

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>

F08_d

观摩：WebView的 PhoneGap小框架设计(d)

By 高煥堂

4、使用Proxy-Stub模式 来包装IPlugin接口

- 基于上述的PhoneGap框架，可以写JavaScript来呼叫Java插件。在许多范例里，你会看到如下的一段程序代码：

```
var myProxy=function(){};
myProxy.prototype.play = function(success, fail, para1, para2){
    return PhoneGap.exec( success, fail,
        'myPlugin',          //java类别名称，注册于plugins.xml中
        'PLAY',              //在Plugin里用来配对的action名称
        [para1, para2]       //所传递的参数(parameter)，Array结构
    );
};
```

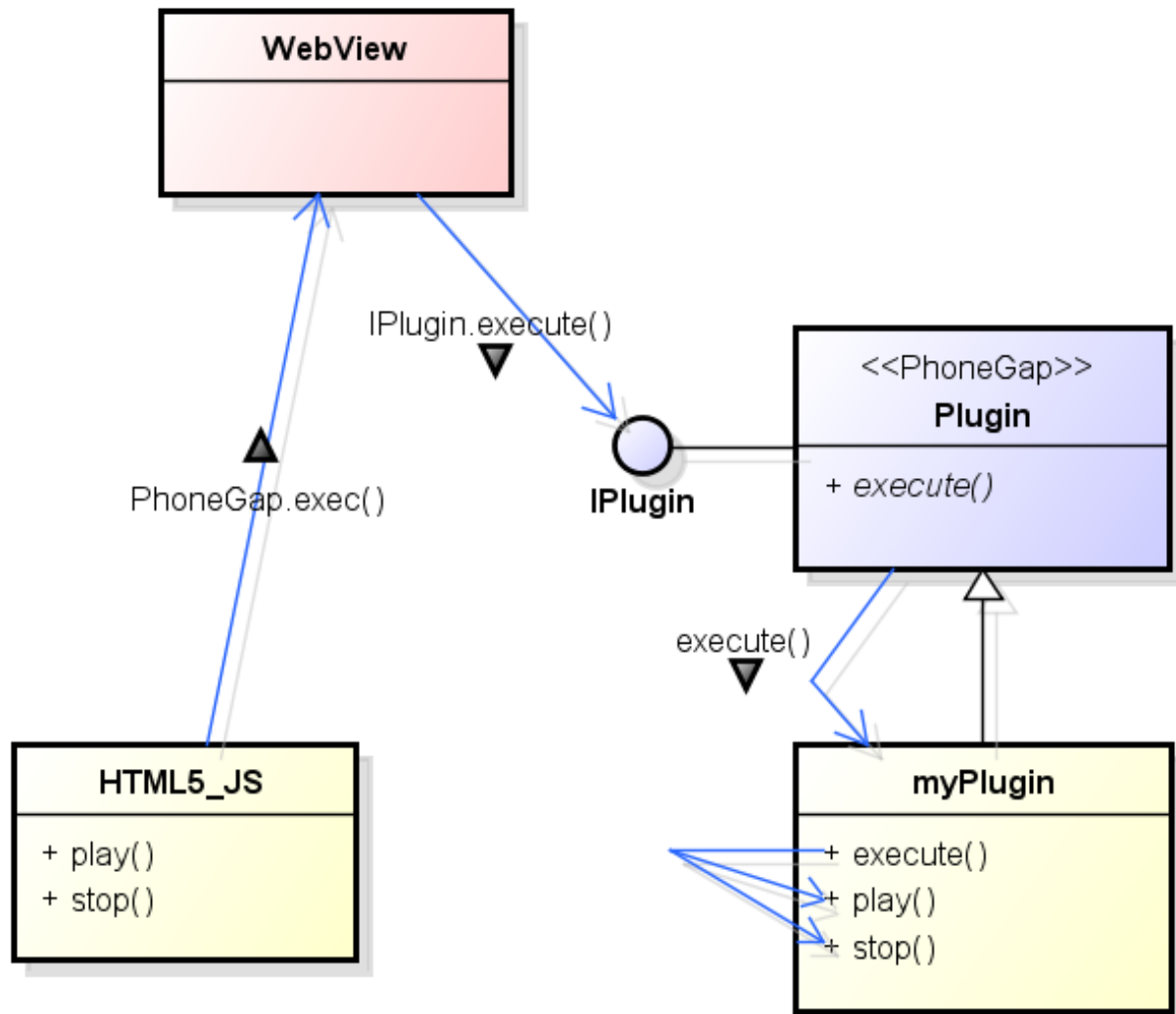
```
myProxy.prototype.stop = function(success, fail, para1, para2){  
    return PhoneGap.exec( success, fail,  
        'myPlugin',      //java类别名称，注册于plugins.xml中  
        'STOP',          //在Plugin里用来配对的action名称  
        [para1, para2]   //所传递的参数(parameter)，Array结构  
    );  
};
```

```
PhoneGap.addConstructor(function() {  
    PhoneGap.addPlugin('myProxy', new myProxy());  
});
```

