(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111338710 A
(43)申请公布日 2020.06.26

(21)申请号 202010120683.X

(22)申请日 2020.02.26

(71)申请人 腾讯科技(深圳)有限公司
地址 518064 广东省深圳市南山区高新区科技中一路腾讯大厦35层

(72)发明人 龚健飞

(74)专利代理机构 深圳市智圈知识产权代理事务所(普通合伙) 44351
代理人 韩绍君

(51)Int.Cl.
G06F 9/445(2018.01)

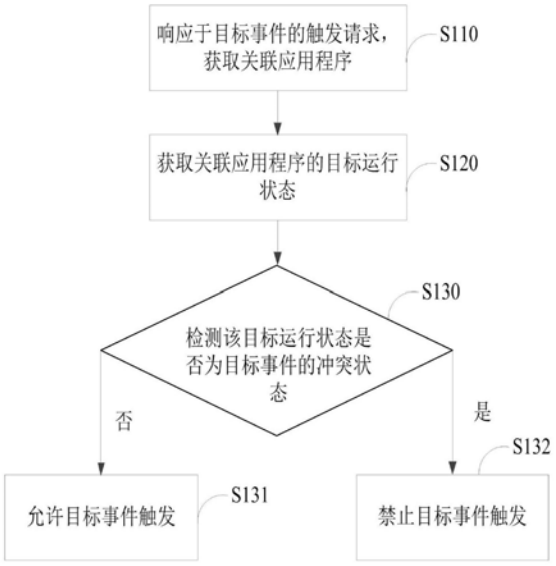
权利要求书2页 说明书19页 附图12页

(54)发明名称

应用程序控制方法、装置、电子设备及存储介质

(57)摘要

本申请实施例公开了一种应用程序控制方法、装置、电子设备及存储介质。所述方法包括：响应于目标事件的触发请求，获取关联应用程序，所述关联应用程序为与发起所述触发请求的应用程序关联的应用程序；获取所述关联应用程序的目标运行状态，所述目标运行状态为所述关联应用程序的当前运行状态中与所述目标事件对应的运行状态；当所述目标运行状态为所述目标事件的冲突状态时，禁止所述目标事件触发。该方法通过获取关联应用程序的目标运行状态的方式，实现了在有目标事件触发请求时，可以根据目标运行状态是否为与目标事件的冲突状态来确定是否允许目标事件触发，进而避免目标事件触发后与关联应用程序之间产生运行冲突。



CN 111338710 A

1.一种应用程序控制方法,其特征在于,所述方法包括:

响应于目标事件的触发请求,获取关联应用程序,所述关联应用程序为与发起所述触发请求的应用程序关联的应用程序;

获取所述关联应用程序的目标运行状态,所述目标运行状态为所述关联应用程序的当前运行状态中与所述目标事件对应的运行状态;

当所述目标运行状态为所述目标事件的冲突状态时,禁止所述目标事件触发。

2.根据权利要求1所述的方法,其特征在于,所述获取所述关联应用程序的目标运行状态,包括:

基于指定的通信接口与所述关联应用程序建立通信通道;

通过所述通信通道,向所述关联应用程序发送第一状态获取请求;

接收所述关联应用程序响应所述第一状态获取请求通过所述通信通道返回的所述目标运行状态。

3.根据权利要求2所述的方法,其特征在于,所述方法还包括:

接收第二状态获取请求;

当所述关联应用程序包括发送所述第二状态获取请求的应用程序时,向发送所述第二状态获取请求的应用程序返回所请求的运行状态。

4.根据权利要求2或3所述的方法,其特征在于,所述指定的通信接口包括内容提供者组件;所述基于指定的通信接口与所述关联应用程序建立通信通道,包括:

基于所述内容提供者组件与所述关联应用程序建立通信通道。

5.根据权利要求1所述的方法,其特征在于,所述响应于目标事件触发请求,获取关联应用程序之前还包括:

获取第一应用程序名单,以及获取第二应用程序名单,其中,所述第一应用程序名单的获取途径和所述第二应用程序名单的获取途径不同;

将所述第一应用程序名单和所述第二应用程序名单均包括的应用程序作为所述关联应用程序并进行存储;

所述获取关联应用程序,包括:

获取所述存储的关联应用程序。

6.根据权利要求5所述的方法,其特征在于,所述获取第一应用程序名单,包括:

从指定服务器下载应用程序名单作为所述第一应用程序名单。

7.根据权利要求6所述的方法,其特征在于,所述从指定服务器下载应用程序名单作为所述第一应用程序名单,包括:

向所述指定服务器发送携带第一特征数据的获取请求,所述第一特征数据为基于历史下载的应用程序名单计算得到的数据;

接收所述指定服务器返回的应用程序名单,其中,基于所述返回的应用程序名单计算得到的第二特征数据,与所述第一特征数据不同;

将所述返回的应用程序名单作为所述第一应用程序名单。

8.根据权利要求5所述的方法,其特征在于,所述获取第二应用程序名单,包括:

获取具有指定通信接口的应用程序;

生成所述第二应用程序名单,所述第二应用程序名单中包括所述具有指定通信接口的

应用程序。

9.根据权利要求1所述的方法，其特征在于，所述获取关联应用程序，包括：

获取参考运行状态，所述参考运行状态为发起所述触发请求的应用程序的当前运行状态中与所述目标事件对应的运行状态；

当所述参考运行状态与所述目标事件对应的运行状态不同时，则获取关联应用程序。

10.根据权利要求1所述的方法，其特征在于，所述目标事件包括显示独立子页面的事件；所述目标运行状态包括前后台运行状态；

所述若所述目标运行状态为所述目标事件的冲突状态，禁止所述目标事件触发，包括：

若所述目标运行状态表征关联应用程序在前台运行，判定所述目标运行状态为所述显示独立子页面的事件的冲突状态，禁止所述显示独立子页面的事件触发。

11.一种应用程序控制方法，其特征在于，所述方法包括：

接收应用程序发送的状态获取请求，所述状态获取请求中携带有表征所述应用程序待触发的目标事件的事件标识；

获取发送所述状态获取请求的应用程序的关联应用程序；

获取所述关联应用程序的目标运行状态，所述目标运行状态为所述关联应用程序的当前运行状态中与所述事件标识表征的目标事件对应的运行状态；

若所述目标运行状态为所述事件标识所表目标征事件的冲突状态，禁止所述应用程序触发所述目标事件。

12.一种应用程序控制装置，其特征在于，所述装置包括：

关联程序获取单元，用于响应于目标事件的触发请求，获取关联应用程序，所述关联应用程序为与发起所述触发请求的应用程序关联的应用程序；

运行状态获取单元，用于获取所述关联应用程序的目标运行状态，所述目标运行状态为所述关联应用程序的当前运行状态中与所述目标事件对应的运行状态；

程序控制单元，用于若所述目标运行状态为所述目标事件的冲突状态，禁止所述目标事件触发。

13.一种应用程序控制装置，其特征在于，所述装置包括：

请求管理单元，用于接收应用程序发送的状态获取请求，所述状态获取请求中携带有表征所述应用程序待触发的目标事件的事件标识；

关联程序获取单元，用于获取发送所述状态获取请求的应用程序的关联应用程序；

运行状态获取单元，用于获取所述关联应用程序的目标运行状态，所述目标运行状态为所述关联应用程序的运行状态中与所述事件标识表征的目标事件对应的运行状态；

程序控制单元，用于若所述目标运行状态为所述事件标识所表目标征事件的冲突状态，禁止所述应用程序触发所述目标事件。

14.一种电子设备，其特征在于，包括处理器以及存储器；一个或多个程序被存储在所述存储器中并被配置为由所述处理器执行以实现权利要求1-11所述的方法。

15.一种计算机可读存储介质，其特征在于，所述计算机可读存储介质中存储有程序代码，其中，在所述程序代码被处理器运行时执行权利要求1-11任一所述的方法。

应用程序控制方法、装置、电子设备及存储介质

技术领域

[0001] 本申请涉及计算机技术领域，更具体地，涉及一种应用程序控制方法、装置、电子设备及存储介质。

背景技术

[0002] 随着软件技术的发展，电子设备中所安装的应用程序的数量越来越多。其中，不同的应用程序在运行过程中可能需要占用一定的资源来完成相关的功能。因此，不同的应用程序在运行过程中可能会出现运行冲突。

发明内容

[0003] 鉴于上述问题，本申请提出了一种应用程序控制方法、装置、电子设备及存储介质，以改善上述问题。

[0004] 第一方面，本申请提供了一种应用程序控制方法，所述方法包括：响应于目标事件的触发请求，获取关联应用程序，所述关联应用程序为与发起所述触发请求的应用程序关联的应用程序。获取所述关联应用程序的目标运行状态，所述目标运行状态为所述关联应用程序的当前运行状态中与所述目标事件对应的运行状态，当所述目标运行状态为所述目标事件的冲突状态时，禁止所述目标事件触发。

[0005] 第二方面，本申请提供了一种应用程序控制方法，所述方法包括：接收应用程序发送的状态获取请求，所述状态获取请求中携带有表征所述应用程序待触发的目标事件的事件标识。获取发送所述状态获取请求的应用程序的关联应用程序。获取所述关联应用程序的目标运行状态，所述目标运行状态为所述关联应用程序的当前运行状态中与所述事件标识表征的目标事件对应的运行状态，当所述目标运行状态为所述事件标识所表目标征事件的冲突状态时，禁止所述应用程序触发所述目标事件。

[0006] 第三方面，本申请提供了一种应用程序控制装置，所述装置包括参考运行状态。关联程序获取单元，用于响应于目标事件的触发请求，获取关联应用程序，所述关联应用程序为与发起所述触发请求的应用程序关联的应用程序。运行状态获取单元，用于获取所述关联应用程序的目标运行状态，所述目标运行状态为所述关联应用程序的当前运行状态中与所述目标事件对应的运行状态。程序控制单元，用于当所述目标运行状态为所述目标事件的冲突状态时，禁止所述目标事件触发。

[0007] 第四方面，本申请提供了一种应用程序控制装置，所述装置包括请求管理单元、关联程序获取单元、运行状态获取单元以及程序控制单元。请求管理单元，用于接收应用程序发送的状态获取请求，所述状态获取请求中携带有表征所述应用程序待触发的目标事件的事件标识。关联程序获取单元，用于获取发送所述状态获取请求的应用程序的关联应用程序。运行状态获取单元，用于获取所述关联应用程序的目标运行状态，所述目标运行状态为所述关联应用程序的运行状态中与所述事件标识表征的目标事件对应的运行状态。程序控制单元，用于当所述目标运行状态为所述事件标识所表目标征事件的冲突状态时，禁止所

述应用程序触发所述目标事件。

[0008] 第五方面，本申请提供了一种电子设备，包括处理器以及存储器；一个或多个程序被存储在所述存储器中并被配置为由所述处理器执行以实现上述的方法。

[0009] 第六方面，本申请提供了一种计算机可读存储介质，所述计算机可读存储介质中存储有程序代码，其中，在所述程序代码被处理器运行时执行上述的方法。

[0010] 本申请提供的一种应用程序控制方法、装置、电子设备及存储介质，通过在响应于目标事件的触发请求时，获取与发起所述触发请求的应用程序关联的关联应用程序，获取所述关联应用程序的当前运行状态中与所述目标事件对应的运行状态作为目标运行状态，进而在所述目标运行状态为所述目标事件的冲突状态的情况下，禁止所述目标事件触发。从而通过获取关联应用程序的目标运行状态的方式，实现了在有目标事件触发请求时，可以根据目标运行状态是否为与目标事件的冲突状态来确定是否允许目标事件触发，进而避免目标事件触发后与关联应用程序之间产生运行冲突。

附图说明

[0011] 为了更清楚地说明本申请实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0012] 图1示出了本申请实施例提中的一种弹窗显示的示意图；

