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

B05_c

IPC的Proxy-Stub设计模式(c)

By 高焕堂

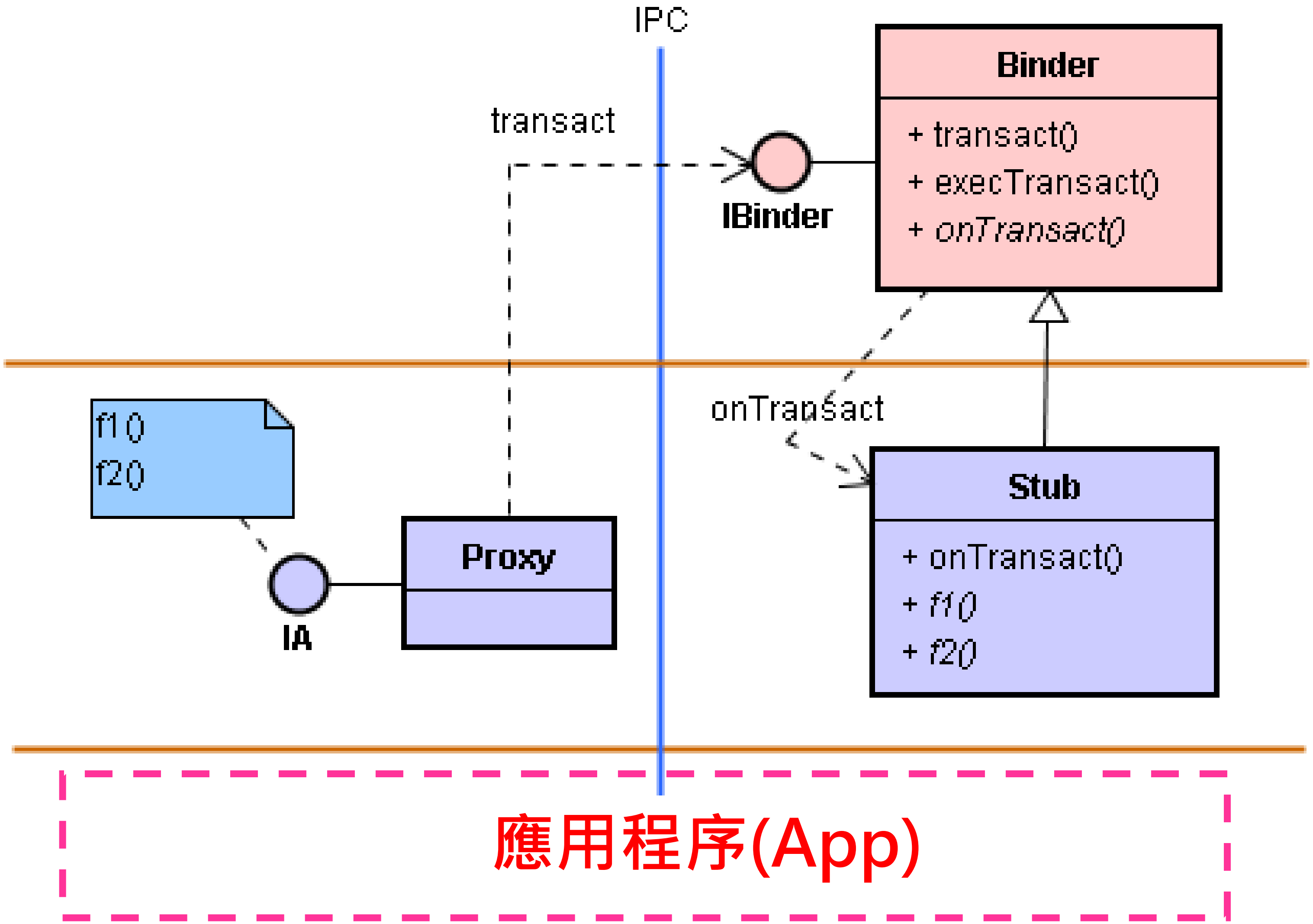
- 如何写Proxy及Stub类
  →包装IBinder接口

- 谁来写Proxy及Stub类
  天子、曹操、地头蛇?

