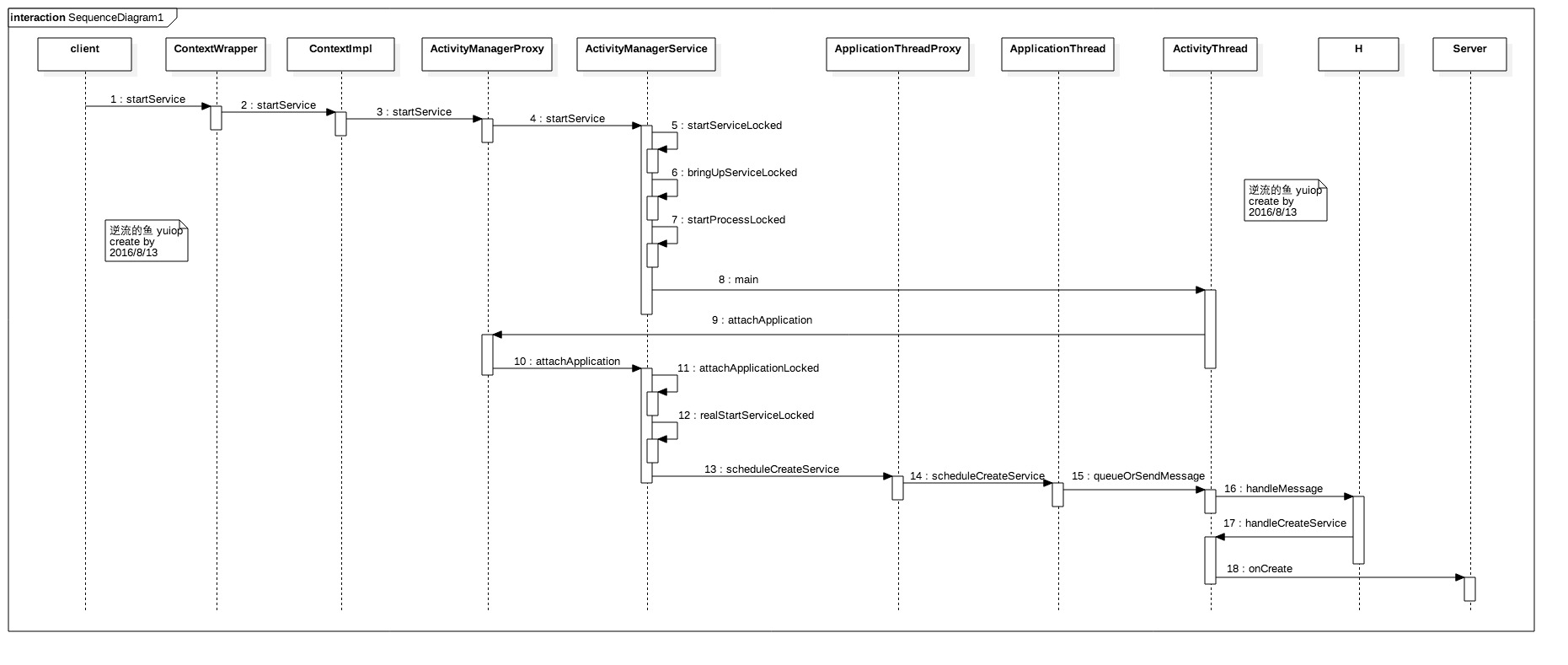
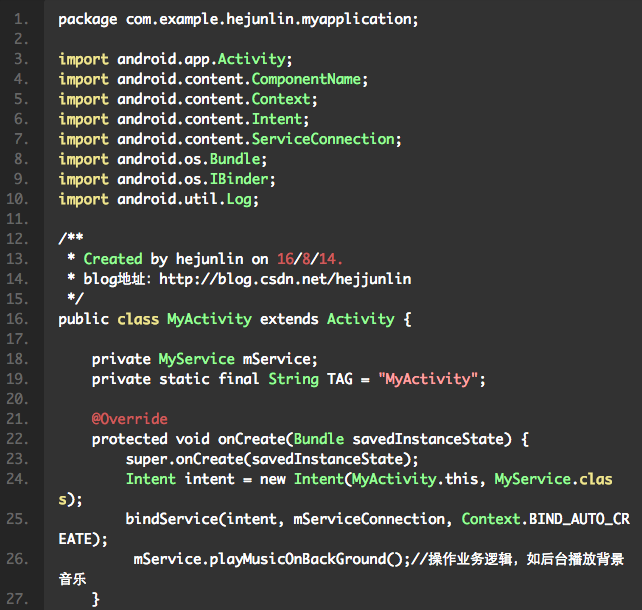
[插件占坑，四大组件动态注册前奏（二） 系统Service的启动流程](http://blog.csdn.net/hejjunlin/article/details/52203903)

