

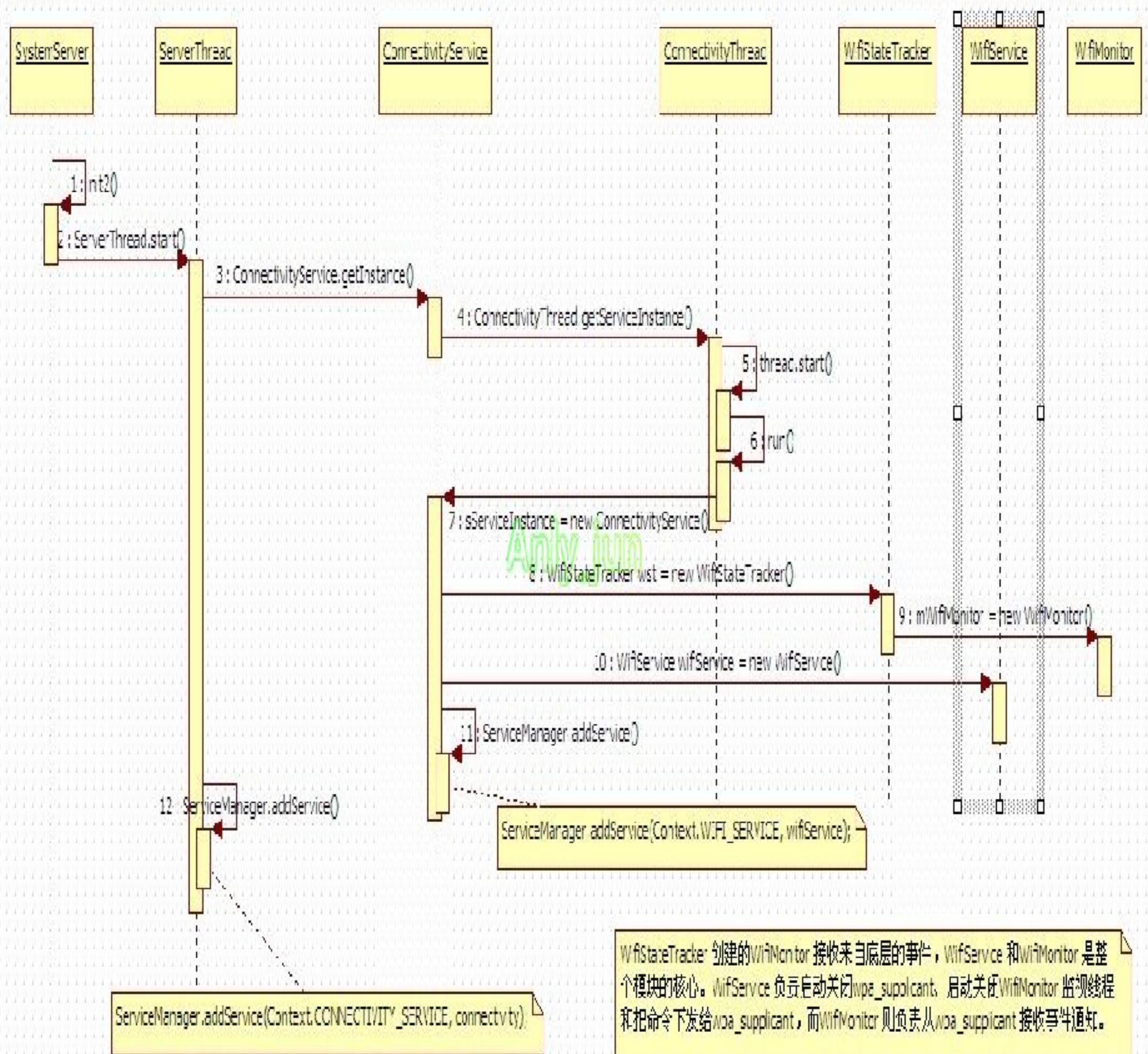
WIFI 学习

一、为了更好学习 WIFI 的这个无线网络，大家要有一起 ANDROID 手机，并且安装好驱动。只有这样才能够写出 WIFI 的程序来，并且大家可以一起来上网。

二：Wifi 模块的初始化：

在 SystemServer 启动的时候，会生成一个 ConnectivityService 的实例，

```
try {  
    Log.i(TAG, "Starting Connectivity Service.");  
    ServiceManager.addService(Context.CONNECTIVITY_SERVICE, new  
ConnectivityService(context));  
} catch (Throwable e) {  
    Log.e(TAG, "Failure starting Connectivity Service", e);  
}
```



