(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104067258 A
(43)申请公布日 2014.09.24

(21)申请号 201280068675.9
(22)申请日 2012.04.26
(85)PCT国际申请进入国家阶段日
2014.07.31
(86)PCT国际申请的申请数据
PCT/US2012/035138 2012.04.26
(87)PCT国际申请的公布数据
WO2013/162561 EN 2013.10.31
(71)申请人 惠普发展公司,有限责任合伙企业
地址 美国德克萨斯州
(72)发明人 B.A. 希普 R. 巴拉赫瓦
T.S. 特里普 K.L. 威尔逊
M. 霍伊普特尔
(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 谢攀 徐红燕
(51)Int.Cl.
G06F 15/16(2006.01)
G06F 9/44(2006.01)

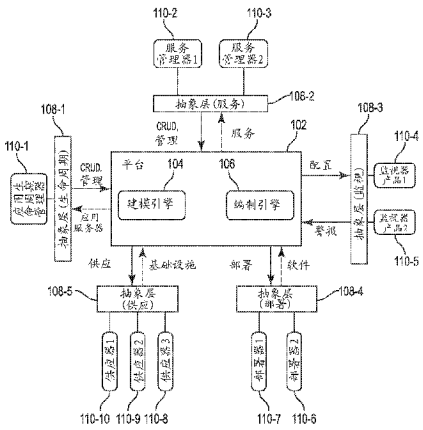
权利要求书2页 说明书9页 附图7页

(54)发明名称

平台运行时间抽象

(57)摘要

本发明提供了用于平台运行时间抽象的系统、方法以及机器可读和可执行指令。平台运行时间抽象可以包括在平台内创建多个模板模型,其中所述多个模板模型对应于提供器模型,以及创建允许提供器在运行时间插入到平台中的多个抽象层,其中提供器包括提供器子系统和提供器模型。平台运行时间抽象还可以包括允许提供器在运行时间通过多个抽象层之一插入到平台中,其中提供器保持独立于平台并且其中提供器保持独立于平台硬件基础设施和平台操作系统基础设施。



CN 104067258 A

1. 一种用于平台运行时间抽象的方法，包括：

在平台内创建多个模板模型，其中所述多个模板模型对应于提供器模型；

创建允许提供器在运行时间插入到平台中的多个抽象层，其中所述提供器包括提供器子系统和提供器模型；以及

允许提供器通过多个抽象层之一在运行时间插入到平台中，其中所述提供器保持独立于平台，并且其中所述提供器保持独立于平台硬件基础设施和平台操作系统基础设施。

2. 根据权利要求 1 的方法，其中所述方法包括向平台提供建模引擎，所述建模引擎促进平台模型中所存储的多个工件的使用，其中所述多个工件包括提供器工件并且其中所述提供器工件在提供器插入到平台中时被创建。

3. 根据权利要求 2 的方法，其中所述方法包括向平台提供编制引擎，所述编制引擎对多个工件的控制流进行协调。

4. 根据权利要求 1 的方法，其中创建允许提供器在运行时间插入到平台中的多个抽象层包括创建特定于提供器类型的抽象层。

5. 根据权利要求 1 的方法，其中允许提供器通过抽象层在运行时间插入到平台中包括平台从提供器接收多个消息以及平台通过抽象层向提供器发送多个消息以将提供器模型与多个模板模型之一进行绑定。

6. 一种非临时计算机可读介质，其存储由于平台运行时间抽象的指令，所述可由计算机执行而使所述计算机进行以下动作：

提供平台，所述平台包括用以促进平台模型中所存储的多个工件的使用的建模引擎，其中所述多个工件包括对应于多个提供器的多个提供器工件，以及包括用以对多个工件的控制流进行协调的编制引擎；

在平台内创建多个模板模型，其中所述多个模板模型对应于多个提供器模型；

创建允许多个提供器在运行时间插入到平台中的多个抽象层，其中所述多个提供器包括多个子系统和多个提供器模型；以及

允许多个提供器通过抽象层在运行时间插入到平台中，其中所述多个提供器模型保持独立于平台基础设施。

7. 根据权利要求 6 的介质，其中所述多个提供器包括消耗提供器资源的第一提供器群组，提供提供器资源的第二提供器群组，以及消耗提供器资源并提供提供器资源的第三提供器群组。

8. 根据权利要求 7 的介质，其中第一提供器群组和第二提供器群组在不与第二提供器群组和第三提供器群组直接交互的情况下使用第二提供器群组和第三提供器群组的提供器资源。

9. 根据权利要求 6 的介质，其中所述多个提供器工件包括在多个提供器向平台注册并加载作为插件时所创建的多个提供器工件，并且其中对多个提供器的控制通过所述多个工件进行管理。

10. 根据权利要求 9 的介质，其中提供器资源流由编制引擎来进行管理，并且其中编制引擎通过管理提供器工件而对资源流进行管理。

11. 一种平台运行时间抽象系统，包括：

具有多个模板模型的平台，其中所述平台在运行时间注册并加载多个提供器；

抽象层，其在运行时间促进多个提供器模型的抽象，其中所述多个提供器包括多个提供器模型；

抽象层内的加标签层，其促进在多个提供器模型的运行时间利用对应于多个模板模型的标签而加标签；和

多个提供器，其在多个提供器注册并加载到平台中之后使用多个平台资源，并且其中所述多个提供器保持独立于平台硬件基础设施以及独立于多个提供器硬件基础设施。

12. 根据权利要求 11 的系统，其中所述抽象层和所述加标签层包括用于对多个提供器中的每一个内的提供器模板库加标签以及将每个提供器模板库链接到平台内的内容目录中的抽象层和加标签层，并且其中每个提供器模板库包括由多个提供器中的每一个提供的服务的描述。

13. 根据权利要求 12 的系统，其中所述内容目录包括平台通过多个提供器而向特定提供器提供的服务的列表的描述以及特定提供器向多个提供器提供的服务的列表的描述。

14. 根据权利要求 11 的系统，其中所述多个提供器包括与平台操作系统基础设施和多个提供器操作系统基础设施保持独立的多个提供器。

15. 根据权利要求 11 的系统，其中所述多个提供器包括：

对产品的生命周期进行管理的应用生命周期管理提供器，其中所述生命周期包括从产品的开发到生产进展的产品的多个阶段；

向平台提供多个硬件资源的供应器提供器，其中所述多个硬件资源包括具有多个不同操作系统的多个不同硬件资源；

在多个供应器提供器上配置并安装软件的部署器提供器；

对多个提供器进行监视并且在检测到性能问题时警告平台的监视器提供器；以及

对通过多个提供器提供的多个服务进行管理的服务管理器提供器。

平台运行时间抽象

背景技术

[0001] 作为私有或公共云的云服务正在获得势头。保持在云系统和其它类型的系统上运行的应用的可用性是重要的。混合云系统由于私有云系统寻求扩展至公共云功能中而变得日益普及。将私有云系统绑定(binding)至公共云系统会影响在混合云系统上运行的应用的可用性。

附图说明

[0002] 图 1 图示了根据本公开的用于运行时间抽象的平台的示例。

[0003] 图 2 图示了根据本公开的用于注册提供器的示例。

[0004] 图 3 图示了根据本公开的用于加标签于模板库的示例。

[0005] 图 4 图示了根据本公开的平台之间的多个绑定的示例。

[0006] 图 5 图示了示出根据本公开的平台运行时间(runtime)抽象的示例的流程图。

[0007] 图 6 是图示用于平台运行时间抽象的方法的示例的流程图。

[0008] 图 7 图示了根据本公开的与用于平台运行时间抽象的处理资源进行通信的机器可读介质的示例的框图。

具体实施方式

[0009] 本公开的示例可以包括用于平台运行时间抽象的方法和系统。用于平台运行时间抽象的示例方法可以包括：在平台内创建多个模板模型，其中所述多个模板模型对应于提供器模型；以及创建允许提供器在运行时间插入到平台中的多个抽象层，其中该提供器包括提供器子系统和提供器模型。用于平台运行时间抽象的示例方法还可以包括允许提供器在运行时间通过多个抽象层中的一个插入到平台中，其中提供器保持独立于平台并且其中提供器保持独立于平台硬件基础设施和平台操作系统基础设施。

[0010] 此处的附图遵循编号惯例，其中最前的一个或多个位对应于附图编号而其余位则识别该附图中的元件或组件。不同示图之间的类似元件或组件可以通过使用类似的位进行识别。例如，102 可以引用图 1 中的元件“02”，而类似要素在图 2 中可以被 202 所引用。

