(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 106716404 B
(45)授权公告日 2020.12.11

(21)申请号 201580047318.8
(22)申请日 2015.04.27
(65)同一申请的已公布的文献号
申请公布号 CN 106716404 A
(43)申请公布日 2017.05.24
(30)优先权数据
62/054,613 2014.09.24 US
14/696,186 2015.04.24 US
(85)PCT国际申请进入国家阶段日
2017.03.03
(86)PCT国际申请的申请数据
PCT/US2015/027757 2015.04.27
(87)PCT国际申请的公布数据
WO2016/048418 EN 2016.03.31
(73)专利权人 甲骨文国际公司
地址 美国加利福尼亚
(72)发明人 N·汉达 N·卡温特泽斯
R·斯瑞瓦斯塔瓦
(74)专利代理机构 中国贸促会专利商标事务所有限公司 11038
代理人 李玲
(51)Int.Cl.
G06F 16/958(2019.01)
G06F 16/957(2019.01)
H04L 29/06(2006.01)
(56)对比文件
CN 102447708 A,2012.05.09
US 2008140857 A1,2008.06.12
US 2013227291 A1,2013.08.29
CN 1512707 A,2004.07.14
审查员 邓丽婉

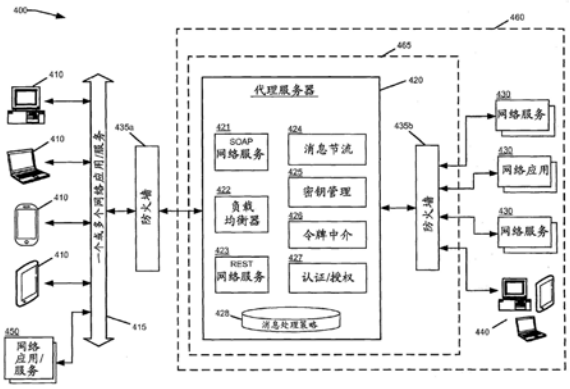
权利要求书4页 说明书23页 附图13页

(54)发明名称

计算机子网内的代理服务器

(57)摘要

本发明的实施例包括用于处理在计算机网络之间传输的消息的技术。在一些实施例中，可以在多个计算机网络之间传输消息(诸如，对各种类型的网络服务、应用和其他网络内容的请求和响应)。一个或多个中介装置或应用(诸如，在物理或逻辑子网内实现的代理服务器)可以对通信端点之间的消息进行接收、处理和发送。在一些实施例中，代理服务器可以被配置成在互联网计算机网络的子网内操作，从而将内部计算机网络的各种网络应用和/或服务暴露于外部计算机网络。这种代理服务器可以基于各种消息特征和针对消息的预定处理流程中的当前点来选择用于处理消息的特定策略。在选择了将应用于消息的特定策略之后，代理服务器可以按照策略来处理消息并且将消息转发到该消息的所期望目的地。



CN 106716404 B

1.一种处理在计算机网络之间传输的消息的方法,所述方法包括:

在内部计算机网络的子网内的代理服务器处从客户端装置接收第一消息,所述代理服务器将所述内部计算机网络的网络应用或服务的集合暴露于外部计算机网络;

确定从其接收所述第一消息的所述客户端装置是在所述内部计算机网络内操作的内部客户端装置还是在所述外部计算机网络内操作的外部客户端装置;

至少基于所述第一消息是从内部客户端装置还是外部客户端装置接收的,确定包括所述第一消息的应用处理流程是针对代理应用还是针对虚拟应用的;

基于对于所述处理流程的客户端装置是针对虚拟应用还是针对代理应用的确定,确定所述代理服务器充当正向代理还是反向代理;

确定所述代理服务器内的针对所述第一消息的所述应用处理流程中的当前点;

从所述代理服务器内的用于处理消息的多个策略中选择用于处理所述第一消息的策略,其中所述选择基于所述应用处理流程中的所述当前点和确定所述代理服务器充当正向代理还是反向代理;

按照所选择的策略来处理所述第一消息;以及

在处理所述第一消息之后,向目的地传输所述第一消息。

2.根据权利要求1所述的方法,其中所述代理服务器包括安全代理,并且其中,所述所选择的策略包括调用一个或多个网络服务安全策略的机器可执行代码。

3.根据权利要求1或2所述的方法,其中所述代理服务器包括在所述内部计算机网络的物理子网内操作的计算机系统。

4.根据权利要求1或2所述的方法,其中所述代理服务器包括在所述内部计算机网络的逻辑子网内执行的代理服务器应用。

5.根据权利要求1或2所述的方法,其中选择用于处理所述第一消息的策略包括:

确定在针对所述第一消息的处理流程期间已经发生错误;以及

基于确定已经发生错误来选择所述策略。

6.根据权利要求1或2所述的方法,其中选择用于处理所述第一消息的策略包括:

确定针对所述第一消息的到来消息格式;

确定所述第一消息需要被变换成与所述到来消息格式不同的所需要的输出消息格式;以及

基于确定所述第一消息需要被变换,选择所述策略。

7.根据权利要求1或2所述的方法,其中,确定针对所述第一消息的所述处理流程中的所述当前点包括以下至少一个:

确定所述第一消息对应于来自所述外部计算机网络中的客户端装置的请求;或者

确定所述代理服务器将向所述内部计算机网络中的网络应用或网络服务传输请求。

8.根据权利要求1或2所述的方法,其中,确定针对所述第一消息的所述处理流程中的所述当前点包括以下至少一个:

确定所述第一消息对应于来自所述内部计算机网络中的网络应用或网络服务对由所述代理服务器传输的之前消息的响应;或者

确定所述代理服务器将向所述外部计算机网络中的客户端装置传输响应。

9.根据权利要求1或2所述的方法,还包括:

