

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>

B01_f

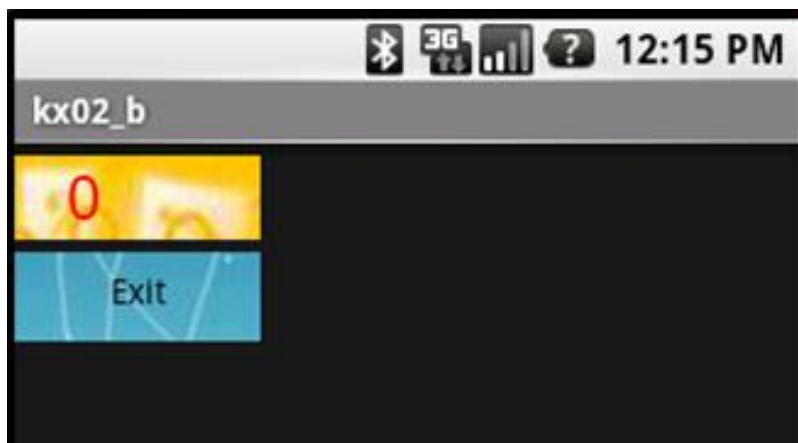
认识进程与IPC架构(f)

By 高煥堂

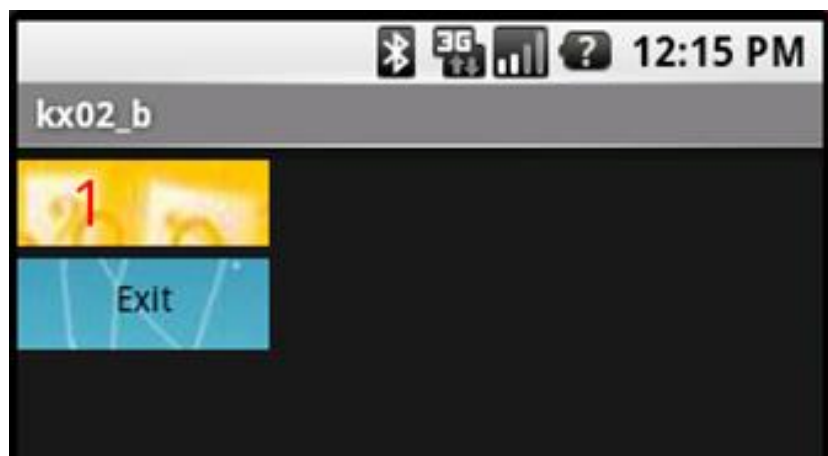
7、短程通信 vs. 远程通信

範例：短程通信

- 首先出现ac01画面，立即启动myService，定时连续传来数字，如下：



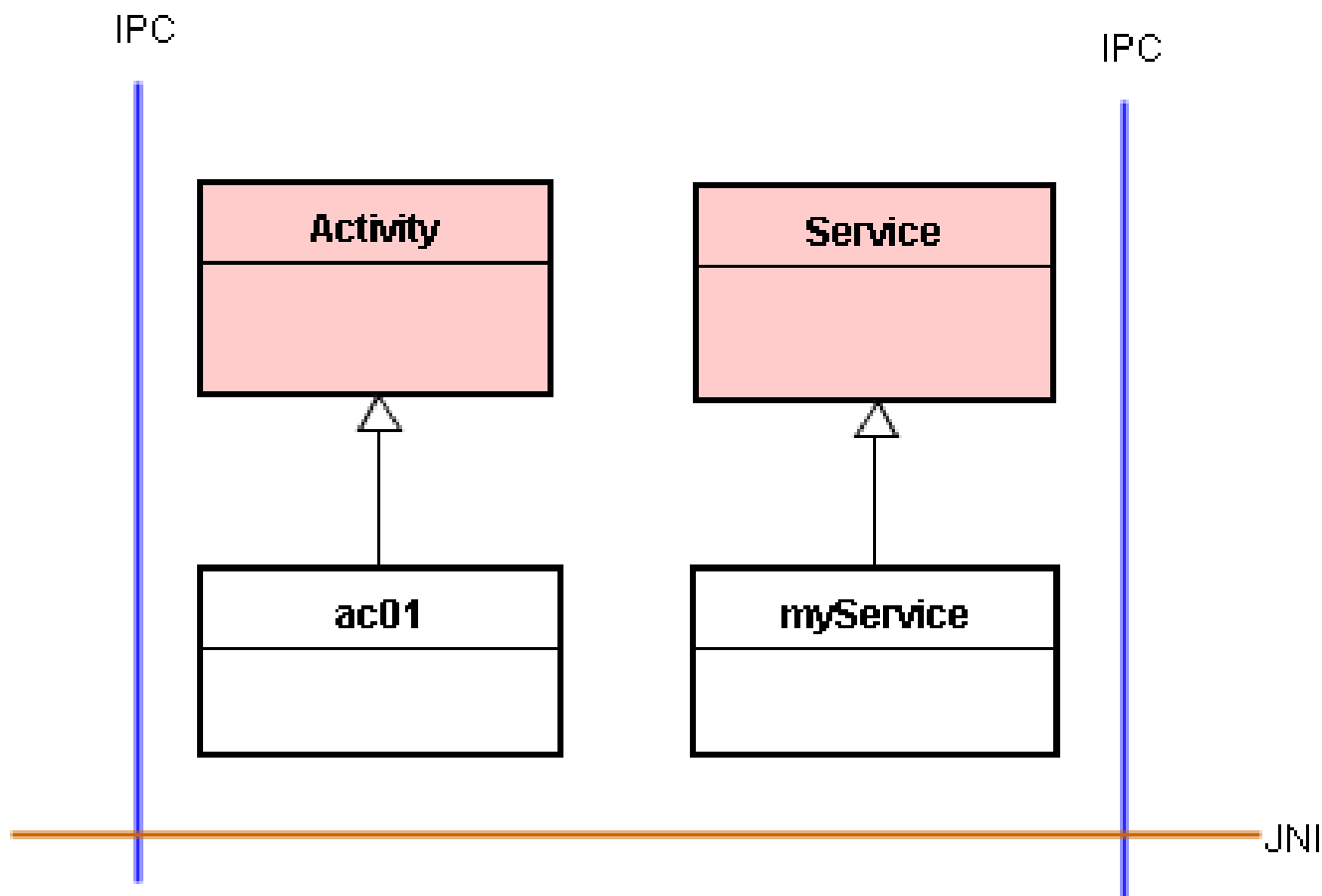
- 由于定时连续传来数字，所以数字会持续递增，如下：



// AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.misoo.pkm"
    android:versionCode="1"
    android:versionName="1.0.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <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>
        <service android:name=".myService" > </service>
    </application>
</manifest>
```

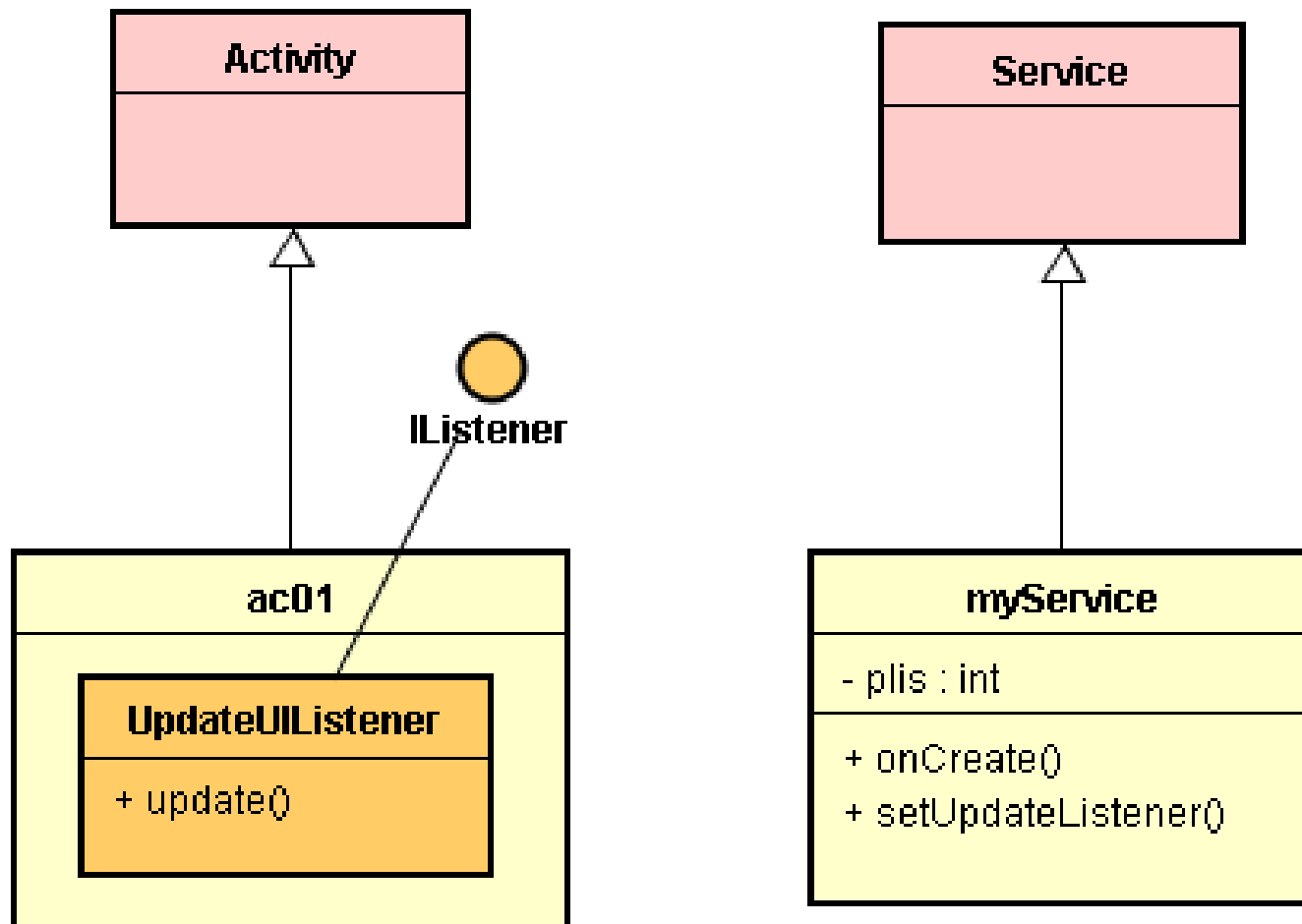
- 这AndroidManifest.xml指定了：ac01和myService是在同一进程里执行。



撰写步骤

Step-1: 定义 IListener接口：

```
// IListener.java  
package com.misoo.ppvv;  
public interface IListener {  
    public void update(String s);  
}
```

