



(12) 发明专利申请

(10) 申请公布号 CN 114880020 A

(43) 申请公布日 2022. 08. 09

(21) 申请号 202210619979.5

(22) 申请日 2022.06.01

(71) 申请人 北京百度网讯科技有限公司
地址 100085 北京市海淀区上地十街10号
百度大厦2层

(72) 发明人 侯峰 张少辉 王晓荣

(74) 专利代理机构 北京英赛嘉华知识产权代理
有限责任公司 11204
专利代理师 王达佐 马晓亚

(51) Int. Cl.
G06F 8/71 (2018.01)
G06F 8/10 (2018.01)

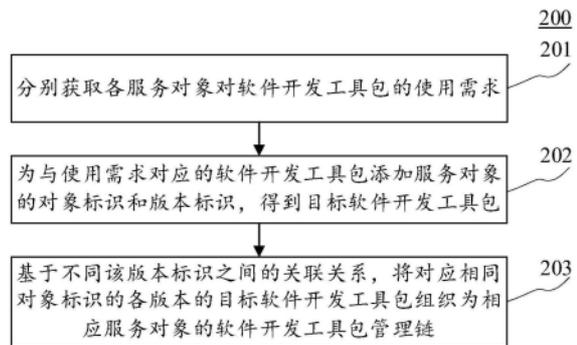
权利要求书2页 说明书10页 附图4页

(54) 发明名称

软件开发工具包管理方法、相关装置及计算机程序产品

(57) 摘要

本公开提供了一种软件开发工具包管理方法、装置、电子设备、计算机可读存储介质及计算机程序产品,涉及资源管理、云存储、智能云等人工智能技术领域。该方法的一具体实施方式包括:分别获取各服务对象对软件开发工具包的使用需求后,为与使用需求对应的软件开发工具包添加该服务对象的对象标识和版本标识,得到目标软件开发工具包,并基于不同该版本标识之间的关联关系,将对应相同对象标识的各版本的目标软件开发工具包组织为相应服务对象的软件开发工具包管理链。该实施方式可利用软件开发工具包管理链实现软件开发工具包的管理,以便于利用同一管理体系为不同服务对象提供软件开发工具包配置服务,降低管理和开发成本。



1. 一种软件开发工具包管理方法,包括:
 - 分别获取各服务对象对软件开发工具包的使用需求;
 - 为与所述使用需求对应的软件开发工具包添加所述服务对象的对象标识和版本标识,得到目标软件开发工具包;
 - 基于不同所述版本标识之间的关联关系,将对应相同对象标识的各版本的目标软件开发工具包组织为相应服务对象的软件开发工具包管理链。
2. 根据权利要求1所述的方法,还包括:
 - 响应于第一服务对象的首个目标软件开发工具包与第二服务对象的目标历史软件开发工具包的功能相同,在与所述第二服务对象对应的软件开发工具包管理链上的目标历史软件开发工具包处,以创建分支链的方式创建所述第一服务对象的软件开发工具包管理链。
3. 根据权利要求1所述的方法,还包括:
 - 接收所述服务对象传入的再次开发请求;
 - 将所述再次开发请求指示的目标软件开发工具包,确定为再次开发起点的基础版本软件开发工具包;
 - 根据所述再次开发请求中的再次开发需求,对所述基础版本软件开发工具包进行再次开发,得到附加新版本标识的新目标软件开发工具包,所述新版本标识基于所述基础版本软件开发工具包的版本标识确定;
 - 在所述软件开发工具包管理链中与所述基础版本软件开发工具包对应的位置,生成分支链或延续链,并将所述新目标软件开发工具包置入所述分支链或所述延续链。
4. 根据权利要求1所述的方法,其中,所述为与所述使用需求对应的软件开发工具包添加所述服务对象的对象标识和版本标识,得到目标软件开发工具包,包括:
 - 根据所述使用需求确定需求代码;
 - 从预先配置的代码库中获取与所述需求代码对应的目标代码文件;
 - 打包所述目标代码文件得到与所述使用需求对应的软件开发工具包,并添加所述服务对象的对象标识和版本标识,得到所述目标软件开发工具包。
5. 根据权利要求1所述的方法,其中,所述关联关系包括:各所述版本标识分别对应的版本号之间的大小关系,对应的,所述软件开发工具包管理链包括:各所述目标软件开发工具包按版本号从小到大、从左至右的顺序组织出的时序链。
6. 根据权利要求1-5中任一项所述的方法,还包括:
 - 分别确定不同服务对象的数据存储路径;
 - 将与各所述服务对象对应的软件开发工具包管理链,通过相应的数据存储路径存储至相应的数据存储空间;其中,所述数据存储路径包括:公有云路径和私有云路径,所述数据存储空间包括:公有云空间和私有云空间。
7. 一种软件开发工具包管理装置,包括:
 - 使用需求获取单元,被配置成分别获取各服务对象对软件开发工具包的使用需求;
 - 工具包生成单元,被配置成为与所述使用需求对应的软件开发工具包添加所述服务对象的对象标识和版本标识,得到目标软件开发工具包;
 - 管理链组织单元,被配置成基于不同所述版本标识之间的关联关系,将对应相同对象

标识的各版本的目标软件开发工具包组织为相应服务对象的软件开发工具包管理链。

8. 根据权利要求7所述的装置,还包括:

分支链创建单元,被配置成响应于第一服务对象的首个目标软件开发工具包与第二服务对象的目标历史软件开发工具包的功能相同,在与所述第二服务对象对应的软件开发工具包管理链上的目标历史软件开发工具包处,以创建分支链的方式创建所述第一服务对象的软件开发工具包管理链。

9. 根据权利要求7所述的装置,还包括:

再次开发请求接收单元,被配置成接收所述服务对象传入的再次开发请求;

基础工具包确定单元,被配置成将所述再次开发请求指示的目标软件开发工具包确定为再次开发起点的基础版本软件开发工具包;

新工具包生成单元,被配置成根据所述再次开发请求中的再次开发需求,对所述基础版本软件开发工具包进行再次开发,得到附加新版本标识的新目标软件开发工具包,所述新版本标识基于所述基础版本软件开发工具包的版本标识确定;

管理链更新单元,被配置成在所述软件开发工具包管理链中与所述基础版本软件开发工具包对应的位置,生成分支链或延续链,并将所述新目标软件开发工具包置入所述分支链或所述延续链。

10. 根据权利要求7所述的装置,其中,所述工具包生成单元被进一步配置成:

根据所述使用需求确定需求代码;

从预先配置的代码库中获取与所述需求代码对应的目标代码文件;

打包所述目标代码文件得到与所述使用需求对应的软件开发工具包,并添加所述服务对象的对象标识和版本标识,得到所述目标软件开发工具包。

11. 根据权利要求7所述的装置,其中,所述关联关系包括:各所述版本标识分别对应的版本号之间的大小关系,对应的,所述软件开发工具包管理链包括:各所述目标软件开发工具包按版本号从小到大、从左至右的顺序组织出的时序链。

