

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_c

介绍Android的 Java层应用框架(c)

By 高煥堂

5、认识Android框架 的实践技术

4个嫡系基类

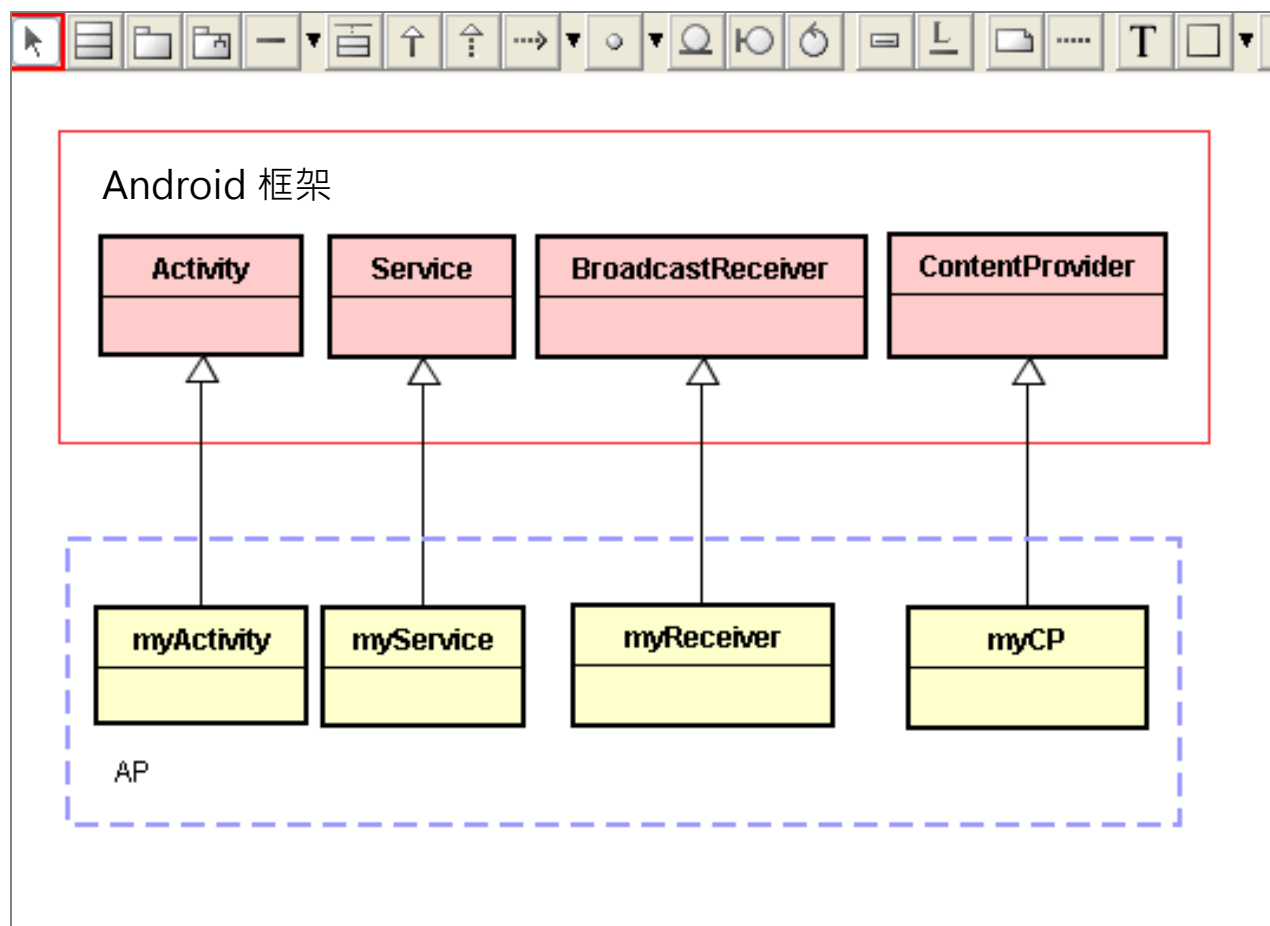
- Android框架里提供了4个嫡系的基类，包括：
 - **Activity**：处理UI互动的事情
 - **Service**：幕后服务(如播放背景音乐)
 - **BroadcastReceiver**：接收讯息及事件
 - **ContentProvider**：储存共享数据



