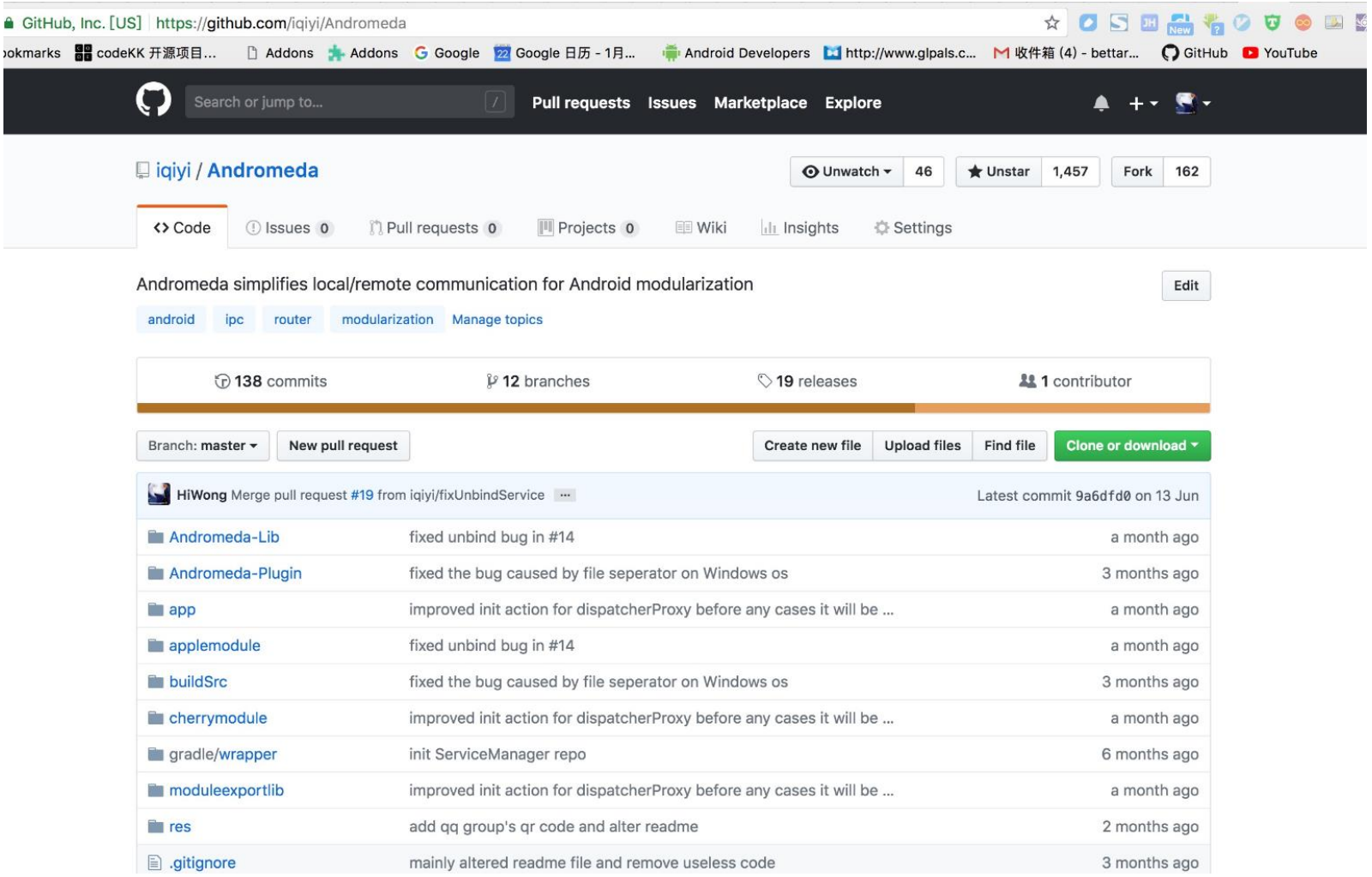
Andromeda:适用于多进程架构的组件通信框架

主讲人：王龙海

时间：35min


GitHub, Inc. [US] https://github.com/iqiyi/Andromeda
Search or jump to...
Pull requests
Issues
Marketplace
Explore
iqiyi / Andromeda
Unwatch 46
Unstar 1,457
Fork 162
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
Andromeda simplifies local/remote communication for Android modularization
Edit
android
ipc
router
modularization
Manage topics
138 commits
12 branches
19 releases
1 contributor
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
HiWong Merge pull request #19 from iqiyi/fixUnbindService
Latest commit 9a6dfd0 on 13 Jun
Andromeda-Lib fixed unbind bug in #14 a month ago
Andromeda-Plugin fixed the bug caused by file seperator on Windows os 3 months ago
app improved init action for dispatcherProxy before any cases it will be ... a month ago
applemodule fixed unbind bug in #14 a month ago
buildSrc fixed the bug caused by file seperator on Windows os 3 months ago
cherrymodule improved init action for dispatcherProxy before any cases it will be ... a month ago
gradle/wrapper init ServiceManager repo 6 months ago
moduleexportlib improved init action for dispatcherProxy before any cases it will be ... a month ago
res add qq group's qr code and alter readme 2 months ago
.gitignore mainly altered readme file and remove useless code 3 months ago