[0013] 图2示出了本申请一实施例提出的一种应用程序控制方法的流程图；

[0014] 图3示出了本申请另一实施例提出的一种应用程序控制方法的流程图；

[0015] 图4示出了本申请再一实施例提出的一种应用程序控制方法的流程图；

[0016] 图5示出了本申请实施例中一种获取关联应用程序的示意图；

[0017] 图6示出了本申请实施例中一种从指定服务器请求关联应用程序名单的示意图；

[0018] 图7示出了本申请实施例中一种从指定服务器请求关联应用程序名单的流程图；

[0019] 图8示出了本申请又一实施例提出的一种应用程序控制方法的流程图；

[0020] 图9示出了本申请实施例提出的应用程序控制方法应用于弹窗显示场景的流程图；

[0021] 图10示出了本申请又一实施例提出的一种应用程序控制方法的流程图；

[0022] 图11示出了本申请又一实施例提出的一种应用程序控制方法的流程图；

[0023] 图12示出了本申请实施例提出的一种应用程序控制装置的结构框图；

[0024] 图13示出了本申请另一实施例提出的一种应用程序控制装置的结构框图；

[0025] 图14示出了本申请再一实施例提出的一种应用程序控制装置的结构框图；

[0026] 图15示出了用于执行根据本申请实施例的应用程序控制方法的另一种电子设备的结构框图；

[0027] 图16示出了本申请实施例的用于保存或者携带实现根据本申请实施例的应用程序控制方法的程序代码的存储单元。

具体实施方式

[0028] 下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

[0029] 为了满足用户的不同需求，电子设备中所安装的应用程序的数量和种类越来越多。例如，为了实现拍照功能，电子设备中可以安装有相机软件，为了实现与其他电子设备用户之前的即时沟通，电子设备中可以安装有即时通信软件，而为了实现对电子设备中的运行产生的垃圾进行清理，电子设备可以安装有垃圾清理软件。

[0030] 发明人在研究中发现，随着软件数量越来越多，不同的应用程序在运行过程中可能产生运行冲突。例如，当前在后台运行的应用程序准备在电子设备的屏幕上触发显示一个弹窗，但是，当前在前台运行的是另一个应用程序，那么在该后台运行的应用程序直接触发弹窗显示的情况下，就会造成显示的弹窗覆盖在处于前台运行的应用程序的界面上，造成所显示的弹窗与前台运行的应用程序出现显示冲突，并且同时还会造成干扰用户对前台运行的应用程序的使用。

[0031] 示例性的，如图1所示，在图1所示的场景中包括有前台运行的应用程序的运行界面10以及弹窗11，其中，弹窗11为后台运行的应用程序触发。在图1所示的这种情况下，可以发现在后台运行的应用程序触发的弹窗11会覆盖在前台运行的应用程序的运行界面10上，造成出现显示冲突。并且，如果该弹窗11配置有页面跳转功能，那么在检测到用户误触控该弹窗11，后电子设备会将当前前台运行的应用程序切换为触发该弹窗11的应用程序，进而给用户造成了不便。

[0032] 而发明人在进一步的研究后发现，造成不同的应用程序在运行过程中可能会产生运行冲突的原因在于应用程序在触发某个事件而执行某个功能前，并不会去获取其他应用程序的运行状态，进而也就无法了解到当前自身所要执行的功能是否会与其他应用程序的运行状态会出现冲突。发明人进一步的还发现，对于相关的应用程序在执行某个功能前，并不会去获取其他应用程序的运行状态的原因由在于，在相关的电子设备中应用程序需要通过权限配置才能使得其他的应用程序获取到自身的运行状态，但是在相关的权限配置过程中，配置路径较为复杂，并且不同类型的电子设备的配置路径以及配置方式并不相同，因此，造成权限配置的成功率并不高，所以应用程序相互之间并不能有效的相互获取到的运行状态。

[0033] 还有，对于一些相互之间具有一定关联的应用程序，如果相互之间出现运行冲突，就可能会造成资源的浪费。其中，具有一定关联的应用程序可以理解为属于同一个公司开发的应用程序，或者可以理解为具有相同的功能的应用程序。

[0034] 因此，为了改善上述问题，本申请实施例提供了一种应用程序控制方法、装置、电子设备及存储介质，通过在响应于目标事件的触发请求时，获取与发起所述触发请求的应用程序关联的关联应用程序，获取所述关联应用程序的当前运行状态中与所述目标事件对应的运行状态作为目标运行状态，进而在所述目标运行状态为所述目标事件的冲突状态的情况下，禁止所述目标事件触发。从而实现了在有目标事件触发请求时，可以根据目标运行状态是否为与目标事件的冲突状态来确定是否允许目标事件触发，进而避免目标事件触发

后与关联应用程序之间产生运行冲突。

[0035] 在对本申请实施例进行进一步详细说明之前，对本申请实施例中涉及的名词和术语进行说明，本申请实施例中涉及的名词和术语适用于如下的解释。

[0036] 事件：事件可以用于描述应用程序要执行的某个功能。例如，当应用程序要执行显示弹窗的功能时，则对应为显示弹窗的事件。当应用程序要执行数据运算功能时，则对应为处理器的处理资源占用事件。当应用程序要执行音频输出功能时，则对应为音频流输出事件。

[0037] 前台运行的应用程序：前台运行的应用程序指存在界面可与用户进行交互的应用程序，如用户在使用即时通信软件进行聊天时，即时通信软件的界面可以用于与进行交互以便用户输入文本信息或者语音信息，在这种情况下，即时通信软件即为前台运行的应用程序。

[0038] 后台运行的应用程序：当应用程序不存在可与用户交互的界面且依然处于运行状态时则为后台运行的应用程序，如用户将当前显示的界面由即时通信软件的界面切换为桌面时，即时通信软件即为后台运行的应用程序。

[0039] SDK：Software Development Kit，软件开发工具包，一般指为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件的开发工具的集合，包括相关文档、范例、工具等。

[0040] 关联应用程序：也可以理解为矩阵应用程序，是指同一公司、或同一部门、或同一开发者开发、或接入同一SDK的应用程序，在功能上存在协助、互补作用。其中，接入同一SDK可以理解为均配置有与同一SDK进行数据通信的接口。在本申请实施例中，可以通过配置应用程序名单来获取关联应用程序。例如，对于应用程序A可以对应配置一个应用程序名单，该应用程序名单中包括有应用程序B、应用程序C以及应用程序D，那么应用程序A通过该应用程序名单就可以检测到应用程序B、应用程序C以及应用程序D为关联应用程序。

[0041] 下面将结合附图具体描述本申请的各实施例。

[0042] 请参阅图2，图2所示为本申请一实施例提出的一种应用程序控制方法的流程图，该方法包括：

[0043] S110：响应于目标事件的触发请求，获取关联应用程序，关联应用程序为与发起触发请求的应用程序关联的应用程序。

[0044] 其中，目标事情为当前待触发的事件，对应的目标事件的触发请求可以理解为当前待触发该目标事件的请求，进而在获取到该触发请求时，就可以确定发起该触发请求的应用程序待触发该目标事件，以执行该目标事件所对应的功能。例如，若应用程序A检测到自身产生了目标事件的触发请求时，就可以响应于目标事件的触发请求，获取与应用程序A关联的关联应用程序。

[0045] 在本实施例中，可以有多种的获取关联应用程序的方式。作为一种方式，可以基于配置的应用程序名单来获取关联应用程序。在这种方式下，可以通过查询该配置的应用程序名单来获取到关联应用程序。再者，作为另外一种方式，可以通过检测是否具有相同的属性来关联应用程序。其中，该属性可以为通信接口标识、安装包签名或者所嵌入的SDK插件的标识等。

[0046] S120：获取关联应用程序的目标运行状态，目标运行状态为关联应用程序的当前

运行状态中与目标事件对应的运行状态。

[0047] 需要说明的是，应用程序在运行过程中会因为执行不同的功能而占用不同的资源。例如，若应用程序要执行与用户交互的功能，那么就需要切换为前台运行的应用程序，进而就会占用显示资源。再例如，若应用程序要执行数据运算的功能，那么就需要利用处理器来完成数据的运算，进而就会占用处理器的处理资源。再例如，若应用程序要执行音频输出的功能，那么就需要音频输出通道来将要输出的音频流传输给音频输出器件(例如，听筒或者扬声器)，进而就会占用音频输出通道资源。可选的，在本实施例中所涉及的资源占用中可以为全部占用，也可以为部分占用。例如，在有2条音频输出通道的情况下，应用程序可以在需要输出音频流时，仅占用其中的1条音频输出通道，也可以同时占用该2条音频输出通道。

[0048] 在本实施例中，对应应用程序的每种资源占用情况可以分别配置一个运行状态，以便对资源占用情况进行标识。可选的，对于显示资源的占用情况可以对应配置的前后台运行状态，其中前后台运行状态表征应用程序是在前台运行或者在后台运行，其中，若表征在前台运行则表征当前占用了显示资源。再者，对于处理器的处理资源的占用情况，可以对应配置处理器占用状态，该处理器占用状态表征应用程序是否占用有处理器的处理资源。还有，对于音频输出通道资源的占用情况，可以对应配置音频输出通道占用状态，该音频输出通道占用状态可以表征应用程序是否占用有音频输出通道。因此，对于每个应用程序对应的运行状态至少可以包括前述的前后台运行状态、处理器占用状态以及音频输出通道占用状态。

[0049] 需要说明的是，前述的前后台运行状态、处理器占用状态以及音频输出通道占用状态等运行状态只是示例性的介绍，对于应用程序所对应的运行状态可以包括除了前后台运行状态、处理器占用状态以及音频输出通道占用状态等运行状态以外更多的状态，例如，还可以包括图像处理器占用状态等。

[0050] 可以理解的是，应用程序的运行状态会根据应用程序当前所实际的占用资源情况而动态的进行更新，进而可以理解的是，应用程序的当前运行状态表征的是应用程序的运行状态当前的实时所表征的情况。例如，应用程序的当前前后台运行状态，表征的是应用程序当前实际是在前台运行，还是在后台运行。再例如，应用程序的当前处理器占用状态，表征的是应用程序当前实际是否占用了处理器的处理资源。基于前述内容可以理解的是，应用程序的当前运行状态可以包括当前前后台运行状态、当前处理器占用状态以及当前音频输出通道占用状态等。

[0051] 作为一种方式，应用程序可以对应配置有一个配置文件，在该配置文件中存储应用程序的当前运行状态。在这种方式下，应用程序可以通过查询配置文件的方式来获取到自身的当前运行状态。示例性的，在配置文件中配置有第一参数、第二参数以及第三参数，其中，第一参数表征当前前后台运行状态，第二参数表征当前处理器占用状态，第三参数表征当前音频输出通道占用状态。可选的，对于每个参数都可以对应配置有0和1两种值以对应不同的状态。例如，若第一参数为0，则可以表征应用程序当前在未在前台运行，对应的，若第一参数为1，则可以表征应用程序当前在前台运行。例如，若第二参数为0，则可以表征应用程序当前未占用处理器的处理资源，对应的，第二参数为1，则可以表征应用程序当前占用有处理器的处理资源。再例如，若第三参数为0，则可以表征应用程序当前未占用音频

输出通道,对应的,若第三参数为1,则可以表征应用程序当前有占用音频输出通道。

[0052] 再者,在本实施例中可以配置有目标事件与应用程序的当前运行状态的对应关系,进而通过该对应关系就可以确定当前运行状态中与所述目标事件对应的运行状态。如前述内容所示,应用程序的每个运行状态表征的是对某种资源的占用情况,而可以理解的是应用程序在触发某个事件时,也会对应的占用一定的资源。

[0053] 例如,若请求触发的是显示子页面(例如,弹窗)的事件,那么在显示子页面时就需要占用显示资源。再例如,若请求触发的是一个数据运算的事件,那么在该数据运算的事件被触发后,就会占用处理器的处理资源。因此,作为一种方式,可以将所需占用的资源相同的事件和运行状态配置为具有对应关系。其中,需要说明的是,事件所需占用的资源可以理解为事件被触发后所占用的资源。运行状态所需占用的资源可以理解为应用程序处于某个运行状态时所需占用的资源。示例性的,在应用程序的当前运行状态中,可以配置当前前后台运行状态与显示子页面的事件对应,配置当前处理器占用状态与数据运算事件对应,配置当前音频输出通道占用状态与音频输出事件对应。作为一种方式,对应事件和运行状态的对应关系也可以存储在前述所指出的配置文件中,以便应用程序通过查询配置文件就可以获取到与某个事件对应的运行状态。

[0054] 基于前述对于应用程序对应的当前运行状态的介绍,在获取到关联应用程序后,就可以获取关联应用程序的当前运行状态中与该目标事件对应的运行状态作为目标运行状态。示例性的,若应用程序A获取到与自身关联的关联应用程序包括有应用程序B、应用程序C以及应用程序D。应用程序A就可以向应用程序B发送第一状态获取请求,应用程序B在接收到该第一状态获取请求后,就可以通过查询前述的配置文件来获取到与该目标事件对应的运行状态作为目标运行状态返回给应用程序A,类似的,应用程序A也可以通过相同的方式获取到应用程序C以及应用程序D返回的目标运行状态。

[0055] S130:检测该目标运行状态是否为目标事件的冲突状态。

[0056] 其中,可选的,处于冲突状态表征的是所需占用的资源相同。目标运行状态为目标事件的冲突状态可以理解为目标运行状态所表征的应用程序的某个当前运行状态所需占用的资源和目标事件被触发后所需占用的资源相同。

