



(12) 发明专利

(10) 授权公告号 CN 112559098 B

(45) 授权公告日 2022. 04. 26

(21) 申请号 201910919665.5

G06F 16/16 (2019.01)

(22) 申请日 2019.09.26

G06T 15/00 (2011.01)

(65) 同一申请的已公布的文献号  
申请公布号 CN 112559098 A

(56) 对比文件  
CN 110209444 A, 2019.09.06

(43) 申请公布日 2021.03.26

审查员 王菲

(73) 专利权人 华为技术有限公司  
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72) 发明人 李军 杨昌俊 余枝强

(74) 专利代理机构 北京同达信恒知识产权代理有限公司 11291  
代理人 赵玲

(51) Int. Cl.  
G06F 9/451 (2018.01)  
G06F 16/11 (2019.01)

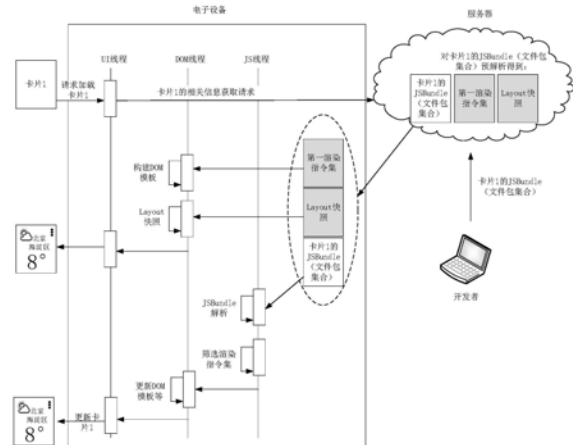
权利要求书2页 说明书16页 附图12页

(54) 发明名称

一种卡片渲染方法及电子设备

(57) 摘要

一种卡片渲染方法及电子设备,涉及终端技术领域。该方法包括:电子设备向服务器发送目标卡片的相关信息获取请求;并接收服务器发送的目标卡片的相关信息。电子设备根据第一布局快照,对目标卡片进行初步渲染,并在显示屏上显示目标卡片;并对目标卡片的第一文件包集合进行解析,得到第二渲染指令集;筛选出第二渲染指令集中除第一渲染指令集以外的渲染指令;根据筛选出的渲染指令,对目标卡片进行再次渲染,更新在显示屏上显示的目标卡片。这种技术方案有助于使得电子设备能够快速向用户呈现卡片,提高用户体验。



1. 一种卡片渲染的方法,其特征在于,所述方法包括:  
电子设备向服务器发送目标卡片的相关信息获取请求;  
所述电子设备接收所述服务器发送的所述目标卡片的相关信息,所述目标卡片的相关信息包括所述目标卡片的第一文件包集合、第一渲染指令集和第一布局快照;  
所述电子设备根据所述第一布局快照,对所述目标卡片进行初步渲染,并在显示屏上显示初步渲染后的所述目标卡片;  
所述电子设备对所述目标卡片的第一文件包集合进行解析,得到第二渲染指令集;  
所述电子设备筛选出所述第二渲染指令集中除所述第一渲染指令集以外的渲染指令;  
所述电子设备根据所述筛选出的渲染指令,对所述目标卡片进行再次渲染,更新在所述显示屏上显示的所述目标卡片。
2. 如权利要求1所述的方法,其特征在于,所述电子设备根据所述第一布局快照,对所述目标卡片进行初步渲染,包括:  
所述电子设备根据所述第一渲染指令集,构建第一DOM模板;  
所述电子设备根据所述第一DOM模板和所述第一布局快照,对所述目标卡片进行初步渲染。
3. 如权利要求1所述的方法,其特征在于,所述目标卡片的相关信息还包括第一DOM模板;  
所述电子设备根据所述第一布局快照,对所述目标卡片进行初步渲染,包括:  
所述电子设备根据所述第一DOM模板和所述第一布局快照,对所述目标卡片进行初步渲染。
4. 如权利要求1至3任一所述的方法,其特征在于,所述电子设备向服务器发送目标卡片的相关信息获取请求,包括:  
所述电子设备接收到加载所述目标卡片的请求、或者搜索所述目标卡片的请求时,向所述服务器发送目标卡片的相关信息获取请求。
5. 如权利要求1至3任一所述的方法,其特征在于,所述第一渲染指令集包括所述第二渲染指令集中构建DOM模板的起始指令和构建DOM模板的结束指令之间的渲染指令,且包括所述构建DOM模板的起始指令和所述构建DOM模板的结束指令。
6. 如权利要求4所述的方法,其特征在于,所述第一渲染指令集包括所述第二渲染指令集中构建DOM模板的起始指令和构建DOM模板的结束指令之间的渲染指令,且包括所述构建DOM模板的起始指令和所述构建DOM模板的结束指令。
7. 如权利要求1至3任一所述的方法,其特征在于,所述方法还包括:  
所述电子设备保存所述第二渲染指令集、和所述第一布局快照。
8. 如权利要求4所述的方法,其特征在于,所述方法还包括:  
所述电子设备保存所述第二渲染指令集、和所述第一布局快照。
9. 如权利要求5所述的方法,其特征在于,所述方法还包括:  
所述电子设备保存所述第二渲染指令集、和所述第一布局快照。
10. 如权利要求7所述的方法,其特征在于,所述方法还包括:  
所述电子设备再次接收到加载所述目标卡片的请求、或者搜索所述目标卡片的请求,向所述服务器发送目标卡片的文件包集合获取请求;  
所述电子设备根据预先存储的所述第二渲染指令集、和所述第一布局快照,对所述目

标卡片进行渲染,并在显示屏上显示所述目标卡片;

所述电子设备接收所述服务器发送的第二文件包集合;

所述电子设备对所述第二文件包集合解析,得到所述目标卡片的第三渲染指令集;

所述电子设备根据所述第三渲染指令集,校验所述第二渲染指令集;

所述电子设备当所述第二渲染指令集校验失败时,根据所述第三渲染指令集,重新对所述目标卡片进行渲染,并在所述显示屏上显示重新渲染后的所述目标卡片。

11. 如权利要求10所述的方法,其特征在于,所述电子设备当所述第二渲染指令集校验失败时,还包括:

所述电子设备删除所述第二渲染指令集、和所述第一布局快照,存储所述第三渲染指令集、以及根据所述第三渲染指令集得到的布局快照。

12. 一种电子设备,其特征在于,所述电子设备包括处理器和存储器;

所述存储器中存储有程序指令;

当所述程序指令被执行时,使得所述电子设备执行如权利要求1至11任一所述的方法。

13. 一种芯片,其特征在于,所述芯片与电子设备中的存储器耦合,使得所述芯片在运行时调用所述存储器中存储的程序指令,实现如权利要求1至11任一所述的方法。

14. 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包括程序指令,当所述程序指令在设备上运行时,使得所述设备执行如权利要求1至11任一项所述的方法。

## 一种卡片渲染方法及电子设备

### 技术领域

[0001] 本申请涉及终端技术领域,特别涉及一种卡片渲染方法及电子设备。

### 背景技术

[0002] 卡片是一种新的服务形态,可以直接向用户呈现用户关心或感兴趣的信息,更好的满足用户的需求。例如,对于天气类卡片,可以向用户呈现用户所在位置的天气信息。再例如,对于股票类卡片,可以向用户呈现用户已购买的股票、或者感兴趣的股票的信息。

[0003] 其中,电子设备对卡片渲染完成后才能够向用户呈现卡片上的信息。而现有技术中,电子设备基于JS-Native架构至少需要依次经历对卡片的文件包集(JSBundle)解析、文档对象模型(document object model,DOM)模板构建、以及布局计算三个步骤,才能实现对卡片渲染,呈现给用户,渲染效率较低。因此,针对如何渲染卡片,使得电子设备能够快速向用户呈现卡片上的信息的研究,对于提高用户体验有重要意义。

### 发明内容

[0004] 本申请实施例提供一种卡片渲染方法及电子设备,使得电子设备能够快速呈现卡片,有助于提高用户体验。

[0005] 第一方面,为本申请实施例的一种卡片渲染的方法,该方法包括:

[0006] 电子设备向服务器发送目标卡片的相关信息获取请求;并接收所述服务器发送的所述目标卡片的相关信息,所述目标卡片的相关信息包括所述目标卡片的第一文件包集合、第一渲染指令集和第一布局快照;然后,电子设备根据所述第一布局快照,对所述目标卡片进行初步渲染,并在显示屏上显示初步渲染后的所述目标卡片;并对所述目标卡片的第一文件包集合进行解析,得到第二渲染指令集;筛选出所述第二渲染指令集中除所述第一渲染指令集以外的渲染指令;最后,根据所述筛选出的渲染指令,对所述目标卡片进行再次渲染,更新在所述显示屏上显示的所述目标卡片。

[0007] 本申请实施例中,服务器由于可以将第一渲染指令集和第一布局快照发送给电子设备,使得电子设备可以先根据第一渲染指令集和第一布局快照先对卡片渲染,使得电子设备在显示屏上呈现卡片,减少了解析文件包集合、以及布局计算的步骤,因而提高了卡片渲染的效率,从而有助于使得电子设备能够快速向用户呈现卡片,提高用户体验。

[0008] 在一种可能的设计中,所述电子设备根据所述第一渲染指令集,构建第一文档对象模型DOM模板;并根据所述第一DOM模板和所述第一布局快照,对所述目标卡片进行初步渲染。有助于简化实现方式。

[0009] 在一种可能的设计中,所述目标卡片的相关信息还包括第一DOM模板;所述电子设备根据所述第一DOM模板和第一布局快照,对所述目标卡片进行初步渲染。有助于进一步提高电子设备向用户呈现卡片的速度。

[0010] 在一种可能的设计中,所述电子设备接收到加载目标卡片的请求时,向服务器发送目标卡片的相关信息获取请求。有助于简化实现方式。

[0011] 在一种可能的设计中,所述电子设备接收到搜索目标卡片的请求时,向服务器发送目标卡片的相关信息获取请求。有助于简化实现方式。

[0012] 在一种可能的设计中,所述第一渲染指令集包括所述第二渲染指令集中构建DOM模版的起始指令和构建DOM模版的结束指令之间的渲染指令,且包括所述构建DOM模版的起始指令和构建DOM模版的结束指令。有助于降低实现的复杂性。

[0013] 在一种可能的设计中,所述电子设备保存所述第二渲染指令集、和所述第一布局快照。

[0014] 在一种可能的设计中,所述电子设备再次接收到加载所述目标卡片的请求、或者搜索所述目标卡片的请求,向所述服务器发送目标卡片的文件包集合获取请求;并在接收到加载所述目标卡片的请求时,根据预先存储的所述第二渲染指令集、和所述第一布局快照,对所述目标卡片进行渲染,并在显示屏上显示所述目标卡片;接收所述服务器发送的第二文件包集合;并对所述第二文件包集合解析,得到所述目标卡片的第三渲染指令集;根据所述第三渲染指令集,校验所述第二渲染指令集;并当所述第二渲染指令集校验失败时,根据所述第三渲染指令集,重新对所述目标卡片进行渲染,并在所述显示屏上显示重新渲染后的所述目标卡片。

[0015] 由于电子设备可以根据预先存储的目标卡片的第一相关信息对目标卡片进行渲染,与电子设备根据从服务器获取的文件包集合对卡片进行渲染相比,可以提高电子设备在显示屏上呈现卡片的速度,提高用户体验。

[0016] 在一种可能的设计中,所述电子设备当所述第二渲染指令集校验失败时,删除所述第二渲染指令集、和所述第一布局快照,存储所述第三渲染指令集、以及根据所述第三渲染指令集得到的布局快照。有助于提高卡片渲染的可靠性。

[0017] 第二方面,本申请实施例的另一卡片渲染的方法,该方法包括:

[0018] 电子设备向服务器发送目标卡片的文件包集合获取请求;根据预先存储的所述目标卡片的第一相关信息,对所述目标卡片进行渲染,并在显示屏上显示所述目标卡片;所述第一相关信息是基于第一文件包集合得到的,所述第一文件包集合为所述电子设备之前渲染所述目标卡片所使用的文件包集合;所述电子设备在接收所述服务器发送的第二文件包集合后,根据所述第二文件包集合,得到所述目标卡片的第二相关信息;

