(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 112416612 B
(45)授权公告日 2023.03.28

(21)申请号 202010814044.3
(22)申请日 2020.08.13
(65)同一申请的已公布的文献号
申请公布号 CN 112416612 A
(43)申请公布日 2021.02.26
(73)专利权人 上海哔哩哔哩科技有限公司
地址 200433 上海市杨浦区政立路485号国正中心3号楼
(72)发明人 耿万鹏
(74)专利代理机构 北京英特普罗知识产权代理有限公司 11015
专利代理师 程超
(51)Int.Cl.
G06F 9/54(2006.01)
G06F 16/955(2019.01)

(56)对比文件
CN 110377438 A,2019.10.25
CN 110308900 A,2019.10.08
CN 109445923 A,2019.03.08
CN 107203535 A,2017.09.26
CN 111045833 A,2020.04.21
US 2014214923 A1,2014.07.31

审查员 谢璐

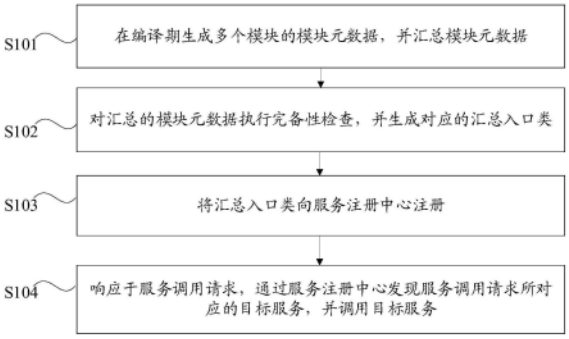
权利要求书2页 说明书8页 附图4页

(54)发明名称

服务调用方法、装置、计算机设备和可读存储介质

(57)摘要

本申请提供了一种基于路由网关的服务调用方法、装置、计算机设备和可读存储介质。该方法包括：在编译期生成多个模块的模块元数据，并汇总所述模块元数据；对汇总的模块元数据执行完备性检查，并生成对应的汇总入口类；将所述汇总入口类向服务注册中心注册；以及响应于服务调用请求，通过所述服务注册中心发现所述服务调用请求所对应的目标服务，并调用所述目标服务。通过本申请，可以确保无环及对应的依赖存在，大大提高团队协作开发的效率。



CN 112416612 B

1.一种基于路由网关的服务调用方法,其特征在于,包括:

在编译期生成多个模块的模块元数据,并汇总所述模块元数据,其中,所述模块为待调用服务的模块,所述模块元数据声明在包、类、字段、方法、局部变量或方法参数的前面,用来对这些元素进行说明和注释;

对汇总的模块元数据执行完备性检查,若所述完备性检查被成功执行,生成对应的汇总入口类,其中,所述完备性检查包括:依赖关系检查和循环依赖检查,其中,所述依赖关系检查用于检查依赖关系所涉及的需要暴露的服务是否完整,所述循环依赖检查用于检查各模块的API之间是否相互依赖;

将所述汇总入口类向服务注册中心注册;以及

响应于服务调用请求,通过所述服务注册中心发现所述服务调用请求所对应的目标服务,并调用所述目标服务。

2.根据权利要求1所述的服务调用方法,其特征在于,对汇总的模块元数据执行完备性检查的步骤包括:

生成汇总的模块元数据对应的配置文件;

基于所述配置文件对所述汇总的模块元数据执行完备性检查。

3.根据权利要求1至2中任一项所述的服务调用方法,其特征在于,响应于服务调用请求,通过所述服务注册中心发现所述服务调用请求所对应的目标服务,并调用所述目标服务的步骤包括:

基于预设的路径规则对目标资源标识符执行匹配;

跳转至匹配所得的目标页面,并加载第一模块;

响应于所述第一模块的服务调用请求,通过所述服务注册中心发现所述服务调用请求所对应的目标服务,并调用所述目标服务。

4.根据权利要求3所述的服务调用方法,其特征在于,基于预设的路径规则对目标资源标识符执行匹配的步骤包括:

确定目标资源标识符对应的路由类型;

基于所述路由类型及预设的路径规则对目标资源标识符执行匹配。

5.根据权利要求1至2中任一项所述的服务调用方法,其特征在于,响应于服务调用请求,通过所述服务注册中心发现所述服务调用请求所对应的目标服务,并调用所述目标服务的步骤包括:

响应于第一模块的服务调用请求,通过所述服务注册中心发现所述服务调用请求所对应的目标服务,并调用所述目标服务;

其中所述目标服务由第二模块提供,所述第一模块与所述第二模块分别为Gradle模块并相互依赖;所述第一模块包含第一实现组件和第一接口组件,所述第二模块包含第二实现组件和第二接口组件,所述目标服务基于所述第二接口组件而被调用。

6.一种基于路由网关的服务调用装置,其特征在于,包括:

元数据生成模块,用于在编译期生成多个模块的模块元数据,并汇总所述模块元数据,其中,所述模块为待调用服务的模块,所述模块元数据声明在包、类、字段、方法、局部变量或方法参数的前面,用来对这些元素进行说明和注释;

完备性检查模块,用于对汇总的模块元数据执行完备性检查,若所述完备性检查被成

功执行,生成对应的汇总入口类,其中,所述完备性检查包括:依赖关系检查和循环依赖检查,其中,所述依赖关系检查用于检查依赖关系所涉及的需要暴露的服务是否完整,所述循环依赖检查用于检查各模块的API之间是否相互依赖;