[0057] 例如,若目标运行状态为当前前后台运行状态,且目标事件为显示子页面的事件,在这种情况下若当前前后台运行状态表征的是应用程序在前台运行,也就表征关联应用程序当前占用了显示资源,而若目标事件为显示子页面的事件,也就意味着该目标事件在触发以后也需要占用显示资源,进而可以将表征应用程序在前台运行的当前前后台运行状态确定为显示子页面的事件的冲突状态。对应的,表征应用程序在后台运行的当前前后台运行状态则不是显示子页面的事件的冲突状态。

[0058] 作为一种方式,可以预先配置有冲突对应关系来表征对于每个当前运行状态而言,在什么情况下会为某个事件的冲突状态。其中,该冲突对应关系可以以文本的形式存储在前述的配置文件中,也可以数据表的形式存储在数据库中。在这种方式下,在获取到目标运行状态以后,就可以通过查询该冲突对应关系来确定所获取到的目标运行状态是否为目标事件的冲突状态。

[0059] 示例性的,对于当前处理器占用状态而言,可以表征应用程序占用有处理器的处理资源以及应用程序未占用处理器的处理资源这两种情况。根据前述内容可知,与当前处

理器占用状态对应的事件为数据运算的事件，那么当目标事件为数据运算的事件时，且当前处理器占用状态表征的是应用程序占用有处理器的处理资源，就可以确定当前处理器占用状态为数据运算的事件的冲突状态，对应的，当前处理器占用状态表征的是应用程序未占用有处理器的处理资源时，就可以确定当前处理器占用状态不是数据运算的事件的冲突状态。再例如，对于当前音频输出通道占用状态而言，可以表征应用程序占用有音频输出通道，以及未占用音频输出通道两种情况。在目标事件为音频输出事件的情况下，若当前音频输出通道占用状态表征的是应用程序有占用音频输出通道，就可以确定当前音频输出通道占用状态为音频输出事件的冲突状态，对应的，若当前音频输出通道占用状态表征的是应用程序未占用音频输出通道，就可以确定当前音频输出通道占用状态不是音频输出事件的冲突状态。

[0060] S131：若目标运行状态不是目标事件的冲突状态，允许目标事件触发。

[0061] S132：若目标运行状态为目标事件的冲突状态，禁止目标事件触发。

[0062] 本申请提供的一种应用程序控制方法，通过在响应于目标事件的触发请求时，获取与发起该触发请求的应用程序关联的关联应用程序，获取该关联应用程序的当前运行状态中与该目标事件对应的运行状态作为目标运行状态，进而在该目标运行状态为该目标事件的冲突状态的情况下，禁止该目标事件触发。从而通过获取关联应用程序的目标运行状态的方式，实现了在有目标事件触发请求时，可以根据目标运行状态是否为与目标事件的冲突状态来确定是否允许目标事件触发，进而避免目标事件触发后与关联应用程序之间产生运行冲突。

[0063] 请参阅图3，图3所示为本申请一实施例提出的一种应用程序控制方法的流程图，该方法包括：

[0064] S210：响应于目标事件的触发请求，获取关联应用程序，关联应用程序为与发起触发请求的应用程序关联的应用程序。

[0065] S220：基于指定的通信接口与关联应用程序建立通信通道。

[0066] 在本实施例中，执行应用程序控制方法的应用程序可以通过多种方式来与关联应用程序建立通信通道。

[0067] 作为一种方式，在操作系统为Android的情况下，该指定的通信接口可以为内容提供者组件(Content Provider)。在这种方式下该基于指定的通信接口与该关联应用程序建立通信通道，包括：基于该内容提供者组件与该关联应用程序建立通信通道。当前应用程序的内容提供者组件，在接收到其他应用程序通过内容提供组件发送的操作请求(例如，状态获取请求)后，可以通过URL来标识其它应用要访问的数据，通过ContentResolver的增加、删除、更高、查询方法实现对共享数据的操作。其中，该共享数据为当前应用程序分享给其他应用程序的数据，例如，当前应用程序的当前运行状态。

[0068] 作为另外一种方式，该指定的通信接口还可以为AIDL(Android Interface Definition Language)服务提供的跨进程通信接口。

[0069] S230：通过通信通道向关联应用程序发送第一状态获取请求。

[0070] 本实施例中第一状态获取请求，可以理解为执行本实施例提供的应用程序控制方法的应用程序，发送给关联应用程序的状态获取请求。例如，本实施例提供的应用程序控制方法由应用程序A来执行，那么应用程序A在获取到关联应用程序包括应用程序B、应用程序

C以及应用程序D的情况下，应用程序A发送给应用程序B、应用程序C以及应用程序D的状态获取请求则为第一状态获取请求。

[0071] 作为一种方式，可以基于私有的协议来生成第一状态获取请求，进而可以携带有需要了解的当前运行状态的同时，也可以提升状态请求过程中的数据安全性。示例性的，在第一状态获取请求中可以携带有询问前后台运行状态的标识，例如，该询问前后台运行状态的标识可以为“/foreground?”。

[0072] 需要说明的是，对于所获取的关联应用程序可以有多个(即至少2个)，也可以有仅有1个。在有多个关联应用程序的情况下，可以依次多个关联应用程序的目标运行状态，也可以并行同时获取多个关联应用程序的目标运行状态。

[0073] S240：接收关联应用程序响应第一状态获取请求通过通信通道返回的目标运行状态，目标运行状态为关联应用程序的当前运行状态中与目标事件对应的运行状态。

[0074] 可以理解的是，在基于已经建立的通信通道向关联应用程序发送第一状态获取请求后，关联应用程序也会同样的基于该建立的通信通道返回目标运行状态。例如，若关联应用程序在获取携带有“/foreground?”的第一状态获取请求后，关联应用程序会查询当前自己是否在前台运行，若在前台运行则可以通过该通信通道返回表征处于前台运行的状态标识，若在后台运行则可以通过该通信通道返回表征处于后台运行的状态标识。其中，该表征处于前台运行的状态标识可以为1，对应的，表征处于后台运行的状态标识则可以为0。

[0075] S250：检测该目标运行状态是否为目标事件的冲突状态。

[0076] S251：若目标运行状态不是目标事件的冲突状态，允许目标事件触发。

[0077] S252：若目标运行状态为目标事件的冲突状态，禁止目标事件触发。

[0078] 可以理解的是，对于发送该第一状态获取请求的应用程序而言，同样也可以是该关联应用程序的关联应用程序。例如，若应用程序A获取到关联应用程序包括应用程序B、应用程序C以及应用程序D的情况下，当应用程序B在获取关联应用程序时，也会获取到应用程序A为自身的关联应用程序。对应的，应用程序A除了可以向应用程序B、应用程序C以及应用程序D分别发送第一状态获取请求外，也会接收到应用程序B、应用程序C以及应用程序D所发送的状态获取请求，以便将自身的运行状态返回给应用程序B、应用程序C以及应用程序D。其中，在本实施例中，作为一种方式，对于发送第一状态获取请求的应用程序，可以将所接收到的状态获取请求作为第二状态获取请求。那么在接收到第二状态获取请求后，就可以对应的查询所请求的运行状态，并返回给发送该第二状态请求的应用程序。

[0079] 需要说明的是，在本实施例中执行应用程序控制方法的应用程序与关联应用程序之间建立通信通道时，所采用的指定的通信接口可以为通用的通信接口。其中，通用的通信接口可以理解为每个应用程序都可以使用的通信接口。在这种情况下即使不互为关联应用程序，也可以基于该指定的通信接口相互发送数据。示例性的，应用程序A获取到的关联应用程序包括应用程序B、应用程序C以及应用程序D。在这种情况下，应用程序A、应用程序B、应用程序C以及应用程序D互相都为关联应用程序。若应用程序A、应用程序B、应用程序C以及应用程序D互相通信所采用的指定的通信接口为通用接口，那么就会造成应用程序A、应用程序B、应用程序C以及应用程序D可以接收到关联应用程序以外的应用程序(例如，应用程序E)所发送的数据(例如，状态获取请求)，进而就会造成数据泄露。

[0080] 为了改善该数据泄露的问题，在这种方式下，该方法还包括：接收第二状态获取请

求;检测关联应用程序是否包括发送第二状态获取请求的应用程序;若是,向发送第二状态获取请求的应用程序返回所请求的运行状态(例如,前述的目标运行状态)。在这种方式下,在接收到第二状态获取请求后,可以先检测发送给第二状态获取请求的应用程序是否属于关联应用程序,若是,则再返回对应所请求的运行状态。若不是,则可以直接返回空消息(例如,消息的内容为空或者为null),或者不对获取的第二状态获取请求进行响应。示例性的,若应用程序A获取到的关联应用程序包括应用程序B、应用程序C以及应用程序D,在确定到发送第二状态获取请求的应用程序为应用程序B时,可以判定发送第二状态获取请求的应用程序属于关联应用程序,则可以直接返回所请求的目标运行状态,但是在确定发送第二状态获取请求的应用程序为应用程序E时,则判定发送第二状态获取请求的应用程序不属于关联应用程序,则可以直接返回空消息,或者不对获取的第二状态获取请求进行响应。

[0081] 如前述内容所示,应用程序的当前运行状态可以包括有当前前后台运行状态、当前处理器占用状态以及当前音频输出通道占用状态等。对应的,对于关联应用程序的当前运行状态同样会包括有当前前后台运行状态、当前处理器占用状态以及当前音频输出通道占用状态等。在运行状态有较多分类的情况下,为了便于关联应用程序可以快速的识别当前所接收到的请求具体是什么类型的请求,以及具体要请求什么内容,在生成请求时可以基于配置的私有协议来进行生成。

[0082] 示例性的,可以基于下列的格式生成请求:“#befin#”+“&&”+“#request#”+“&&”+“#request_type#”+“&&”+“#end#”。在上述格式中,“#befin#”表征的是这条数据的开始,“#request#”表征的是这条数据的目的是为了进行数据请求。对应的,若要表征这条数据的目的是为了更改关联应用程序中的数据,可以将“#request#”替换为“#update#”。再者,其中的“#request_type#”表征具体所要请求的数据是什么。例如,在前述所指出的询问前后台运行状态的标识可以理解为该“#request_type#”,那么在所生成的请求为第一状态获取请求的情况下,基于该格式所生成的第一状态请求中该request_type的值就可以为“/foreground?”。其中,“&&”表征的是数据分隔符,用于将前述的多个字段分割开。类似的,对于关联应用程序在返回目标运行状态时,也可以基于类型的格式生成返回数据。例如,可以为:“#befin#”+“&&”+“#back#”+“&&”+“#end#”。其中,“#back#”则可以用于携带所返回的目标运行状态。例如,第一状态获取请求所请求的目标运行状态为当前前后台运行状态,且关联应用程序处于前台运行时,该“#back#”所携带的值可以为1,而若关联应用程序处于后台运行时,该“#back#”所携带的值可以为0。

[0083] 本申请提供的一种应用程序控制方法,通过在响应于目标事件的触发请求时,获取与发起该触发请求的应用程序关联的关联应用程序,基于指定的通信接口与该关联应用程序建立通信通道,通过该通信通道获取该关联应用程序的当前运行状态中与该目标事件对应的运行状态作为目标运行状态,进而在该目标运行状态为该目标事件的冲突状态的情况下,禁止该目标事件触发。从而通过获取关联应用程序的目标运行状态的方式,实现了在有目标事件触发请求时,可以根据目标运行状态是否为与目标事件的冲突状态来确定是否允许目标事件触发,进而避免目标事件触发后与关联应用程序之间产生运行冲突。

[0084] 请参阅图4,图4所示为本申请一实施例提出的一种应用程序控制方法的流程图,该方法包括:

[0085] S310:获取第一应用程序名单,以及获取第二应用程序名单,其中,该第一应用程

序名单的获取途径和该第二应用程序名单的获取途径不同。

[0086] 作为一种方式,对于每个应用程序的关联应用程序可以通过配置关联应用程序名单的方式来进行记录。再者,除了可以通过配置关联应用程序名单的方式来进行记录外,作为另外一种方式,还可以通过检测是否具有相同的属性来确定关联应用程序。但是,对于在单独采用上述两种方式来进行关联应用程序获取的过程中都会有一定缺陷,进而造成无法准确的获取本地实际存在的关联应用程序。

[0087] 例如,对于通过配置关联应用程序名单的这种方式中,存在于名单中的关联应用程序可能并没有在本地进行安装,那么即使对未在名单中的关联应用程序发送了通信通道建立请求,也无法得到有效的响应。例如,应用程序A通过配置的关联应用程序名单获取到关联应用程序包括有应用程序B、应用程序C、应用程序D以及应用程序E。但是,应用程序E实际并未在本地进行安装,那么应用程序A即使发送了关于应用程序E的通信通道建立请求,也无法得到响应。

