



(12) 发明专利申请

(10) 申请公布号 CN 104580098 A

(43) 申请公布日 2015. 04. 29

(21) 申请号 201310499615. 9

(22) 申请日 2013. 10. 22

(71) 申请人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼资本大厦一座四
层 847 号邮箱

(72) 发明人 张志鹏 钱霄

(74) 专利代理机构 北京安信方达知识产权代理
有限公司 11262

代理人 解婷婷 栗若木

(51) Int. Cl.

H04L 29/06(2006. 01)

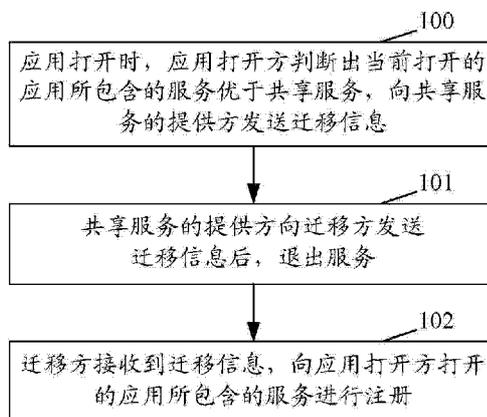
权利要求书2页 说明书7页 附图1页

(54) 发明名称

一种服务共享方法及装置

(57) 摘要

一种服务共享方法及装置,包括应用打开时,应用打开方判断出当前打开的应用所包含的服务优于当前共享服务,向当前共享服务的提供方发送迁移信息;当前共享服务的提供方向迁移方发送迁移信息后,退出服务,迁移方接收到迁移信息,向应用打开方打开的应用所包含的服务进行注册。通过本发明方法,无需在应用中安装独立的服务包,同时,对于使用服务的多个应用来讲,只需要创建一个当前最优的共享服务实例,从而节约电量与流量。



1. 一种服务共享方法,其特征在于,包括:

应用打开时,应用打开方判断出当前打开的应用所包含的服务优于当前共享服务,向当前共享服务的提供方发送迁移信息;

所述当前共享服务的提供方向迁移方发送迁移信息后,退出服务;

所述迁移方接收到迁移信息,向所述应用打开方打开的应用所包含的服务进行注册。

2. 根据权利要求 1 所述的服务共享方法,其特征在于,所述向当前共享服务的提供方发送迁移信息包括:

所述应用打开方从全局配置信息中查询所述当前共享服务的信息,如果所述当前打开的应用所包含的服务的级别高于查询到的信息显示的当前共享服务的级别,则所述应用打开方向当前共享服务的提供方发送迁移信息;

所述应用打开方修改所述全局配置以指向所述当前打开的应用所包含的服务。

3. 根据权利要求 2 所述的服务共享方法,其特征在于,如果判断出所述当前打开的应用所包含的服务的级别低于所述当前共享服务的级别,该方法还包括:

所述应用打开方与所述当前共享服务的提供方之间进行验证,并在双方验证均通过后,所述应用打开方向所述当前共享服务进行注册,结束本流程。

4. 根据权利要求 3 所述的服务共享方法,其特征在于,所述验证未通过,该方法还包括:所述应用打开方修改自身所在应用的全局配置以指向当前打开的应用所包含的服务。

5. 根据权利要求 1 所述的服务共享方法,其特征在于,所述当前共享服务的提供方收到迁移信息后,向迁移方发送迁移信息之前,该方法还包括:

所述当前共享服务的提供方与所述应用打开方之间进行验证,并在双方验证均通过后,向所述迁移方发送迁移信息;

如果验证未通过,结束本流程。

6. 根据权利要求 5 所述的服务共享方法,其特征在于,所述迁移方包括已注册在所述当前共享服务的所有应用。

7. 根据权利要求 2 所述的服务共享方法,其特征在于,所述迁移方向应用打开方打开的应用所包含的服务进行注册包括:

所述迁移方根据所述全局配置信息中的共享服务信息,向所述应用打开方打开的应用所包含的服务进行注册。

8. 根据权利要求 1、3 或 7 所述的服务共享方法,其特征在于,所述注册包括:所述迁移方向所述应用打开方发送注册信息;

所述应用打开方利用注册信息获得应用 ID;

所述应用打开方为发起注册的所述迁移方建立应用 ID,绑定该应用 ID 与应用回调钩子;