注册模块,用于将所述汇总入口类向服务注册中心注册;以及

服务调用模块,用于响应于服务调用请求,通过所述服务注册中心发现所述服务调用请求所对应的目标服务,并调用所述目标服务。

7.一种计算机设备,包括存储器、处理器以及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1至5任一项所述方法的步骤。

8.一种计算机可读存储介质,其上存储有计算机程序,其特征在于:所述计算机程序被处理器执行时实现权利要求1至5任一项所述方法的步骤。

服务调用方法、装置、计算机设备和可读存储介质

技术领域

[0001] 本申请涉及应用发布技术领域，尤其涉及基于路由网关的服务调用方法、装置、计算机设备和可读存储介质。

背景技术

[0002] 安卓(Android)是一种基于Linux内核(不包含GNU组件)的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由美国谷歌公司(Google)和开放手机联盟领导及开发。

[0003] 程序的执行过程就是方法的调用过程，有方法调用，必然会促使对象跟对象之间产生依赖，除非一个对象不参与程序的运行，这样的对象就像一座孤岛，与其它对象没有任何交互，但是这样的对象也就没有任何存在价值。因此，在程序代码中，任何一个对象必然会与其它一个甚至更多个对象产生依赖关系。“方法调用”是最常见产生依赖的原因，一个对象与其它对象必然会通信(除非所有的代码逻辑全部写在了这个对象内部)，通信通常情况下就意味着有方法的调用，亦即意味着这两个对象之间存在依赖关系(至少要有其它对象的引用才能调用方法)，另外常见的一种产生依赖的原因是继承。

[0004] 其中，若干个对象循环嵌套引用会产生循环依赖，而循环依赖的对象在运行时一直循环调用，直至内存溢出报错。为了衡量对象之间依赖程度的高低，人们引进了“耦合”这一概念，耦合度越高，说明对象之间的依赖程度越高。在多团队共同开发一个Android应用的情形下，需要一种通用技术降低团队之间的代码耦合度，最常用技术包括总线、路由、ServiceLoader(一种依赖倒置的开发方式)等，这几种方案最核心的思想，都是通过一个中间SDK来注册并分发消息、对象或者启动页面。

[0005] 但是，尝试用针对路径或者约定的固定格式的key来分发所有内容，缺少强类型的约束，此外不能很好的处理URI(资源标识符)，比如需要对HTTP URI进行映射或者转义；类似于ServiceLoader的依赖倒置的处理方式缺少生命周期以及服务之间的依赖检查，且十分繁琐。

发明内容

[0006] 本申请的目的是提供一种基于路由网关的服务调用方法、装置、计算机设备和可读存储介质，用于解决现有技术中的上述技术问题。

[0007] 一方面，为实现上述目的，本申请提供了一种服务调用方法。

[0008] 该服务调用方法包括：在编译期生成多个模块的模块元数据，并汇总所述模块数据；对汇总的模块元数据执行完备性检查，并生成对应的汇总入口类；将所述汇总入口类向服务注册中心注册；以及响应于服务调用请求，通过所述服务注册中心发现所述服务调用请求所对应的目标服务，并调用所述目标服务。

[0009] 进一步地，对汇总的模块元数据执行完备性检查，并生成对应的汇总入口类的步骤包括：对汇总的模块元数据执行完备性检查；若所述完备性检查被成功执行，生成对应的

汇总入口类。

[0010] 进一步地，对汇总的模块元数据执行完备性检查的步骤包括：生成汇总的模块元数据对应的配置文件；基于所述配置文件对所述汇总的模块元数据执行完备性检查。

[0011] 进一步地，所述完备性检查包括：依赖关系检查；循环依赖检查。

[0012] 进一步地，响应于服务调用请求，通过所述服务注册中心发现所述服务调用请求所对应的目标服务，并调用所述目标服务的步骤包括：基于预设的路径规则对目标资源标识符执行匹配；跳转至匹配所得的目标页面，并加载第一模块；以及响应于所述第一模块的服务调用请求，通过所述服务注册中心发现所述服务调用请求所对应的目标服务，并调用所述目标服务。

[0013] 进一步地，基于预设的路径规则对目标资源标识符执行匹配的步骤包括：确定目标资源标识符对应的路由类型；基于所述路由类型及预设的路径规则对目标资源标识符执行匹配。

[0014] 进一步地，响应于服务调用请求，通过所述服务注册中心发现所述服务调用请求所对应的目标服务，并调用所述目标服务的步骤包括：响应于第一模块的服务调用请求，通过所述服务注册中心发现所述服务调用请求所对应的目标服务，并调用所述目标服务；其中所述目标服务由第二模块提供，所述第一模块与所述第二模块分别为Gradle模块并相互依赖；所述第一模块包含第一实现组件和第一接口组件，所述第二模块包含第二实现组件和第二接口组件，所述目标服务基于所述第二接口组件而被调用。