[0088] 再者,在通过检测是否具有相同的属性来确定关联应用程序的这种方式中,对于其他实际不是关联应用程序的应用程序,可以将自己原本的属性修改为与关联应用程序的属性相同,进而实现伪装成关联应用程序。以属性为通信接口为例,在Android系统中对应有AndroidManifest.xml文件,该AndroidManifest.xml文件是Android应用的入口文件,它描述了package中暴露的组件(activities,services,等等),以及他们各自的实现类,各种能被处理的数据和启动位置。

[0089] 在AndroidManifest.xml文件中的“<provider>”表征应用程序会注册一个内容提供者组件(Content Provider),进而通过将“<provider>”的action name字段配置为ASSIST_PRODUCTION的方式,使得该ASSIST_PRODUCTION作为了区别关联应用程序的属性。在这种方式下,应用程序在被安装时首先会在系统中注册自身的内容提供者组件。在联应用程序的获取过程中,可以将配置的内容提供组件中携带有该ASSIST_PRODUCTION的应用程序为关联应用程序。但是,在这种方式下,对于原本并不是属于关联应用程序的应用程序可以通过将自己的action name字段的值也配置为ASSIST_PRODUCTION来伪装成关联应用程序。

[0090] 因此,为了改善该单独采用一种方式来获取关联应用程序造成的无法在本地准确的获取关联应用程序的问题,可以结合前述两种方式共同来获取关联应用程序。进而可以通过一种方式所采用的途径来获取到第一应用程序名单,然后通过另一个方式所采用的途径来获取第二应用程序名单,然后将第一应用程序名单和第二应用程序名单共同包括的应用程序作为本地实际存在的关联应用程序。

[0091] 示例性的,如图5所示,图5示出了获取第一应用程序名单12以及第二程序名单13。其中,第一应用程序名单12包括有应用程序A、应用程序B、应用程序C以及应用程序D。第二应用程序名单13包括有应用程序A、应用程序B、应用程序E以及应用程序F。在将该第一应用程序名单12以及第二程序名单13取交集后,得到它们均包括的应用程序为应用程序A以及应用程序B,进而所得到的关联应用程序名单14包括应用程序A以及应用程序B。

[0092] 作为一种方式,获取第一应用程序名单可以包括:从指定服务器下载应用程序名单作为第一应用程序名单。在这种方式中,对于配置的关联应用程序名单可以预先存储在指定服务器中。如图6所示,当应用程序要获取关联应用程序名单时,承载发送获取请求的

应用程序的电子设备15会将该获取请求发送给指定服务器16，该指定服务器16在接收到获取请求后，可以响应该名单获取请求返回关联应用程序名单12，应用程序进而将该返回的应用程序名单12作为第一应用程序名单。其中，对于返回的应用程序名单12可以存储在应用程序的私有目录下。其中，私有目录可以理解为存储该应用程序的安装文件以及运行过程中所缓存资源的目录。

[0093] 在这种方式下，可以理解的是，指定服务器中所存储的关联应用程序名单可能会进行更新。例如，当指定服务器的管理人员期望增加或者剔除应用程序时，就会对存储的联应用程序名单进行更新。再者，指定服务器也可以主动周期性的对存储的关联应用程序名单进行更新。为了能够及时的获取到指定服务器中更新的熬最新版本的关联应用程序名单，在向指定服务器发送获取请求时，可以增加表征本地的关联应用程序的数据，以便指定服务器可以确定是否需要返回本地存储的最新版本的关联应用程序名单。其中，指定服务器可以为预先配置在发送该获取请求的应用程序中的网络地址所指向的服务器。可选的，该网络地址可以由开发人员预先配置在该应用程序中，以便该应用程序在要获取关联应用程序名单时，可以访问该网络地址所指向的服务器以获取到关联应用程序名单。

[0094] 作为一种方式，如图7所示，该从指定服务器下载应用程序名单作为该第一应用程序名单，包括：

[0095] S311：向指定服务器发送携带第一特征数据的获取请求，第一特征数据为基于历史下载的应用程序名单计算得到的数据。

[0096] 可选的，第一特征数据为基于历史下载的应用程序名单中的应用程序标识计算得到。需要说明的是，其中的历史下载的应用程序名单为发送该获取请求之前最近一次下载的应用程序名单。作为一种方式，其中的应用程序标识可以为应用程序的包名，该包名为应用程序的安装包的名称，通常格式为com.xxxx.yyyy。可以理解的是，其中的“xxxx”和“yyyy”只是替代字符，并不表示实际的字符内容。在这种方式下，可以基于历史下载的应用程序名单中的应用程序的包名进行哈希(例如，MD5算法)计算得到该第一特征数据。其中，MD5算法是一种密码散列函数，可以产生出一个128位(16字节)的散列值，用于确保信息传输完整一致。

[0097] S312：指定服务器基于本地最新版本的关联应用程序名单计算得到第二特征数据。

[0098] 可选的，指定服务器计算第二特征数据的方式要和前述计算第一特征数据的方式相同。可以为基于本地最新版本的关联应用程序名单中的应用程序标识计算得到第二特征数据。其中，若第一特征数据是基于哈希算法计算得到，那么第二特征数据也需要是基于哈希算法计算得到，以便于准确的比对第一特征数据和第二特征数据是否相同。

[0099] S313：若指定服务器检测第二特征数据与第一特征数据不同，将本地存储的最新版本的关联应用程序名单返回给发送获取请求的应用程序。

[0100] 可以理解的是，若第二特征数据和第一特征数据不同的，则表示指定服务器中存储的关联应用程序名单已经进行了更新。

[0101] S314：接收指定服务器返回的应用程序名单，其中，基于返回的应用程序名单的应用程序标识计算得到的第二特征数据与第一特征数据不同。

[0102] 若指定服务器检测到该第一特征数据和第二特征数据相同，则判断发起获取请求

的应用程序在本地存储的关联应用程序名单和指定服务器本地存储的最新版本的关联应用程序名单是相同的，进而指定服务器也就不用再进行关联应用程序名单的返回，以节约网络资源。

[0103] 可以理解的是，所接收到的服务器返回的应用程序名单即为前述指定服务器本地存储的最新版本的关联应用程序名单。

[0104] S315：将返回的应用程序名单作为第一应用程序名单。

[0105] 再者，可以通过在本地进行指定通信接口查找的方式来获取到第二应用程序名单。作为一种方式，该获取第二应用程序名单，包括：获取具有指定通信接口的应用程序；生成该第二应用程序名单，该第二应用程序名单包括该具有指定通信接口的应用程序。

[0106] S320：将第一应用程序名单和第二应用程序名单均包括的应用程序作为所述关联应用程序并进行存储。

[0107] S330：响应于目标事件的触发请求，获取存储的关联应用程序。

[0108] S340：获取关联应用程序的目标运行状态，目标运行状态为关联应用程序的当前运行状态中与目标事件对应的运行状态。

[0109] S350：检测该目标运行状态是否为目标事件的冲突状态。

[0110] S351：若目标运行状态不是目标事件的冲突状态，允许目标事件触发。

[0111] S352：若目标运行状态为目标事件的冲突状态，禁止目标事件触发。

[0112] 需要说明的是，在本实施例中S310和S320所对应的流程，和后续S330到S350对应的流程之间不一定具有必然的顺序性。例如，S310和S320可以独立的周期性执行，而不一定要在执行S350后才能再次执行S310，以便可以及时的根据指定服务器中最新版的关联应用程序名单对所存储的关联应用程序进行更新，只是在执行S330时，可以直接读取基于S310和S320所存储的关联应用程序，以提升关联应用程序的获取速率。

[0113] 本申请提供的一种应用程序控制方法，实现了在有目标事件触发请求时，可以根据目标运行状态是否为与目标事件的冲突状态来确定是否允许目标事件触发，进而避免目标事件触发后与关联应用程序之间产生运行冲突。并且，在本实施例中，可以预先通过获取第一应用程序名单以及第二应用程序名单来得到关联应用程序并进行存储，从而在检测到有目标事件的触发请求时，可以直接获取已经存储好的关联应用程序，而不用再实时再去实时获取第一应用程序名单以及获取第二应用程序名单来得到关联应用程序，进而提升了数据的处理效率，从而使得可以更加快速的判断关联应用程序的当前运行状态是否与该目标事件的冲突。

[0114] 请参阅图8，图8所示为本申请一实施例提出的一种应用程序控制方法的流程图，该方法包括：

[0115] S410：响应于目标事件的触发请求，获取参考运行状态，所述参考运行状态为发起所述触发请求的应用程序的当前运行状态。

[0116] S411：检测所述参考运行状态与所述目标事件对应的运行状态是否相同。

[0117] 需要说明的是，若目标事件触发后所需占用的资源和关联应用程序当前已经占用的资源相同，就可能会产生运行冲突。在本申请实施例中，获取关联应用程序的目标运行状态的目的是为了确定该目标事件在被触发后会不会与关联应用程序之间产生运行冲突。因此，若发起该触发请求的应用程序本身就已经占用了该目标事件触发所需的资源，也就意

味着发起所述触发请求的应用程序的当前运行状态中与所述目标事件对应的运行状态，与所述目标事件对应的运行状态是相同的，那么就不用再对关联应用程序的目标运行状态进行判断。其中，当前运行状态与事件的对应关系可以参见前述实施例中的介绍，此处就不再赘述。

[0118] 示例性的，若目标事件为显示子页面的事件，那么该显示子页面的事件在触发后所需占用的为显示资源。而发送该触发请求的应用程序的当前前后台运行状态(即当前运行状态中与该显示子页面的事件对应的运行状态)表征该应用程序目前已经处于前台运行状态的情况下，表征该发送该触发请求的应用程序已经占用了显示资源，进而就可以判定发送该触发请求的应用程序已经处于显示子页面的事件对应的运行状态。再例如，若目标事件为音频输出事件，在该音频输出事件被触发后所需占用的资源音频输出通道资源。而发送该触发请求的应用程序的当前音频输出通道占用状态(即当前运行状态中与该音频输出事件对应的运行状态)表征该发送该触发请求的应用程序处于占用音频输出通道状态是，则可以确定已经占用有音频输出通道资源，进而就可以判定发送该触发请求的应用程序已经处于音频输出事件对应的运行状态。

[0119] S412:若相同，执行该目标事件。

[0120] 可以理解的是，若检测到发起所述触发请求的应用程序处于目标事件对应的运行状态，就表征该目标事件触发后并不会与关联应用程序之间产生运行冲突，进而就可以直接执行该目标事件。

[0121] S420:若不相同，获取第一应用程序名单，以及获取第二应用程序名单，其中，该第一应用程序名单的获取途径和该第二应用程序名单的获取途径不同。

[0122] 在本实施例中获取第一应用程序名单和获取第二应用程序名单的方式可以参考前述实施例中所介绍的方式，此处不再赘述。

[0123] S430:将该第一应用程序名单和该第二应用程序名单均包括的应用程序作为关联应用程序。

[0124] S440:获取当次运行状态获取过程中，当次进行状态获取的关联应用程序的目标运行状态。

[0125] 需要说明的是，在获取到的关联应用程序包括两个甚至更多的情况下，可以依次获取关联应用程序的目标运行状态，并且在检测到某个关联应用程序的目标运行状态为目标事件的冲突状态时，若还有剩余未获取目标运行状态的应用程序，则可以停止获取。在依次获取关联应用程序的目标运行状态的过程中，可以先与当前要获取的关联应用程序建立通信通道，而在获取到该当前要获取的关联应用程序返回的目标运行状态后，就可以与该当前要获取的关联应用程序断开该通信通道，然后再与下一个要获取目标运行状态的关联应用程序建立通信通道。

[0126] S450:判断当次进行状态获取的关联应用程序的目标运行状态是否为该目标事件的冲突状态。

[0127] S460:若是，禁止目标事件触发。

[0128] S470:若否，进入一下次的运行状态获取过程。

[0129] 需要说明的是，在关联应用程序有至少两个的情况下，若检测到该至少两个关联应用程序的目标运行状态均不为目标事件的冲突状态，则可以允许该目标事件触发。

[0130] 示例性的，若应用程序A获取到关联应用程序包括应用程序B、应用程序C以及应用程序D。且应用程序A确定的目标运行状态获取顺序依次为应用程序B、应用程序C以及应用程序D。在获取应用程序B的目标运行状态的过程中，当次进行状态获取的关联应用程序即为该应用程序B，并且若后续检测应用程序B的目标运行状态为目标事件的冲突状态，则就不用再获取应用程序C以及应用程序D的目标运行状态。但是，若后续检测应用程序B的目标运行状态不是目标事件的冲突状态，需要继续检测应用程序C和应用程序D的目标运行状态，若应用程序C和应用程序D的目标运行状态均不是目标事件的冲突状态，则可以允许该目标事件触发。