- 有些人就会问到：为什么要写这段程序代码呢？我们可看看：
- <http://www.cnblogs.com/zyip/archive/2012/05/12/2497029.html>
- 在这篇文章里，作者写到：

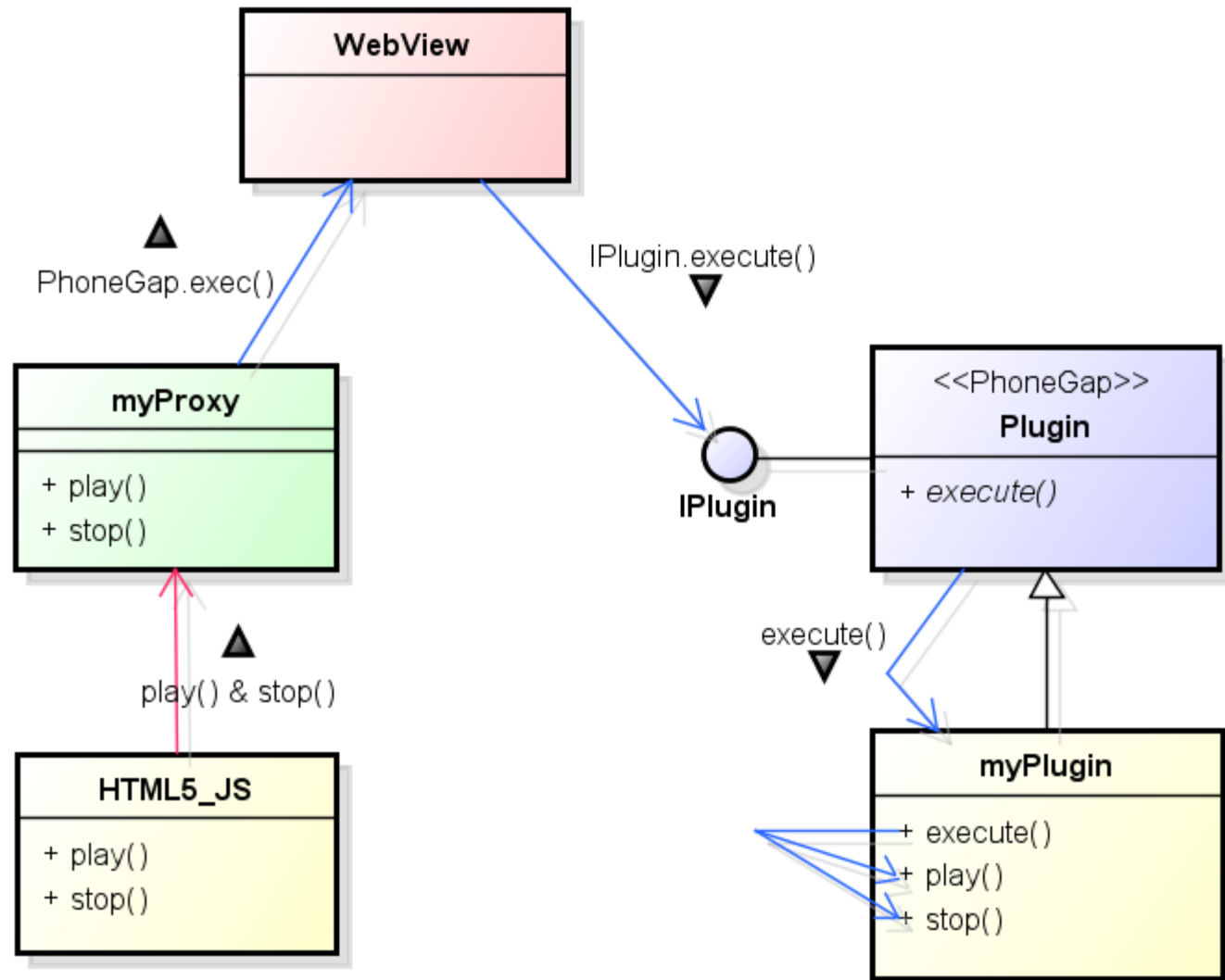
“You will need to create a class with functions that will mirror the logic exposed by the native code.” (你必须创建一个类别，来扮演Java本地代码的逻辑的映像)

