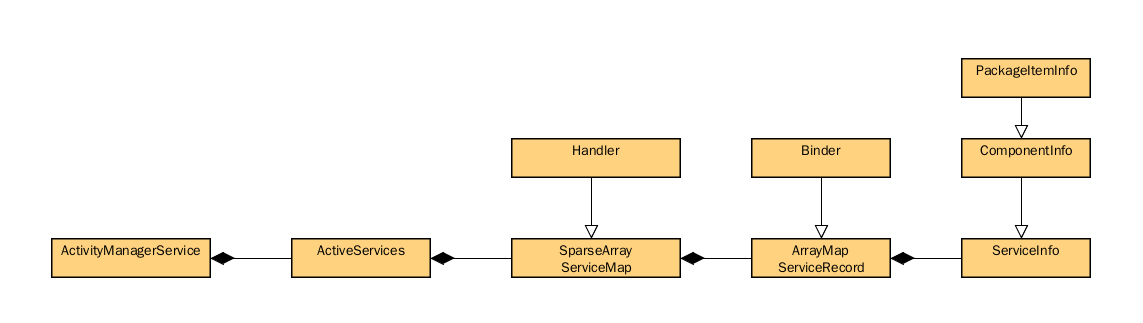
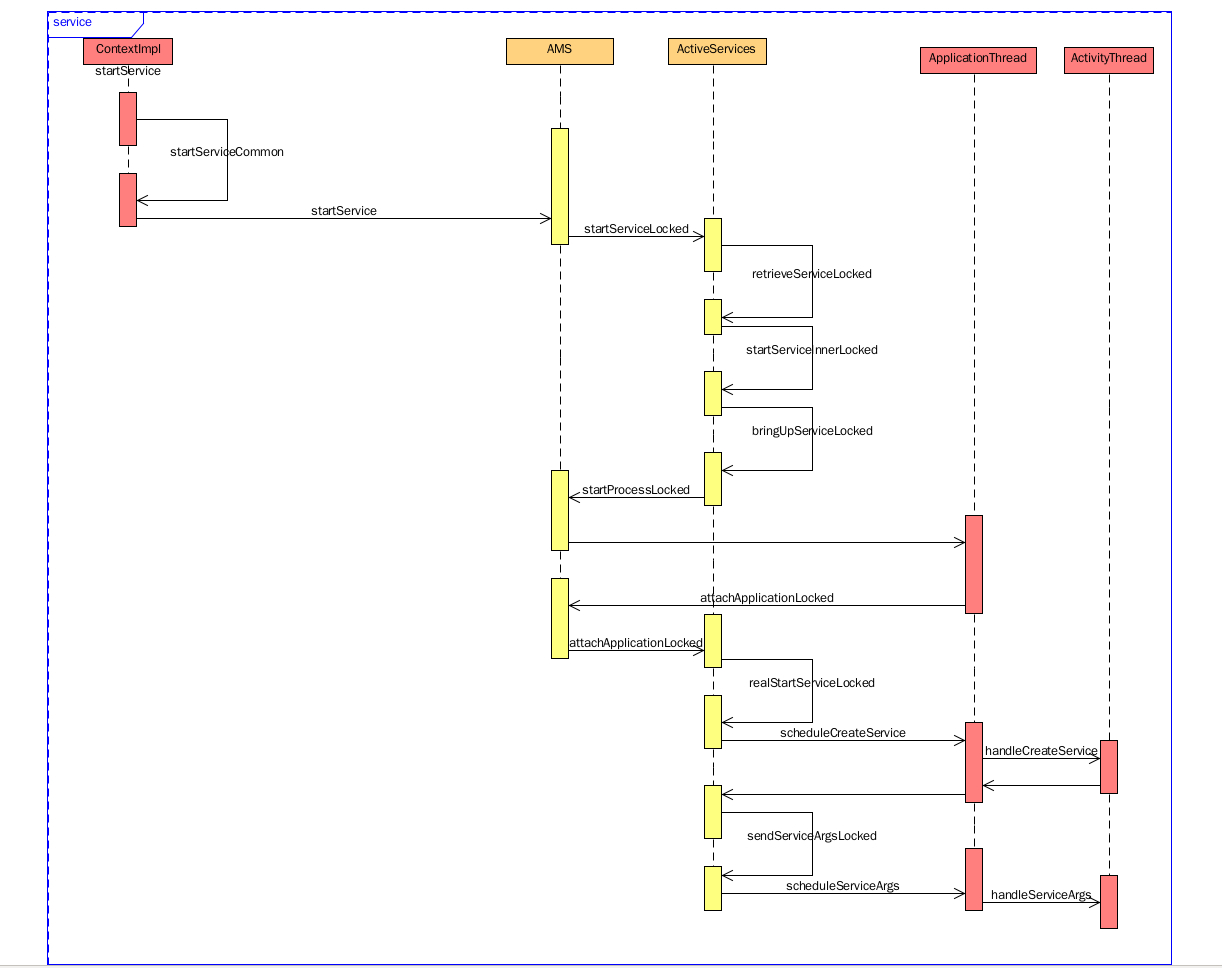
1. **Service結構圖**