[0011] 如此处所使用的，“一个”或“多个”某物可以指的是一个或多个这样的事物。例如，“多个小部件(widget)”可以指的是一个或多个小部件。

[0012] 混合云系统可以将两个或更多云系统进行组合。例如，混合云系统可以将私有云系统和公共云系统进行组合。公共云系统可以包括通过服务提供商而使得应用、储存和 / 或其它资源可供公众使用的云系统。私有云系统可以包括仅由一个实体操作以供该实体使用的云系统。

[0013] 混合云系统可以通过组合两个或更多云系统的资源而对两个或更多云系统进行组合。例如，混合云系统能够将公共云系统的硬件资源和私有云系统的监视资源进行组合。

[0014] 与第一云系统相关联的硬件资源和软件资源可能限制在第二云系统中应用的使用。例如，如果第一云系统与第一操作系统相关联，则第二云系统仅能够使用与第二云系统

相兼容的监视资源(例如,应用)。在本公开的多个示例中,混合云系统能够在不限制应用使用的情况下将独立于彼此的两个或更多云系统进行组合。

[0015] 图 1 图示了根据本公开的用于运行时间抽象的平台 102。平台 102 可以将多个提供器彼此绑定。提供器可以包括提供器子系统和提供器模型。提供器可以包括硬件子系统、软件子系统和 / 或硬件子系统和软件子系统的组合。提供器还可以包括提供资源,其中提供器可以将提供器资源提供给平台。在本公开的多个示例中,平台 102 可以将应用生命周期管理器提供器 110-1、第一服务管理器提供器 110-2、第二服务管理器提供器 110-3、第一监视器提供器 110-4、第二监视器提供器 110-5、第二部署提供器 110-6、第一部署提供器 110-7、第三供应(provisioning)提供器 110-8、第二供应提供器 110-9 和第一供应提供器 110-10 (一般称为提供器)彼此绑定。

[0016] 绑定多个提供器 110 可以包括在多个提供器 110 插入到平台 102 中时将多个提供器 110 彼此松散绑定并且松散绑定至平台 102。松散绑定可以包括允许多个提供器 110 在不依赖于彼此的情况下互相通信或者处于互相通信的形式。平台 102 可以通过用作多个提供器的实现引擎而松散绑定多个提供器。

[0017] 平台 102 能够通过从提供器之一接收多个请求以及通过向不同提供器进行所述请求而用作多个提供器的实现引擎。也就是说,平台 102 能够在不要求提供器之一 110-1 直接与多个提供器进行通信的情况下实现所述提供器的请求。

[0018] 平台 102 可以包括建模引擎 104。建模引擎 104 可以促进平台中所存储的多个工件(artifact)的使用。多个工件可以包括多个提供器的内部表示以使得当提供器向平台 102 注册时创建工件。工件可以对应于所注册的提供器并且对应于提供器模型。工件可以被表示为平台内的对象和 / 或结构。

[0019] 平台 102 还可以包括编制(orchestration)引擎 106。编制引擎 106 可以对多个工件的控制流进行协调。也就是说,平台 102 能够通过编制引擎而对多个工件的控制进行管理。例如,编制引擎 106 能够通过将对应于第一提供器的工件的控制给予第二提供器而将第一提供器的提供器资源的控制给予第二提供器。编制引擎 106 能够通过创建控制流并且通过终止控制流而对工件的控制流进行协调。能够在给予提供器对工件的控制时创建控制流。在本公开的多个示例中,工件能够由多个提供器所控制和 / 或提供器能够控制多个工件。在多个示例中,工件能够由单个提供器所控制和 / 或提供器能够控制单个工件。

[0020] 多个提供器 110 能够通过多个抽象层而被插入到平台 102 中。例如,多个抽象层可以包括生命周期抽象层 108-1、服务抽象层 108-2、监视抽象层 108-3、部署抽象层 108-4 和供应抽象层 108-5 (一般称为抽象层 108)。多个提供器 110 能够在运行时间通过多个抽象层 108 而向平台 102 注册。多个提供器 110 能够通过平台 102 而被彼此松散绑定。绑定能够在提供器 110 向平台 102 注册时创建。提供器可以包括提供器子系统和提供器模型。提供器子系统可以包括硬件子系统和 / 或软件子系统。硬件子系统可以包括多个硬件资源。多个软件子系统可以包括多个软件资源。多个软件资源可以包括编译软件和 / 或非编译软件。

[0021] 抽象层 108 可以包括允许多个提供器 110 与平台 102 进行通信的接口。接口可以包括代码集和 / 或应用编程接口(API)。抽象层 108 可以允许提供器 110 通过对提供器 110 和平台 102 之间例如通信的消息集进行规范化而与平台 102 进行通信。可以通过允许具有

不同通信格式的两个系统(例如提供器 110 和平台 102)彼此通信而对消息集合进行规范化。例如,当第一系统基于第一操作系统而第二系统基于第二操作系统时,例如提供器 110 和平台 110 的两个系统可以具有不同的通信格式。此外,当两个系统共享类似操作系统但是包括通信的消息被不同地格式化时,两个系统可以具有不同的通信格式。例如,不同通信格式可以包括发送和接收以第一格式进行格式化的消息的第一系统,以及发送和接收以第二格式进行格式化的消息的第二系统,其中该第一格式不同于第二格式。

[0022] 抽象层 108 可以包括特定于提供器类型的抽象层。提供器类型可以通过多个提供器的子集所共享的特性来识别。例如,第一提供器类型可以包括共享通信格式的多个提供器。共享通信格式可以包括大体上类似的多个通信格式和 / 或共享关键相似性的多个通信格式。关键相似性可以包括平台 102 所要求的通信格式的特性。在多个示例中,提供器类型可以包括提供相似资源和 / 或消耗相似资源的多个提供器。当抽象层能够改变以所述提供器类型的提供器所使用的第一格式的消息以使得平台 102 能够接收第二格式的经改变消息时,抽象层可以特定于该提供器类型。例如,应用生命周期管理器提供器 110-1 能够通过生命周期抽象层 108-1 而与平台 102 进行通信,第一服务管理器提供器 110-2 和第二服务管理器提供器 110-3 能够通过服务抽象层 108-2 而与平台 102 进行通信,第一监视器产品提供器 110-4 和第二监视器产品提供器 110-5 能够通过监视抽象层 108-3 而与平台 102 进行通信,第二部署提供器 110-6 和第一部署提供器 110-7 能够通过部署抽象层 108-4 而与平台 102 进行通信,并且第三供应提供器 110-8、第二供应提供器 110-9 和第一供应提供器 110-10 能够通过供应抽象层 108-5 而与平台 102 进行通信。

[0023] 在图 1、图 2 和图 3 中,提供器 110、提供器 210 和提供器 310 与平台 102、平台 202 和平台 302 之间通过抽象层 108、抽象层 208 和抽象层 308 的通信能够由实线和虚线所表示。实线可以指示请求和回复。虚线包括资源。由虚线所标识的资源可以包括对资源的引用,其中引用可以包括资源的描述和 / 或资源特性。

[0024] 在本公开的多个示例中,抽象层 108 可以是平台 102 的部分。在本公开的一些示例中,抽象层 108 可以与平台 102 分离。

[0025] 提供器模型可以包括提供器子系统的内部表示,其中提供器模型可以在提供器内部。提供器模型可以被用来向平台 102 注册提供器。

[0026] 多个提供器可以包括第一群组,其中提供器的第一群组消耗资源。例如,多个服务管理器提供器 110-2 和 110-3 可以消耗多个提供器 110 能够提供的多个资源。多个提供器还可以包括提供资源的提供器的第二群组。例如,多个供应提供器 110-8、110-9 和 110-10 可以提供资源。多个提供器可以进一步包括消耗资源和提供资源的提供器的第三群组。例如,多个部署提供器 110-6 和 110-7 可以消耗资源和提供资源。

[0027] 供应提供器 110-8、110-9 和 110-10 可以包括多个处理系统、储存系统和 / 或连网系统。供应提供器可以包括以上示例中所未包括的其它类型的硬件系统。供应提供器还可以包括创建硬件基础设施的供应提供器。例如,供应提供器可以接收针对供应的请求,其中供应请求可以包括特定基础设施配置。特定基础设施配置可以包括具有满足供应请求中发现的规范的特定软件配置的特定服务器。特定基础设施配置可以包括更多和 / 或更少的组件。例如,特定基础设施配置可以包括网络配置和 / 或功率使用配置。供应提供器可以通过提供满足供应请求中发现的规范的基础设施而对供应请求进行响应。供应提供器能够包

