



(12) 发明专利申请

(10) 申请公布号 CN 112150136 A

(43) 申请公布日 2020.12.29

(21) 申请号 202010924134.8

(22) 申请日 2020.09.04

(71) 申请人 京东数字科技控股股份有限公司
地址 100176 北京市大兴区北京经济技术开发区科创十一街18号C座2层221室

(72) 发明人 李腾

(74) 专利代理机构 北京华夏泰和知识产权代理有限公司 11662
代理人 卢万腾 曾军

(51) Int.Cl.

G06Q 20/22 (2012.01)

G06Q 20/38 (2012.01)

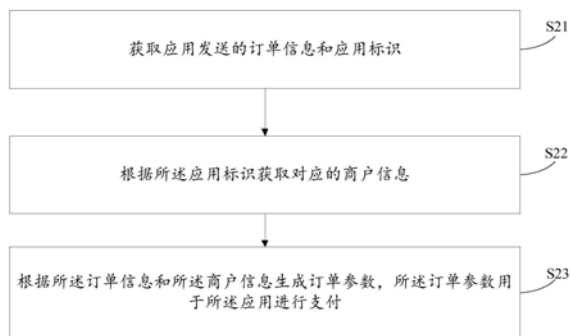
权利要求书3页 说明书14页 附图8页

(54) 发明名称

一种应用中内嵌网页的支付方法、系统及装置

(57) 摘要

本申请提供一种应用中内嵌网页的支付方法、系统及装置,服务器获取应用发送的订单信息和应用标识,根据应用标识获取对应的商户信息,根据商户信息和订单信息生成订单参数,订单参数用于应用进行原生支付。在本方案中,应用向服务器发送应用标识,服务器根据应用标识进行商户信息的匹配,确定与获取到的应用标识相匹配的商户信息,在应用时可以在服务器中预先设置多个商户与各自对应的应用的应用标识之间的对应关系,从而使得同一服务器可以应用于多个不同的商户对应的不同的应用的支付,提高了服务器的适用性,相比于现在每个商户都需要开发一个服务器,本方案减少了开发所需要的人力、物力。



1. 一种应用中内嵌网页的支付方法,其特征在于,应用于服务器,所述方法包括:
获取应用发送的订单信息和应用标识;
根据所述应用标识获取对应的商户信息;
根据所述订单信息和所述商户信息生成订单参数,所述订单参数用于所述应用进行支付。
2. 根据权利要求1所述的方法,其特征在于,获取应用发送的订单信息和应用标识,包括:
接收应用中内嵌的网页发送的下单请求,所述下单请求中包含订单信息;
基于所述下单请求,生成与所述订单信息对应的订单号;
基于所述订单号生成标识请求,并将所述标识请求发送至所述应用,所述标识请求用于请求获取应用标识;
接收所述应用响应于所述标识请求发送的应用标识和订单号;
根据接收的订单号获取对应的订单信息。
3. 根据权利要求2所述的方法,其特征在于,根据接收的订单号获取对应的订单信息之前,所述方法还包括:
判断接收的订单号与所述订单信息对应的订单号是否一致;
若一致,则执行根据接收的订单号获取对应的订单信息的步骤;
若不一致,则不执行根据接收的订单号获取对应的订单信息的步骤。
4. 根据权利要求1所述的方法,其特征在于,根据所述商户信息和所述订单信息生成订单参数,包括:
将所述订单信息与所述商户信息进行组合,得到组合信息;
采用预设的加密算法对所述组合信息进行加密,将加密后的组合信息作为订单参数。
5. 根据权利要求1所述的方法,其特征在于,所述订单参数用于所述应用标识对应的应用进行原生支付,包括:
所述应用接收所述服务器发送的所述订单参数,将所述订单参数发送至支付系统,所述支付系统根据所述订单参数生成支付单号,并将所述支付单号发送至所述应用从而使所述应用根据所述支付单号进行支付。
6. 一种应用中内嵌网页的支付方法,其特征在于,应用于应用,所述应用包含内嵌的网页,所述方法包括:
当检测到应用中内嵌的网页有下单操作时,向服务器发送订单信息和应用标识,以使所述服务器生成与所述订单信息和应用信息对应的订单参数;
获取支付系统基于所述订单参数生成的支付单号;
根据所述支付单号在目标原生页面调用原生支付接口,所述原生支付接口用于实现原生支付。
7. 根据权利要求6所述的方法,其特征在于,获取支付系统基于所述订单参数生成的支付单号,包括:
接收所述服务器返回的订单参数;
携带所述订单参数跳转至目标原生页面;
将所述订单参数发送至支付系统,以使所述支付系统根据所述订单参数生成对应的支

付单号；

接收所述支付系统发送的所述支付单号。

8. 根据权利要求6所述的方法,其特征在於,向服务器发送订单信息和应用标识,包括:应用中内嵌的网页向服务器发送下单请求,所述下单请求中包含订单信息,所述下单请求用于请求所述服务器生成与所述订单信息对应的订单号;

接收所述服务器发送的标识请求,所述标识请求用于请求获取应用标识,所述标识请求中包含所述订单号;

在确定所述网页内嵌在所述应用中时,将所述应用的应用标识和所述订单号发送至所述服务器。

9. 根据权利要求6所述的方法,其特征在於,根据所述支付单号在目标原生页面调用原生支付接口,包括:

获取应用标识、时间戳、随机串和签名方式;

根据所述原生支付调用请求中包含的支付单号生成数据包;

在所述目标原生页面根据所述应用标识、时间戳、随机串、签名方式和数据包调用原生支付接口。

10. 一种应用中内嵌网页的支付方法,其特征在於,应用于支付系统,所述方法包括:

接收订单参数;

调用预设接口,所述预设接口用于生成与所述订单参数对应的支付单号;

将所述支付单号发送至与所述订单参数对应的应用,以使所述应用根据所述支付单号进行支付。

11. 一种系统,其特征在於,所述系统用于实现权利要求1-10任一所述的应用中内嵌网页的支付方法,所述系统包括应用、服务器和支付系统,所述应用包含内嵌网页;

所述应用在检测到内嵌的网页中有下单操作时,向所述服务器发送订单信息和应用标识;

所述服务器根据所述应用标识获取对应的商户信息,并根据所述商户信息和所述订单信息生成订单参数;

所述支付系统根据所述订单参数生成对应的支付单号;

所述应用根据所述支付单号调用原生支付接口进行支付。

12. 一种应用中内嵌网页的支付装置,其特征在於,应用于服务器,所述装置包括:

第一获取模块,用于获取应用发送的订单信息和应用标识;

第二获取模块,用于根据所述应用标识获取对应的商户信息;

订单参数生成模块,用于根据所述订单信息和所述商户信息生成订单参数,所述订单参数用于所述应用进行支付。

13. 一种应用中内嵌网页的支付装置,其特征在於,应用于应用,所述应用包含内嵌网页,所述装置包括:

发送模块,用于当检测到应用中内嵌的网页有下单操作时,向服务器发送订单信息和应用标识,以使所述服务器生成与所述订单信息和应用信息对应的订单参数;

获取模块,用于获取支付系统基于所述订单参数生成的支付单号;

支付模块,用于根据所述支付单号在目标原生页面调用原生支付接口,所述原生支付

接口用于实现原生支付。

14. 一种应用中内嵌网页的支付装置,其特征在于,应用于支付系统,所述装置包括:
接收模块,用于接收订单参数;

支付单号生成模块,用于调用预设接口,所述预设接口用于生成与所述订单参数对应的支付单号。

发送模块,用于将所述支付单号发送至与所述订单参数对应的应用,以使所述应用根据所述支付单号进行支付。

15. 一种电子设备,其特征在于,包括:处理器和存储器,所述处理器用于执行所述存储器中存储的数据处理程序,以实现权利要求1-10任一所述的应用中内嵌网页的支付方法。

16. 一种存储介质,其特征在于,所述存储介质存储有一个或者多个程序,所述一个或者多个程序可被一个或者多个处理器执行,以实现权利要求1-10任一所述的应用中内嵌网页的支付方法。

一种应用中内嵌网页的支付方法、系统及装置

技术领域

[0001] 本申请涉及应用领域,尤其涉及一种应用中内嵌网页的支付方法、系统及装置。

背景技术

[0002] 微信小程序作为一种应用的展现方式,依托微信的流量在社交领域中有着极广泛的应用,本着用完即走、即开即用的原则,在低频快应用的场景中发挥着重要作用。因此从小程序诞生开始,各大应用相继推出了自己的微信小程序版本。

[0003] 微信小程序作为开发平台,提供了非常丰富的组件,能够满足应用程序(Application,简称app)的不同应用场景。此外微信小程序也提供了类似app的webview形式,开发者可以通过内嵌H5的形式迅速搭建应用,实现快速迭代。

[0004] 在互联网场景中,引导用户快速完成支付是黄金流程的关键。在微信小程序中可以使用内嵌webview的形式跳转收银台,并且收银台的提供方有很多,类似京东支付、银联支付等等,但以上的收银台均为H5形式,需要用户输入密码或短信验证码进行支付,无法使用指纹、人脸等生物识别技术,在安全性与便捷性方面大打折扣,并且微信环境下打开非微信支付可能会被微信封禁域名。