```
// ac01.java
```

```
// .....
```

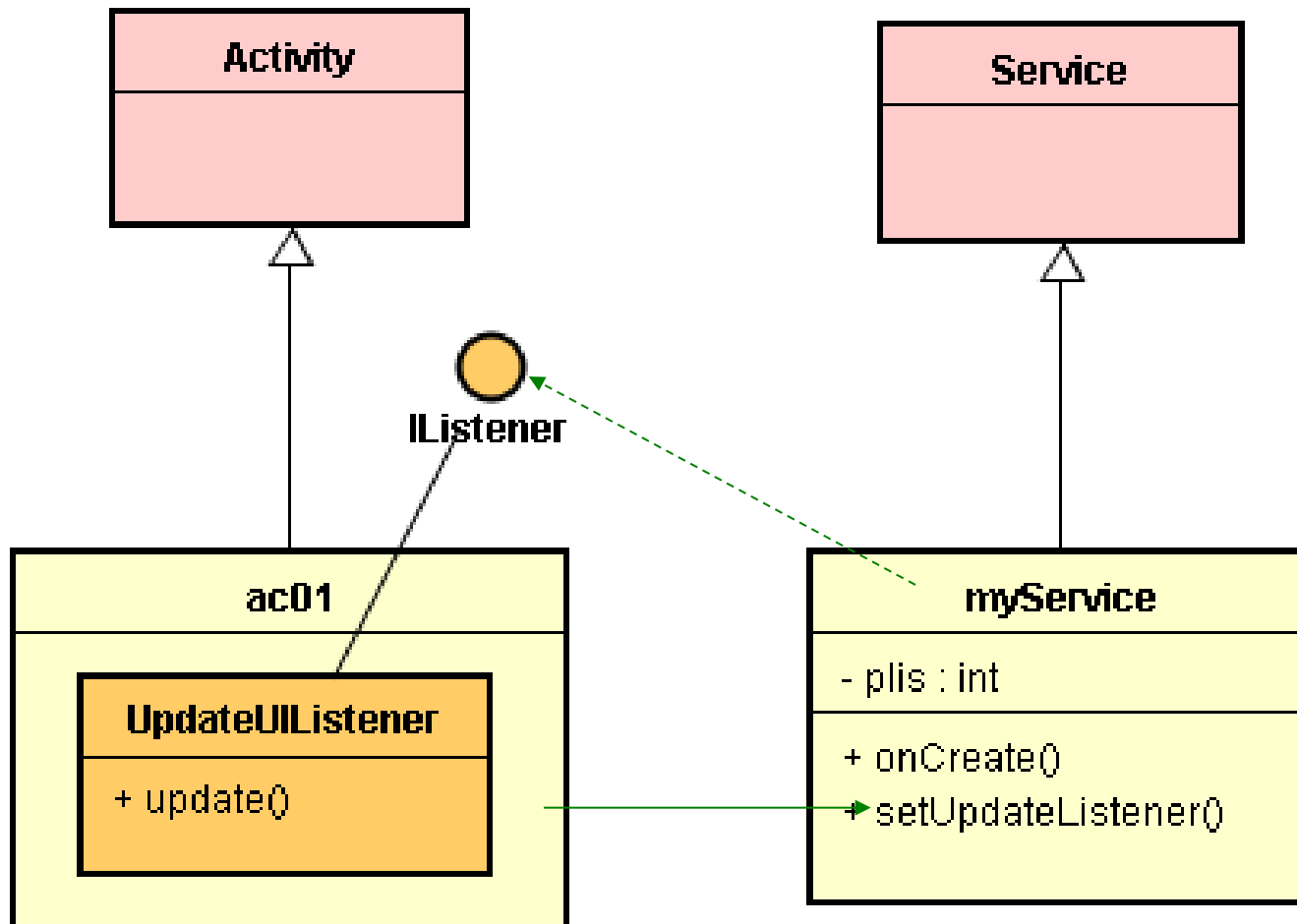
```
public class ac01 extends Activity implements OnClickListener {  
    private final int WC =  
        LinearLayout.LayoutParams.WRAP_CONTENT;  
    private TextView tx;  
    private Button btn;  
    @Override public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
        LinearLayout layout = new LinearLayout(this);  
        layout.setOrientation(LinearLayout.VERTICAL);  
        LinearLayout.LayoutParams param =  
            new LinearLayout.LayoutParams(100, WC);  
        param.topMargin = 5;  
        tx = new TextView(this);    tx.setTextSize(26);  
        tx.setTextColor(Color.RED);  
        tx.setBackgroundResource(R.drawable.x_yellow);  
        layout.addView(tx, param);  
    }  
}
```

```

    btn = new Button(this);    btn.setText("Exit");
    btn.setBackgroundResource(R.drawable.earth);
    btn.setOnClickListener(this);
    layout.addView(btn, param);
    this.setContentView(layout);
    //-----
    myService.setUpdateListener( new UpdateUIListener() );
    Intent svc = new Intent(this, myService.class);
    startService(svc);
}
@Override protected void onDestroy() {
    super.onDestroy();
    Intent svc = new Intent(this, myService.class);
    stopService(svc);
}
class UpdateUIListener implements IListener {
    public void update(String s) tx.setText(" " + s);
}
public void onClick(View v) { this.finish(); }
}

```

- 诞生一个UpdateUIListner的对象，然后将此对象之参考值(Reference)传递给myService，存于plis参考变量里。