12. 根据权利要求7-11中任一项所述的装置,还包括:

存储路径获取单元,被配置成分别确定不同的服务对象的数据存储路径;

管理链存储单元,被配置成将与各所述服务对象对应的软件开发工具包管理链,通过相应的数据存储路径存储至相应的数据存储空间;其中,所述数据存储路径包括:公有云路径和私有云路径,所述数据存储空间包括:公有云空间和私有云空间。

13. 一种电子设备,包括:

至少一个处理器;以及

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行权利要求1-6中任一项所述的软件开发工具包管理方法。

14. 一种存储有计算机指令的非瞬时计算机可读存储介质,所述计算机指令用于使所述计算机执行权利要求1-6中任一项所述的软件开发工具包管理方法。

15. 一种计算机程序产品,包括计算机程序,所述计算机程序在被处理器执行时实现根据权利要求1-6中任一项所述的软件开发工具包管理方法。

软件开发工具包管理方法、相关装置及计算机程序产品

技术领域

[0001] 本公开涉及计算机技术领域,具体涉及资源管理、云存储、智能云等人工智能技术领域,尤其涉及软件开发工具包管理方法、装置、电子设备、计算机可读存储介质及计算机程序产品。

背景技术

[0002] 当前,医疗机构往往选择与外部机构进行合作的方式,利用由外部机构配置的软件开发工具包(Software Development Kit,简称SDK)实现医疗机构的智能化建设。

[0003] 为方便外部机构对软件开发工具包进行管理、配置,以及方便各医疗机构方便、快速的获取相关的软件开发工具包,外部机构往往通过共有云部署、私有云部署两种方式,实现指向医疗机构的软件开发工具包配置服务。

发明内容

[0004] 本公开实施例提出了一种软件开发工具包管理方法、装置、电子设备、计算机可读存储介质及计算机程序产品。

[0005] 第一方面,本公开实施例提出了一种软件开发工具包管理方法,包括:分别获取各服务对象对软件开发工具包的使用需求;为与该使用需求对应的软件开发工具包添加该服务对象的对象标识和版本标识,得到目标软件开发工具包;基于不同该版本标识之间的关联关系,将对应相同对象标识的各版本的目标软件开发工具包组织为相应服务对象的软件开发工具包管理链。

[0006] 第二方面,本公开实施例提出了一种软件开发工具包管理装置,包括:使用需求获取单元,被配置成分别获取各服务对象对软件开发工具包的使用需求;工具包生成单元,被配置成为与该使用需求对应的软件开发工具包添加该服务对象的对象标识和版本标识,得到目标软件开发工具包;管理链组织单元,被配置成基于不同该版本标识之间的关联关系,将对应相同对象标识的各版本的目标软件开发工具包组织为相应服务对象的软件开发工具包管理链。

[0007] 第三方面,本公开实施例提供了一种电子设备,该电子设备包括:至少一个处理器;以及与至少一个处理器通信连接的存储器;其中,存储器存储有可被至少一个处理器执行的指令,该指令被至少一个处理器执行,以使至少一个处理器执行时能够实现如第一方面中任一实现方式描述的软件开发工具包管理方法。

[0008] 第四方面,本公开实施例提供了一种存储有计算机指令的非瞬时计算机可读存储介质,该计算机指令用于使计算机执行时能够实现如第一方面中任一实现方式描述的软件开发工具包管理方法。

[0009] 第五方面,本公开实施例提供了一种包括计算机程序的计算机程序产品,该计算机程序在被处理器执行时能够实现如第一方面中任一实现方式描述的软件开发工具包管理方法。

[0010] 本公开实施例提供的软件开发工具包管理方法、装置、电子设备、计算机可读存储介质及计算机程序产品,分别获取各服务对象对软件开发工具包的使用需求后,为与使用需求对应的软件开发工具包添加该服务对象的对象标识和版本标识,得到目标软件开发工具包,并基于不同该版本标识之间的关联关系,将对应相同对象标识的各版本的目标软件开发工具包组织为相应服务对象的软件开发工具包管理链。

[0011] 本公开可利用软件开发工具包管理链实现软件开发工具包的管理,以便于利用同一管理体系为不同服务对象提供软件开发工具包配置服务,降低管理和开发成本。

[0012] 应当理解,本部分所描述的内容并非旨在标识本公开的实施例的关键或重要特征,也不用于限制本公开的范围。本公开的其它特征将通过以下的说明书而变得容易理解。

附图说明

[0013] 通过阅读参照以下附图所作的对非限制性实施例所作的详细描述,本公开的其它特征、目的和优点将会变得更明显:

[0014] 图1是本公开可以应用于其中的示例性系统架构;

[0015] 图2为本公开实施例提供的一种软件开发工具包管理方法的流程图;

[0016] 图3为本公开实施例提供的另一种软件开发工具包管理方法的流程图;

[0017] 图4为本公开实施例提供的在一应用场景下的软件开发工具包管理方法的流程示意图;

[0018] 图5为本公开实施例提供的一种软件开发工具包管理装置的结构框图;

[0019] 图6为本公开实施例提供的一种适用于执行软件开发工具包管理方法的电子设备的结构示意图。

具体实施方式

[0020] 以下结合附图对本公开的示范性实施例做出说明,其中包括本公开实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本公开的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。需要说明的是,在不冲突的情况下,本公开中的实施例及实施例中的特征可以相互组合。

[0021] 此外,本公开涉及的技术方案中,所涉及的用户个人信息的获取、存储、使用、加工、运输、提供和公开等处理,均符合相关法律法规的规定,且不违背公序良俗。

[0022] 图1示出了可以应用本公开的软件开发工具包管理方法、装置、电子设备及计算机可读存储介质的实施例的示例性系统架构100。

[0023] 如图1所示,系统架构100可以包括终端设备101、102、103,网络104和服务器105。网络104用以在终端设备101、102、103和服务器105之间提供通信链路的介质。网络104可以包括各种连接类型,例如有线、无线通信链路或者光纤电缆等等。

[0024] 用户可以使用终端设备101、102、103通过网络104与服务器105交互,以接收或发送消息等。终端设备101、102、103和服务器105上可以安装有各种用于实现两者之间进行信息通讯的应用,例如软件开发工具包配置类应用、软件开发工具包开发类应用、即时通讯类应用等。

[0025] 终端设备101、102、103和服务器105可以是硬件,也可以是软件。当终端设备101、102、103为硬件时,可以是具有显示屏的各种电子设备,包括但不限于智能手机、平板电脑、膝上型便携计算机和台式计算机等等;当终端设备101、102、103为软件时,可以安装在上述所列举的电子设备中,其可以实现成多个软件或软件模块,也可以实现成单个软件或软件模块,在此不做具体限定。当服务器105为硬件时,可以实现成多个服务器组成的分布式服务器集群,也可以实现成单个服务器;服务器为软件时,可以实现成多个软件或软件模块,也可以实现成单个软件或软件模块,在此不做具体限定。