第三部分：Wifi 模块的启动（使能）

WirelessSettings 在初始化的时候配置了由 WifiEnabler 来处理 Wifi 按钮，

```
private void initToggles() {  
mWifiEnabler = new WifiEnabler(this,  
                                (WifiManager) getSystemService(WIFI_SERVICE),  
                                (CheckBoxPreference) findPreference(KEY_TOGGLE_WIFI));
```

当用户按下 Wifi 按钮后，Android 会调用 WifiEnabler 的 onPreferenceChange，再由 WifiEnabler 调用 WifiManager 的 setWifiEnabled 接口函数，通过 AIDL，实际调用的是 WifiService 的 setWifiEnabled 函数，WifiService 接着向自身发送一条 MESSAGE_ENABLE_WIFI 消息，在处理该消息的代码中做真正的使能工作：首先装载 WIFI 内核模块（该模块的位置硬编码为“/system/lib/modules/wlan.ko”），然后启动 wpa_supplicant（配置文件硬编码为“/data/misc/wifi/wpa_supplicant.conf”），再通过 WifiStateTracker 来启动 WifiMonitor 中的监视线程。

```
private boolean setWifiEnabledBlocking(boolean enable) {  
    final int eventualWifiState = enable ? WIFI_STATE_ENABLED : WIFI_STATE_DISABLED;  
    updateWifiState(enable ? WIFI_STATE_ENABLING : WIFI_STATE_DISABLING);  
    if (enable) {  
        if (!WifiNative.loadDriver()) {  
            Log.e(TAG, "Failed to load Wi-Fi driver.");  
            updateWifiState(WIFI_STATE_UNKNOWN);
```

```

        return false;
    }
    if (!WifiNative.startSupplicant()) {
        WifiNative.unloadDriver();
        Log.e(TAG, "Failed to start supplicant daemon.");
        updateWifiState(WIFI_STATE_UNKNOWN);
        return false;
    }
    mWifiStateTracker.startEventLoop();
}

// Success!
persistWifiEnabled(enable);
updateWifiState(eventualWifiState);
return true;
}

```

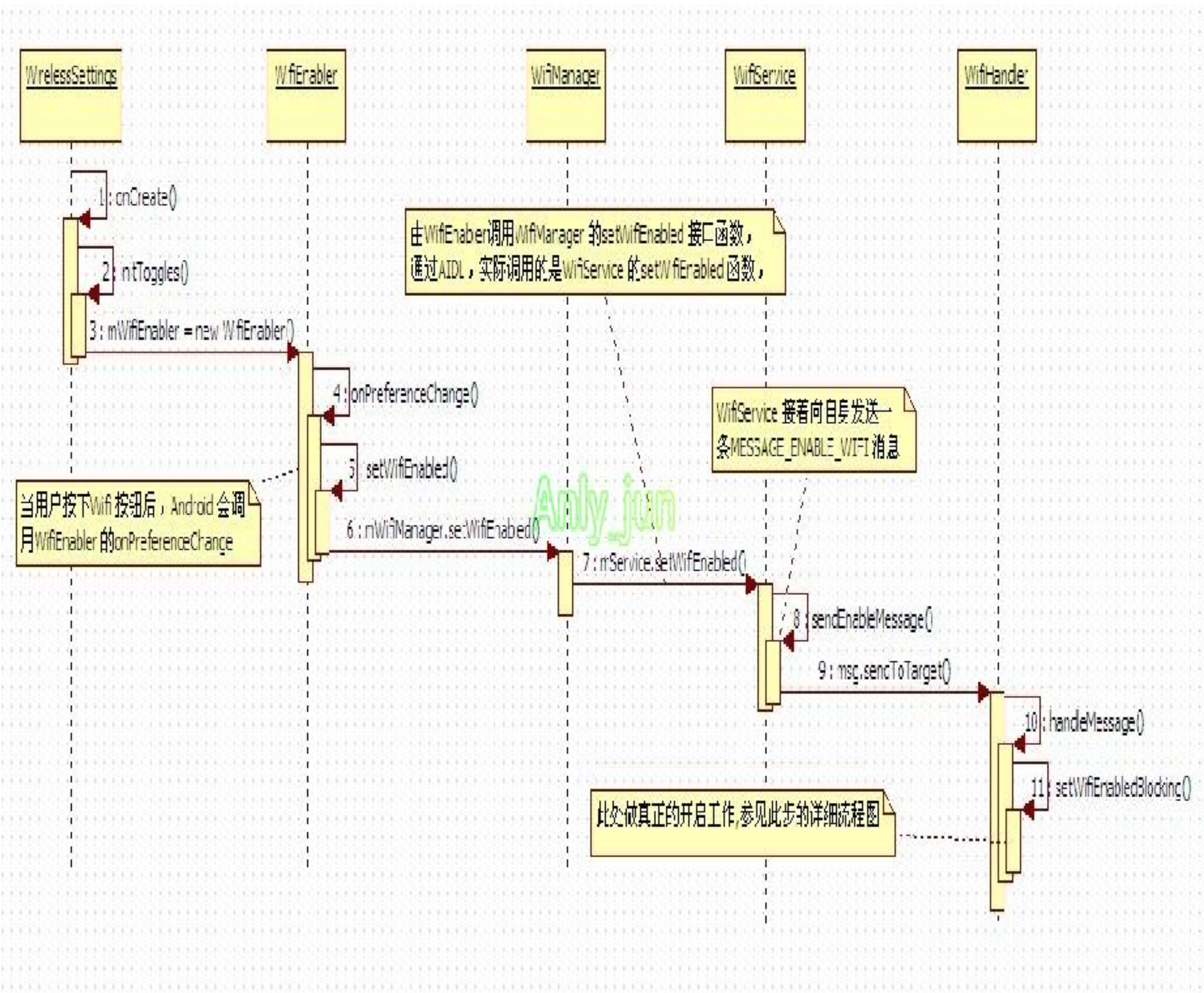
当使能成功后，会广播发送 `WIFI_STATE_CHANGED_ACTION` 这个 Intent 通知外界 WIFI 已经成功使能了。WifiEnabler 创建的时候就会向 Android 注册接收 `WIFI_STATE_CHANGED_ACTION`，因此它会收到该 Intent，从而开始扫描。