发明内容

[0005] 为了解决上述微信小程序在webview环境下无法使用指纹、人脸等生物识别技术进行支付的技术问题或者至少部分地解决上述技术问题,本申请提供了一种应用中内嵌网页的支付方法、系统及装置。

[0006] 第一方面,本申请提供了一种应用中内嵌网页的支付方法,应用于服务器,所述方法包括:

[0007] 获取应用发送的订单信息和应用标识;

[0008] 根据所述应用标识获取对应的商户信息;

[0009] 根据所述订单信息和所述商户信息生成订单参数,所述订单参数用于所述应用进行支付。

[0010] 在一种可能的实现方式中,获取应用发送的订单信息和应用标识,包括:

[0011] 接收应用中内嵌的网页发送的下单请求,所述下单请求中包含订单信息;

[0012] 基于所述下单请求,生成与所述订单信息对应的订单号;

[0013] 基于所述订单号生成标识请求,并将所述标识请求发送至所述应用,所述标识请求用于请求获取应用标识;

[0014] 接收所述应用响应于所述标识请求发送的应用标识和订单号;

[0015] 根据接收的订单号获取对应的订单信息。

[0016] 在一种可能的实现方式中,根据接收的订单号获取对应的订单信息之前,所述方法还包括:

[0017] 判断接收的订单号与所述订单信息对应的订单号是否一致;

- [0018] 若一致,则执行根据接收的订单号获取对应的订单信息的步骤;
- [0019] 若不一致,则不执行根据接收的订单号获取对应的订单信息的步骤。
- [0020] 在一种可能的实现方式中,根据所述商户信息和所述订单信息生成订单参数,包括:
- [0021] 将所述订单信息与所述商户信息进行组合,得到组合信息;
- [0022] 采用预设的加密算法对所述组合信息进行加密,将加密后的组合信息作为订单参数。
- [0023] 在一种可能的实现方式中,所述订单参数用于所述应用标识对应的应用进行原生支付,包括:
- [0024] 所述应用接收所述服务器发送的所述订单参数,将所述订单参数发送至支付系统,所述支付系统根据所述订单参数生成支付单号,并将所述支付单号发送至所述应用从而使所述应用根据所述支付单号进行支付。
- [0025] 第二方面,本申请实施例还提供了一种应用中内嵌网页的支付方法,应用于应用,所述应用包含内嵌的网页,所述方法包括:
- [0026] 当检测到应用中内嵌的网页有下单操作时,向服务器发送订单信息和应用标识,以使所述服务器生成与所述订单信息和应用信息对应的订单参数;
- [0027] 获取支付系统基于所述订单参数生成的支付单号;
- [0028] 根据所述支付单号在目标原生页面调用原生支付接口,所述原生支付接口用于实现原生支付。
- [0029] 在一种可能的实现方式中,获取支付系统基于所述订单参数生成的支付单号,包括:
- [0030] 接收所述服务器返回的订单参数;
- [0031] 携带所述订单参数跳转至目标原生页面;
- [0032] 将所述订单参数发送至支付系统,以使所述支付系统根据所述订单参数生成对应的支付单号;
- [0033] 接收所述支付系统发送的所述支付单号。
- [0034] 在一种可能的实现方式中,向服务器发送订单信息和应用标识,包括:
- [0035] 应用中内嵌的网页向服务器发送下单请求,所述下单请求中包含订单信息,所述下单请求用于请求所述服务器生成与所述订单信息对应的订单号;
- [0036] 接收所述服务器发送的标识请求,所述标识请求用于请求获取应用标识,所述标识请求中包含所述订单号;
- [0037] 在确定所述网页内嵌在所述应用中时,将所述应用的应用标识和所述订单号发送至所述服务器。
- [0038] 在一种可能的实现方式中,根据所述支付单号在目标原生页面调用原生支付接口,包括:
- [0039] 获取应用标识、时间戳、随机串和签名方式;
- [0040] 根据所述原生支付调用请求中包含的支付单号生成数据包;
- [0041] 在所述目标原生页面根据所述应用标识、时间戳、随机串、签名方式和数据包调用原生支付接口。

[0042] 第三方面,本申请实施例还提供了一种应用中内嵌网页的支付方法,应用于支付系统,所述方法包括:

[0043] 接收订单参数;

[0044] 调用预设接口,所述预设接口用于生成与所述订单参数对应的支付单号;

[0045] 将所述支付单号发送至与所述订单参数对应的应用,以使所述应用根据所述支付单号进行支付。

[0046] 第四方面,本申请实施例还提供了一种系统,所述系统用于实现第一方面、第二方面和第三方面任一所述的应用中内嵌网页的支付方法,所述系统包括应用、服务器和支付系统,所述应用包含内嵌网页;

[0047] 所述应用在检测到内嵌的网页中有下单操作时,向所述服务器发送订单信息和应用标识;

[0048] 所述服务器根据所述应用标识获取对应的商户信息,并

[0049] 根据所述商户信息和所述订单信息生成订单参数;

[0050] 所述支付系统根据所述订单参数生成对应的支付单号;

[0051] 所述应用根据所述支付单号调用原生支付接口进行支付。

[0052] 第五方面,本申请实施例还提供了一种应用中内嵌网页的支付装置,应用于服务器,所述装置包括:

[0053] 第一获取模块,用于获取应用发送的订单信息和应用标识;

[0054] 第二获取模块,用于根据所述应用标识获取对应的商户信息;

[0055] 订单参数生成模块,用于根据所述订单信息和所述商户信息生成订单参数,所述订单参数用于所述应用进行支付。

[0056] 第六方面,本申请实施例还提供了一种应用中内嵌网页的支付装置,应用于应用,所述应用包含内嵌网页,所述装置包括:

[0057] 发送模块,用于当检测到应用中内嵌的网页有下单操作时,向服务器发送订单信息和应用标识,以使所述服务器生成与所述订单信息和应用信息对应的订单参数;

[0058] 获取模块,用于获取支付系统基于所述订单参数生成的支付单号;

[0059] 支付模块,用于根据所述支付单号在目标原生页面调用原生支付接口,所述原生支付接口用于实现原生支付。

[0060] 第七方面,本申请实施例还提供了一种应用中内嵌网页的支付装置,应用于支付系统,所述装置包括:

[0061] 接收模块,用于接收订单参数;

[0062] 支付单号生成模块,用于调用预设接口,所述预设接口用于生成与所述订单参数对应的支付单号。

[0063] 发送模块,用于将所述支付单号发送至与所述订单参数对应的应用,以使所述应用根据所述支付单号进行支付。

[0064] 第八方面,本申请实施例还提供了一种电子设备,包括:处理器和存储器,所述处理器用于执行所述存储器中存储的数据处理程序,以实现第一方面、第二方面和第三方面任一所述的应用中内嵌网页的支付方法。

[0065] 第九方面,本申请实施例还提供了一种存储介质,所述存储介质存储有一个或者

多个程序,所述一个或者多个程序可被一个或者多个处理器执行,以实现第一方面、第二方面和第三方面任一所述的应用中内嵌网页的支付方法。

[0066] 相比现有技术,本申请实施例提供的上述技术方案与现有技术相比具有如下优点:

[0067] 本申请实施例提供了一种应用中内嵌网页的支付方法,服务器获取应用发送的订单信息和应用标识,根据应用标识获取对应的商户信息,根据商户信息和订单信息生成订单参数,订单参数用于应用进行支付。在本方案中,应用向服务器发送应用标识,服务器根据应用标识进行商户信息的匹配,确定与获取到的应用标识相匹配的商户信息,在应用时可以在服务器中预先设置多个商户与各自对应的应用的应用标识之间的对应关系,从而使同一服务器可以应用于多个不同的商户对应的不同的应用的支付,提高了服务器的适用性,相比于现在每个商户都需要开发一个服务器,本方案减少了开发所需要的人力、物力。

附图说明

[0068] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本发明的实施例,并与说明书一起用于解释本发明的原理。

[0069] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,对于本领域普通技术人员而言,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0070] 图1为本申请实施例提供了一种系统的支付方法流程图;

[0071] 图2为本申请实施例提供了一种应用于服务器的应用中内嵌网页的支付方法流程图;

[0072] 图3为本申请实施例提供了一种服务器获取应用发送的订单信息和应用信息的流程图;

[0073] 图4为本申请实施例提供了一种应用于应用的应用中内嵌网页的支付方法流程图;

[0074] 图5为本申请实施例提供了一种应用界面的示意图;

[0075] 图6为本申请实施例提供了一种应用向服务器发送订单信息和应用标识的流程图;

[0076] 图7为本申请实施例提供了一种应用获取支付系统基于所述订单参数生成的支付单号的流程图;

[0077] 图8为本申请实施例提供了一种应用于支付系统的应用中内嵌网页的支付方法流程图;

[0078] 图9为本申请实施例提供了一种应用于服务器的应用中内嵌网页的支付装置的框图;

[0079] 图10为本申请实施例提供了一种应用于应用的应用中内嵌网页的支付装置的框图;

[0080] 图11为本申请实施例提供了一种应用于支付系统的应用中内嵌网页的支付装置的框图;

[0081] 图12为本申请实施例提供了一种电子设备的示意图。

具体实施方式

[0082] 为使本申请实施例的目的、技术方案和优点更加清楚,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,所描述的实施例是本申请的一部分实施例,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做出创造性劳动的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。