1. **Service 的使用**



1. StartService

**startServiceLocked：**

通過IApplicationThread獲取ams中調用者的ProcessRecord。

**retrieveServiceLocked 函數的解析：**

在mServicesByName、mServicesByIntent容器中通過intent來查詢相應的ServiceRecord對象是否已經存在，當不存在的時候通過PackageManager獲取ServiceInfo來創建相應的ServiceRecord 並且放入到兩個容器中。

**bringUpServiceLocked：**

1.當service已經啟動調用sendServiceArgsLocked，調用ServiceRecord中的ProcessRecord. IApplicationThread binder調用到service的宿主進程的scheduleServiceArgs（onStartCommand）

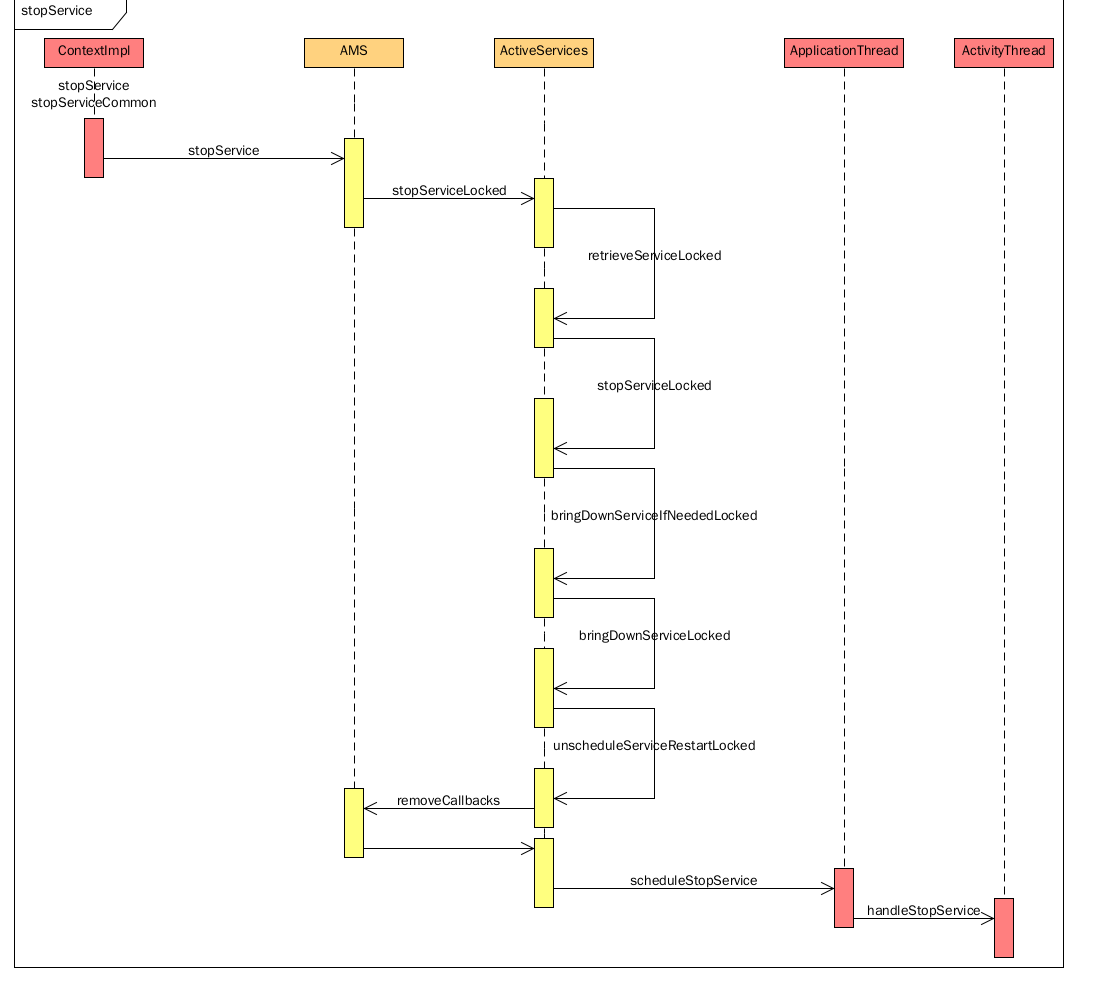
2.當宿主進程已經啟動（ProcessRecord不為空），調用realStartServiceLocked函數，