[0019] 所述电子设备根据所述第二相关信息,校验所述第一相关信息;

[0020] 所述电子设备当所述第一相关信息校验失败时,根据所述第二相关信息,重新对所述目标卡片进行渲染,并在所述显示屏上显示重新渲染后的所述目标卡片。

[0021] 本申请实施例中,由于电子设备可以根据预先存储的目标卡片的第一相关信息对目标卡片进行渲染,与电子设备根据从服务器获取的文件包集合对卡片进行渲染相比,可以提高电子设备在显示屏上呈现卡片的速度,提高用户体验。

[0022] 在一种可能的设计中,所述电子设备当所述第一相关信息校验失败时,删除所述第一相关信息,存储所述第二相关信息。有助于提高电子设备呈现卡片的可靠性。

[0023] 在一种可能的设计中,所述第一相关信息包括:第一渲染指令集;所述电子设备对所述第二文件包集合解析,得到第二渲染指令集;

[0024] 所述电子设备判断所述第一渲染指令集与所述第二渲染指令集是否相同;所述第一渲染指令集与所述第二渲染指令集相同时,所述第一渲染指令集校验成功;所述第一渲

染直联机与所述第二渲染指令集不同时,所述第一渲染指令集校验失败。有助于简化第一相关信息的校验方式。

[0025] 在一种可能的设计中,所述电子设备接收到加载目标卡片的请求或者搜索目标卡片的请求时,向所述服务器发送的目标卡片的相关信息获取请求。有助于简化实现方式。

[0026] 第三方面,本申请实施例提供的一种电子设备,包括执行本申请实施例上述各个方面以及各个方面涉及的任一可能设计的方法的装置。

[0027] 第四方面,本申请实施例提供的一种芯片,包括:处理器和接口,用于从存储器中调用并运行所述存储器中存储的程序指令,执行本申请实施例上述各个方面以及各个方面涉及的任一可能设计的方法。

[0028] 第五方面,本申请实施例的一种计算机存储介质,该计算机存储介质存储有程序指令,当所述程序指令在电子设备上运行时,使得电子设备执行本申请实施例上述各个方面以及各个方面涉及的任一可能设计的方法。

[0029] 第六方面,本申请实施例的一种计算机程序产品,当所述计算机程序产品在电子设备上运行时,使得所述电子设备执行实现本申请实施例上述各个方面以及各个方面涉及的任一可能设计的方法。

[0030] 另外,第三方面至第六方面中任一种可能设计方式所带来的技术效果可参见方法部分相关中不同设计方式所带来的技术效果,此处不再赘述。

## 附图说明

[0031] 图1为本申请实施例的一种JS-Native架构的示意图;

[0032] 图2为本申请实施例的一种卡片渲染方法的流程示意图;

[0033] 图3为本申请实施例的一种电子设备的硬件架构示意图;

[0034] 图4为本申请实施例的另一卡片渲染方法的流程示意图;

[0035] 图5为本申请实施例的一种界面的示意图;

[0036] 图6为本申请实施例的一种卡片设置界面的示意图;

[0037] 图7为本申请实施例的另一界面的示意图;

[0038] 图8为本申请实施例的另一界面的示意图;

[0039] 图9为本申请实施例的渲染指令集的关系示意图;

[0040] 图10为本申请实施例的另一卡片的示意图;

[0041] 图11为本申请实施例的另一卡片渲染方法的流程示意图;

[0042] 图12为本申请实施例的另一卡片渲染方法的流程示意图;

[0043] 图13为本申请实施例的另一卡片渲染方法的流程示意图;

[0044] 图14为本申请实施例的另一卡片渲染方法的流程示意图;

[0045] 图15为本申请实施例的电子设备的结构示意图;

[0046] 图16为本申请实施例的另一电子设备的结构示意图。

## 具体实施方式

[0047] 应理解,本申请中除非另有说明,“/”表示或的意思。例如,A/B可以表示A或B。本申请中的“和/或”仅仅是一种描述关联对象的关联关系,表示可以存在三种关系。例如,A和/

或B,可以表示:单独存在A,同时存在A和B,单独存在B三种情况。“至少一个”是指一个或者多个,“多个”是指两个或两个以上。

[0048] 在本申请中,“示例的”、“在一些实施例中”、“在另一些实施例中”等用于表示作例子、例证或说明。本申请中被描述为“示例”的任何实施例或设计方案不应被解释为比其它实施例或设计方案更优选或更具优势。确切而言,使用示例的一词旨在以具体方式呈现概念。

[0049] 另外,本申请中涉及的“第一”、“第二”等词汇,仅用于区分描述的目的,而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量,也不能理解为指示或暗示顺序。

[0050] 首先,对本申请涉及相关术语进行相应的解释,以便于本领域技术人员理解。

[0051] 1、电子设备。本申请实施例中的电子设备可以为便携式终端,比如手机、平板电脑、可穿戴设备、增强现实(augmented reality,AR)/虚拟现实(virtual reality,VR)设备等。具体的,电子设备的示例性实施例包括但不限于搭载IOS®、Android®、Microsoft®或者其它操作系统。在另一些实施例中,电子设备还可以为智慧屏、笔记本电脑、车载终端等。

[0052] 2、卡片。卡片是一种电子设备向用户呈现信息的形式,可以包括有关同一个主题的图片、文本、链接、控件等信息。例如,天气类卡片、股票类卡片和新闻类卡片等。在一些实施例中,卡片还可以为与卡片对应的应用程序(application,APP)的入口,用户可以通过对卡片操作,打开与卡片对应的应用程序,使得电子设备向用户呈现与该卡片对应的应用程序的界面,从而使得用户可以在该界面上查看更详细的信息。进一步的,用户还可以在打开卡片对应的应用程序后,在应用程序的界面上进行相应的操作,来满足自身的需求。例如,天气类卡片对应的应用程序为天气。用户可以通过对天气类卡片的操作,打开应用程序天气,使得电子设备向用户呈现天气的界面。用户可以通过对天气的界面进行操作,可以设置天气类卡片上显示的信息、或者查看某一城市的天气情况等,来满足自身的需求。需要说明的是,与卡片对应的应用程序可以安装在电子设备上,也可以未安装在电子设备上,对此不作限定。

[0053] 3、宿主应用。电子设备上用于呈现卡片的应用程序,例如负一屏、语音助手等。例如,负一屏上可以呈现天气类卡片、股票类卡片等。例如,电子设备可以响应于用户的语音指令,在语音助手中呈现相应的卡片。需要说明的是,宿主应用可以安装在电子设备上,也可以不安装在电子设备上,对此不作限定。

[0054] 4、文件包集合。文件包集合为用于描述卡片的文件及其相关资源的集合。文件包集合又可以称之为JSBundle。其中,不同卡片的JSBundle不同。

[0055] 5、文档对象模型(document object model,DOM)模板。DOM模块用于描述卡片上不同元素之间的关系。需要说明的是,元素可以理解为卡片上的控件、图片等信息。

[0056] 6、服务器。本申请实施例中服务器可以为电子设备、通用服务器、或者云服务器等,用于为电子设备提供卡片的相关信息,例如,文件包集合、DOM模板等。

[0057] 通常,电子设备当宿主应用请求加载某一卡片时,先要对该卡片进行渲染,并在对卡片渲染完成后才能够向用户呈现该卡片上的信息。示例的,电子设备可以基于Web架构或者JS-Native架构实现对卡片的渲染。然而,基于Web架构实现对卡片渲染的方案中,由于性能和体验等问题限制较多,因此通常采用基于JS-Native架构实现对卡片的渲染。

[0058] 其中,JS-Native架构采用JavaScript动态语言。示例的,如图1所示,为一种JS-Native架构的示意图。如图所示,JS-Native架构包括JSFramework和JS-Native引擎。JSFramework为JavaScript提供了各种功能基础。JS-Native引擎包括JS引擎(V8/JSCore)和JS-Native渲染引擎。JS引擎为JavaScript提供运行环境,JS-Native渲染引擎用于DOM模板的构建、Layout布局计算以及卡片的渲染。另外,JS-Native架构还包括用户界面(user interface,UI)线程、DOM线程和JS线程。具体的,UI线程可用于渲染卡片,DOM线程可用于构建DOM模板、Layout布局计算等,JS线程可用于JSBundle的解析。

[0059] 以卡片1为例,对电子设备进行卡片渲染的方法进行介绍。示例的,如图2所示,电子设备在UI线程对JS-Native引擎初始化后,当宿主应用请求加载卡片1时,向服务器发送卡片1的文件包集合获取请求。服务器在接收到卡片1的文件包集合获取请求后,将卡片1的JSBundle返回给电子设备。电子设备从服务器获取到卡片1的JSBundle后,将卡片1的JSBundle传递给JSFramework。在JS线程上,由JSFramework对JSBundle进行解析,得到卡片1的渲染指令集。卡片1的渲染指令集包括多个用于渲染卡片1的渲染指令。然后,在DOM线程上,根据卡片1的渲染指令集,构建DOM模板,并进行Layout布局计算,得到Layout快照文件。最后,电子设备在UI线程上,根据DOM模板和Layout快照文件,对卡片1进行渲染,向用户呈现卡片1。

[0060] 换句话说,电子设备需要至少经过JSBundle解析、DOM模板构建以及Layout布局计算的三个步骤,才能实现对卡片的渲染,呈现给用户。因此,上述卡片渲染方法的耗时较长,容易导致电子设备向用户呈现卡片的速度较慢。尤其是在宿主应用首次请求加载某一卡片、或者宿主应用请求加载多张卡片时,电子设备向用户呈现卡片的速度较慢。其中,其中,宿主应用请求加载卡片越多,卡顿现象可能会更为严重。

[0061] 有鉴于此,本申请实施例提供了一种卡片渲染的方法,使得电子设备能够快速向用户呈现卡片,减少电子设备的卡顿,提高用户体验。

[0062] 以下介绍电子设备、和用于使用这样的电子设备的实施例。示例的,如图3所示,为本申请实施例的一种电子设备的硬件结构示意图。如图所示,电子设备包括处理器110、内部存储器121、外部存储器接口122、摄像头130、显示屏140、传感器模块150、音频模块160、扬声器161、受话器162、麦克风163、耳机接口164、用户标识模块(subscriber identification module,SIM)卡接口171、通用串行总线(universal serial bus,USB)接口172、充电管理模块180、电源管理模块181、电池182、移动通信模块191和无线通信模块192。此外,在另一些实施例中,电子设备还可以包括马达、指示器和按键等。

[0063] 应理解,图3所示的硬件结构仅是一个示例。本申请实施例的电子设备可以具有比图中所示电子设备更多的或者更少的部件,可以组合两个或更多的部件,或者可以具有不同的部件配置。图中所示出的各种部件可以在包括一个或多个信号处理和/或专用集成电路在内的硬件、软件、或硬件和软件的组合中实现。

[0064] 其中,处理器110可以包括一个或多个处理单元。例如,处理器110可以包括应用处理器(application processor,AP)、调制解调器、图形处理器(graphics processing unit,GPU)、图像信号处理器(image signal processor,ISP)、视频编解码器、数字信号处理器(digital signal processor,DSP)、基带处理器、和/或神经网络处理器(neural-network processing unit,NPU)等。在具体实现时,不同的处理单元可以是独立的器件,也



可以集成在一个或多个器件中。

[0065] 在一些实施例中,处理器110中还可以设置缓存器,用于存储程序和/或数据。需要说明的是,本申请实施例中的程序又可以称之为程序指令、计算机程序、代码指令、指令等,对此不作限定。示例的,处理器110中的缓存器可以为高速缓冲存储器。该缓存器可以用于保存处理器110刚用过的、生成的、或循环使用的程序和/或数据。如果处理器110需要使用该程序和/或数据,可从该缓存器中直接调用。有助于减少处理器110获取程序或数据的时间,从而提高系统的效率。