目录

1 背景

2 解决方案

3 InterStellar

4 展望

ARouter

统跳协议

DeepLink Dispatch

DDComponent ForAndroid

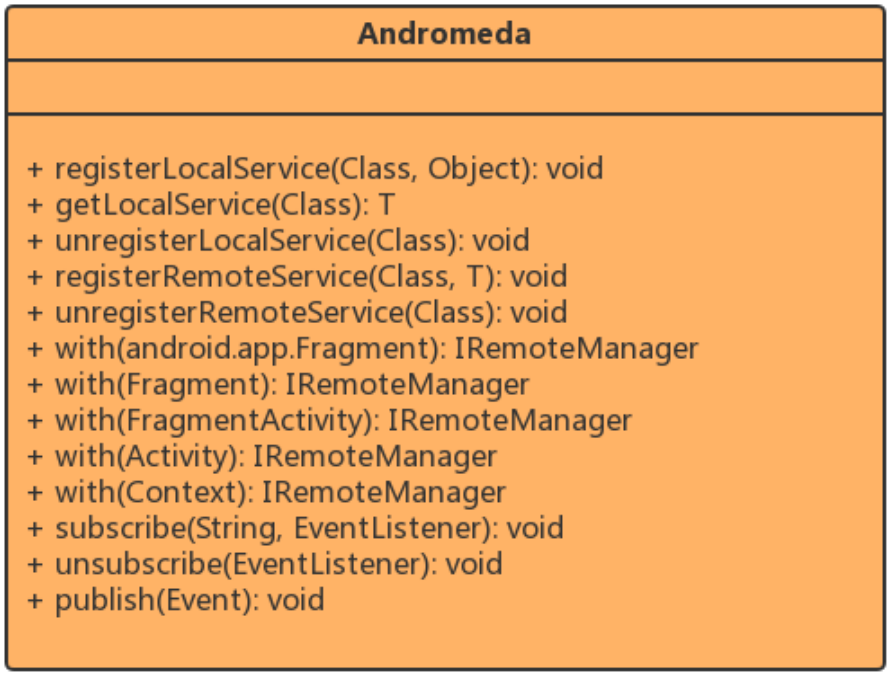
同进程通信

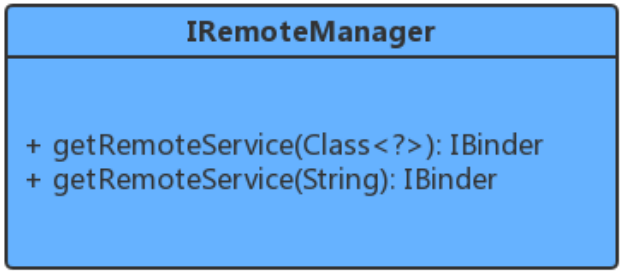
IPC通信

阻塞式调用

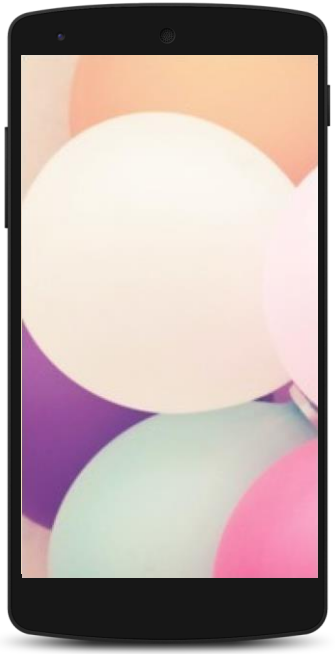
真.事件总线

微信 Event	爱奇艺 ModuleBean	接口+ 数据结构
全局事件	字段膨胀	组件间接口依赖


Andromeda
+ registerLocalService(Class, Object): void
+ getLocalService(Class): T
+ unregisterLocalService(Class): void
+ registerRemoteService(Class, T): void
+ unregisterRemoteService(Class): void
+ with(android.app.Fragment): IRemoteManager
+ with(Fragment): IRemoteManager
+ with(FragmentActivity): IRemoteManager
+ with(Activity): IRemoteManager