本节主要记录系统Service的启动流程：   
先看时序图：   


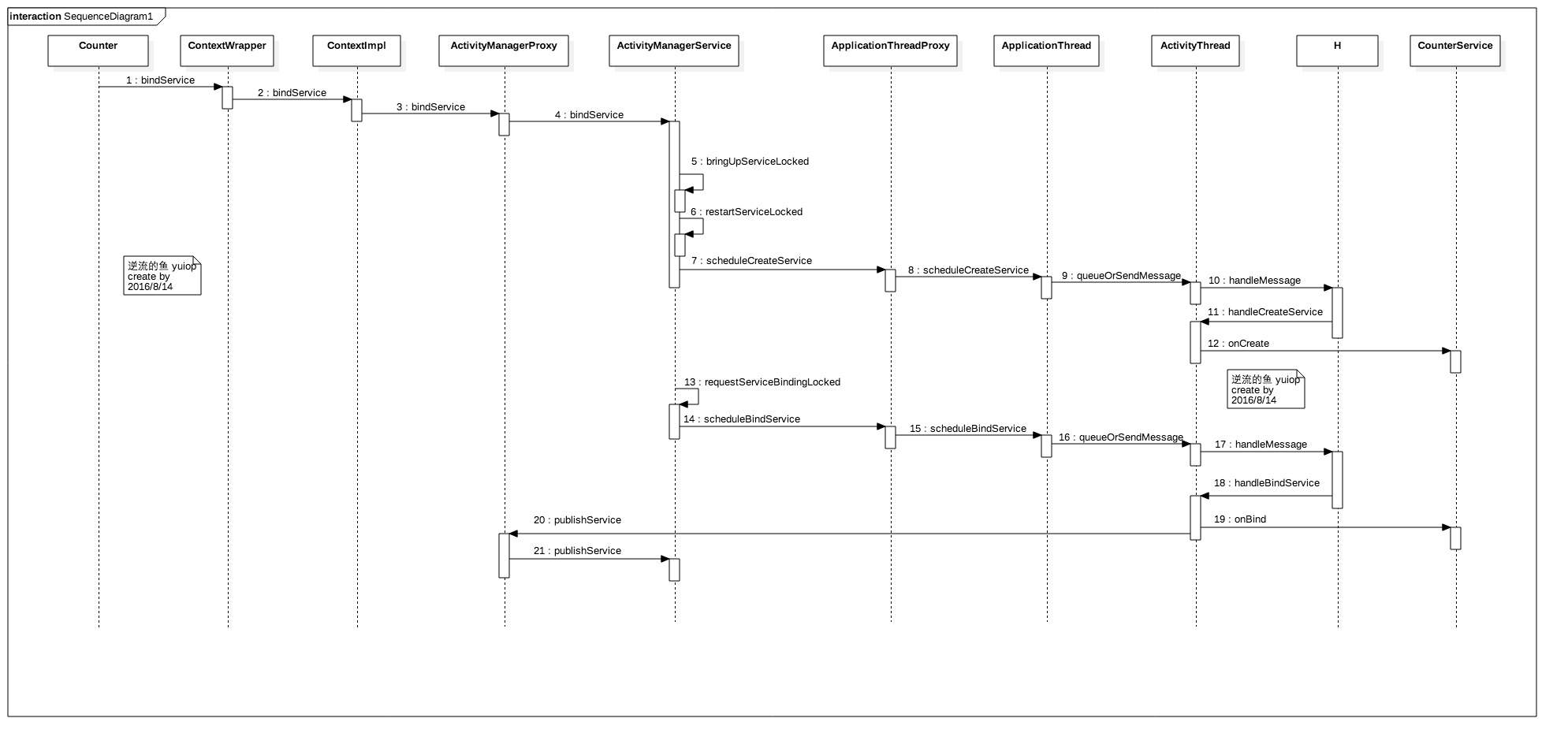
与Activity组件的启动方式很像，Service启动分为隐式和显式两种，对于隐式启动Service组件来说，我们只需要知道它的组件名称，而对于显示的Service组件来说，需要知道它的类名称。   
以一个后台播放音乐场景来说明：   
通过实现一个MyService来实现一个异步任务来播放后台音乐