[0066] 内部存储器121可以用于存储程序和/或数据。在一些实施例中,内部存储器121包括存储程序区和存储数据区。其中,存储程序区可以用于存储操作系统(如Android、IOS等操作系统)、至少一个功能所需的程序(比如显示功能)等。存储数据区可以用于存储电子设备使用过程中所创建、预先设置和/或获取的数据(比如JSBundle、Layout快照文件)等。示例的,处理器110可以通过调用内部存储器121中存储的程序和/或数据,使得电子设备执行相应的方法,从而实现一种或多种功能。例如,处理器110调用内部存储器中的某些程序和/或数据,使得电子设备执行本申请实施例中所提供的卡片渲染方法,以使得电子设备能够快速向用户呈现卡片,有助于提高用户体验。其中,内部存储器121可以采用高速随机存取存储器、和/或非易失性存储器等。例如,非易失性存储器可以包括一个或多个磁盘存储器件、闪存器件、和/或通用闪存存储器(universal flash storage,UFS)等中的至少一个。

[0067] 外部存储器接口122可以用于连接外部存储卡(例如,Micro SD卡),实现扩展电子设备的存储能力。外部存储卡通过外部存储器接口122与处理器110通信,实现数据存储功能。例如,电子设备可以通过外部存储器接口122将图像、音乐、视频、文档等内容保存在外部存储卡中。

[0068] 摄像头130可以用于捕获动、静态图像等。通常情况下,摄像头130包括镜头和图像传感器。其中,物体通过镜头生成的光学图像投射到图像传感器上,然后转换为电信号,在进行后续处理。示例的,图像传感器可以是电荷耦合器件(charge coupled device,CCD)或互补金属氧化物半导体(complementary metal-oxide-semiconductor,CMOS)光电晶体管。图像传感器把光信号转换成电信号,之后将电信号传递给ISP转换成数字图像信号。需要说明的是,本申请实施例中,电子设备可以包括一个或多个摄像头130,对此不作限定。示例的,电子设备包括5个摄像头130,例如,3个后置摄像头和2个前置摄像头。又示例的,电子设备包括3个摄像头130,例如2个后置摄像头和1个前置摄像头。

[0069] 显示屏140可以包括显示面板。用户可以根据自身的需要,使得显示屏140上显示不同的界面,从而满足用户的需求。具体的,显示面板可以采用液晶显示屏(liquid crystal display,LCD)、有机发光二极管(organic light-emitting diode,OLED)、有源矩阵有机发光二极体或主动矩阵有机发光二极体(active-matrix organic light emitting diode,AMOLED)、柔性发光二极管(flex light-emitting diode,FLED)、Miniled、MicroLed、Micro-oLed、量子点发光二极管(quantum dot light emitting diodes,QLED)等。示例的,电子设备100可以通过GPU、显示屏140、应用处理器等实现显示功能。需要说明的是,本申请实施例中,电子设备可以包括一个或多个显示屏140,对此不作限定。其中,显示屏140可以为可折叠屏幕,也可以为不可折叠屏幕,对此不作限定。

[0070] 传感器模块150可以包括一个或多个传感器。例如,触摸传感器150A、压力传感器

150B等。在另一些实施例中,传感器模块150还可以包括陀螺仪、加速度传感器、指纹传感器、环境光传感器、距离传感器、接近光传感器、骨传导传感器、温度传感器、定位传感器(如全球定位系统(global positioning system,GPS)传感器)等中的一个或多个,对此不作限定。

[0071] 触摸传感器150A也可称为“触控面板”。触摸传感器150A可以设置于显示屏140。当触摸传感器150A设置于显示屏140时,由触摸传感器150A与显示屏140组成触摸屏,也可以称为“触控屏”。触摸传感器150A用于检测作用于其上或附近的触摸操作。触摸传感器150A可以将检测到的触摸操作传递给应用处理器,以确定触摸事件类型。电子设备可以通过显示屏140提供与触摸操作相关的视觉输出等。例如,电子设备可以响应于触摸器150A检测到作用于其上或附近的触摸操作,进行界面切换,并在显示屏140上显示切换后的界面。在另一些实施例中,触摸传感器150A也可以设置于电子设备的表面,与显示屏140所处的位置不同。

[0072] 压力传感器150B用于感受压力信号,可以将压力信号转换成电信号。示例的,压力传感器150B可以设置于显示屏140。其中,作用于相同触摸位置,但不同触摸操作强度的触摸操作,可以对应不同的操作指令。

[0073] 电子设备可以通过音频模块160、扬声器161、受话器162、麦克风163、耳机接口164以及应用处理器等实现音频功能。例如,音频播放功能、录音功能、语音唤醒功能等。

[0074] 音频模块160可以用于对音频数据进行数模转换、和/或模数转换,还可以用于对音频数据进行编码和/或解码。示例的,音频模块160可以设置于处理器110中,或将音频模块160的部分功能模块设置于处理器110中。

[0075] 扬声器161,也称“喇叭”,用于将音频数据转换为声音,并播放声音。例如,电子设备100可以通过扬声器161收听音乐、接听免提电话、或者发出语音提示等。

[0076] 受话器162,也称“听筒”,用于将音频数据转换成声音,并播放声音。例如,当电子设备接听电话时,可以通过将受话器162靠近人耳进行接听。

[0077] 麦克风163,也称“话筒”、“传声器”,用于采集声音(例如周围环境声音,包括人发出的声音、设备发出的声音等),并将声音转换为音频电数据。当拨打电话或发送语音时,用户可以通过人嘴靠近麦克风163发出声音,麦克风163采集用户发出的声音。需要说明的是,电子设备100可以设置至少一个麦克风163。例如,电子设备中设置两个麦克风163,除了采集声音,还可以实现降噪功能。又示例如,电子设备中还可以设置三个、四个或更多个麦克风163,从而可以在实现声音采集、降噪的基础上,还可以实现声音来源的识别、或定向录音功能等。

[0078] 耳机接口164用于连接有线耳机。耳机接口164可以是USB接口170,也可以是3.5mm的开放移动电子设备平台(open mobile terminal platform,OMTP)标准接口、美国蜂窝电信工业协会(cellular telecommunications industry association of the USA,CTIA)标准接口等。

[0079] SIM卡接口171用于连接SIM卡。SIM卡可以通过插入SIM卡接口171,或从SIM卡接口115拔出,实现和电子设备的接触和分离。电子设备可以支持1个或N个SIM卡接口,N为大于1的正整数。SIM卡接口171可以支持Nano SIM卡、Micro SIM卡、SIM卡等。同一个SIM卡接口171可以同时插入多张卡。所述多张卡的类型可以相同,也可以不同。SIM卡接口171也可以

兼容不同类型的SIM卡。SIM卡接口171也可以兼容外部存储卡。电子设备通过SIM卡和网络交互,实现语音通话、视频通话以及数据通信等功能。在一些实施例中,电子设备采用eSIM,即:嵌入式SIM卡。eSIM卡可以嵌在电子设备中,不能和电子设备分离。

[0080] USB接口172是符合USB标准规范的接口,具体可以是Mini USB接口、Micro USB接口、USB Type C接口等。USB接口172可以用于连接充电器为电子设备充电,也可以用于电子设备与外围设备之间传输数据。也可以用于连接耳机,通过耳机播放声音。示例的,USB接口172除了可以为耳机接口164以外,还可以用于连接其他电子设备,例如AR设备、计算机等。

[0081] 充电管理模块180用于从充电器接收充电输入。其中,充电器可以是无线充电器,也可以是有线充电器。在一些有线充电的实施例中,充电管理模块180可以通过USB接口170接收有线充电器的充电输入。在一些无线充电的实施例中,充电管理模块180可以通过电子设备的无线充电线圈接收无线充电输入。充电管理模块180为电池182充电的同时,还可以通过电源管理模块180为电子设备供电。

[0082] 电源管理模块181用于连接电池182、充电管理模块180与处理器110。电源管理模块181接收电池182和/或充电管理模块180的输入,为处理器110、内部存储器121、摄像头130、显示屏140等供电。电源管理模块181还可以用于监测电池容量、电池循环次数、电池健康状态(漏电、阻抗)等参数。在其他一些实施例中,电源管理模块181也可以设置于处理器110中。在另一些实施例中,电源管理模块181和充电管理模块180也可以设置于同一个器件中。

[0083] 移动通信模块191可以提供应用在电子设备上的包括2G/3G/4G/5G等无线通信的解决方案。移动通信模块191可以包括滤波器、开关、功率放大器、低噪声放大器(low noise amplifier,LNA)等。

[0084] 无线通信模块192可以提供应用在电子设备上的包括无线局域网(wireless local area networks,WLAN)(如无线保真(wireless fidelity,Wi-Fi)网络)、蓝牙(Bluetooth,BT)、全球导航卫星系统(global navigation satellite system,GNSS)、调频(frequency modulation,FM)、近距离无线通信技术(near field communication,NFC)、红外技术(infrared,IR)等无线通信的解决方案。无线通信模块192可以是集成至少一个通信处理模块的一个或多个器件。

[0085] 在一些实施例中,电子设备的天线1和移动通信模块191耦合,天线2和无线通信模块192耦合,使得电子设备可以与其他设备通信。具体的,移动通信模块191可以通过天线1与其它设备通信,无线通信模块191可以通过天线2与其它设备通信。

[0086] 以下实施例均可以在具有上述硬件结构的电子设备中实现。下面以对卡片1为例,对本申请实施例的卡片渲染的方法进行相应的介绍。

[0087] 示例一:

[0088] 如图4所示,为本申请实施例的一种卡片渲染的方法的流程示意图,具体包括以下步骤。

[0089] 步骤401,电子设备向服务器发送卡片1的相关信息获取请求。

[0090] 在一些实施例中,电子设备接收到加载卡片1的请求时,向服务器发送卡片1的相关信息获取请求。示例的,电子设备可以响应于打开宿主应用的操作,使得宿主应用向电子设备发送加载卡片1的请求。例如,以宿主应用为负一屏为例。电子设备当前在显示屏上显

示主界面,例如,主界面如图5中的A所示的界面510。电子设备检测到用户在主界面上向左滑动的操作,响应于该操作,在显示屏上显示负一屏。例如,负一屏如图5中的B所示的界面520。如图5中的B所示的界面上包括卡片501和卡片502。为了使得电子设备在负一屏上能够正常显示卡片501和卡片502,电子设备还响应于用户在主界面上向左滑动的操作,使得负一屏请求加载卡片501和卡片502。以卡片1为卡片501为例,电子设备当负一屏请求加载卡片501时,向服务器发送卡片501的相关信息获取请求。

[0091] 示例的,电子设备在负一屏上显示的卡片可以是根据用户需要订阅的,也可以是电子设备当基于预设策略(例如用户的历史使用记录等),确定是否在负一屏上呈现某一个或多个卡片的。例如,电子设备可以通过在情境智能设置界面上根据需求订阅相应的卡片。例如,情境智能设置界面为如图6所示的界面600,包括用于控制情境智能开启或关闭的虚拟按键、一个或多个订阅选项等。其中,用于控制情境智能开启或关闭的虚拟按键开启(ON)时,用户可以根据自身需求选择相应的订阅选项。例如,用户选中的订阅选项为应用使用情况和天气,则电子设备在负一屏上显示天气卡片和应用使用情况卡片。再例如,用户当开启电子设备的情境智能时,电子设备可以在上班时间段,在负一屏上显示上班路线以及方式的卡片。电子设备还可以在购买火车票或飞机票后,在负一屏上显示出行时间和车次的卡片等。

[0092] 再例如,以宿主应用为快应用为例。电子设备在当前的显示屏上显示主界面,例如,主界面如图7中的A所示的界面710。电子设备检测到用户点击快应用图标701的操作,响应于该操作,在显示屏上显示快应用的界面。例如快应用的界面为图7中B所示的界面720。界面720上包括卡片702。为了使得电子设备在界面720上正常显示卡片702,电子设备还响应于用户点击快应用图标701的操作,使得快应用请求加载卡片702。以卡片1为卡片702为例,电子设备当接收到加载卡片702的请求时,向服务器发送卡片702的相关信息获取请求。