确定所述第一消息调用所述内部计算机网络的简单对象访问协议SOAP虚拟服务内的一个或多个SOAP操作或者确定所述第一消息是所述一个或多个SOAP操作的部分；

基于所确定的SOAP操作和所述SOAP虚拟服务来选择用于处理所述第一消息的策略；以及

在按照所选择的策略处理了所述第一消息之后，使用所述第一消息内的数据来调用所确定的一个或多个SOAP操作。

10.根据权利要求1或2所述的方法，还包括：

确定所述第一消息对应于与所述内部计算机网络的表征状态转移REST虚拟服务或虚拟网络应用关联的一种或多种超文本传输协议HTTP方法；

基于所确定的所述HTTP方法和所述REST虚拟服务或虚拟网络应用来选择用于处理所述第一消息的策略；以及

在按照所选择的策略处理了所述第一消息之后，使用所述第一消息内的数据来调用所确定的一种或多种HTTP方法。

11.根据权利要求1或2所述的方法，还包括：

接收与所述第一消息关联的一个或多个用户凭证，其中所述第一消息对应于来自所述外部计算机网络中的客户端装置访问所述内部计算机网络的第一网络服务的请求；以及

使用所述用户凭证来认证与所述请求关联的第一用户。

12.根据权利要求11所述的方法，还包括：

确定需要第一令牌类型的认证令牌来访问所述第一网络服务；

从所述内部计算机网络的网络服务获取第一认证令牌，其中所述第一认证令牌是所述第一令牌类型的并且与所述第一用户关联；以及

使用所述第一认证令牌按照所述请求来访问所述第一网络服务。

13.根据权利要求1或2或12所述的方法，还包括：

基于所述应用处理流程中的所确定的所述当前点来执行OnRequest()、OnInvoke()、OnResponse()或onError()方法。

14.一种处理在计算机网络之间传输的消息的系统，所述系统包括：

处理单元，包括一个或多个处理器；以及

存储器，与所述处理单元耦合并且能由所述处理单元读取，并且所述存储器中存储了指令的集合，所述指令的集合当由所述处理单元执行时致使所述处理单元：

从客户端装置接收第一消息，其中所述系统被配置成在内部计算机网络的子网内操作，所述系统将所述内部计算机网络的网络应用或服务的集合暴露于外部计算机网络；

确定从其接收所述第一消息的所述客户端装置是在所述内部计算机网络内操作的内部客户端装置还是在所述外部计算机网络内操作的外部客户端装置；

至少基于所述第一消息是从内部客户端装置还是外部客户端装置接收的，确定包括所述第一消息的应用处理流程是针对代理应用还是针对虚拟应用的；

基于对于所述处理流程的客户端装置是针对虚拟应用还是针对代理应用的确定，确定所述系统充当正向代理还是反向代理；

确定针对所述第一消息的所述应用处理流程中的当前点；

从用于处理消息的多个策略中选择用于处理所述第一消息的策略，其中所述选择基于

所述应用处理流程中的所述当前点和确定所述系统充当正向代理还是反向代理；

按照所选择的所述策略来处理所述第一消息；以及

在处理所述第一消息之后，向目的地传输所述第一消息。

15.根据权利要求14所述的系统，其中，所述系统被配置成在所述内部计算机网络的物理子网内进行操作。

16.根据权利要求14或15所述的系统，所述存储器中还存储了其他指令，所述其他指令当由所述处理单元执行时致使所述处理单元：

确定所述第一消息调用所述内部计算机网络的简单对象访问协议SOAP虚拟服务内的一个或多个SOAP操作或者确定所述第一消息是所述一个或多个SOAP操作的部分；

基于所确定的SOAP操作和所述SOAP虚拟服务来选择用于处理所述第一消息的策略；以及

在按照所选择的策略处理了所述第一消息之后，使用所述第一消息内的数据来调用所确定的一个或多个SOAP操作。

17.根据权利要求14或15所述的系统，所述存储器中还存储了其他指令，所述其他指令当由所述处理单元执行时致使所述处理单元：

确定所述第一消息对应于与所述内部计算机网络的表征状态转移REST虚拟服务或虚拟网络应用关联的一种或多种超文本传输协议HTTP方法；

基于所确定的HTTP方法和所述REST虚拟服务或虚拟网络应用来选择用于处理所述第一消息的策略；以及

在按照所选择的策略处理了所述第一消息之后，使用所述第一消息内的数据来调用所确定的一种或多种HTTP方法。

18.一种非暂态计算机可读存储器，所述非暂态计算机可读存储器包括其中存储的指令的集合，所述指令的集合当由系统的处理器执行时致使所述处理器：

从客户端装置接收第一消息，其中所述系统被配置成在内部计算机网络的子网内操作，所述系统将所述内部计算机网络的网络应用或服务的集合暴露于外部计算机网络；

确定从其接收所述第一消息的所述客户端装置是在所述内部计算机网络内操作的内部客户端装置还是在所述外部计算机网络内操作的外部客户端装置；

至少基于所述第一消息是从内部客户端装置还是外部客户端装置接收的，确定包括所述第一消息的应用处理流程是针对代理应用还是针对虚拟应用的；

基于对于所述处理流程的客户端装置是针对虚拟应用还是针对代理应用的确定，确定所述系统充当正向代理还是反向代理；

确定针对所述第一消息的所述应用处理流程中的当前点；

从用于处理消息的多个策略中选择用于处理所述第一消息的策略，其中，所述选择基于所述应用处理流程中的当前点和确定所述系统充当正向代理还是反向代理；

按照所选择的策略来处理所述第一消息；以及

在处理所述第一消息之后，向目的地传输所述第一消息。

19.根据权利要求18所述的计算机可读存储器，所述计算机可读存储器还包括在其中存储的其他指令，所述其他指令当由所述处理器执行时致使所述处理器：

确定所述第一消息调用所述内部计算机网络的简单对象访问协议SOAP虚拟服务内的

一个或多个SOAP操作或者确定所述第一消息是所述一个或多个SOAP操作的部分；

基于所确定的SOAP操作和所述SOAP虚拟服务来选择用于处理所述第一消息的策略；以及

在按照所选择的策略处理了所述第一消息之后，使用所述第一消息内的数据来调用所确定的一个或多个SOAP操作。

20.根据权利要求18或19所述的计算机可读存储器，所述计算机可读存储器还包括在其中存储的其他指令，所述其他指令当由所述处理器执行时致使所述处理器：

确定所述第一消息对应于与所述内部计算机网络的表征状态转移REST虚拟服务或虚拟网络应用关联的一种或多种超文本传输协议HTTP方法；

基于所确定的HTTP方法和所述REST虚拟服务或虚拟网络应用来选择用于处理所述第一消息的策略；以及

在按照所选择的策略处理了所述第一消息之后，使用所述第一消息内的数据来调用所确定的一种或多种HTTP方法。

计算机子网内的代理服务器

[0001] 相关申请

[0002] 本申请要求2015年4月24日提交的、名称为“PROXY SERVERS WITHIN COMPUTER SUBNETWORKS”的美国非临时申请No.14/696,186的权益和优先权，该美国非临时申请要求2014年9月24日提交的、名称为“MOBILE SECURITY ACCESS SERVER(MSAS)”的美国临时专利申请No.62/054,613的权益和优先权。上述专利申请的全部内容出于所有目的以引用方式并入本文中。

背景技术

[0003] 本公开一般涉及用于提供安全服务的系统、方法和机器可读介质。更具体地，本公开涉及用于提供移动装置和企业应用之间的连接的安全服务的系统、方法和机器可读介质，这些安全服务包括认证、授权、审计、单点登陆、安全策略实施、密钥管理和分配、安全通信、安全数据存储和安全数据共享等。

发明内容

[0004] 本文中描述的一些方面提供了用于处理在计算机网络之间传输的消息的各种技术。在一些实施例中，可以在多个计算机网络之间传输消息(诸如，针对各种类型的网络服务、应用和其他网络内容的请求和响应)。一个或多个中介装置或应用(诸如，在物理或逻辑子网内实现的代理服务器)可以对通信端点之间的消息进行接收、处理和传输。例如，代理服务器可以被配置成在内部计算机网络的子网内操作，从而将内部计算机网络的各种网络应用和/或服务暴露于外部计算机网络。

[0005] 在某些实施例中，代理服务器可以接收从内部网络中的端点传输到外部系统中的端点(或反之亦然)的消息。可以分析该消息，以确定消息的所期望目的地，和/或确定代理服务器在处理消息时应该充当正向代理还是反向代理。另外，代理服务器可以确定预定处理流程(诸如，用于处理特定消息的端到端策略模型)中的当前点。基于对消息的分析和预定处理流程中的当前点，代理服务器可以选择将应用于消息的一个或多个策略。这些策略可以包括安全策略以及其他通信管理策略，例如以认证消息、提供安全令牌仲裁(mediation)和密钥管理、执行协议和有效负荷仲裁、执行基于装置的安全、支持隔离区(DMZ)威胁保护等。在选择了将应用于消息的特定策略之后，代理服务器可以按照这些策略来处理消息并且将消息转发到该消息的所期望目的地。