Android 框架



- 基于这些基类，就可以写App子类，如下：



誰來創建子類的對象呢？

- 这4种应用子类都是由Android框架来负责创建(Create or New)其对象(Object)的。
不过有趣的是：
 - 撰写框架基类在先
 - 撰写应用子类在后

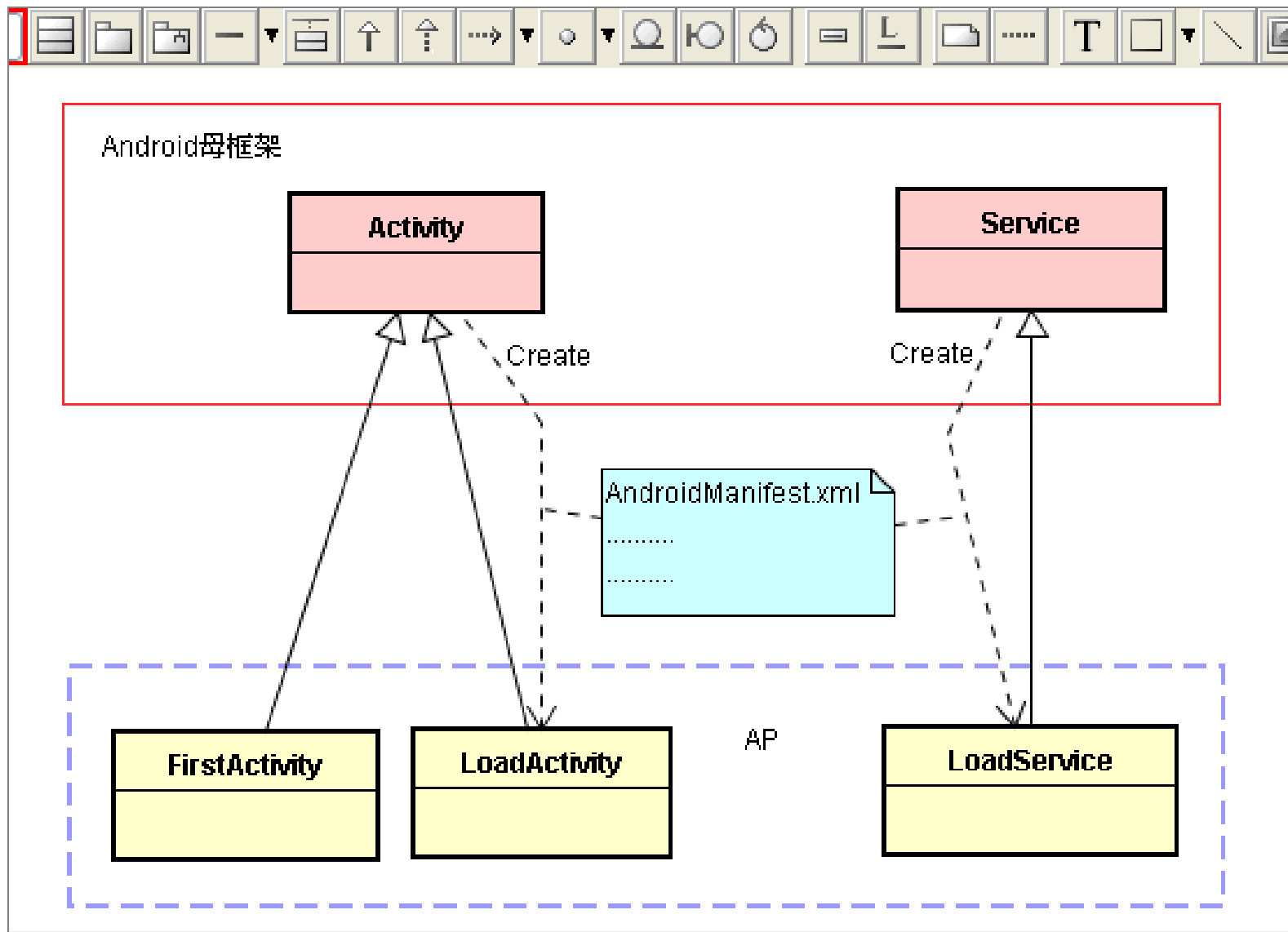
- 那么框架事先又如何知道开发者后来撰写的应用子类的名称呢？如果不知道应用子类的名称，又如何创建应用子类的对象呢？
- 答案是：依赖AndroidManifest.xml文档。
例如：