ServiceRecord加入到ProcessRecord的services容器中，app.thread.scheduleCreateService調用宿主進程的scheduleCreateService（onCreate） 函數來創建service，之後調用sendServiceArgsLocked函數，從而調用ServiceRecord中的ProcessRecord. IApplicationThread binder調用到service的宿主進程的scheduleServiceArgs（onStartCommand）。

3.當宿主進程沒有啟動，調用AMS的startProcessLocked函數來啟動進程，同時

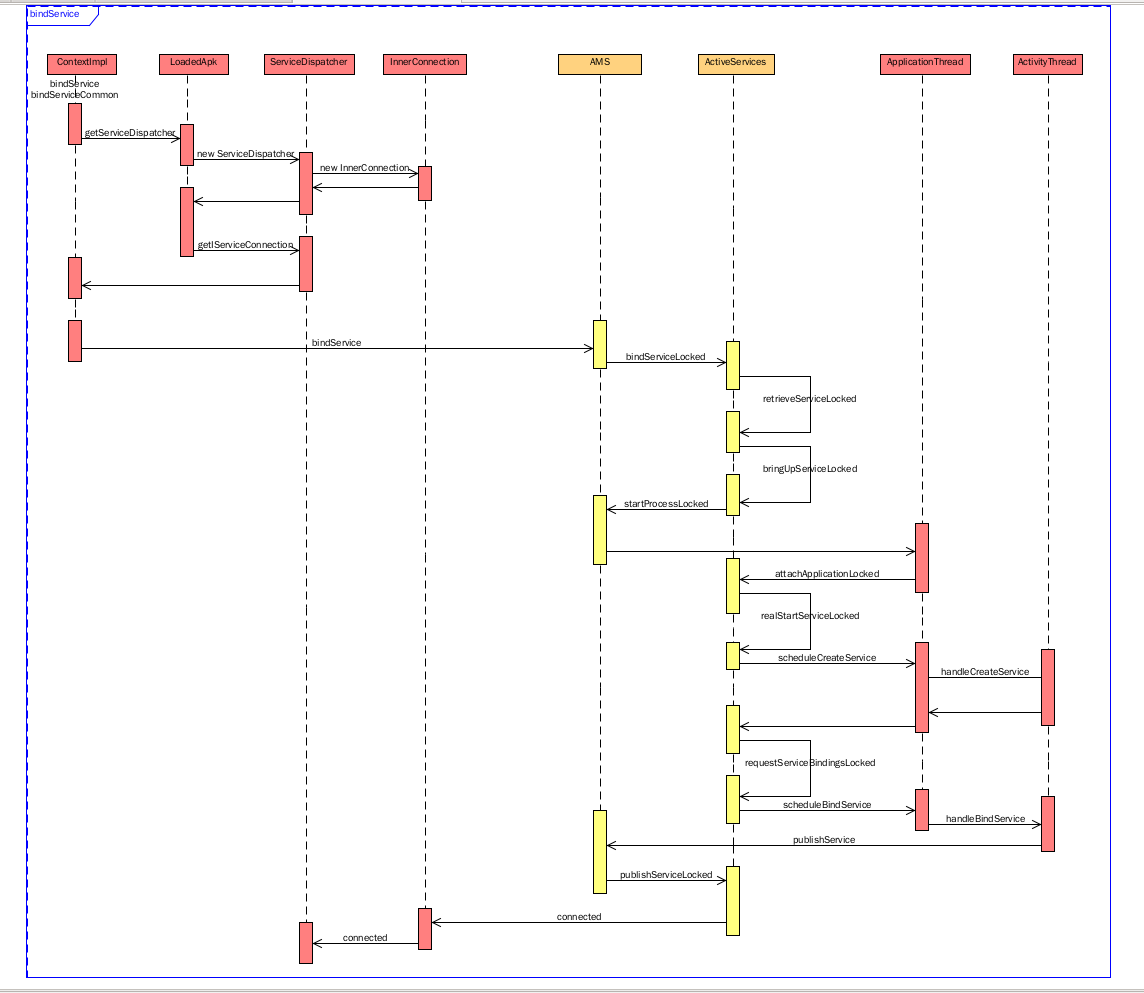
ServiceRecord 加入到mPendingServices容器中，在進程啟動的過程中AMS中的attachApplicationLocked函數會被調用，同時該函數會調用ActiveServices的attachApplicationLocked函數，在該函數中會遍歷mPendingServices容器，調用realStartServiceLocked函數來創建service。

4.service的stopService：



bringDownServiceLocked函數中從mServiceMap、mServicesByIntent、mPendingServices中的mServicesByName容器中移除ServiceRecord，同時從ProcessRecord的services容器中移除ServiceRecord，調用到宿主進程的handleStopService(onDestroy)函數。

1. BindService



ArrayMap<Context, ArrayMap<ServiceConnection, LoadedApk.ServiceDispatcher>> mServices

ServiceDispatcher:包含發起進程的組件Context、主線程的handler、用戶自定義的ServiceConnection，以及繼承與IServiceConnection.Stub 的匿名服務。