[0093] 以宿主应用为语音助手为例。如图8中的A所示,电子设备在显示屏上显示语音助手的界面810。电子设备接收到用户查看天气的语音指令,响应于用户查看天气的语音指令,在显示屏上显示图8中B所示的界面820。其中,界面820上包括卡片802。卡片802用于显示天气情况。为了使得电子设备在界面820上正常显示卡片802,电子设备还响应于用户查看天气的语音指令,使得语音助手请求加载卡片802。以卡片1为卡片802为例,电子设备当接收到加载卡片802的请求时,向服务器发送卡片802的相关信息获取请求。

[0094] 在另一些实施例中,电子设备接收到搜索卡片1的请求时,向服务器发送卡片1的相关信息获取请求。示例的,电子设备可以响应于用户对某一宿主应用的操作,使得宿主应用向电子设备发送搜索卡片1的请求。以图7中B所示的界面720为例,电子设备响应于在搜索框中输入天气的操作,快应用向电子设备发送搜索天气卡片的请求。

[0095] 又示例的,以电子设备为智慧屏为例。电子设备还可以响应于用户开机的操作、或者将电子设备从睡眠状态唤醒的操作,使得宿主应用向电子设备发送加载卡片1的请求,向服务器发送卡片1的相关信息获取请求。

[0096] 在另一些实施例中,电子设备还可以当检测到宿主应用的界面的刷新时,向服务器发送卡片1的相关信息获取请求。

[0097] 步骤402,服务器接收到卡片1的相关信息获取请求,向电子设备发送卡片1的相关信息。卡片1的相关信息可以包括卡片1的JSBundle、第一渲染指令集和Layout快照。第一渲

染指令集用于构建卡片1的DOM模板。在另一些实施例中,卡片1的相关信息还可以包括卡片1的DOM模板。

[0098] 需要说明的是,Layout快照,又可以称之为布局快照、快照文件、布局文件或布局数据等,为Layout布局计算的结果,又可以称之为Layout快照,用于描述卡片上个元素的布局,例如元素在卡片上的位置、大小等。

[0099] 示例的,对于服务器来说,卡片1的相关信息可以是开发者通过电子设备(例如笔记本电脑等)上传给服务器。例如,卡片1的JSBundle、第一渲染指令集、Layout快照等可以是开发者通过电子设备上传给服务器的。或者,开发者通过电子设备将卡片1的JSBundle上传给服务器,服务器在接收到卡片1的JSBundle后,对JSBundle解析,得到渲染指令集。然后从渲染指令集中筛选出第一渲染指令集,并保存;根据渲染指令集,构造DOM模板,以及进行Layout布局计算得到Layout快照,并保存Layout快照。在一些实施例中,服务器还保存DOM模块。

[0100] 其中,第一渲染指令集为渲染指令集中第一目标指令与第二目标指令之间的渲染指令,且包括第一目标指令和第二目标指令。例如,服务器可以通过判断渲染指令集中的渲染指令是否位于第一目标指令和第二目标指令之间,来筛选出第一渲染指令集。第一目标指令和第二目标指令可以是服务器根据预设策略确定的,也可以是预配置在服务器中的。例如,第一目标指令为构建DOM模板的起始指令,例如CreateBody,第二目标指令为构建DOM模板的结束指令,例如CreateFinish。

[0101] 由于服务器除了可以将卡片1的JSBundle发送给电子设备以外,还可以将第一渲染指令集、Layout快照或者DOM模板发送给电子设备,使得电子设备在卡片渲染时,可以使用第一渲染指令集、Layout快照或者DOM模板先对卡片渲染,呈现给用户,无需先JSBundle解析,从而有助于提高卡片的渲染效率。

[0102] 步骤403,电子设备接收到卡片1的相关信息后,根据第一渲染指令集构建DOM模板;并根据DOM模板以及Layout快照对卡片进行初步渲染,在显示屏上显示卡片1。

[0103] 需要说明的是,电子设备在卡片1的相关信息中包括DOM模板的情况下,电子设备接收到卡片1的相关信息后,可以根据DOM模板以及Layout快照对卡片进行初步渲染,在显示屏上显示卡片1。

[0104] 示例的,电子设备可以接收到卡片1的相关信息后,通过宿主应用的应用程序编程接口(application programming interface,API)将卡片1的第一渲染指令集、Layout快照传递给JS-Native架构中的JS-Native引擎,由JS-Native引擎在DOM线程上,根据第一渲染指令集构建DOM模板,并在UI线程上根据DOM模板以及Layout快照,对卡片进行初步渲染,呈现在显示屏上。

[0105] 步骤404,电子设备对卡片1的相关信息中的JSBundle进行解析,得到渲染指令集。

[0106] 示例的,电子设备可以接收到卡片1的相关信息后,通过宿主应用的API将卡片1的JSBundle传递给JS-Native架构中的JSFramework。由JSFramework在JS线程上,对JSBundle进行解析,得到渲染指令集。

[0107] 步骤405,电子设备从渲染指令集中筛选出除第一渲染指令集以外的渲染指令。

[0108] 需要说明的是,电子设备每次采用示例一的方式对卡片渲染时,均需要进行渲染指令集的筛选,以避免渲染出重复的卡片视图,造成内存泄露。

[0109] 示例的,第一渲染指令集和渲染指令集如图7所示。渲染指令集为电子设备对卡片1的相关信息中的JSBundle进行解析得到的。第一渲染指令集时服务器发送给电子设备的。而图9中阴影部分的渲染指令为渲染指令集中出第一渲染指令集以外的渲染指令。

[0110] 例如,第一渲染指令集为第一目标指令和第二目标指令之间的渲染指令的集合。电子设备可以判断渲染指令集中的渲染指令是否不在第一目标指令和第二目标指令之间,从而筛选出渲染指令集中除第一渲染指令集以外的渲染指令。比如,第一目标指令与第二目标指令可以是电子设备根据第一渲染指令集中起始位置和结束位置的渲染指令确定的。比如,如图7所示,第一渲染指令集的起始位置的渲染指令为CreateBody,第一渲染指令集的结束位置的渲染指令为CreateFinish。再例如,第一目标指令和第二目标指令可以是预先配置在电子设备中的。

[0111] 在一些实施例中,电子设备可以通过JSFramework在JS线程上,从渲染指令集中筛选出除第一渲染指令集以外的渲染指令。

[0112] 步骤406,电子设备根据筛选出的除第一渲染指令集以外的渲染指令,对卡片进行二次渲染,更新在显示屏上显示的卡片1。

[0113] 示例的,电子设备对卡片进行二次渲染,可以包括:电子设备基于筛选出的第一渲染指令集以外的渲染指令更新DOM模板、和/或卡片上的信息等。从而有助于避免电子设备根据解析JSBundle得到渲染指令集渲染出重复的卡片视图,造成内存泄露。

[0114] 需要说明的是,通常执行步骤403和步骤404是在不同的线程上,例如,步骤403否构建DOM模板在DOM线程上,对卡片进行渲染在UI线程上,而步骤404解析JSBundle在JS线程上,因此电子设备可以同时执行步骤403和步骤404,也可以先执行步骤403在执行步骤404,对此不作限定。本申请实施例中,由于服务器可以将第一渲染指令集、Layout快照或者DOM模板发送给电子设备,使得电子设备可以先根据DOM模板和Layout快照对卡片进行初步渲染,呈现给用户,从而有助于提高电子设备向用户呈现卡片的效率,降低电子设备呈现卡片时导致的卡顿的可能性。另外,由于电子设备还能够对卡片的JSBundle进行解析,并继续根据解析得到的渲染指令除第一渲染指令集以外的渲染指令继续对卡片渲染,从而使得电子设备可以响应于用户对卡片的操作,进行删除、分享、编辑等。以图10所示的卡片801为例。电子设备在执行步骤406后,用户可以对虚拟按钮800进行操作,使得电子设备响应于对虚拟按钮1000的操作,在显示屏上显示菜单栏。其中,菜单栏中包括多个虚拟按钮。例如,用户当选中移除时,负一屏上不再显示卡片1001。然而,当电子设备执行步骤403后,用户对虚拟按钮1000操作,电子设备可能无法响应用户对虚拟按钮1000操作的事件。

[0115] 以卡片1的相关信息为卡片1的JSBundle、第一渲染指令集和Layout快照为例。如图11所示,电子设备在UI线程检测到宿主应用请求加载卡片1时,向服务器发送卡片1的相关信息获取请求。服务器接收到卡片1的相关信息的获取请求后,将卡片1的相关信息发送给电子设备。电子设备可以将卡片1的相关信息缓存到某一存储器中,例如处理器中的缓存器、或者内部存储器、或者与外部存储器接口连接的外部存储器、或者专用存储器中等。电子设备通过宿主应用的API将存储器中的第一渲染指令集传递给JS-Native引擎,并由JS-Native引擎在DOM线程上根据第一渲染指令集构建DOM模板,以及通过宿主应用的API将存储器中的Layout快照传递给JS-Native引擎,由JS-Native引擎在UI线程上根据DOM模板和Layout快照,对卡片1进行初步渲染,并在显示屏上呈现卡片1。进一步的,电子设备通过宿

主应用的API将存储器的卡片1的JSBundle传递给JSFramework,并由JSFramework在JS线程上对JSBundle解析得到渲染指令集,然后从渲染指令集中筛选出除第一渲染指令集以外的渲染指令。在DOM线程上,根据筛选出的渲染指令,更新DOM模板。最后,在UI线程上根据更新后的DOM模板等,更新在显示屏上显示的卡片1,本流程结束。其中,卡片1的JSBundle是开发者上传到服务器的,第一渲染指令集、Layout快照文件是服务器对开发者上传的JSBundle解析得到的,也可以是开发者上传到服务器的。

[0116] 与图2所示的卡片渲染方法相比,图12中所示的本申请实施例提供的渲染卡片的方法,可以先构建DOM模板,并根据Layout快照文件实现对卡片的初步渲染呈现给用户,从而大大提高了电子设备向用户呈现卡片的效率。

[0117] 需要说明的是,如图4所示的卡片渲染的方法中,由服务器的实现功能还可以集成在集成开发环境(integrated development environment,IDE)上实现。

[0118] 示例二:

[0119] 如图13所示,为本申请实施例的另一种卡片渲染的方法的流程示意图,具体包括以下步骤。

[0120] 步骤1301,电子设备向服务器发送卡片1的文件包集合获取请求。

[0121] 步骤1301可以参见步骤401中电子设备向服务器发送卡片1的相关信息的获取请求的相关介绍,在此不再赘述。

[0122] 步骤1302,电子设备根据预先存储的卡片1的相关信息,对卡片1进行初步渲染,并在显示屏上显示卡片1。

[0123] 示例的,预先存储的卡片1的相关信息,可以为电子设备之前对卡片1渲染时存储的,例如,预先存储的卡片1的相关信息可以为电子设备最近一次对卡片1渲染时存储的。其中,预先存储的卡片1的相关信息可以包括对电子设备之前对卡片1渲染所使用的JSBundle解析得到的渲染指令集、Layout快照;或者预先存储的卡片1的相关信息包括对电子设备之前对卡片1渲染所使用的JSBundle解析得到的渲染指令集;或者预先存储的卡片1的相关信息包括对电子设备之前对卡片1渲染所使用的JSBundle解析得到的渲染指令集、DOM模板和Layout快照等,对此不作限定。

[0124] 以预先存储的卡片1的相关信息包括第一渲染指令集、和Layout快照为例。电子设备根据第一渲染指令集,构建DOM模板。然后根据DOM模板和Layout快照,对卡片1进行初步渲染,并在显示屏上显示卡片1。以预先存储的卡片1的相关信息为对电子设备之前对卡片1渲染所使用的JSBundle解析得到的渲染指令集为例,电子设备可以根据渲染指令集,构建DOM模板,以及进行Layout布局计算,得到Layout快照,然后根据DOM模板和Layout快照对卡片1进行初步渲染。

[0125] 例如,电子设备可以通过宿主应用的API将预先存储的卡片1的第一渲染指令集和Layout快照传递给JS-Native引擎,由JS-Native引擎在DOM线程上根据第一渲染指令集,构建DOM模板,并在UI线程上根据DOM模板和Layout快照,对卡片1进行初步渲染,并向用户呈现初步渲染后的卡片1。