所述应用打开方绑定所述注册信息与迁移回调钩子。

9. 根据权利要求 1 所述的服务共享方法,其特征在于,所述当前共享服务的应用被卸载或由于其他原因无法继续提供服务时,该方法还包括:

所述当前共享服务所在应用清空全局配置中指向自身的配置项,并向当前所有已注册在所述当前共享服务所在的应用发送通知,以重新确定共享服务。

10. 一种服务共享装置,其特征在于,包括两个或两个以上应用单元;

第一应用单元,用于在应用打开、并判断出当前打开的应用所包含的服务优于当前共享服务时,向提供当前共享服务的第二应用单元发送迁移信息,更新共享服务为当前打开的应用所包含的服务;

第二应用单元,用于将接收到的迁移信息发送给已注册在自身的所有应用单元;

已注册在当前共享服务的应用单元,一个或一个以上,均用于接收迁移信息,向第一应用单元的应用所包含的服务进行注册。

11. 根据权利要求 10 所述的服务共享装置,其特征在于,

所述第一应用单元,还用于在判断出当前打开的应用所包含的服务比当前共享服务的级别低时,与所述第二应用单元之间进行验证,并在双方验证均通过后,向所述当前共享服务进行注册;

所述第二应用单元具体用于,与所述第一应用单元之间进行验证,并在双方验证均通过后,向所有所述已注册在自身的应用单元发送迁移信息,之后退出服务。

12. 根据权利要求 11 所述的服务共享方法,其特征在于,

所述已注册在当前共享服务的应用单元,具体用于向所述第一应用单元的应用所包含的服务发送注册信息;

所述第一应用单元,还用于利用注册信息获得应用 ID;为已注册在当前共享服务的应用单元建立应用 ID;绑定所述应用 ID 与应用回调钩子;绑定注册信息与迁移回调钩子。

13. 根据权利要求 10 ~ 12 任一项所述的服务共享装置,其特征在于,所述应用单元包括服务模块、代理模块和应用模块,其中,

服务模块,用于在应用启动后作为服务的提供方;接受来自代理单元的注册;在收到迁移信息并确认迁移后,向所有已注册在当前共享服务的应用的代理模块发送迁移信息;退出服务并释放资源;

代理模块,用于启动服务注册过程;在判断出当前打开的应用所包含的服务优于当前共享服务时,向提供当前共享服务所在应用单元的服务模块发送迁移信息;更新共享服务为当前打开的应用所包含的服务;更新共享服务为当前打开的应用所包含的服务;收到迁移信息,重新启动服务注册过程;

应用模块,用于使用共享服务。

一种服务共享方法及装置

技术领域

[0001] 本申请涉及移动互联网领域,尤其涉及一种服务共享方法及装置。

背景技术

[0002] 在移动系统如常用的安卓(Android)系统中,目前,为应用提供服务的方法有两种:

[0003] 一种是,将服务实现于单独的 Android 安装包(Apk,Android Package)中,用户通过安装 Apk 包,在需求服务时,指定服务所在 Apk 包的包名与服务名以连接上服务。该方法要求用户独立安装服务所在 Apk 包,容易被用户拒绝。如果用户拒绝安装,则客户端将无法使用服务提供的功能。因此,若采用该方法,还需要具有服务缺失的备选方案。如目前的支付宝插件即采取此种方案,插件使用者须进行判断,若用户设备中已经安装支付宝插件,则通过支付宝插件支付,若用户设备中没有安装,则通过 Wap 页面的方式进行支付。

[0004] 另一种是,在所有应用中均设置服务单元,用于各应用间通过创建自身的独立服务实例来提供服务。该方法需要在所有的应用中都设置服务单元,并且,对应同一服务,所有应用应该设置相同的服务名,当有应用请求服务时,由系统从提供同名服务的提供者中随机选取一个进行服务。显然,不同应用很可能会创建不同的服务实例,导致用户设备中创建多个服务实例,造成电量和流量的浪费。

发明内容

[0005] 为了解决上述技术问题,本申请提供了一种服务共享方法及装置,无需在应用中安装独立的服务包,同时对于使用服务的多个应用只需要创建一个共享服务实例,从而节约电量与流量。

[0006] 为了达到本申请目的,本申请提供一种服务共享方法,包括:

