(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 110347445 B
(45)授权公告日 2024.12.27

(21)申请号 201910631634.X

(22)申请日 2019.07.12

(65)同一申请的已公布的文献号
申请公布号 CN 110347445 A

(43)申请公布日 2019.10.18

(73)专利权人 财付通支付科技有限公司
地址 518000 广东省深圳市南山区高新科技园科技中一路腾讯大厦8层

(72)发明人 潘健聪 邓德胜

(74)专利代理机构 北京三高永信知识产权代理有限责任公司 11138
专利代理师 邢少真

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

(56)对比文件
CN 109814943 A,2019.05.28
CN 109710340 A,2019.05.03

审查员 赵园

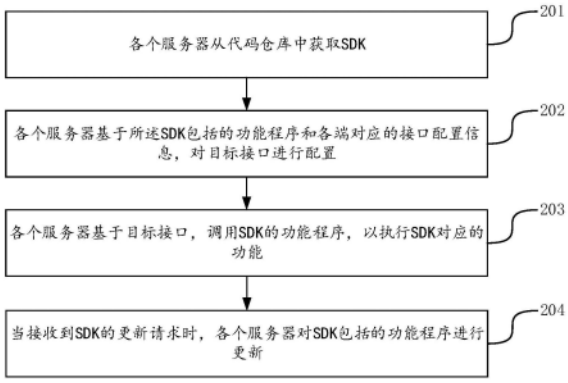
权利要求书2页 说明书7页 附图6页

(54)发明名称

SDK调用方法、装置、服务器及存储介质

(57)摘要

本发明公开了一种SDK调用方法、装置、服务器及存储介质,属于计算机技术领域。所述方法包括:获取软件开发工具包SDK,所述SDK包括多端对应的接口配置信息和功能程序;基于所述SDK包括的当前端对应的接口配置信息和功能程序,对目标接口进行配置;基于所述目标接口,调用所述SDK的功能程序,以执行所述SDK对应的功能。本发明实施例通过采用调用SDK的方式,在多端实现相同的功能,在调用过程中可以根据功能程序和各端对应的接口配置信息,对接口进行配置,从而可以直接对SDK的功能程序进行调用,以实现相应的功能,无需针对不同端各自开发一个工程文件,大大减少了开发时间和人工成本,该SDK调用方式的通用性好。



CN 110347445 B

1.一种软件开发工具包SDK调用方法,其特征在于,所述方法包括:

获取软件开发工具包SDK,所述SDK包括多个功能程序和多端对应的接口配置信息;

基于所述SDK包括的每个功能程序和当前端对应的接口配置信息,对多个目标接口中每个目标接口进行配置,每个目标接口与每个功能程序对应;

基于所述目标接口,调用所述SDK的功能程序,以执行所述SDK对应的功能。

2.根据权利要求1所述的方法,其特征在于,所述多个功能程序包括样式配置程序和业务逻辑程序;

所述基于所述SDK包括的每个功能程序和当前端对应的接口配置信息,对多个目标接口中每个目标接口进行配置,包括:

基于所述SDK包括的样式配置程序以及当前端对应的接口配置信息,对显示接口进行配置;

基于所述SDK包括的业务逻辑程序以及当前端对应的接口配置信息,对业务处理接口进行配置。

3.根据权利要求1所述的方法,其特征在于,所述方法还包括:

当接收到所述SDK的更新请求时,对所述SDK包括的功能程序进行更新。

4.根据权利要求3所述的方法,其特征在于,所述SDK包括的功能程序基于语法抽象树实现;

所述当接收到所述SDK的更新请求时,对所述SDK包括的功能程序进行更新,包括:

当接收到所述SDK的更新请求时,基于所述更新请求的内容,对所述语法抽象树的结构进行更新。

5.根据权利要求1所述的方法,其特征在于,所述方法还包括:

当接收到对所述SDK的调用请求时,执行获取软件开发工具包SDK、配置和调用的步骤,所述调用请求基于对所述SDK的任一功能程序对应的插件进行的交互操作触发,所述调用请求包括所述插件的信息;

所述基于所述目标接口,调用所述SDK的功能程序,以执行所述SDK对应的功能,包括:

基于所述目标接口和所述插件的信息,调用所述SDK的所述插件对应的所述任一功能程序,执行所述任一功能程序对应的功能。

6.一种软件开发工具包SDK调用装置,其特征在于,所述装置包括:

获取模块,用于获取软件开发工具包SDK,所述SDK包括多个功能程序和多端对应的接口配置信息;

配置模块,用于基于所述SDK包括的每个功能程序和当前端对应的接口配置信息,对多个目标接口中每个目标接口进行配置,每个目标接口与每个功能程序对应;

调用模块,用于基于所述目标接口,调用所述SDK的功能程序,以执行所述SDK对应的功能。

7.根据权利要求6所述的装置,其特征在于,所述多个功能程序包括样式配置程序和业务逻辑程序;所述配置模块,用于:

基于所述SDK包括的样式配置程序以及当前端对应的接口配置信息,对显示接口进行配置;

基于所述SDK包括的业务逻辑程序以及当前端对应的接口配置信息,对业务处理接口

进行配置。

8.根据权利要求6所述的装置,其特征在于,所述装置还包括:

更新模块,用于当接收到所述SDK的更新请求时,对所述SDK包括的功能程序进行更新。

9.根据权利要求8所述的装置,其特征在于,所述SDK包括的功能程序基于语法抽象树实现;

所述更新模块,用于当接收到所述SDK的更新请求时,基于所述更新请求的内容,对所述语法抽象树的结构进行更新。

10.根据权利要求6所述的装置,其特征在于,所述装置还包括:

所述获取模块,所述配置模块或所述调用模块,还用于当接收到对所述SDK的调用请求时,执行获取软件开发工具包SDK、配置和调用的步骤,所述调用请求基于对所述SDK的任一功能程序对应的插件进行的交互操作触发,所述调用请求包括所述插件的信息;

所述调用模块,用于基于所述目标接口和所述插件的信息,调用所述SDK的所述插件对应的所述任一功能程序,执行所述任一功能程序对应的功能。

11.一种服务器,其特征在于,所述服务器包括一个或多个处理器和一个或多个存储器,所述一个或多个存储器中存储有至少一条指令,所述指令由所述一个或多个处理器加载并执行以实现如权利要求1至权利要求5任一项所述的软件开发工具包SDK调用方法所执行的操作。