- 这样的说明，可能让许多人仍然模糊不清。于此，让我来说明一下。
- 在PhoneGap框架里，由HTML5_JS模块呼叫Java插件的基本架构如下：

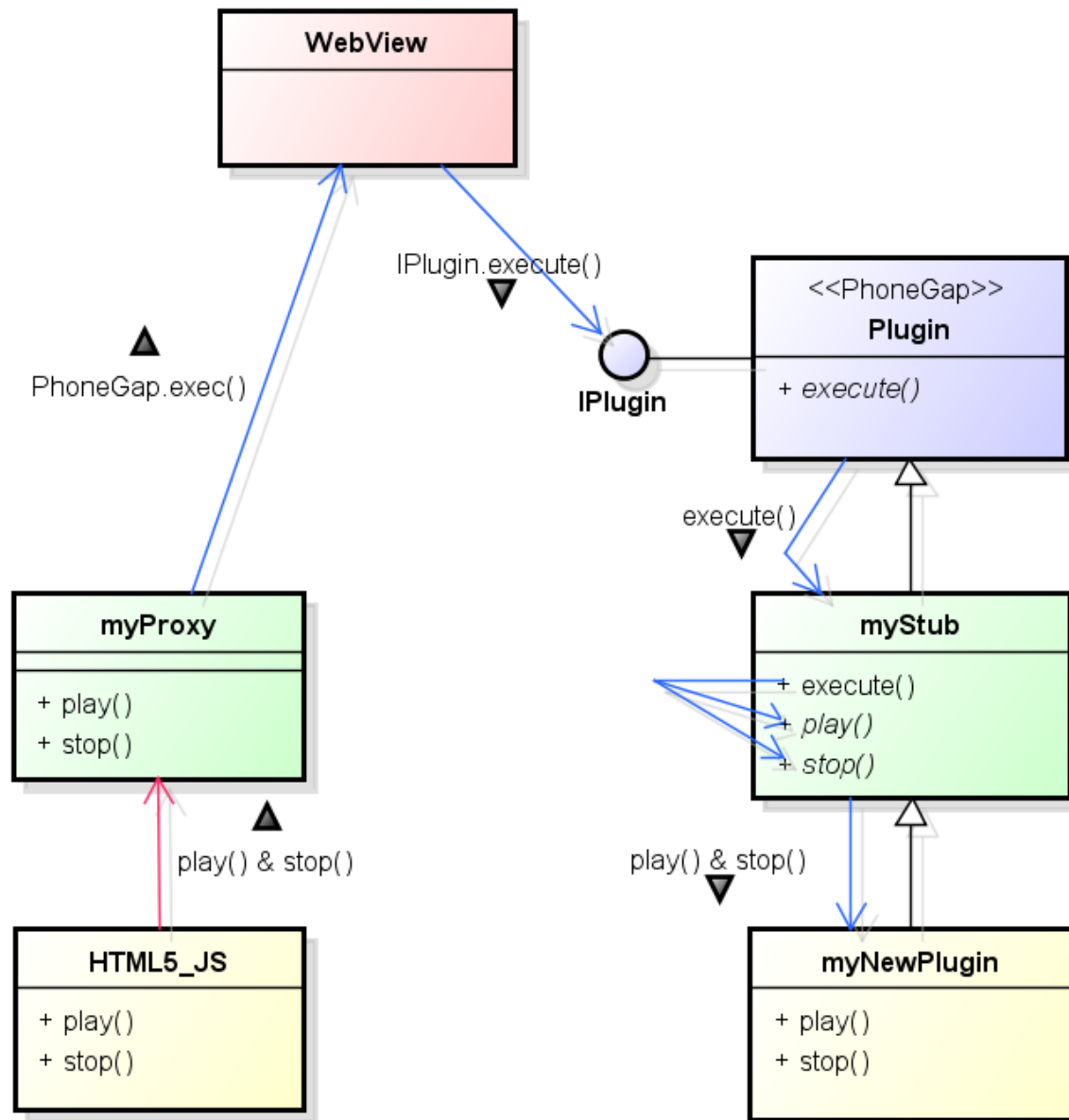


- 由于IPlugin接口只提供单一函数：
`execute()`；使得PhoneGap也只能提供单一函数：`exec()`。
- 然而，在HTML5_JS里可能有多个函数，例如`play()`和`stop()`等。
- 于是，在HTML5_JS里，必须从`play()`函数转而呼叫`PhoneGap.exec()`。