[0026] 服务器105通过内置的各种应用可以提供各种服务,以可以提供软件开发工具包配置服务的软件开发工具包配置类应用为例,服务器105在运行该软件开发工具包配置类应用时可实现如下效果:首先,服务器105通过网络104从终端设备101、102、103中获取各服务对象(即使用终端设备101、102、103的各用户)中,分别获取各服务对象对软件开发工具包的使用需求;然后,服务器105为与该使用需求对应的软件开发工具包添加该服务对象的对象标识和版本标识,得到目标软件开发工具包;最后,服务器105基于不同该版本标识之间的关联关系,将对应相同对象标识的各版本的目标软件开发工具包组织为相应服务对象的软件开发工具包管理链。

[0027] 需要指出的是,各服务对象的使用需求除可以从终端设备101、102、103通过网络104获取到之外,也可以通过各种方式预先存储在服务器105本地。因此,当服务器105检测到本地已经存储有这些数据时(例如开始处理之前留存的软件开发工具包的配置任务),可选择直接从本地获取这些数据。

[0028] 由于进行目标软件开发工具包的开发、配置需要占用较多的运算资源和较强的运算能力,因此本公开后续各实施例所提供的软件开发工具包管理方法一般由拥有较强运算能力、较多运算资源的服务器105来执行,相应地,软件开发工具包管理装置一般也设置于服务器105中。

[0029] 应该理解,图1中的终端设备、网络和服务器的数目仅仅是示意性的。根据实现需要,可以具有任意数目的终端设备、网络和服务器。

[0030] 请参考图2,图2为本公开实施例提供的一种软件开发工具包管理方法的流程图,其中流程200包括以下步骤:

[0031] 步骤201,分别获取各服务对象对软件开发工具包的使用需求。

[0032] 在本实施例中,由软件开发工具包管理方法的执行主体(例如图1所示的服务器105)分别获取各服务对象的软件开发包工具包使用需求,该使用需求可以包括软件开发包工具包的功能需求、更新软件开发包工具包的配置需求、优化软件开发包工具包的优化需求等。

[0033] 在实践中,在获取各服务对象对软件开发工具包的使用需求,还可以一并获取各服务对象的软件开发工具包的使用环境需求,以便于基于该使用环境需求对服务对象进行有针对性的软件开发工具包,避免因运行环境配置不当影响服务对象对于软件开发工具包的使用。

[0034] 步骤202,为与使用需求对应的软件开发工具包添加服务对象的对象标识和版本标识,得到目标软件开发工具包。

[0035] 在本实施例中,在基于上述步骤201中确定各服务对象对软件开发工具包的使用

需求后,分别对应各使用需求确定对应的软件开发工具包,并将与该使用需求所对应的服务对象的对象标识和该软件开发工具包的版本标识添加至该软件开发工具包,得到目标软件开发工具包,即该目标软件开发工具包中添加有使用需求与该目标软件开发工具包的功能相对应的服务对象的对象标识和该目标软件开发工具包的版本标识。

[0036] 需要指出的是,添加对象标识和版本标识之前的软件开发工具包可以由上述执行主体直接从本地的存储设备获取,也可以从非本地的存储设备(例如图1所示的终端设备101、102、103)中获取。本地的存储设备可以是设置在上述执行主体内的一个数据存储模块,例如服务器硬盘,在此种情况下,该软件开发工具包可以在本地快速读取到;非本地的存储设备还可以为其它任何被设置用于存储数据的电子设备,例如一些用户终端等,在此种情况下,上述执行主体可以通过向该电子设备发送获取命令来获取所需的软件开发工具包。

[0037] 在实践中,该版本标识优选的体现为多数位的版本号,以便于基于版本中所包括的各数位体现软件开发工具包更新过程中产生的变化大小。

[0038] 步骤203,基于不同该版本标识之间的关联关系,将对应相同对象标识的各版本的目标软件开发工具包组织为相应服务对象的软件开发工具包管理链。

[0039] 在本实施例中,在基于上述步骤202中对应各使用需求完成对象标识和版本标识的添加,得到多个目标软件开发工具包后,将对应相同对象标识(即属于同一服务对象的使用需求、目标软件开发工具包)进行收集,以一具体的服务对象为例,基于该服务对象(对象标识)的各目标软件开发工具包的版本标识之间的关联关系,将该服务对象的各版本的目标软件开发工具包组织为该服务对象的软件开发工具包管理链,其余不同的服务对象均可参照该过程完成各服务对象所对应的软件开发工具包管理链的组织,此处不再赘述。

[0040] 本公开实施例提供的软件开发工具包管理方法,可利用软件开发工具包管理链实现软件开发工具包的管理,以便于利用同一管理体系为不同服务对象提供软件开发工具包配置服务,降低管理和开发成本。

[0041] 在本实施例的一些可选的实现方式中,该软件开发工具包管理方法,还可以包括:响应于该服务对象之前已组织有历史软件开发工具包管理链,将为该服务对象新创建的软件开发工具包管理链与该历史软件开发工具包管理链进行合并,以统一管理。

[0042] 具体的,在对该服务对象完成了当前的软件开发工具包管理链的组织操作后,若同时发现该服务对象之前已组织有历史软件开发工具包管理链,则将本次组织得到的软件开发工具包管理链与该历史软件开发工具包管理链合并,形成包括本次组织得到的软件开发工具包管理链与历史软件开发工具包管理链中所有内容的完整的软件开发工具包管理链,以实现对该服务对象的软件开发工具包管理链的统一管理。

[0043] 在本实施例的一些可选的实现方式中,该软件开发工具包管理方法,还可以包括:响应于第一服务对象的首个目标软件开发工具包与第二服务对象的目标历史软件开发工具包的功能相同,在与第二服务对象对应的软件开发工具包管理链上的目标历史软件开发工具包处,以创建分支链的方式创建第一服务对象的软件开发工具包管理链。

[0044] 具体的,在根据第一服务对象的首个使用需求确定对应的首个目标软件开发工具包在功能上实际第二服务对象的某个目标历史软件开发工具包(即之前已经根据第二服务对象的使用需求开发好的软件开发工具包)相同时,可以在与第二服务对象对应的软件开

发工具包管理链上的目标历史软件开发工具包处,以创建分支链的方式创建第一服务对象的软件开发工具包管理链。即此时第一服务对象的软件开发工具包管理链为第二服务对象的软件开发工具包管理链上的一条分支链(或称子链),以更为完备的体现软件开发工具包管理链中的目标软件开发工具包开发进程,提升软件开发工具包管理链的内容丰富度、使用质量。