// *AndroidManifest.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.misoo.pkm">
    <uses-
        permissionxmlns:android="http://schemas.android.com/apk/res/andr
        oid"
            android:name="android.permission.INTERNET">
    </uses-permission>
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
    <activity android:name=" .FirstActivity"
        android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER"
        />
        </intent-filter>
```

```
<activity android:name= ".LoadActivity">
    <intent-filter>
        <category android:name="android.intent.category.DEFAULT"
    />
    </intent-filter>
</activity>
<service android:name= ".LoadService" android:process=":remote">
    <intent-filter>
        <action android:name="com.misoo.pkm.REMOTE_SERVICE"
    />
    </intent-filter>
</service>
</application>
</manifest>
```

- 在执行阶段(Run-time)，Android框架读取这个由开发者所写的XML文档。于是Android框架得知开发者撰写了3个嫡系应用子类，就会去创建这些应用子类的对象，如下图：



6、Intent-based Programming 技术

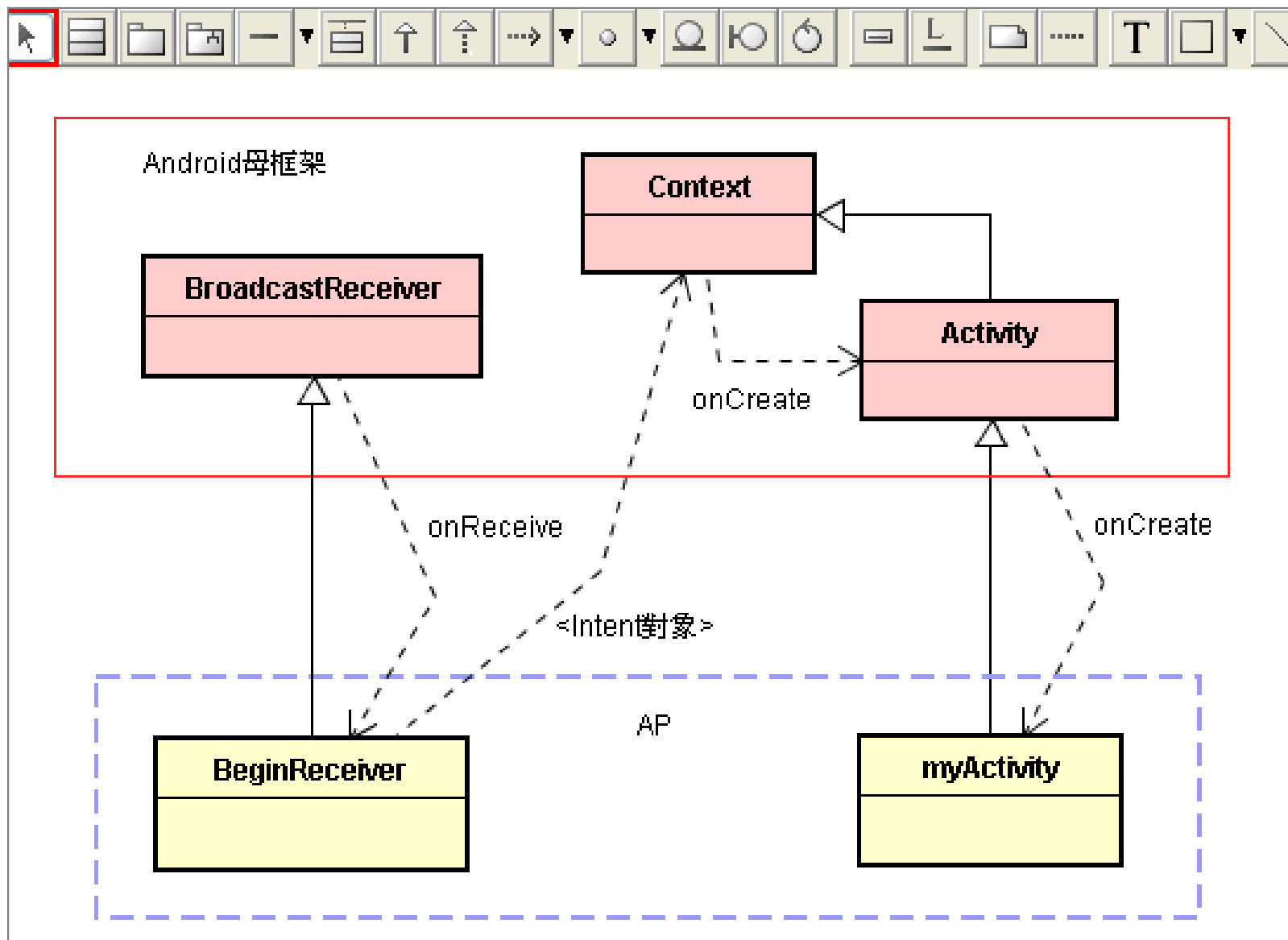
嫡系应用子类之间如何互相沟通呢？