12.一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至少一条指令,所述指令由处理器加载并执行以实现如权利要求1至权利要求5任一项所述的软件开发工具包SDK调用方法所执行的操作。

SDK调用方法、装置、服务器及存储介质

技术领域

[0001] 本发明涉及计算机技术领域,特别涉及一种SDK调用方法、装置、服务器及存储介质。

背景技术

[0002] 随着计算机技术的发展,多端同构技术的应用越来越广泛,在很多场景中,通常需要在多端上实现相同的功能。例如,在网页端、应用端、小程序端均可以接入某个相同的广告功能。为了满足多端同构的需求,则需要在多端对应的服务器上部署实现相同功能的工程文件,该工程文件中可以包括功能程序。

[0003] 目前,不同端的运行环境不同,如果需要在多端实现同一功能,则需要由技术人员为每端各自开发一个工程文件。该过程中所需的开发时间较长,需要耗费很大的人工成本,工程文件的通用性差。

发明内容

[0004] 本发明实施例提供了一种软件开发工具包(Software Development Kit,SDK)调用方法、装置、服务器及存储介质,可以解决相关技术中开发时间长、耗费的人工成本大和通用性差的问题。所述技术方案如下:

[0005] 一方面,提供了一种SDK调用方法,所述方法包括:

[0006] 获取软件开发工具包SDK,所述SDK包括功能程序和多端对应的接口配置信息;

[0007] 基于所述SDK包括的功能程序和当前端对应的接口配置信息,对目标接口进行配置;

[0008] 基于所述目标接口,调用所述SDK的功能程序,以执行所述SDK对应的功能。

[0009] 一方面,提供了一种SDK调用装置,所述装置包括:

[0010] 获取模块,用于获取软件开发工具包SDK,所述SDK包括功能程序和多端对应的接口配置信息;

[0011] 配置模块,用于基于所述SDK包括的功能程序和当前端对应的接口配置信息,对目标接口进行配置;

[0012] 调用模块,用于基于所述目标接口,调用所述SDK的功能程序,以执行所述SDK对应的功能。

[0013] 在一种可能实现方式中,所述多个功能程序包括样式配置程序和业务逻辑程序;

[0014] 所述配置模块用于:

[0015] 基于所述SDK包括的样式配置程序以及当前端对应的接口配置信息,对显示接口进行配置;

[0016] 基于所述SDK包括的业务逻辑程序以及当前端对应的接口配置信息,对业务处理接口进行配置。

[0017] 在一种可能实现方式中,所述装置还包括:

[0018] 更新模块,用于当接收到所述SDK的更新请求时,对所述SDK包括的功能程序进行更新。

[0019] 在一种可能实现方式中,所述SDK包括的功能程序可以基于语法抽象树实现;

[0020] 所述更新模块用于当接收到所述SDK的更新请求时,基于所述更新请求的内容,对所述语法抽象树的结构进行更新。

[0021] 在一种可能实现方式中,所述装置还包括:

[0022] 所述获取模块,所述配置模块或所述调用模块,还用于当接收到对所述SDK的调用请求时,执行所述获取软件开发工具包SDK、配置和调用的步骤,所述调用请求基于对所述SDK的任一功能程序对应的插件进行的交互操作触发,所述调用请求包括所述插件的信息;

[0023] 所述调用模块用于基于所述目标接口和所述插件的信息,调用所述SDK的所述插件对应的所述任一功能程序,执行所述任一功能程序对应的功能。

[0024] 一方面,提供了一种服务器,所述服务器包括一个或多个处理器和一个或多个存储器,所述一个或多个存储器中存储有至少一条指令,所述指令由所述一个或多个处理器加载并执行以实现所述SDK调用方法所执行的操作。

[0025] 一方面,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至少一条指令,所述指令由处理器加载并执行以实现所述SDK调用方法所执行的操作。

[0026] 本发明实施例通过采用调用SDK的方式,在多端实现相同的功能,在调用过程中可以根据功能程序和各端对应的接口配置信息,对接口进行配置,从而可以直接对SDK的功能程序进行调用,以实现相应的功能,无需针对不同端各自开发一个工程文件,大大减少了开发时间和人工成本,该SDK调用方式的通用性好。

附图说明

[0027] 为了更清楚地说明本发明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发明的一些实施例,对于本领域普通技术人员来讲,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。

[0028] 图1是本发明实施例提供的一种SDK调用方法的实施环境;

[0029] 图2是本发明实施例提供的一种SDK调用方法的流程图;

[0030] 图3是本发明实施例提供的一种SDK调用方法的流程图;

[0031] 图4是本发明实施例提供的一种SDK调用方法的流程图;

[0032] 图5是本发明实施例提供的一种SDK调用方法的流程图;

[0033] 图6是相关技术中对工程文件进行构建部署的流程图;

[0034] 图7是本发明实施例提供的一种SDK调用装置的结构示意图;

[0035] 图8是本发明实施例提供的一种服务器的结构示意图。

具体实施方式

[0036] 为使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明实施方式作进一步地详细描述。

[0037] 图1是本发明实施例提供的一种SDK调用方法的实施环境,参见图1,该实施环境可

以包括多个服务器101和代码仓库102。其中,该代码仓库102可以存储于一个或多个服务器中。每个服务器101可以为同一种网络服务的各个端。例如,该各个端可以包括网页(World Wide Web,简称Web)端、小程序端、轻应用端和应用(Application,APP)端。

[0038] 在本发明实施例中,该代码仓库102用于提供软件开发工具包(SDK),每个服务器101可以从代码仓库102中获取SDK,并在运行过程调用SDK,以在不同端实现同一功能。

[0039] 图2是本发明实施例提供的一种SDK调用方法的流程图,该方法可以应用于服务器中,该服务器可以为上述图1所示的服务器101。参见图2,该方法可以包括以下步骤:

[0040] 201、各个服务器从代码仓库中获取SDK。

[0041] 其中,各个服务器可以是指同一种网络服务的各个端。在本发明实施例中,仅以各个端均从代码仓库中获取SDK,并调用SDK为例进行说明,对于任一个端,均可以执行同理的步骤,本发明实施例对此不作限定。