[0083] 为便于对本申请实施例的理解,下面首先对本申请实施例涉及到的一些概念名词进行简要说明:

[0084] **webview**:是指网页视图。可以内嵌在移动端,实现前端的混合式开发,大多数混合式开发框架都是基于webview模式进行二次开发的。

[0085] **H5**:是指第5代HTML,也指用H5语言制作的一切数字产品,所谓HTML是超文本标记语言的英文缩写,平时上网所看到的网页多数都是由HTML写成的,超文本是指页面内可以包含图片,链接,甚至音乐,程序等非文字元素,而标记指的是这些超文本必须由包含属性的开头与结尾标志来标记,浏览器通过解码HTML,就可以把网页内容显示出来,它也构成了互联网兴起的基础。

[0086] **原生组件**:是由客户端创建的组件,原生组件脱离在webview渲染流程外。

[0087] **支付系统**:Payment System是由提供支付清算服务的中介机构和实现支付指令传送及资金清算的专业技术手段共同组成,用以实现债权债务清偿及资金转移的一种金融安排,有时也称为清算系统(Clear System)。

[0088] 现有技术中为了方便用户使用,提供了一种不需要下载安装即可使用的应用,例如微信小程序,它实现了应用“触手可及”的梦想,用户扫一扫或搜一下即可打开应用。

[0089] 为了减少微信小程序的开发成本,微信中提供了一种web view组件,在创建小程序时可以直接通过webview组件在小程序中内嵌网页,例如H5网页,这样网页中的内容就可以直接在小程序中显示了。

[0090] 为了实现小程序的微信支付,微信应用提供有唤起微信支付的**原生组件**,小程序在支付时通过调用微信支付的**原生组件**,可以使用微信的免密支付、指纹支付等能力。但是微信支付的**原生组件**需要在小程序的**原生页面**唤起,故对于webview环境下唤起**微信原生支付**并不友好。

[0091] 但是通过查看微信小程序的支付文档,发现小程序如果想要调用微信支付只需要得到小程序Id、时间戳、随机串、签名方式和数据包这5个支付参数即可,其中小程序Id是微信分配的appid,时间戳是当前时间,随机串是随机字符串,可以由随机数生成算法生成,数据包为统一下单接口返回的prepay_Id参数值,签名方式为预先设置的默认的签名类型,由此可见,在小程序端只要能通过某种方式得到webview内嵌网页下单后生成的上述相关参数就能调起微信支付从而实现webview组件内嵌网页的**微信原生支付**。

[0092] 基于上述原因,本申请提供了一种应用中内嵌网页的支付方法,通过该方法可以在小程序的webview环境下唤起**微信的原生支付**,实现了在web view环境下的**原生支付**,实现免密支付和指纹支付等。

[0093] 图1为本申请实施例提供的一种系统的支付方法流程图,如图1所示,该系统包括应用、服务器和支付系统,其中应用可以为包含内嵌网页的应用客户端,例如微信小程序,服务器为与应用对应的业务系统服务器,可以为实体服务器也可以为虚拟服务器,支付系

统是与应用对应的提供支付清算服务的中介机构,如图1所示,当应用的内嵌网页接收到用户的下单操作时,网页会向服务器发送下单请求,服务器接收到下单请求后,创建订单,并向应用发送标识请求,其中标识请求主要有两个作用,一个作用是请求获取应用标识,一个作用是请求验证发送下单请求的网页当前是否处于应用环境下,因为服务器接收到下单请求后无法判断这个请求是由一个内嵌在应用中的网页发送的还是由一个未内嵌在应用中的网页发送的,由于内嵌在应用中的网页和未内嵌在应用中的网页在支付时其对应的支付流程不同,所以服务器需要知道当前网页是否是内嵌在应用中的,应用接收到服务器发送的标识请求后,应用中的网页会判断当前是否处于应用环境下,若网页处于应用环境下,则应用响应于标识请求向服务器发送应用标识,服务器接收到应用返回的应用标识,则确定当前网页是内嵌在应用中,执行与内嵌网页对应的后续支付流程,若网页未处于应用环境下,则应用不会向服务器发送应用标识,服务器若没有接收到应用返回的应用标识,则说明当前网页就是一个单独的网页并没有内嵌到应用中,则此时执行正常网页的支付流程,本方案主要研究内嵌网页的支付,并不涉及正常网页的支付流程,故对此不进行过多描述,服务器中预先设置有应用标识与商户信息的对应关系,其中商户信息通常包含有商户名、商户Id等,通常一个应用对应一个商户,所谓商户通常指应用的开发商或所有者,当服务器接收到应用发送的应用标识后,获取与应用标识对应的商户信息,并将商户信息与下单请求中包含的订单信息进行组合和加密,从而生成订单参数,将订单参数发送给应用,应用需要将订单参数发送给支付系统用以请求支付单号,但是由于应用中的内嵌网页无法调用与支付系统进行通信的API,所以在应用接收到订单参数后需要携带订单参数先跳转至一个预设的目标原生页面,在目标原生页面调用相应的API从而将订单参数发送至支付系统,支付系统接收到订单参数后按照预先约定好的加密方法对其进行解密,然后调用预设接口(例如若应用为微信小程序,则此时调用微信的统一下单接口)生成与订单参数对应的支付单,支付单中包含支付单号,将支付单号发送至应用,应用接收到支付单号后即可调用用于实现原生支付的原生支付组件,从而完成原生支付。

[0094] 通过上述方式可以实现应用中内嵌网页的原生支付。

[0095] 下面分别以应用、服务器和支付系统为执行主体对本申请提供的应用中内嵌网页的支付方法进行详细说明。

[0096] 图2为以服务器为执行主体的一种应用中内嵌网页的支付方法的流程图,如图2所示,该方法可以包括:

[0097] S21. 获取应用发送的订单信息和应用标识。

[0098] 在一实施例中订单信息可以包含订单金额、订单商品Id、商品名称等等,应用标识可以为应用的appid,至于如何获取订单信息和应用标识,在下文中进行描述,此处先不详述。

[0099] S22. 根据所述应用标识获取对应的商户信息。

[0100] 在一实施例中,可以预先在服务器存储应用标识与商户信息的对应关系集合,集合中可以包含多个应用标识与商户信息的对应关系,但是在设置对应关系时需要保证一个应用标识只能对应一种商户信息,这就使得根据应用标识只会确定出唯一的商户信息,一种商户信息中可以包含一个商户的一种或多种信息,例如可以包含商户名称和/或商户Id等。

[0101] S23. 根据所述订单信息和所述商户信息生成订单参数,所述订单参数用于所述应用进行支付。

[0102] 其中所述订单参数用于所述应用进行支付指的是订单参数应用于应用进行订单支付的流程。

[0103] 本实施例提供的一种应用中内嵌网页的支付方法,服务器获取应用发送的订单信息和应用标识,根据应用标识获取对应的商户信息,根据商户信息和订单信息生成订单参数,订单参数用于应用进行支付。在本方案中,应用向服务器发送应用标识,服务器根据应用标识进行商户信息的匹配,确定与获取到的应用标识相匹配的商户信息,可以在服务器中预先设置多个商户的商户信息与应用的应用标识之间的对应关系,从而使得同一服务器可以实现对多种商户信息的获取,从而可以应用于多个不同的商户对应的应用的支付,提高了服务器的适用性,相比于现在每个商户都需要单独开发一个服务器以用于对应应用的支付,本方案减少了系统开发所需要的成本,提升了系统使用效率。

[0104] 在上述实施例的基础上,如图3所示,S21可以通过下述方式获取应用发送的订单信息和应用标识:

[0105] S211. 接收应用中内嵌的网页发送的下单请求,所述下单请求中包含订单信息。

[0106] 当应用中内嵌的网页中检测到有下单操作时,网页会向服务器发送下单请求。

[0107] S212. 基于所述下单请求,生成与所述订单信息对应的订单号。

[0108] 在一实施例中,对下单请求进行解析,得到下单请求中包含的订单信息,生成一个与订单信息生成的一个订单号,并对订单信息进行存储,根据该订单号即可获取对应的订单信息。

[0109] S213. 基于所述订单号生成标识请求,并将所述标识请求发送至所述网页,所述标识请求用于请求应用标识。

[0110] 在一实施例中,向网页发送标识请求主要是想要验证发送下单请求的网页当前是否处于应用环境下,以及获取应用标识,若网页处于应用环境下,则网页所处的应用会响应于标识请求返回应用标识,若网页不处于应用环境下,则其不会返回应用标识,也就是说服务器不会接收到应用标识。

[0111] 标识请求中还包含有订单号,当网页处于应用环境下时,响应于标识请求返回的信息中还包含有该订单号,目的是使服务器可以基于订单号获取对应的订单信息。

[0112] S214. 接收所述应用响应于所述标识请求发送的应用标识和订单号。

[0113] 在一实施例中,当服务器接收到返回的应用标识和订单号时,就可以确定网页当前处于应用环境下,此时执行S215。

[0114] 在一实施例中,若预设时长内没有接收到应用标识,或者接收到表示网页当前没有处于微信环境下的相关信息时,则确定网页不处于应用环境,则执行与不处于应用环境下的网页对应的支付流程,其支付流程可以为现有常用的网页支付流程,由于其与本申请无关,故不再具体描述。