[0015] 另一方面，为实现上述目的，本申请提供了一种基于路由网关的服务调用装置。

[0016] 该基于路由网关的服务调用装置包括：元数据生成模块，用于在编译期生成多个模块的模块元数据，并汇总所述模块元数据；完备性检查模块，用于对汇总的模块元数据执行完备性检查，并生成对应的汇总入口类；注册模块，用于将所述汇总入口类向服务注册中心注册；以及服务调用模块，用于响应于服务调用请求，通过所述服务注册中心发现所述服务调用请求所对应的目标服务，并调用所述目标服务。

[0017] 为实现上述目的，本申请还提供一种计算机设备，包括存储器、处理器以及存储在存储器上并可在处理器上运行的计算机程序，该处理器执行计算机程序时实现上述方法的步骤。

[0018] 为实现上述目的，本申请还提供计算机可读存储介质，其上存储有计算机程序，该计算机程序被处理器执行时实现上述方法的步骤。

[0019] 本申请提供的基于路由网关的服务调用方法、装置、计算机设备和可读存储介质，在编译期生成多个待调用服务的服务元数据，并汇总服务元数据；对汇总的服务元数据执行完备性检查，并生成待调用服务对应的汇总入口类；将汇总入口类向服务注册中心注册；响应于服务调用请求，通过服务注册中心发现服务调用请求所对应的目标服务，并调用目标服务；从而在编译期保障待调用服务模块的完备性，确保无环及对应的依赖存在，大大提高团队协作开发的效率。

附图说明

[0020] 通过阅读下文优选实施方式的详细描述，各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的，而并不认为是对本申请

的限制。而且在整个附图中，用相同的参考符号表示相同的部件。在附图中：

[0021] 图1为本申请实施例一提供的服务调用方法的流程图；

[0022] 图2a和图2b分别是现有技术中处理循环依赖的解决方案的示意；

[0023] 图3为本申请一实施例中URI的路径匹配示意；

[0024] 图4为本申请实施例二提供的服务调用装置的框图；

[0025] 图5为本申请实施例三提供的计算机设备的硬件结构图。

具体实施方式

[0026] 为了使本申请的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本申请进行进一步详细说明。应当理解，此处所描述的具体实施例仅用以解释本申请，并不用于限定本申请。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。

[0027] 就现有技术而言，降低团队之间的代码耦合度最常用的技术包括总线、路由、ServiceLoader(一种依赖倒置的开发方式)等，这几种方案最核心的思想，都是通过一个中间SDK来注册并分发消息、对象或者启动页面。但是，尝试用针对路径或者约定的固定格式的key来分发所有内容本质上是一种动态调用，基于约定来达成，缺少强类型的约束，此外不能很好的处理URI，比如需要对HTTP URI进行映射或者转义；类似于ServiceLoader的依赖倒置的处理方式虽然很好利用接口来保证了调用的类型，但是每个被倒置的接口均是无状态游离的节点，缺少生命周期以及服务之间的依赖检查。此外(参考图2a)，基于ServiceLoader，各模块(例如模块A、模块B)的接口定义，要么需要下沉到一个公共模块，要么需要针对每个需要暴露服务的模块另外单独新建一个接口(API)模块(参考图2b)，从而达到屏蔽实现的目的，十分繁琐。

[0028] 基于发明人的上述研究，本申请提出一种基于路由网关的服务调用方法、装置、计算机设备和可读存储介质。基于本申请提供的基于路由网关的服务调用方法，在编译期生成多个模块的模块元数据，并汇总模块元数据；对汇总的模块元数据执行完备性检查，并生成对应的汇总入口类；将汇总入口类向服务注册中心注册；以及响应于服务调用请求，通过服务注册中心发现服务调用请求所对应的目标服务，并调用目标服务；从而在编译期保障待调用服务的模块的完备性，确保无环及对应的依赖存在，大大提高团队协作开发的效率。

[0029] 关于本申请提供的基于路由网关的服务调用方法、装置、计算机设备和可读存储介质的具体实施例，将在下文中详细描述。

[0030] 实施例一

[0031] 本申请实施例提供了一种基于路由网关的服务调用方法，通过在编译期保障待调用服务的模块的完备性，确保无环及对应的依赖存在，大大提高团队协作开发的效率。具体地，图1为本申请实施例一提供的基于路由网关的服务调用方法的流程图，如图1所示，该实施例提供的基于路由网关的服务调用方法包括如下的步骤S101至步骤S104。

[0032] 步骤S101：在编译期生成多个模块的模块元数据，并汇总模块元数据。

[0033] 具体而言，在一个实施例中，通过JavaAPT(Annotation Processing Tool，注解处理器)技术生成相应的元数据，再根据元数据生成相应的代码，即通过APT接口生成相应的Java文件。注解(Annotation)，也叫元数据，一种代码级别的说明，是JDK1.5及以后版本引

入的一个特性，与类、接口、枚举是在同一个层次，它可以声明在包、类、字段、方法、局部变量、方法参数等的前面，用来对这些元素进行说明和注释。

[0034] 步骤S102：对汇总的模块元数据执行完备性检查，并生成对应的汇总入口类。

[0035] 其中，对元数据执行完备性检查的一个目的是确保信息的完整性，以及在构建应用(APP)时汇总进行检查以确保无循环依赖，从而在编译期保障待调用服务的模块的完备性，确保无环及对应的依赖存在，大大提高团队协作开发的效率。