[0007] 应用打开时,应用打开方判断出当前打开的应用所包含的服务优于当前共享服务,向当前共享服务的提供方发送迁移信息;

[0008] 所述当前共享服务的提供方向迁移方发送迁移信息后,退出服务;

[0009] 所述迁移方接收到迁移信息,向所述应用打开方打开的应用所包含的服务进行注册。

[0010] 所述向当前共享服务的提供方发送迁移信息包括:

[0011] 所述应用打开方从全局配置信息中查询所述当前共享服务的信息,如果所述当前打开的应用所包含的服务的级别高于查询到的信息显示在当前共享服务的级别,则所述应用打开方向当前共享服务的提供方发送迁移信息;

[0012] 所述应用打开方修改所述全局配置以指向所述当前打开的应用所包含的服务。

[0013] 如果判断出所述当前打开的应用所包含的服务的级别低于所述当前共享服务的级别,该方法还包括:

[0014] 所述应用打开方与所述当前共享服务的提供方之间进行验证,并在双方验证均通

过后,所述应用打开方向所述当前共享服务进行注册,结束本流程。

[0015] 所述验证未通过,该方法还包括:所述应用打开方修改自身所在应用的全局配置以指向当前打开的应用所包含的服务。

[0016] 所述当前共享服务的提供方收到迁移信息后,向迁移方发送迁移信息之前,该方法还包括:

[0017] 所述当前共享服务的提供方与所述应用打开方之间进行验证。并在双方验证均通过后,向所述迁移方发送迁移信息;

[0018] 如果验证未通过,结束本流程。

[0019] 所述迁移方包括已注册在所述当前共享服务的所有应用。

[0020] 所述迁移方向应用打开方打开的应用所包含的服务进行注册包括:

[0021] 所述迁移方根据所述全局配置信息中的共享服务信息,向所述应用打开方打开的应用所包含的服务进行注册。

[0022] 所述注册包括:所述迁移方向所述应用打开方发送注册信息;

[0023] 所述应用打开方利用注册信息获得应用 ID;

[0024] 所述应用打开方为发起注册的所述迁移方建立应用 ID,绑定该应用 ID 与应用回调钩子;

[0025] 所述应用打开方绑定所述注册信息与迁移回调钩子。

[0026] 所述当前共享服务的应用被卸载或由于其他原因无法继续提供服务时,该方法还包括:

[0027] 所述当前共享服务所在应用清空全局配置中指向自身的配置项,并向当前所有已注册在所述当前共享服务所在的应用发送通知,以重新确定共享服务。

[0028] 本申请还提供一种服务共享装置,包括两个或两个以上应用单元;

[0029] 第一应用单元,用于在应用打开、并判断出当前打开的应用所包含的服务优于当前共享服务时,向提供当前共享服务的第二应用单元发送迁移信息,更新共享服务为当前打开的应用所包含的服务;

[0030] 第二应用单元,用于将接收到的迁移信息发送给已注册在自身的所有应用单元;

[0031] 已注册在当前共享服务的应用单元,一个或一个以上,均用于接收迁移信息,向第一应用单元的应用所包含的服务进行注册。

[0032] 所述第一应用单元,还用于在判断出当前打开的应用所包含的服务比当前共享服务的级别低时,与所述第二应用单元之间进行验证,并在双方验证均通过后,向所述当前共享服务进行注册;

[0033] 所述第二应用单元具体用于,与所述第一应用单元之间进行验证,并在双方验证均通过后,向所有所述已注册在自身的应用单元发送迁移信息,之后退出服务。

[0034] 所述已注册在当前共享服务的应用单元,具体用于向所述第一应用单元的应用所包含的服务发送注册信息;

[0035] 所述第一应用单元,还用于利用注册信息获得应用 ID;为已注册在当前共享服务的应用单元建立应用 ID;绑定所述应用 ID 与应用回调钩子;绑定注册信息与迁移回调钩子。

[0036] 所述应用单元包括服务模块、代理模块和应用模块,其中,

[0037] 服务模块,用于在应用启动后作为服务的提供方;接受来自代理单元的注册;在收到迁移信息并确认迁移后,向所有已注册在当前共享服务的应用的代理模块发送迁移信息;退出服务并释放资源;