[0115] S215. 根据接收的订单号获取对应的订单信息。

[0116] 此处接收的订单号指的是S214中接收到的应用响应于所述标识请求发送的订单号。

[0117] 因为在S212中已经存储了订单信息,所以根据订单号即可获取到对应的订单信

息。

[0118] 在本实施例中,通过上述流程可以验证当前下单的网页是否处于应用环境下,并且在处于应用环境下时,获取到应用标识,以便根据应用标识确定对应的商户信息。

[0119] 在上述实施例的基础上,在执行S215之前,还可以包括下述步骤:

[0120] 判断接收的订单号与所述订单信息对应的订单号是否一致,若一致,则执行根据接收的订单号获取对应的订单信息的步骤,若不一致,则不执行根据接收的订单号获取对应的订单信息的步骤。

[0121] 也就是在接收到响应于标识请求返回的订单号后,将该订单号与S212生成的订单号进行对比,若两者一致,则确定当前返回的应用标识是与S211中接收的下单请求对应的应用标识,然后获取与订单号对应的订单信息,根据订单信息和应用标识生成订单参数,若两者不一致,则说明当前接收到的应用标识与之前接收到的下单请求并不对应,可能出现了订单错误等问题,为了保证用户的财产安全此时可以终止支付流程。

[0122] 在本实施例中,对订单号进行验证,保证了订单的准确性,提高了支付的安全性。

[0123] 在上述实施例的基础上,S23根据所述订单信息和所述商户信息生成订单参数,可以包括:

[0124] 将订单信息和商户信息进行组合,得到组合信息,采用预设的加密算法对所述组合信息进行加密,将加密后的组合信息作为订单参数。

[0125] 在一实施例中,预设加密算法可以为根据需求设定的任意加密算法,例如AES (Advanced Encryption Standard,AES) 等加密算法。

[0126] 在本实施例中,为了防止订单参数被篡改对订单参数进行加密,保证了数据安全性。

[0127] 在上述实施例的基础上,S24中订单参数用于所述应用进行支付指的是:

[0128] 服务器将订单参数发送至所述应用,所述应用接收到所述服务器发送的所述订单参数后,将所述订单参数发送至支付系统,所述支付系统根据所述订单参数生成对应的支付单号,并将所述支付单号发送至所述应用从而使所述应用根据所述支付单号进行支付。

[0129] 在本实施例中,通过服务器将订单参数发送给应用,再由应用将订单参数发送至支付系统,减少了服务器与支付系统的通信,因为在本实施例中服务器可以用于多个应用的支付,而这些应用可能对应同一个支付系统,所以若通过服务器将订单参数发送至支付系统可能会由于应用过多,导致通信堵塞,降低了支付效率。

[0130] 在另一实施例中,S24中订单参数用于所述应用进行支付还可以为:

[0131] 服务器将订单参数发送至支付系统,支付系统根据订单参数生成对应的支付单号,将支付单号发送至所述应用,所述应用根据所述支付单号进行支付。

[0132] 在本实施例中,服务器直接将订单参数发送至支付系统,减少了一次服务器与应用交互,减少了订单参数丢失或被篡改的风险,提高了数据的安全性。

[0133] 图4为以应用为执行主体的一种应用中内嵌网页的支付方法的流程图,应用中包含内嵌的网页,如图4所示,该方法可以包括:

[0134] S41.当检测到应用中内嵌的网页有下单操作时,向服务器发送订单信息和应用标识。

[0135] 在一实施例中,应用可监听用户在所展示的应用界面上的触发事件,比如点击事

件、双击事件、滑动事件等。当应用的应用界面中展示的是网页时，应用在监听到任一上述触发事件时，获取触发时间的描述信息，根据确定接收到的触发事件的描述信息确定触发事件是否是下单操作。

[0136] 在一例子中，触发事件的描述信息包含触发事件在应用界面中对应的坐标位置，图5为应用界面的一个示意图，如图5所示，当应用界面中的“提交订单”这一图标被点击时就确定发生下单操作，因此可以预先存储“提交订单”这一图标的坐标位置，当检测到触发事件时将触发事件的坐标位置与“提交订单”这一图标的坐标位置进行对比，若触发事件的坐标位置落在了“提交订单”这一图标的坐标位置范围内，则确定检测到下单操作。

[0137] 在一实施例中，还可以通过语音检测模块来确定是否有下单操作，通过语音检测模块接收用户的语音信息，并对语音信息进行识别，当识别出语音信息中包含“下单”“提交订单”等关于下单之类的描述时，就确定检测到下单操作。

[0138] 上述是关于检测下单操作的描述只是示例，除了上述方式还可以采用其他方式来确定不是否有下单操作。

[0139] 至于是如何向服务器发送订单信息和应用标识的在下文中进行描述，这里先不详述。

[0140] S42.接收支付系统发送的基于所述订单参数生成的支付单号。

[0141] 在一实施例中，支付系统为与应用对应的提供支付清算等服务的中介机构，例如如果应用是京东旗下商户对应的应用，则其对应的支付系统可以为京东支付系统。至于是如何接收支付系统发送的支付单号的，在下文中进行描述，此处先不详述。

[0142] S43.根据所述支付单号在目标原生页面调用原生支付接口，所述原生支付接口用于实现原生支付。

[0143] 本申请实施例提供的一种应用中内嵌网页的支付方法，当应用中的内嵌的网页接收到下单操作后，向服务器发送订单信息和应用标识，以使服务器根据应用标识和订单信息生成订单参数，接收支付系统发送的基于订单参数生成的支付单号，然后根据支付单号在目标原生页面调用原生支付接口从而实现原生支付。在本方案中，应用向服务器发送应用标识，从而使得服务器可以根据应用标识进行商户信息的匹配，确定与获取到的应用标识相匹配的商户信息，可以在服务器中预先设置多个商户与各自对应的应用的应用标识之间的对应关系，从而使得同一服务器可以应用于多个不同的商户对应的不同的应用的支付，提高了服务器的适用性，相比于现在每个商户都需要开发一个服务器，本方案减少了开发所需要的人力、物力。

[0144] 在上述实施例的基础上，如图6所示，S41向服务器发送订单信息和应用标识可以包括：

[0145] S411.应用中内嵌的网页向服务器发送下单请求，所述下单请求中包含订单信息，所述下单请求用于请求所述服务器生成与所述订单信息对应的订单号。

[0146] S412.接收所述服务器发送的标识请求，所述标识请求用于请求所述应用发送应用标识，所述标识请求中包含所述订单号。

[0147] S413.在确定所述网页内嵌在所述应用中时，将所述应用的应用标识和所述订单号发送至所述服务器。

[0148] 在一实施例中，应用接收到服务器发送的标识请求后，发送下单请求的网页会调

用相应的接口来检测当前是否处于应用环境下,也就是判断网页是否内嵌在应用中,例如应用为微信小程序,则网页内可通过`window.__wxjs_environment`变量判断是否在小程序环境下,在确定网页处于应用环境下时,应用会将自身的应用标识发送给服务器,若确定网页未处于应用环境下,则不会发送应用标识。

[0149] 通过上述方式,既可以检测网页是否处于应用环境下,又可以使服务器获取到应用标识。

[0150] 在上述实施例的基础上,如图7所示,S42获取支付系统基于所述订单参数生成的支付单号可以包括:

[0151] S421.接收所述服务器返回的订单参数。

[0152] S422.携带所述订单参数跳转至目标原生页面。

[0153] 因为一般内嵌在应用中的网页无法直接调用用于与支付系统通信的接口,所以为了将订单参数发送至支付系统,需要先跳转至可以调用相应接口的目标原生页面。

[0154] 在一实施例中,目标原生页面为预先设置的原生页面,功能上该原生页面主要是为了收到web view网页的订单参数和支付参数,该原生页面可以为空白的页面,从用户使用角度上来说该原生页面是一个支付页面,因此可以在该页面中添加一些用户交互的组件,例如用于表示加载中的动画组件,比如转圈等。

[0155] 在一实施例中,可以通过调用相应的函数由当前内嵌的网页跳转至目标原生页面,比如若应用为微信小程序,则可以通过`wx.miniProgram.navigateBack`这一函数有小程序中的网页跳转至目标原生页面。

[0156] S423.将所述订单参数发送至支付系统,以使所述支付系统根据所述订单参数生成对应的支付单号。

[0157] S424.接收所述支付系统发送的所述支付单号。

[0158] 在本实施例中,通过服务器将订单参数发送给应用,再由应用将订单参数发送至支付系统,减少了服务器与支付系统的通信,因为在本实施例中服务器可以用于多个应用的支付,而这些应用可能对应同一个支付系统,所以若通过服务器将订单参数发送至支付系统可能会由于应用过多,导致通信堵塞,降低了支付效率。

[0159] 在另一实施例中,S42获取支付系统基于所述订单参数生成的支付单号还可以采用下述方式:

[0160] 服务器将订单参数发送至支付系统,支付系统根据订单参数生成与订单参数对应的支付单号,将支付单号发送至应用,从而使应用获取到支付单号。

[0161] 在本实施例中,服务器直接将订单参数发送至支付系统,减少了一次服务器与应用的交互,减少了订单参数丢失或被篡改的风险,提高了数据的安全性。

[0162] 在上述实施例的基础上,若S42采用的是S421-S424的方式获取支付单号,接收到支付单后执行S43即可,但是若S42采用的是上述另一种获取支付单号的方法获取的支付单号,则在接收到支付单号后还需先携带所述支付单号跳转至目标原生页面,然后再执行S43,因为网页无法直接调用原生支付接口。