[0036] 步骤S103：将所述汇总入口类向服务注册中心注册。

[0037] 其中，各个模块在对外提供服务前，必须首先到注册中心进行注册；而所有访问通过服务网关进行访问，然后由服务网关路由到对应服务中心进行交互访问。具体而言，在网关层接受各个模块的注册，模块可包含URI(资源)、服务(接口或者类)，在加载模块或者获取服务时执行模块的生命周期或者任务，降低模块之间的耦合。

[0038] 步骤S104：响应于服务调用请求，通过所述服务注册中心发现所述服务调用请求所对应的目标服务，并调用所述目标服务。

[0039] 其中，该服务调用请求由调用目标服务的模块发出。

[0040] 至此，服务调用过程完毕。

[0041] 可选地，在一种实施例中，上述步骤S102包括：

[0042] 子步骤S102a(未示出)：对汇总的模块元数据执行完备性检查；

[0043] 子步骤S102b：若所述完备性检查被成功执行，生成对应的汇总入口类。

[0044] 即，在汇总的模块元数据的完备性检查通过后，再生成对应的汇总入口类，否则报错，以便检查相关内容(包括依赖关系所涉及的需要暴露的服务)是否完整；其中，在完备性检查涉及循环依赖检查时，若报错，操作人员还需检查并消除循环依赖。

[0045] 可选地，在一种实施例中，上述完备性检查是基于新增的配置文件完成的。具体而言，在上述子步骤S102a中，生成汇总的模块元数据对应的配置文件，并基于所述配置文件对所述汇总的模块元数据执行完备性检查。该配置文件包含模块元数据的汇总信息，因此除用于执行前述完备性检查外，在一个实施例中，该配置文件还被用于生成前述汇总入口类，从而无需重复获取多个模块的信息，因此处理效率大大提高。即，在上述子步骤S102b中，若所述完备性检查被成功执行，基于所述配置文件生成对应的汇总入口类。

[0046] 可选地，在一种实施例中，上述完备性检查包括依赖关系检查和循环依赖检查。

[0047] 其中，依赖关系检查主要用于检查相关内容(包括依赖关系所涉及的需要暴露的服务)是否完整，若不完整(例如某个模块声明依赖于另一个模块，但是被依赖的模块却没有暴露相应的服务)则报错；而循环依赖检查则主要用于检查各模块的API之间是否相互依赖，若存在API的相互依赖则报错。通过依赖关系检查和循环依赖检查，可以保障待调用服务模块的完备性，确保无环及对应的依赖存在，以提高团队协作开发的效率。

[0048] 可选地，在一种实施例中，上述步骤S104包括：

[0049] 子步骤S104a(未示出)：基于预设的路径规则对目标资源标识符(URI)执行匹配；

[0050] 子步骤S104b(未示出)：跳转至匹配所得的目标页面，并加载第一模块；

[0051] 子步骤S104c(未示出)：响应于所述第一模块的服务调用请求，通过所述服务注册中心发现所述服务调用请求所对应的目标服务，并调用所述目标服务。

[0052] 其中，本申请提供的服务调用方法提供SDK作为一个微服务的网关层，对外屏蔽内

部微服务的细节，只暴露URI，针对路由跳转内置统一路由协议，用于特定的路由逻辑。其中，该统一的路由协议为基于URI的格式封装的路由协议，分为声明和匹配两个阶段。

[0053] 对于声明阶段：

[0054] 路径规则的声明支持如下3种方式(优先级从高到低)：

[0055] 1.复合，例如：(?<capture_name>path1|path2|pathN)，其中?<capture_name>是可选的，括号是可嵌套的，如(bilibili|http|https)与(bilibili|htt(p|ps))匹配规则是一样的，注意括号中的字符串不能为空，即暂不支持http(|s)这样的格式，段内可以有多个；

[0056] 2.通配，例如{capture_name}或者*，其中*等价{}，即匹配但是不捕获，段内只能有一个；

[0057] 3.前缀，**，只能存在末尾，如https://bilibili.com/**，会匹配https://bilibili.com/a/b/n，但是不能匹配https://bilibili.com，一定会产生捕获，capture_name为空字符串。

[0058] 其中，前两类路由规则不能跨越“/”，且不同路由规则是互斥的，每个段内只能应用一类。同时，允许添加字符串KV对声明为路由的属性用于匹配。

[0059] 对于匹配阶段：

[0060] 传入一个标准的URI，其中一些query作为协议的一部分。

[0061] 例如：

[0062] 1.name以-B开头，保留几个固定的name用于协议的一部分，其他的-B开头的当做特殊参数传给目标，其中一个特殊的是路由类型，如-Btype＝xxx，指明目标URI的类型，如不存在则代码中默认依次尝试全部类型。

[0063] 2.name以-A开头的，去除-A的部分为URI中的属性，用于匹配过程，例如https://www.bilibili.com?-Abrowser＝main中可提取出browser＝main，参与路由匹配。

[0064] 可选地，在一种实施例中，对于匹配过程，上述子步骤S104a包括：

[0065] 确定目标资源标识符对应的路由类型，并基于所述路由类型及预设的路径规则(例如上述路径规则)对目标资源标识符执行匹配。

[0066] 通过引入对路由类型，在路由类型未指定的情况下本实施例可以实现根据统一路由协议进行路由查询，因此该方案接受一切URI的注册，而无需对进入的URI进行转移或者做特别处理，相较于现有的处理方式而言大大简化了处理过程。