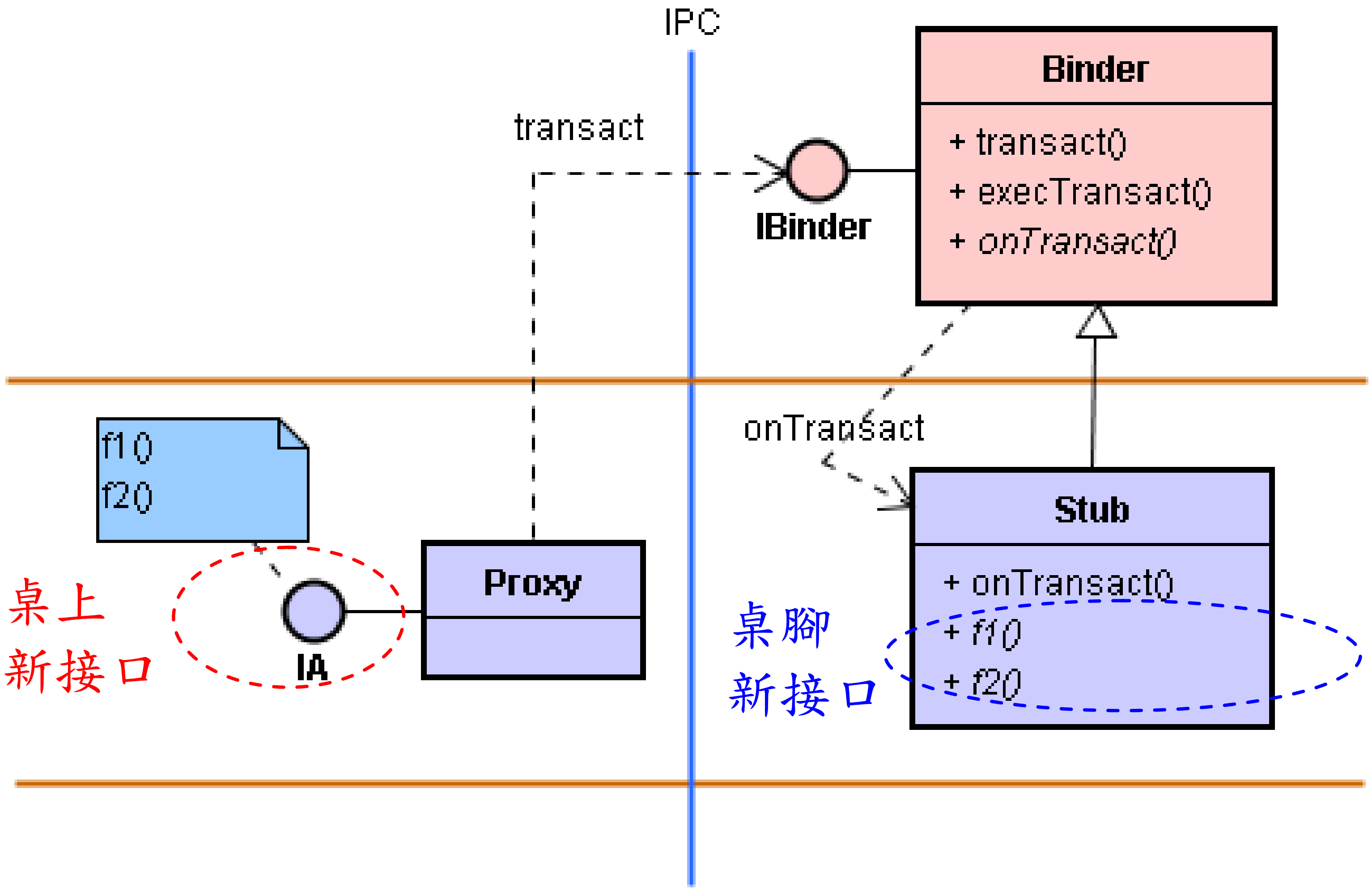
3、包裝IBinder接口

-- 使用Proxy-Stub设计模式