括多个硬件系统和多个软件系统。供应提供器可以包括多个不同硬件配置与多个不同软件配置。不管硬件配置和/或软件配置如何，供应抽象层可以允许供应提供器与平台 102 进行通信。供应提供器并不限于硬件资源，而是可以包括虚拟资源和/或云资源。

[0028] 部署提供器能够在多个硬件资源上安装并配置软件。在本公开的多个示例中，在任意硬件系统上能够使用多个不同的部署提供器，因为可通过抽象层 108-4 而使得部署提供器资源可用。部署抽象层 108-4 能够允许多个部署提供器与平台 102 进行通信而不管部署提供器的配置如何。也就是说，多个部署提供器能够被设计为利用多种不同的操作系统运行。例如，第一部署提供器可以被设计为利用第一操作系统运行而第二部署提供器能够被设计为利用第二操作系统运行，以使得第一部署提供器与第二操作系统不兼容并且第二部署提供器与第一操作系统不兼容。平台 102 能够通过多个抽象层对第一部署提供器和第二操作系统之间的交互进行规范化。第一部署提供器能够作出第一请求，其中该第一请求被配置为由第一操作系统接收。第一请求能够被配置为使得第一操作系统能够理解该请求而第二操作系统则无法理解该请求。如果第一请求通过第一抽象层而被规范化，则平台 102 能够接收第一请求并且对第一请求进行配置以使得第二操作系统能够理解第二请求。对请求进行规范化可以包括接收对平台 120 已知的格式的请求。不管部署提供器如何，部署抽象层 108-4 都能够允许部署提供器与平台 102 进行通信。

[0029] 部署提供器 110-7 和 110-6 能够通过部署抽象层 108 从平台 102 接收部署请求。部署请求可以包括部署软件的请求。部署软件可以包括在服务器上安装、激活、适配和更新软件。部署软件可以包括比以上所列出的那些更多和/或更少的服务。例如，部署软件可以包括去激活软件。部署提供器 110-7 和 110-6 能够通过返回所部署软件的实例而对部署请求进行响应。所部署软件可以包括已经被安装在服务器和/或其它计算设备上的软件。

[0030] 监视产品提供器 110-4 和 110-5 能够包括对对象进行监视的提供器。对象可以包括不同提供器、应用、平台和/或任意其它对象。监视产品提供器能够接收例如在图 1 中被表示为配置的一组配置并且能够在所监视对象以预定义方式表现和/或并未以预定义方式表现时向平台 102 发送多个警报。

[0031] 应用生命周期管理器提供器 110-1 能够贯穿应用的多个生命周期而对应用进行管理。多个生命周期包括应用的不同阶段。应用的不同阶段可以包括构思(conception)、测试和实现。应用生命周期管理器提供器 110-1 能够通过生命周期抽象层 108-1 与平台 102 进行通信。应用生命周期管理器提供器 110-1 能够创建、读取、更新和删除(例如，在图 1 中被表示为 CRUD)不同应用版本，其中应用的每个版本对应于应用的不同生命周期。应用生命周期管理器提供器 110-1 能够接收针对被管理的应用和/或与被管理应用相关联的多个服务器的多个引用。此外，应用生命周期管理器提供器 110-1 能够向平台中进行多个调用，这对应用的不同生命周期的管理有所影响。应用生命周期管理器提供器 110-1 能够接收多个应用版本和/或针对那些应用版本所能够存储之处的多个引用。

[0032] 服务管理提供器 110-2 和 110-3 能够对多种服务进行管理。例如，服务管理提供器 110-2 和 110-3 可以包括开发服务管理提供器 110-2 和/或电子邮件服务管理提供器 110-3。然而，服务管理提供器可以包括对并未包括在以上示例中的多种服务的管理。服务管理提供器能够创建、读取、更新和删除(例如，在图 1 中被表示为 CRUD)与被管理的服务有关的多个对象。服务管理提供器能够通过经由服务抽象层 108-2 向平台 102 进行管理调用

而创建、读取、更新和删除多个对象。服务管理提供器 110-2 和 110-3 能够对平台 102 进行管理调用。对平台 102 的管理调用能够管理服务。平台 102 能够返回被管理的服务资源，其在图 1 中由将平台 102 连接至抽象层 108-2 的以虚线标记的服务所表示。例如，平台 102 能够向电子邮件服务管理提供器 110-3 返回电子邮件。

[0033] 图 2 图示了根据本公开的用于注册提供器 210 的示例。平台 202 可以包括多个模板模型 220。多个模板模型 220 可以包括多种提供器模型的内部表示，其例如处于平台 202 内部。提供器模型 230 可以包括提供器资源的表示。提供器 210 可以通过对提供器模型 230 加标签以使得模板模型 220 表示提供器模型 230 和提供器资源而向平台 202 注册 228。

[0034] 平台 202 可以通过经由一组共用 API 请求规范化输入而注册提供器 210。规范化输入可以包括以对平台 202 可访问的格式的提供器资源的表示。抽象层 208 可以接收针对规范化输入的请求并且通过提供器实现的 API 将该请求转发至加标签层 222。提供器实现的 API 可以包括定义对提供器 210 可访问的通信格式的 API。加标签层 222 能够从抽象层 208 接收请求并且发送经评估的输入请求。经评估的输入请求可以包括针对提供器 210 的特定调用。

[0035] 提供器 210 能够对经评估的输入请求进行处理并且利用包括提供器资源的消息进行响应，其中该消息和提供器资源为对提供器 210 可访问的格式。提供器资源能够关于提供器模型 230 进行定义。加标签层 220 能够接收包括提供器资源的消息并且对例如提供器模型 230 的提供器资源加标签，以使得经加标签的资源包括平台 202 所能够处理的格式的提供器模型 230 的表示。抽象层 208 能够接收经加标签的资源并且向平台 202 发送规范化资源。规范化资源可以包括对平台 202 可访问的消息，其中该消息包括经加标签的提供器模型 230。

[0036] 在本公开的多个示例中，加标签层 222 可以是抽象层 208 的部分。在本公开的一些示例中，加标签层 222 可以独立于抽象层 208。加标签层可以包括模型映射模块 224，其接收与提供器模型 230 相关联的参数、与提供器模型 230 相关联的要求以及与提供器模型 230 相关联的能力。模型映射模块 224 然后例如通过过滤模块 226 对包括在提供器模型 230 中的提供器资源的列表进行过滤以识别出模板模型 220 之一所请求的一组模型资源。提供器资源的列表和一组模型资源可以包括由提供器 210 所提供的多个资源的参数、要求和能力。模型映射模块 224 能够返回一组映射值。也就是说，提供器模型 230 能够被映射到模板模型 220 上。在多个示例中，该映射可以在运行时间进行。在一些示例中，该映射可以在运行时间之后发生。例如，如果提供器 210 能够向主控服务提供动态分配的网际协议（IP）地址，则加标签层 222 能够在运行时间之后对动态分配的 IP 地址加标签。

[0037] 在本公开的一些示例中，模型映射模块 224 和过滤模块 226 能够处于加标签层 222 和抽象层 208 的内部。在本公开的多个示例中，模型映射模块 224 和过滤模块 226 能够独立于抽象层 208 和加标签层 222。

[0038] 提供器 210 能够在运行时间向平台 202 注册。也就是说，提供器 210 能够在运行时间插入到平台 202 中。在运行时间将提供器 210 插入到平台 202 中通过改变抽象层 208 和 / 或通过对提供器模型 230 进行改变而允许提供器模型 230 与多个平台一起重用。提供器模型 230 的重用允许提供器 210 的便携性，其中提供器 210 的便携性包括在多个平台中使用提供器 210 同时允许提供器模型 230 被重用的能力。多个平台可以包括包含不同提供器

的多个平台。例如，第一平台可以包括第一提供器和第二提供器而第二平台可以包括第三提供器。在不对第一提供器进行修改而使其与第三提供器相兼容的情况下，第一提供器可以被重用并被插入到第二平台中。能够在运行时间链接至提供器模型 230 的模板模型 220 能够在运行时间之后被更新。更新能够由平台 202 和 / 或提供器 210 在提供器模型 230 发生变化时而发起。