[0067] 为直观起见，参考图3，对匹配阶段的工作举例如下：

[0068] 当一条URI进入匹配过程、但不指定路由类型时，会依次在NATIVE/H5中找到第一条匹配成功的目标路由，如待匹配的URI为https://bilibili.com/video/123则会在NATIVE中匹配成功，而如果待匹配URI为https://bilibili.com/music时，由于NATIVE中匹配失败，而在H5中匹配成功。

[0069] 如上所述的引入了路由类型的统一路由协议，可灵活地做到降级、互备，并且不对待匹配URI做出假定(如只限定scheme为bilibili或者其他)。

[0070] 可选地，在一种实施例中，上述步骤S104包括：

[0071] 响应于第一模块的服务调用请求，通过所述服务注册中心发现所述服务调用请求所对应的目标服务，并调用所述目标服务；其中所述目标服务由第二模块提供，所述第一模

块与所述第二模块分别为Gradle模块并相互依赖;所述第一模块包含第一实现组件和第一接口组件,所述第二模块包含第二实现组件和第二接口组件,所述目标服务基于所述第二接口组件而被调用。

[0072] 该方式可直接在一个模块中声明接口API与实现,以自动暴露模块接口,既可在模块直接被依赖时只暴露其API,亦可在发布时同时自动发布出API与实现这两个模块。相较于现有技术而言,该方式避免了接口下沉形成超级公共接口模块,也无需为每个暴露服务的模块单独新建API模块。

[0073] 举例来说,在一个具体实现方式中,对于两个功能模块,使它们在Gradle模块中互相依赖,从而各自屏蔽实现,而仅暴露接口API。一种做法是在src->main目录下分别添加相应模块的API目录,并各自独立编译API部分的代码,从而使其脱离Android插件的编译范围,从根本上避免了循环依赖的产生。

[0074] 实施例二

[0075] 对应于上述实施例一,本申请实施例二提供了一种基于路由网关的服务调用装置,相关技术特征的详细描述和对应的技术效果可参考上述实施例一,该处不再赘述。图4为本申请实施例二提供的业务数据的服务调用装置的框图,如图4所示,该装置包括元数据生成模块201、完备性检查模块202、注册模块203、服务调用模块204。

[0076] 其中,元数据生成模块201用于在编译期生成多个模块的模块元数据,并汇总模块元数据;完备性检查模块202,用于对汇总的模块元数据执行完备性检查,并生成对应的汇总入口类;注册模块203,用于将汇总入口类向服务注册中心注册;服务调用模块204,用于响应于服务调用请求,通过服务注册中心发现服务调用请求所对应的目标服务,并调用目标服务。

[0077] 可选地,在一种实施例中,完备性检查模块202包括第一子模块,用于对汇总的模块元数据执行完备性检查;还包括第二子模块,用于若所述完备性检查被成功执行,生成对应的汇总入口类。。

[0078] 可选地,在一种实施例中,第一子模块用于生成汇总的模块元数据对应的配置文件,并基于所述配置文件对所述汇总的模块元数据执行完备性检查。

[0079] 可选地,在一种实施例中,完备性检查包括:依赖关系检查;循环依赖检查。

[0080] 可选地,在一种实施例中,服务调用模块204包括第三子模块、第四子模块和第五子模块。其中,第三子模块用于基于预设的路径规则对目标资源标识符执行匹配;第四子模块用于跳转至匹配所得的目标页面,并加载第一模块;第五子模块用于响应于所述第一模块的服务调用请求,通过所述服务注册中心发现所述服务调用请求所对应的目标服务,并调用所述目标服务。

[0081] 可选地,在一种实施例中,第三子模块用于确定目标资源标识符对应的路由类型,并基于所述路由类型及预设的路径规则对目标资源标识符执行匹配。

[0082] 可选地,在一种实施例中,服务调用模块204用于响应于第一模块的服务调用请求,通过所述服务注册中心发现所述服务调用请求所对应的目标服务,并调用所述目标服务;其中所述目标服务由第二模块提供,所述第一模块与所述第二模块分别为Gradle模块并相互依赖;所述第一模块包含第一实现组件和第一接口组件,所述第二模块包含第二实现组件和第二接口组件,所述目标服务基于所述第二接口组件而被调用。

[0083] 实施例三

[0084] 本实施例三还提供一种计算机设备，如可以执行程序的智能手机、平板电脑、笔记本电脑、台式计算机、机架式服务器、刀片式服务器、塔式服务器或机柜式服务器(包括独立的服务器，或者多个服务器所组成的服务器集群)等。如图5所示，本实施例的计算机设备01至少包括但不限于：可通过系统总线相互通信连接的存储器011、处理器012，如图5所示。需要指出的是，图5仅示出了具有组件存储器011和处理器012的计算机设备01，但是应理解的是，并不要求实施所有示出的组件，可以替代的实施更多或者更少的组件。