+ with(Context): IRemoteManager
+ subscribe(String, EventListener): void
+ unsubscribe(EventListener): void
+ publish(Event): void


IRemoteManager
+ getRemoteService(Class<?>): IBinder
+ getRemoteService(String): IBinder

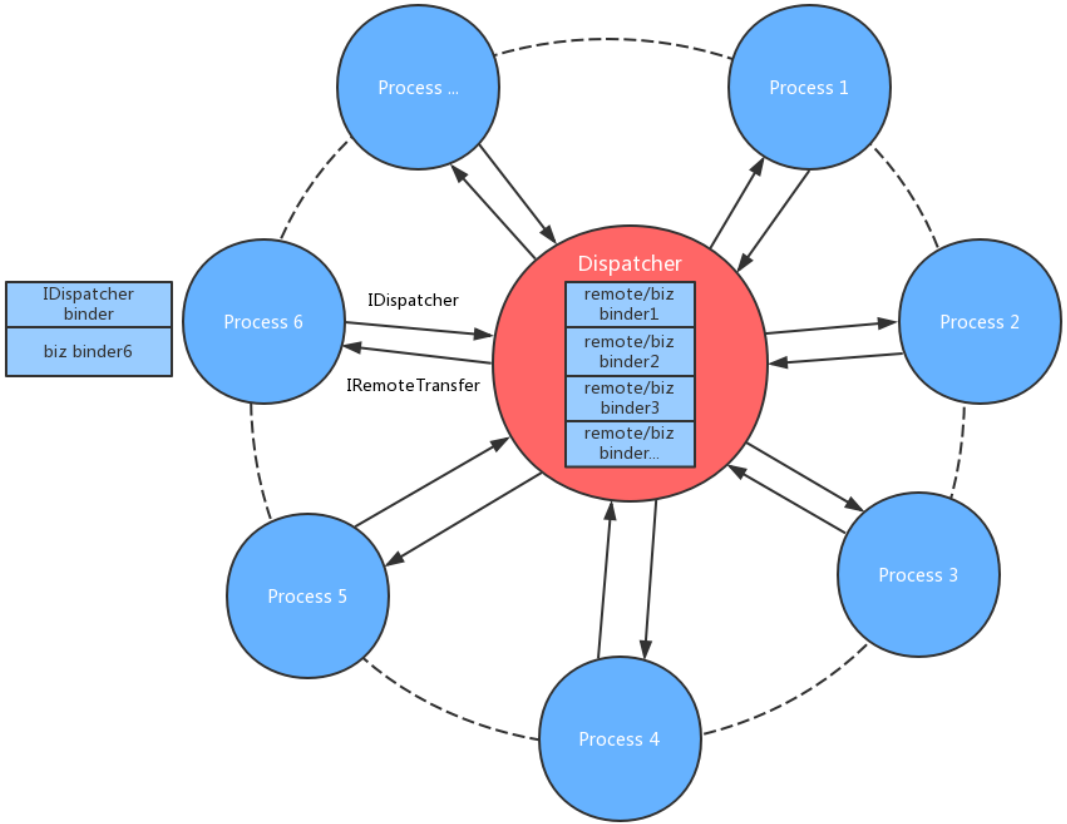
```
private void useBuyAppleService() {
    IBinder binder = Andromeda.with( activity: this).getRemoteService(IBuyApple.class);
    if (binder == null) {
        return;
    }
    IBuyApple buyApple = IBuyApple.Stub.asInterface(binder);
    if (buyApple == null) {
        return;
    }
    try {
        buyApple.buyAppleInShop( userId: 29);
    } catch (RemoteException ex) {
        ex.printStackTrace();
    }
}
```

