

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>

A10\_d

# 介绍Android的 Java层应用框架(d)

By 高煥堂

## 7、四大组件之相互启动关系

- 在Android框架里已经定义了这些组件间之互动关系。互动关系又分为二：

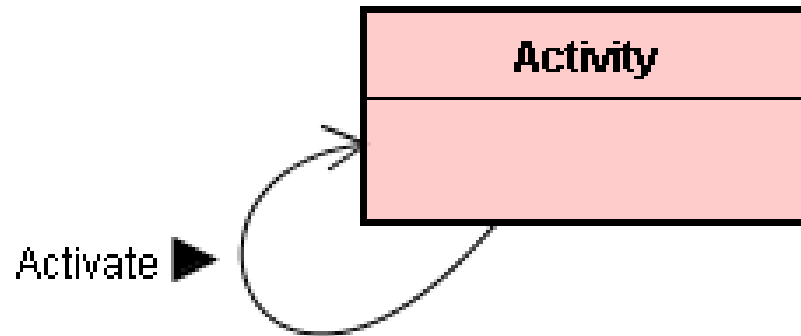
1. 启动(Activation)关系。

2. 数据传输或交换(Data Exchange)关系。

- 于此，从启动关系来初步认识它们。由于CP(Content Provider)组件大多基于被动角色、属于被启动的一方，其角色单纯不复杂。因之，先针对Activity  $\leftrightarrow$  Receiver  $\leftrightarrow$  Service间之三角关系，做个仔细の説明。

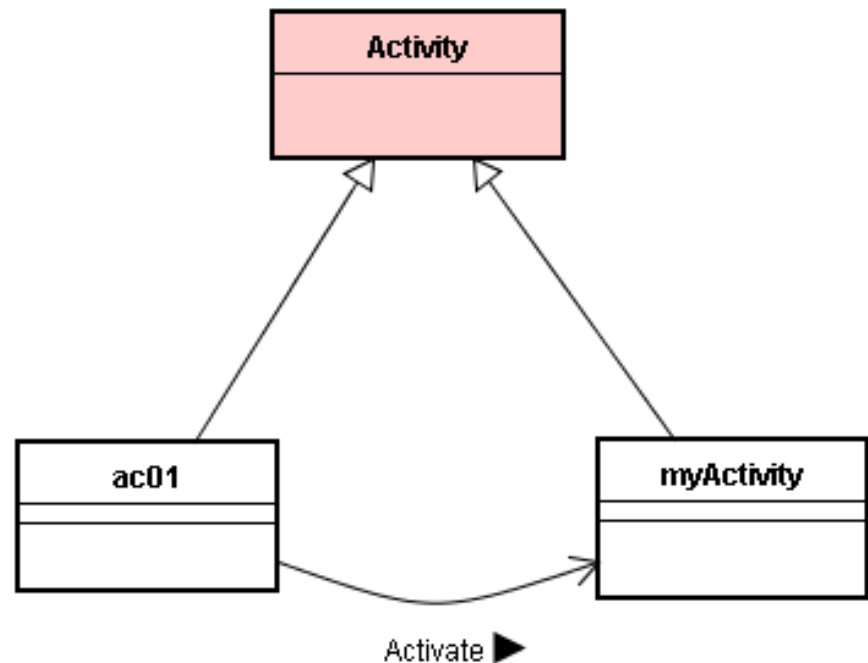
## ☆ 1. AA模式

这是由Android先启动一个Activity的子类(如ac01)之对象，然后ac01对象视需要而要求Android启动另一个Activity的子类(如myActivity)的对象。



```
public class ac01 extends Activity {  
    // .....  
    startActivity(new Intent(this, myActivity.class));  
    // .....  
}
```

**\*\* Activity Manager**  
启动ac01



- ac01就要求Android(里的ActivityManager)去启动myActivity子类别之对象了。
- 当myActivity被启动时，立即执行myActivity的onCreate()函数。