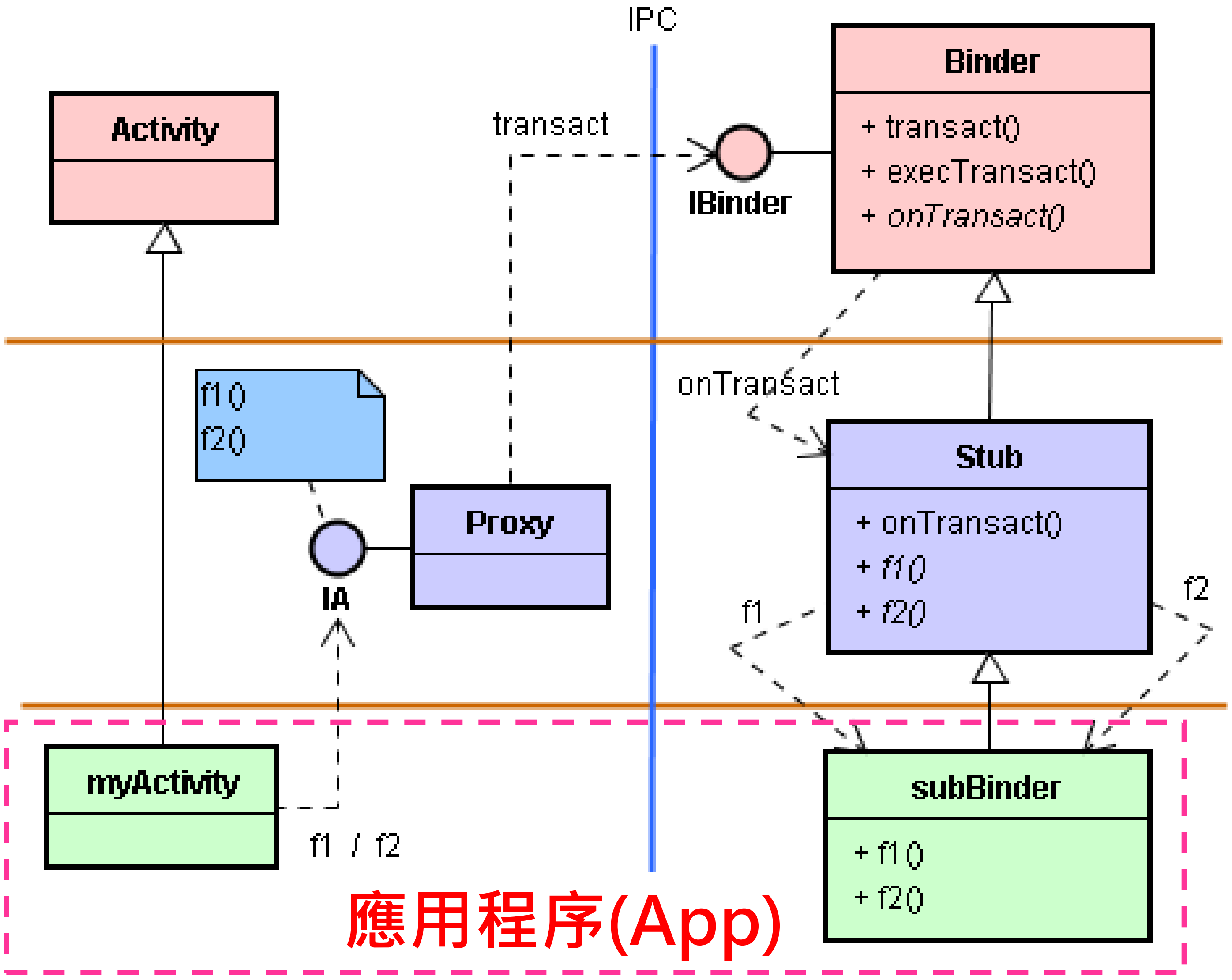
- 采用Proxy-Stub设计模式将IBinder接口包装起来，让App与IBinder接口不再产生高度相依性。