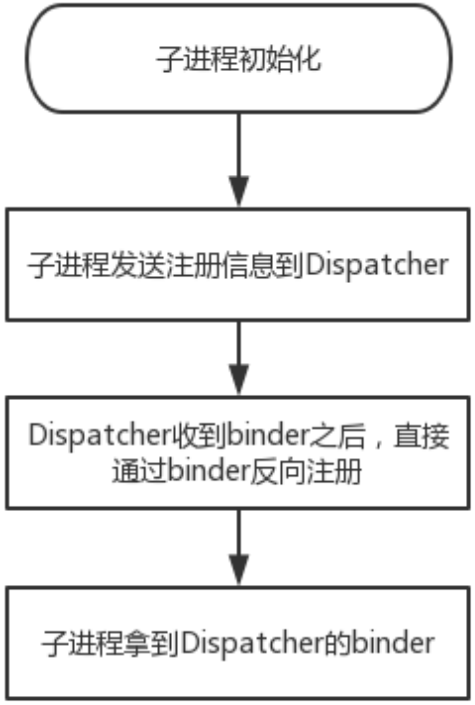


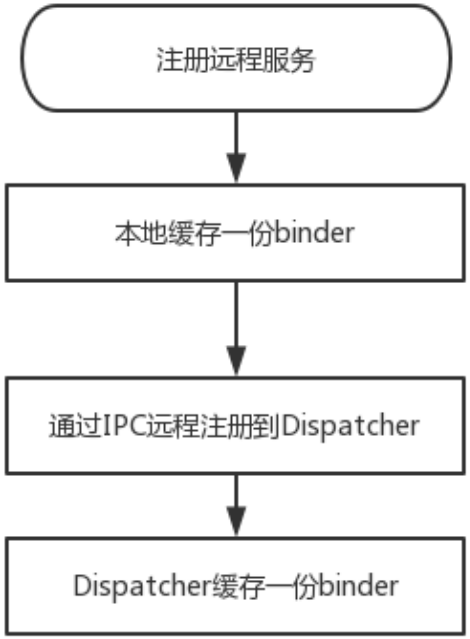
bind Service

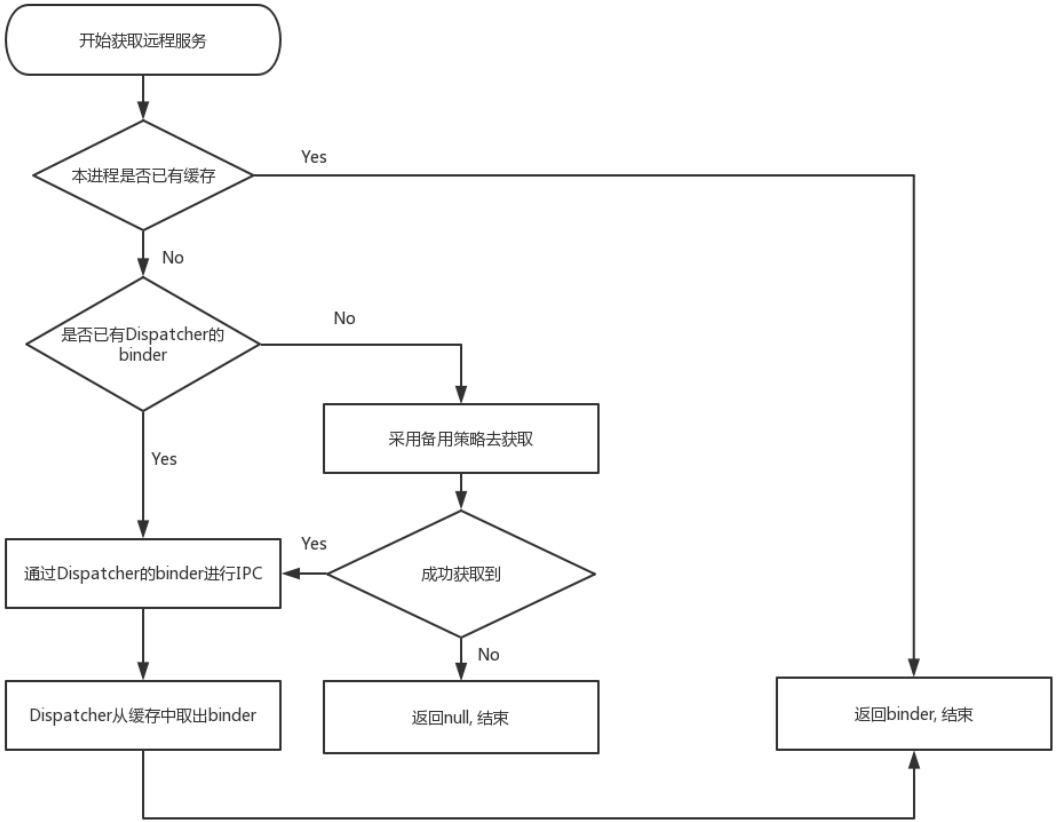
Hermes

Content Provider


Process ...
Process 1
Process 2
Process 3
Process 4
Process 5
Process 6
Dispatcher
remote/biz binder1
remote/biz binder2
remote/biz binder3
remote/biz binder...
IDispatcher
IRemoteTransfer
IDispatcher binder
biz binder6


子进程初始化
子进程发送注册信息到Dispatcher
Dispatcher收到binder之后，直接通过binder反向注册
子进程拿到Dispatcher的binder


注册远程服务
本地缓存一份binder
通过IPC远程注册到Dispatcher
Dispatcher缓存一份binder


开始获取远程服务
本进程是否已有缓存
Yes
No
是否已有Dispatcher的binder
No
采用备用策略去获取
Yes
通过Dispatcher的binder进行IPC
Yes
成功获取到
No
Dispatcher从缓存中取出binder
返回null, 结束
返回binder, 结束

Nexus 6p

8.0.1系统上，Application进行bindService,bind flag为BINDAUTO_CREATE|BIND_ADJUST_WITH_ACTIVITY				
process/oom_adj	打开BindServiceDemo,调用startService和bindService()后	切换到home之后	打开两个其他的App,并且停留在第二	又重新回到BindServiceDemo后
MainActivity	0(fore,top-activity)	3(prev,previous)	5(cch,cch-act)	0(fore, top-activity)
CherryService	1(vis,service)	4(prev,service)	6(cch,cch-client-act)	6(fore, service)
AppleService	10(svc,started-services)	15(svc,started-services)	18(cch+2, cch-empty)	18(cch+2,cch-empty)