**\*\* Activity Manager**

如何选择ac01与myActivity呢？



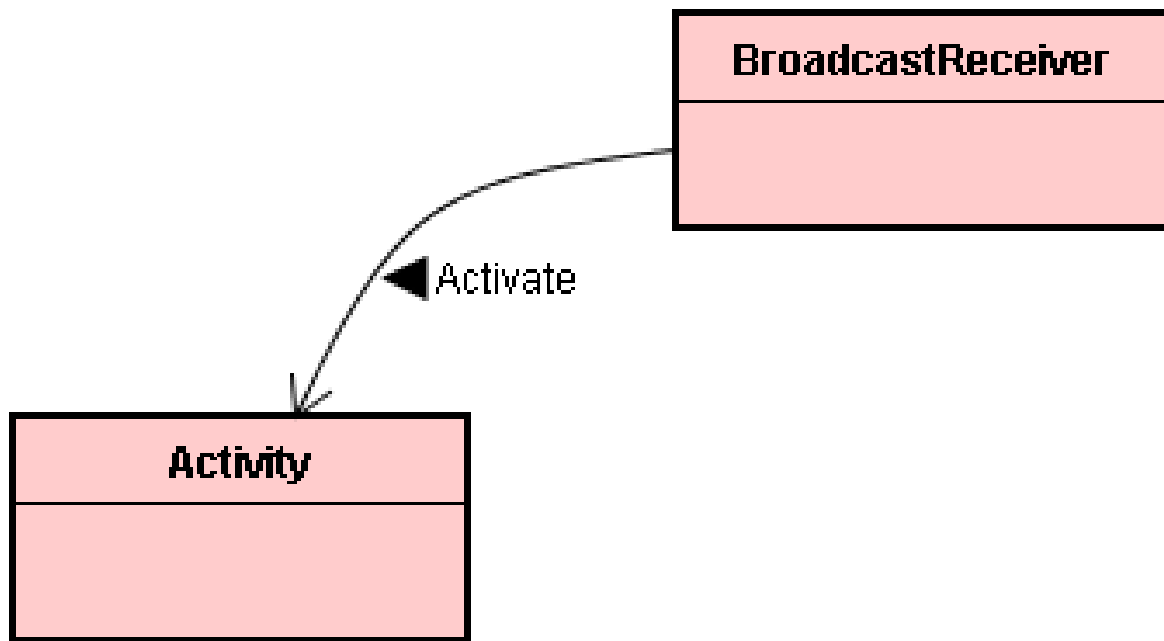
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.misoo.kx01">
    <application android:icon="@drawable/icon">
        <activity android:name=".ac01" android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".myActivity" android:label="myActivity">
        </activity>
    </application>
</manifest>
```

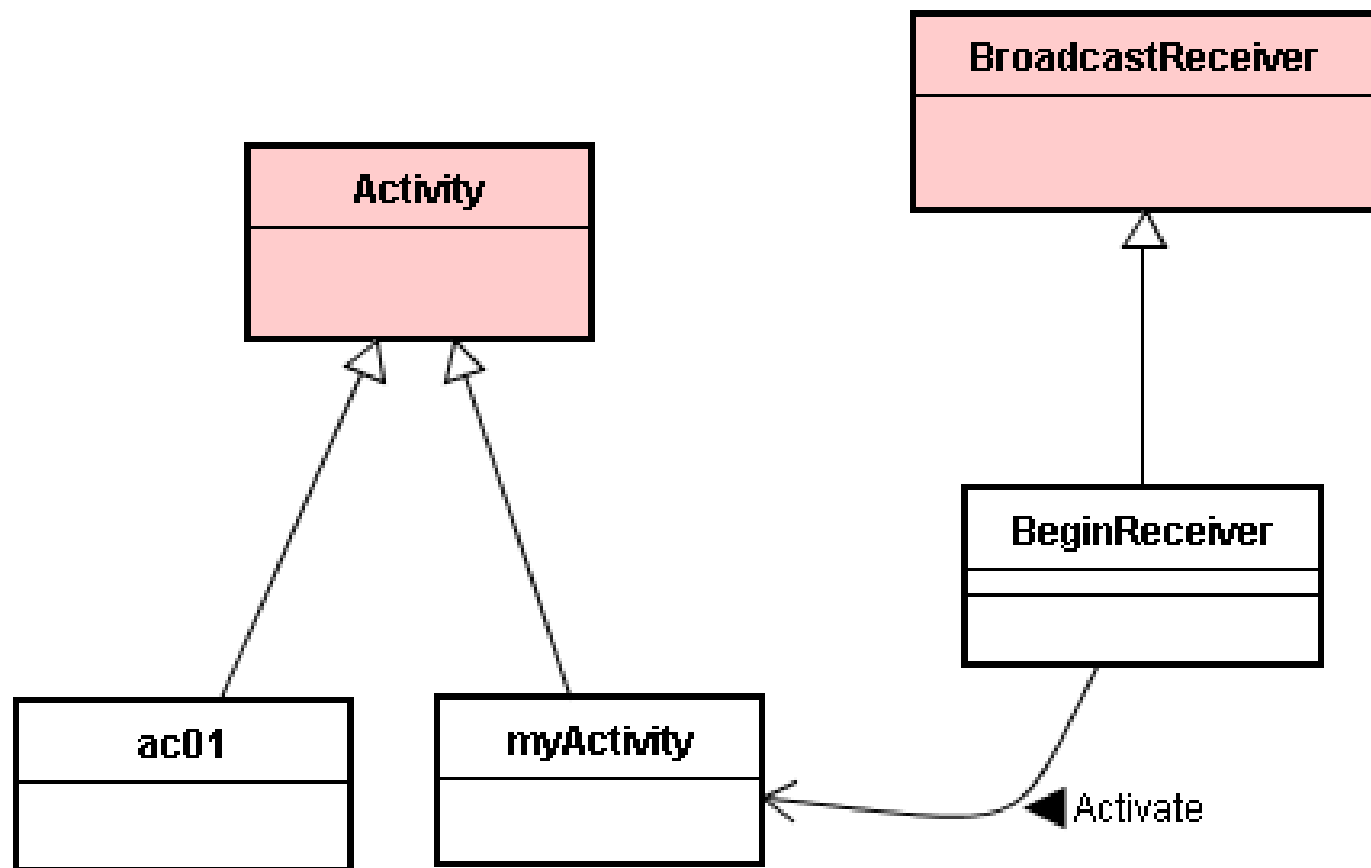
留意：

AndroidManifest.xml文檔的用途：**给予各子类的配对条件**

## ☆ 2. RA模式

在此模式里，是由R(BroadcastReceiver)启动A(Activity)。但是谁来启动R呢？在本范例里，将由开机的事件来启动R。





在AndroidManifest.xml裡宣告如下：