[0045] 在本实施例的一些可选的实现方式中,该为与该使用需求对应的软件开发工具包添加该服务对象的对象标识和版本标识,得到目标软件开发工具包的步骤,可以具体包括:根据该使用需求确定需求代码;从预先配置的代码库中获取与该需求代码对应的目标代码文件;打包该目标代码文件得到与该使用需求对应的软件开发工具包,并添加该服务对象的对象标识和版本标识,得到该目标软件开发工具包。

[0046] 具体的,根据该使用需求确定需求代码,从预先配置的代码库中获取与该需求代码对应的目标代码文件,并执行一个替换引用静态资源地址的脚本,统一增加数据网络的内容分发网络域名后,把引用的静态资源打入到生成的区域(Dist)文件夹中,然后添加该服务对象的对象标识和版本标识,最后动态更改打包后的软件开发工具包的文件名和Package.json中的Version版本,记录最新版本号,完成添加该服务对象的对象标识和版本标识操作,以通过代码库对各类软件开发工具包所需的代码进行整理、存储,后续可基于该代码库中的代码文件进行拼接、组装得到目标软件开发工具包,提升软件开发工具包的配置效率及上述软件开发工具包管理方法的可用性。

[0047] 在本实施例的一些可选的实现方式中,该关联关系包括:各版本标识分别对应的版本号之间的大小关系,对应的,软件开发工具包管理链包括:各目标软件开发工具包按版本号从小到大、从左至右的顺序组织出的时序链,以便于通过按版本号从小到大、从左至右组织出的时序链清楚直观的明确各版本的目标软件开发工具包之间的开发时序关联。

[0048] 进一步的,因软件开发工具包的开发、配置,以及服务对象的使用需求是一个持续发展、变化的过程,伴随服务对象的使用需求的变化,实际应用场景中存在着对历史生成的目标软件开发包进行再次开发以满足后续更新的使用需求的需求,因此,为实现对该情况下的软件开发工具包管理链的管理,本公开实施例还提供了另一种软件开发工具包管理方法的流程图,具体请参考图3,其中流程300包括以下步骤:

[0049] 步骤301,分别获取各服务对象对软件开发工具包的使用需求。

[0050] 步骤302,为与使用需求对应的软件开发工具包添加服务对象的对象标识和版本标识,得到目标软件开发工具包。

[0051] 步骤303,基于不同该版本标识之间的关联关系,将对应相同对象标识的各版本的目标软件开发工具包组织为相应服务对象的软件开发工具包管理链。

[0052] 以上步骤301-303与如图2所示的步骤201-203一致,相同部分内容请参见上一实施例的相应部分,此处不再进行赘述。

[0053] 步骤304,接收服务对象传入的再次开发请求。

[0054] 在本实施例中,接收服务对象传入的再次开发请求,该再次开发请求可以是针对该软件开发工具包管理链上任意的目标软件开发工具包的开发请求。

[0055] 应当理解的是,该再次开发请求可针对该服务对象所对应的软件开发工具包管理链上任意的目标软件开发工具包发出,也可以经授权后,针对该服务对象进行授权的其

他服务对象所对应的软件开发工具包管理链上的任意的目标软件开发工具包发出。

[0056] 步骤305,将该再次开发请求指示的目标软件开发工具包确定为再次开发起点的基础版本软件开发工具包。

[0057] 在本实施例中,将基于上述步骤304中所确定的目标软件开发工具包确定为基础版本软件开发工具包,并以该基础版本软件开发工具包作为再次开发起点。

[0058] 步骤306,根据该再次开发请求中的再次开发需求,对该基础版本软件开发工具包进行再次开发,得到附加新版本标识的新目标软件开发工具包。

[0059] 在本实施例中,根据上述步骤304中的再次开发请求中的再次开发需求,对基于步骤305确定的基础版本软件开发工具包进行再次开发后,为其添加新版本标识,进而得到附加新版本标识的新目标软件开发工具包,其中,对于基础版本软件开发工具包进行再次开发的过程以及对开发完成的软件开发工具包中添加新版本标识的过程,均可参见上述图2所示实施例及相关实现方式中的说明,此处不再赘述。

[0060] 此外,在再次开发请求中所请求的基础版本软件开发工具包为其他服务对象的软件开发工具包管理链上的任意的目标软件开发工具包时,还可以在经过再次开发得到的新目标软件开发工具包中添加该服务对象的标记,以区分对基础版本软件开发工具包进行再次开发的服务对象。

[0061] 步骤307,在软件开发工具包管理链中与基础版本软件开发工具包对应的位置,生成分支链或延续链,并将新目标软件开发工具包置入分支链或延续链。

[0062] 在本实施例中,在软件开发工具包管理链中与基础版本软件开发工具包对应的位置,生成分支链或延续链,并将新目标软件开发工具包置入分支链或延续链。其中,分支链的生成代表着再次开发请求所针对的基础版本软件开发工具包并不是之前管理链上的最新版本(例如管理链上的最新版本为5.0版本,但再次开发请求所针对的基础版本为2.0版本),因此就区别原有链,需要在基础版本软件开发工具包的位置创建分支链,从而将新目标软件开发工具包置入合适的位置(例如将新目标软件开发工具包的版本号命名为2.1,即此时版本号为2.0、2.1的两个软件开发工具包形成了一条分支链)。对应的,延续链的生成则代表再次开发请求所针对的基础版本软件开发工具包是之前管理链上的最新版本,因此只需要对原有的管理链在末端进行延续即可。

[0063] 在上述任一实施例的基础上,该软件开发工具包管理方法,还可以包括:分别确定不同的服务对象的数据存储路径;将与各服务对象对应的软件开发工具包管理链,通过相应的数据存储路径存储至相应的数据存储空间;其中,数据存储路径包括:公有云路径和私有云路径,数据存储空间包括:公有云空间和私有云空间。

[0064] 具体的,在得到与服务对象的使用需求对应的目标软件开发工具包后,还可以分别确定不同的、各服务对象的数据存储路径,该数据存储路径包括以下中的至少一种:公有云路径、私有云路径,并将与各服务对象对应的软件开发工具包管理链,通过相应的数据存储路径存储至相应的数据存储空间,以便于各服务对象在各自对应的数据存储空间获取存入的各目标软件开发工具包,并实现对公有云路径、私有云路径下的服务对象的目标软件开发工具包的统一管理。

[0065] 为加深理解,本公开还结合一个具体应用场景,给出了一种具体的实现方案,其中,在该应用场景下,利用配置服务器位服务对象A和服务对象B进行目标软件开发工具包

的配置,以及软件开发工具包管理链的组织,具体如下:

[0066] 首先,配置服务器分别获取服务对象A、B各自对软件开发工具包的使用需求。

[0067] 然后,在配置服务器中为对应服务对象A的使用需求的软件开发工具包添加服务对象的对象标识和版本标识后,得到目标软件开发工具包“目标软件开发工具包A-1.1”、“目标软件开发工具包A-1.2”、“目标软件开发工具包A-1.3”、“目标软件开发工具包A-2.1”、“目标软件开发工具包A-2.2”、“目标软件开发工具包A-2.3”,其中以目标软件开发工具包“目标软件开发工具包A-1.1”为例,其中“A-1.1”中的“A”为服务对象A的对象标识,“1.1”为该目标软件开发工具包的版本标识,同样的配置服务器中为对应服务对象B的使用需求的软件开发工具包添加服务对象的对象标识和版本标识后,得到目标软件开发工具包“目标软件开发工具包B-2.1”、“目标软件开发工具包B-2.2”、“目标软件开发工具包B-2.3”和“目标软件开发工具包B-2.4”。

[0068] 接下来,基于不同版本标识之间的关关系,将对应服务对象A的各版本的目标软件开发工具包组织为该服务对象A的软件开发工具包管理链A,以及将对应服务对象B的各版本的目标软件开发工具包组织为该服务对象B的软件开发工具包管理链B,并在服务对象B的软件开发工具包管理链中首位的目标软件开发工具包(目标软件开发工具包B-2.1)与服务对象A的历史软件开发工具包管理链(软件开发工具包管理链A)中的历史目标软件开发工具包(目标软件开发工具包A-2.1)的功能相同时进行响应,在服务对象A的历史软件开发工具包管理链中对应目标软件开发工具包A-2.1的位置以直链方式链入服务对象B的软件开发工具包管理链B。

[0069] 最后,分别确定服务对象A的数据存储路径(公有云路径)以及服务对象B的数据存储路径(私有云路径)后,将“目标软件开发工具包A-1.1”、“目标软件开发工具包A-1.2”、“目标软件开发工具包A-1.3”、“目标软件开发工具包A-2.1”、“目标软件开发工具包A-2.2”、“目标软件开发工具包A-2.3”存入该公有云路径,以便于利用公有云与该服务对象A进行交互、配置(目标)软件开发工具包,将“目标软件开发工具包B-2.1”、“目标软件开发工具包B-2.2”、“目标软件开发工具包B-2.3”和“目标软件开发工具包B-2.4”先通过公有云路径存入公有云,然后再通过该公有云与私有云之间的传输通路存入该私有云,以便于利用私有云与该服务对象B进行交互、配置(目标)软件开发工具包。

[0070] 另外,在其它场景下,也可以将“目标软件开发工具包B-2.1”、“目标软件开发工具包B-2.2”、“目标软件开发工具包B-2.3”和“目标软件开发工具包B-2.4”直接通过私有云路径存入私有云,而不再通过公有云作为中间传输桥梁,具体选用哪种可根据实际情况而定。

[0071] 进一步参考图5,作为对上述各图所示方法的实现,本公开提供了一种软件开发工具包管理装置的一个实施例,该装置实施例与图2所示的方法实施例相对应,该装置具体可以应用于各种电子设备中。

[0072] 如图5所示,本实施例的软件开发工具包管理装置500可以包括:使用需求获取单元501、工具包生成单元502、管理链组织单元503。其中,使用需求获取单元501,被配置成分别获取各服务对象对软件开发工具包的使用需求;工具包生成单元502,被配置成为与该使用需求对应的软件开发工具包添加该服务对象的对象标识和版本标识,得到目标软件开发工具包;管理链组织单元503,被配置成基于不同该版本标识之间的关联关系,将对应相同对象标识的各版本的目标软件开发工具包组织为相应服务对象的软件开发工具包管理链。

[0073] 在本实施例中,软件开发工具包管理装置500中:使用需求获取单元501、工具包生成单元502、管理链组织单元503的具体处理及其所带来的技术效果可分别参考图2对应实施例中的步骤201-203的相关说明,在此不再赘述。

[0074] 在本实施例的一些可选的实现方式中,该软件开发工具包管理装置500,还包括:

[0075] 分支链创建单元,被配置成响应于第一服务对象的首个目标软件开发工具包与第二服务对象的目标历史软件开发工具包的功能相同,在与第二服务对象对应的软件开发工具包管理链上的目标历史软件开发工具包处,以创建分支链的方式创建第一服务对象的软件开发工具包管理链。

[0076] 在本实施例的一些可选的实现方式中,该软件开发工具包管理装置500,还包括:

[0077] 再次开发请求接收单元,被配置成接收服务对象传入的再次开发请求;

[0078] 基础工具包确定单元,被配置成将再次开发请求指示的目标软件开发工具包确定为再次开发起点的基础版本软件开发工具包;

[0079] 新工具包生成单元,被配置成根据再次开发请求中的再次开发需求,对基础版本软件开发工具包进行再次开发,得到附加新版本标识的新目标软件开发工具包,新版本标识基于基础版本软件开发工具包的版本标识确定;

[0080] 管理链更新单元,被配置成在软件开发工具包管理链中与基础版本软件开发工具包对应的位置,生成分支链或延续链,并将新目标软件开发工具包置入分支链或延续链。

[0081] 在本实施例的一些可选的实现方式中,工具包生成单元502可以被进一步配置成:

[0082] 根据使用需求确定需求代码;

[0083] 从预先配置的代码库中获取与需求代码对应的目标代码文件;

[0084] 打包目标代码文件得到与使用需求对应的软件开发工具包,并添加服务对象的对象标识和版本标识,得到目标软件开发工具包。

[0085] 在本实施例的一些可选的实现方式中,关联关系包括:各版本标识分别对应的版本号之间的大小关系,对应的,软件开发工具包管理链包括:各目标软件开发工具包按版本号从小到大、从左至右的顺序组织出的时序链。

[0086] 在本实施例的一些可选的实现方式中,该软件开发工具包管理装置500,还可以包括:

[0087] 存储路径获取单元,被配置成分别确定不同的服务对象的数据存储路径;

[0088] 管理链存储单元,被配置成将与各服务对象对应的软件开发工具包管理链,通过相应的数据存储路径存储至相应的数据存储空间;其中,数据存储路径包括:公有云路径和私有云路径,数据存储空间包括:公有云空间和私有云空间。

[0089] 本实施例作为对应于上述方法实施例的装置实施例存在,本实施例提供的软件开发工具包管理装置,可利用软件开发工具包管理链实现软件开发工具包的管理,以便于利用同一管理体系为不同服务对象提供软件开发工具包配置服务,降低管理和开发成本。

[0090] 根据本公开的实施例,本公开还提供了一种电子设备、一种可读存储介质和一种计算机程序产品。

[0091] 图6示出了可以用来实施本公开的实施例的示例电子设备600的示意性框图。电子设备旨在表示各种形式的数字计算机,诸如,膝上型计算机、台式计算机、工作台、个人数字助理、服务器、刀片式服务器、大型计算机、和其它适合的计算机。电子设备还可以表示各种