8.0.1系统上，Application进行startService和bindService,bind flag为BINDAUTO_CREATE|BIND_IMPORTANT				
process/oom_adj	打开BindServiceDemo,调用startService和bindService()后	切换到home之后	打开两个其他的App,并且停留在第二	又重新回到BindServiceDemo后
MainActivity	0(fore,top-activity)	3(prev,previous)	5(cch,cch-act)	0(fore, top-activity)
CherryService	1(fore,service)	4(prev,service)	6(cch,cch-client-act)	6(fore, service)
AppleService	11(svc,started-services)/10(有一次测到是10)	13(svc,started-services)	16(svcb, cch-empty)	16(svcb,cch-empty)

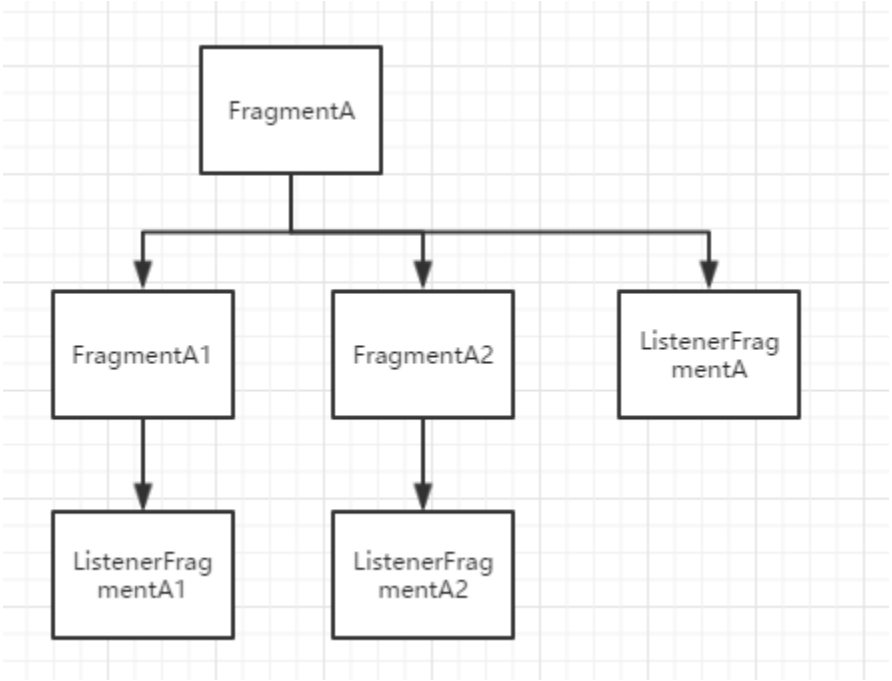
华为Mate 9

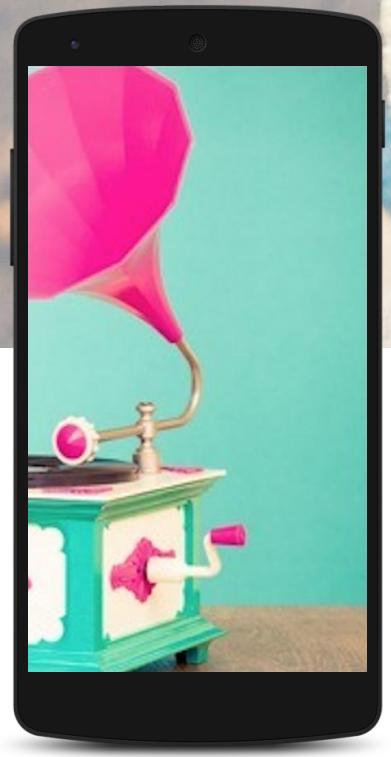
7.1.0系统上，Activity进行startService和bindService,bind flag为BINDAUTO_CREATE|BIND_IMPORTANT				
process/oom_adj	打开BindServiceDemo,调用startService和bindService()后	切换到home之后	打开两个其他的App,并且停留在第二个App时	又重新回到BindServiceDemo后
MainActivity	0(fore,top-activity)	1(prev,previous)	3(cch,cch-act)	0(fore, top-activity)
CherryService	1(fore,service)	2(prev,service)	4(cch,cch-client-act)	3(fore, service)
AppleService	13(svc,started-services)	24(svc,started-services)	26(svc, cch-empty)	27(svcb,cch-empty)
7.1.0系统上，Application进行startService和bindService,bind flag为BINDAUTO_CREATE|BIND_IMPORTANT				
process/oom_adj	打开BindServiceDemo,调用startService和bindService()后	切换到home之后	打开两个其他的App,并且停留在第二个App时	又重新回到BindServiceDemo后
MainActivity	0(fore,top-activity)	1(prev,previous)	3(cch,cch-act)	0(fore, top-activity)
CherryService	1(fore,service)	2(prev,service)	4(cch,cch-client-act)	4(fore, service)
AppleService	15(svc,started-services)	24(svc,started-services)	26(svc, cch-empty)	28(svcb,cch-empty)