[0085] 本实施例中，存储器011(即可读存储介质)包括闪存、硬盘、多媒体卡、卡型存储器(例如，SD或DX存储器等)、随机访问存储器(RAM)、静态随机访问存储器(SRAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、可编程只读存储器(PROM)、磁性存储器、磁盘、光盘等。在一些实施例中，存储器011可以是计算机设备01的内部存储单元，例如该计算机设备01的硬盘或内存。在另一些实施例中，存储器011也可以是计算机设备01的外部存储设备，例如该计算机设备01上配备的插接式硬盘，智能存储卡(Smart Media Card,SMC)，安全数字(Secure Digital,SD)卡，闪存卡(Flash Card)等。当然，存储器011还可以既包括计算机设备01的内部存储单元也包括其外部存储设备。本实施例中，存储器011通常用于存储安装于计算机设备01的操作系统和各类应用软件，例如实施例二的基于路由网关的服务调用装置的程序代码等。此外，存储器011还可以用于暂时地存储已经输出或者将要输出的各类数据。

[0086] 处理器012在一些实施例中可以是中央处理器(Central Processing Unit,CPU)、控制器、微控制器、微处理器、或其他数据处理芯片。该处理器012通常用于控制计算机设备01的总体操作。本实施例中，处理器012用于运行存储器011中存储的程序代码或者处理数据，例如应用组件的生成方法等。

[0087] 实施例四

[0088] 本实施例四还提供一种计算机可读存储介质，如闪存、硬盘、多媒体卡、卡型存储器(例如，SD或DX存储器等)、随机访问存储器(RAM)、静态随机访问存储器(SRAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、可编程只读存储器(PROM)、磁性存储器、磁盘、光盘、服务器、App应用商城等等，其上存储有计算机程序，程序被处理器执行时实现相应功能。本实施例的计算机可读存储介质用于基于路由网关的服务调用装置，被处理器执行时实现实施例一的应用组件的生成方法。

[0089] 需要说明的是，在本文中，术语“包括”、“包含”或者其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、物品或者装置不仅包括那些要素，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、物品或者装置所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

[0090] 上述本申请实施例序号仅仅为了描述，不代表实施例的优劣。

[0091] 通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。

[0092] 以上仅为本申请的优选实施例，并非因此限制本申请的专利范围，凡是利用本申

请说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本申请的专利保护范围内。

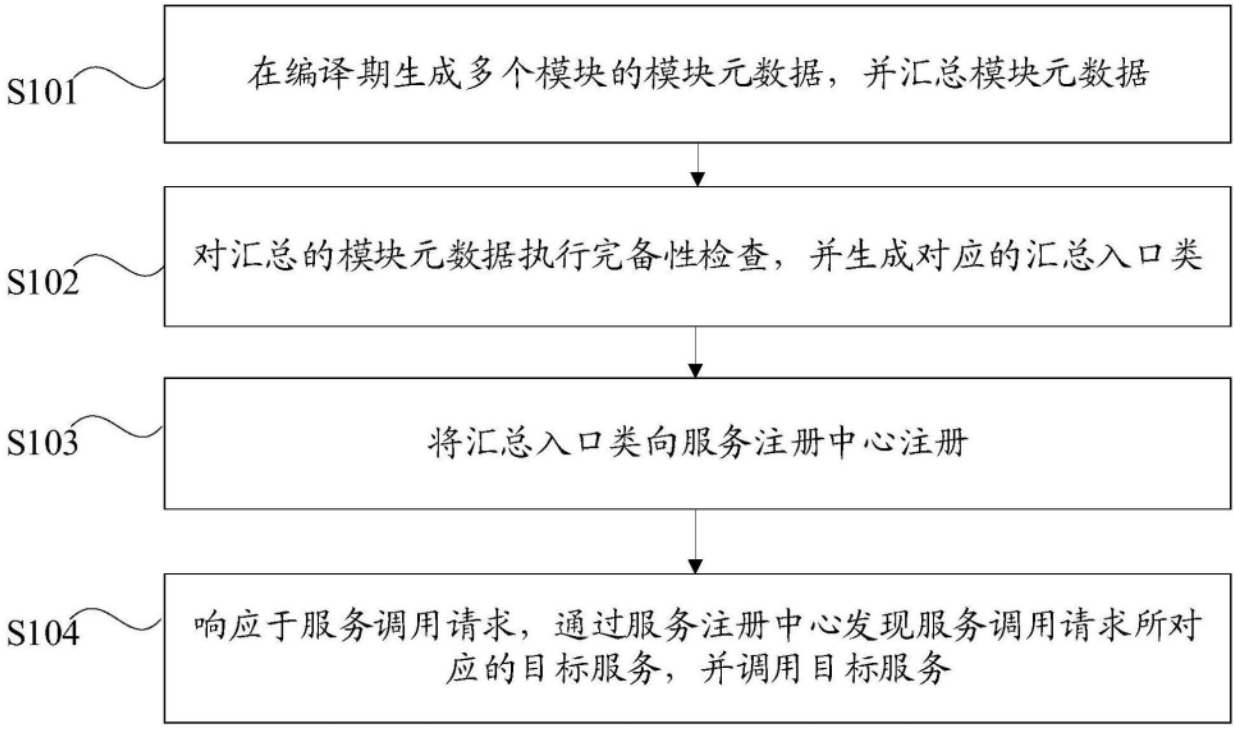

S101
在编译期生成多个模块的模块元数据，并汇总模块元数据
S102
对汇总的模块元数据执行完备性检查，并生成对应的汇总入口类
S103
将汇总入口类向服务注册中心注册
S104
响应于服务调用请求，通过服务注册中心发现服务调用请求所对应的目标服务，并调用目标服务