[0126] 步骤1303,服务器接收到卡片1的文件包集合获取请求,向电子设备发送卡片1的JSBundle。

[0127] 步骤1304,电子设备接收到服务器发送的卡片1的JSBundle后,根据JSBundle,得

到卡片1的相关信息。

[0128] 示例的,电子设备接收到服务器发送的卡片1的JSBundle后,通过宿主应用的API将JSBundle传递给JS-Native架构中的JSFramework。由JSFramework对JSBundle进行解析得到渲染指令集,并将渲染指令集传递给JS-Native引擎,由JS-Native引擎根据渲染指令集构建DOM模板、以及进行Layout布局计算得到Layout快照,从而得到卡片1的相关信息。

[0129] 步骤1305,电子设备根据服务器发送的JSBundle得到卡片1的相关信息,对预先存储的卡片1的相关信息进行校验。

[0130] 需要说明的是,根据服务器发送的卡片1的JSBundle得到卡片1的相关信息,与预先存储的卡片1的相关信息相关。示例的,预先存储的卡片1的相关信息包括渲染指令集时,步骤1304中,电子设备可以在接收到服务器发送的卡片1的JSBundle后,对JSBundle解析,得到卡片1的渲染指令集,通过比较预先存储的渲染指令集与对接收到的来自服务器的JSBundle解析得到的渲染指令集是否相同,对预先存储的卡片1的相关信息进行校验。但预先存储的渲染指令集与对接收到的来自服务器的JSBundle解析得到的渲染指令集不同时,校验失败。当预先存储的渲染指令集与对接收到的来自服务器的JSBundle解析得到的渲染指令集相同时,校验成功。

[0131] 再例如,预先存储的卡片1的相关信息包括卡片1的DOM模板。示例的,步骤1104中,电子设备可以在接收到服务器发送的卡片1的JSBundle后,对JSBundle解析,得到卡片1的渲染指令集,并根据得到的卡片1的渲染指令集,构建DOM模板。通过比较基于接收到的来自服务器的JsBundle构建的DOM模板与预先存储的卡片1的DOM模板是否相同。当基于接收到的来自服务器的JsBundle构建的DOM模板与预先存储的卡片1的DOM模板不同时,校验失败;当基于接收到的来自服务器的JsBundle构建的DOM模板与预先存储的卡片1的DOM模板相同时,校验成功。

[0132] 步骤1306,电子设备当校验失败时,删除预先存储的卡片1的相关信息,并存储根据服务器发送的卡片1的JSBundle得到的卡片1的相关信息;以及根据服务器发送的卡片1的JSBundle得到的卡片1的相关信息,对卡片1进行重新渲染,并在显示屏上呈现重现渲染后的卡片1。

[0133] 示例的,电子设备当校验成功时,电子设备确定完成渲染。

[0134] 在一些实施例中,电子设备中未存储卡片1的相关信息时,电子设备可以对从服务器获取的卡片1的JSBundle进行解析,得到渲染指令集。然后根据渲染指令集,构建DOM模板,以及进行Layout布局计算,得到Layout快照,然后对卡片进行渲染,在显示屏上显示该卡片。

[0135] 与示例一相比,示例二中卡片渲染的方法无需服务器对卡片的JSBundle进行预解析,而且,电子设备先基于预先存储的卡片1的相关信息进行卡片渲染,然后再对预先存储的卡片1的相关信息进行校验,从而有助于提高显示屏向用户呈现卡片的效率。而且,在校验失败后,根据服务器发送个JSBundle重新渲染卡片,有助于保证电子设备向用户呈现的卡片的准确性。

[0136] 需要说明是,步骤1301和步骤1302没有必然的先后顺序。或者,电子设备可以一边根据服务器发送的JSBundle对预先存储的卡片1的相关信息进行校验,一边根据预先存储的卡片1的相关信息对卡片进行渲染,再校验失败后,再根据服务器发送的JSBundle重新渲



染卡片。

[0137] 另外,还需要说明的是,本申请实施例中电子设备预先存储的卡片1的相关信息还可以为第一渲染指令集、Layout快照,其中第一渲染指令集为对电子设备之前对卡片进行渲染所使用的JSBundle解析得到的渲染指令集中用于构建DOM模板的渲染指令集。在这种情况下,如果校验成功,电子设备还需要对从服务器接收到的JSBundle解析得到的渲染指令集中筛选出除第一渲染指令集以外的渲染指令,并根据筛选出的渲染指令更新初步渲染的卡片。

[0138] 以预先存储的卡片1的相关信息为卡片1的渲染指令集和Layout快照为例。如图14所示,电子设备在UI线程检测到宿主应用请求加载卡片1时,向服务器发送卡片1的文件包集合获取请求。服务器接收到卡片1的文件包集合的获取请求后,将卡片1的JSBundle发送给电子设备。电子设备可以将卡片1的JSBundle缓存到某一存储器中,例如处理器中的缓存器、或者内部存储器、或者与外部存储器接口连接的外部存储器、或者专用存储器中等。电子设备在DOM线程上从相应的存储器中获取预先存储的卡片1的渲染指令集,并根据渲染指令集构建DOM模板,以及从相应的存储器中获取Layout快照,并根据Layout快照进行初始化布局,然后在UI线程上根据Layout快照和DOM模板,对卡片1进行初步渲染,并在显示屏上呈现卡片1。进一步的,电子设备在JS线程上对从服务器接收到的卡片1的JSBundle进行解析得到渲染指令集,然后根据对从服务器接收到的卡片1的JSBundle进行解析得到渲染指令集,对预先存储的渲染指令集进行校验。当检验失败时,在DOM线程上,根据对从服务器接收到的卡片1的JSBundle进行解析得到渲染指令集,构建DOM模板,以及进行Layout布局计算得到Layout快照。最后,在UI线程上根据基于从服务器接收到的JSbundle得到的DOM模板和Layout快照,重新渲染卡片1,并在显示屏上显示的重新渲染后的卡片1,本流程结束。

[0139] 需要说明的是,如图13所示的卡片渲染的方法中,由服务器的实现功能还可以集成IDE)实现。

[0140] 还需要说明的是,对于示例一和示例二来说,不同宿主应用可以共用同一JS-Native架构,也可以不同的宿主应用分别对应不同的JS-Native架构,对此不作限定。

[0141] 上述各个实施例可以单独使用,也可以相互结合使用,以达到不同的技术效果。例如,对于电子设备来说,当电子设备首次渲染卡片1时,可以采用示例一所示的卡片渲染方法,当电子设备非首次渲染卡片1时,可采用示例二所示的卡片渲染方法。

[0142] 上述本申请提供的实施例中,从电子设备作为执行主体的角度对本申请实施例提供的方法进行了介绍。为了实现上述本申请实施例提供的方法中的各功能,电子设备可以包括硬件结构和/或软件模块,以硬件结构、软件模块、或硬件结构加软件模块的形式来实现上述各功能。上述各功能中的某个功能以硬件结构、软件模块、还是硬件结构加软件模块的方式来执行,取决于技术方案的特定应用和设计约束条件。

[0143] 基于相同的构思,图15所示为本申请提供的一种装置1500,用于执行图4、或图13所示的卡片渲染方法。示例的,该装置1500可以为电子设备,也可以为芯片。示例的,装置1500包括处理模块1501、通信模块1502和显示模块1503。

[0144] 示例的,处理模块1501用于根据卡片的相关信息进行渲染。通信模块1502用于与服务器进行交互,例如向服务器发送卡片的相关信息的获取请求、接收来自服务器的卡片的相关信息等。显示模块1503用于在卡片完成渲染后,显示卡片。例如,装置1500为电子设

备时,通信模块1502可以为收发器。再例如,装置1502为芯片时,通信模块1502可以为接口。

[0145] 基于相同的构思,图16所示为本申请提供的一种装置1600。装置1600包括至少一个处理器1601、通信接口1602和显示屏1603。示例的,该装置1600可以为电子设备,也可以为芯片。在本申请实施例中,处理器1601可以是通用处理器、数字信号处理器、专用集成电路、现场可编程门阵列或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件,可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件处理器执行完成,或者用处理器中的硬件及软件模块组合执行完成。

[0146] 通信接口1602用于与服务器进行交互。示例性地,通信接口1602可以是收发器、电路、总线、模块或其它类型的通信接口等。例如,装置1600为电子设备时,通信接口1602为收发器。再例如,装置1600为芯片时,通信接口1602为接口。

[0147] 显示屏1603用于显示卡片。

[0148] 在一些实施例中,装置1600还包括存储器1604。存储器1604用于存储程序指令和/或数据。在本申请实施例中,存储器1604可以是非易失性存储器,比如硬盘(hard disk drive,HDD)或固态硬盘(solid-state drive,SSD)等,还可以是易失性存储器(volatile memory),例如随机存取存储器(random-access memory,RAM)。存储器是能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质,但不限于此。

[0149] 其中,处理器1601与存储器1604、通信接口1602、和显示屏1603耦合,本申请实施例中的耦合是装置、单元或模块之间的间接耦合或通信连接,可以是电性,机械或其它的形式,用于装置、单元或模块之间的信息交互。本申请实施例中不限定上述通信接口1602、处理器1601、显示屏1603以及存储器1604之间可以通过总线连接,所述总线可以分为地址总线、数据总线、控制总线等。

[0150] 应理解,该装置1300和装置1600可以用于实现本申请实施例的图4和图13所示的方法,相关特征可以参照上文,此处不再赘述。

[0151] 所属领域的技术人员可以清楚地了解到本申请实施例可以用硬件实现,或固件实现,或它们的组合方式来实现。当使用软件实现时,可以将上述功能存储在计算机可读介质中或作为计算机可读介质上的一个或多个指令或代码进行传输。计算机可读介质包括计算机存储介质和通信介质,其中通信介质包括便于从一个地方向另一个地方传送计算机程序的任何介质。存储介质可以是计算机能够存取的任何可用介质。以此为例但不限于:计算机可读介质可以包括RAM、ROM、电可擦可编程只读存储器(electrically erasable programmable read only memory,EEPROM)、只读光盘(compact disc read-Only memory,CD-ROM)或其他光盘存储、磁盘存储介质或者其他磁存储设备、或者能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质。此外。任何连接可以适当的成为计算机可读介质。例如,如果软件是使用同轴电缆、光纤光缆、双绞线、数字用户线(digital subscriber line,DSL)或者诸如红外线、无线电和微波之类的无线技术从网站、服务器或者其他远程源传输的,那么同轴电缆、光纤光缆、双绞线、DSL或者诸如红外线、无线和微波之类的无线技术包括在所属介质的定义中。如本申请实施例所使用的,盘(disk)和碟(disc)包括压缩光碟(compact disc,CD)、激光碟、光碟、数字通用光碟

(digital video disc,DVD)、软盘和蓝光光碟,其中盘通常磁性的复制数据,而碟则用激光来光学的复制数据。上面的组合也应当包括在计算机可读介质的保护范围之内。

[0152] 总之,以上所述仅为本申请的实施例而已,并非用于限定本申请的保护范围。凡根据本申请的揭露,所作的任何修改、等同替换、改进等,均应包含在本申请的保护范围之内。

