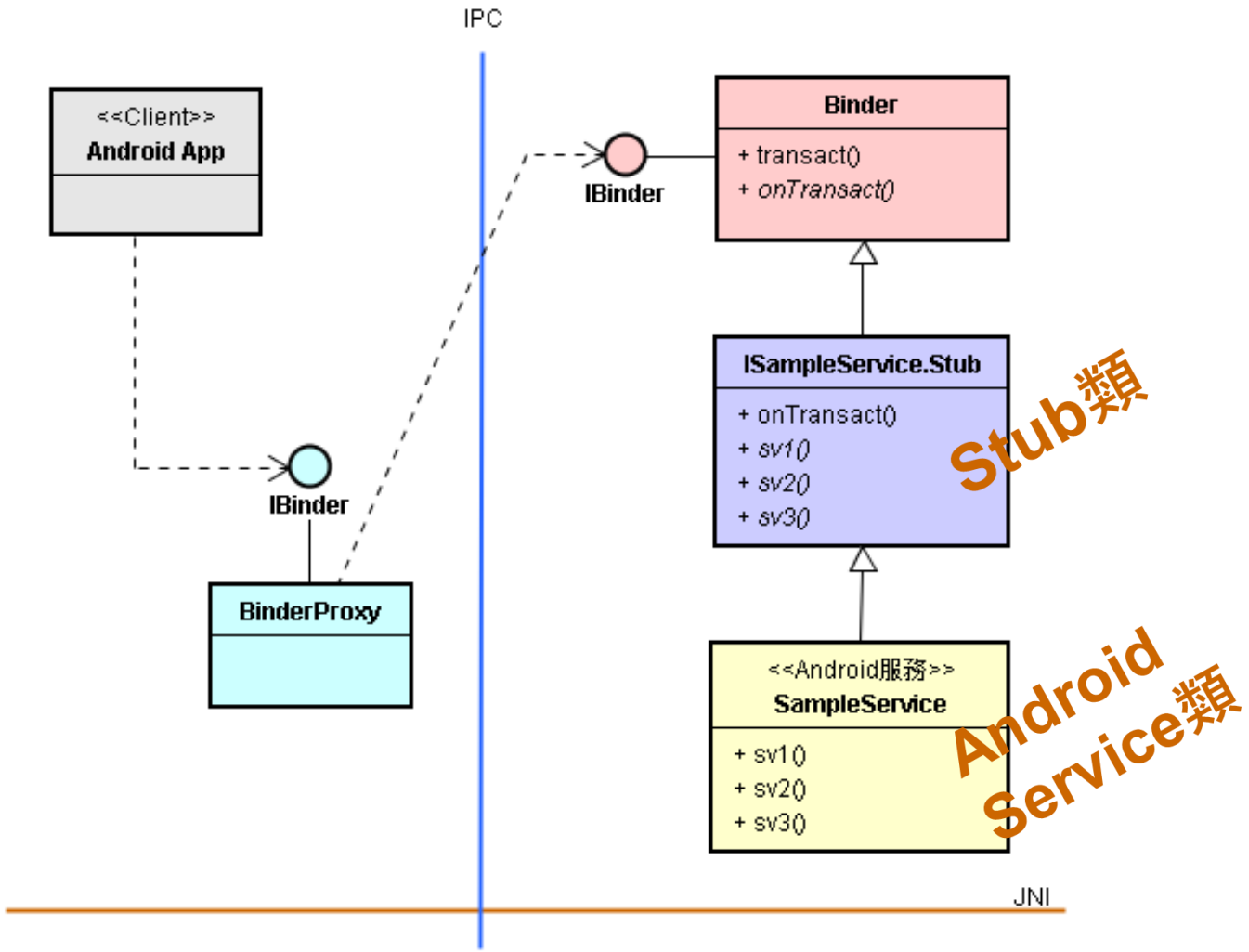
```
class ISampleService.Stub extends Binder
        implements ISampleService {
            //.....
    }
```