[0042] 其中,该SDK可以包括功能程序和多端对应的接口配置信息。该功能程序为该SDK可以实现的功能对应的程序,通过执行该功能程序,可以实现对应的功能。该SDK适用于多端,通过该多端对应的接口配置信息,可以在多端所在服务器上实现该SDK的调用以及功能实现。

[0043] 例如,在一些场景中,该SDK可以为第三方开发者提供的软件包、软件框架、硬件平台、操作系统等创建应用软件开发工具的集合,该SDK可以作为插件,应用到现有的产品中,比如可以作为广告插件、图表插件、投票插件、天气插件或其他自定义插件等,应用到相应的项目和产品中。

[0044] 由于不同端的运行环境不同,对该SDK调用时,各个服务器所需执行的步骤也可以不同。因而,各个服务器可以先获取各端的信息,基于各端的信息,采用相应的处理方式。对于一个服务器来说,则可以先获取当前端的信息,基于当前端的信息,采用相应的步骤。

[0045] 在一种可能实现方式中,该各个服务器从代码仓库中获取该SDK可以基于对该SDK的调用请求触发,对于任一端,用户可以进行操作,服务器检测到用户的操作,基于该操作,调用该SDK。

[0046] 在一个具体的可能实施例中,该SDK可以包括一个功能程序,也可以包括多个功能程序,每个功能程序可以对应一种功能。对于任一个功能程序,均可以作为相应的插件的形式显示于该服务器所提供的显示界面中,从而用户可以对该插件进行交互操作,进而触发上述对SDK的调用请求。

[0047] 该调用请求可以基于对该SDK的任一功能程序对应的插件进行的交互操作触发,该调用请求包括所述插件的信息。因而,该步骤201至步骤203中,各个服务器可以基于该目标接口和该插件的信息,调用该SDK的该插件对应的所述任一功能程序,执行该任一功能程序对应的功能。

[0048] 202、各个服务器基于该SDK包括的功能程序和各端对应的接口配置信息,对目标接口进行配置。

[0049] 各个服务器获取到SDK后,该SDK的功能程序均封装于该SDK中,如果要调用该SDK的功能程序,则需要进行配置。

[0050] 在一种可能实现方式中,该步骤202之前,各个服务器获取到SDK后,在执行功能程序之前,可以先对SDK进行初始化。例如,该初始化过程中可以初始化SDK的执行环境,导入

调用SDK所需的头文件,初始化执行参数等,进而可以在SDK初始化后,对目标接口进行配置。

[0051] 在上述步骤201中的一个具体的可能实施例中,该SDK可以包括多个功能程序。其中,每个功能程序可能对应于不同的目标接口。该步骤202则可以为:各个服务器基于该SDK包括的每个功能程序和各端对应的接口配置信息,对多个目标接口中每个目标接口进行配置,每个目标接口与每个功能程序对应。对于任一端,该端所在服务器则可以基于该述SDK包括的每个功能程序和当前端对应的接口配置信息,对多个目标接口中每个目标接口进行配置。

[0052] 在对该目标接口进行配置时,可以将该SDK的接口配置信息写入该目标接口,通过该写入步骤,可以调整该目标接口的参数,进而通过调整过参数后的目标接口,可以调用该SDK。当然,如果SDK包括多个功能程序,在对多个目标接口进行配置时,可以将该每个功能程序的信息写入对应的目标接口,二者建立起对应关系,可以基于目标接口,调用对应的功能程序。

[0053] 在一个具体的可能实施例中,该多个功能程序包括样式配置程序和业务逻辑程序。该样式配置程序可以用于指示显示样式。该业务逻辑程序用于指示所需执行的处理逻辑。例如,如果确定某个插件的显示样式,可以基于该样式配置程序,确定其显示样式。从一个服务器角度来说,在用户进行了交互操作后,服务器可以基于该业务逻辑程序,确定该交互操作所需反馈的交互结果。

[0054] 在该实施例中,服务器针对不同的功能程序,执行不同的配置处理的过程可以为:服务器可以基于该SDK包括的样式配置程序以及当前端对应的接口配置信息,对显示接口进行配置,基于该SDK包括的业务逻辑程序以及当前端对应的接口配置信息,对业务处理接口进行配置。

[0055] 203、各个服务器基于所该目标接口,调用该SDK的功能程序,以执行该SDK对应的功能。

[0056] 各个服务器对目标接口进行配置后,则可以基于目标接口对SDK的功能程序进行调用,实现对应的功能。

[0057] 在上述SDK包括多个功能程序的实现方式中,各个服务器基于任一目标接口,可以根据该任一目标接口的配置参数,调用该任一目标接口对应的功能程序,执行该功能程序对应的功能。对于一端,该端所在服务器则可以基于任一目标接口,可以根据该任一目标接口的配置参数,调用该任一目标接口对应的功能程序,执行该功能程序对应的功能。

[0058] 例如,在一个具体示例中,如图3所示,以在Web端调用该SDK为例进行说明,在SDK初始化之后,可以对目标接口进行配置。在该示例中,将每个功能程序作为一个功能模块,SDK包括用户界面(User Interface,UI)渲染模块、事件绑定模块、业务逻辑模块和数据上报模块,该SDK还可以包括形态配置。对于该多个功能模块,形态配置即为上述样式配置程序,基于形态配置,可以确定进行UI渲染时的显示样式。对于UI渲染显示的插件,如果用户对插件进行了交互操作,可以根据事件配置模块,确定该交互操作这一事件绑定的处理方式,从而调用业务逻辑模块来去进行相应的业务处理,在处理后,还可以将数据上报给服务器,或反馈给用户交互结果。Web接口(目标接口)可以包括显示接口(Render Application Programming Interface,Render API)、事件绑定接口(API)、业务逻辑API和数据上报API。

则在对目标接口进行配置过程,则可以配置各个功能模块与API之间的对应关系,通过各个目标接口,可以调用各个功能模块,实现相应的功能。

[0059] 在上述具体示例中,如图4所示,该SDK调用的具体流程具体可以为以下几个步骤,下述流程中,以Web端为例,按照该服务器执行不同的功能,对该服务器的各个功能结构进行了命名。具体如下:

[0060] 步骤一、用户访问Web端产品,并启用特定插件。其中,该特定插件即为该SDK所对应的功能插件。

[0061] 步骤二、SDK初始化模块初始化该插件的SDK环境,也即是对SDK进行初始化,初始化成功后,可以返回初始化成功。

[0062] 步骤三、SDK初始化模块使用Web接口初始化框架。

[0063] 步骤四、SDK初始化模块在Web接口注册该多端同构框架。

[0064] 该步骤三和步骤四的过程也即是对目标接口进行配置的过程。

[0065] 步骤五、配置成功后,多端同构框架可以读取样式配置进行UI渲染及事件绑定。

[0066] 步骤六、UI渲染成功,即显示插件给用户观看,用户可以对插件进行交互操作。

[0067] 步骤七、多端同构框架中的功能模块,可以确定对交互操作这一事件进行响应,按照业务逻辑进行处理,并将数据上报给服务器。

[0068] 步骤八、基于Web接口向用户返回交互结果。

[0069] 通过上述内容,对于该SDK的调用过程,可以如图5所示,该SDK支持多端调用,例如,WEB端、小程序端、小游戏端以及其他平台。该SDK可以作为插件的形式安装于各个平台,作为某个业务的实现。可以对该SDK进行初始化,可以进行各个平台的接口实现,该在任一个平台进行的调用过程中,SDK按照多端同构框架的规范提供当前平台的接口实现。例如,WEB接口实现、小程序接口实现、小游戏接口实现或其他接口实现,通过该接口实现,可以完成多端同构框架的注册和调用,进而实现对SDK的调用,该多端共用同一SDK,或者说该多端共用同一多端同构框架,可以实现统一业务逻辑。

[0070] 相较之下,如图6所示,相关技术中,开发者通常会开发一套框架,该框架可以包括前端逻辑代码,采用抽象语法树,可以将代码转换拆解为模板(template)文件、业务逻辑文件和样式配置文件,从而通过工程构建器将该多个文件构建为一个过程,部署于不同端。在部署时,需要根据要部署的端,进行框架转换,构建该端所适用的框架,再完成部署。则该相关技术中的部署过程的使用场景受限,该框架必须整体开发,无法像本发明实施例提供的SDK调用方法一样适用于插件类的产品开发,也无法在已有产品中引入。且该相关技术中的框架开发一般是针对于某个或某些平台开发的,有些平台则不支持,开发者需重新开发新的框架。该相关技术中的新特性接入受限,对于各个平台最新支持的新特性,开发者要么只能等待框架版本更新,要么只能手动修改已有的框架源码,而框架源码复杂度极高,开发者难以改动。

[0071] 本发明实施例提供的方法,将通用的业务逻辑在框架中进行统一管理,有利于减少开发时间及降低人力成本,有利于程序代码的维护及产品的长远发展。例如,在一种广告插件的应用场景中,假设广告插件分别支持三种不同的平台,在各个平台上开发新广告样式或新功能平均需要5人/天,那么通过上述相关技术的方式,三个平台的开发量总和是15人/天。而本发明实施例提供的方法由于样式配置及业务逻辑由同构框架统一管理,开发者

只需在框架内实现新广告样式或新功能即可,总开发量仅需5人/天。

[0072] 另外,本发明实施例提供的方法中,各个平台各自提供接口的实现,框架只负责统一的调度,有利于新平台及新特性的接入,有利于产品的快速发展。

[0073] 204、当接收到该SDK的更新请求时,各个服务器对该SDK包括的功能程序进行更新。

[0074] 如果SDK中的功能程序发生了更新,各个服务器可以直接对SDK中的功能程序进行更新,而无需像相关技术中,等待平台版本更新或重新开发,更新便利、快捷。

[0075] 在一种可能实现方式中,所述SDK包括的功能程序还可以基于语法抽象树实现,该步骤204则可以为:当接收到所述SDK的更新请求时,各个服务器基于所述更新请求的内容,对所述语法抽象树的结构进行更新。采用抽象语法树的方式,非常有利于专门开发业务逻辑程序的开发者对功能程序进行更新。对于任一端,也可以执行上述同理的步骤,对此不多做赘述。

[0076] 本发明实施例通过采用调用SDK的方式,在多端实现相同的功能,在调用过程中可以根据功能程序和各端对应的接口配置信息,对接口进行配置,从而可以直接对SDK的功能程序进行调用,以实现相应的功能,无需针对不同端各自开发一个工程文件,大大减少了开发时间和人工成本,该SDK调用方式的通用性好。

[0077] 上述所有可选技术方案,可以采用任意结合形成本发明的可选实施例,在此不再一一赘述。

[0078] 图7是本发明实施例提供的一种SDK调用装置的结构示意图,参见图7,该装置可以包括:

[0079] 获取模块701,用于获取软件开发工具包SDK,所述SDK包括功能程序和多端对应的接口配置信息;

[0080] 配置模块702,用于基于所述SDK包括的功能程序和当前端对应的接口配置信息,对目标接口进行配置;

[0081] 调用模块703,用于基于所述目标接口,调用所述SDK的功能程序,以执行所述SDK对应的功能。

[0082] 在一种可能实现方式中,所述SDK包括多个功能程序;

[0083] 所述配置模块702用于基于所述SDK包括的每个功能程序和当前端对应的接口配置信息,对多个目标接口中每个目标接口进行配置,每个目标接口与每个功能程序对应。

[0084] 在一种可能实现方式中,所述多个功能程序包括样式配置程序和业务逻辑程序;

[0085] 所述配置模块702用于:

[0086] 基于所述SDK包括的样式配置程序以及当前端对应的接口配置信息,对显示接口进行配置;

[0087] 基于所述SDK包括的业务逻辑程序以及当前端对应的接口配置信息,对业务处理接口进行配置。

[0088] 在一种可能实现方式中,所述装置还包括:

[0089] 更新模块,用于当接收到所述SDK的更新请求时,对所述SDK包括的功能程序进行更新。

[0090] 在一种可能实现方式中,所述SDK包括的功能程序可以基于语法抽象树实现;

[0091] 所述更新模块用于当接收到所述SDK的更新请求时,基于所述更新请求的内容,对所述语法抽象树的结构进行更新。

[0092] 在一种可能实现方式中,所述装置还包括:

[0093] 所述获取模块701,所述配置模块702或所述调用模块703,还用于当接收到对所述SDK的调用请求时,执行所述获取软件开发工具包SDK、配置和调用的步骤,所述调用请求基于对所述SDK的任一功能程序对应的插件进行的交互操作触发,所述调用请求包括所述插件的信息;

[0094] 所述调用模块703用于基于所述目标接口和所述插件的信息,调用所述SDK的所述插件对应的所述任一功能程序,执行所述任一功能程序对应的功能。

[0095] 本发明实施例提供的装置,通过采用调用SDK的方式,在多端实现相同的功能,在调用过程中可以根据功能程序和各端对应的接口配置信息,对接口进行配置,从而可以直接对SDK的功能程序进行调用,以实现相应的功能,无需针对不同端各自开发一个工程文件,大大减少了开发时间和人工成本,该SDK调用方式的通用性好。

[0096] 需要说明的是:上述实施例提供的SDK调用装置在调用SDK时,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将服务器的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。另外,上述实施例提供的SDK调用装置与SDK调用方法实施例属于同一构思,其具体实现过程详见方法实施例,这里不再赘述。

[0097] 图8是本发明实施例提供的一种服务器的结构示意图,该服务器800可因配置或性能不同而产生比较大的差异,可以包括一个或多个处理器(central processing units,CPU)801和一个或多个的存储器802,其中,所述一个或多个存储器802中存储有至少一条指令,所述至少一条指令由所述一个或多个处理器801加载并执行以实现上述各个方法实施例提供的SDK调用方法。当然,该服务器800还可以具有有线或无线网络接口、键盘以及输入输出接口等部件,以便进行输入输出,该服务器800还可以包括其他用于实现设备功能的部件,在此不做赘述。

[0098] 在示例性实施例中,还提供了一种计算机可读存储介质,例如包括指令的存储器,上述指令可由处理器执行以完成上述实施例中的SDK调用方法。例如,该计算机可读存储介质可以是只读存储器(Read-Only Memory,ROM)、随机存取存储器(Random Access Memory,RAM)、只读光盘(Compact Disc Read-Only Memory,CD-ROM)、磁带、软盘和光数据存储设备等。

[0099] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成,也可以通过程序来指令相关的硬件完成,该程序可以存储于一种计算机可读存储介质中,上述提到的存储介质可以是只读存储器,磁盘或光盘等。

[0100] 上述仅为本发明的较佳实施例,并不用以限制本发明,凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