IPC
transact
IBinder
Binder
+ transact()
+ execTransact()
+ onTransact()
onTransact
f1()
f2()
IA
Proxy
Stub
+ onTransact()
+ f1()
+ f2()
應用程序(App)

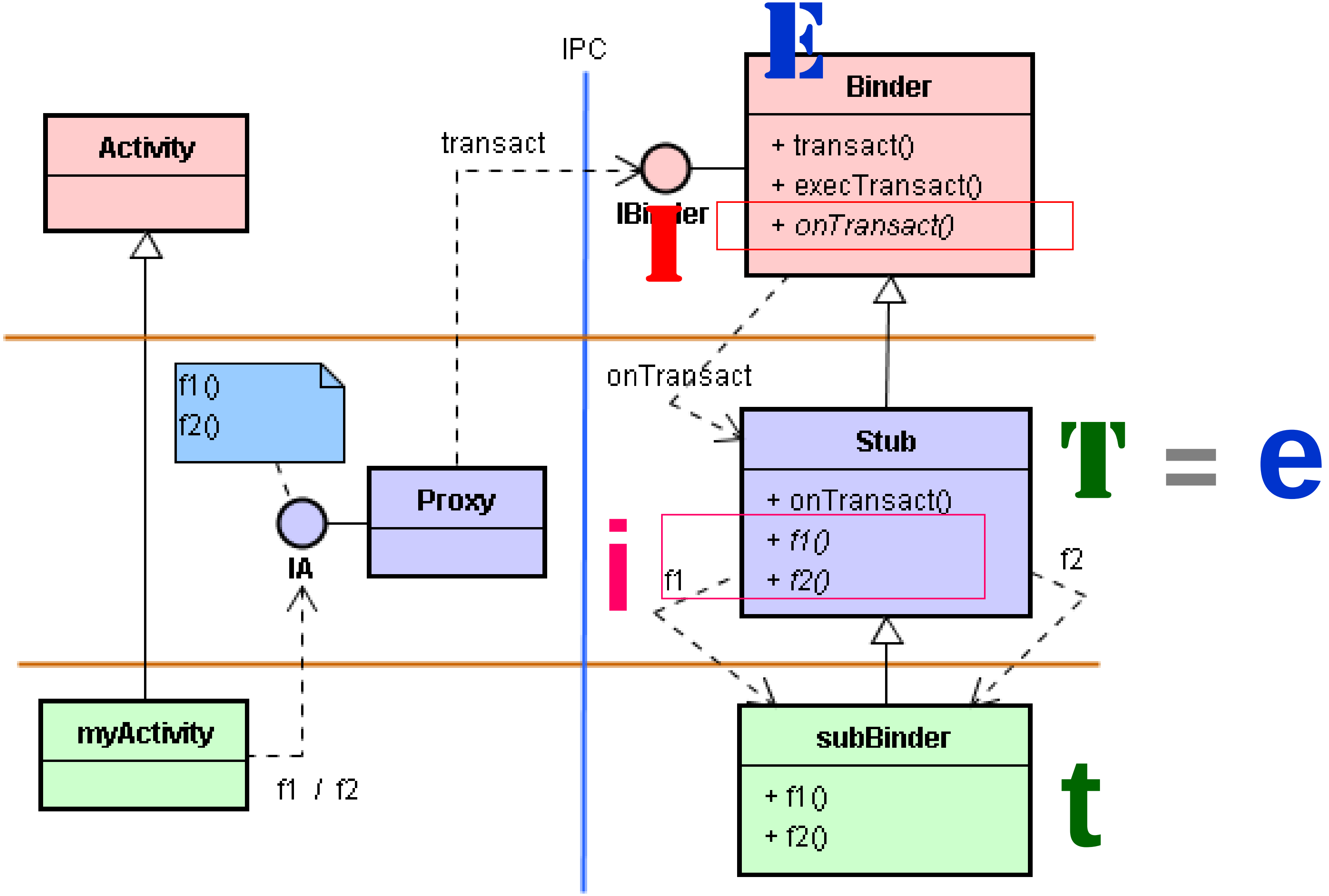
- 其将IBinder接口包装起来，转换出更好用的新接口：



- Proxy类提供较好用的IA接口给Client使用。
- Stub类别则是屏蔽了Binder基类的onTransact()函数，然后将IA接口里的f1()和f2()函数定义为抽象函数。于是简化了App开发的负担：