- 这个PhoneGap.exec()再透过WebView而呼叫到IPlugin.execute()。
- 如果我们在上述架构里面，加上一个myProxy类别，它包装了PhoneGap.exec()函数，扮演Java Plugin的Proxy角色，如下图所示：



- 此时，HTML5_JS里面就看不到PhoneGap.exec()函数了，其好处是：
- 一方面让HTML5_JS程序代码比较单纯；
- 另一方面，让HTML5_JS与PhoneGap.exec()两者相依性(Dependency)降低，让HTML5_JS不受制于PhoneGap的框架界面。也因而提升了PhoneGap框架接口的变动自由度。
- 通常，在框架设计里，myProxy和myStub会是成对的，这称为Proxy-Stub模式。



- 其中的指令：

```
PhoneGap.addPlugin('myProxy', new myProxy());
```

- 这诞生了一个myProxy类别的对象，然后将其指针存入于window.plugins.myProxy里。于是，在HTML5_JS里面就能呼叫myProxy.play()或myProxy.stop()，如下述的程序代码：

```
<script type="text/javascript" src="cordova-1.7.0.js"></script>
<script type="text/javascript" src="json.js"></script>
<script type="text/javascript">
var hello = function(){
var success = function(data){    alert("data : " + data.para1);    };
var fail = function(e){    alert(e);    };
window.plugins.myProxy.play(success, fail, "Hello", " Mike");
}
```