[0006] 另外，如本文中讨论的示例例示的，各种实施例可以支持动态策略模型，在该模型中，可以在DMZ或其他逻辑或物理子网内在消息的整个端到端处理流程中的各种不同处理点处应用不同安全策略和其他通信管理策略。可以使用这些动态策略模型框架来建立并实现各种类型的计算机网络和系统安全以及可能在通信端点内不可能或者不优选的其他通信策略。

附图说明

[0007] 图1是例示可以实现本发明的各种实施例的示例性分布式系统的组件的框图。

[0008] 图2是例示通过其可以将本发明的实施例所提供的服务作为云服务供应的系统环境的组件的框图。

[0009] 图3是例示可以实现本发明的实施例的示例性计算机系统的框图。

[0010] 图4是在高级(high-level)例示根据本发明的一个或多个实施例的包括用于对计算装置和/或系统之间的消息进行处理和传输的代理服务器的计算环境的框图。

[0011] 图5是例示根据本发明的一个或多个实施例的使用所选择的消息处理策略来接收并处理消息的过程的流程图。

[0012] 图6A和图6B是例示根据本发明的一个或多个实施例的预定消息处理流程的示例的标记语言文档。

[0013] 图7A至图7D是例示根据本发明的一个或多个实施例的与一个或多个消息处理流程内的不同点对应的消息处理策略的示例模板的标记语言文档。

[0014] 图8是例示根据本发明的一个或多个实施例的从外部客户端装置发送到内部网络服务的网络服务请求的端到端处理流程的流程图。

[0015] 图9是例示根据本发明的一个或多个实施例的从内部客户端装置发送到外部网络服务或应用的网络服务或应用请求的端到端处理流程的流程图。

具体实施方式

[0016] 在下面的描述中,出于说明的目的,阐述了众多具体细节以便提供对于本发明的各种实施例的彻底理解。然而,本领域的技术人员应该清楚的是,可以在没有这些具体细节中的一些的情况下实践本发明的实施例。在其他情形下,用框图形式示出熟知的结构和装置。

[0017] 接下来的描述只提供示例性实施例并且不旨在限制本公开的范围、适用性或配置。确切地,示例性实施例的接下来的描述将为本领域的技术人员提供能够实现示例性实施例的描述。应该理解的是,可以在不脱离所附权利要求书中阐述的本发明的精神和范围的情况下,对元件的功能和布置做出各种改变。

[0018] 在下面的描述中给出具体细节以提供对实施例的彻底理解。然而,本领域的普通技术人员应该理解的是,可以在没有这些具体细节的情况下实践实施例。例如,为了不以不必要的细节模糊实施例,可以以框图形式将电路、系统、网络、处理和其他组件示出为组件。在其他情形下,可以在没有不必要细节的情况下示出熟知的电路、处理、算法、结构和技术,以避免模糊实施例。

[0019] 另外,要注意的是,个体实施例可以被描述为过程,其被描绘为流程图、流程示图、数据流程图、结构示图或框图。虽然流程图可以将操作描述为顺序过程,但许多操作可以并行或同时地执行。另外,操作的次序可以被重排。过程在其操作完成时终止,但可以具有附图中没有包括的附加步骤。过程可以对应于方法、函数、过程、子例程(subroutine)、子程序(subprogram)等。当过程对应于函数时,该过程的终止可以对应于该函数向调用函数或主函数的返回。

[0020] 术语“计算机可读介质”包括但不限于非暂态介质(诸如,便携式或固定的存储装

置、光学存储装置和能够存储、包含或承载指令和/或数据的各种其他介质)。代码段或计算机可执行指令可以代表过程、函数、子程序、程序、例程、子例程、模块、软件包、类、或指令、数据结构或程序语句的任何组合。可以通过传递和/或接收信息、数据、变元、参数或存储器内容来将代码段耦合到另一个代码段或硬件电路。可以经由任何合适手段(包括存储器共享、消息传递、令牌传递、网络传输等)来传递、转发或传输信息、变元、参数、数据等。

[0021] 此外,可以用硬件、软件、固件、中间件、微代码、硬件描述语言或其任何组合来实现实施例。当用软件、固件、中间件或微代码来实现时,用于执行必要任务的程序代码或代码段可以被存储在机器可读介质中。处理器可以执行这些必要任务。

[0022] 本文中描述的各种技术(例如,方法、系统、存储可以由一个或多个处理器执行的多个指令的非暂态计算机可读存储存储器等)用于处理在计算机网络之间传输的消息。在一些实施例中,可以在多个计算机网络之间传输消息(诸如,针对各种类型的网络服务、应用和其他网络内容的请求和响应)。一个或多个中介装置或应用(诸如,在物理或逻辑子网内实现的代理服务器)可以对通信端点之间的消息进行接收、处理和传输。例如,代理服务器可以被配置成在内部计算机网络的子网内操作,从而将内部计算机网络的各种网络应用和/或服务暴露于外部计算机网络。

[0023] 在一些实施例中,代理服务器可以接收从内部网络中的端点传输到外部系统中的端点(或反之亦然)的消息。可以分析该消息,以确定消息的所期望目的地,和/或确定代理服务器在处理消息时应该充当正向代理还是反向代理。另外,代理服务器可以确定预定处理流程(诸如,用于处理特定消息的端到端策略模型)中的当前点。基于对消息的分析和预定处理流程中的当前点,代理服务器可以选择将应用于消息的一个或多个策略。这些策略可以包括安全策略以及其他通信管理策略,例如以认证消息、提供安全令牌仲裁和密钥管理、执行协议和有效负荷仲裁、执行基于装置的安全、支持隔离区(DMZ)威胁保护等。在选择了将应用于消息的特定策略之后,代理服务器可以按照这些策略来处理消息并且将该消息转发到该消息的所期望目的地。以下将参照附图来描述本发明的实施例的各种另外细节。

[0024] 图1是例示可以实现本发明的各种实施例的示例性分布式系统的组件的框图。在例示的实施例中,分布式系统100包括一个或多个客户端计算装置102、104、106和108,这些装置被配置成通过(一个或多个)网络110执行并操作客户端应用(诸如,网络浏览器、专有客户端(例如,Oracle Forms)等)。服务器112可以经由网络110与远程客户端计算装置102、104、106和108通信地耦合。

[0025] 在各种实施例中,服务器112可以适于运行由系统的一个或多个组件提供的一个或多个服务或软件应用。在一些实施例中,这些服务可以被作为基于网络的服务或云服务或者在软件即服务(SaaS)模型下供应到客户端计算装置102、104、106和/或108的用户。操作客户端计算装置102、104、106和/或108的用户可以进而利用一个或多个客户端应用与服务器112交互,以利用由这些组件提供的服务。

[0026] 在图中描绘的构造中,系统100的软件组件118、120和122被示出为在服务器112上实现。在其他实施例中,系统100的一个或多个组件和/或由这些组件提供的服务还可以由客户端计算装置102、104、106和/或108中的一个或多个来实现。操作客户端计算装置的用户随后可以利用一个或多个客户端应用来使用这些组件所提供的服务。可以用硬件、固件、软件或其组合来实现这些组件。应该理解的是,可能与分布式系统100不同的各种不同系统