图1

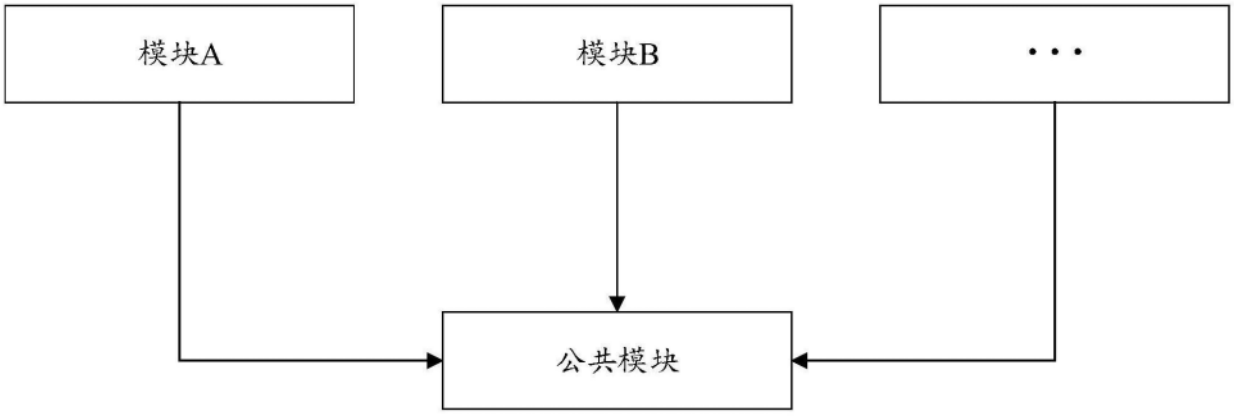

模块A
模块B
• • •
公共模块

图2a

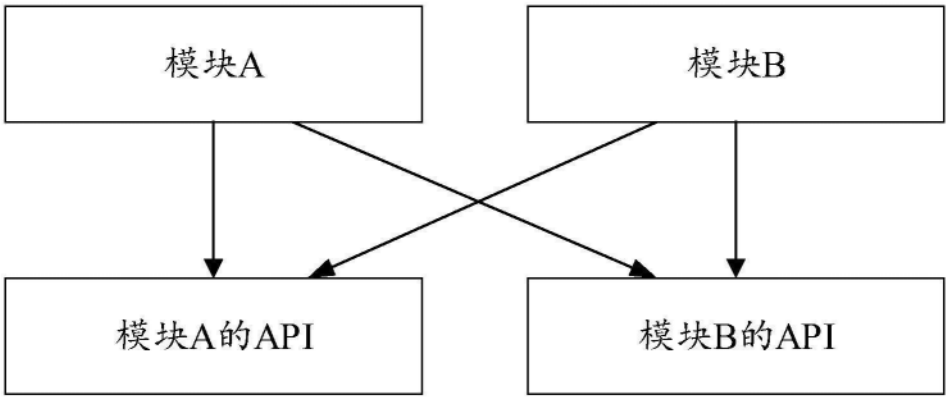

模块A
模块B
模块A的API
模块B的API

图2b

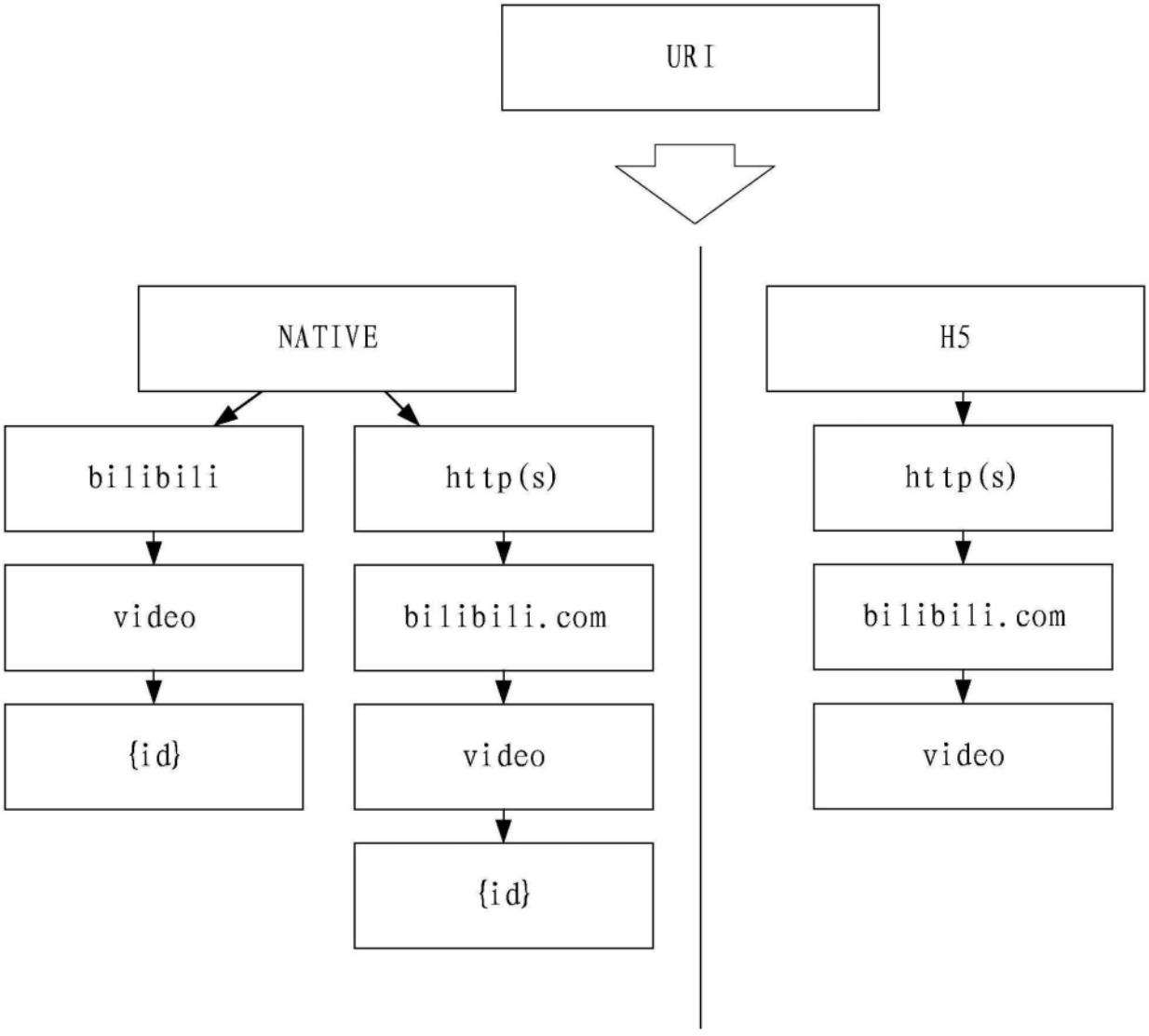