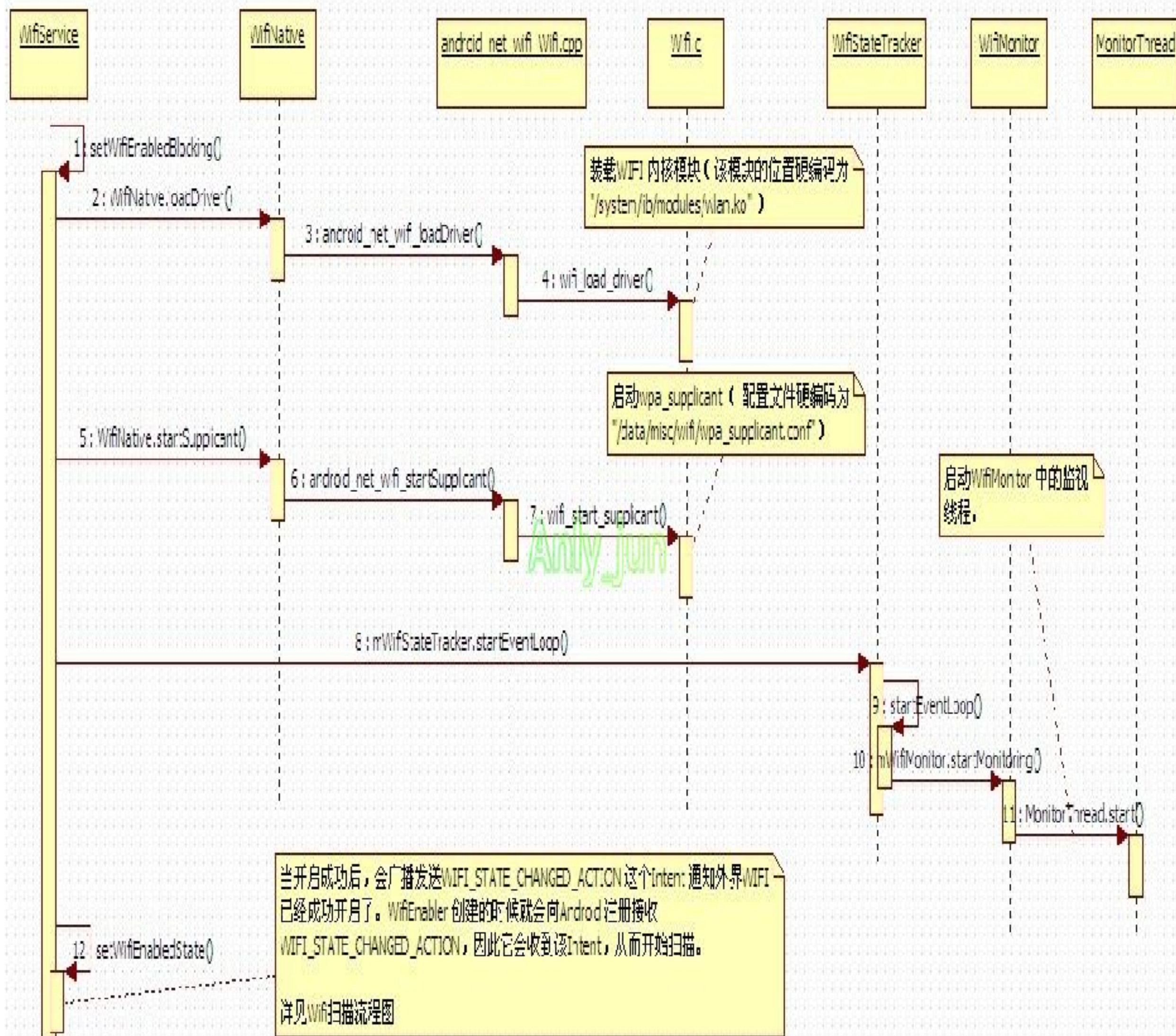
```

private void handleWifiStateChanged(int wifiState) {
    if (wifiState == WIFI_STATE_ENABLED) {
        loadConfiguredAccessPoints();
        attemptScan();
    }
}