```
<application android:icon="@drawable/icon">  
    // .....  
    <receiver android:name=".BeginReceiver">  
        <intent-filter>  
            <action android:name=  
                "android.intent.action.BOOT_COMPLETED" />  
        </intent-filter>  
    </receiver>  
</application>
```

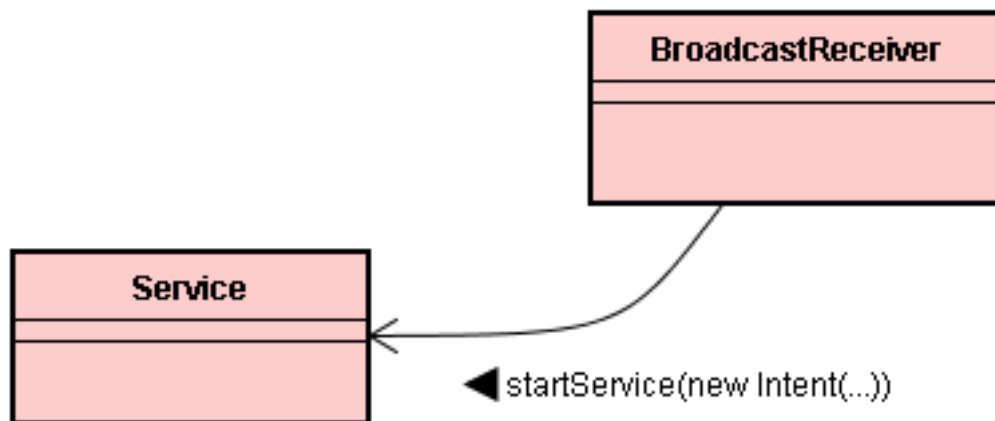
- 重开模拟器或手机，开机了，此时会发送开机事件给myReceiver。Android就诞生一个BeginReceiver对象，并执行onReceiver()函数：

```
public void onReceive(Context context, Intent intent){  
    Intent i = new Intent(context, myActivity.class);  
    i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
    context.startActivity(i);  
}
```

- 这就发出一个Intent对象给Android，请求启动myActivity。

### ☆ 3. RS模式

在此模式里，是由R(BroadcastReceiver)启动S(Service)。



在AndroidManifest.xml裡宣告如下：