[0039] 将提供器 210 在运行时间插入到平台 202 中可以包括将提供器模型 230 映射到模板模型 220 上。将提供器模型 230 映射到模板模型 220 上可以包括将提供器 210 与能够插入到平台 202 中的多个提供器进行绑定。抽象层 208 和加标签层 222 允许提供器 210 插入到平台 202 中，而提供器 210 保持独立于平台 202。提供器 210 能够通过保持独立于与平台 202 相关联的硬件和 / 或软件组件以及被绑定和 / 或能够被绑定至平台 202 的多个提供器而保持独立于平台 202。

[0040] 硬件可以包括与平台 202 相关联的多个硬件组件。例如，硬件可以包括与平台 202 相关联的计算机系统。硬件并不局限于计算机系统，而是可以包括连网组件、存储器组件以及与平台相关联的其它计算相关组件。硬件还可以包括能够与多个提供器相关联的多个硬件组件。例如，包括提供器 210 在内的多个提供器能够被插入到平台中并且所述多个提供器中的每一个可以与能够用来提供多个提供器资源的多个硬件组件相关联。每个提供器能够保持独立于其它提供器的硬件组件。

[0041] 提供器 210 能够保持独立于平台的软件组件以及多个提供器的软件组件。软件组件可以包括多个操作系统和 / 或计算机可读指令。软件组件并不局限于软件组件的以上示例而是可以包括软件组件的其它示例。

[0042] 由于平台 202 和提供器 210 之间的通信能够通过抽象层 208 进行规范化，所以提供器 210 能够保持独立于平台 202。由于提供器 210 与插入到平台 202 中的多个提供器之间的交互能够通过平台 202 进行规范化，所以提供器 210 能够保持独立于插入到平台 202 中的多个提供器。也就是说，抽象层 208 能够允许提供器 210 与平台 202 和多个提供器进行通信而无需改变提供器本地的通信格式和 / 或提供器模型 230。

[0043] 图 3 图示了根据本公开的用于对模板库 354 加标签的示例。模板库 354 能够包括提供器 310 本地的内容。例如，模板库 354 可以包括能够通过多种格式进行表达的内容。内容可以包括提供器 310 本地的文本、公式和 / 或其它内容表达。模板库 354 可以包括内容集合。

[0044] 通过用户界面 (UI)，设计者 352 能够通过抽象层 308 和 / 或过滤层 326 对提供器 310 中的模板库 354 进行浏览。设计者 352 能够包括构建平台 302 的用户。用户能够通过选择将被插入平台 302 中的多个提供器而构建平台 302。用户可以包括物理用户或自动化用户。设计者 352 能够从模板库 354 中选择多个模板。设计者 352 选择的模板能够通过过滤层 326 和抽象层 308 返回至平台作为建模的内容。建模的内容能够被包括在内容目录 356 中以使得内容目录 356 可以包括来自多个提供器的多个模板。

[0045] 平台 302 能够通过对多个模板加标签 322 而向用户和 / 或应用提供例如内容目录 356 的内容。平台 302 不必须理解所述多个模板以对多个模板加标签。也就是说，用户和 / 或应用能够在用户和 / 或平台 302 并不完全理解内容的情况下从内容目录 356 选择内容。

[0046] 在图 3 中，经由加标签而被链接 322 的模板库 354 和内容目录 356 之间的连接能

够构成加标签过程的概念表示并且被包括在平台 302、抽象层 308、过滤层 326 和提供器 310 之间的流程之中。

[0047] 图 4 图示了根据本公开的平台之间的多个绑定的示例。在本公开的多个示例中，应用提供器 410-1 能够通过平台绑定到多个提供器。也就是说，能够创建并设计平台以支持应用提供器。平台能够通过使得多个资源通过多个提供器可供应用提供器 410-1 使用而支持应用提供器 410-1。应用提供器可以包括表示应用资源的应用模型。完全构建的平台 405 可以包括与多个平台的多个绑定。

[0048] 多个提供器可以包括安全提供器 410-2、基础设施提供器 410-3、工作负载管理提供器 410-4、控制提供器 410-5、部署提供器 410-6、代理提供器 410-7、操作系统提供器 410-8、策略提供器 410-9 和用户 / 域提供器 410-10（一般称为提供器 410）。在本公开的多个示例中，多个提供器可以包括比以上示例中所包括的那些更多或更少的提供器。此外，以上所列出的提供器是说明性的并且可以包括提供不同资源的不同提供器。

[0049] 应用提供器 410-1 能够被松散绑定到多个提供器 410。例如，应用提供器 410-1 可以通过绑定 440-1 而被绑定至安全提供器 410-2，通过绑定 440-2 而被绑定至基础设施提供器 410-3，通过绑定 440-3 而被绑定至工作负载管理提供器 410-4，通过绑定 440-4 而被绑定至控制提供器 410-5，通过绑定 440-5 而被绑定至部署提供器 410-6，通过绑定 440-6 而被绑定至代理提供器 410-7，通过绑定 440-7 而被绑定至操作系统提供器 410-7，通过绑定 440-8 而被绑定至策略提供器，以及通过绑定 440-9 而被绑定至用户 / 域提供器 410-10。

[0050] 应用提供器 410-1 可以在运行时间被绑定至多个提供器 410。平台能够将应用提供器 410-1 绑定至多个提供器 410。该绑定可以包括松散绑定，因为应用提供器 410-1 可以在并不对应用提供器 410-1 内的应用模型进行改变并且应用提供器 410-1 并不直接与多个提供器 410 进行通信的情况下而被绑定至多个提供器 410。该运行时间的构建块构造能够允许创建包括应用提供器 410-1 和多个提供器 410 的平台。该平台能够在运行时间动态改变而并不影响应用提供器 410-1 和多个提供器 410。例如，如果平台包括作为开源部署提供器 410-6 的部署提供器 410-6，则该开源部署提供器 410-6 能够与专有部署提供器进行交换而并不影响作为该平台的部分的其它提供器。

[0051] 图 5 图示了示出根据本公开的平台运行时间抽象的示例的流程图 507。设计者 552 能够通过选择应用提供器以及绑定至平台的多个提供器而设计平台 502。设计者 552 能够从应用目录 560-1 和多个提供器目录中选择多个提供器。多个提供器目录可以包括安全目录 560-2、基础设施目录 560-3、工作负载管理目录 560-4、控制目录 560-5、部署目录 560-6（未示出）、代理目录 560-7（未示出）、操作系统目录 560-8（未示出）、策略目录 560-9（未示出）和用户 / 域目录 560-10（一般称为目录 560）。

[0052] 应用目录和多个提供器目录可以包括提供器描述，其使得设计者 552 能够选择将一起运作以支持应用提供器的提供器群组。例如，设计者 552 能够确定应用提供器需要多个提供器资源来进行运作。设计者 552 能够查看对多个提供器进行描述的多个提供器目录 560。设计者 552 能够从多个提供器中选择能够提供所需提供器资源的那些提供器。设计者 552 能够在运行时间之前选择应用提供器 560-1 和多个提供器 560。在本公开的多个示例中，设计者 552 能够在运行时间选择应用提供器 560-1 和多个提供器 560。在一些示例中，设计者 552 能够在运行时间之后修改选择。

[0053] 模型组装器(assembler)570能够在运行时间构造提供器模型。模型组装器570可以是例如图1中的建模引擎104的建模引擎的部分。模型组装器570能够在运行时间将应用提供器和多个提供器绑定至平台。完全构造的平台505可以由模型实现器572在运行时间所实现574,其中平台实现器572可以是例如图1中的编制引擎106的编制引擎的部分。

[0054] 图6是图示用于平台运行时间抽象的方法示例的流程图。在676,能够在平台内创建多个模板模型,其中多个模板模型对应于提供器模型。在678,能够创建允许提供器在运行时间插入到平台中的多个抽象层,其中提供器包括提供器子系统和提供器模型。在680,允许提供器通过多个抽象层之一在运行时间插入到平台中,其中该提供器保持独立于平台硬件基础设施和平台操作系统基础设施。

[0055] 在本公开的多个示例中,平台可以包括建模引擎,其促进平台中所存储的多个工件的使用,其中多个工件包括提供器工件并且其中提供器工件可以在提供器插入到平台中时被创建。平台还可以包括编制引擎。该编制引擎能够通过给予工件的提供器控制并且通过终止工件的控制而对多个工件的控制流进行协调。在本公开的多个示例中,工件能够由多个提供器所控制和/或提供器能够控制多个工件。在本公开的多个示例中,工件能够由单个提供器所控制和/或提供器能够控制单个工件。