```

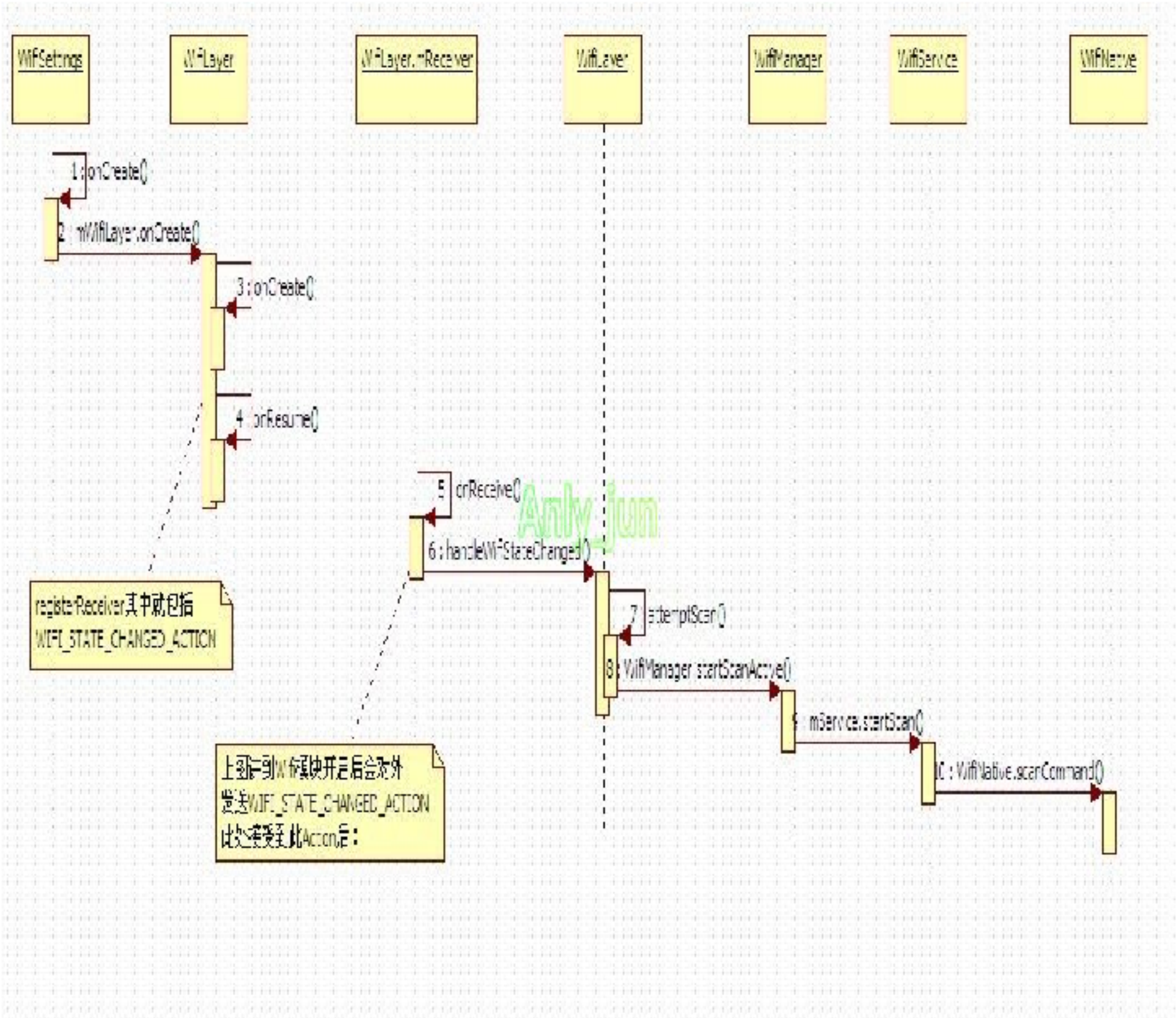

具体流程如下流程图所示:





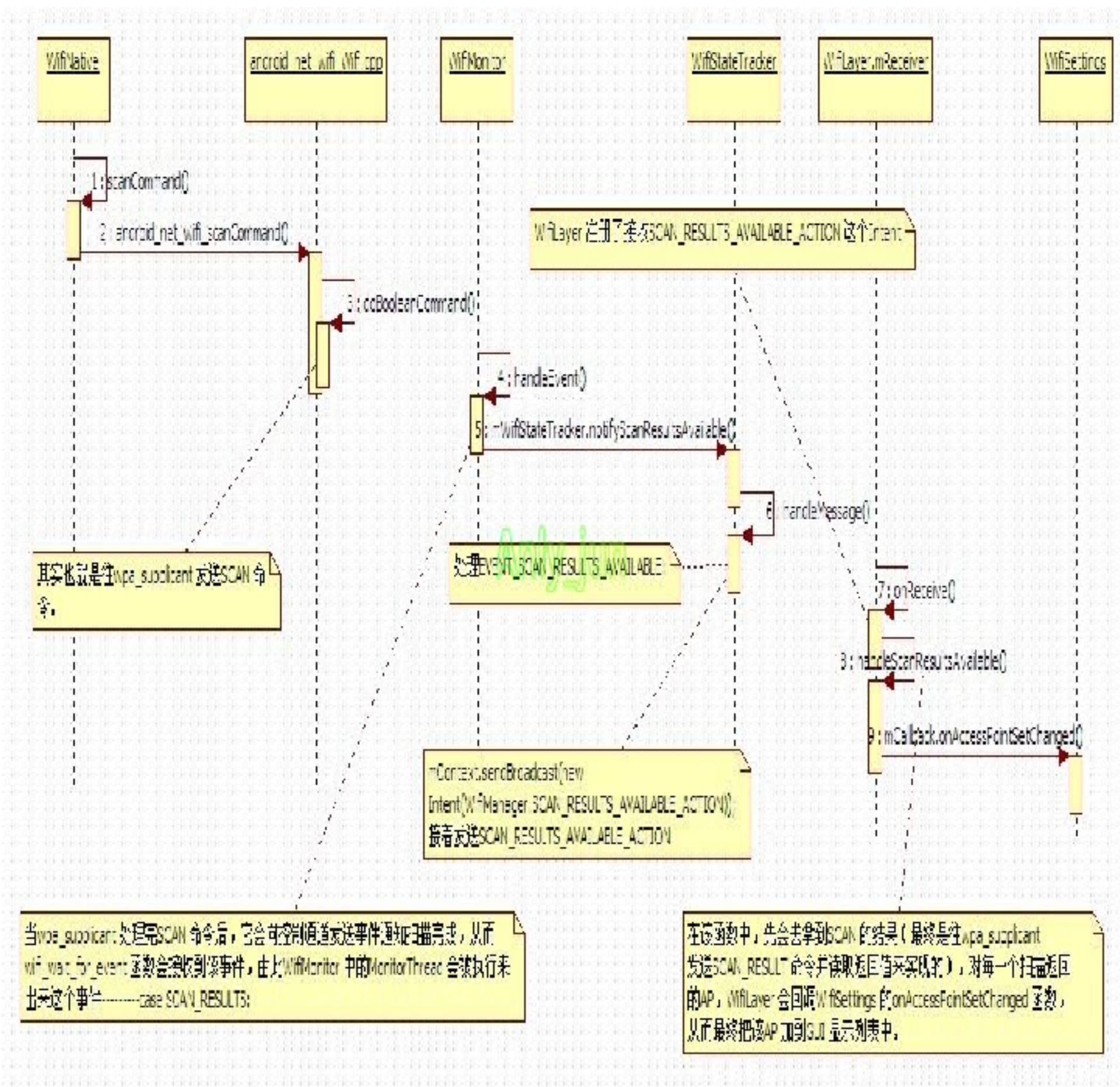
第四部分：查找热点（AP）

上一节（第三部分：Wifi 开启）中讲到 Wifi 模块开启后会对外发送 WIFI_STATE_CHANGED_ACTION。WifiLayer 中注册了 Action 的 Receiver。