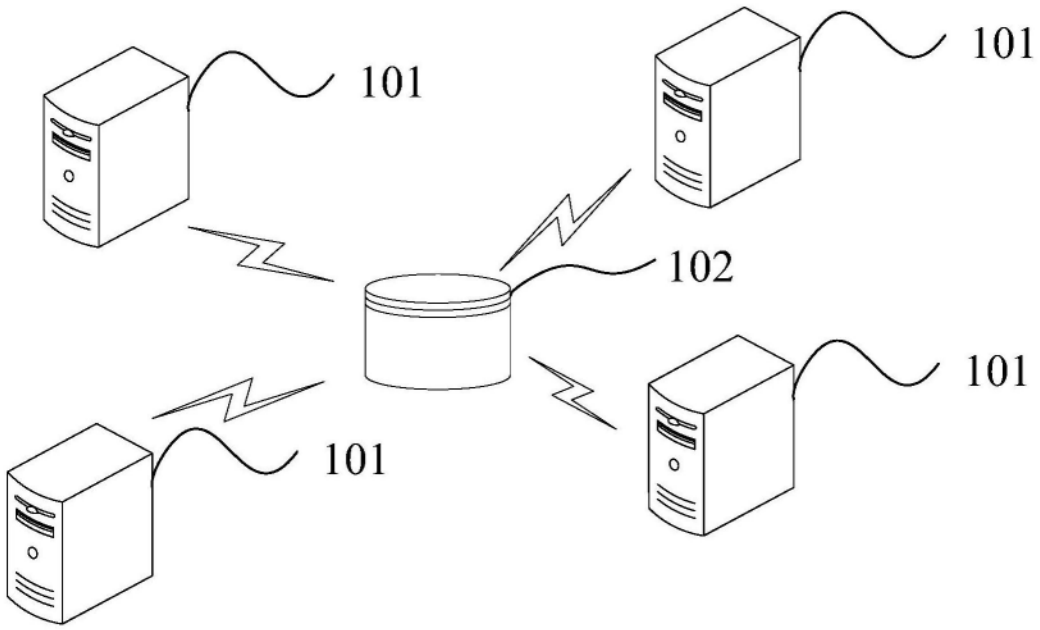

101
101
102
101
101

图1

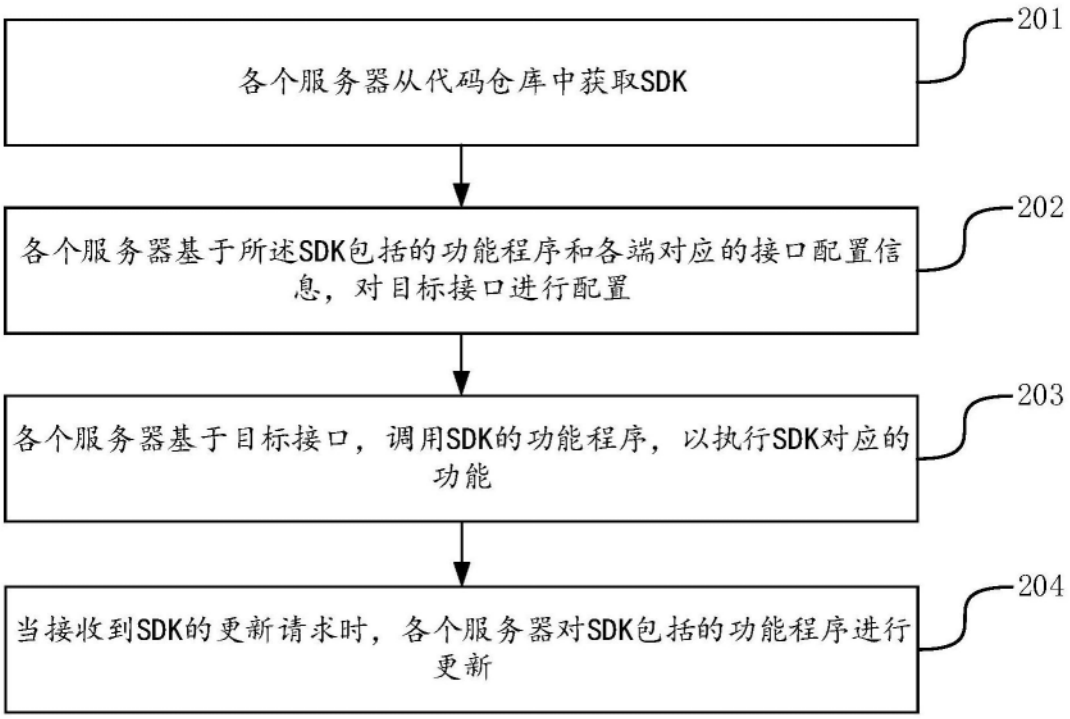

201
各个服务器从代码仓库中获取SDK
202
各个服务器基于所述SDK包括的功能程序和各端对应的接口配置信息，对目标接口进行配置
203
各个服务器基于目标接口，调用SDK的功能程序，以执行SDK对应的功能
204
当接收到SDK的更新请求时，各个服务器对SDK包括的功能程序进行更新

图2

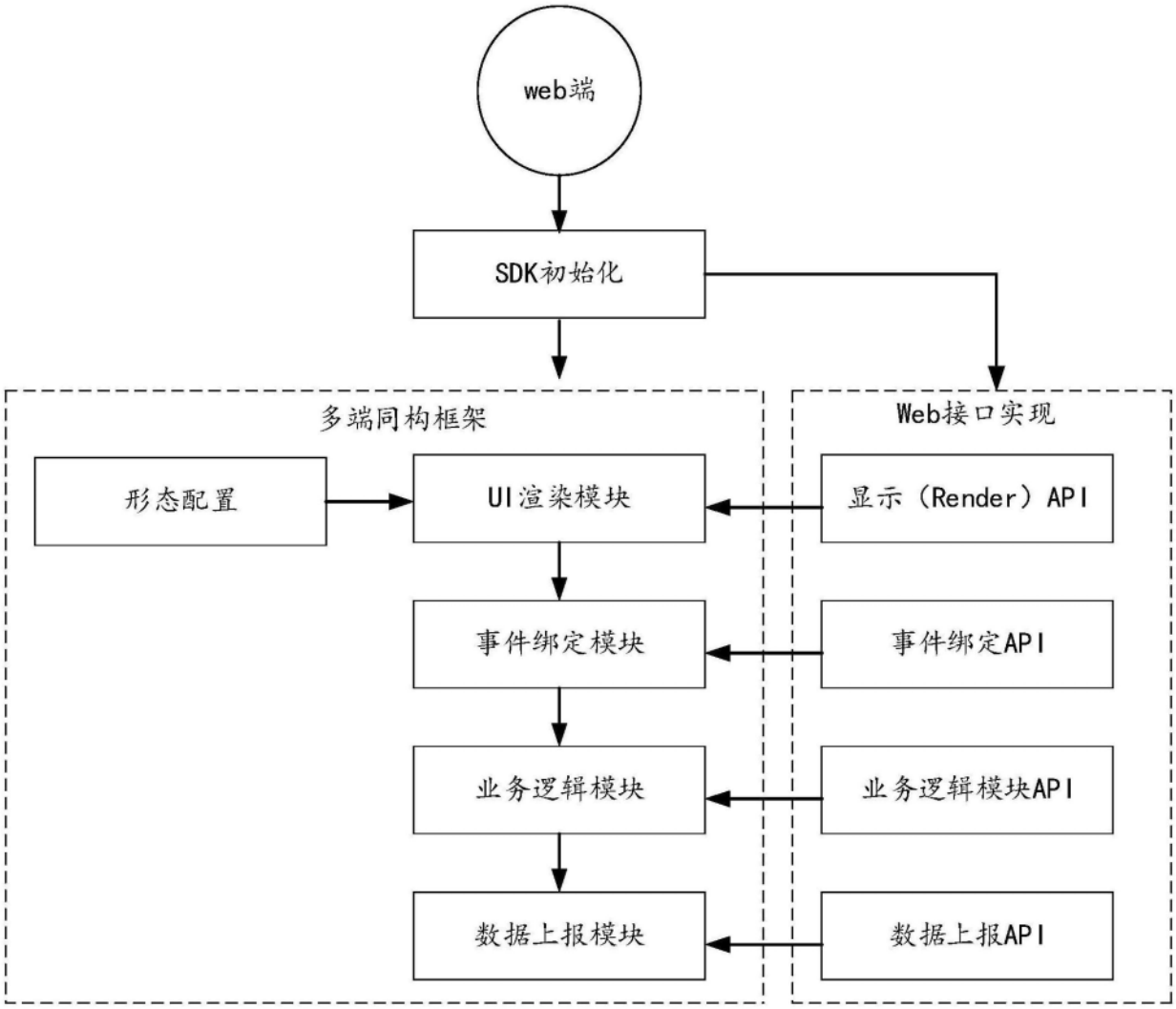

web端
SDK初始化
多端同构框架
Web接口实现
形态配置
UI渲染模块
显示（Render）API
事件绑定模块
事件绑定API
业务逻辑模块
业务逻辑模块API
数据上报模块
数据上报API