URI
NATIVE
H5
bilibili
http(s)
http(s)
video
bilibili.com
bilibili.com
{id}
video
video
{id}

图3

图4

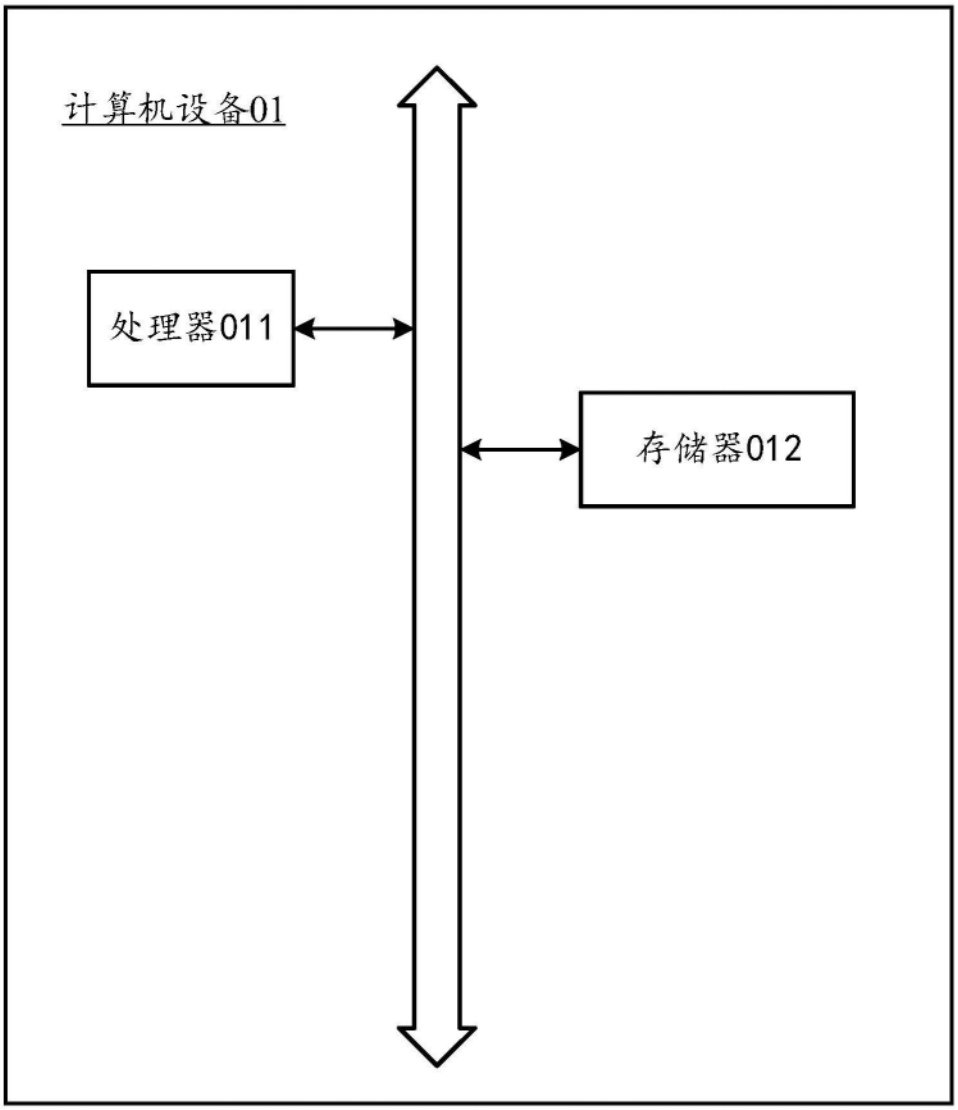


图5