[0038] 代理模块,用于启动服务注册过程;在判断出当前打开的应用所包含的服务优于当前共享服务时,向提供当前共享服务所在应用单元的服务模块发送迁移信息;更新共享服务为当前打开的应用所包含的服务;更新共享服务为当前打开的应用所包含的服务;收到迁移信息,重新启动服务注册过程;

[0039] 应用模块,用于使用共享服务。

[0040] 本申请提供的方案包括,应用打开时,应用打开方判断出当前打开的应用所包含的服务优于当前共享服务,向当前共享服务的提供方发送迁移信息;当前共享服务的提供方向迁移方发送迁移信息后,退出服务,迁移方接收到迁移信息,向应用打开方打开的应用所包含的服务进行注册。通过本发明方法,无需在应用中安装独立的服务包,同时,对于使用服务的多个应用来讲,只需要创建一个当前最优的共享服务实例,从而节约电量与流量。

附图说明

[0041] 附图用来提供对本申请技术方案的进一步理解,并且构成说明书的一部分,与本申请的实施例一起用于解释本申请的技术方案,并不构成对本申请技术方案的限制。

[0042] 图 1 为本申请服务共享方法的流程图;

[0043] 图 2 为本申请服务共享装置的组成结构示意图。

具体实施方式

[0044] 为使本申请的目的、技术方案和优点更加清楚明白,下文中将结合附图对本申请的实施例进行详细说明。需要说明的是,在不冲突的情况下,本申请中的实施例及实施例中的特征可以相互任意组合。

[0045] 在本申请一个典型的配置中,计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0046] 内存可能包括计算机可读介质中的非永久性存储器,随机存取存储器(RAM)和/或非易失性内存等形式,如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

[0047] 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括,但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带,磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质,可用于存储可以被计算设备访问的信息。按照本文中的界定,计算机可读介质不包括非暂存电脑可读媒体(transitory media),如调制的数据信号和载波。

[0048] 在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中

执行。并且,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤。

[0049] 图 1 为本申请服务共享方法的流程图,如图 1 所示,包括以下步骤:

[0050] 步骤 100:应用打开时,应用打开方判断出当前打开的应用所包含的服务优于当前共享服务,向当前共享服务的提供方发送迁移信息。其中,当前共享服务就是当前被多个应用所共享的服务实例。

[0051] 本步骤中,判断当前打开的应用所包含的服务是否优于当前共享服务包括:应用打开方从全局配置信息中查询当前共享服务的信息,如果当前打开的应用所包含的服务的级别高于查询到的信息显示在当前共享服务的级别,比如当前打开的应用所包含的服务的版本号更高等,则应用打开方向当前共享服务的提供方发送迁移信息;同时,应用打开方修改应用的全局配置以指向当前打开的应用所包含的服务,从而更新了共享服务,实现了一次服务自选举过程。

[0052] 本步骤还包括:如果判断出当前打开的应用所包含的服务比当前共享服务的级别低,则应用打开方与当前共享服务的提供方之间进行验证,并在双方验证均通过后,应用打开方向当前共享服务进行注册,结束本流程;如果验证未通过即非法,则应用打开方修改自身所在应用的全局配置以指向当前打开的应用所包含的服务。其中,验证的具体实现属于本领域技术人员的常用技术手段,实现方式也很多如验证身份等,这里不再赘述。

[0053] 本步骤就是一个服务自选举过程,而且都是动态的,实现了根据实际应用情况,选择出最优的服务。

[0054] 步骤 101:当前共享服务的提供方向迁移方发送迁移信息后,退出服务。

[0055] 本步骤中,迁移方为已注册在当前共享服务的所有应用,包括当前共享服务所在的应用。迁移信息就是一个是预先预定的符号,比如“101”等,用于表示进行共享服务迁移。

[0056] 在当前共享服务的提供方收到迁移信息后,本步骤之前,还包括:

[0057] 当前共享服务的提供方与迁移信息的发送方即应用打开方之间进行验证,并在双方验证均通过后,向迁移方发送迁移信息,之后退出服务;如果验证未通过,则当前共享服务的提供方忽略接收到的迁移信息。其中,验证的具体实现属于本领域技术人员的常用技术手段,实现方式也很多如验证身份等,这里不再赘述。