当 WifiLayer 收到此 Action 后开始 scan 的流程，具体如下：



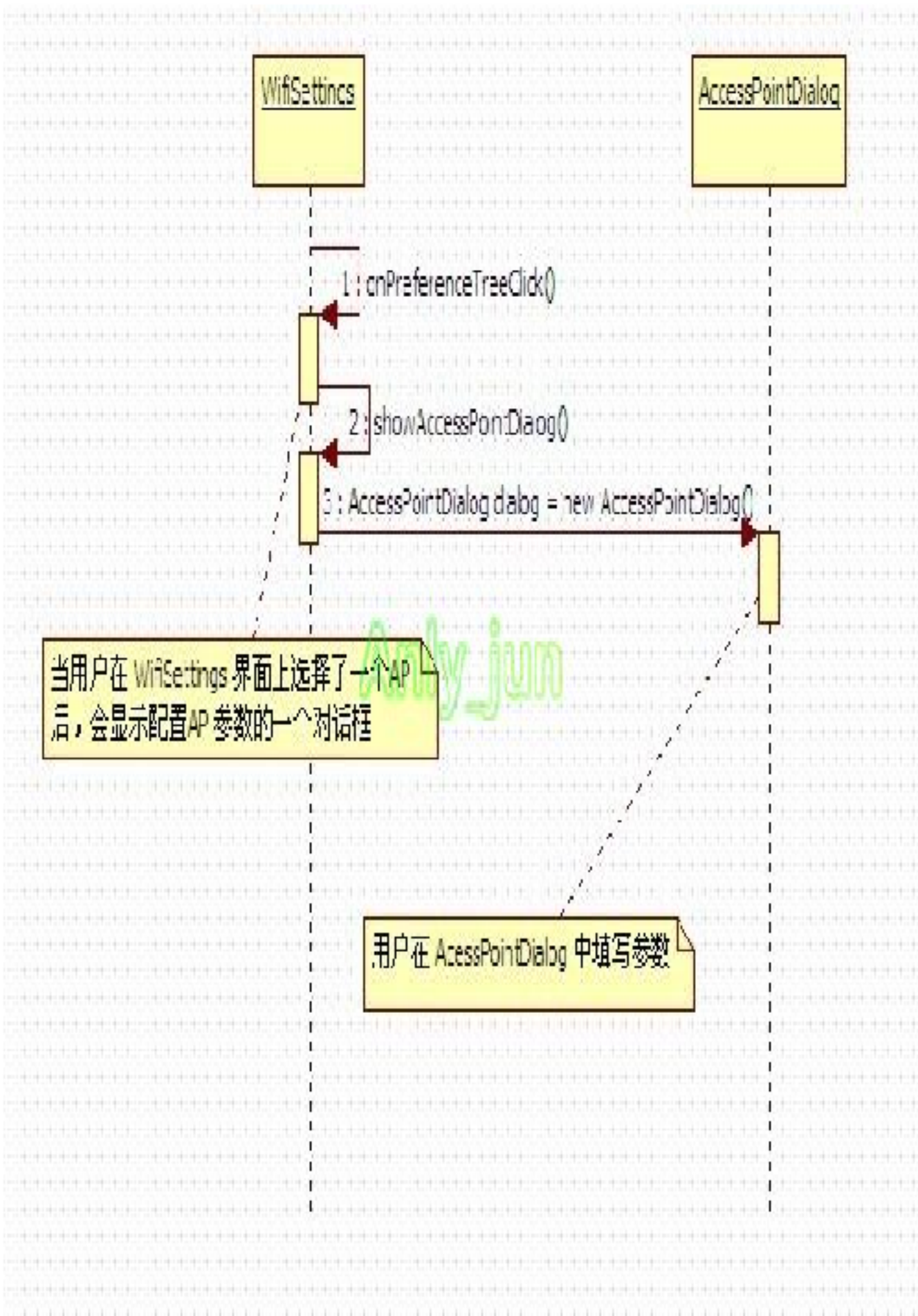
当 wpa_supplicant 处理完 SCAN 命令后，它会向控制通道发送事件通知扫描完成，从 wifi_wait_for_event 函数会接收到该事件，由此