一加3t

7.1.0系统上，Activity进行startService和bindService,bind flag为BINDAUTO_CREATE|BIND_IMPORTANT				
process/oom_adj	打开BindServiceDemo,调用startService和bindService()后	切换到home之后	打开两个其他的App,并且停留在第二个App时	又重新回到BindServiceDemo后
MainActivity	0(fore,top-activity)	1(prev,previous)	3(cch,cch-act)	0(fore, top-activity)
CherryService	1(fore,service)	2(prev,service)	4(cch,cch-client-act)	4(fore, service)
AppleService	24(svc,started-services)	26(svc,started-services)	28(svc, cch-empty)	29(svcb,cch-empty)

7.1.0系统上，Application进行startService和bindService,bind flag为BINDAUTO_CREATE|BIND_IMPORTANT				
process/oom_adj	打开BindServiceDemo,调用startService和bindService()后	切换到home之后	打开两个其他的App,并且停留在第二个App时	又重新回到BindServiceDemo后
MainActivity	0(fore,top-activity)	1(prev,previous)	3(cch,cch-act)	0(fore, top-activity)
CherryService	1(fore,service)	3(prev,service)	5(cch,cch-client-act)	8(fore, service)
AppleService	24(svc,started-services)	13(svc,started-services)	28(svcb, cch-empty)	16(svcb,cch-empty)


FragmentA
FragmentA1
FragmentA2
ListenerFragmentA
ListenerFragmentA1
ListenerFragmentA2