[0131] 下面结合弹窗显示场景对本实施例的内容进行说明。如图9所示，包括：

[0132] S480：检测自身是否处于前台运行。

[0133] 在目标事件为在前台进行弹窗显示的情况下，响应于在前台进行弹窗显示的触发请求，触发请求在前台进行弹窗显示的应用程序会检测自身是否处于前台运行，进而获取到与在前台进行弹窗显示这个目标事件对应的参考运行状态。

[0134] 若是则流程终止。可以理解的是，若触发请求在前台进行弹窗显示的应用程序检测到自身已经处于前台运行，那么即使直接触发在前台进行弹窗显示也不会对其他矩阵应用程序造成干扰，进而就可以不用再进行后续的流程。

[0135] S481：若否，获取最新网络矩阵应用名单。

[0136] 触发请求在前台进行弹窗显示的应用程序检测到自身不在前台运行的情况下，为了避免对矩阵应用造成干扰，就会进一步的获取最新网络矩阵应用名单。可选的，可以从运行在网络端的配置服务获取最新网络矩阵应用名单。对于网络端存储的网络矩阵应用名单可以由管理员进行更新。需要说明的是，这里获取的最新网络矩阵应用名单可以理解为前述实施例中的第一应用程序名单。

[0137] S482：通过内容提供者组件接口查找本地矩阵应用名单。

[0138] 需要说明的是，这里通过内容提供者组件(Content Provider)接口查找本地矩阵应用名单可以理解为前述实施例中的第二应用程序名单。

[0139] S483：获取可用矩阵应用名单。

[0140] 将最新网络矩阵应用名单和本地矩阵应用名单中的内容取交集就可以得到可用矩阵应用名单。需要说明的是，在可用矩阵应用名单中所存储的矩阵应用可以理解为前述实施例中的关联应用程序。

[0141] S484：遍历可用矩阵名单。

[0142] S485：使用内容提供者组件接口与对应应用建立通信通道。

[0143] 需要说明的是，对于可用矩阵名单中的矩阵应用，会分多次获取各自对应的前台运行状态，这里建立通信通道的对应应用可以理解为前述的当次进行状态获取的矩阵应用，对应的，使用ContentProvider接口与对应应用建立通信通道可以理解为使用ContentProvider接口与当次进行状态获取的矩阵应用建立通信通道。

[0144] S486：根据协议向对应应用发送询问请求。

[0145] 可选的，询问请求的内容可以为“/foreground?”。

[0146] S487：对应应用检测自身是否在前台。

[0147] 对应应用在接收到内容为“/foreground?”的请求后就可以开始检测自身是否处

于前台运行，并生成检测结果。

[0148] S488：对应应用反馈检测结果并断开通信通道。

[0149] S489：对应应用是否处于前台运行。

[0150] 在接收到对应应用反馈的检测结果后，就可以从检测结果中识别对应应用是否处于前台运行。

[0151] 若是，则流程终止。需要说明的是，每次询问的对应应用都是属于可用矩阵名单中的矩阵应用，那么若检测到对应应用在前台运行，也就是表征已经确定矩阵应用有应用程序处于前台运行，进而就可以让流程终止。

[0152] S490：若否，则判断遍历是否结束。

[0153] 可用矩阵名单中的矩阵应用可能是会有多个的，那么在当次检测过程中的对应应用未处于前台运行的状态下，可以先判断是否已经遍历了所有的矩阵应用。

[0154] S491：若是，则流程终止，并返回空结果。

[0155] 若否，则再次回到步骤S484，以便继续下一次的检测。

[0156] 从而通过前述步骤，就实现了触发请求在前台进行弹窗显示的应用程序可以根据矩阵应用是否在前台运行来确定是否允许触发执行在前台进行弹窗显示。

[0157] 需要说明的是，在弹窗显示场景中，S481和S482可以在步骤S480之前就预先执行，并且可以将获取到的可用矩阵名单进行存储，在这种情况下，在执行S483时获取的可用矩阵名单为预先存储的可用矩阵名单。

[0158] 本申请提供的一种应用程序控制方法，会先检测自身是否处于目标事件对应的运行状态，进而在未处于目标事件对应的运行状态的情况下，再进一步的获取关联应用程序，以便在当次进行状态获取的关联应用程序的目标运行状态是否为目标事件的冲突状态时禁止目标事件触发。从而实现了在有目标事件触发请求时，可以根据目标运行状态是否为与目标事件的冲突状态来确定是否允许目标事件触发，进而避免目标事件触发后与关联应用程序之间产生运行冲突。

[0159] 请参阅图10，图10所示为本申请一实施例提出的一种应用程序控制方法的流程图，该方法包括：

[0160] S510：响应于显示独立子页面的事件的触发请求，获取关联应用程序，该关联应用程序为与发起触发请求的应用程序关联的应用程序。

[0161] S520：获取关联应用程序的前后台运行状态。

[0162] S530：检测该前后台运行状态是否为显示独立子页面的事件的冲突状态。

[0163] S531：若前后台运行状态表征关联应用程序在后台运行，判定目标运行状态为显示独立子页面的事件的冲突状态，允许显示独立子页面的事件触发。

[0164] S532：若目标运行状态表征关联应用程序在前台运行，判定目标运行状态为显示独立子页面的事件的冲突状态，禁止显示独立子页面的事件触发。

[0165] 需要说明的是，在本申请实施例中可以有多种方式来确定应用程序是否在前台运行。

[0166] 作为一种方式，根据Android系统的机制，一个应用可以运行多个进程，而所有进程都对应着一个进程优先级，进程优先级可以对应该进程所属的应用程序在系统当前处于的状态(前台运行状态、后台运行状态等)。当一个应用程序的界面正在显示时，则运行该界

面的进程的优先级为IMPORTANCE_FOREGROUND＝100。而Android系统允许应用在运行时可以获取当前自身应用所有进程的优先级，那么利用该原理，应用程序对自身所有进程的优先级并进行遍历，当应用程序检测到自己所包括的进程中存在某个进程优先级＝IMPORTANCE_FOREGROUND时，则判断为自身时处于前台运行。

[0167] 本申请提供的一种应用程序控制方法，通过获取关联应用程序的前后台运行状态的方式，实现了在有显示独立子页面的事件的触发请求时，可以根据关联应用程序的前后台运行状态是否为与目标事件的冲突状态来确定是否允许显示独立子页面，进而避免显示独立子页面后覆盖在该关联应用程序的运行界面上，进而与关联应用程序之间产生运行冲突。

[0168] 请参阅图11，图11所示为本申请一实施例提出的一种应用程序控制方法的流程图，应用于管理平台，该方法包括：

[0169] S610：接收应用程序发送的状态获取请求，状态获取请求中携带有表征该应用程序待触发的目标事件的事件标识。

[0170] 需要说明的是，在本实施例中管理平台可以为一系统程序，也可以为运行在某个应用程序中的插件。在本实施例中，作为一种方式，可以基于前述的生成请求的格式来生成状态获取请求。并且，在该管理平台对应的存储区域中还可以存储有目标事件和运行状态的对应关系，进而使得管理平台在接收到状态获取请求并从中提取出事件标识后，可以确定需要获取哪个类型的当前运行状态。

[0171] S620：获取发送状态获取请求的应用程序的关联应用程序。

[0172] 在本实施例中，管理平台可以基于前述实施例中获取关联应用程序的方式，此处不再赘述。

[0173] S630：获取关联应用程序的目标运行状态，目标运行状态为关联应用程序的运行状态中与事件标识表征的目标事件对应的运行状态。

[0174] 示例性的，管理平台对应的存储区域所存储的目标事件和运行状态的对应关系可以包括：当前前后台运行状态与显示子页面的事件对应，当前处理器占用状态与数据运算事件对应，当前音频输出通道占用状态与音频输出事件对应。当管理平台获取到的事件标识对应的为显示子页面的事件，那么可以确定目标运行状态为与显示子页面的事件对应的当前前后台运行状态。

[0175] S640：检测该目标运行状态是否为目标事件的冲突状态。

[0176] S641：若目标运行状态不是目标事件的冲突状态，允许目标事件触发。

[0177] S642：若目标运行状态为事件标识所表目标征事件的冲突状态，禁止应用程序触发目标事件。

[0178] 在本实施例中，管理平台可以向该应用程序发送一条指令，进而应用程序通过该指令的内容来是否可以执行该目标事件。那么在若目标运行状态为事件标识所表目标征事件的冲突状态的情况下，管理平台会发送表征禁止应用程序触发目标事件的指令给应用程序。

[0179] 本申请提供的一种应用程序控制方法，管理平台通过获取关联应用程序的目标运行状态的方式，实现了在有目标事件触发请求时，管理平台可以根据目标运行状态是否为与目标事件的冲突状态来确定是否允许目标事件触发，进而避免目标事件触发后与关联应

用程序之间产生运行冲突。

[0180] 请参阅图12，图12所示为本申请一实施例提出的一种应用程序控制装置700的结构框图，该装置700包括：

[0181] 关联程序获取单元710，用于响应于目标事件的触发请求，获取关联应用程序，该关联应用程序为与发起该触发请求的应用程序关联的应用程序。

[0182] 作为一种方式，关联程序获取单元710，具体用于若发起该触发请求的应用程序，未处于该目标事件对应的运行状态，获取关联应用程序。

[0183] 运行状态获取单元720，用于获取该关联应用程序的目标运行状态，该目标运行状态为该关联应用程序的当前运行状态中与该目标事件对应的运行状态。

[0184] 作为一种方式，运行状态获取单元720，具体用于基于指定的通信接口与该关联应用程序建立通信通道；通过该通信通道向该关联应用程序发送第一状态获取请求；接收该关联应用程序响应该第一状态获取请求通过该通信通道返回的该目标运行状态。在这种方式下，运行状态获取单元720，还具体用于接收第二状态获取请求；检测该关联应用程序是否包括发送该第二状态获取请求的应用程序；若是，向发送该第二状态获取请求的应用程序返回所请求的运行状态。其中，可选的，该指定的通信接口包括内容提供者组件。运行状态获取单元720，具体用于基于该内容提供者组件与该关联应用程序建立通信通道。

[0185] 其中，作为一种方式，运行状态获取单元720，具体用于从指定服务器下载应用程序名单作为该第一应用程序名单。在这种方式下，运行状态获取单元720，具体用于向该指定服务器发送携带第一特征数据的获取请求，该第一特征数据为基于历史下载的应用程序名单中的应用程序标识计算得到；接收该指定服务器返回的应用程序名单，其中，基于该返回的应用程序名单的应用程序标识计算得到的第二特征数据与该第一特征数据不同；将该返回的应用程序名单作为该第一应用程序名单。

[0186] 作为一种方式，运行状态获取单元720，具体用于获取具有指定通信接口的应用程序；生成该第二应用程序名单，该第二应用程序名单包括该具有指定通信接口的应用程序。

[0187] 程序控制单元730，用于若该目标运行状态为该目标事件的冲突状态，禁止该目标事件触发。

[0188] 作为一种方式，如图13所示，该装置700，还包括：

[0189] 关联程序存储单元740，用于获取第一应用程序名单，以及获取第二应用程序名单，其中，该第一应用程序名单的获取途径和该第二应用程序名单的获取途径不同；将该第一应用程序名单和该第二应用程序名单均包括的应用程序存储为该关联应用程序。在这种方式下，运行状态获取单元720，具体用于获取该存储的关联应用程序。

[0190] 作为一种方式，该目标事件包括显示独立子页面的事件；该目标运行状态包括前后台运行状态，该前后台运行状态表征应用程序是在前台运行或者在后台运行。在这种方式下，程序控制单元730，具体用于若该目标运行状态表征关联应用程序在前台运行，判定该目标运行状态为该显示独立子页面的事件的冲突状态，禁止该显示独立子页面的事件触发。