```
// myService.java
```

```
// .....
```

```
public class myService extends Service {
```

```
    private static IListener plis;
```

```
    private static int k = 0;
```

```
    private Timer timer = new Timer();
```

```
    public Handler handler = new Handler(){
```

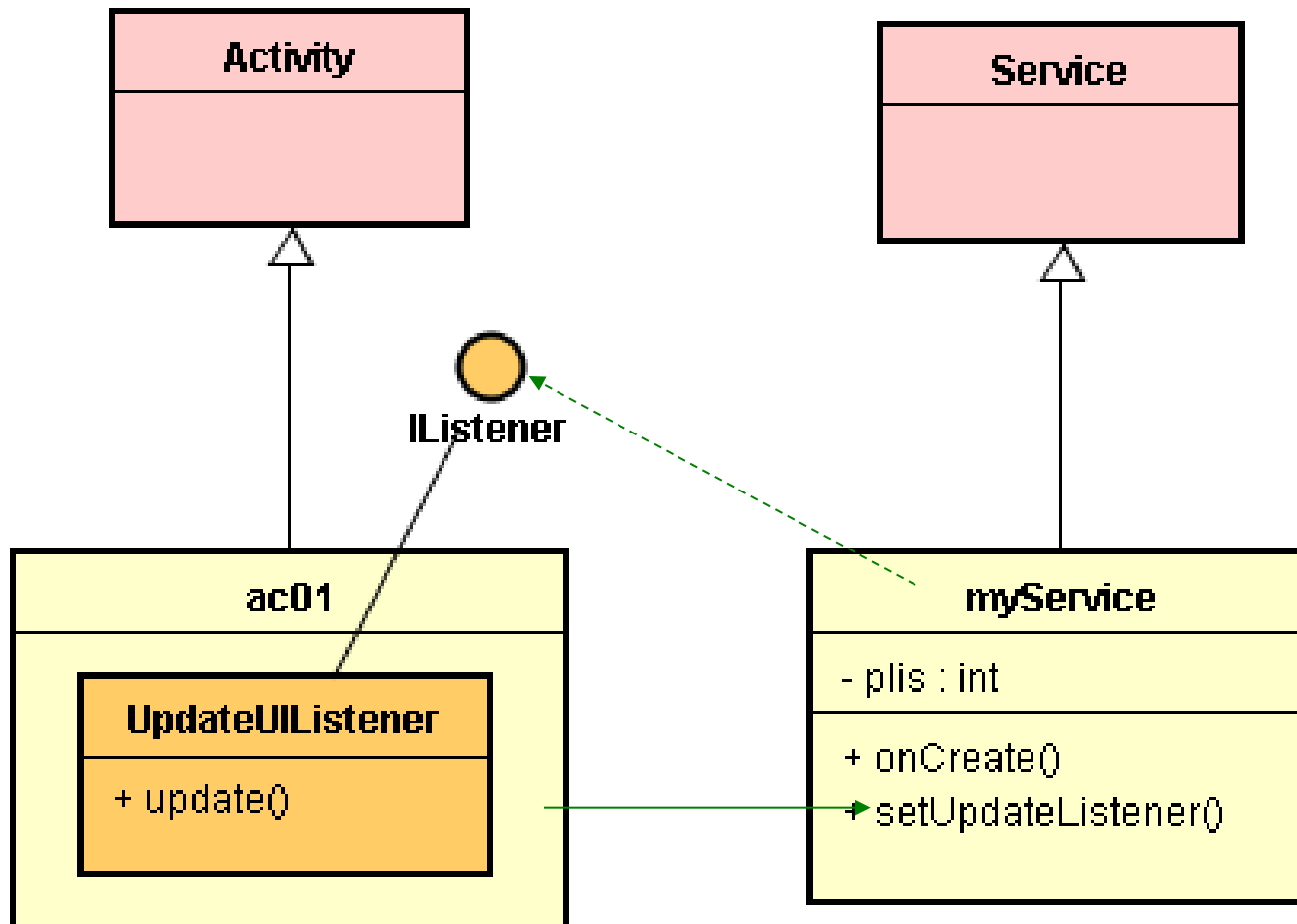
```
        public void handleMessage(Message msg) {
```