构造是可能的。图中示出的实施例因此是用于实现实施例系统的分布式系统的一个示例并且不旨在是限制。

[0027] 客户端计算装置102、104、106和/或108可以是便携式手持装置(例如，iPhone®、蜂窝电话、iPad®、计算平板、个人数字助理(PDA))或可穿戴装置(例如，Google Glass®头戴式显示器)、运行软件(诸如Microsoft Windows Mobile®)和/或各种移动操作系统(诸如，iOS、Windows Phone、Andriod、BlackBerry 10、Palm OS等)，并且启用了互联网、电子邮件、短消息服务(SMS)、Blackberry®或其他通信协议。客户端计算装置可以是通用个人计算机，包括(举例来说)运行各种版本的Microsoft Windows®、Apple Macintosh®和/或Linux操作系统的个人计算机和/或膝上型计算机。客户端计算装置可以是运行各种商购的UNIX®或类UNIX的操作系统(包括而不限于诸如例如Google Chrome OS的各种GNU/Linux操作系统)中的任一个的工作站计算机。可替代地，或另外地，客户端计算装置102、104、106和108可以是任何其他电子装置，诸如，瘦(thin)客户端计算机、启用了互联网的游戏系统(例如，带有或不带有Kinect ®姿势输入装置的Microsoft Xbox游戏控制器)和/或能够通过(一个或多个)网络110进行通信的个人消息收发装置。

[0028] 虽然示例性的分布式系统100被示出为具有四个客户端计算装置，但可以支持任何数量的客户端计算装置。其他装置(诸如，具有传感器的装置等)可以与服务器112交互。

[0029] 分布式系统100中的(一个或多个)网络110可以是本领域的技术人员熟悉的、可以使用各种商购协议中的任一种来支持数据通信的任何类型的网络，这些协议包括而不限于TCP/IP(传输控制协议/互联网协议)、SNA(系统网络架构)、IPX(互联网分组交换)、AppleTalk等。仅仅举例来说，(一个或多个)网络110可以是局域网(LAN)(诸如，基于以太网、令牌环和/或类似物的网络)。(一个或多个)网络110可以是广域网和互联网。它可以包括虚拟网络，包括而不限于虚拟私人网络(VPN)、内联网、外联网、公共交换电话网络(PSTN)、红外网络、无线网络(例如，在电子电气工程协会(IEEE)802.11协议集、Bluetooth®和/或任何其他无线协议中的任一种下操作的网络)、和/或这些和/或其他网络的任何组合。

[0030] 可以由一个或多个通用计算机、专用服务器计算机(包括举例来说PC(个人计算机)服务器、UNIX®服务器、中程服务器、大型计算机、机架式服务器等)、服务器群组、服务器集群或任何其他适宜的布置和/或组合来构成服务器112。在各种实施例中，服务器112可以适于运行在以上公开中描述的一个或多个服务或软件应用。例如，服务器112可以对应于用于执行以上根据本公开的实施例描述的处理的服务器。

[0031] 服务器112可以运行操作系统，该操作系统包括以上讨论的操作系统中的任一个以及任何商购的服务器操作系统。服务器112还可以运行各种另外的服务器应用和/或中间层应用中的任一个，这些服务器应用和/或中间层应用包括HTTP(超文本传输协议)服务器、FTP(文件传输协议)服务器、CGI(通用网关接口)服务器、JAVA ®服务器、数据库服务器等。示例性的数据库服务器包括而不限于商购自Oracle、Miscrosoft、Sybase、IBM(国际商业机器)等的数据库服务器。

[0032] 在一些实现方式中，服务器112可以包括分析和合并从客户端计算装置102、104、106和108的用户接收的数据供给和/或事件更新的一个或多个应用。例如，数据供给和/或事件更新可以包括但不限于Twitter®供给、Facebook ®更新或从一个或多个第三方信息源接收的实时更新和连续数据流，这些连续数据流可以包括与传感器数据应用、金融股票、网络性能测量工具(例如，网络监视和流量管理应用)、点击流分析工具、汽车交通监视等有关的实时事件。服务器112还可以包括经由客户端计算装置102、104、106和108的一个或多个显示装置来显示数据供给和/或实时事件的一个或多个应用。

[0033] 分布式系统100还可以包括一个或多个数据库114和116。数据库114和116可以驻留在各种位置。举例来说，数据库114和116中的一个或多个可以驻留在服务器112本地的(和/或驻留在其中的)非暂态存储介质上。可替代地，数据库114和116可以远离服务器112并且经由基于网络的或专用的连接与服务器112通信。在实施例的一个集合中，数据库114和116可以驻留在存储区网络(SAN)中。类似地，用于执行归因于服务器112的功能的任何必要文件可以酌情本地存储在服务器112上和/或远程地存储。在实施例的一个集合中，数据库114和116可以包括适于响应于SQL格式化的命令存储、更新和获取数据的关系数据库(诸如，Oracle所提供的数据库)。

[0034] 图2是例示通过其可以将本发明的实施例所提供的服务作为云服务供应的系统环境的组件的框图。在例示实施例中，系统环境200包括可以被用户用来与提供云服务的云基础设施系统202交互的一个或多个客户端计算装置204、206和208。客户端计算装置可以被配置成操作客户端应用(诸如，网络浏览器、专属客户端应用(例如，Oracle Forms)或一些其他应用)，该客户端应用可以被客户端计算装置的用户使用来与云基础设施系统202交互以使用云基础设施系统202所提供的服务。

[0035] 应该理解的是，图中描绘的云基础设施系统202可以具有除了所描绘组件外的组件。另外，图中示出的实施例只是可以结合本发明的实施例的云基础设施系统的一个示例。在某些其他实施例中，云基础设施系统202可以具有比图中示出更多或更少的组件，可以合并两个或更多个组件，或者可以具有不同配置或布置的组件。

[0036] 客户端计算装置204、206和208可以是与以上针对102、104、106和108描述的装置类似的装置。

[0037] 虽然示例性的系统环境200被示出为具有三个客户端计算装置，但可以支持任何数量的客户端计算装置。其他装置(诸如，带有传感器的装置等)可以与云基础设施系统202交互。

[0038] (一个或多个)网络210可以促进客户端204、206和208与云基础设施系统202之间的数据通信和交换。每个网络可以是本领域的技术人员熟悉的、可以使用各种商购协议(包括以上针对(一个或多个)网络110描述的协议)中的任一种来支持数据通信的任何类型的网络。

[0039] 云基础设施系统202可以包含可以包括以上针对服务器112描述的计算机和/或服务器的一个或多个计算机和/或服务器。

[0040] 在某些实施例中，云基础设施系统所提供的服务可以包括云基础设施系统的用户按需可用的大批服务(诸如，在线数据存储和备份解决方案、基于网络的电子邮件服务、托管办公套件和文档协同服务、数据库处理、受管理技术支持服务等)。云基础设施系统所提

供的服务可以动态缩放以满足其用户的需要。云基础设施系统所提供的服务的具体实例化在本文中被称为“服务实例”。通常，来自云服务供应商系统的经由通信网络(诸如，互联网)对用户可用的任何服务被称为“云服务”。通常，在公共云环境中，构成云服务供应商系统的服务器和系统不同于顾客自己的内部部署(on-premises)服务器和系统。例如，云服务供应商系统可以托管应用，并且用户可以经由通信网络(诸如，互联网)按需订购和使用该应用。

[0041] 在一些示例中，计算机网络云架构中的服务可以包括对存储设备、托管数据库、托管网络服务器、软件应用、或云供应商提供给用户的其他服务或本领域中已知的其他方式提供的服务的受保护的计算机网络访问。例如，服务可以包括通过互联网对云上的远程存储设备的受密码保护的访问。又如，服务可以包括联网开发方自用的基于网络服务的托管关系数据库和脚本语言中间件引擎。又如，服务可以包括对托管在云供应商网址上的电子邮件软件应用的访问。