[0191] 请参阅图14，图14所示为本申请一实施例提出的一种应用程序控制装置800的结构框图，该装置800包括：

[0192] 请求管理单元810，用于接收应用程序发送的状态获取请求，该状态获取请求中携

带有表征该应用程序待触发的目标事件的事件标识。

[0193] 关联程序获取单元820，用于获取发送该状态获取请求的应用程序的关联应用程序。

[0194] 运行状态获取单元830，用于获取该关联应用程序的目标运行状态，该目标运行状态为该关联应用程序的运行状态中与该事件标识表征的目标事件对应的运行状态。

[0195] 程序控制单元840，用于若该目标运行状态为该事件标识所表目标征事件的冲突状态，禁止该应用程序触发该目标事件。

[0196] 本申请提供的一种应用程序控制装置，通过在响应于目标事件的触发请求时，获取与发起该触发请求的应用程序关联的关联应用程序，获取该关联应用程序的当前运行状态中与该目标事件对应的运行状态作为目标运行状态，进而在该目标运行状态为该目标事件的冲突状态的情况下，禁止该目标事件触发。从而通过获取关联应用程序的目标运行状态的方式，实现了在有目标事件触发请求时，可以根据目标运行状态是否为与目标事件的冲突状态来确定是否允许目标事件触发，进而避免目标事件触发后与关联应用程序之间产生运行冲突。

[0197] 需要说明的是，本申请中装置实施例与前述方法实施例是相互对应的，装置实施例中具体的原理可以参见前述方法实施例中的内容，此处不再赘述。

[0198] 下面将结合图15对本申请提供的一种电子设备进行说明。

[0199] 请参阅图15，基于上述的应用程序控制方法，本申请实施例还提供的另一种包括可以执行前述应用程序控制方法的处理器104的电子设备200，该电子设备200可以为智能手机、平板电脑、计算机或者便携式计算机等设备。电子设备200还包括存储器104、网络模块106以及屏幕108。其中，该存储器104中存储有可以执行前述实施例中内容的程序，而处理器102可以执行该存储器104中存储的程序。

[0200] 其中，处理器102可以包括一个或者多个用于处理数据的核以及消息矩阵单元。处理器102利用各种接口和线路连接整个电子设备200内的各个部分，通过运行或执行存储在存储器104内的指令、程序、代码集或指令集，以及调用存储在存储器104内的数据，执行电子设备200的各种功能和处理数据。可选地，处理器102可以采用数字信号处理(Digital Signal Processing,DSP)、现场可编程门阵列(Field—Programmable Gate Array,FPGA)、可编程逻辑阵列(Programmable Logic Array,PLA)中的至少一种硬件形式来实现。处理器102可集成中央处理器(Central Processing Unit,CPU)、图像处理器(Graphics Processing Unit,GPU)和调制解调器等中的一种或几种的组合。其中，CPU主要处理操作系统、用户界面和应用程序等；GPU用于负责显示内容的渲染和绘制；调制解调器用于处理无线通信。可以理解的是，上述调制解调器也可以不集成到处理器102中，单独通过一块通信芯片进行实现。

[0201] 存储器104可以包括随机存储器(Random Access Memory,RAM)，也可以包括只读存储器(Read-Only Memory)。存储器104可用于存储指令、程序、代码、代码集或指令集。存储器104可包括存储程序区和存储数据区，其中，存储程序区可存储用于实现操作系统的指令、用于实现至少一个功能的指令(比如触控功能、声音播放功能、图像播放功能等)、用于实现下述各个方法实施例的指令等。存储数据区还可以存储终端100在使用中所创建的数据(比如电话本、音视频数据、聊天记录数据)等。

[0202] 所述网络模块106用于接收以及发送电磁波，实现电磁波与电信号的相互转换，从而与通讯网络或者其他设备进行通讯，例如和音频播放设备进行通讯。所述网络模块106可包括各种现有的用于执行这些功能的电路元件，例如，天线、射频收发器、数字信号处理器、加密/解密芯片、用户身份模块(SIM)卡、存储器等等。所述网络模块106可与各种网络如互联网、企业内部网、无线网络进行通讯或者通过无线网络与其他设备进行通讯。上述的无线网络可包括蜂窝式电话网、无线局域网或者城域网。例如，网络模块106可以与基站进行信息交互。

[0203] 所述屏幕108可以进行界面内容的显示，也可以用于响应触控手势。

[0204] 需要说明的是，为了实现更多的功能，电子设备200还可以保护更多的器件，例如，还可以保护用于进行人脸信息采集的结构光传感器或者还可以保护用于采集虹膜的摄像头等。

[0205] 请参考图16，其示出了本申请实施例提供的一种计算机可读存储介质的结构框图。该计算机可读介质1100中存储有程序代码，所述程序代码可被处理器调用执行上述方法实施例中所描述的方法。

[0206] 计算机可读存储介质1100可以是诸如闪存、EEPROM(电可擦除可编程只读存储器)、EPROM、硬盘或者ROM之类的电子存储器。可选地，计算机可读存储介质1100包括非易失性计算机可读介质(non-transitory computer-readable storage medium)。计算机可读存储介质1100具有执行上述方法中的任何方法步骤的程序代码810的存储空间。这些程序代码可以从一个或者多个计算机程序产品中读出或者写入到这一个或者多个计算机程序产品中。程序代码1110可以例如以适当形式进行压缩。

[0207] 综上所述，本申请提供的一种应用程序控制方法、装置、电子设备及存储介质，通过在响应于目标事件的触发请求时，获取与发起所述触发请求的应用程序关联的关联应用程序，获取所述关联应用程序的当前运行状态中与所述目标事件对应的运行状态作为目标运行状态，进而在所述目标运行状态为所述目标事件的冲突状态的情况下，禁止所述目标事件触发。从而通过获取关联应用程序的目标运行状态的方式，实现了在有目标事件触发请求时，可以根据目标运行状态是否为与目标事件的冲突状态来确定是否允许目标事件触发，进而避免目标事件触发后与关联应用程序之间产生运行冲突。

[0208] 最后应说明的是：以上实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不驱使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

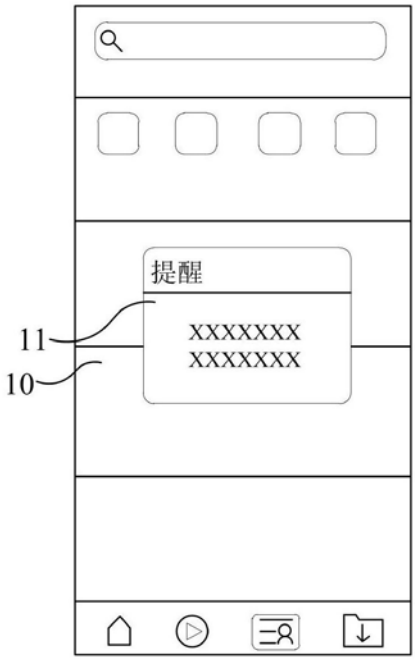

提醒
XXXXXXX
XXXXXXX
11
10

图1

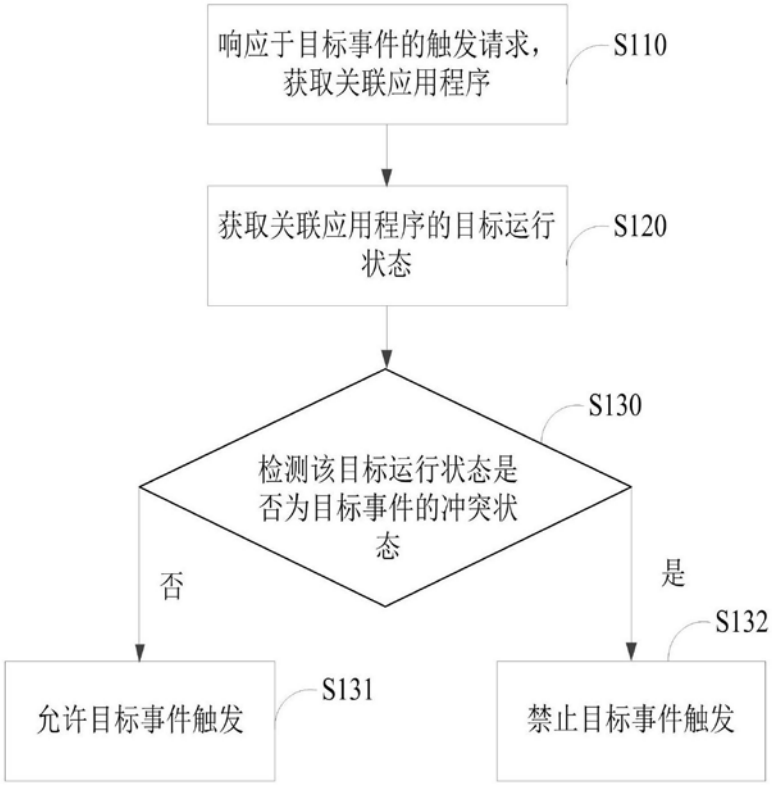

响应于目标事件的触发请求，获取关联应用程序
S110
获取关联应用程序的目标运行状态
S120
检测该目标运行状态是否为目标事件的冲突状态
S130
否
是
允许目标事件触发
S131
禁止目标事件触发
S132

图2

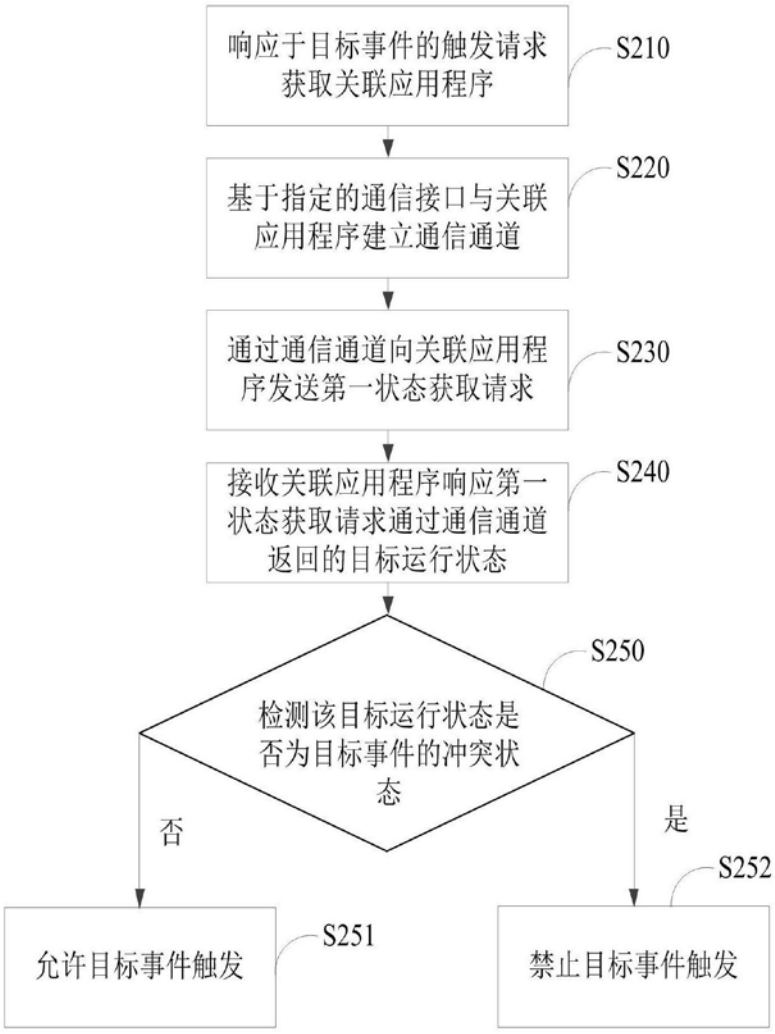

响应于目标事件的触发请求获取关联应用程序
S210
基于指定的通信接口与关联应用程序建立通信通道
S220
通过通信通道向关联应用程序发送第一状态获取请求
S230
接收关联应用程序响应第一状态获取请求通过通信通道返回的目标运行状态
S240
检测该目标运行状态是否为目标事件的冲突状态
S250
否
是
允许目标事件触发
S251
禁止目标事件触发
S252