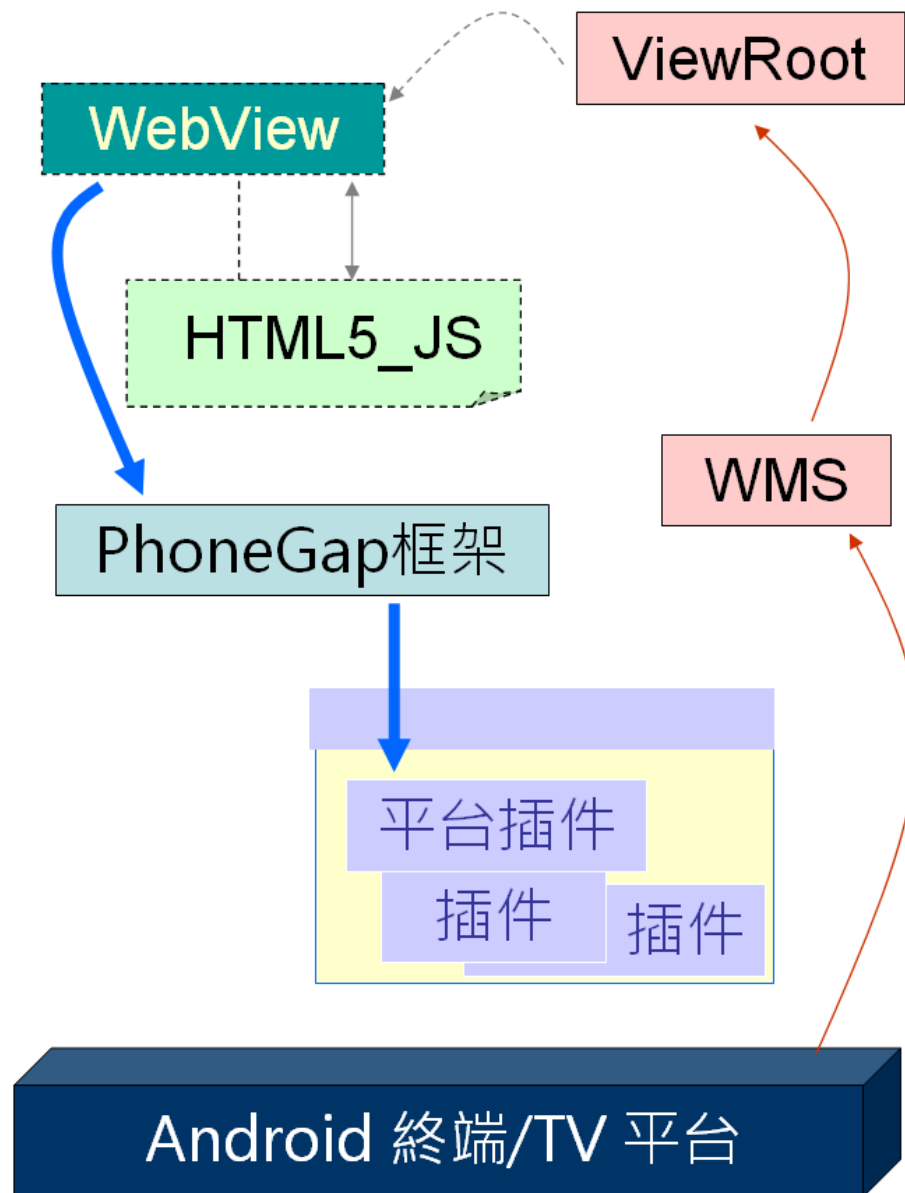

- 于此，你已经学会了使用proxy-stub模式来封装PhoneGap.exec()界面。除了PhoneGap框架之外，在Android框架里，也使用同样的技巧来封装跨进程的IBinder接口。
- 这项技巧是各种框架开发时，经常使用到的。