```
            plis.update(String.valueOf(k++));
```

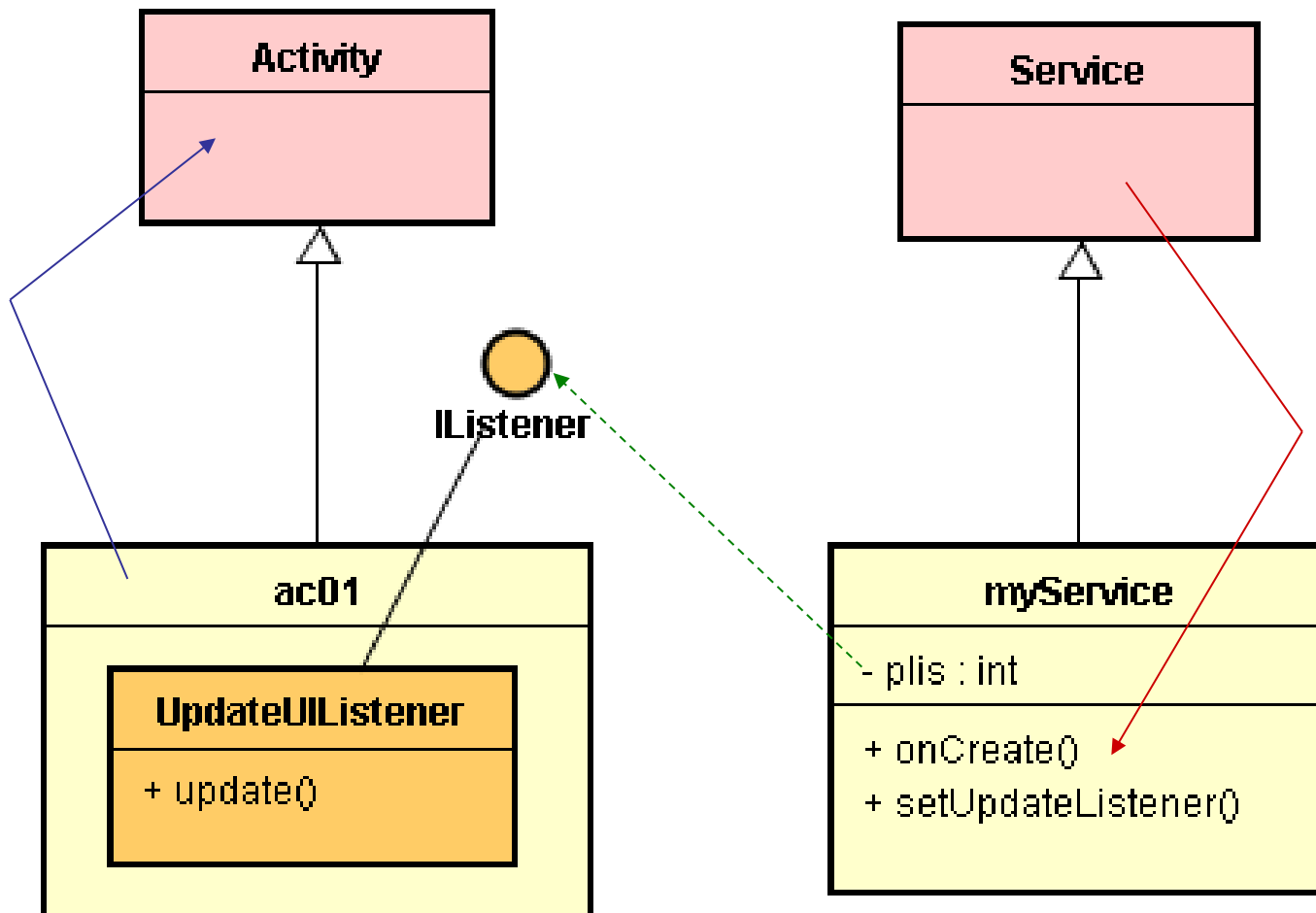
```
        }
```