15

Extension

Manifest

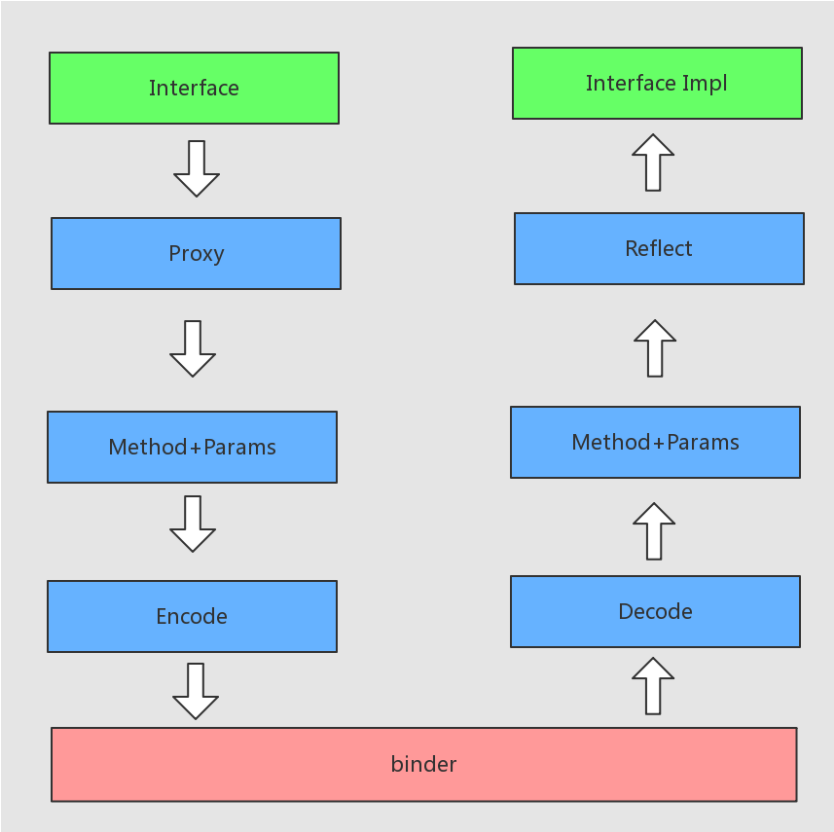
Stub

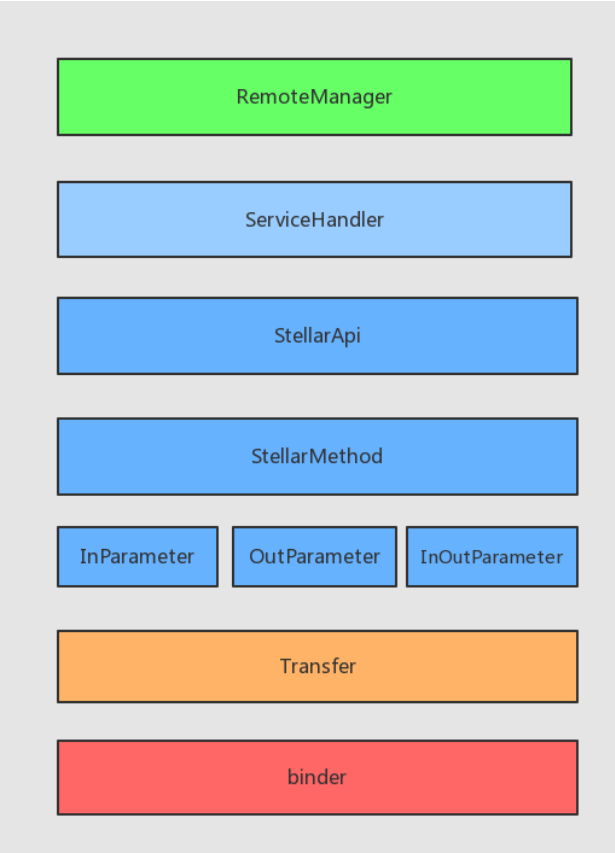
	Convenience	IPC efficiency	IPC	IPC EventBus	IPC Callback
Andromeda	good	high	Yes	Yes	Yes
DDComponentForAndroid	normal	--	No	No	No
ModularizationArchitecture	normal	low	Yes	No	No

```java
public interface IAppleService {

    int getApple(int money);

    float getAppleCalories(int appleNum);

}
```

```java
public class AppleService implements IAppleService {

    @Override
    public int getApple(int money) {
        return money / 2;
    }

    @Override
    public float getAppleCalories(int appleNum) {
        return appleNum * 5;
    }
}
```

```
IAppleService appleService = InterStellar.with( fragmentActivity: BananaActivity.this)
        .getRemoteService(IAppleService.class);
if (appleService != null) {
    int appleNum = appleService.getApple( money: 120);
    Logger.d( msg: "appleNum:" + appleNum);
    float calories = appleService.getAppleCalories( appleNum: 30);
    Logger.d( msg: "calories:" + calories);
}
```


Interface
Proxy
Method+Params
Encode
binder
Interface Impl
Reflect
Method+Params
Decode


RemoteManager
ServiceHandler
StellarApi
StellarMethod
InParameter
OutParameter
InOutParameter
Transfer
binder

1.0 功能完全

2.0 Fragment

3.0 arch-components

4.0 混合

Andromeda交流群

扫一扫二维码，加入群聊。

Thank you for your time!
Questions?