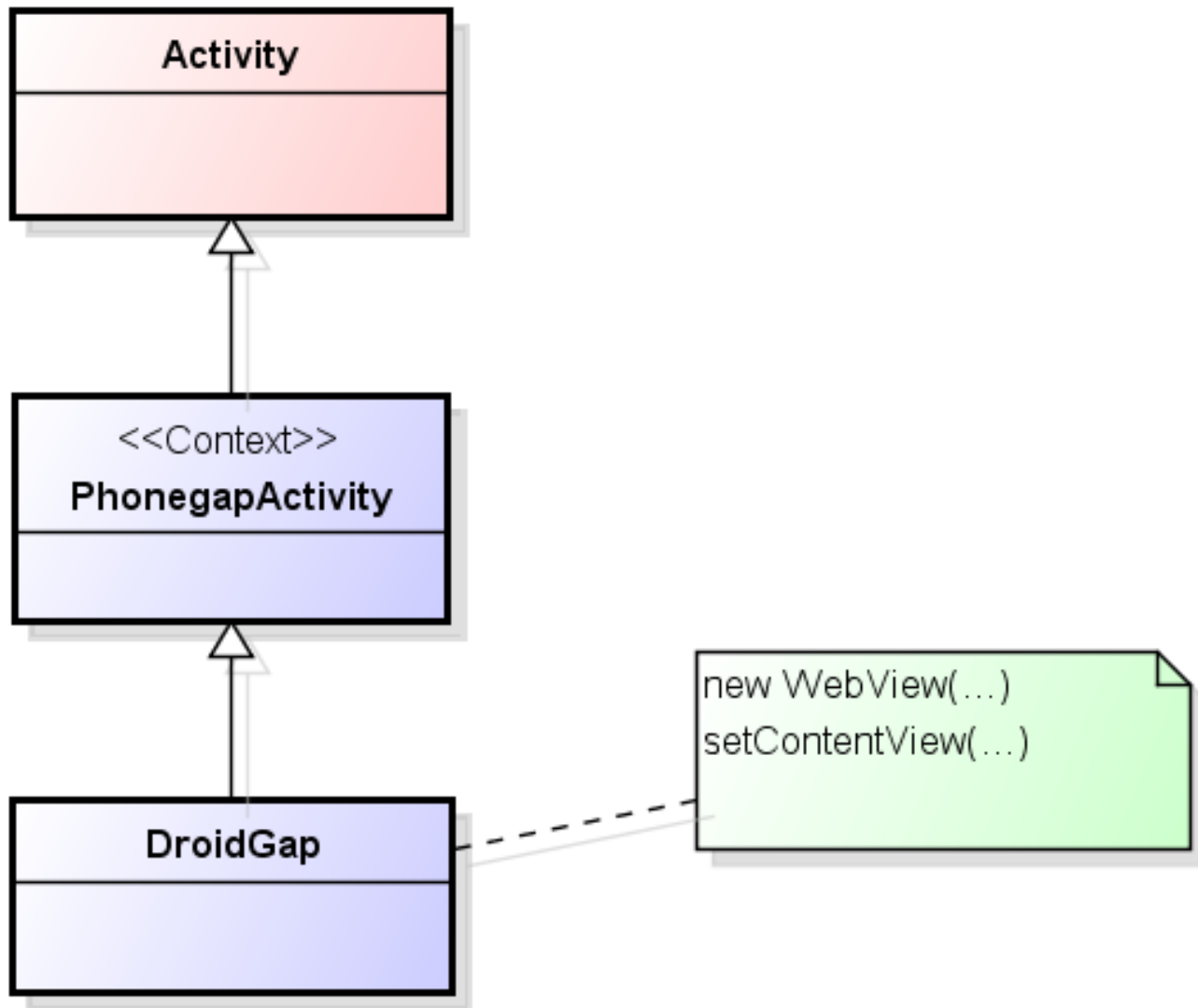
5、一般App如何使用PhoneGap框架呢？

PhoneGap框架原本的信息流向

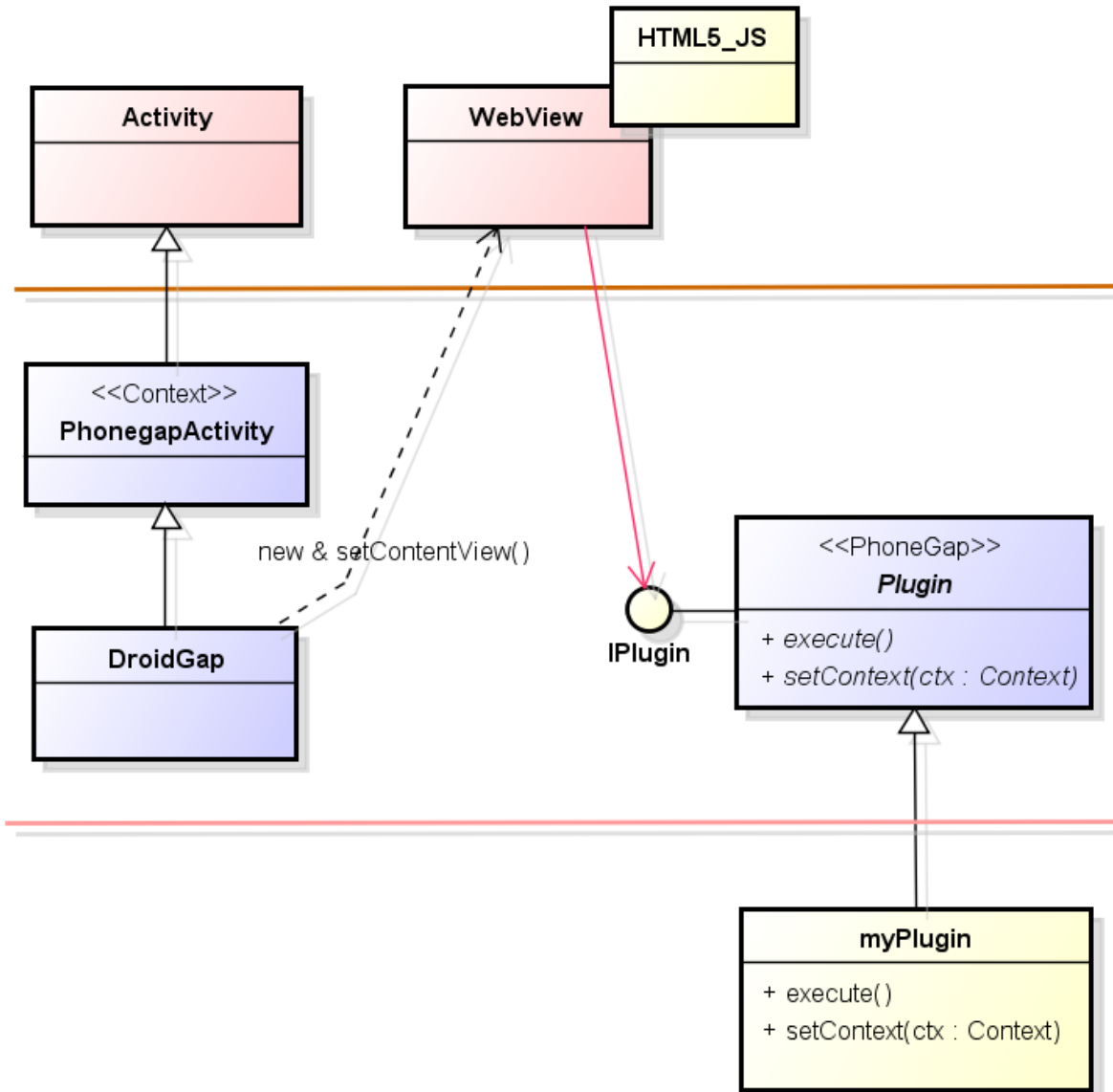
~ 承接来自WebView的事件 ~

- 原本PhoneGap是用来支持Web App开发
- Web App是以HTML5/JS语言开发的
- WebView是UI事件的来源



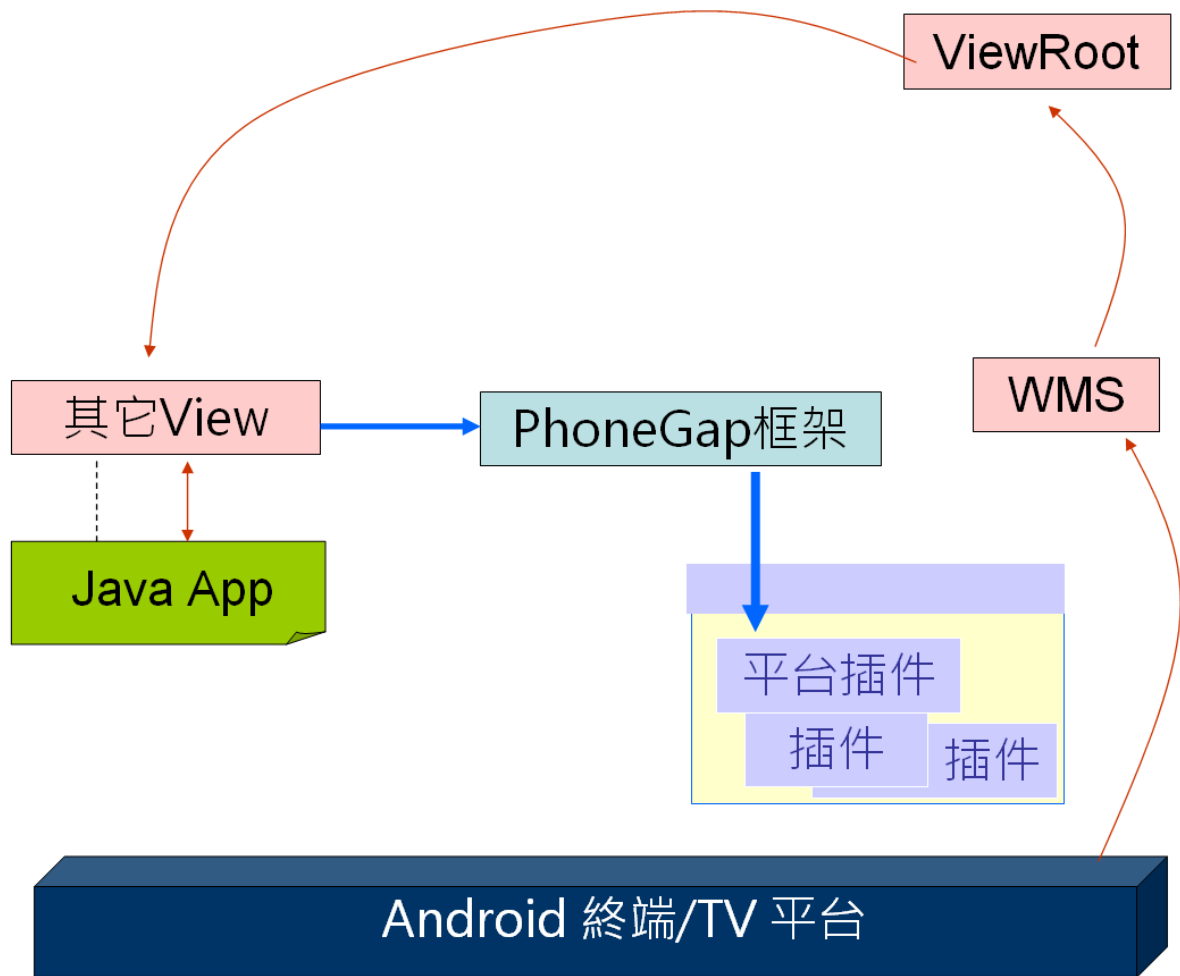


```
public class DroidGap extends PhonegapActivity {  
    // .....  
    protected WebView appView;  
    protected LinearLayout root;  
    // .....  
    public void onCreate(Bundle savedInstanceState) {  
        // .....  
        root = new LinearLayoutSoftKeyboardDetect(this, width, height);  
        // .....  
    }  
    public void init() {  
        this.appView = new WebView(DroidGap.this);  
        // .....  
        root.addView(this.appView);  
        setContentView(root);  
    }  
}
```