IPC
Activity
transact
IBinder
Binder
+ transact()
+ execTransact()
+ onTransact()
onTransact
f1 ()
f2()
Proxy
IA
Stub
+ onTransact()
+ f1 ()
+ f2()
f1
f2
myActivity
f1 / f2
subBinder
+ f1 ()
+ f2()
應用程序(App)

EIT造形的雙層組合！！

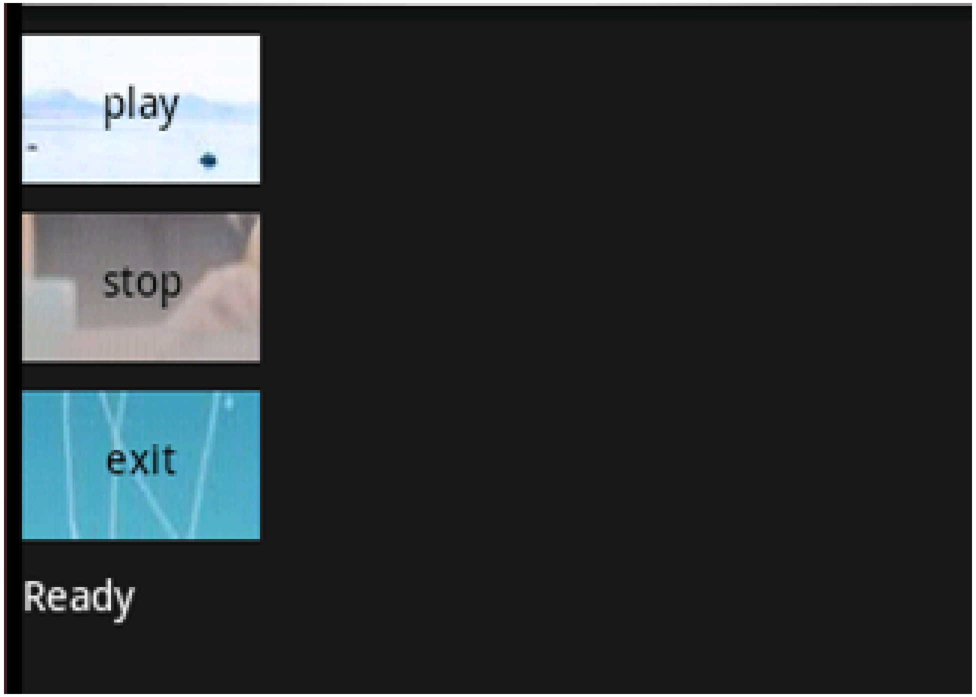

E
IPC
Binder
+ transact()
+ execTransact()
+ onTransact()
Activity
transact
IBinder
I
onTransact
Stub
+ onTransact()
+ f1()
+ f2()
T = e
f1()
f2()
Proxy
IA
i
f1
f2
myActivity
f1 / f2
subBinder
+ f1()
+ f2()
t

4、谁来写Proxy及Stub类呢?