形式的移动装置,诸如,个人数字处理、蜂窝电话、智能电话、可穿戴设备和其它类似的计算装置。本文所示的部件、它们的连接和关系、以及它们的功能仅仅作为示例,并且不意在限制本文中描述的和/或者要求的本公开的实现。

[0092] 如图6所示,设备600包括计算单元601,其可以根据存储在只读存储器 (ROM) 602中的计算机程序或者从存储单元608加载到随机访问存储器 (RAM) 603中的计算机程序,来执行各种适当的动作和处理。在RAM 603中,还可存储设备600操作所需的各种程序和数据。计算单元601、ROM 602以及RAM 603通过总线604彼此相连。输入/输出 (I/O) 接口605也连接至总线604。

[0093] 设备600中的多个部件连接至I/O接口605,包括:输入单元606,例如键盘、鼠标等;输出单元607,例如各种类型的显示器、扬声器等;存储单元608,例如磁盘、光盘等;以及通信单元609,例如网卡、调制解调器、无线通信收发机等。通信单元609允许设备600通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

[0094] 计算单元601可以是各种具有处理和计算能力的通用和/或专用处理组件。计算单元601的一些示例包括但不限于中央处理单元 (CPU)、图形处理单元 (GPU)、各种专用的人工智能 (AI) 计算芯片、各种运行机器学习模型算法的计算单元、数字信号处理器 (DSP)、以及任何适当的处理器、控制器、微控制器等。计算单元601执行上文所描述的各个方法和处理,例如软件开发工具包管理方法。例如,在一些实施例中,软件开发工具包管理方法可被实现为计算机软件程序,其被有形地包含于机器可读介质,例如存储单元608。在一些实施例中,计算机程序的部分或者全部可以经由ROM 602和/或通信单元609而被载入和/或安装到设备600上。当计算机程序加载到RAM 603并由计算单元601执行时,可以执行上文描述的软件包管理方法的一个或多个步骤。备选地,在其他实施例中,计算单元601可以通过其他任何适当的方式(例如,借助于固件)而被配置为执行软件开发工具包管理方法。

[0095] 本文中以上描述的系统和技术和各种实施方式可以在数字电子电路系统、集成电路系统、场可编程门阵列 (FPGA)、专用集成电路 (ASIC)、专用标准产品 (ASSP)、芯片上系统的系统 (SOC)、负载可编程逻辑设备 (CPLD)、计算机硬件、固件、软件、和/或它们的组合中实现。这些各种实施方式可以包括:实施在一个或者多个计算机程序中,该一个或者多个计算机程序可在包括至少一个可编程处理器的可编程系统上执行和/或解释,该可编程处理器可以是专用或者通用可编程处理器,可以从存储系统、至少一个输入装置、和至少一个输出装置接收数据和指令,并且将数据和指令传输至该存储系统、该至少一个输入装置、和该至少一个输出装置。

[0096] 用于实施本公开的方法的程序代码可以采用一个或多个编程语言的任何组合来编写。这些程序代码可以提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器或控制器,使得程序代码当由处理器或控制器执行时使流程图和/或框图中所规定的功能/操作被实施。程序代码可以完全在机器上执行、部分地在机器上执行,作为独立软件包部分地在机器上执行且部分地在远程机器上执行或完全在远程机器或服务器上执行。

[0097] 在本公开的上下文中,机器可读介质可以是有形的介质,其可以包含或存储以供指令执行系统、装置或设备使用或与指令执行系统、装置或设备结合地使用的程序。机器可读介质可以是机器可读信号介质或机器可读储存介质。机器可读介质可以包括但不限于电子的、磁性的、光学的、电磁的、红外的、或半导体系统、装置或设备,或者上述内容的任何合

适组合。机器可读存储介质的更具体示例会包括基于一个或多个线的电气连接、便携式计算机盘、硬盘、随机存取存储器 (RAM)、只读存储器 (ROM)、可擦除可编程只读存储器 (EPROM 或快闪存储器)、光纤、便捷式紧凑盘只读存储器 (CD-ROM)、光学储存设备、磁储存设备、或上述内容的任何合适组合。

[0098] 为了提供与用户的交互,可以在计算机上实施此处描述的系统和技术,该计算机具有:用于向用户显示信息的显示装置(例如,CRT(阴极射线管)或者LCD(液晶显示器)监视器);以及键盘和指向装置(例如,鼠标或者轨迹球),用户可以通过该键盘和该指向装置来将输入提供给计算机。其它种类的装置还可以用于提供与用户的交互;例如,提供给用户的反馈可以是任何形式的传感反馈(例如,视觉反馈、听觉反馈、或者触觉反馈);并且可以用任何形式(包括声输入、语音输入或者、触觉输入)来接收来自用户的输入。

[0099] 可以将此处描述的系统和技术实施在包括后台部件的计算系统(例如,作为数据服务器)、或者包括中间件部件的计算系统(例如,应用服务器)、或者包括前端部件的计算系统(例如,具有图形用户界面或者网络浏览器的用户计算机,用户可以通过该图形用户界面或者该网络浏览器来与此处描述的系统和技术实施方式交互)、或者包括这种后台部件、中间件部件、或者前端部件的任何组合的计算系统中。可以通过任何形式或者介质的数字数据通信(例如,通信网络)来将系统的部件相互连接。通信网络的示例包括:局域网(LAN)、广域网(WAN)和互联网。

[0100] 计算机系统可以包括客户端和服务端。客户端和服务端一般远离彼此并且通常通过通信网络进行交互。通过在相应的计算机上运行并且彼此具有客户端-服务器关系的计算机程序来产生客户端和服务端的关系。服务器可以是云服务器,又称为云计算服务器或云主机,是云计算服务体系中的一项主机产品,以解决传统物理主机与虚拟专用服务器(VPS,Virtual Private Server)服务中存在的管理难度大,业务扩展性弱的缺陷。服务器也可以分为分布式系统的服务器,或者是结合了区块链的服务器。

[0101] 根据本公开实施例的技术方案,可利用软件开发工具包管理链实现软件开发工具包的管理,以便于利用同一管理体系为不同服务对象提供软件开发工具包配置服务,降低管理和开发成本。

[0102] 应该理解,可以使用上面所示的各种形式的流程,重新排序、增加或删除步骤。例如,本公开中记载的各步骤可以并行地执行也可以顺序地执行也可以不同的次序执行,只要能够实现本公开提供的技术方案所期望的结果,本文在此不进行限制。

[0103] 上述具体实施方式,并不构成对本公开保护范围的限制。本领域技术人员应该明白的是,根据设计要求和因素,可以进行各种修改、组合、子组合和替代。任何在本公开的精神和原则之内所作的修改、等同替换和改进等,均应包含在本公开保护范围之内。