[0058] 步骤 102:迁移方接收到迁移信息,向应用打开方打开的应用所包含的服务进行注册。具体地,迁移方接收到迁移信息,根据全局配置信息中的共享服务信息,向更新后的服务即应用打开方打开的应用所包含的服务进行注册。

[0059] 其中,注册具体包括:

[0060] 迁移方向更新后的共享服务的提供方即步骤 100 中的应用打开方发送注册信息,注册信息包含有迁移方设备标识(ID)和应用名称;

[0061] 更新后的共享服务的提供方利用注册信息获得应用 ID;

[0062] 更新后的共享服务的提供方为注册的迁移方建立应用 ID,并绑定该应用 ID 与应用回调钩子。其中,应用回调钩子由迁移方的应用提供,用于业务事件发生时回调,如有新的消息送达时,这样,可以通过调用应用回调钩子,将服务器的服务内容提供给对应的应用;

[0063] 更新后的共享服务的提供方绑定注册信息与迁移回调钩子。其中,迁移回调钩子

由迁移方设备提供,用于当有迁移事件发生时,对应用而言透明的切换共享服务的提供方。这样,在共享服务的提供方发生改变时,通过调用迁移回调钩子,通知注册在共享服务的应用迁移到改变后的服务上。

[0064] 其中,这些钩子如应用回调钩子、或迁移回调钩子,主要是在消息到达时调用通知应用有消息到达,以及服务提供者收到迁移消息时通知连接到该共享服务的提供方的应用进行迁移即寻找新的共享服务的提供方。其具体实现属于本领域技术人员的惯用技术手段,这里不再赘述。

[0065] 根据系统与使用场景的不同,上述钩子可以是特定的消息,比如 Android 系统中通过信使 (Messenger) 发送的消息 (Message)、延迟发送的系统事件,比如 Android 系统中的未决意向 (PendingIntent) 等。

[0066] 在当前共享服务的应用被卸载或由于其他原因无法继续提供服务时,本发明方法还包括:该应用清空全局配置中指向自身的配置项,并向当前所有已注册在被卸载或其他原因无法继续提供服务的服务所在的应用发送通知,以通知重新开始服务自选举过程即重新选择合适的服务。

[0067] 通过本发明方法,无需在应用中安装独立的服务包,同时,对于使用服务的多个应用来讲,只需要创建一个当前最优的共享服务实例,从而节约电量与流量。

[0068] 图 2 为本申请服务共享装置的组成结构示意图,如图 2 所示,包括两个或两个以上应用单元,其中,

[0069] 第一应用单元,用于在应用打开、并判断出当前打开的应用所包含的服务优于当前共享服务时,向提供当前共享服务的第二应用单元发送迁移信息,更新共享服务为当前打开的应用所包含的服务;

[0070] 第二应用单元,用于将接收到的迁移信息发送给已注册在自身的所有应用单元;

[0071] 第三应用单元~第 N 应用单元,为已注册在当前共享服务的应用单元,均用于接收迁移信息,向第一应用单元的应用所包含的服务进行注册。

[0072] 进一步地,

[0073] 第一应用单元,还用于在判断出当前打开的应用所包含的服务比当前共享服务的级别低时,与第二应用单元之间进行验证,并在双方验证均通过后,向当前共享服务进行注册;

[0074] 第二应用单元具体用于,与第一应用单元之间进行验证,并在双方验证均通过后,向所有已注册在自身的应用单元发送迁移信息,之后退出服务。

[0075] 进一步地,

[0076] 已注册在当前共享服务的应用单元,具体用于向第一应用单元的应用所包含的服务发送注册信息;

[0077] 第一应用单元,还用于利用注册信息获得应用 ID;为已注册在当前共享服务的应用单元建立应用 ID,并绑定该应用 ID 与应用回调钩子;绑定注册信息与迁移回调钩子。

[0078] 本发明中的每个应用单元都包括服务模块、代理模块和应用模块,其中,

[0079] 服务模块,用于在应用启动后作为服务的提供方;接受来自代理单元的注册;在收到迁移信息并确认迁移后,向所有已注册在当前共享服务的应用的代理模块发送迁移信息;退出服务并释放资源;

[0080] 代理模块,用于启动服务注册过程;在判断出当前打开的应用所包含的服务优于当前共享服务时,向提供当前共享服务所在应用单元的服务模块发送迁移信息;更新共享服务为当前打开的应用所包含的服务;更新共享服务为当前打开的应用所包含的服务;收到迁移信息,重新启动服务注册过程;