bindServiceCommon：將mainThread的匿名applicationThread binder 對象和IServiceConnection匿名對象傳送至AMS。

bindServiceLocked：ServiceRecord . retrieveAppBindingLocked

總結：

1.當service宿主進程die之後，ams會重新啟動該進程。

2.Service 創建後保存的容器：

1. ActiveServices中的mServiceMap

2. ProcessRecord中的services

3.ActivityThread中的mServices

3.startservice可以實現的接口和使用的場景：

onCreate、onStartCommand、onDestroy

可以啟動一個後台線程成為一個服務比如接受broadcast來處理等等。

4.bindService可以自定義進程間的調用的接口：

5.eclipse創建aidl文件：

New->file: ITestService.aidl

package com.example.testfl;

interface ITestService {

void test();

}

ADT 會自動在gen目錄下生成相應的ITestService.java 文件。

注意點：aidl文件所在的包名一定要相同。

6.bindService 通過onBind函數返回的匿名binder對象不會因為client端調用unbindService銷毀，只要進程存在就會一直可用。

7.service是什麼：service就是宿主進程中有固定方法的對象，ams來負責這些對象的生命週期（記錄對象的運行狀態，同時對ActivityThread容器mServices中對象進行操作），而且service對象的方法都是在主線中運行。

8. bindService和startservice組合使用。

9.intentservice 不在主线程运行的service