WifiMonitor 中的 MonitorThread 会被执行来出来这个事件：



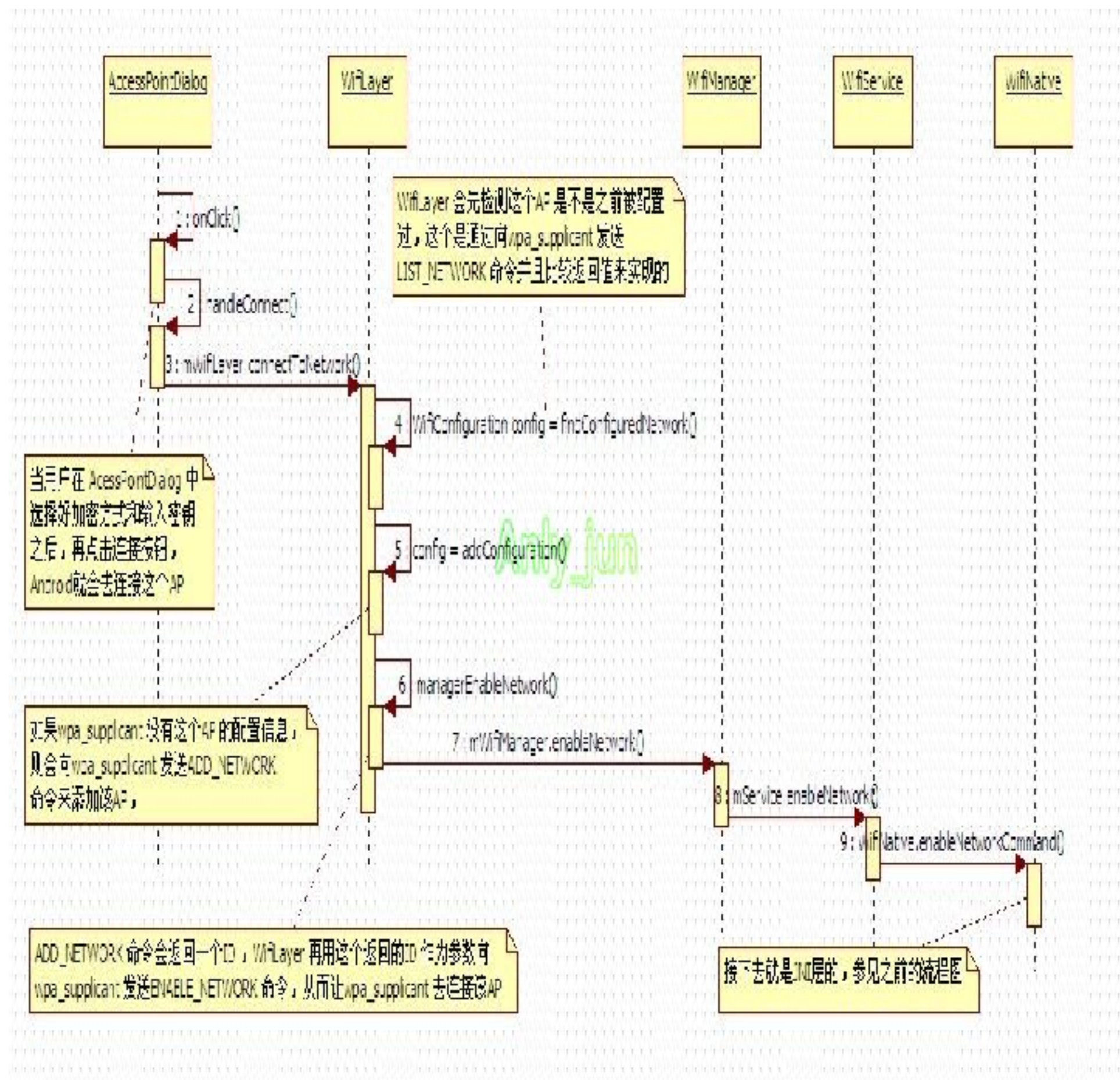
第五部分：配置 AP 参数

当用户在 WifiSettings 界面上选择了一个 AP 后，会显示配置 AP 参数的一个对话框：