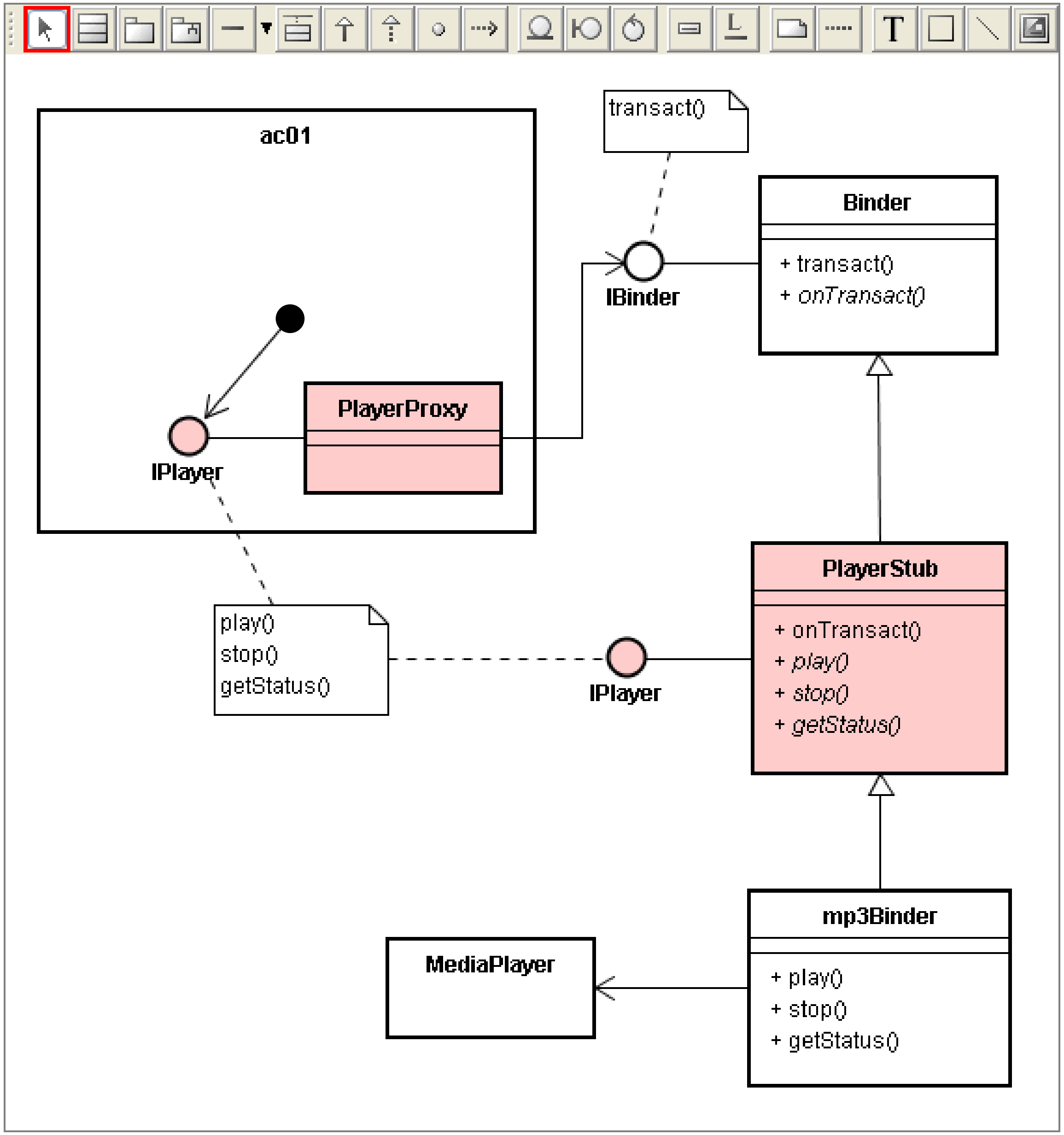
-- 地头蛇(App开发者)自己写

范例

- 兹写一个App范例，程序执行时出现画面如下：



- 在这个范例里，定义了一个IPlayer接口，然后规划了PlayerProxy和PlayerStub两的类，如下图：


ac01
IPlayer
PlayerProxy
transact()
IBinder
Binder
+ transact()
+ onTransact()
PlayerStub
+ onTransact()
+ play()
+ stop()
+ getStatus()
play()
stop()
getStatus()
IPlayer
MediaPlayer
mp3Binder
+ play()
+ stop()
+ getStatus()

曹操（小）框架的代码

定义一个新接口：IPlayer

```
// IPlayer.java
package com.misoo.pkgx;
public interface IPlayer {
            void play();
            void stop();
            String getStatus();
}
```