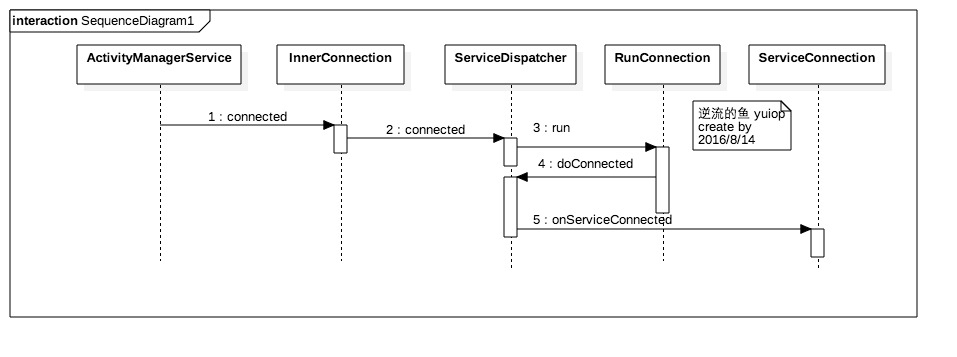
MyActivity.java   
   


MyService.java   


MyActivity组件绑定MyService的过程：

* **1.MyActivity向ActivityManagerService发送一个绑定CounterService组件的进程间通信请求。**
* **2.ActivityManagerService发现用来运行MyService组件的应用程序进程即为MyActivity组件所运行的应用程序进程，因此，它就直接通知应用程序进程将MyService启动起来。**
* **3.MyService组件启动起来后，ActivityManagerService就请求它返回一个Binder本地对象，以便MyActivity可以通过这个Binder本地对象来和MyService组件建立连接。**
* **4.ActivityManagerService将前面从MyService组件中获得的一个Binder本地对象发送给MyActivity组件。**
* **5.MyActivity组件获得了ActivityManagerService给它发送的Binder本地对象之后，就可以通过它来获得MyService组件的一个访问接口，MyActivity组件之后就可以通过这个访问接口来使用MyService组件所提供的服务，这就相当于将MyService绑定在了MyActivity中了。**

那service在系统中绑定是如何的呢？   
同样看下时序图：   




客户端组件启动Server组件的过程：

* **1.Client组件启动ActivityManagerService发送一个启动Server组件的进程间通信请求。**
* 2.ActivityManagerService发现用来运行Server组件的应用程序进程不存在，因此，它就会首先将Server组件的信息保存下来，接着再   
  创建一个新的应用程序进程
* **3.新的应用程序启动完成之后，就会向ActivityManagerService发送一个启动完成进程间通信请求，以便ActivityManagerServices可以继续执行启动Service组件的操作。**
* **4.ActitivtyManagerService将2中保存下来的Service组件信息发送级第2步创建的应用程序进程，以便它可以将Server组件启动起来。**