[0163] 在上述实施例的基础上,S43.根据所述支付单号在目标原生页面调用原生支付接口,所述原生支付接口用于实现原生支付,可以包括:

[0164] 获取应用标识、时间戳、随机串和签名方式;

[0165] 根据所述原生支付调用请求中包含的支付单号生成数据包；

[0166] 在所述目标原生页面根据所述应用标识、时间戳、随机串、签名方式和数据包调用原生支付接口。

[0167] 原生支付接口为预先设置的接口，可以直接调用。

[0168] 在一例子中，如果应用为微信小程序，则原生支付接口可以为 wx.requestPayment，根据应用标识、时间戳、随机串、签名方式和数据包调用上述接口即可调起微信的原生支付，从而实现微信小程序的微信支付。

[0169] 在一实施例中，应用中内嵌网页的支付方法还可以包括：

[0170] 完成原生支付后再跳转至应用中内嵌的网页，并在网页中显示支付结果。

[0171] 通过在网页中显示支付结果可以使用户知悉到支付结果，提升用户体验。

[0172] 图8为以支付系统为执行主体的一种应用中内嵌网页的支付方法的流程图，如图8所示，该方法可以包括：

[0173] S81. 接收订单参数。

[0174] 在一实施例中，订单参数可以是应用发送的。

[0175] 在另一实施中，订单参数可以是服务器发送的。

[0176] 在一实施例中，若订单参数被加密了，则在接收到订单参数后先根据预先约定的订单参数的加密算法对订单参数进行解密，然后基于解密后的订单参数执行S82。

[0177] S82. 调用预设接口，所述预设接口用于生成与所述订单参数对应的支付单号。

[0178] 在一实施例中，若应用为微信小程序则预设接口可以为微信的统一下单接口。

[0179] S83. 将所述支付单号发送至与所述订单参数对应的应用，以使所述应用根据所述支付单号进行支付。

[0180] 在一实施例中，根据订单参数采用微信统一下单接口进行统一下单，即可得到与订单参数对应的支付单号。

[0181] 在本实施例中，通过支付系统接收到订单参数后，即可调用预设接口进行统一下单得到支付单号，可以实现多个应用共用一个支付系统。

[0182] 一个具体的例子：

[0183] 下面以内嵌有网页的微信小程序为例对本申请提供的应用中内嵌网页的支付方法进行描述。

[0184] 微信小程序中展示内嵌网页的内容，当检测到当前展示的网页接收到下单操作时，向小程序的业务系统服务器发送下单请求，下单请求中包含订单信息，订单信息中包含订单金额、订单商品、商品Id等信息，服务器接收到下单请求后创建订单，并生成与订单信息对应的订单号，基于订单号生成标识请求，将标识请求发送至微信小程序，微信小程序接收到标识请求后检测网页是否处于微信小程序，也就是判断网页是否是内嵌与微信小程序中的网页，在确定网页处于微信小程序环境后，将微信小程序的appid和接收到的订单号发送至服务器，服务器根据接收到的appid获取对应的与微信小程序对应的商户信息，根据订单号获取对应的订单信息，将订单信息和商户信息进行拼接组合后采用预设的加密算法进行加密得到订单参数，将订单参数发送在微信小程序，微信小程序携带订单参数跳转至预设的目标原生页面，将订单参数发送中支付系统，支付系统对订单参数进行解密，然后调用微信统一下单接口生成与订单参数对应的支付单号，将支付单号发送至微信小程序，微信

小程序根据预设的支付文档得到微信小程序的appid、时间戳、随机串、签名方式和数据包这5个支付参数,然后根据支付参数即可在目标原生页面调用微信支付的原生组件,从而实现微信小程序中网页的微信支付。

[0185] 目前很多公司可能会维护多个不同主体(商户)的微信小程序,这些小程序的归属主体不同、收款账户也不同,但是这些小程序可以共用同一个支付系统,为了保证提单、资金账目的正确,本申请中在服务器预先配置微信小程序的appid与商户信息之间的对应关系,在微信小程序向服务器请求参数中,在请求中添加appid,从而使得服务器可以根据appid获取对应的商户信息,从而实现利用同一个服务器和支付系统解决多小程序支付的问题。

[0186] 另外本申请中在支付时会涉及到参数的传递,为了防止参数在传递过程中被篡改,对参数进行加密,从而保证了数据的安全。

[0187] 本申请实施例还提供了一种应用于服务器的应用中内嵌网页的支付装置,如图9所示,该装置可以包括:

[0188] 第一获取模块901,用于获取应用发送的订单信息和应用标识;

[0189] 第二获取模块902,用于根据所述应用标识获取对应的商户信息;

[0190] 订单参数生成模块903,用于根据所述订单信息和所述商户信息生成订单参数,所述订单参数用于所述应用进行支付。

[0191] 在一实施例中,第一获取模块901具体用于:

[0192] 接收应用中内嵌的网页发送的下单请求,所述下单请求中包含订单信息;

[0193] 基于所述下单请求,生成与所述订单信息对应的订单号;

[0194] 基于所述订单号生成标识请求,并将所述标识请求发送至所述应用,所述标识请求用于请求获取应用标识;

[0195] 接收所述应用发送的应用标识和订单号;

[0196] 根据接收的订单号获取对应的订单信息。

[0197] 在一实施例中,该装置还可以包括:验证模块(图中未示出)

[0198] 所述验证模块具体用于:

[0199] 判断接收的订单号与所述订单信息对应的订单号是否一致;

[0200] 若一致,则执行根据接收的订单号获取对应的订单信息的步骤;

[0201] 若不一致,则不执行根据接收的订单号获取对应的订单信息的步骤。

[0202] 在一实施例中,订单参数生成模块903具体用于:

[0203] 将所述订单信息与所述商户信息进行组合,得到组合信息;

[0204] 采用预设的加密算法对所述组合信息进行加密,将加密后的组合信息作为订单参数。

[0205] 本申请实施例还提供了一种应用于应用的应用中内嵌网页的支付装置,如图10所示,该装置可以包括:

[0206] 发送模块1001,用于当检测到应用中内嵌的网页有下单操作时,向服务器发送订单信息和应用标识,以使所述服务器生成与所述订单信息和应用信息对应的订单参数;

[0207] 获取模块1002,用于获取支付系统基于所述订单参数生成的支付单号;

[0208] 支付模块1003,用于根据所述支付单号在目标原生页面调用原生支付接口,所述

原生支付接口用于实现原生支付。

[0209] 在一实施例中,所述获取模块1002具体用于:

[0210] 接收所述服务器返回的订单参数;

[0211] 携带所述订单参数跳转至目标原生页面;

[0212] 将所述订单参数发送至支付系统,以使所述支付系统根据所述订单参数生成对应的支付单号;

[0213] 接收所述支付系统发送的所述支付单号。

[0214] 在一实施例中,所述发送模块1001具体用于:

[0215] 应用中内嵌的网页向服务器发送下单请求,所述下单请求中包含订单信息,所述下单请求用于请求所述服务器生成与所述订单信息对应的订单号;

[0216] 接收所述服务器发送的标识请求,所述标识请求用于请求获取应用标识,所述标识请求中包含所述订单号;

[0217] 在确定所述网页内嵌在所述应用中时,将所述应用的应用标识和所述订单号发送至所述服务器。

[0218] 在一实施例中,所述支付模块1003具体用于:

[0219] 获取应用标识、时间戳、随机串和签名方式;

[0220] 根据所述原生支付调用请求中包含的支付单号生成数据包;

[0221] 在所述目标原生页面根据所述应用标识、时间戳、随机串、签名方式和数据包调用原生支付接口。

[0222] 本申请实施例还提供了一种应用于支付系统的应用中内嵌网页的支付装置,如图11所示,该装置可以包括:

[0223] 接收模块1101,用于接收订单参数。

[0224] 在一实施例中,订单参数可以是应用发送的。

[0225] 在另一实施中,订单参数可以是服务器发送的。

[0226] 在一实施例中,若订单参数被加密了,则可以根据预先约定的订单参数的加密算法对订单参数进行解密。

[0227] 支付单号生成模块1102,用于调用预设接口,所述预设接口用于生成与所述订单参数对应的支付单号。

[0228] 发送模块1103,用于将所述支付单号发送至与所述订单参数对应的应用,以使所述应用根据所述支付单号进行支付。

[0229] 在上述实施例的基础上,该装置还可以包括解密模块(图中未示出),所述解密模块用于对接收的订单参数进行解密。

[0230] 在本申请另一实施例中,还提供了一种电子设备,如图12所示,包括处理器1201、通信接口1202、存储器1203和通信总线1204,其中,处理器1201,通信接口1202,存储器1203通过通信总线1204完成相互间的通信;

[0231] 存储器1203,用于存放计算机程序;

[0232] 处理器1201,用于执行存储器1203上所存放的程序时,实现如下步骤:

[0233] 获取订单信息和应用标识,根据所述应用标识获取对应的商户信息,根据所述订单信息和所述商户信息生成订单参数,所述订单参数用于所述应用标识对应的应用进行支