图3

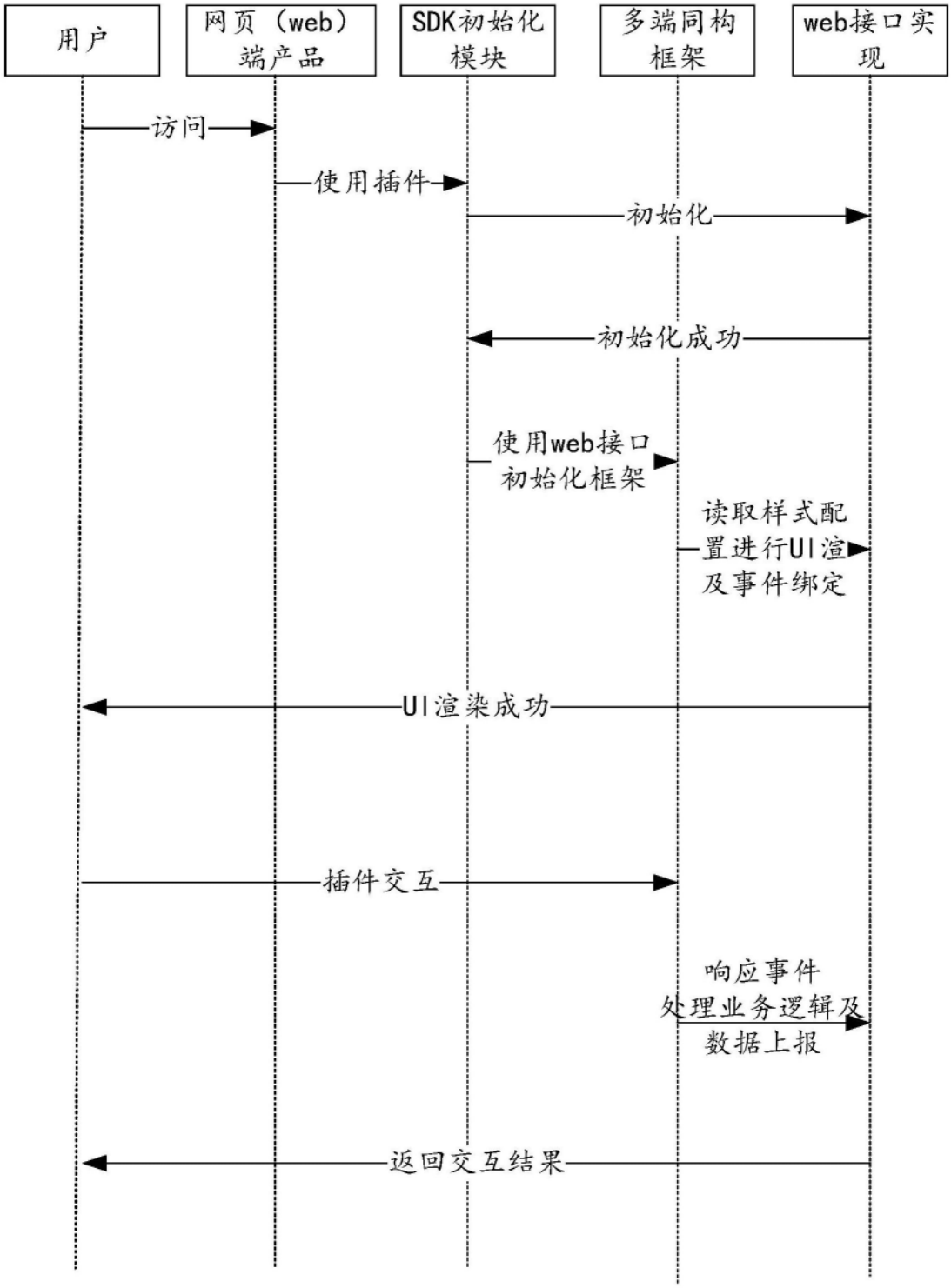

用户
网页（web）端产品
SDK初始化模块
多端同构框架
web接口实现
访问
使用插件
初始化
初始化成功
使用web接口初始化框架
读取样式配置进行UI渲及事件绑定
UI渲染成功
插件交互
响应事件处理业务逻辑及数据上报
返回交互结果