[0056] 多个抽象层可以包括特定于提供器类型的多个抽象层。提供器类型可以包括共享通信格式的多个提供器。每种提供器类型可以包括抽象层,以使得来自多个提供器类型的多个提供器能够通过特定于该提供器类型的多个抽象层而与平台进行通信。

[0057] 提供器能够通过将提供器模型与模板模型进行绑定而插入到平台中。平台可以包括表示多个提供器模型的多个模板模型。绑定可以包括对提供器模型加标签的多个消息,其中在所述加标签中将提供器模型绑定至模板模型。

[0058] 图7图示了根据本公开的与处理资源进行通信以用于平台运行时间抽象的计算机可读介质的示例的框图。计算机可读介质788(例如,有形非临时介质)和/或存储器资源786能够存储可由处理器资源784执行以在平台内创建790多个模板模型的指令集,其中多个模板模型对应于提供器模型。该指令能够被执行以创建792允许提供器在运行时间插入到平台中的多个抽象层,其中提供器包括提供器子系统和提供器模型。该指令能够被执行以允许794提供器在运行时间通过多个抽象层之一插入到平台中,其中提供器保持独立于平台并且其中提供器保持独立于平台硬件基础设施和平台操作系统基础设施。

[0059] 抽象层和加标签层可以对提供器模板库加标签并且将提供器模板链接到内容目录中,其中提供器包括提供器模板库并且平台包括内容目录。提供器模板库可以包括提供器能够提供的服务的列表以及提供器能够提供的服务列表的描述。提供器模板库能够被链接到内容目录中以提供多个提供器能够提供的服务的列表以及多个提供器能够提供的服务列表的描述。内容目录可以在平台内提供以允许多个平台保持彼此独立。

[0060] 提供器能够保持独立于平台以及多个提供器。提供器能够保持独立于平台操作系统基础设施和多个提供器操作系统基础设施。平台操作系统可以包括与平台相关联的操作系统。提供器操作系统可以包括与多个提供器相关联的操作系统。

[0061] 多个提供器可以包括对产品的生命周期进行管理的应用生命周期管理提供器,其中生命周期包括从产品的开发到生产的进展的产品的多个阶段。多个提供器还可以包括向

平台提供多个硬件资源的供应器提供器，其中多个硬件资源包括具有多个不同操作系统的多个不同硬件资源。多个提供器还可以包括在多个供应器提供器上配置并安装软件的部署器提供器。多个提供器可以包括对多个提供器进行监视并且在检测到性能问题时警告平台的监视器提供器。多个提供器可以包括对通过多个提供器提供的多个服务进行管理的服务管理器提供器。

[0062] 此处所描述的方法、技术、系统和装置例如可以通过执行存储在计算机可读存储介质中的指令而以数字电子电路或计算机硬件来实现。实现这些技术的装置可以包括适当的输入和输出设备、计算机处理器和/或存储指令以供处理器执行的有形计算机可读存储介质。

[0063] 实现此处所描述的技术的过程可以通过处理器运行存储在有形计算机可读存储介质上的指令以用于通过对输入数据进行操作并生成适当输出而执行所期望的功能来实现。作为示例，合适的处理器包括通用和专用微处理器二者。用于存储可执行指令的合适的计算机可读存储设备包括所有形式的非易失性存储器，例如包括半导体存储器设备，诸如可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)和闪存设备；磁盘，诸如固定磁盘、软盘和可移动盘；包括磁带的其它磁性介质；以及诸如压缩盘(CD)或数字视频盘(DVD)之类的光学媒体。以上任意部件能够被特别设计的专用集成电路(ASIC)所补充或者被合并于其中。

[0064] 虽然所公开技术的操作在此处可以被描述为以某种次序和/或以某种组合来执行，但是在一些实现方式中，单独的操作可以以不同次序重新排列，与此处所描述的其它操作组合和/或被删除，并且仍然实现所期望的结果。类似地，所公开系统中的组件可以以不同方式进行组合和/或被其它组件所替代或补充，并且仍然可以实现所期望的结果。

[0065] 以上说明书、示例和数据提供了对方法和装置以及本公开的系统和方法的使用的描述。由于可以在不背离本公开的系统和方法的精神和范围的情况下形成许多示例，所以该说明书仅阐述许多可能的实施例配置和实现方式中的部分。

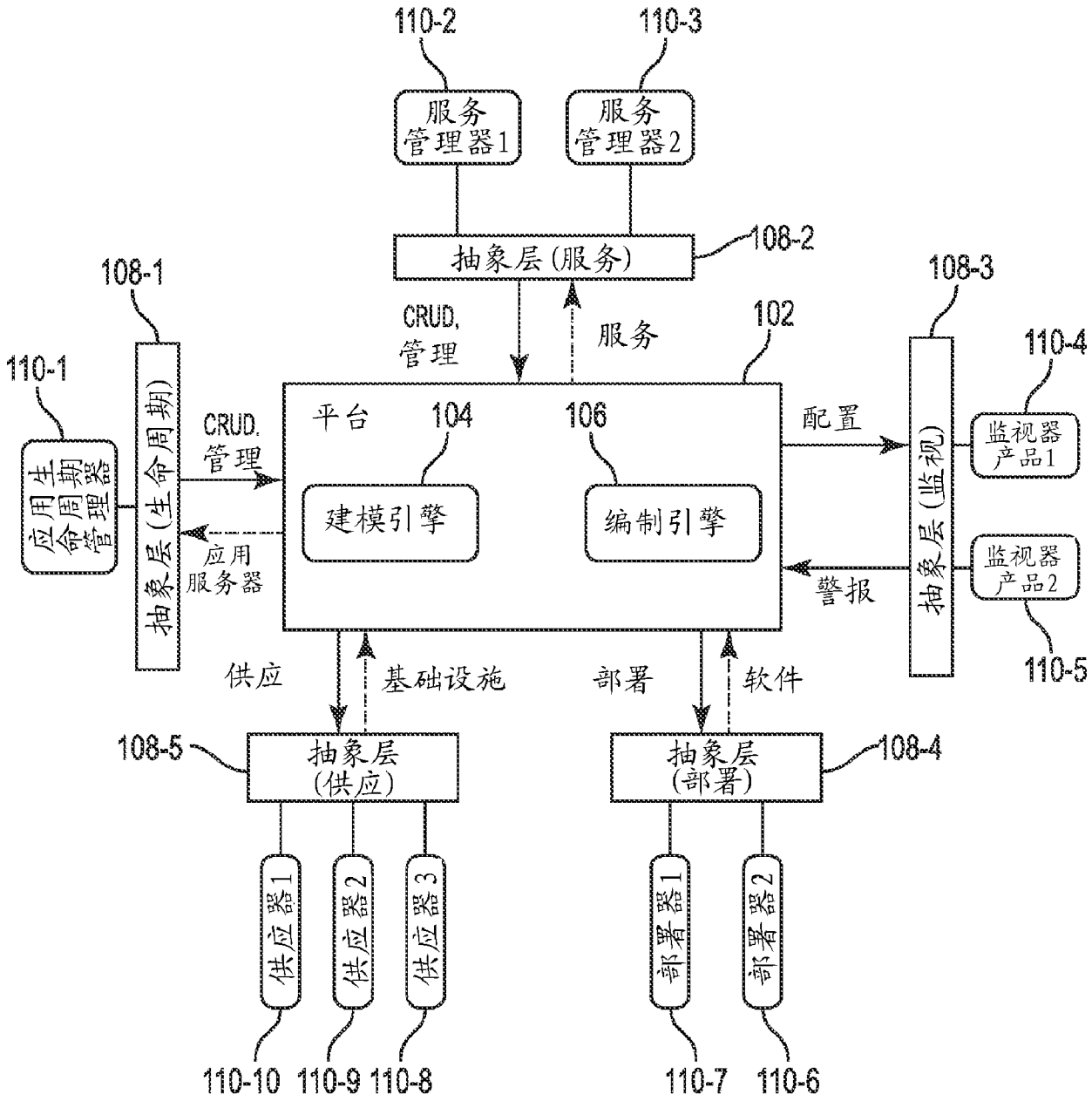

110-2
110-3
服务管理器1
服务管理器2
抽象层(服务)
108-2
108-1
108-3
CRUD,管理
服务
102
110-1
110-4
平台
104
106
配置
CRUD,管理
应用生命周期管理器
抽象层(生命周期)
建模引擎
编制引擎
监视器产品1
抽象层(监视)
应用服务器
监视器产品2
警报
110-5
供应
基础设施
部署
软件
108-5
抽象层(供应)
抽象层(部署)
108-4
供应器1
供应器2
供应器3
部署器1
部署器2
110-10
110-9
110-8
110-7
110-6