图3

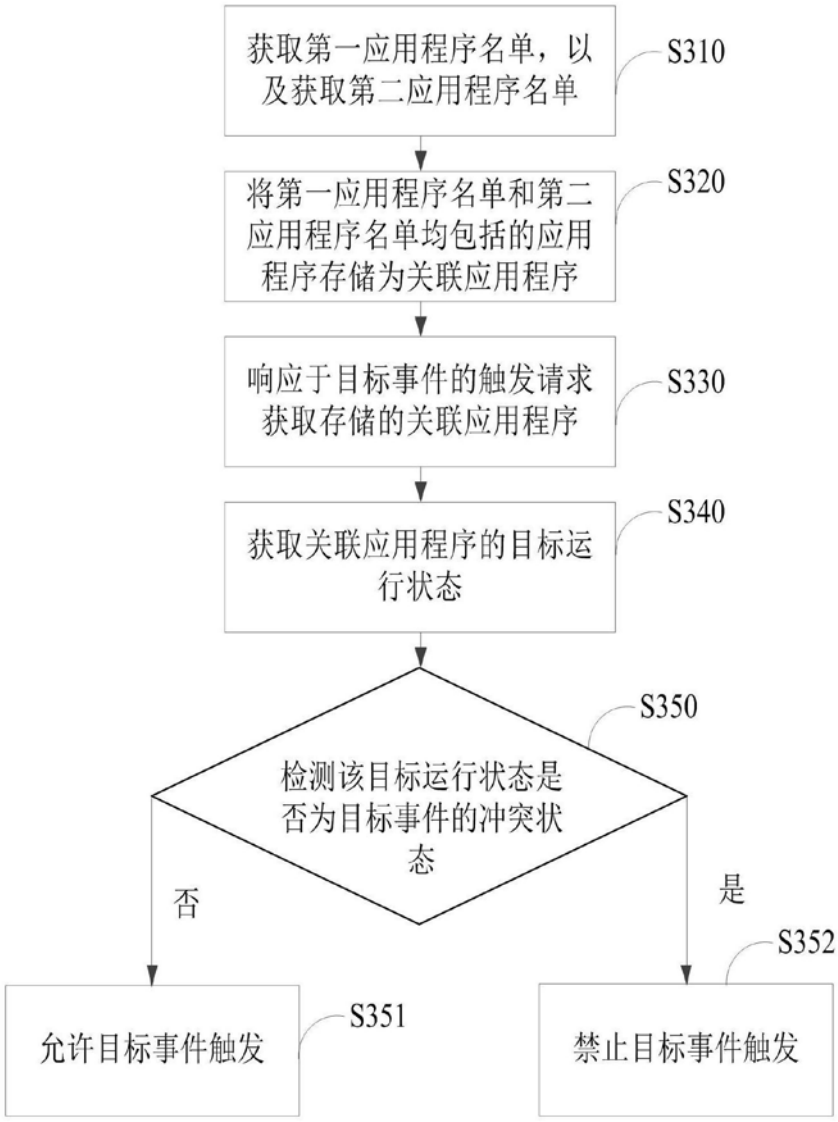

获取第一应用程序名单，以及获取第二应用程序名单
S310
将第一应用程序名单和第二应用程序名单均包括的应用程序存储为关联应用程序
S320
响应于目标事件的触发请求获取存储的关联应用程序
S330
获取关联应用程序的目标运行状态
S340
检测该目标运行状态是否为目标事件的冲突状态
S350
否
是
允许目标事件触发
S351
禁止目标事件触发
S352

图4

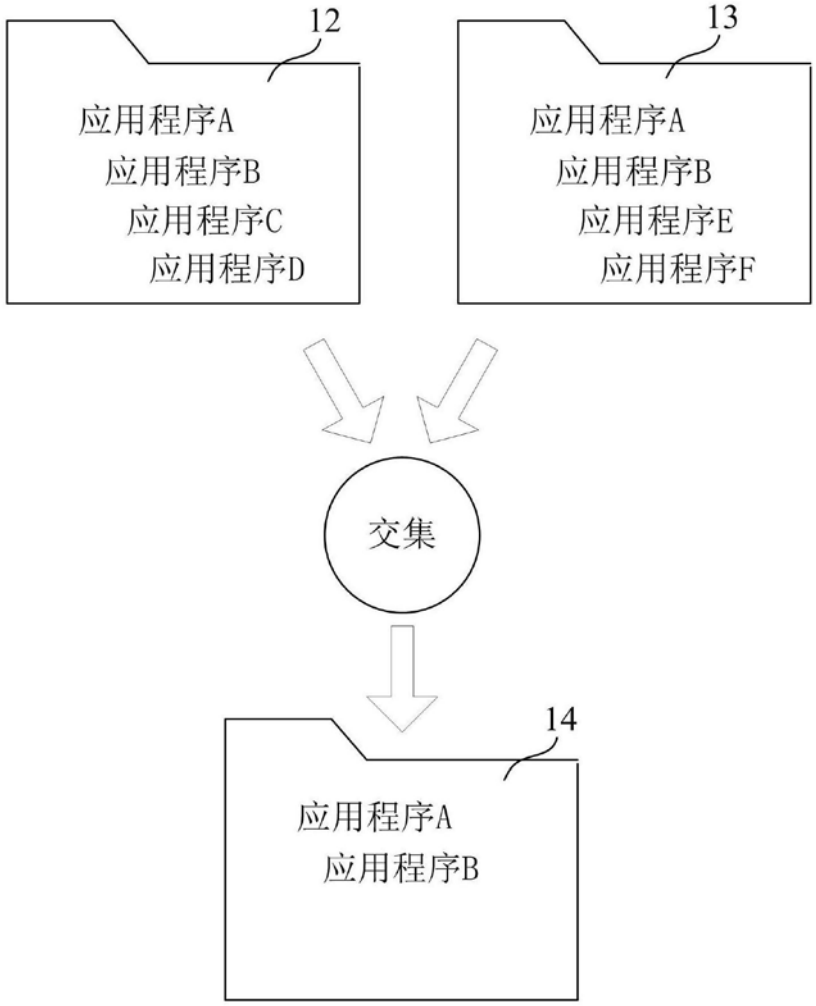

12
应用程序A
应用程序B
应用程序C
应用程序D
13
应用程序A
应用程序B
应用程序E
应用程序F
交集
14
应用程序A
应用程序B

图5

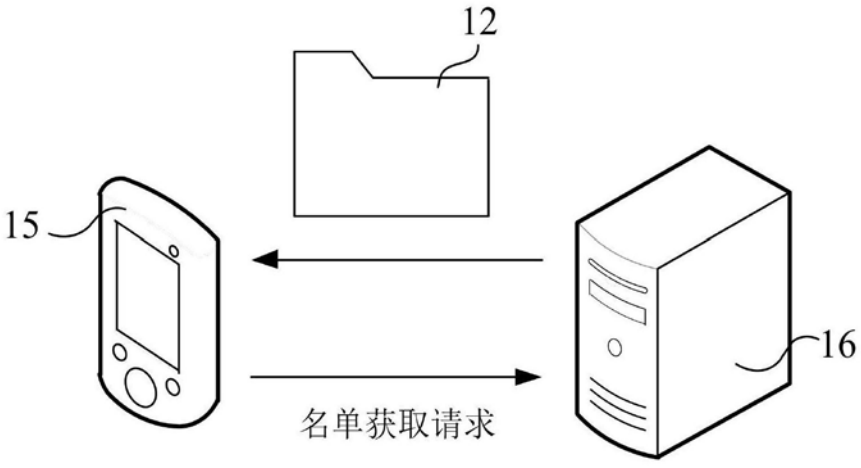

12
15
16
名单获取请求

图6

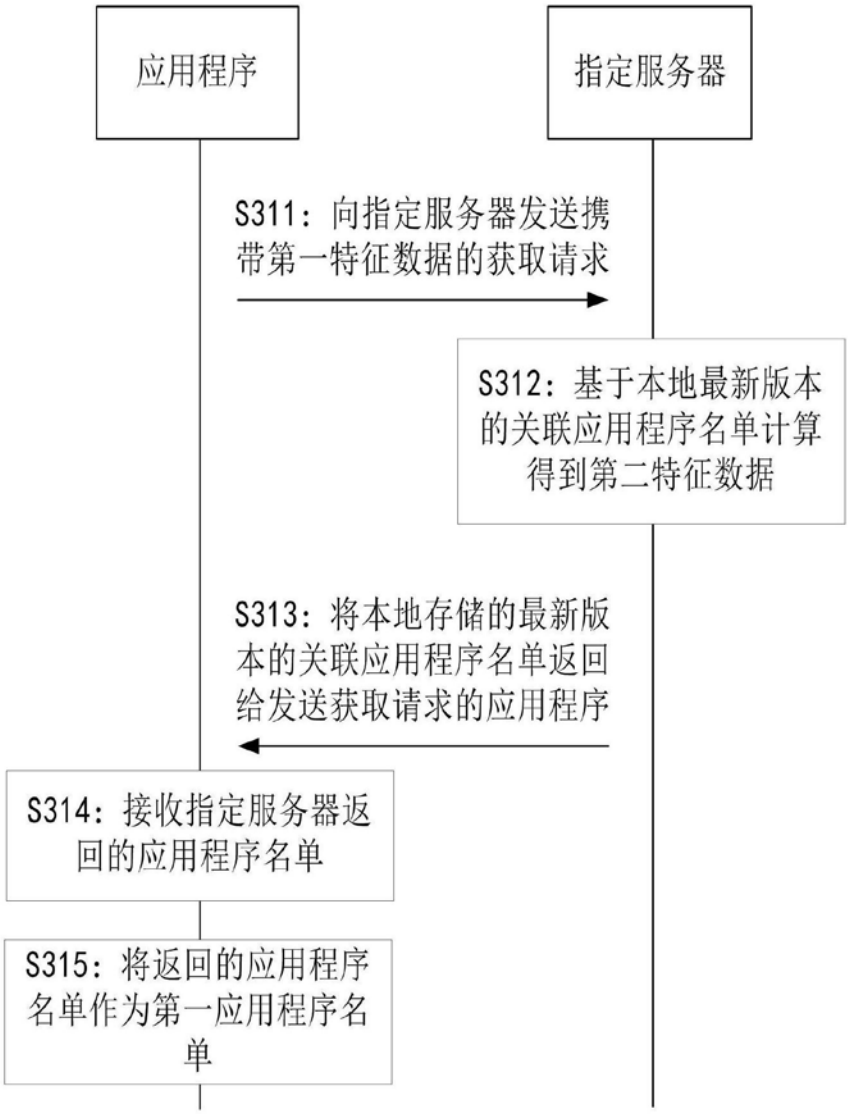

应用程序
指定服务器
S311：向指定服务器发送携带第一特征数据的获取请求
S312：基于本地最新版本的关联应用程序名单计算得到第二特征数据
S313：将本地存储的最新版本的关联应用程序名单返回给发送获取请求的应用程序
S314：接收指定服务器返回的应用程序名单
S315：将返回的应用程序名单作为第一应用程序名单

图7

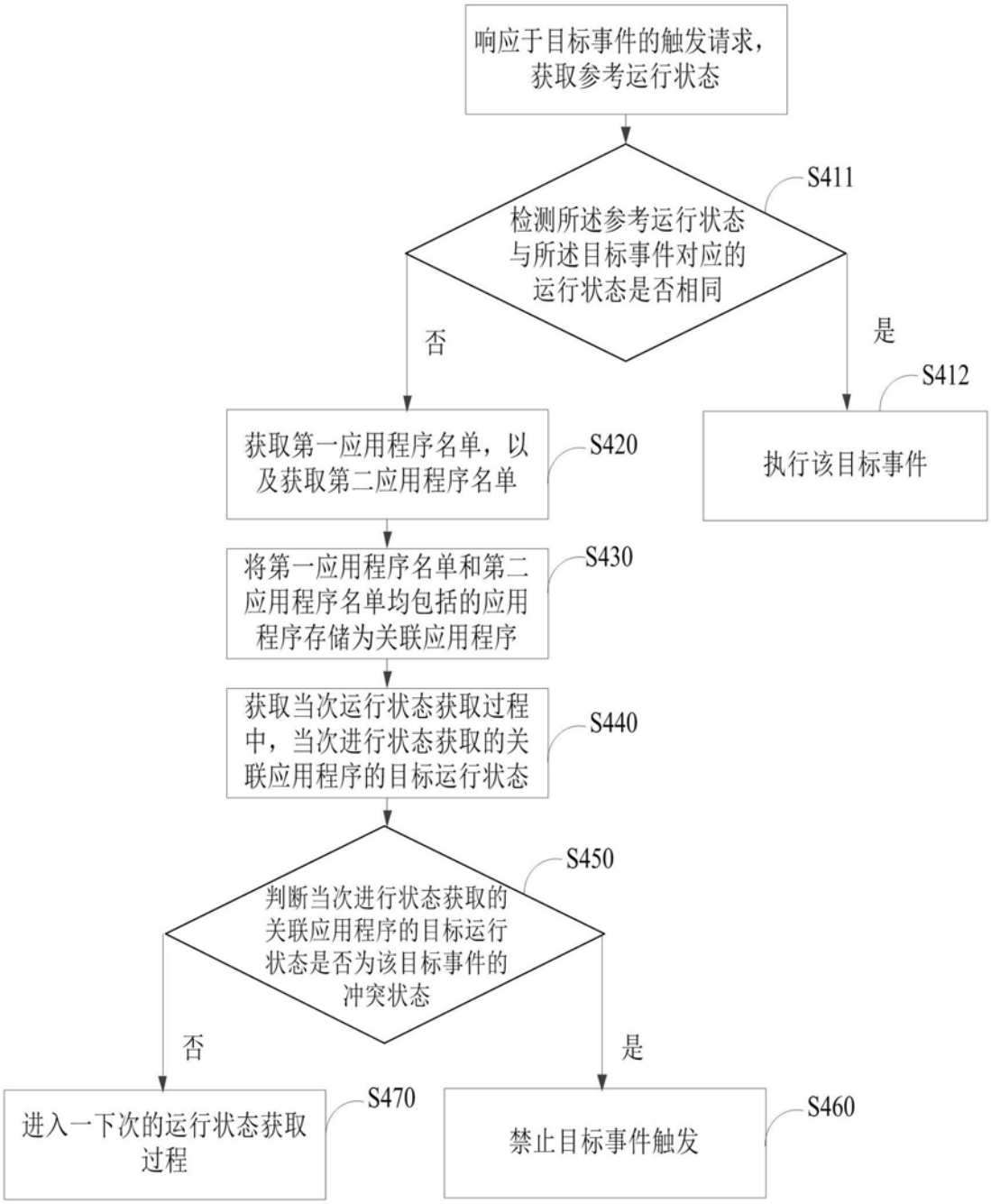

响应于目标事件的触发请求，获取参考运行状态
S411
检测所述参考运行状态与所述目标事件对应的运行状态是否相同
否
是
S412
执行该目标事件
获取第一应用程序名单，以及获取第二应用程序名单
S420
将第一应用程序名单和第二应用程序名单均包括的应用程序存储为关联应用程序
S430
获取当次运行状态获取过程中，当次进行状态获取的关联应用程序的目标运行状态
S440
判断当次进行状态获取的关联应用程序的目标运行状态是否为该目标事件的冲突状态
S450
否
是
进入一下次的运行状态获取过程
S470
禁止目标事件触发
S460