[0081] 应用模块,用于使用共享服务。

[0082] 以淘宝券券与电影票客户端为例,即本实施例中包括淘宝券券和电影票两个应用,在本实施例的应用场景中,淘宝券券与电影票均使用了本发明的服务共享机制,淘宝券券使用其接收远程服务器的通知消息,并且使用客户端中的服务定期更新 GPS 位置信息,电影票则使用其接收远程服务器的通知消息。

[0083] 假设用户先打开了电影票客户端,其代理模块启动注册,应用模块提供当有通知消息需要处理时的 PendingIntent (应用回调钩子),代理模块首先通过 Settings.System 类从全局配置表中获取到当前共享服务的提供方的信息,由于此时还没有服务进行过注册,因此,其将电影票中所带服务在全局配置中进行注册,然后代理模块通过 startService 启动服务,并且通过 Messenger 在注册期间与服务器进行通信,除注册应用提供的 PendingIntent 外,代理层模块还注册处理迁移消息的 PendingIntent (迁移回调钩子),当收到迁移消息时会将应用的 PendingIntent 作为参数启动处理迁移消息的 PendingIntent,代理模块在应用模块不知情的情况下更换所使用的服务,也就是说,服务的更新对应应用模块来讲是透明的,最后,电影票客户端使用了其自身所带服务。而且,服务一旦启动则在后台运行,即使此时电影票前台退出,后台服务仍会继续运行,除非用户卸载服务所在应用,或者所有应用都解除注册 (unregister)。

[0084] 假设淘宝券券中的服务版本高于电影票中的服务,当用户又打开了淘宝券券客户端,则其代理模块首先从全局配置表中获取当前共享服务的提供方的信息,发现是电影票所在应用,查询其所带服务版本,发现淘宝券券自身的比当前共享服务的级别高,则修改全局配置,向当前共享服务的提供方(即电影票客户端所包含的服务)发送迁移消息,同时提供自身的认证信息,并使用自身所带服务。电影票客户端所包含的服务在收到淘宝券券的迁移消息后根据其认证信息验证其身份,通过认证后,向已经注册在电影票客户端所包含的服务的所有应用广播迁移消息即将应用对应的 PendingIntent 作为参数启动代理的 PendingIntent,电影票的代理模块在收到迁移消息后,重新进行注册流程,发现全局配置中的服务的提供方是淘宝券券,而其比自己更加合适,因此,向淘宝券券所包含的服务进行注册。最后电影票的后台服务进程退出运行。

[0085] 对于服务后台,淘宝券券的后台服务进程定期唤醒其所在设备,向远程服务器轮询消息通知,同时更新自身的位置信息,假设在一次轮询周期内淘宝券券和电影票的运营人员都发布了各自的促销信息,则后台服务在下次轮询时接收到这些通知消息,并分别发送给淘宝券券客户端与电影票客户端处理,比如通过启动各应用注册时提供的 PendingIntent 发送通知,参数则通过 Intent Data 传递。

[0086] 假设用户在某一时刻卸载了淘宝券券应用,则淘宝券券后台服务将会清除全局配置中的当前服务提供者信息,然后向所有注册了淘宝券券应用所包含的服务的应用的代理模块发布迁移消息,重新进行自选举过程选出合适的服务的提供方,在该场景中,电影票应用的代理模块重新向电影票所含服务进行注册。

[0087] 本申请针对目前常见的移动系统,如 Android,在不增加独立安装 apk、不对操作系统进行特别处理的情况下,引入共享服务机制。本申请的共享服务可在后台进行对多个应用均适用的操作,如获取通知消息等,从而避免了多个应用自身独立操作引入的电量与流量消耗。同时,本申请中的各应用所包含的服务的自选举过程,保证了系统中的共享服务的提供方为当前最合适的提供方,如版本最新等。另外,本申请还支持服务的无缝迁移,即当服务提供者被卸载或者无法继续提供服务时,重新选取合适的其他服务的提供方继续提供服务,而且该过程对于应用是透明的。