付。

[0234] 或

[0235] 当检测到应用中内嵌的网页有下单操作时,向服务器发送订单信息和应用标识,以使所述服务器生成与所述订单信息和应用信息对应的订单参数,获取支付系统基于所述订单参数生成的支付单号,根据所述支付单号在目标原生页面调用原生支付接口,所述原生支付接口用于实现原生支付。

[0236] 或

[0237] 接收订单参数,调用预设接口生成与所述订单参数对应的支付单号,将所述支付单号发送至与所述订单参数对应的应用,以使所述应用根据所述支付单号进行支付。

[0238] 上述电子设备提到的通信总线1204可以是外设部件互连标准(Peripheral Component Interconnect,简称PCI)总线或扩展工业标准结构(Extended Industry Standard Architecture,简称EISA)总线等。该通信总线1204可以分为地址总线、数据总线、控制总线等。为便于表示,图中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0239] 通信接口1202用于上述电子设备与其他设备之间的通信。

[0240] 存储器1203可以包括随机存取存储器(Random Access Memory,简称RAM),也可以包括非易失性存储器(non-volatile memory),例如至少一个磁盘存储器。可选的,存储器还可以是至少一个位于远离前述处理器的存储装置。

[0241] 上述的处理器1201可以是通用处理器,包括中央处理器(Central Processing Unit,简称CPU)、网络处理器(Network Processor,简称NP)等;还可以是数字信号处理器(Digital Signal Processing,简称DSP)、专用集成电路(Application Specific Integrated Circuit,简称ASIC)、现场可编程门阵列(Field-Programmable Gate Array,简称FPGA)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。

[0242] 在本申请另一实施例中,还提供了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上存储方法程序,所述方法程序被处理器执行时实现上述任一所述的应用中内嵌网页的支付方法的步骤。

[0243] 本发明实施例在具体实现时,可以参阅上述各个实施例,具有相应的技术效果。

[0244] 需要说明的是,在本文中,诸如“第一”和“第二”等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0245] 以上所述仅是本发明的具体实施方式,使本领域技术人员能够理解或实现本发明。对这些实施例的多种修改对本领域的技术人员来说将是显而易见的,本文中所定义的一般原理可以在不脱离本发明的精神或范围的情况下,在其它实施例中实现。因此,本发明将不会被限制于本文所示的这些实施例,而是要符合与本文所申请的原理和新颖特点相一致的最宽的范围。