- 如下图所示：


IPC
<<Client>>
Android App
IBinder
Binder
+ transact()
+ onTransact()
IBinder
ISampleService.Stub
+ onTransact()
+ sv1()
+ sv2()
+ sv3()
Stub類
BinderProxy
JNI

- 这个ISampleService.Stub类别是由Android SDK所提供的aidl.exe工具所自动产生的。
- 我们只要撰写SampleService类别，它来继承上述的ISampleService.Stub类别即可，如下图所示：

从Stub类衍生出Android Service


IPC
<<Client>>
Android App
IBinder
Binder
+ transact()
+ onTransact()
ISampleService.Stub
+ onTransact()
+ sv1()
+ sv2()
+ sv3()
Stub類
IBinder
BinderProxy
<<Android服務>>
SampleService
+ sv1()
+ sv2()
+ sv3()
Android
Service類
JNI

```
/* SampleService.java */
// .....
public class SampleService extends ISampleService.Stub {
    private static final String TAG = “SampleService";
    private SubThread t;
    private SubHandler h;
    private Context mContext;

public SampleService(Context context) {
        super();
        mContext = context;
        t = new SubThread();
        t.start();
   }
```

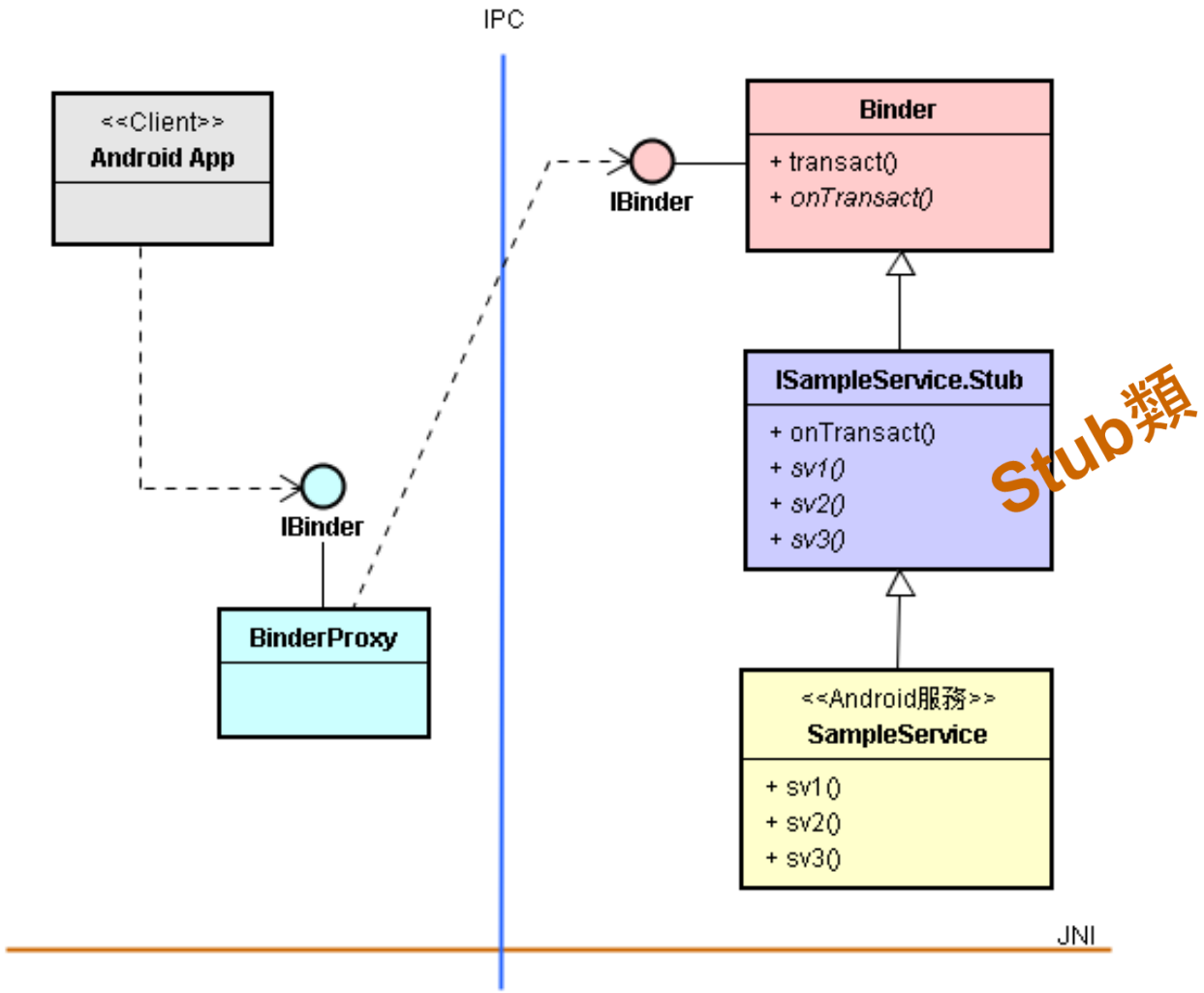
```
 public void  sv1() {
     Message msg = Message.obtain();
     msg.what = 0;
     msg.arg1 = “sv1”;
     h.sendMessage(msg);
 }
public void  sv2() {
     Message msg = Message.obtain();
     msg.what = 0;
     msg.arg1 = “sv2”;
     h.sendMessage(msg);
 }
public void  sv3() {
     Message msg = Message.obtain();
     msg.what = 0;
     msg.arg1 = “sv3”;
     h.sendMessage(msg);
 }
```

```
 private class SubThread extends Thread {
        public void run() {
            Looper.prepare();
            h = new SubHandler();
            Looper.loop();
        }}
private class SubHandler extends Handler {
          @Override
        public void handleMessage(Message msg) {
            try {
                if (msg.what == 0) {
                    // 顯示出 msg.arg1 內容;
                }
            } catch (Exception e) {
                Log.e(TAG, "Exception...", e);
            }
        }
}}
```