```
<activity android:name=".myActivity"
           android:label="myActivity"> </activity>
<receiver android:name=".BeginReceiver">
    <intent-filter>
        <action android:name=
            "android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
<service android:name=".myService" />
```

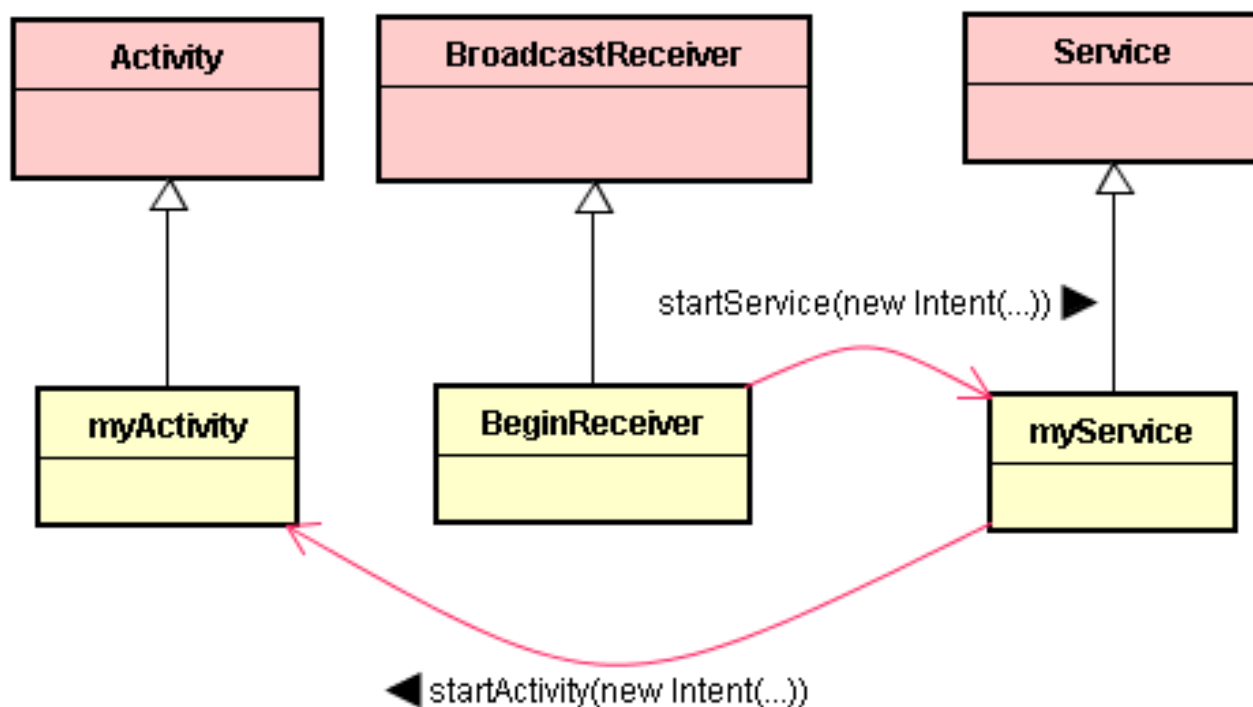
```
public class BeginReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent intent)  
    {  
        // .....  
        context.startService(  
            new Intent(context, myService.class));  
    }  
}
```



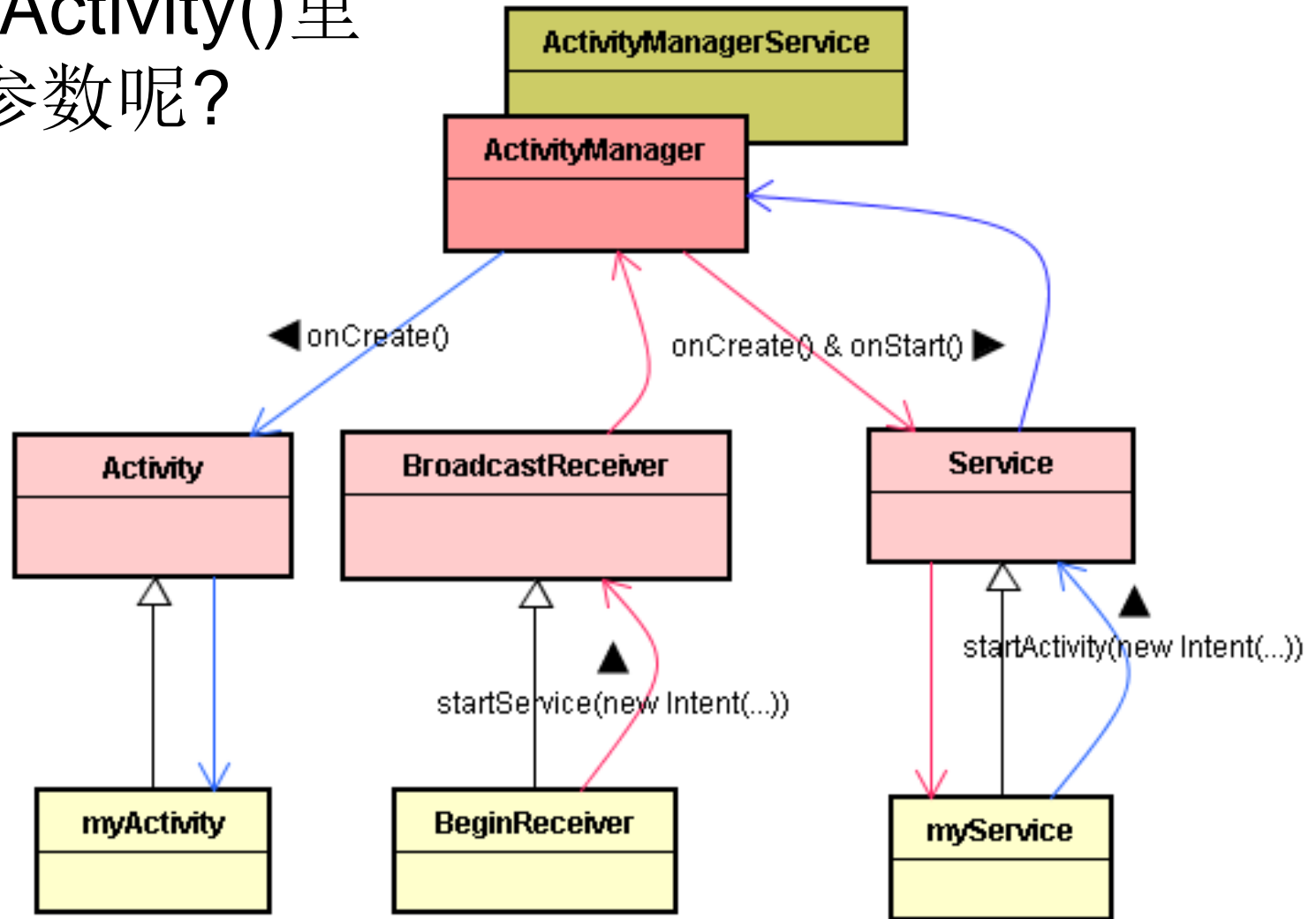
```
public class myService extends Service
{
    private NotificationManager mNMgr;
    @Override public void onCreate(){
        mNMgr =
            (NotificationManager)getSystemService(
                NOTIFICATION_SERVICE);
        showNotification(...);
    }
    @Override public void onStart() { //..... }
}
```

## ☆ 4. RSA模式

在此模式里，是由R(BroadcastReceiver)启动S(Service)，然后S再启动A。



请问：myService里如何  
撰写startActivity()里  
的Intent参数呢？



## AndroidManifest.xml

```
// .....  
<service name=".myService">  
    <intent-filter>  
        <action android:name=  
            "com.misoo.utilities.LocalService"/>  
    </intent-filter>  
</service>
```

```
class BeginReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent  
intent)  
    {  
        // .....  
        context.startService( new Intent(  
            "com.misoo.utilities.LocalService, ...));  
    }  
}
```

## ☆ 5. AR模式(由A啟動R)

### AndroidManifest.xml

```
// .....  
<receiver android:name=".myReceiver">  
    <intent-filter>  
        <action android:name=  
            "com.misoo.kx01.ac01.MY_EVENT" />  
    </intent-filter>  
</receiver>
```

```
public class ac01 extends Activity {  
    private final String MY_EVENT =  
        new String("com.misoo.kx01.ac01.MY_EVENT");  
    //.....  
    Intent in = new Intent(MY_EVENT);  
    this.sendBroadcast(in);  
    //.....  
}
```

## ☆ 其它模式演練

- 除了上述的常见启动模式之外，你还可以活用和变化之，例如：
- **AS模式**
- **SR模式**



~ Continued ~