```
    };
```

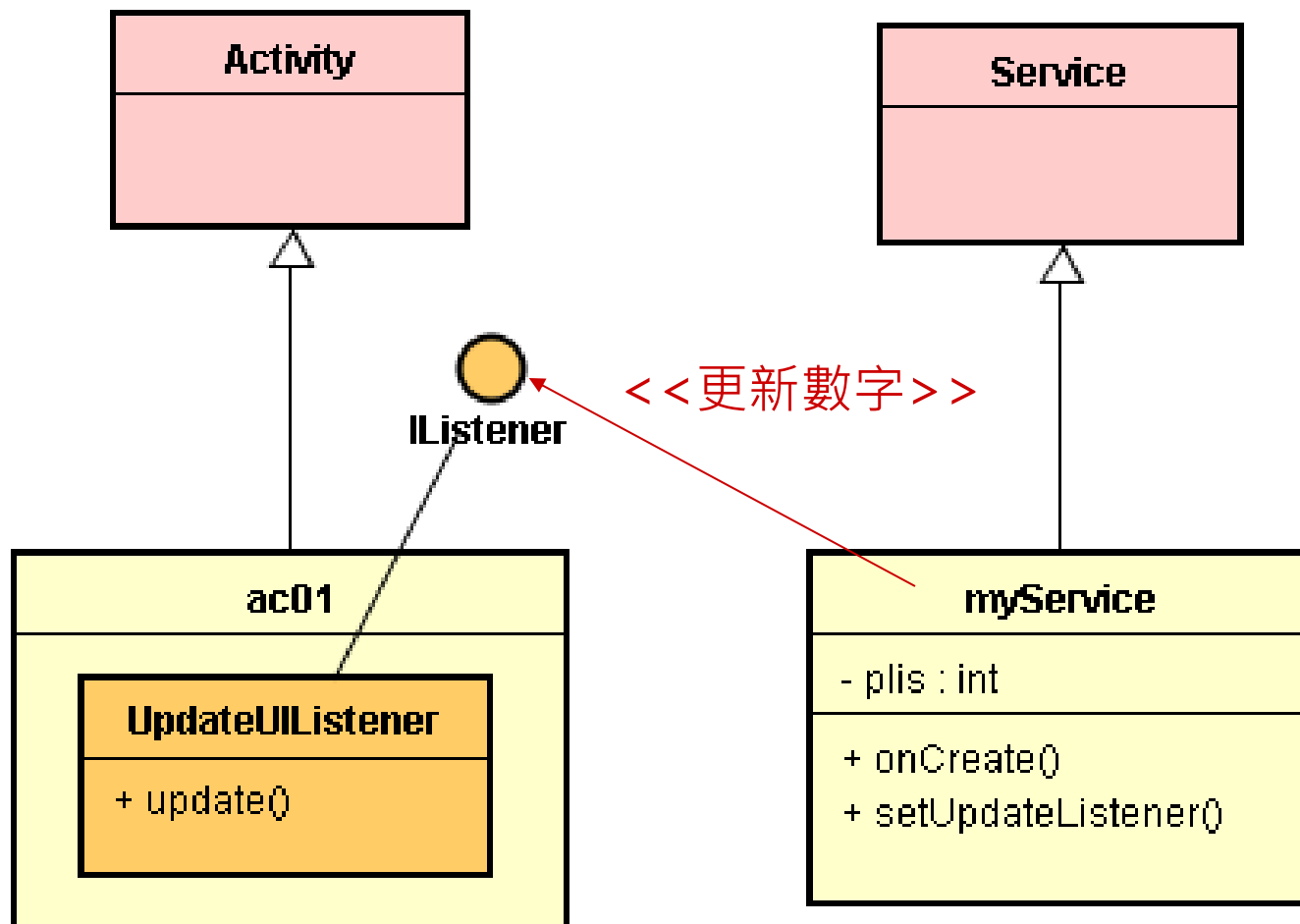
```
@Override public void onCreate() {  
    super.onCreate();  
    TimerTask task = new TimerTask() {  
        @Override public void run() {  
            handler.sendMessage(0);  
        }  
    };  
    timer.schedule(task,1000, 4000);  
}  
@Override public IBinder onBind(Intent intent)  
    { return null; }  
public static void setUpdateListener(Listener listener)  
    { plis = listener; }  
}
```