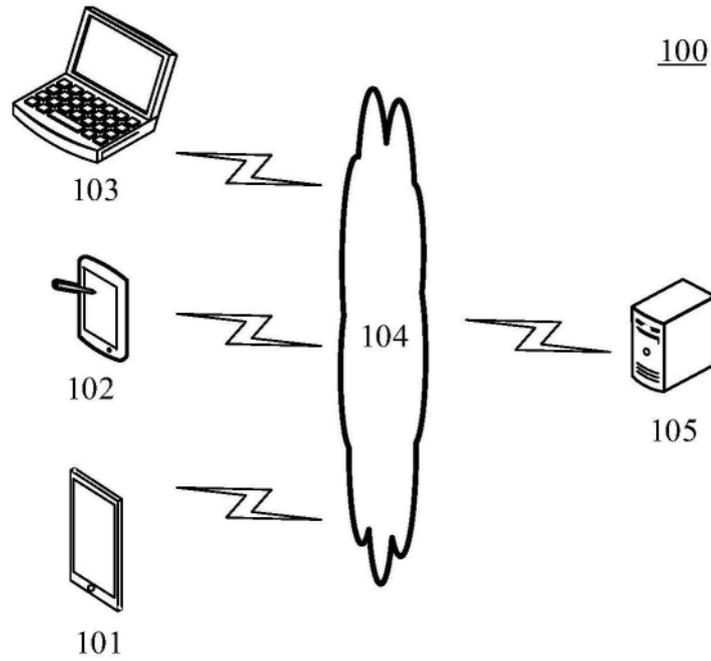


图1

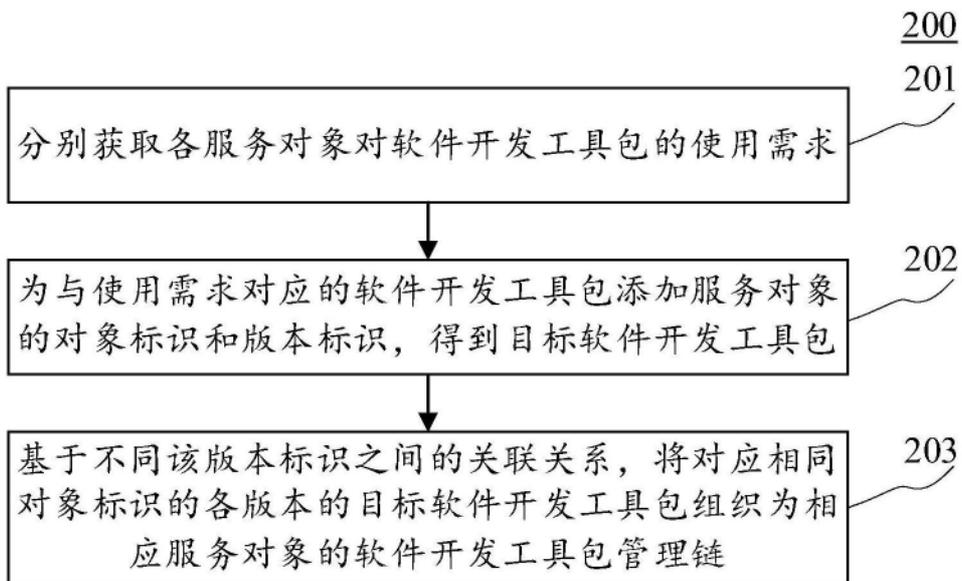


图2

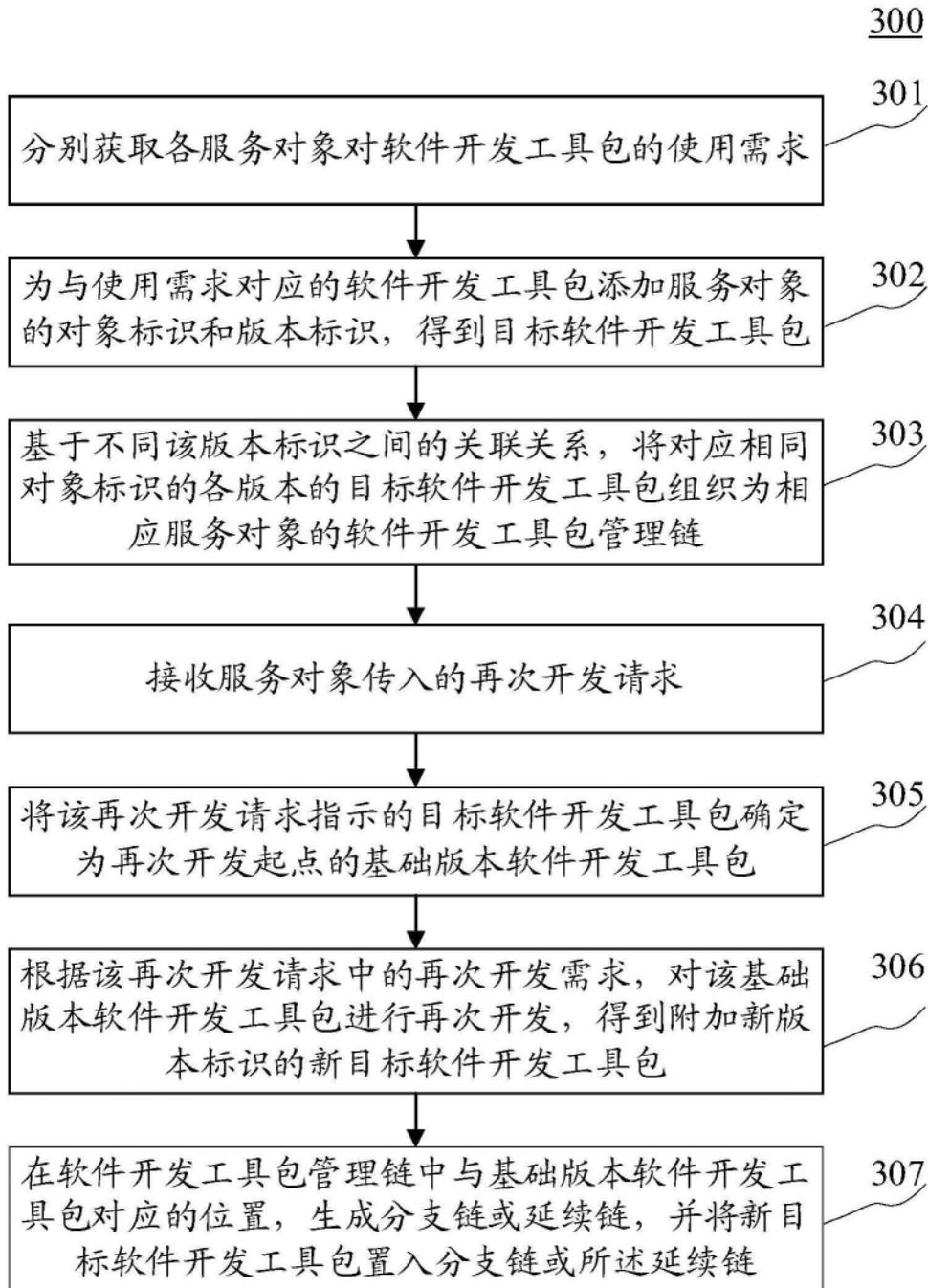


图3