图8

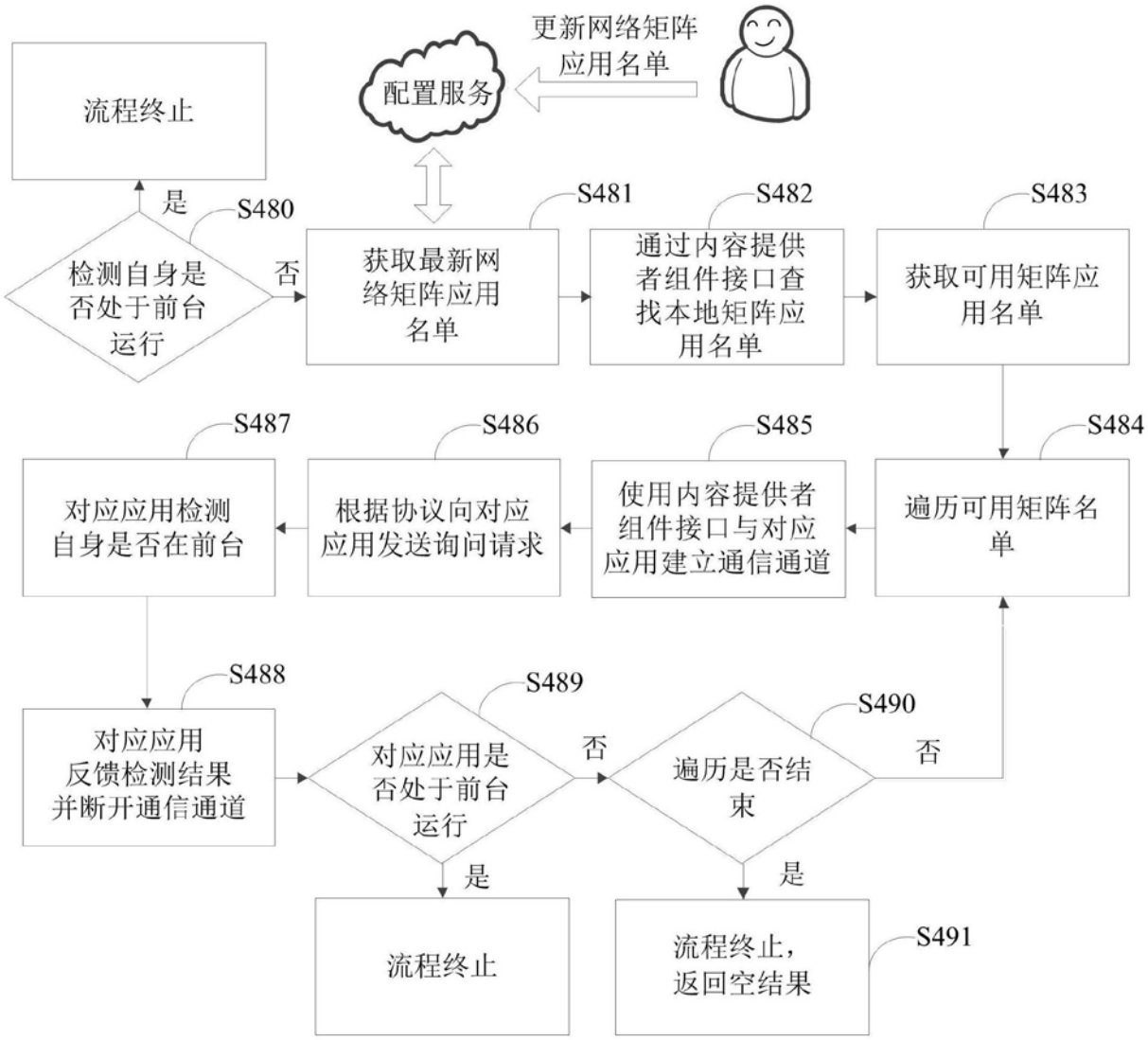

更新网络矩阵
应用名单
配置服务
流程终止
是
S480
检测自身是
否处于前台
运行
否
S481
获取最新网
络矩阵应用
名单
S482
通过内容提供
者组件接口查
找本地矩阵应
用名单
S483
获取可用矩阵应
用名单
S484
遍历可用矩阵名
单
S485
使用内容提供者
组件接口与对应
应用建立通信通道
S486
根据协议向对应
应用发送询问请求
S487
对应应用检测
自身是否在前台
S488
对应应用
反馈检测结果
并断开通信通道
S489
对应应用是
否处于前台
运行
否
S490
遍历是否结
束
否
是
流程终止
是
流程终止，
返回空结果
S491

图9

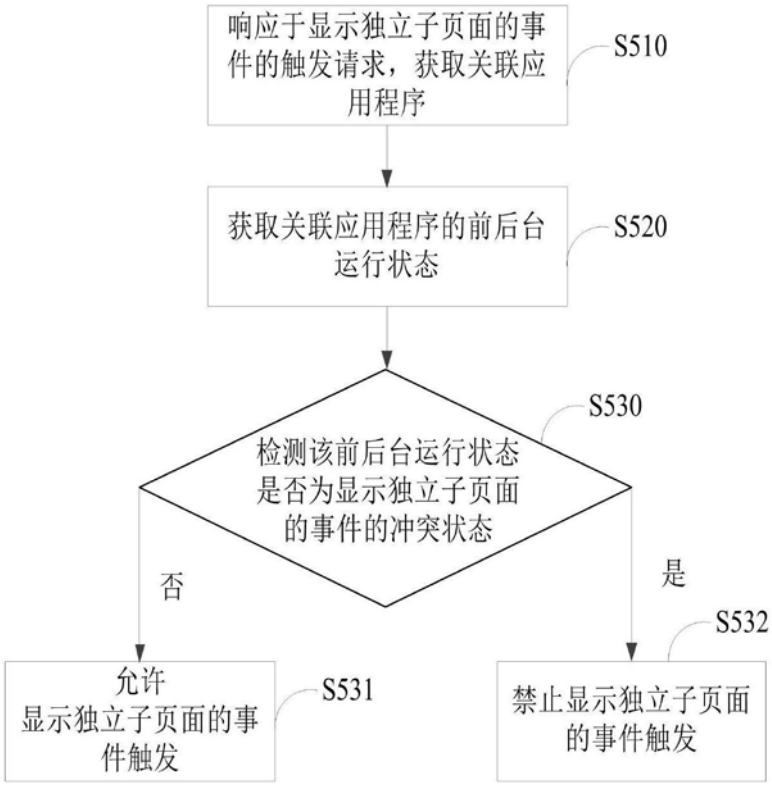

响应于显示独立子页面的事件的触发请求，获取关联应用程序
S510
获取关联应用程序的前后台运行状态
S520
检测该前后台运行状态是否为显示独立子页面的事件的冲突状态
S530
否
是
允许显示独立子页面的事件触发
S531
S532
禁止显示独立子页面的事件触发

图10

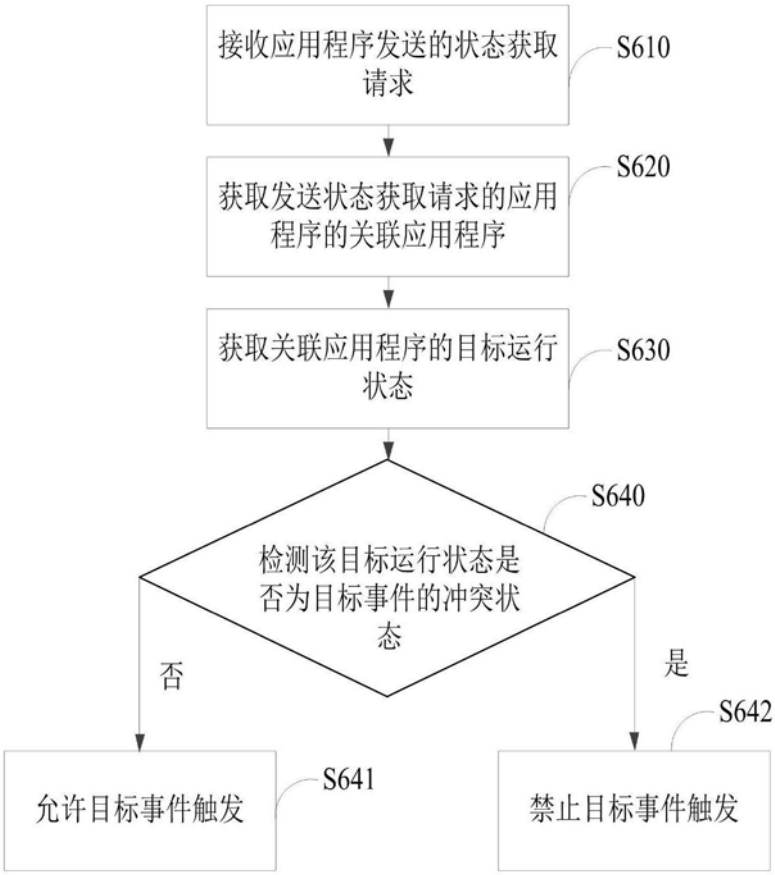

接收应用程序发送的状态获取请求
S610
获取发送状态获取请求的应用程序的关联应用程序
S620
获取关联应用程序的目标运行状态
S630
检测该目标运行状态是否为目标事件的冲突状态
S640
否
是
允许目标事件触发
S641
禁止目标事件触发
S642

图11


装置
700
关联程序获取单元
710
运行状态获取单元
720
程序控制单元
730

图12

图13


装置
800
请求管理单元
810
关联程序获取单元
820
运行状态获取单元
830
程序控制单元
840

图14

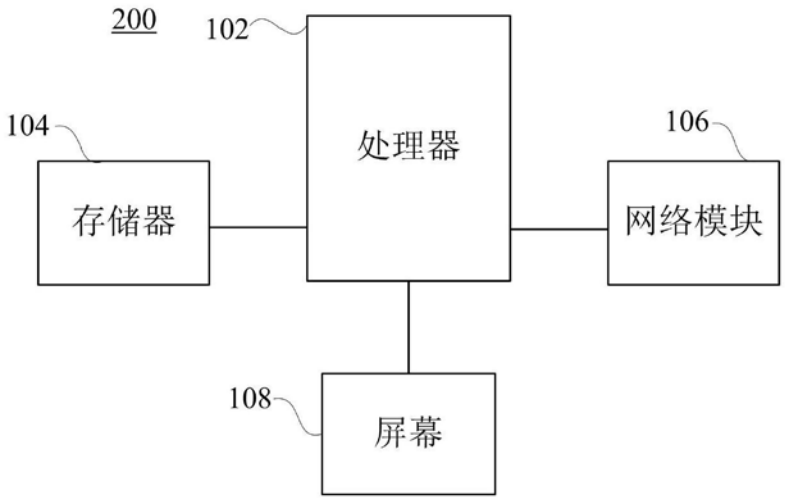

200
102
处理器
104
存储器
106
网络模块
108
屏幕

图15

图16