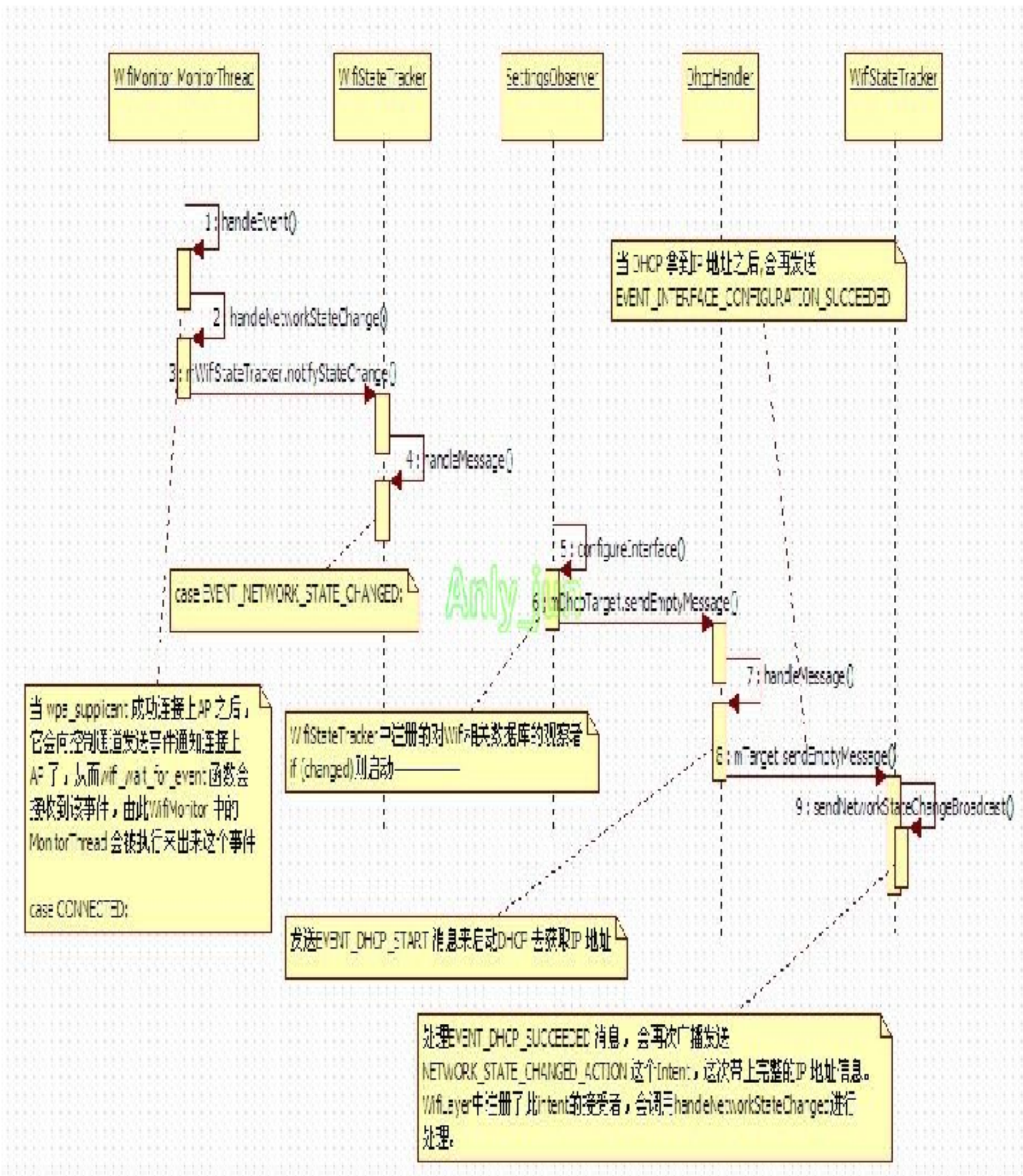
第六部分: Wifi 连接

具体流程参见以下流程图:



第七部分：IP 地址的配置

流程如图：



最后祝大家学习 WIFI 无线，有所进步。

作者: Smart