图 1

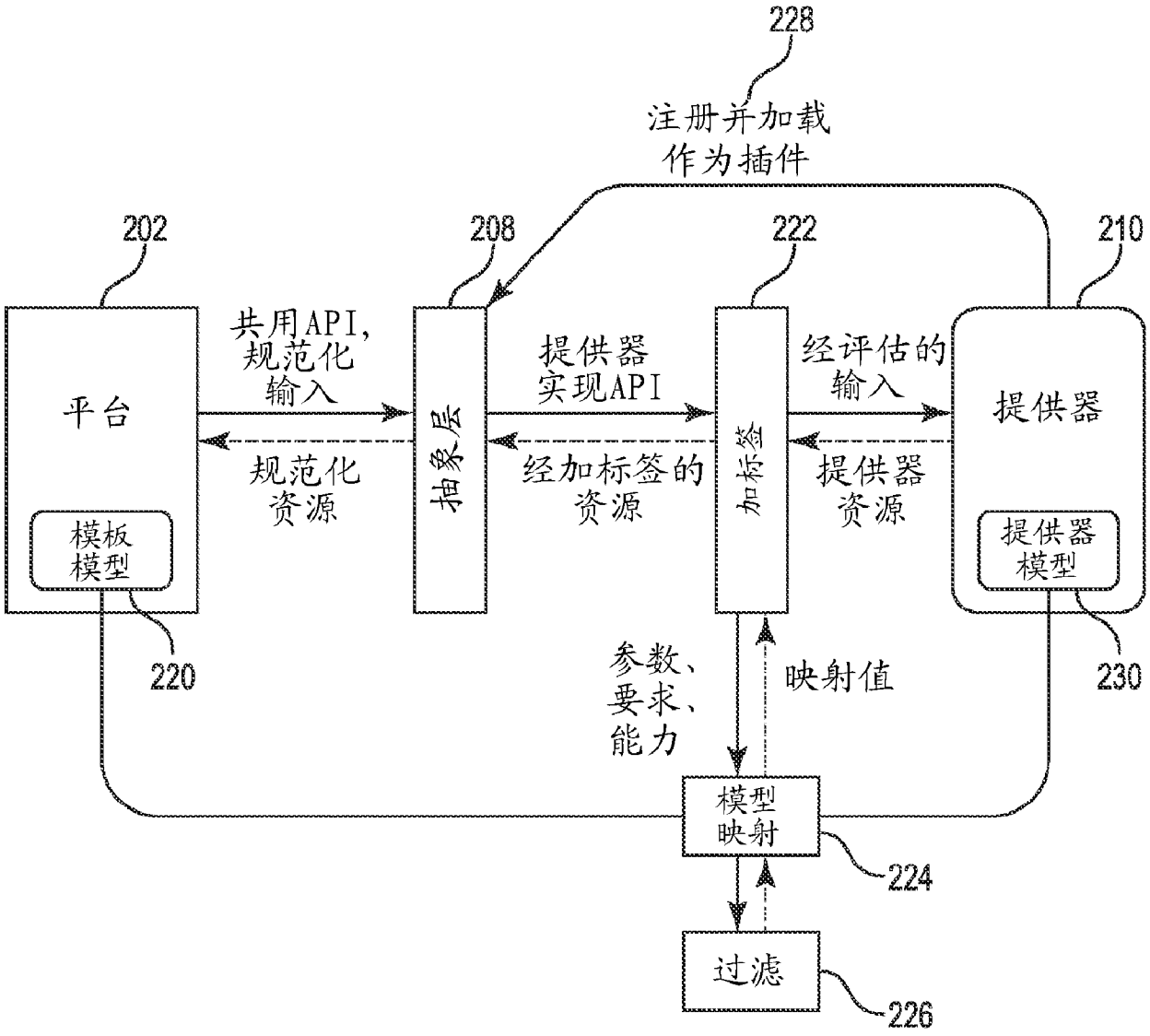

228
注册并加载
作为插件
202
208
222
210
平台
共用API,
规范化
输入
抽象层
提供器
实现API
经加标签的
资源
经评估的
输入
提供器
资源
提供器
规范化
资源
模板
模型
提供器
模型
220
230
参数、
要求、
能力
映射值
模型
映射
224
过滤
226

图 2

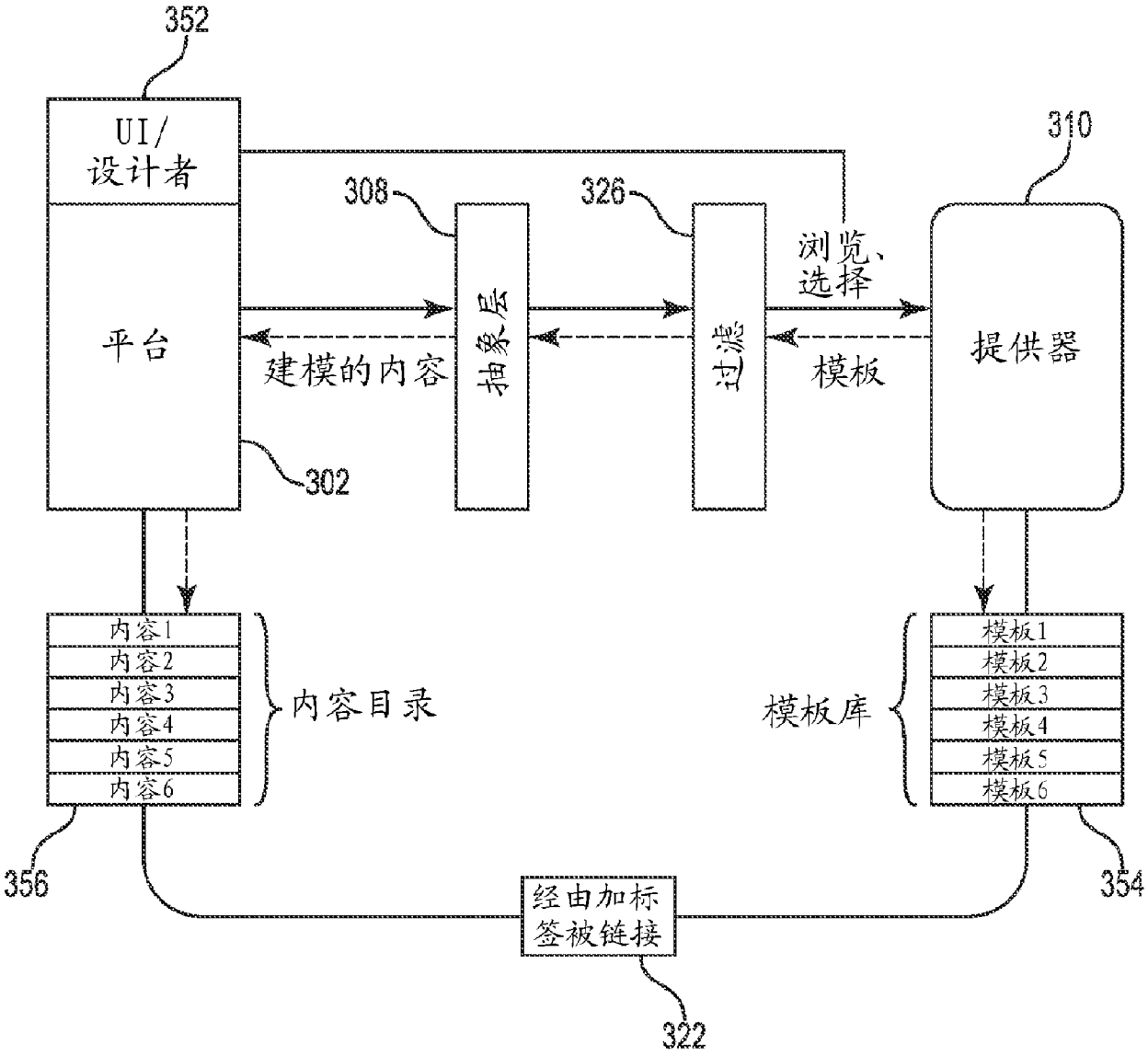

352
UI/
设计者
平台
302
308
抽象层
326
浏览、
选择
模板
建模的内容
310
提供器
内容1
内容2
内容3
内容4
内容5
内容6
内容目录
模板1
模板2
模板3
模板4
模板5
模板6
模板库
356
354
经由加标
签被链接
322