[0088] 本领域的技术人员应该明白,上述的本申请实施例所提供的装置的各组成部分,以及方法中的各步骤,它们可以集中在单个的计算装置上,或者分布在多个计算装置所组成的网络上。可选地,它们可以用计算装置可执行的程序代码来实现。从而,可以将它们存储在存储装置中由计算装置来执行,或者将它们分别制作成各个集成电路模块,或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样,本申请不限制于任何特定的硬件和软件结合。

[0089] 虽然本申请所揭露的实施方式如上,但所述的内容仅为便于理解本申请而采用的实施方式,并非用以限定本申请。任何本申请所属领域内的技术人员,在不脱离本申请所揭露的精神和范围的前提下,可以在实施的形式及细节上进行任何的修改与变化,但本申请的专利保护范围,仍须以所附的权利要求书所界定的范围为准。

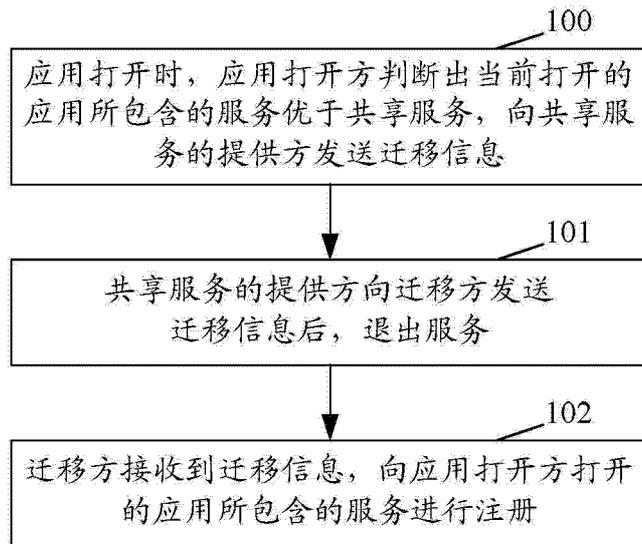


图 1

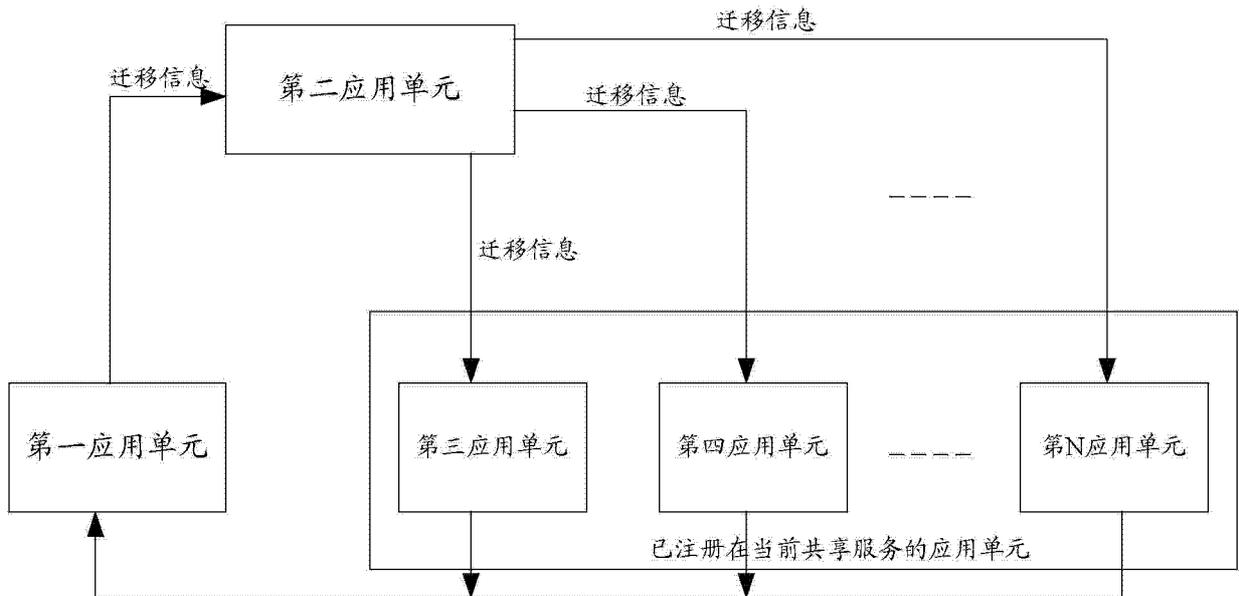


图 2