- 将Android Service登录到SystemServer，并从Init进程启动这项系统服务。

Register service in SystemServer.java

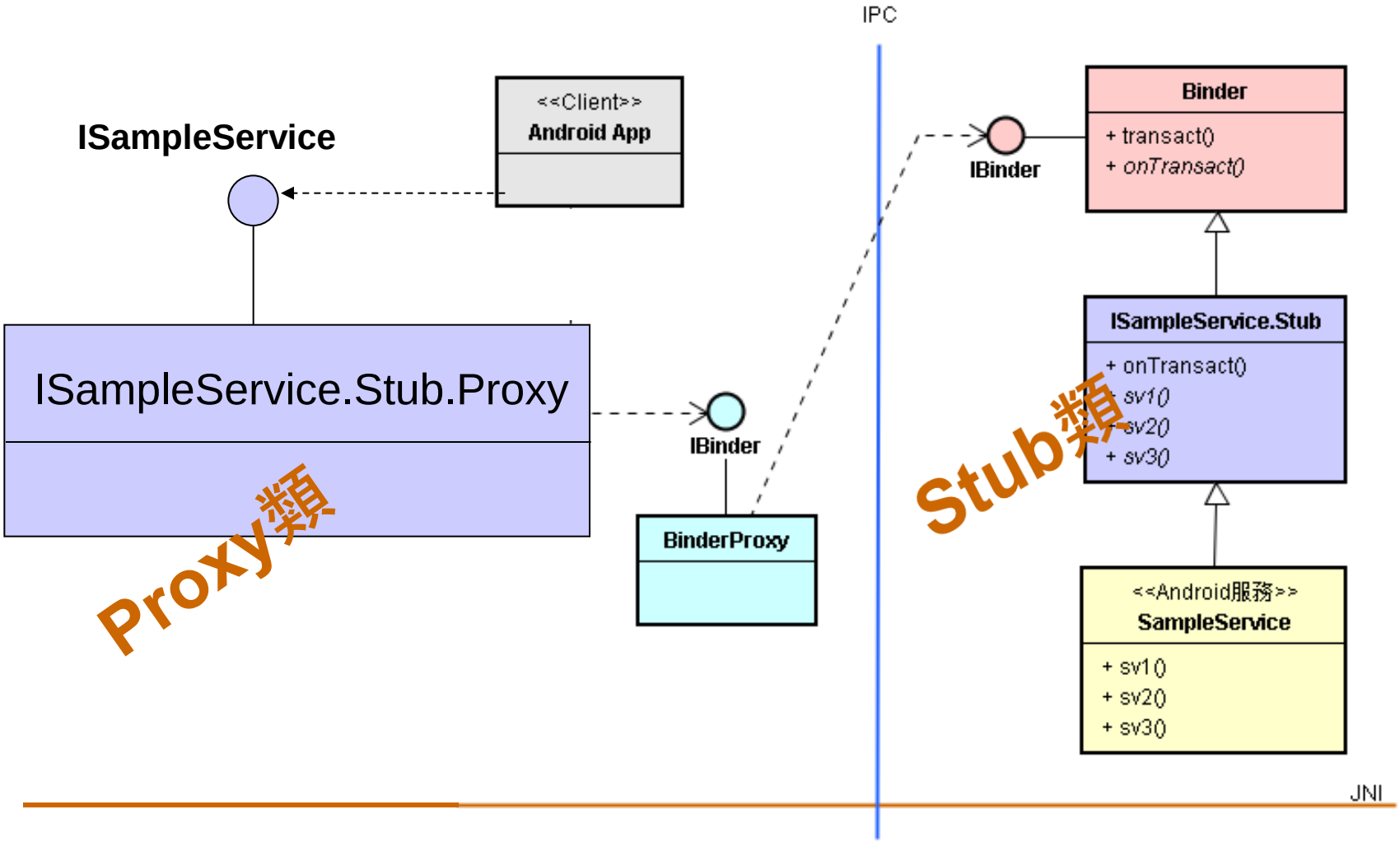
```
/*
 * go to function "@Override public void run()"
 * .....
 * Add following block after line "if (factoryTest !=
SystemServer.FACTORY_TEST_LOW_LEVEL) {"
 */

try {
    Slog.i(TAG, "Test Service");
    ServiceManager.addService(  “CS001”  ,
                        new SampleService(context));
} catch (Throwable e) {
    Slog.e(TAG, "Failure starting SampleService", e);
}
```


IPC
<<Client>>
Android App
IBinder
Binder
+ transact()
+ onTransact()
IBinder
ISampleService.Stub
+ onTransact()
+ sv1()
+ sv2()
+ sv3()
Stub類
BinderProxy
<<Android服務>>
SampleService
+ sv1()
+ sv2()
+ sv3()
JNI

在Client端创建Proxy类的对象

```
/*
 * myActivity.java
 */
// .....
public class myActivity extends Activity {
    private static final String DTAG = “SampleServer";
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
```

```
IBinder m_ib  =  ServiceManager.getService("CS001");
ISampleService m_obj  =  ISampleService.Stub.asInterface( m_ib );
        try {
            Log.d(DTAG, "Going to call service");
            m_obj.f1(255);
            m_obj.f2(17.35);
        }
        catch (Exception e) {
            Log.d(DTAG, "FAILED to call service");
            e.printStackTrace();
        }
    }
}
```


IPC
ISampleService
<<Client>>
Android App
IBinder
Binder
+ transact()
+ onTransact()
ISampleService.Stub.Proxy
IBinder
ISampleService.Stub
+ onTransact()
+ sv1()
+ sv2()
+ sv3()
Proxy類
Stub類
BinderProxy
<<Android服務>>
SampleService
+ sv1()
+ sv2()
+ sv3()
JNI

~ Continued ~