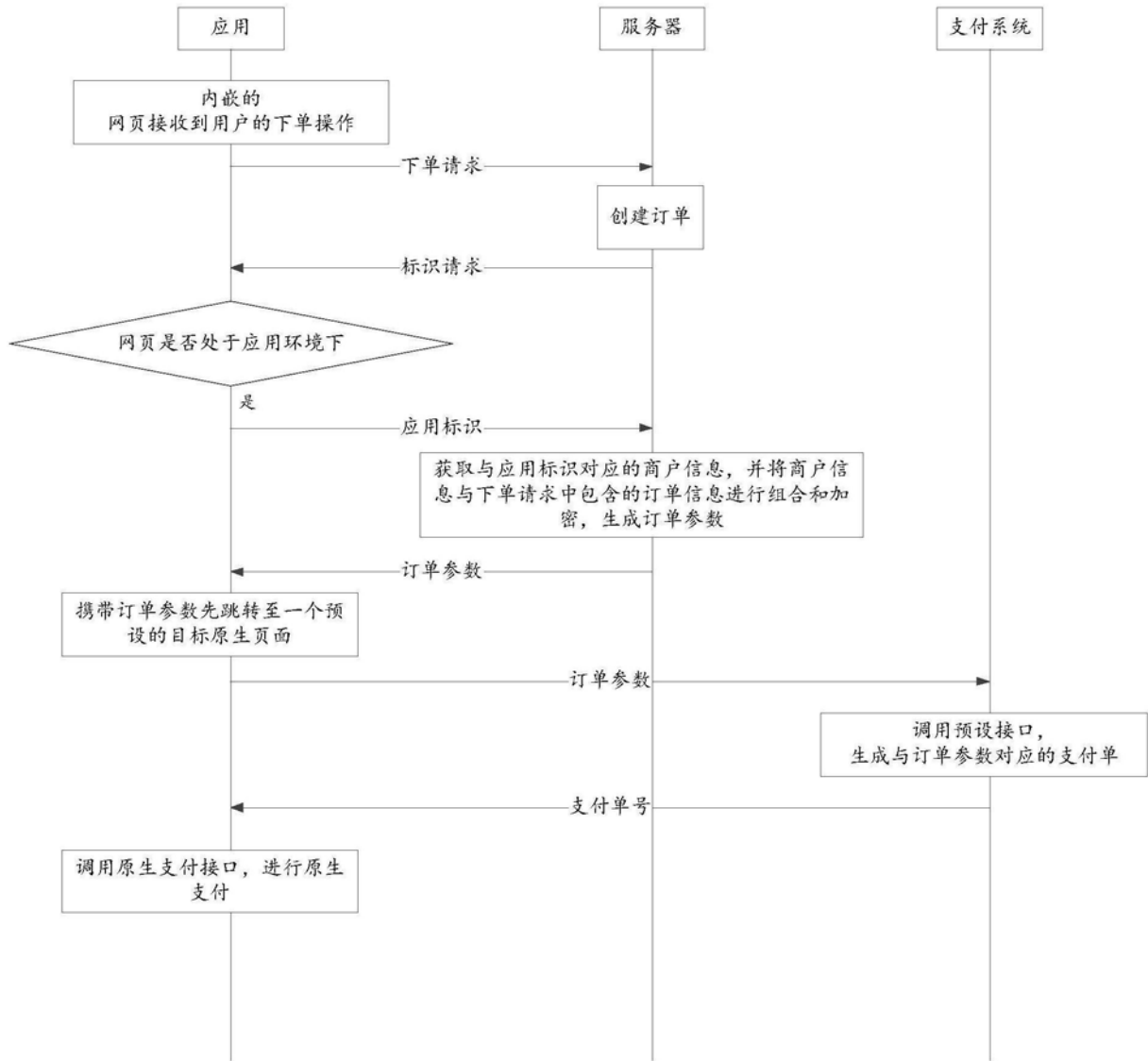


图1

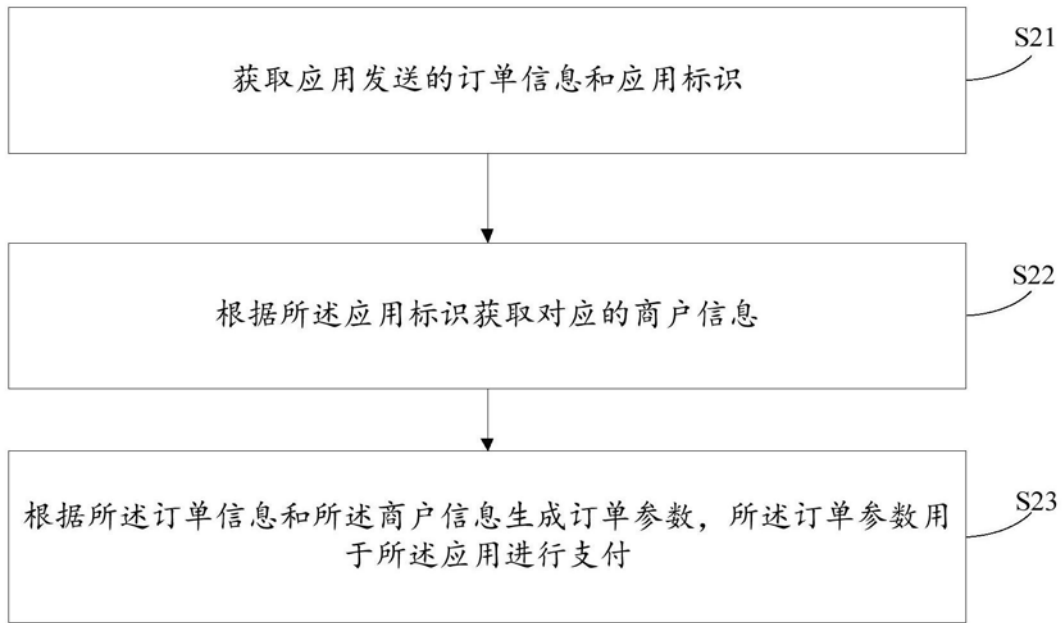


图2

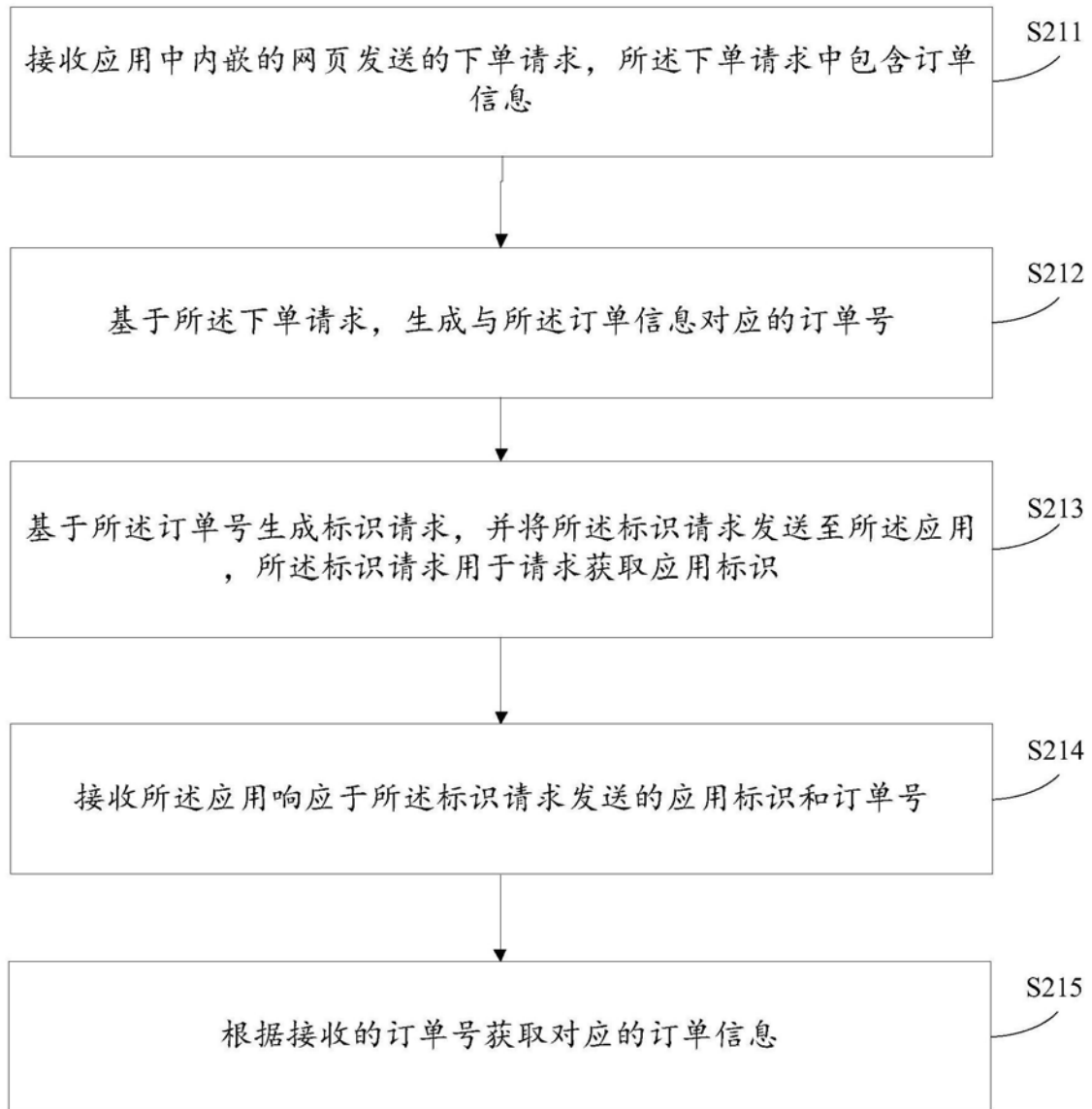


图3

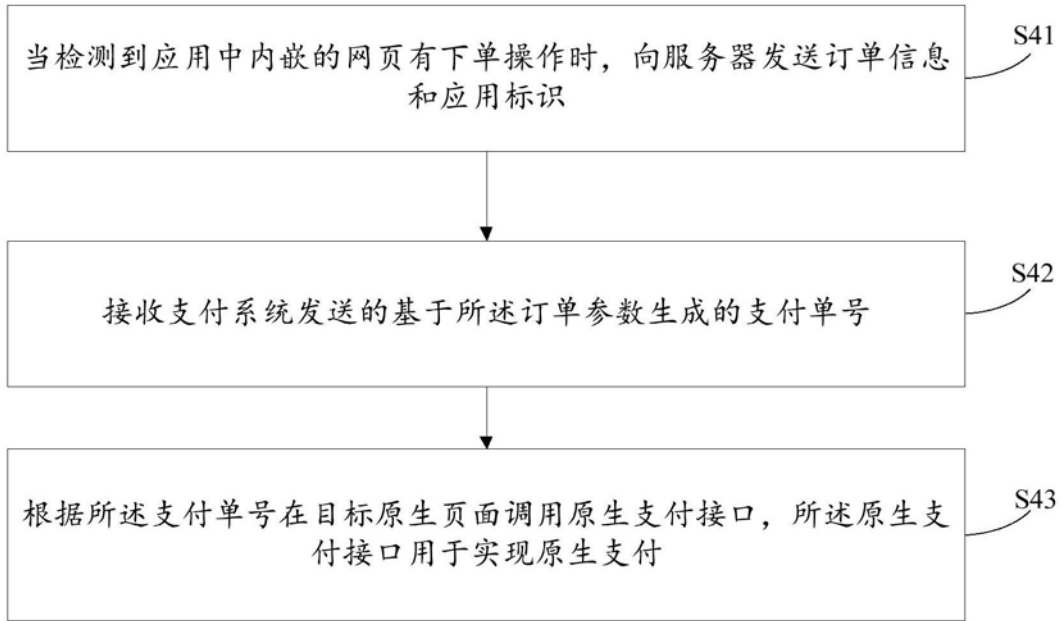


图4

确认订单

收货人: XXX
收货地址: XXXXXXXXXXXXXXX

商品名称: XXXXXX
商品金额: XXXXXX

应付金额: XXX 提交订单

图5

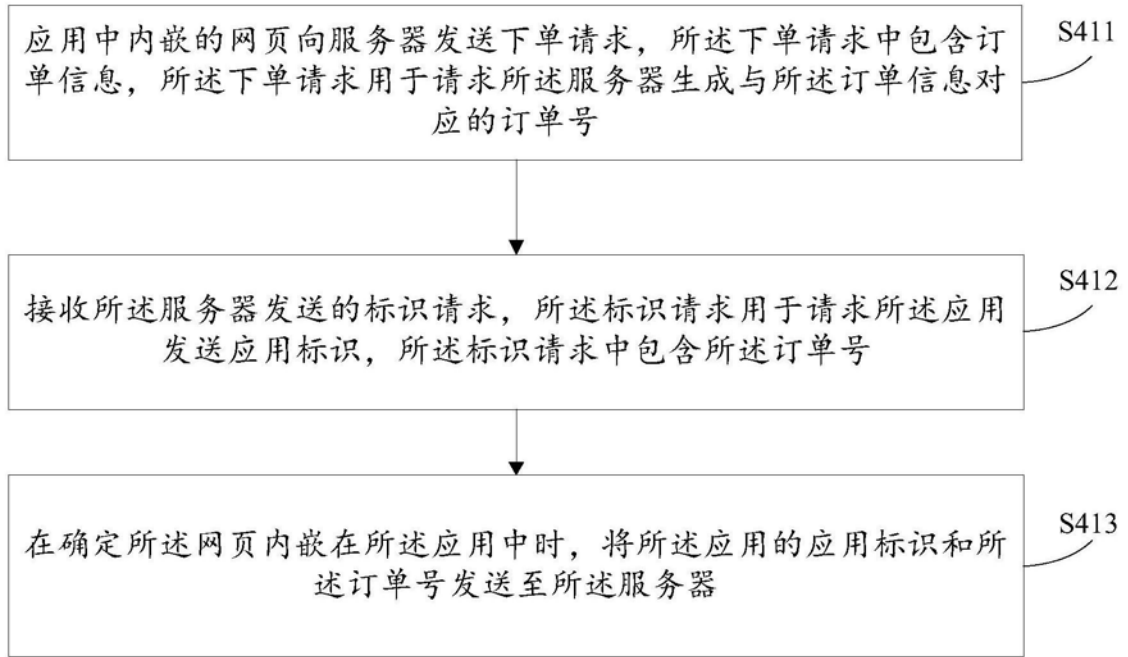