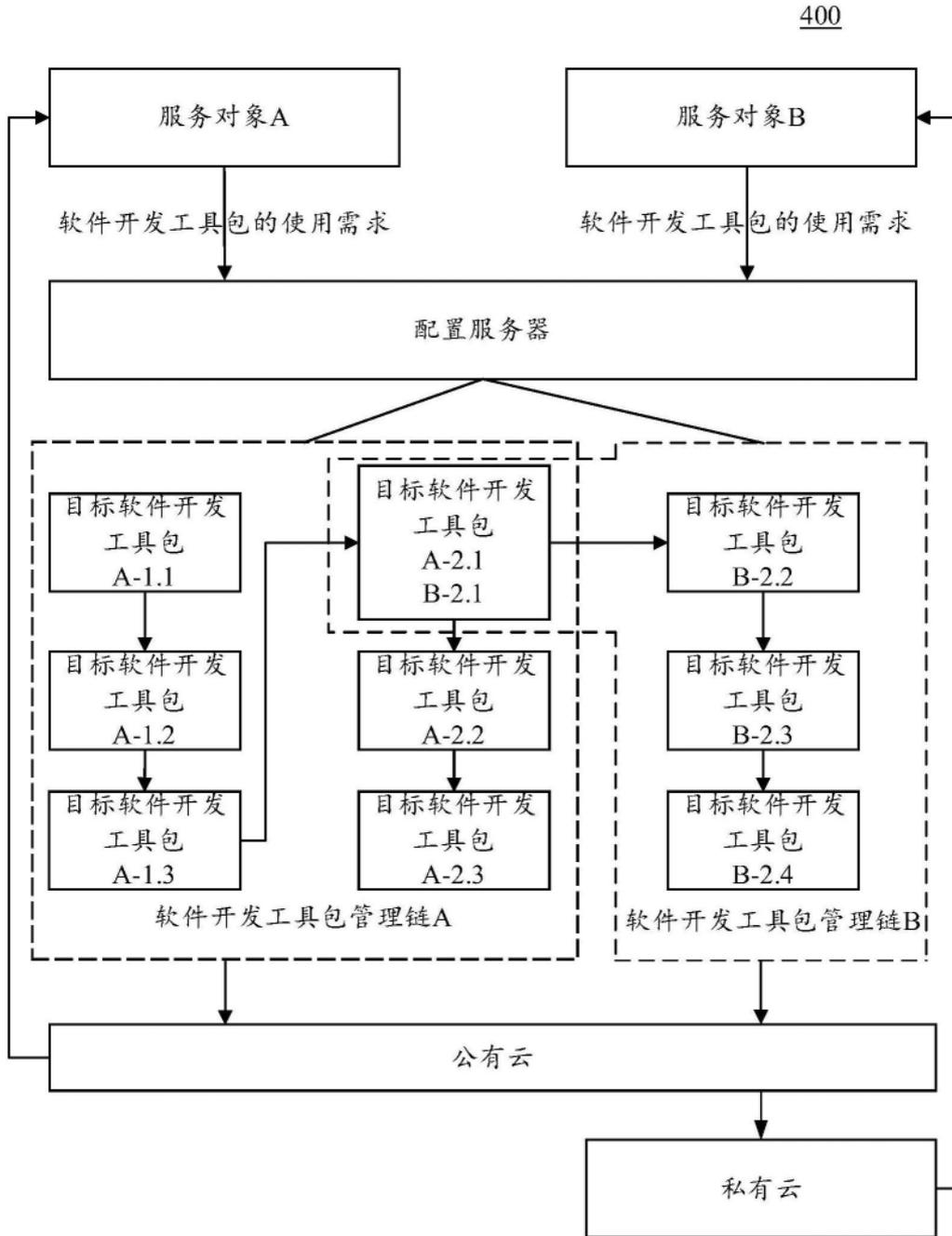


图4

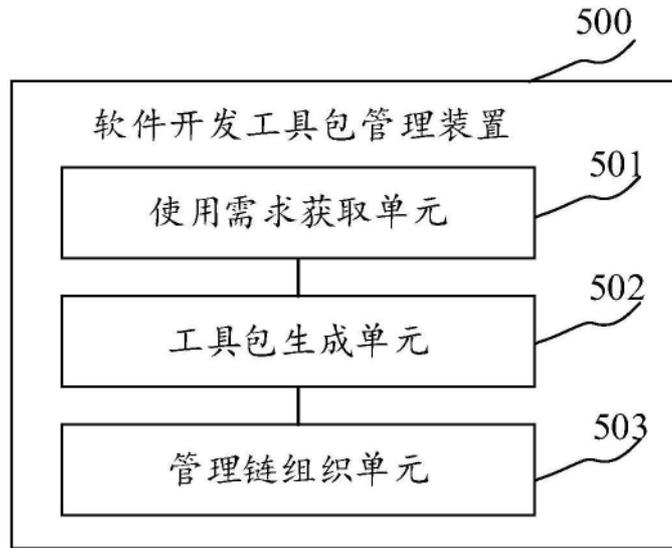


图5

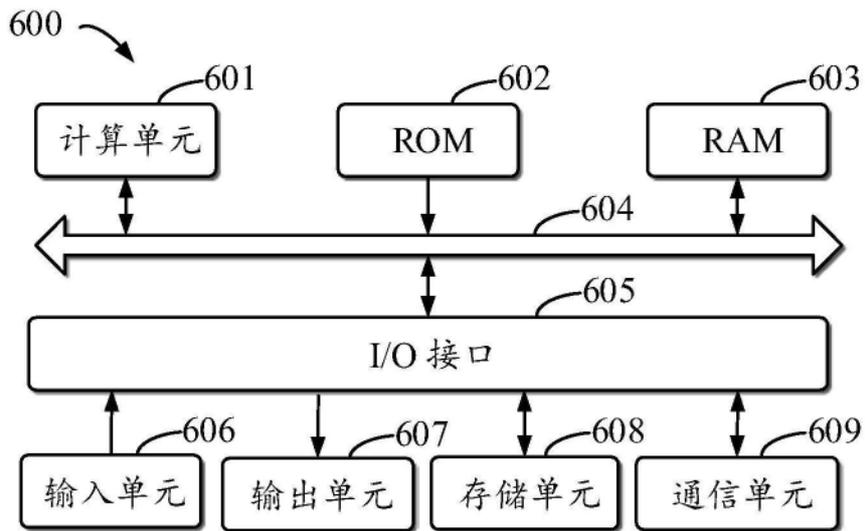


图6