[0042] 在某些实施例中，云基础设施系统202可以包括以自服务、基于订阅(subscription)、灵活可缩放、可靠、高度可用和安全的方式交付到顾客的一套应用、中间件和数据库服务供应物。这种云基础设施系统的示例是本受让人所提供的Oracle Public Cloud。

[0043] 在各种实施例中，云基础设施系统202可以适于自动地提供、管理和跟踪顾客对由云基础设施系统202供应服务的订阅。云基础设施系统202可以经由不同的部署模型来提供云服务。例如，可以在公共云模型下提供服务，在该公共云模型中，销售云服务的组织拥有(例如，Oracle拥有的)云基础设施系统202并且服务对于一般公共或不同工业企业可用。又如，可以在私有云模型下提供服务，在该私有云模型中，云基础设施系统202单独操作用于单个组织并且可以为该组织内的一个或多个实体提供服务。还可以在社区云模型下提供云服务，在该社区云模型中，云基础设施系统202和云基础设施系统202所提供的服务被相关社区中的若干组织共享。还可以在作为两个或更多个不同模型的组合的混合云模型下提供云服务。

[0044] 在一些实施例中，云基础设施系统202所提供的服务可以包括在软件即服务(SaaS)类别、平台即服务(Paas)类别、基础设施即服务(IaaS)类别或包括混合服务的其他服务类别下提供的一个或多个服务。顾客经由订阅订单可以订购由云基础设施系统202提供的一个或多个服务。云基础设施系统202随后执行处理以提供顾客的订阅订单中的服务。

[0045] 在一些实施例中，云基础设施系统202所提供的服务可以包括而不限于应用服务、平台服务和基础设施服务。在一些示例中，云基础设施系统可以经由Saas平台来提供应用服务。Saas平台可以被配置成提供归入SaaS类别的云服务。例如，Saas平台可以提供在集成的开发和部署平台上构建并且交付一套按需应用的能力。Saas平台可以管理并且控制用于提供SaaS服务的底层软件和基础设施。通过利用由Saas平台提供的服务，顾客可以利用在云基础设施系统上执行的应用。顾客可以在不需要顾客购买单独的许可证和支持的情况下获得应用服务。可以提供各种不同的SaaS服务。示例包括而不限于为大型组织提供用于销售业绩管理、企业整合和商业灵活性的解决方案的服务。

[0046] 在一些实施例中，云基础设施系统可以经由PaaS平台提供平台服务。PaaS平台可以被配置成提供归入PaaS类别的云服务。平台服务的示例可以包括而不限于使得组织(诸如，Oracle)能够在共享共同基础设施上整合现有应用以及建立利用平台所提供的共享服

务的新应用的能力的服务。PaaS平台可以管理并控制用于提供PaaS服务的底层软件和基础设施。顾客可以在不需要顾客购买单独的许可证和支持的情况下获得云基础设施系统所提供的PaaS服务。平台服务的示例包括而不限于Oracle Java Cloud Service(JCS)、Oracle Database Cloud Service(DBCS)和其他服务。

[0047] 通过利用PaaS所提供的服务，顾客可以采用云基础设施系统所支持的编程语言和工具并且还控制所部署的服务。在一些实施例中，云基础设施系统所提供的平台服务可以包括数据库云服务、中间件云服务(例如，Oracle Fusion Middleware服务)和Java云服务。在一个实施例中，数据库云服务可以支持共享服务部署模型，该模型使得组织能够汇集数据库资源并且以数据库云的方式向顾客提供数据库即服务(Database as a Service)。在云基础设施系统中，中间件云服务可以向顾客提供开发并部署各种商业应用的平台，并且Java云服务可以向顾客提供部署Java应用的平台。

[0048] 在云基础设施系统中，可以由IaaS平台来提供各种不同的基础设施服务。这些基础设施服务促进底层计算资源(诸如，存储设备、网络和让顾客利用SaaS平台和PaaS平台所提供的服务的其他基础计算资源)的管理和控制。

[0049] 在某些实施例中，云基础设施系统202还可以包括用于提供用于向云基础设施系统的顾客提供各种服务的资源的基础设施资源230。在一个实施例中，基础设施资源230可以包括硬件(诸如，服务器、存储设备和联网资源)的预先集成和优化的组合以执行由SaaS平台和PaaS平台提供的服务。

[0050] 在一些实施例中，云基础设施系统202中的资源可以被多个用户共享并且根据需要动态地再分配。另外，可以将资源分配给在不同的时区的用户。例如，云基础设施系统230可以使第一时区中的第一组用户能够在预定小时数内利用云基础设施系统的资源，然后使相同资源能够被重新分配到位于不同时区的另一组用户，由此使资源利用最大化。

[0051] 在某些实施例中，可以提供数个内部共享服务232，内部共享服务232被云基础设施系统202的不同组件或模块以及被云基础设施系统202所提供的服务共享。这些内部共享服务可以包括而不限于安全和身份服务、集成服务、企业库(repository)服务、企业管理器服务、病毒扫描和白名单服务、高度可用性的备份与恢复服务、用于启用云支持的服务、电子邮件服务、通知服务、文件传送服务等。

[0052] 在某些实施例中，云基础设施系统202可以提供云基础设施系统中的云服务(例如，SaaS、PaaS和IaaS服务)的全面管理。在一个实施例中，云管理功能可以包括用于供应、管理和跟踪由云基础设施系统202接收的顾客订阅等的能力。

[0053] 在一个实施例中，如在图中描绘的，可以由一个或多个模块(诸如，订单管理模块220、订单编排模块222、订单供应模块224、订单管理和监控模块226以及身份管理模块228)来提供云管理功能。这些模块可以包括一个或多个计算机和/或服务器或者使用一个或多个计算机和/或服务器来提供，该计算机和/或服务器可以是通用计算机、专用服务器计算机、服务器群组、服务器集群或任何其他适宜的布置和/或组合。

[0054] 在示例性操作234中，使用客户端装置(诸如，客户端装置204、206或208)的顾客可以通过请求由云基础设施系统202提供的一个或多个服务并且对由云基础设施系统202提供的一个或多个服务的订阅下订单来与云基础设施系统202交互。在某些实施例中，顾客可以访问云用户界面(UI)(云UI 212、云UI 214和/或云UI 216)并且经由这些UI来下订阅订

单。云基础设施系统202响应于顾客下订单而接收的订单信息可以包括识别顾客和顾客期望订阅的云基础设施系统202所供应的一个或多个服务的信息。

[0055] 在顾客已经下订单之后，经由云UI 212、214和/或216来接收订单信息。

[0056] 在操作236处，订单被存储在订单数据库218中。订单数据库218可以是由云基础设施系统218操作并且连同其他系统元件操作的多个数据库中的一个。

[0057] 在操作238处，订单信息被转发到订单管理模块220。在某些情形下，订单管理模块220可以被配置成执行与订单相关的记账和核算功能(诸如，验证订单)，并且在验证通过时接纳(book)订单。

[0058] 在操作240处，关于订单的信息被传达到订单编排模块222。订单编排模块222可以利用订单信息为顾客所下的订单编排服务和资源的供应。在某些情形下，订单编排模块222可以编排资源的供应，以使用订单供应模块224的服务来支持所订阅的服务。