- Android的4种嫡系应用子类之间如何互相沟通呢？其实很简单，就像你打手机去车行叫出租车，而不是直接到街道上叫车。你送给行一个短信或一通电话，表明你的「意图」(Intent)，当车行服务员接到此意图，就依据你的意图的内含条件而去挑选最合适的车，然后派遣它去接你。

- 于是，Android框架里所提供的Intent就扮演它们之间的沟通媒介。这4种嫡系对象都是由框架去配对及启动的，并不是组件之间透过直接呼叫而启动的。「意图」(Intent)本身是定义为一个类(Class)，一个Intent对象表达一个期望(Expectation)，叙述其所期望的服务或动作。

- 框架则根据此Intent对象之叙述，负责配对，找出相配的组件，然后将Intent对象传递给所找到的对象，于是框架的媒婆任务就完成了。
- 因此，Intent对象扮演着媒体中介的角色。例如，提供「Activity组件→Android框架→ContentProvider组件」之间互相沟通的相关信息。

- 这种Intent沟通方式让不同人开发的套件(Package)内之类别可以轻易达成沟通。例如，有两个套件：A与B。假设它们是由不同的开发者所撰写的，当这两个套件都被下载到同一支手里执行时，就能立即互相沟通了，这是Android框架的一项重要实践技术。
- 例如下图：



兹以Java代码实践如下：

```
// BeginReceiver.java
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class BeginReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        Intent intent = new Intent(context, myActivity.class);
        intent.addFlags( Intent.FLAG_ACTIVITY_NEW_TASK );
        context.startActivity( intent );
    }
}
```

- 其中的Intent.FLAG_ACTIVITY_NEW_TASK就是配对条件。
- 而代码：`context.startActivity(intent);` 就将Intent对象发送给母框架里的Context，要求配对到Activity的应用子类。



此圖摘自：<<Deep inside Android>> By Giillles Priintemps



~ Continued ~