图 3

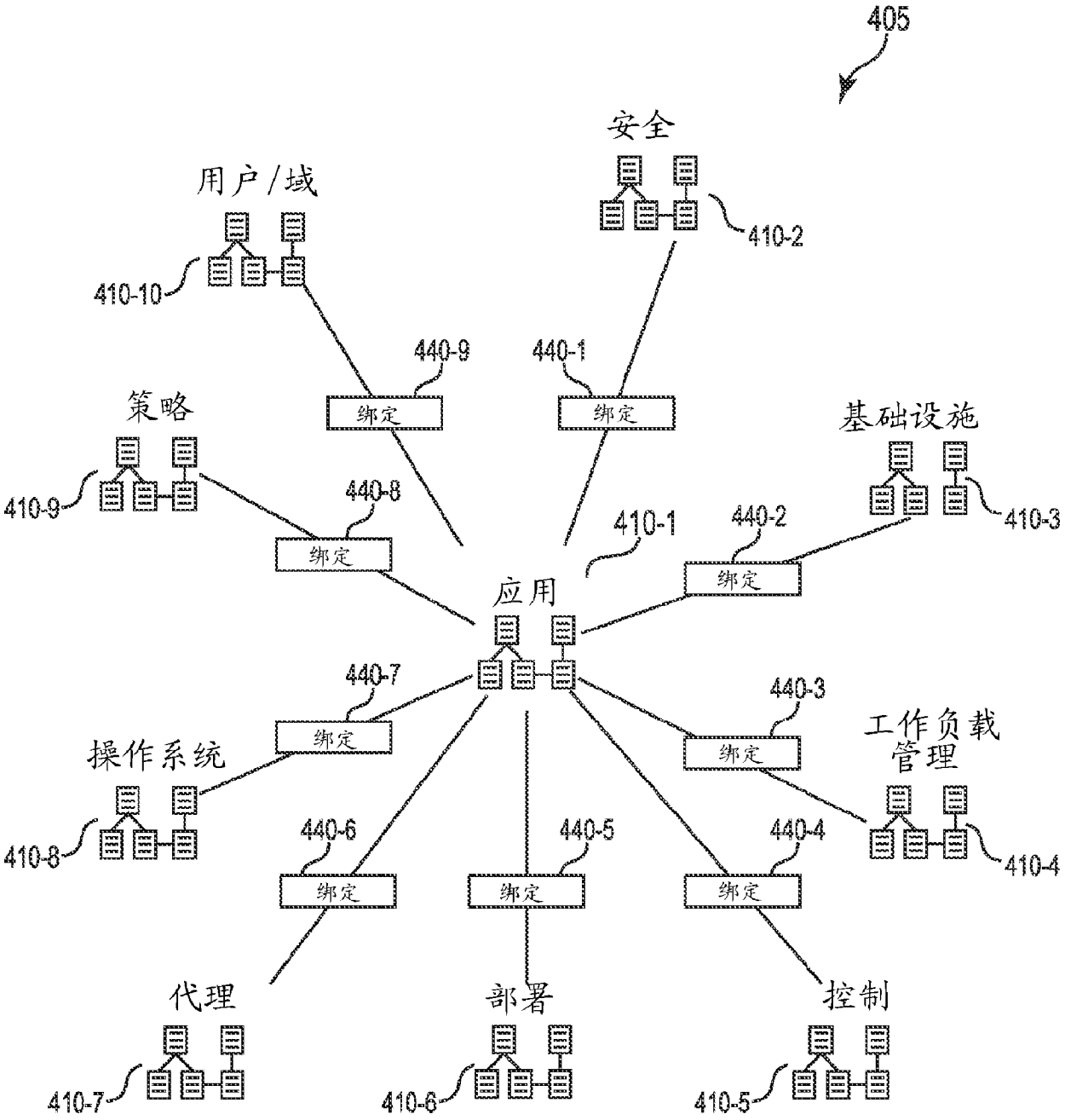

405
安全
410-2
用户/域
410-10
440-9
440-1
绑定
绑定
策略
410-9
440-8
绑定
基础设施
410-3
410-1
440-2
绑定
应用
440-7
绑定
440-3
绑定
操作系统
410-8
工作负载管理
410-4
440-6
440-5
440-4
绑定
绑定
绑定
代理
部署
控制
410-7
410-6
410-5

图 4

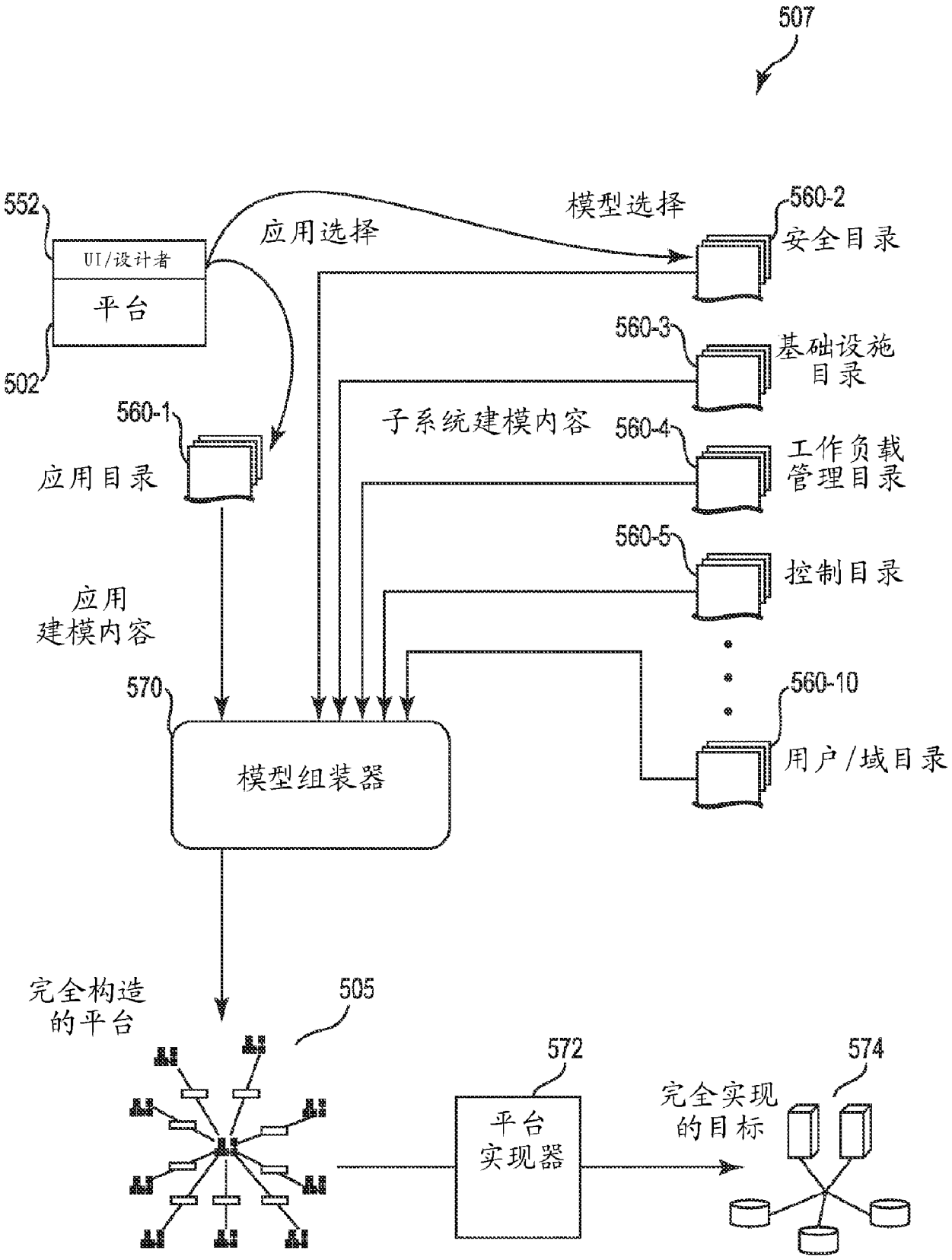

507
552
UI/设计者
平台
502
应用选择
模型选择
560-2
安全目录
560-3
基础设施
目录
560-1
应用目录
子系统建模内容
560-4
工作负载
管理目录
560-5
控制目录
应用
建模内容
570
模型组装器
560-10
用户/域目录
完全构造
的平台
505
572
平台
实现器
574
完全实现
的目标