[0059] 在某些实施例中，订单编排模块222实现与每个订单关联的商业处理的管理并且应用商业逻辑以确定订单是否应该继续到供应。在操作242处，在接收到对新订阅的订单时，订单编排模块222向订单供应模块224发送请求以分配资源并且配置履行订阅订单所需的那些资源。订单供应模块224实现用于顾客所订购的服务的资源分配。订单供应模块224在由云基础设施系统200提供的云服务和用于供应用于提供所请求服务的资源的物理实现层之间提供抽象层。订单编排模块222因此可以与实现细节(诸如，服务和资源实际上是即时(on the fly)供应还是预先供应还是只有请求时才进行分配/指派)隔离。

[0060] 在操作244处，一旦供应了服务和资源，云基础设施系统202的订单供应模块224就可以向客户端装置204、206和/或208上的顾客发送所提供服务的通知。

[0061] 在操作246处，订单管理和监控模块226可以管理和跟踪顾客的订阅订单。在某些情形下，订单管理和监控模块226可以被配置成收集订阅订单中服务的使用统计(诸如，使用的存储设备量、传送的数据量、用户数量、系统开启时间和系统关闭时间的量)。

[0062] 在某些实施例中，云基础设施系统200可以包括身份管理模块228。身份管理模块228可以被配置成提供身份服务(诸如，云基础设施系统200中的访问管理和授权服务)。在一些实施例中，身份管理模块228可以控制关于希望利用由云基础设施系统202提供的服务的顾客的信息。这种信息可以包括认证这种顾客的身份的信息和描述授权这些顾客相对于各种系统资源(例如，文件、目录、应用、通信端口、存储器段等)执行哪些动作的信息。身份管理模块228还可以包括描述性信息的管理，该描述性信息关于每个顾客以及关于可以如何和由谁访问和修改该描述性信息。

[0063] 图3是例示可以实现本发明的实施例的示例性计算机系统的框图。系统300可以用于实现上述计算机系统中的任一个。如图中所示，计算机系统300包括经由总线子系统302与多个外围子系统通信的处理单元304。这些外围子系统可以包括处理加速单元306、I/O子系统308、存储设备子系统318和通信子系统324。存储设备子系统318包括有形计算机可读存储介质322和系统存储器310。

[0064] 总线子系统302提供用于使计算机系统300的各种组件和子系统如期望的那样彼此通信的机制。虽然总线子系统302被示意性示出为单条总线，但总线子系统的替代实施例可以利用多条总线。总线子系统302可以是包括存储器总线或存储器控制器、外围总线和使用各种总线架构中的任一种的局部总线的许多类型的总线结构中的任一种。例如，这些架

构可以包括工业标准架构(ISA)总线、微通道架构(MCA)总线、增强型ISA(EISA)总线、视频电子标准协会(VESA)局部总线和外围组件互连(PCI)总线，该PCI总线总线可以被实现为被制造成IEEE P1386.1标准的Mezzanine总线。

[0065] 可以被实现为一个或多个集成电路(例如，传统的微处理器或微控制器)的处理单元304控制计算机系统300的操作。处理单元304中可以包括一个或多个处理器。这些处理器可以包括单核或多核处理器。在某些实施例中，处理单元304可以被实现为一个或多个独立的处理单元332和/或334，在每个处理单元中包括单核或多核处理器。在其他实施例中，处理单元304还可以被实现为通过将两个双核处理器集成到单个芯片而形成的四核处理单元。

[0066] 在各种实施例中，处理单元304可以响应于程序的代码来执行各种程序并且可以维护多个同时执行的程序或处理。在任何给定时间，待执行的程序代码中的一些或全部可以驻留在(一个或多个)处理器304中和/或存储设备子系统318中。通过合适的编程，(一个或多个)处理器304可以提供上述各种功能。计算机系统300可以另外包括处理加速单元306，处理加速单元306可以包括数字信号处理器(DSP)、专用处理器和/或类似物。

[0067] I/O子系统308可以包括用户接口输入装置和用户接口输出装置。用户接口输入装置可以包括键盘、诸如鼠标或跟踪球的指点装置、结合到显示器中的触摸板或触摸屏、滚轮、点击轮、拨号盘、按钮、开关、小键盘、带有语音命令识别系统的音频输入装置、麦克风和其他类型的输入装置。用户接口输入装置可以包括例如运动感测和/或姿势识别装置(诸如，使用户能够通过使用姿势和语音命令的自然用户接口来控制输入装置(诸如，Microsoft **Xbox®** 360游戏控制器)并且与该输入装置交互的Microsoft **Kinect®** 运动传感器)。用户接口输入装置还可以包括检测来自用户的眼睛活动性(例如，在拍照片和/或进行菜单选择的同时“眨眼”)并且将眼睛姿势变换为对输入装置(例如，Google **Glass®**)的输入的眼睛姿势识别装置(诸如，Google **Glass®** 眨眼检测器)。另外，用户接口输入装置可以包括使能用户通过语音命令与语音识别系统(例如，**Siri®** 导航仪)交互的语音识别感测装置。

[0068] 用户接口输入装置还可以包括而不限于三维(3D)鼠标、游戏操纵杆或指点杆、游戏手柄和图形输入板和音频/可视装置(诸如，扬声器、数字相机、数字摄像机、便携式媒体播放器、网络摄像机、图像扫描仪、指纹扫描仪、条形码读取器3D扫描仪、3D打印机、激光测距仪和视线跟踪装置)。另外，用户接口输入装置可以包括例如医疗成像输入装置(诸如，计算机断层扫描、磁共振成像、正电子发射断层扫描、医疗超声波扫描装置)。用户接口输入装置还可以包括例如音频输入装置(诸如，MIDI键盘、数字音乐仪器等)。

[0069] 用户接口输出装置可以包括显示器子系统、指示器灯、或非可视显示器(诸如，音频输出装置)等。显示器子系统可以是阴极射线管(CRT)、平板装置(诸如，使用液晶显示器(LCD)或等离子体显示器的平板装置)、投影装置、触摸屏等。通常，使用术语“输出装置”旨在包括将来自计算机系统300的信息输出到用户或其他计算机的所有可能类型的装置和机构。例如，用户接口输出装置可以包括而不限于可视地传送文本、图形和音频/视频信息的各种显示装置(诸如，监视器、打印机、扬声器、头戴式耳机、汽车导航系统、绘图仪、语音输出装置和调制解调器)。

[0070] 计算机系统300可以包括存储设备子系统318，其包括被示出为当前位于系统存储器310内的软件元件。系统存储器310可以存储可加载到处理单元304上并且可在处理单元304上执行的程序指令以及在执行这些程序期间生成的数据。

[0071] 依赖于计算机系统300的配置和类型，系统存储器310可以是易失性的(诸如，随机存取存储器(RAM))和/或非易失性的(诸如，只读存储器(ROM)、闪存等)。RAM通常包含处理单元304立即可访问和/或目前正由操作和执行的数据和/或程序模块。在一些实现方式中，系统存储器310可以包括多种不同类型的存储器(诸如，静态随机存取存储器(SRAM)或动态随机存取存储器(DRAM))。在一些实现方式中，包含帮助诸如在启动期间在计算机系统300内的元件之间传递信息的基本例程的基本输入/输出系统(BIOS)可以通常被存储在ROM中。举例来说，而非限制，系统存储器310还例示了应用程序312(可以包括客户端应用、网络浏览器、中间层应用、关系数据库管理系统(RDBMS)等)、程序数据314和操作系统316。举例来说，操作系统316可以包括各种版本的Microsoft **Windows®**、Apple **Macintosh®**和/或Linux操作系统、各种商购的**UNIX®**或类UNIX操作系统(包括而不限于各种GNU/Linux操作系统、Google **Chrome®** OS等)和/或移动操作系统(诸如，iOS、**Windows®** Phone、**Android®** OS、**BlackBerry®** 10OS和**Palm®** OS操作系统)。