图4

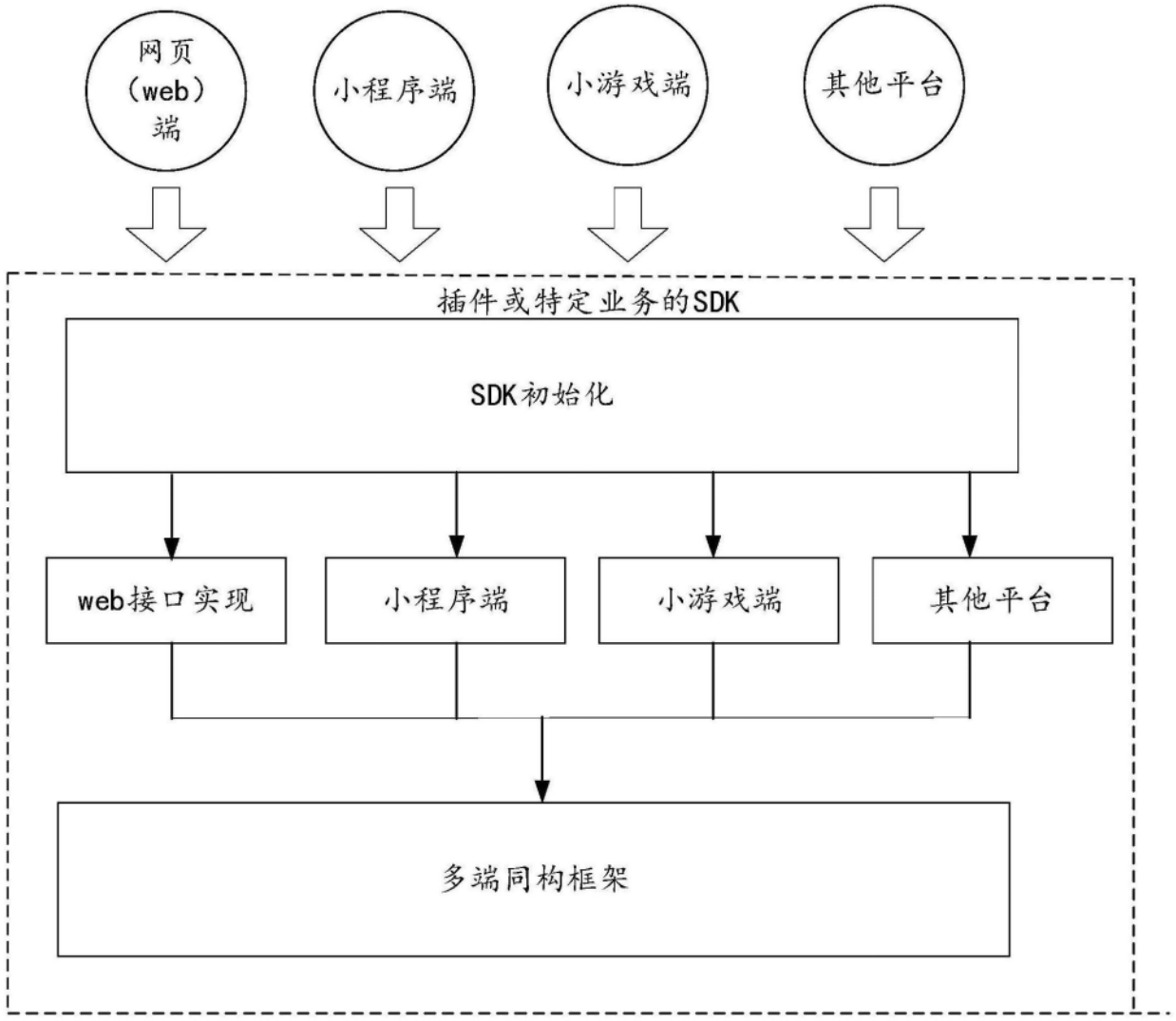

网页（web）端
小程序端
小游戏端
其他平台
插件或特定业务的SDK
SDK初始化
web接口实现
小程序端
小游戏端
其他平台
多端同构框架

图5

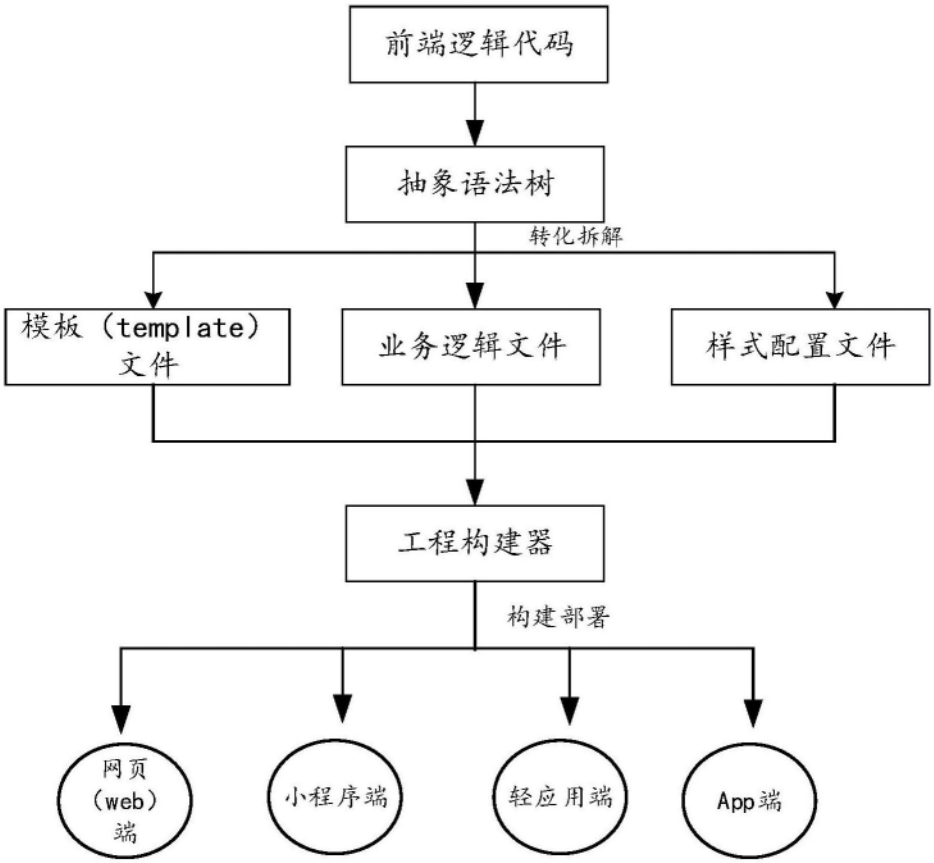

前端逻辑代码
抽象语法树
转化拆解
模板（template）文件
业务逻辑文件
样式配置文件
工程构建器
构建部署
网页（web）端
小程序端
轻应用端
App端

图6

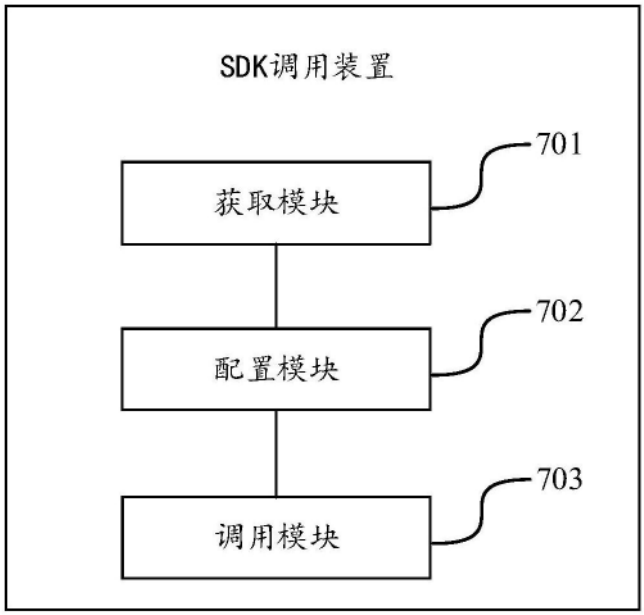

SDK调用装置
701
获取模块
702
配置模块
703
调用模块

图7

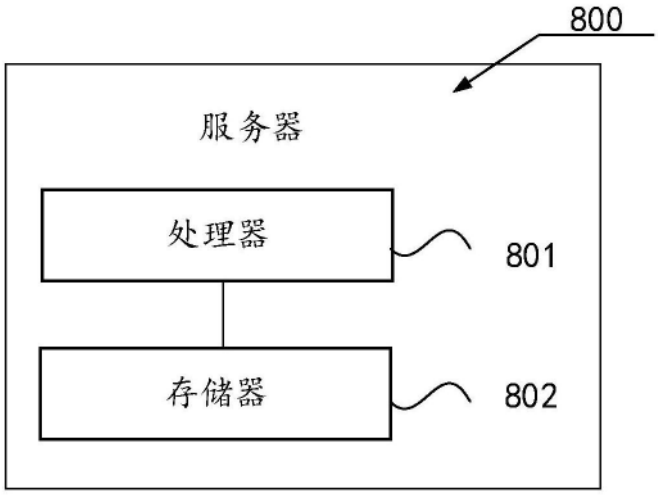


图8