图6

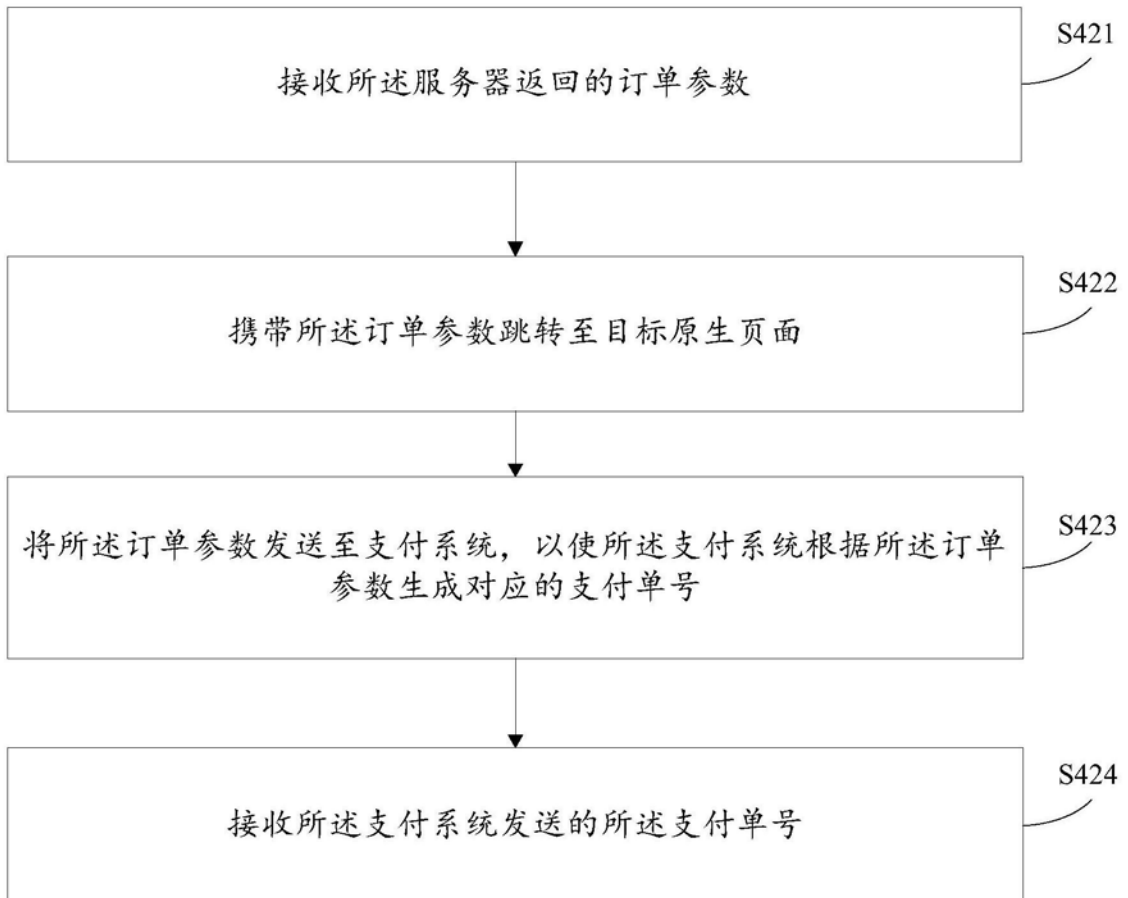


图7

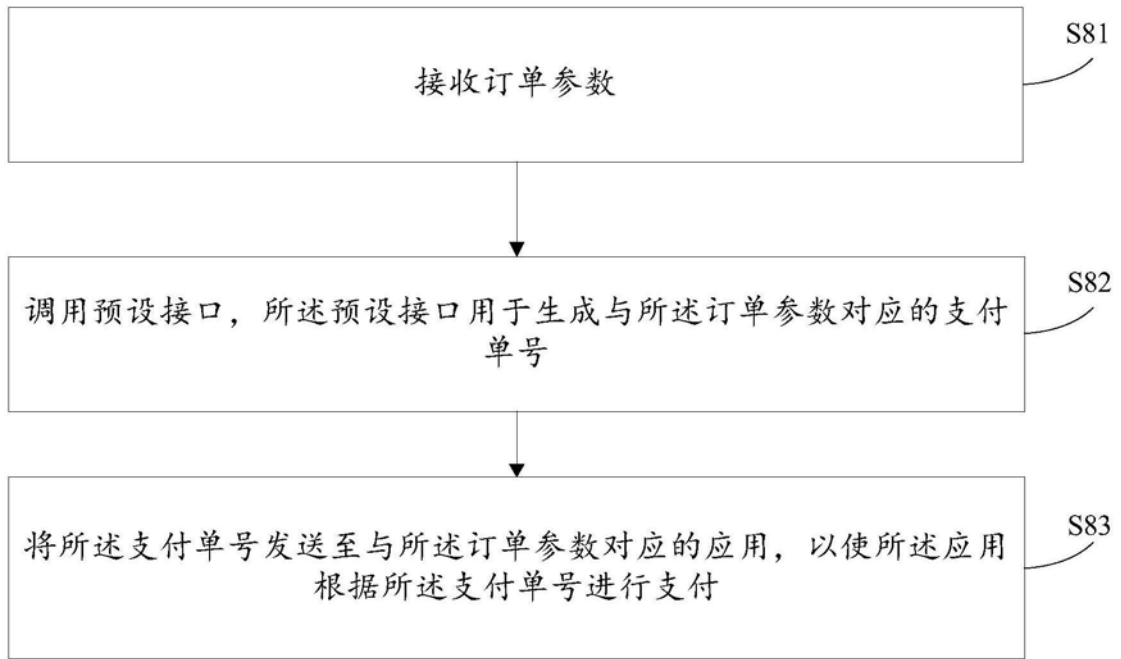


图8

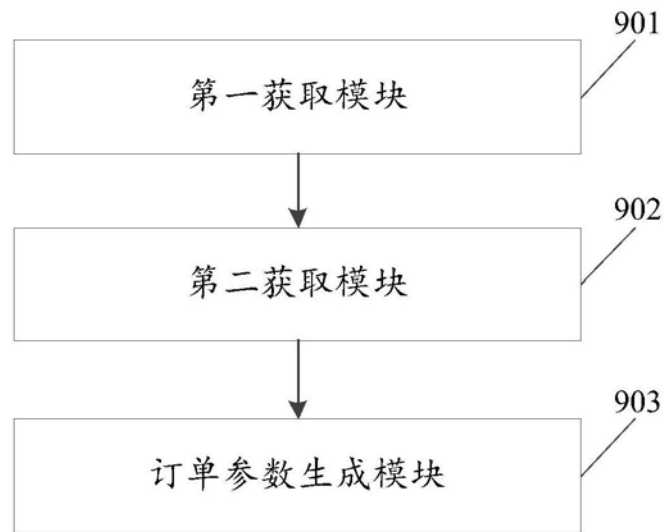


图9

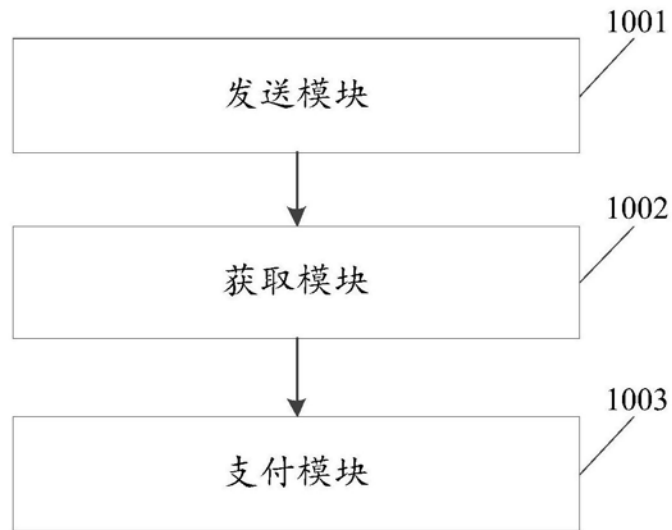


图10

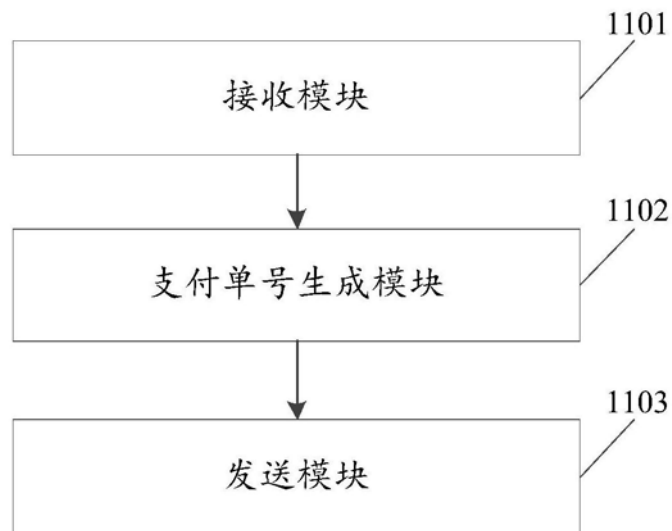


图11

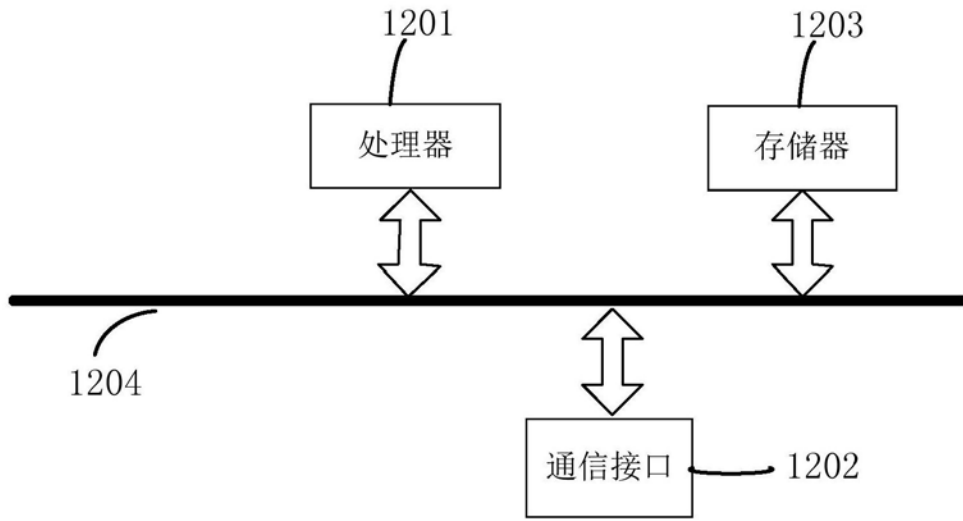


图12