[0072] 存储设备子系统318还可以提供用于存储提供一些实施例的功能的基本编程和数据构造的有形计算机可读存储介质。在被处理器执行时提供上述功能的软件(程序、代码模块、指令)可以被存储在存储设备子系统318中。这些软件模块或指令可以由处理单元304执行。存储设备子系统318还可以提供用于存储按照本发明所使用的数据的储存库。

[0073] 存储设备子系统300还可以包括计算机可读存储介质读取器320，计算机可读存储介质读取器320还可以连接到计算机可读存储介质322。一起地并且可选地与系统存储器310组合，计算机可读存储介质322可以全面地代表远程、本地、固定和/或可移除存储装置加上用于暂时和/或更永久地包含、存储、传输和获取计算机可读信息的存储介质。

[0074] 包含代码或部分代码的计算机可读存储介质322还可以包括本领域中已知或使用的任何适宜介质，包括存储介质和通信介质(诸如但不限于用于存储和/或传输信息的任何方法或技术中实现的易失性和非易失性、可移除和不可移除的介质)。这可以包括非暂态和有形计算机可读存储介质，诸如，RAM、ROM、电可擦除可编程ROM(EEPROM)、闪存或其他存储器技术、CD-ROM、数字通用盘(DVD)、或其他光学存储设备、磁盘、磁带、磁盘存储设备或其他磁存储装置、或其他有形计算机可读介质。这还可以包括非有形计算机可读介质(诸如，数据信号、数据传输、或可以用于传输所期望信息并且可以由计算系统300访问的任何其他介质)。

[0075] 举例来说，计算机可读存储介质322可以包括读取不可移除非易失性磁介质或者向其写入的硬盘驱动器、读取可移除非易失性磁盘或者向其写入的磁盘驱动器、和读取可移除非易失性光盘(诸如，CD ROM、DVD和Blu-**Ray®**盘或其他光学介质、或其他光学介质)或者向其写入的光盘驱动器。计算机可读存储介质322可以包括但不限于**Zip®**驱动、闪存卡、通用串行总线(USB)闪存驱动器、安全数字(SD)卡、DVD盘、数字视频带等。计算机可读存储介质322还可以包括基于非易失性存储器的固态驱动(SSD)(诸如，基于闪存的SSD、商用

闪存驱动、固态ROM等)、基于易失性存储器的SSD(诸如,固态RAM、动态RAM、静态RAM、基于DRAM的SSD、磁阻RAM(MRAM)SSD)和使用DRAM和基于闪存的SSD的组合的混合SSD。盘驱动器及其关联的计算机可读介质可以提供用于计算机系统300的计算机可读指令、数据结构、程序模块和其他数据的非易失性存储。

[0076] 通信子系统324提供到其他计算机系统和网络的接口。通信子系统324用作从其他系统接收数据并且将数据从计算机系统300传输到其他系统的接口。例如,通信子系统324可以使计算机系统300能够经由互联网连接到一个或多个装置。在一些实施例中,通信子系统324可以包括射频(RF)收发器组件、全球定位系统(GPS)接收器组件和/或其他组件,该RF收发器组件用于访问无线语音和/或数据网络(例如,使用蜂窝电话技术、高级数据网络技术,诸如3G、4G或EDGE(全球演进的增强数据速率)、WiFi(IEEE 802.11系列标准、或其他移动通信技术、或其任何组合))。在一些实施例中,通信子系统324可以提供有线网络连接(例如,以太网)作为无线接口的补充或替代。

[0077] 在一些实施例中,通信子系统324还可以代表可以使用计算机系统300的一个或多个用户接收结构化和/或非结构化格式的数据供给326、事件流328、事件更新330等的形式的输入通信。

[0078] 举例来说,通信子系统324可以被配置成从社交网络和/或其他通信服务的用户实时地接收数据供给326,数据供给326诸如**Twitter®**供给、**Facebook®**更新、诸如丰富站点摘要(RRS)供给的网络供给、和/或来自一个或多个第三方信息源的实时更新。

[0079] 另外,通信子系统324还可以被配置成接收连续数据流形式的数据,该连续数据流可以包括实时事件的事件流328和/或事件更新330,该连续数据流形式的数据可以是本质上连续的或者无边界的而没有明确结束。生成连续数据的应用的示例可以包括例如传感器数据应用、金融股票、网络性能测量工具(例如,网络监视和流量管理应用)、点击流分析工具、汽车交通监视等。

[0080] 通信子系统324还可以被配置成向一个或多个数据库输出结构化和/或非结构化的数据供给326、事件流328、事件更新330等,这些数据库可以与耦合到计算机系统300的一个或多个流数据源计算机通信。

[0081] 计算机系统300可以是包括手持便携式装置(例如,**iPhone®**蜂窝电话、**iPad ®**计算平板、PDA)、可穿戴装置(例如,Google**Glass ®**头戴式显示器)、PC、工作站、主机、自助式服务机、服务器机架、或任何其他数据处理系统的各种类型中的一种。

[0082] 由于计算机和网络不断变化的性质,对图中描绘的计算机系统300的描述仅仅是旨在作为特定示例。比图中描绘的系统更多或更少的组件的许多其他配置是可能的。例如,还可以使用定制的硬件和/或可以在硬件、固件、软件(包括小应用程序)或组合来实现特定元件。另外,可以采用与其他计算装置(诸如,网络输入/输出装置)的连接。基于本文中提供的公开和教导,本领域的普通技术人员将理解用于实现各种实施例的其他方式和/或方法。

[0083] 如以上介绍的,本发明的实施例提供了用于处理在计算机网络之间传输的消息的技术。更具体地,某些实施例提供了用于在多个计算机网络之间传输消息(诸如,对用于各种类型的网络服务、应用和其他网络内容的请求和响应)的技术。一个或多个中间装置或应用(诸如,在物理或逻辑子网内实现的代理服务器)可以对通信端点之间的消息进行接收、

处理和传输。在一些实施例中，代理服务器可以接收从内部网络中的端点传输到外部网络的端点(或反之亦然)的消息。可以分析该消息，以便确定消息的所期望目的地，和/或确定代理服务器在处理消息时应该充当正向代理还是反向代理。代理服务器可以确定针对消息的预定处理流程(诸如，用于处理特定消息的端到端策略模型)中的当前点。基于对消息的分析和预定处理流程中的当前点，代理服务器可以选择将应用于消息的一个或多个策略。这种策略可以包括安全策略以及其他通信管理策略，例如以认证消息、提供安全令牌仲裁和密钥管理、执行协议和有效负荷仲裁、执行基于装置的安全、支持隔离区(DMZ)威胁保护等。在选择了将应用于消息的特定策略之后，代理服务器可以按照这些策略来处理消息并且将消息转发到消息的所期望的目的地。

[0084] 图4是例示包括用于处理和传输各种计算机网络中的装置和/或系统之间的消息的代理服务器420的计算环境400的组件的框图。在这一示例中例示的计算环境400可以对应于高级计算机架构，该高级计算机架构被设计成为各种客户端装置提供对计算资源(诸如，网络应用和网络服务)的访问。在各种实施例中，计算环境400的范围可以从小且简单的计算系统到包括被设计成与其他这些系统集成以支持各种组织的计算需要的硬件、软件和网络组件的大且高度复杂的系统。计算环境400可以被实现为多层计算机架构，该多层计算机架构可以包括基于网络和/或基于云的实现方式，并且其中各种端点装置(例如，用户装置410、网络应用或网络服务供应商430等)经由一个或多个中间层系统进行交互。另外，计算环境400中示出的每个组件可以被实现为包括硬件、软件和/或网络组件的各种组合的个体计算机系统。在其他情况下，计算环境400中示出的多个组件可以被实现为与组合的计算机系统一起进行操作的逻辑子组件(例如，在计算机可读介质上实施的软件应用等)。