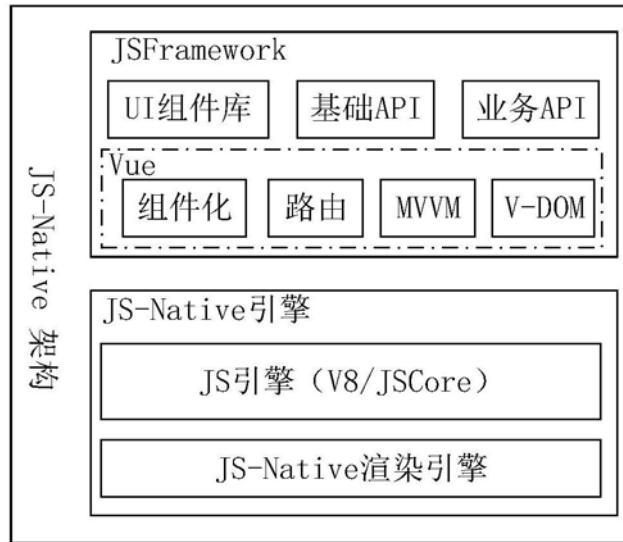


图1

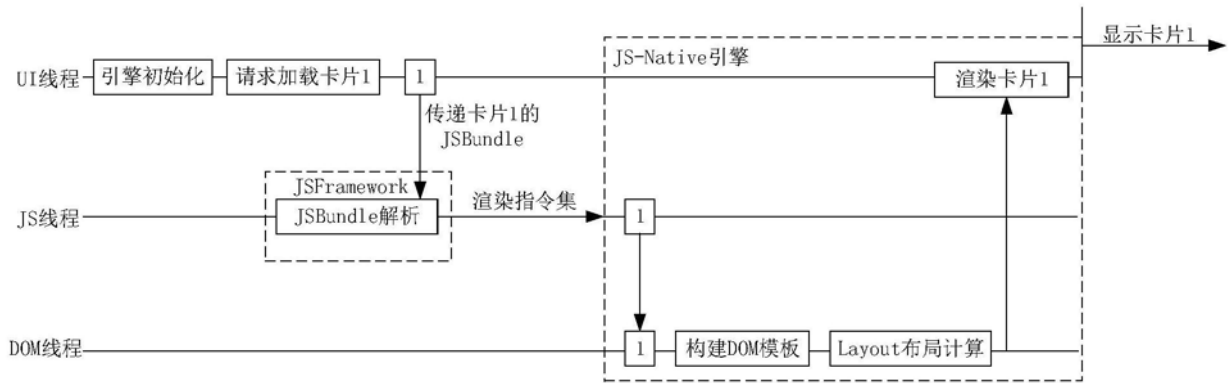


图2

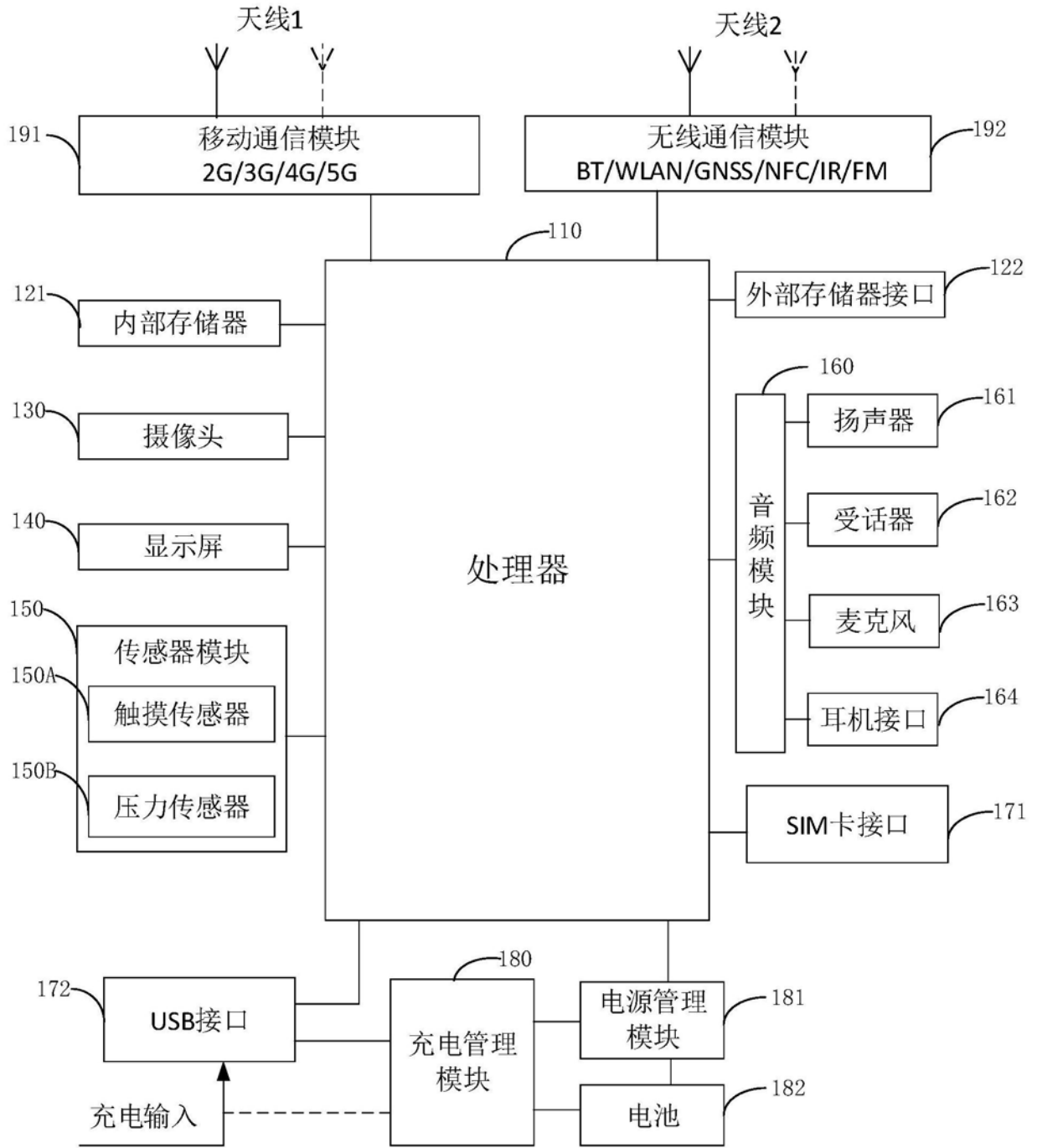


图3

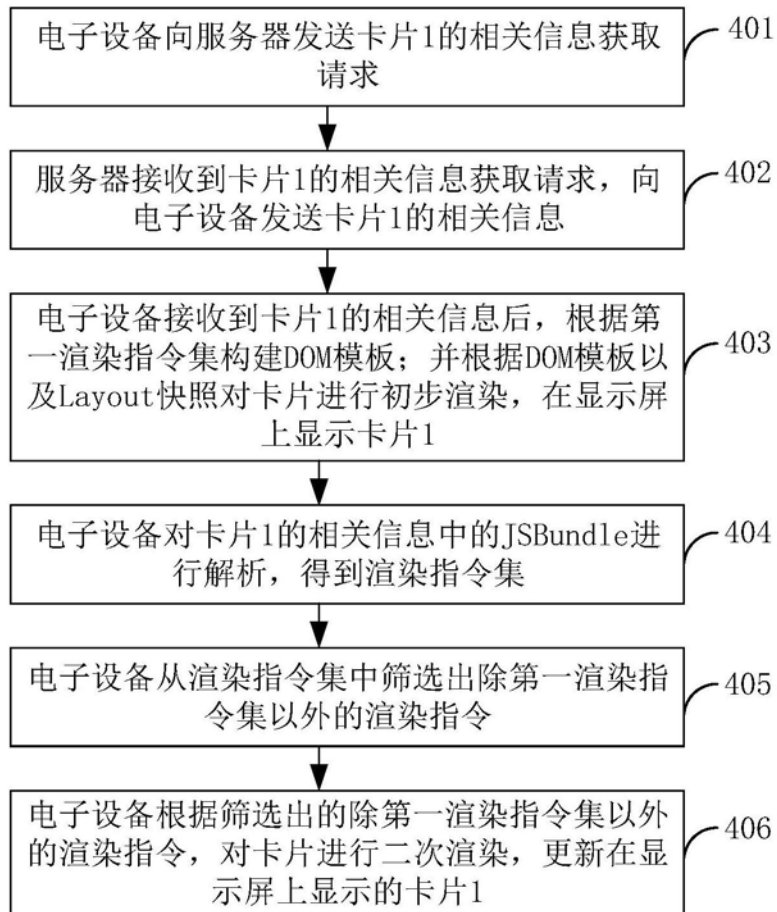


图4

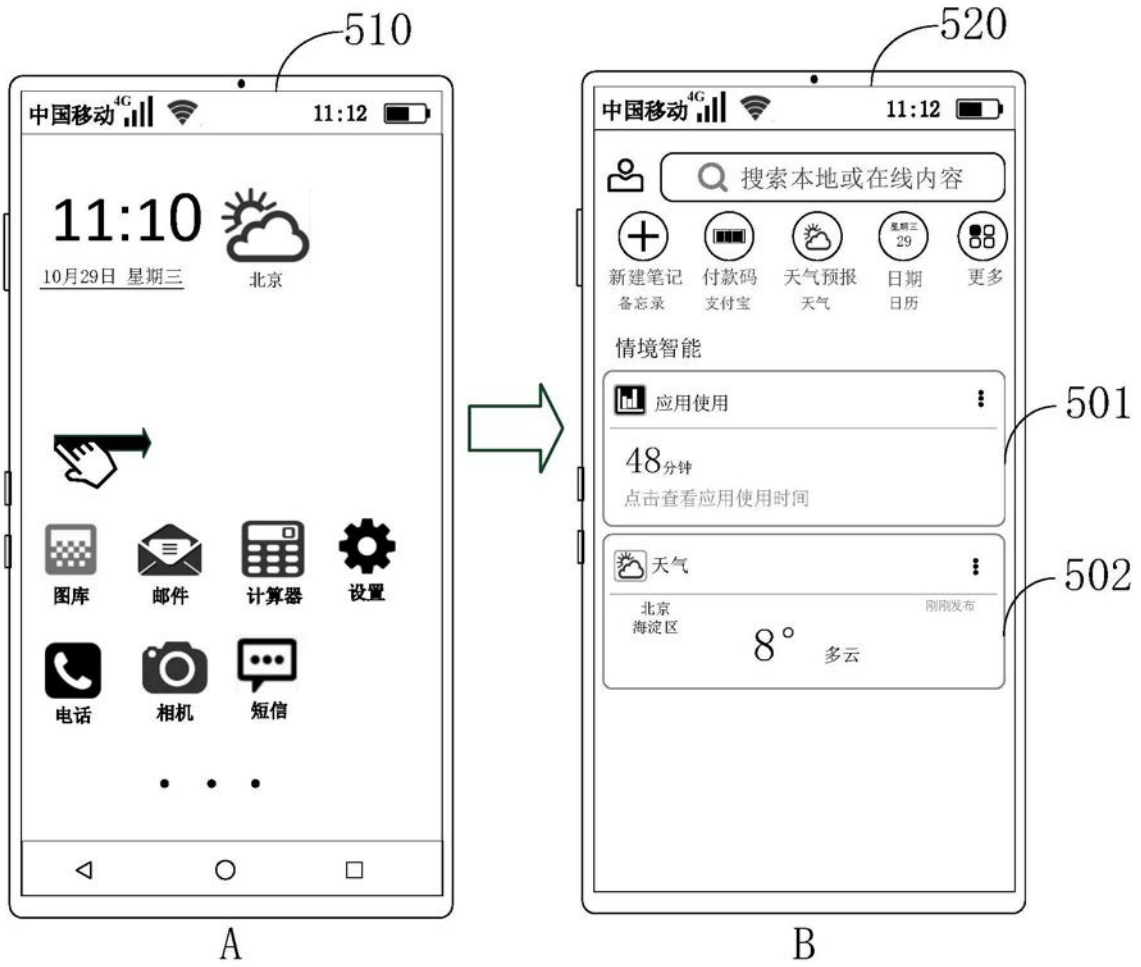


图5