图 5

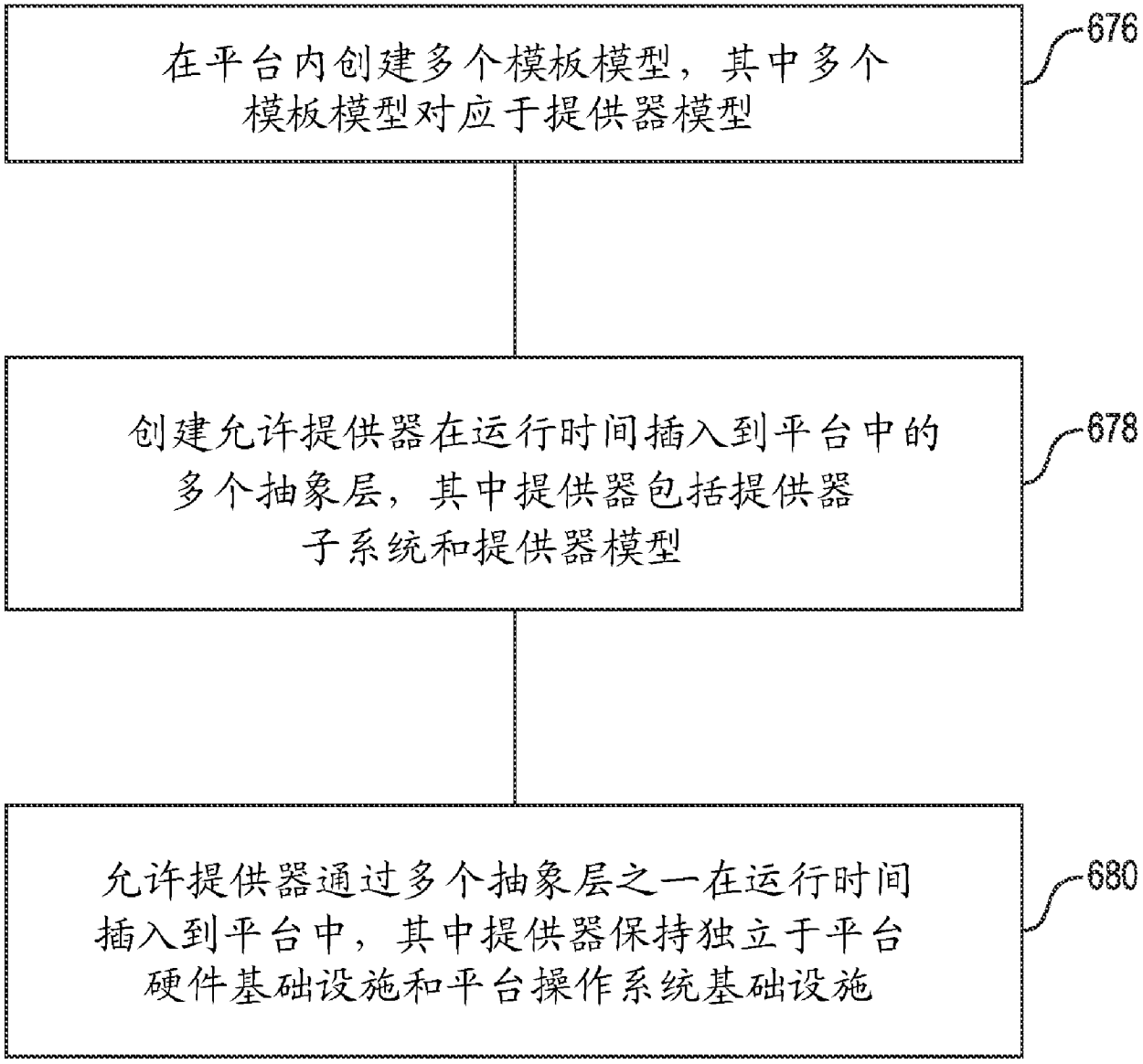

在平台内创建多个模板模型，其中多个模板模型对应于提供器模型
676
创建允许提供器在运行时间插入到平台中的多个抽象层，其中提供器包括提供器子系统和提供器模型
678
允许提供器通过多个抽象层之一在运行时间插入到平台中，其中提供器保持独立于平台硬件基础设施和平台操作系统基础设施
680

图 6

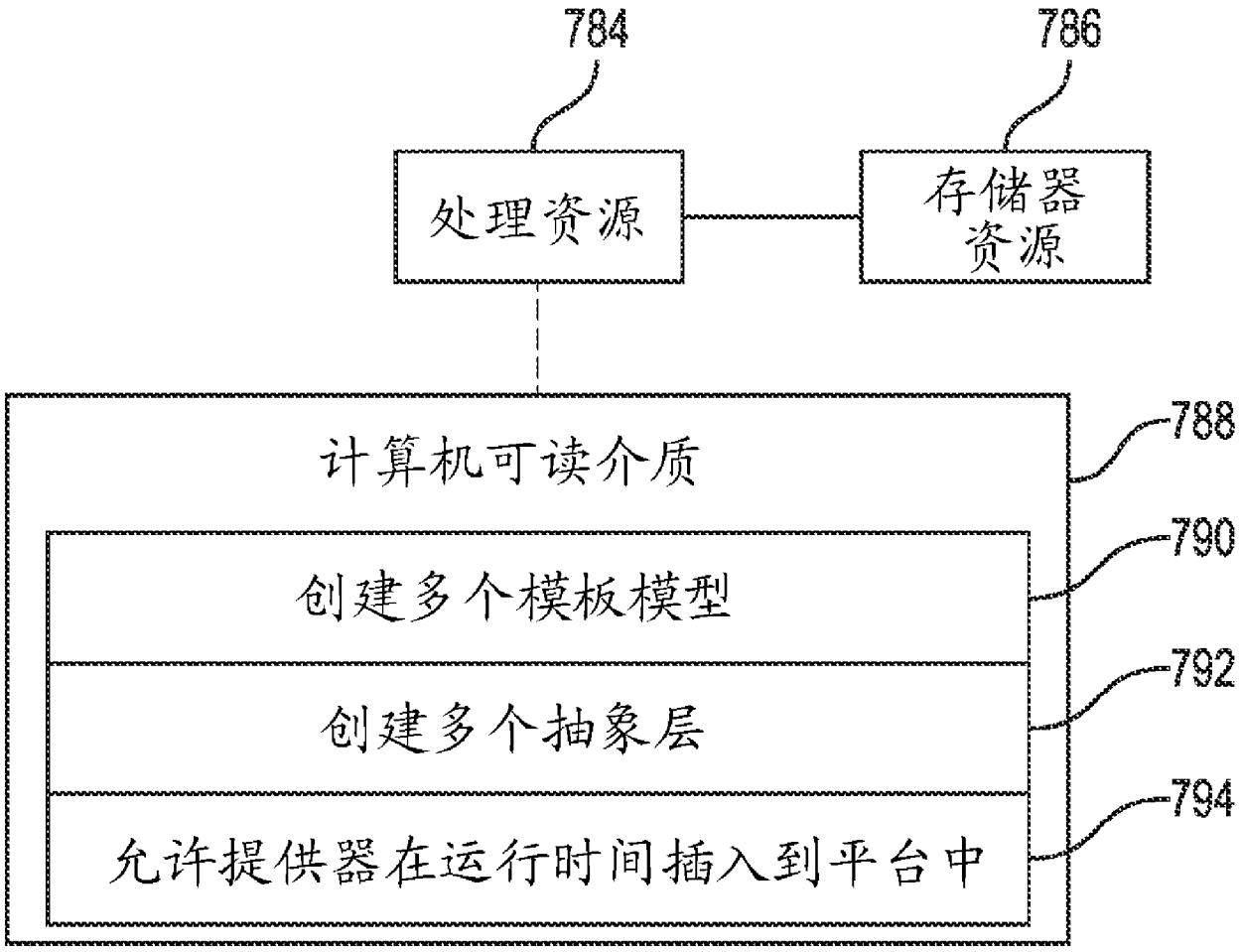

784
786
处理资源
存储器资源
计算机可读介质
788
创建多个模板模型
790
创建多个抽象层
792
允许提供器在运行时间插入到平台中
794

图 7