撰写一个Stub类：PlayerStub

```
// PlayerStub.java
package com.misoo.pkgx;
import android.os.Binder;
import android.os.Parcel;
public abstract class PlayerStub extends Binder implements IPlayer{
    @Override public boolean onTransact(int code, Parcel data,
             Parcel reply, int flags) throws android.os.RemoteException   {
                     reply.writeString(data.readString()+ " mp3");
                     if(code == 1) this.play();
                     else if(code == 2) this.stop();
                     return true;
           }
           public abstract void play();
           public abstract void stop();
           public abstract String getStatus();
}
```

撰写一个Proxy类：PlayerProxy

```
// PlayProxy.java
private class PlayerProxy implements IPlayer{
    private IBinder ib;
    private String mStatus;

    PlayerProxy(IBinder ibinder)
              {  ib = ibinder;   }
    public void play(){
              Parcel data = Parcel.obtain();
              Parcel reply = Parcel.obtain();
              data.writeString("playing");
              try {  ib.transact(1, data, reply, 0);
                     mStatus = reply.readString();
              } catch (Exception e) {   e.printStackTrace();  }
    }
    public void stop(){
          Parcel data = Parcel.obtain();
          Parcel reply = Parcel.obtain();
          data.writeString("stop");
          try {   ib.transact(2, data, reply, 0);
                  mStatus = reply.readString();
          } catch (Exception e) {  e.printStackTrace();           }
        }
     public String getStatus() {  return mStatus;  }
  }
}
```

App的代码

撰写mp3Binder类

```
// mp3Binder.java
// ……..
public class mp3Binder extends PlayerStub{
    private MediaPlayer mPlayer = null;
    private Context ctx;
    public mp3Binder(Context cx){   ctx= cx;       }
    public void play(){
        if(mPlayer != null) return;
        mPlayer = MediaPlayer.create(ctx, R.raw.test_cbr);
        try {  mPlayer.start();
        } catch (Exception e)  {
            Log.e("StartPlay", "error: " + e.getMessage(), e);  }}
    public void stop(){
        if (mPlayer != null)
            {   mPlayer.stop();   mPlayer.release();  mPlayer = null;  }}
    public String getStatus() {   return null;   }
}
```

撰写mp3RemoteService类

```
// mp3RemoteService.java
package com.misoo.pkgx;
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
public class mp3RemoteService extends Service {
          private IBinder mBinder = null;
          @Override       public void onCreate() {
                    mBinder = new mp3Binder(getApplicationContext());
          }
          @Override
public IBinder onBind(Intent intent) {      return mBinder;  }
}
```

撰写mp3RemoteService类

```
// ac01.java
// ………
public class ac01 extends Activity implements OnClickListener {
    //……….
    private PlayerProxy pProxy = null;
    public void onCreate(Bundle icicle) {
       // ………
       startService(new Intent("com.misoo.pkgx.REMOTE_SERVICE"));
       bindService(new Intent("com.misoo.pkgx.REMOTE_SERVICE"),
                      mConnection, Context.BIND_AUTO_CREATE);  }
    private ServiceConnection mConnection =
                      new ServiceConnection() {
        public void onServiceConnected(ComponentName className,
                       IBinder ibinder)
              {   pProxy = new PlayerProxy(ibinder);   }
        public void onServiceDisconnected(ComponentName classNa){}
     };
    public void onClick(View v) {
      switch (v.getId()) {
         case 101:  pProxy.play();  tv.setText(pProxy.getStatus());
                    break;
         case 102:  pProxy.stop();  tv.setText(pProxy.getStatus());
                    break;
         case 103:
             unbindService(mConnection);
             stopService(
                    new Intent("com.misoo.pkgx.REMOTE_SERVICE"));
             finish();  break;
        }
     }
}
```

- PlayerStub类将onTransact()函数隐藏起来，提供一个更具有美感、更亲切的新接口给mp3Binder类使用。
- 隐藏了onTransact()函数之后，mp3Binder类的开发者就不必费心去了解onTransact()函数了。于是，PlayerProxy与PlayerStub两个类遥遥相对，并且将IPC细节知识(例如transact()和onTransact()函数之参数等)包夹起来。

~ Continued ~