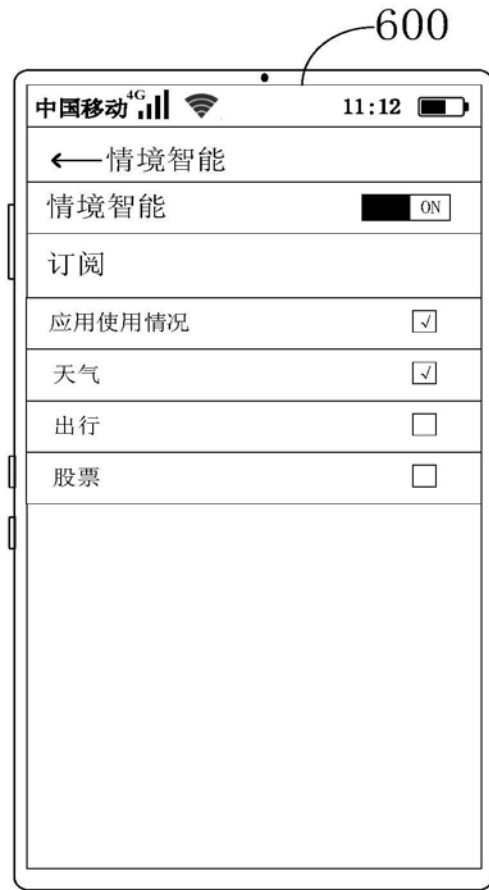


图6



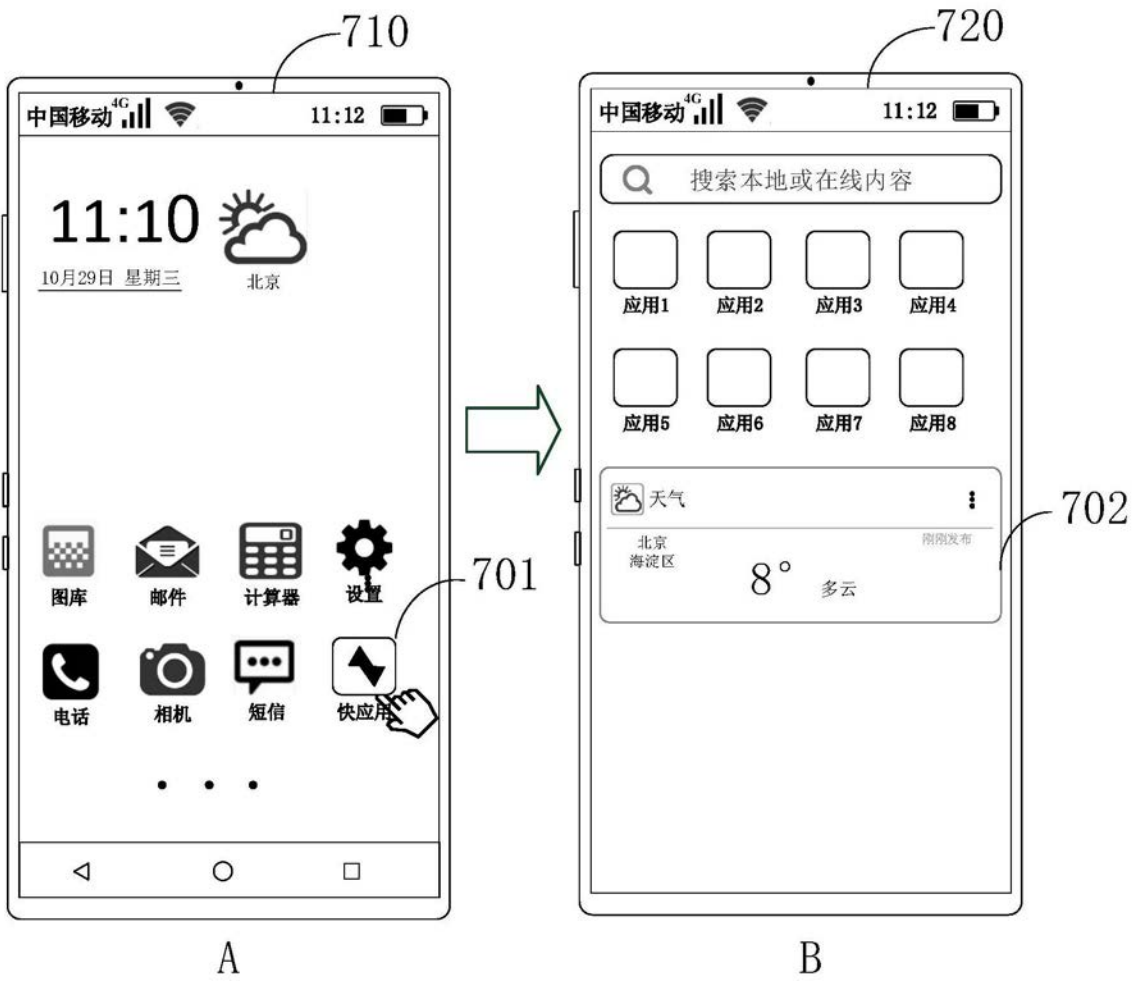


图7

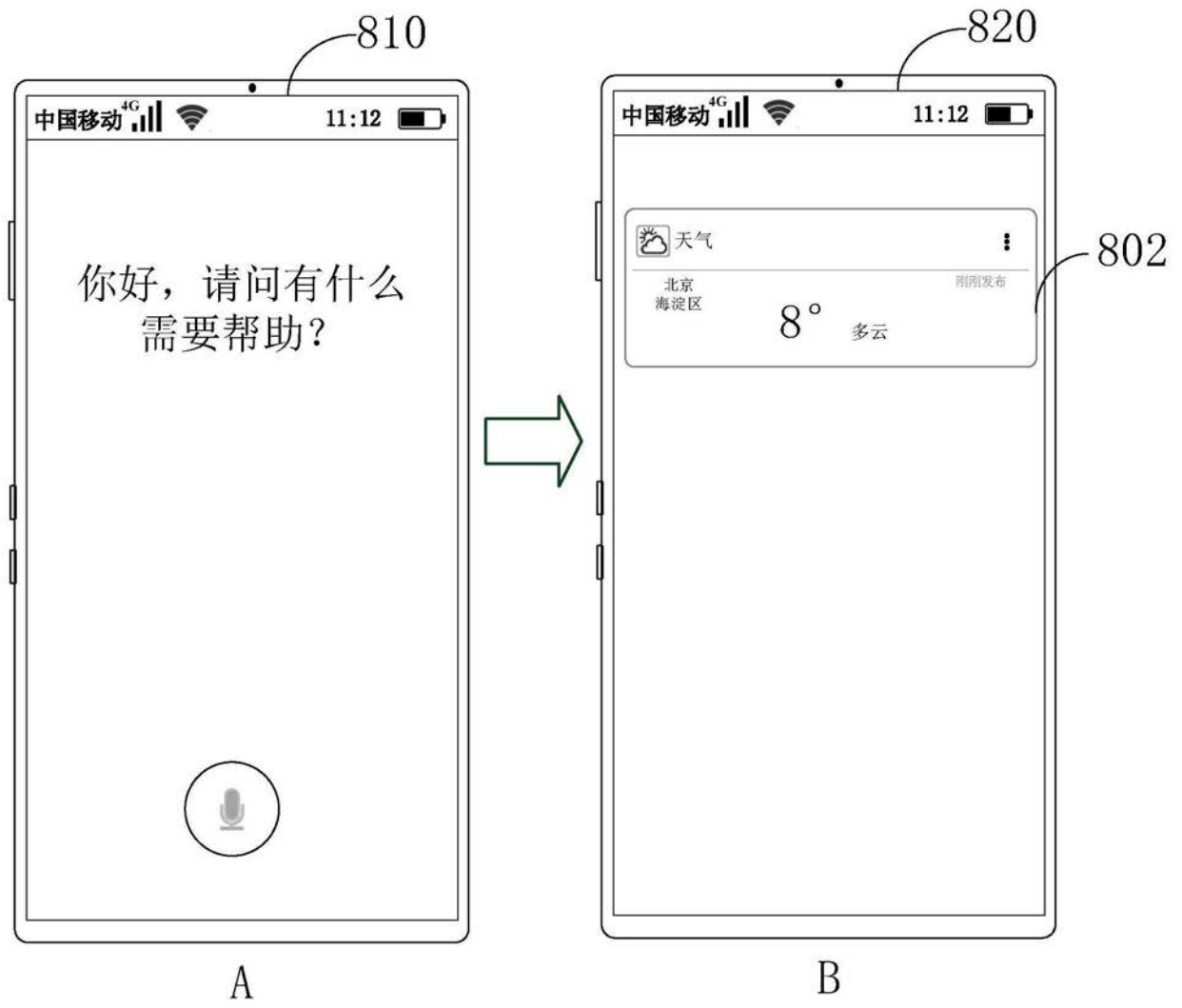


图8

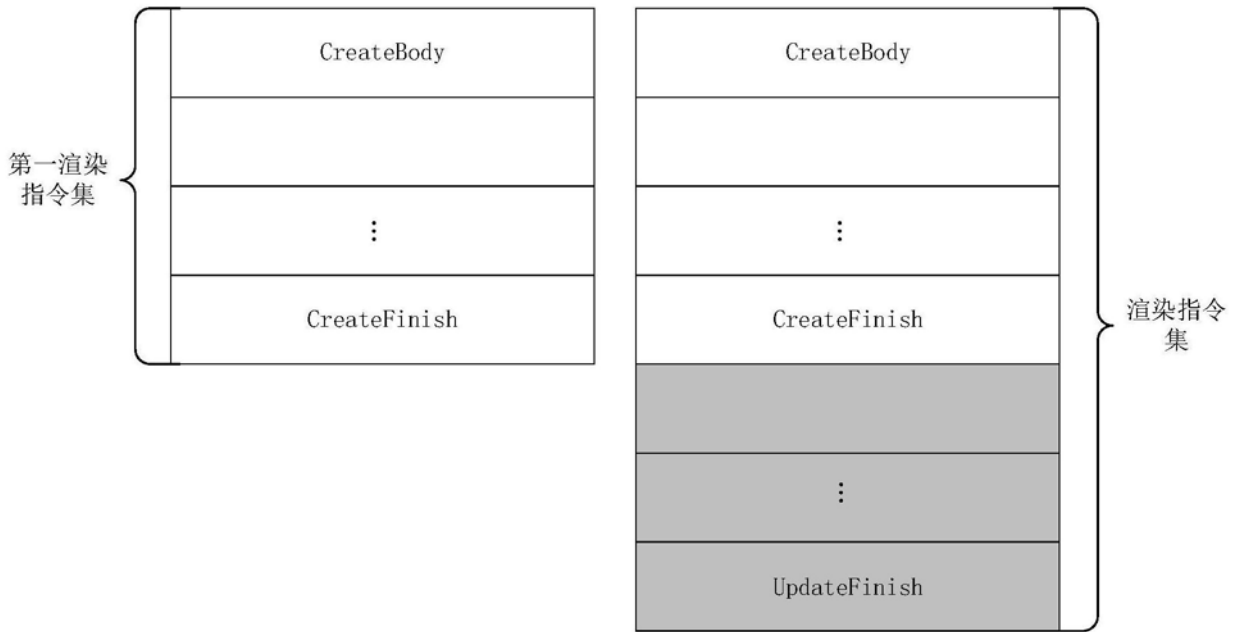


图9



图10

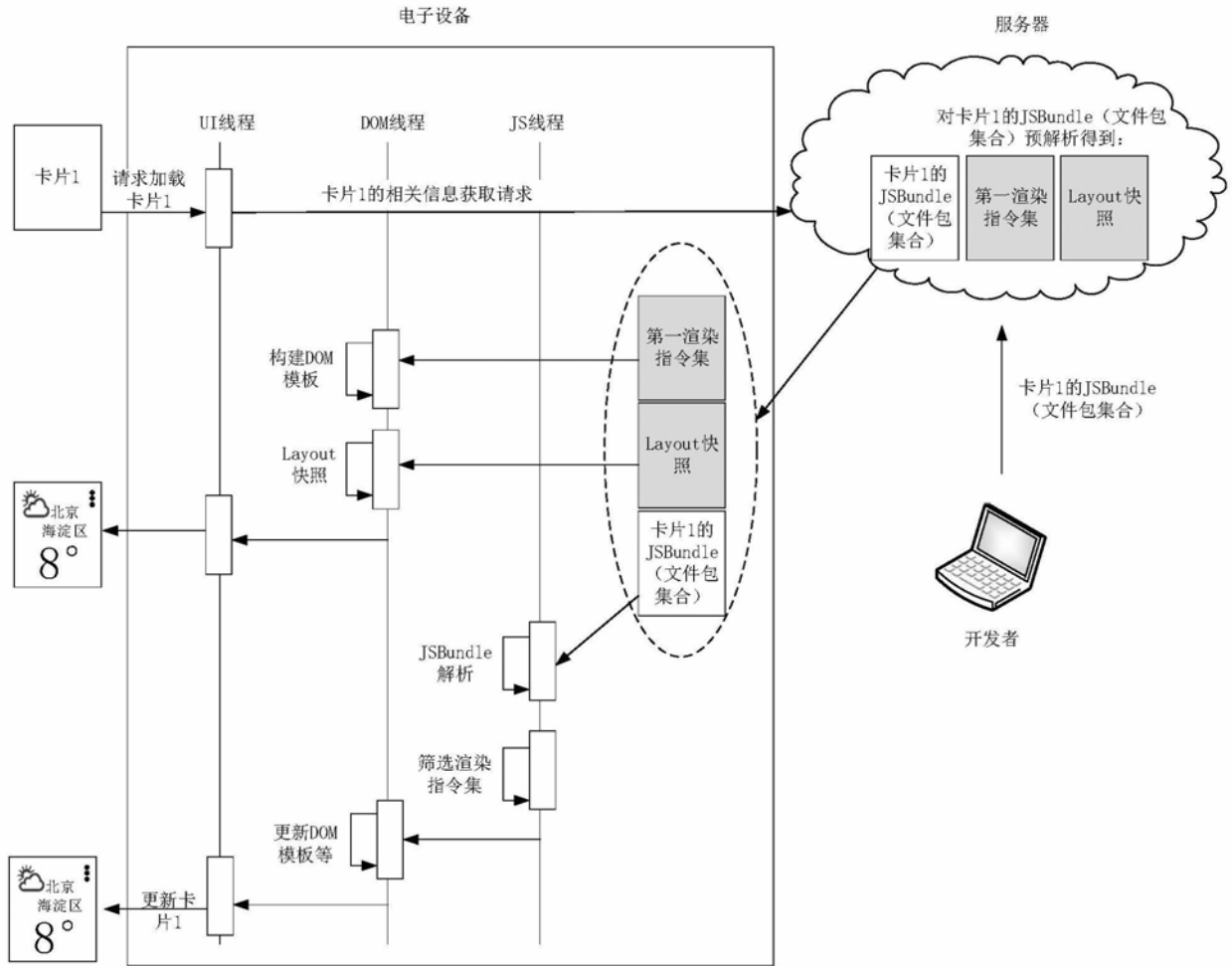


图11