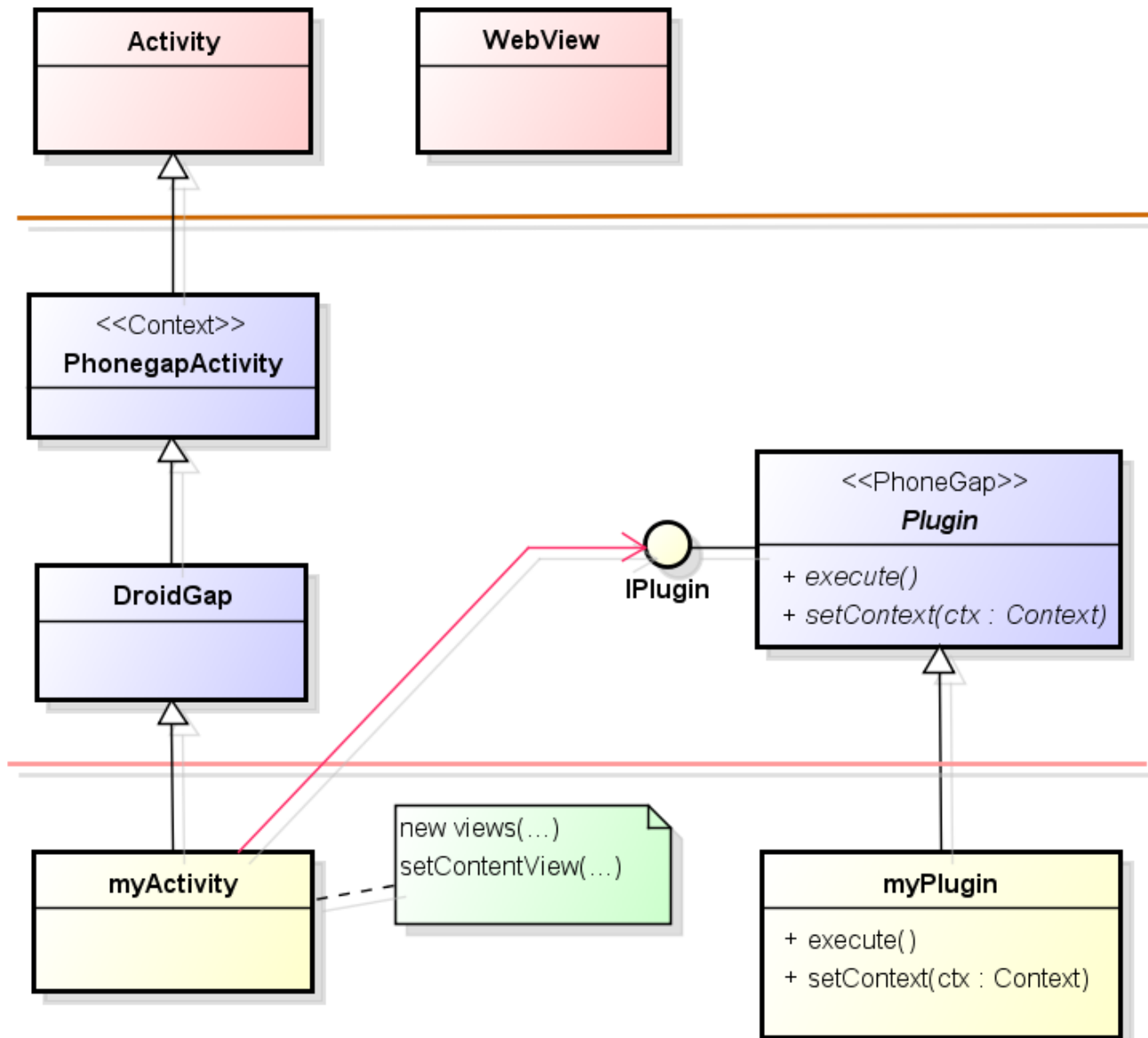


重构PhoneGap框架

- 我们可以依循GPL开源协议而修改PhoneGap的源代码，用来支持一般Android App的开发。
- 在一般App里，会用到许多View控件(如Button等)，而不局限于WebView，而这些View控件都是UI事件的来源。如果事件处理需要使用到Java插件，就能将信息传递给PhoneGap了。



- 使用一般Android App的撰写方式，在myActivity里创建有关的views,如Button、SurfaceView等；并且setContentView()来将views呈现于UI画面上。
- 当myActivity接到 UI事件时，如果需要的话，才调用PhoneGap的API，间接执行到Java插件的代码。



在重构PhoneGap框架中，
获得宝贵的架构设计经验。

Thanks...



高煥堂