- 接着，诞生一个Intent对象去要求Android启动myService。就执行到myService的onCreate()函数。



- 此onCreate()函数诞生一个Timer对象，它会定时呼叫Handler类别的handleMessage()函数。
- 此onCreate()函数就透过plis而将数据传递给ac01里的UpdateUIListner对象，然后将数字呈现于画面上，并且循环下去。

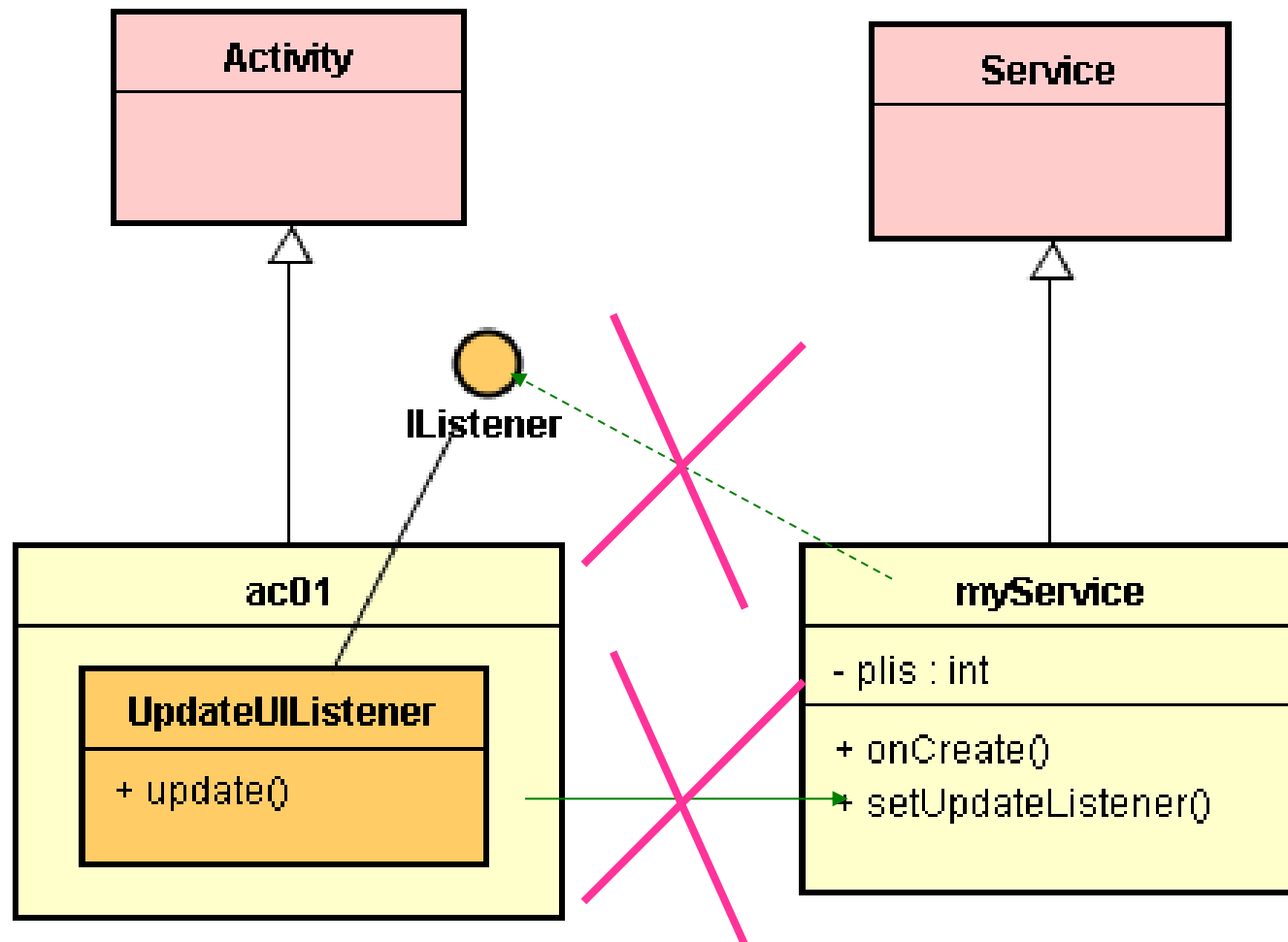


短程通信 vs. 远程通信

// AndroidManifest.xml

//.....

```
<service android:name=".myService" android:process=":remote" >  
    <intent-filter>  
        <action android:name  
            ="com.misoo.pkm.REMOTE_SERVICE" />  
    </intent-filter></service>  
</manifest>
```





~ Continued ~