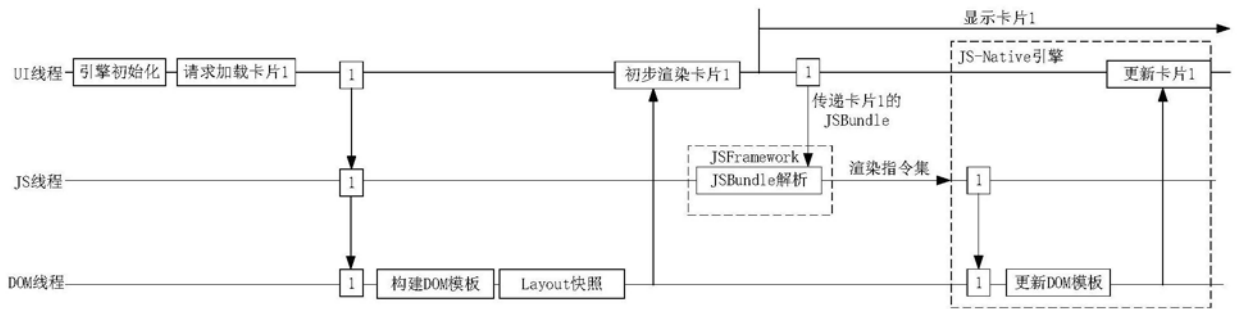


图12

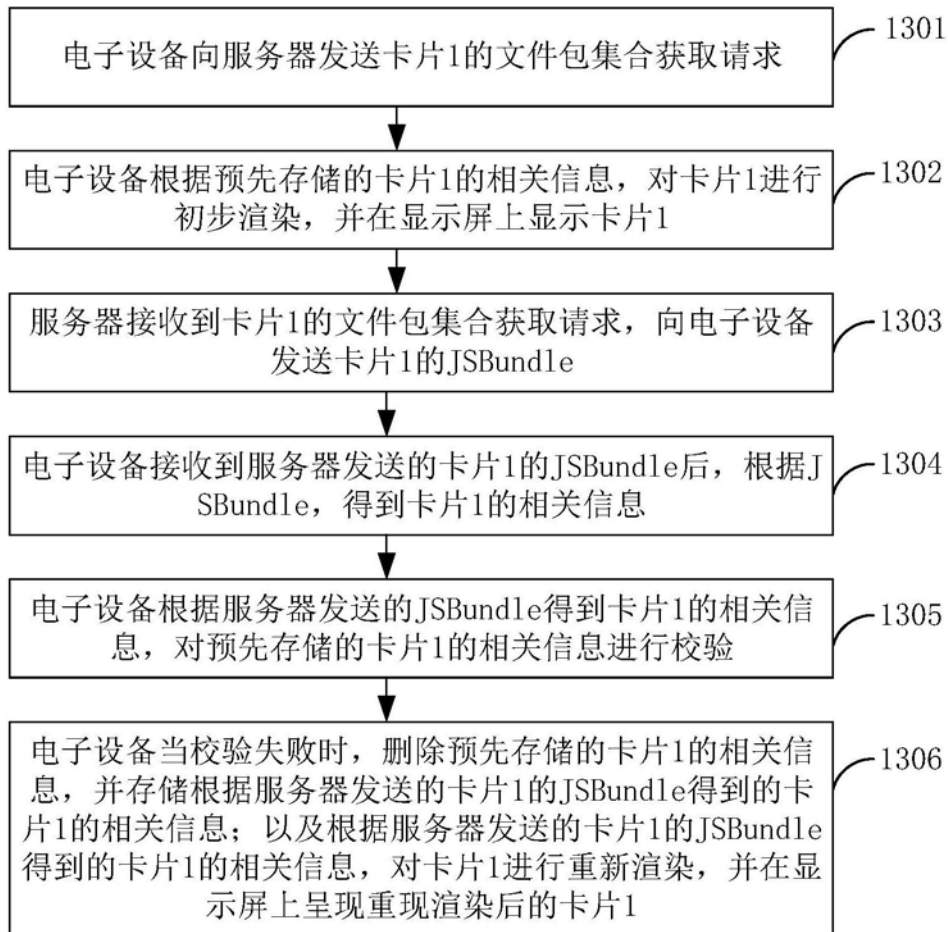


图13

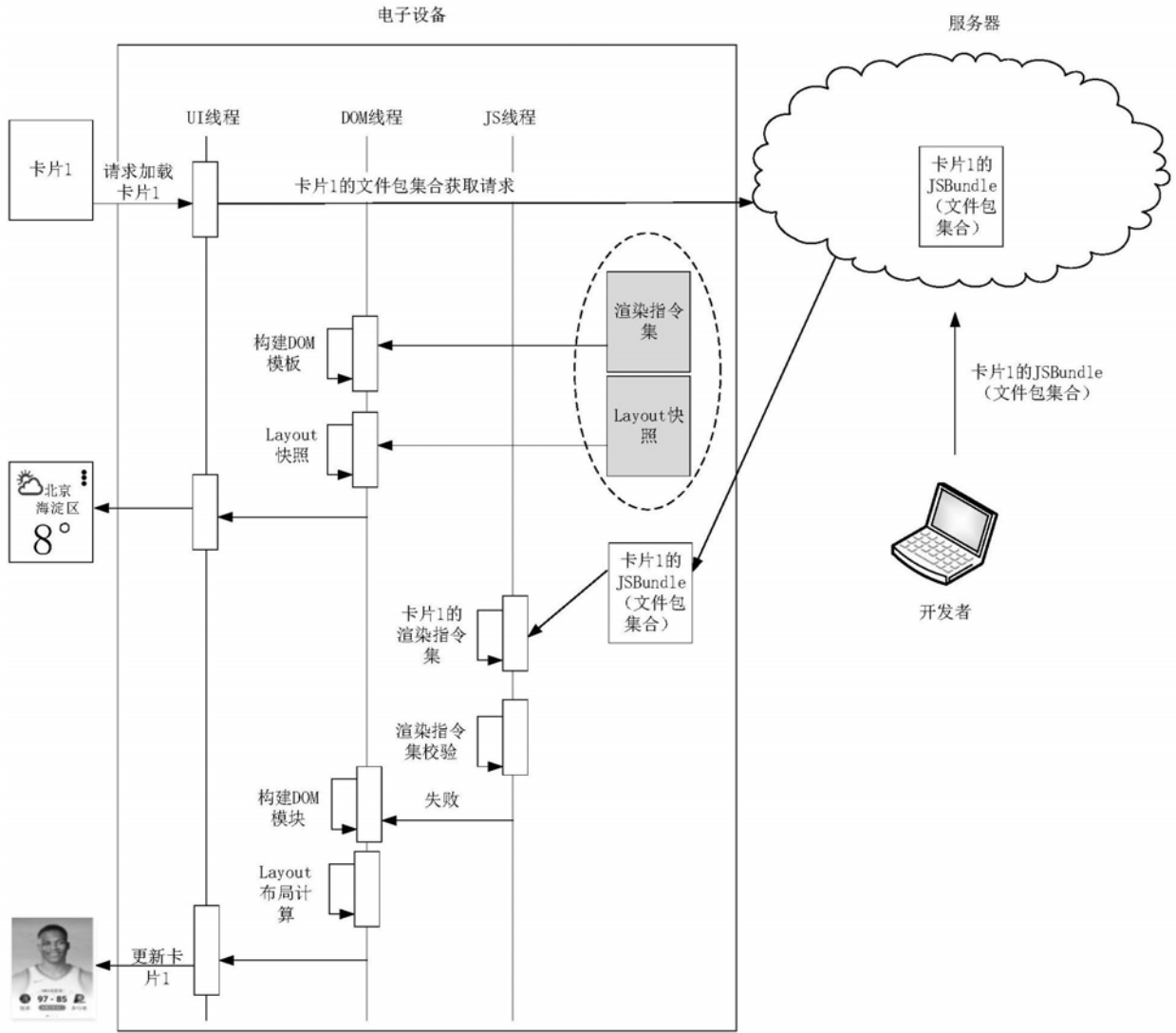


图14

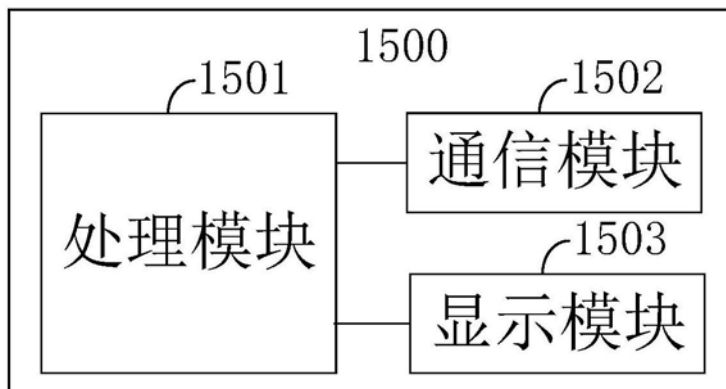


图15

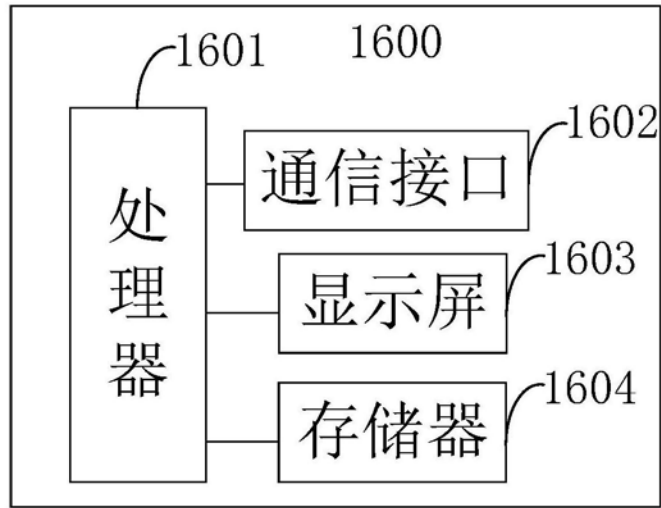


图16