[0085] 如图4中所示，计算环境400可以对应于客户端服务系统，在该客户端服务系统中，客户端装置410可以经由各种(一个或多个)计算机网络415、(一个或多个)防火墙435、代理服务器420、和/或其他中介装置将请求传输到一个或多个后端网络应用或网络服务430。网络应用或服务430可以包括各种系统430所暴露的任何应用编程接口(API)、服务、应用和任何其他信息资产，包括但不限于简单对象访问协议(SOAP)网络服务或API、表征状态转移(REST)网络服务或API、和/或经由超本文传输协议(HTTP)或HTTP安全协议暴露的网络内容。在这些情况下，代理服务器420可以充当在客户端装置410和后端服务/应用430之间提供安全层的反向代理服务器。当充当反向代理时，代理服务器420可以提供用于后端服务/应用430的中央接入点连同与后端服务/应用430关联的各种安全和管理策略的服务虚拟化和实施。当充当反向代理时，代理服务器420可以在虚拟化和模糊后端服务/应用430的同时暴露这些后端服务/应用430。例如，代理服务器420可以只暴露虚拟统一资源定位符(URL)，使得不可信任网络上的客户端装置410可能无法看到或者知悉底层的后端网络服务/应用430。

[0086] 另外地或可替代地，计算环境400可以对应于用于在相反方向上传输的请求-响应的客户端-服务器系统。例如，在与网络服务/应用430相同的内部计算机网络460内操作的客户端装置440可以向超出代理服务器420和(一个或多个)防火墙435的各种外部计算机系统和网络上运行的网络服务或应用450传输请求。在这些情况下，代理服务器420可以充当正向代理服务器，从而在内部网络460内的客户端装置440和外部网络上的后端服务/应用450之间提供安全层。与反向代理操作类似，正向代理操作中的通信可以包括对SOAP网络服

务、REST网络服务、HTTP/HTTPS网络内容等的请求和来自SOAP网络服务、REST网络服务、HTTP/HTTPS网络内容等的响应。当代理服务器420正作为正向代理服务器操作时，内部网络内的客户端装置440可以知道后端服务/应用450，并且这些服务/应用450可以从客户端方配置的代理服务器420接收直接传输。在这种情况下，代理服务器420可以使用任何安全或通信管理策略来为正向代理统一资源标识符(URI)端点提供安全。

[0087] 在正向代理模式或反向代理模式下，代理服务器420可以支持各种安全和认证特征，诸如，基于Kerberos Kinit的认证、基于Kerberos Pkinit的认证、基于授权协议版本2.0的开放标准(OAuth2)的认证、基于TLP的认证，使用简单和受保护GSSAPI协商机制(SPNEGO)令牌、WINDOWS NT LAN管理器(NTLM)令牌、安全确认标记语言(SAML)令牌等来创建后端服务的会话令牌和/或基于挑战的认证。

[0088] 客户端装置410和客户端装置440可以包括台式计算机或膝上型计算机、移动装置和其他各种计算装置/系统，包括以上在图1至图3中的例示计算系统中讨论的硬件、软件和联网组件中的一些或全部。在一些实施例中，客户端装置410和客户端装置440可以包括被配置成请求并且接收来自后端网络服务/应用430和后端网络服务/应用450的数据的一个或多个客户端软件应用(例如，网络浏览器)。客户端装置410和客户端装置440还可以包括必要的硬件和软件组件以建立网络接口、安全和认证能力和内容高速缓存能力，以接收现场内容并且将它实时(或几乎实时)提供给用户。

[0089] 通信网络415可以包括本文中描述的计算机网络和其他通信网络的任何组合。例如，网络415可以包括TCP/IP(传输控制协议/互联网协议)网络(诸如，局域网(LAN)、广域网(WAN)(例如，互联网)和各种无线电信网络)。另外，应该理解的是，通信网络415可以代表将客户端装置410与后端应用/服务430分开的许多不同的物理和逻辑网络的组合。在一个或多个防火墙435之外，各种服务器(诸如，网络服务器、认证服务器)和/或专用联网组件(诸如，防火墙、路由器、网关、负载均衡器等)也可以促进客户端装置410和后端服务/应用430之间的通信。

[0090] 如以下讨论的，代理服务器420可以实现为隔离的计算机系统(例如，代理计算机服务器)或实现为包括专用硬件、软件和网络组件的计算机多个计算系统的组合。可替代地或另外地，代理服务器420可以是在网络装置(例如，网络服务器或防火墙)或者是在可信任网络460内的计算机服务器内执行的代理服务器软件应用。因此，代理服务器420可以驻留在内部计算机网络460的物理子网或逻辑子网465内，并且在任一种情况下，可以充当可信任内部网络上的客户端/服务器和不可信任外部网络上的客户端/服务器之间的中介。另外，代理服务器420内的组件421-428中的每个可以被实现为被配置成与代理服务器420通信的单独计算系统，或者可以作为集成在与代理服务器420相同的计算机服务器内的逻辑子组件进行操作。在任一种情况下，可以使用专用硬件、软件、网络和存储器子系统来实现每个组件421-428以执行本文中描述的技术。

[0091] 在这一示例中，代理服务器420包括被配置成经由通信网络415和/或防火墙435从外部客户端装置410接收消息的负载均衡器422。在一些实施例中，负载均衡器422可以是从任何外部网络到后端服务/应用430的所有TCP、UDP、HTTP和HTTPS流量的进入点。负载均衡器422还可以被配置成与后端服务器通信，并且配置成向客户端装置410传输响应。在接收和解析消息之后，负载均衡器422可以向适宜的网络服务框架(例如，经由Java本地接口

(JNI)或.NET编程框架等)传输消息。例如,在代理服务器420处接收到的来自客户端装置的SOAP请求可以被传输到SOAP网络服务框架421,并且REST请求可以被传输到REST网络服务框架423。可以通过解析请求并且将其传输到各种组件(诸如,URL虚拟化组件或服务)类似地处置网络内容请求。这些网络服务和组件还可以被配置成执行协议转换(诸如,SOAP到REST和REST到SOP消息转换以及JavaScript对象表示法(JSON)到XML或JSON到SOAP,反之亦然)。

[0092] 消息节流系统(或消息节流子组件)424可以被配置成监视从客户端装置410和/或后端服务/应用430接收的网络流量。消息节流系统424可以具有用于特定客户端装置410和/或特定网络服务或应用430的可配置的消息速率限制。消息节流系统424可以使用现有策略以允许特定量的消息到达/来自指定的客户端410、或到达/来自指定的网络服务/应用430。当消息的数量超过消息速率限制时,那么消息节流系统424可以被配置成执行动作(诸如,发送警报、记录日志、或挂起今后的消息传输)。

[0093] 代理服务器420还可以包括被配置成实现代理服务器420内的各种安全策略的各种安全系统或组件。在这一示例中,代理服务器420包括密钥管理系统425、令牌仲裁系统426、和认证和授权系统427。代理服务器420内的这些系统和安全组件可以认证来自客户端装置410的消息,提供安全令牌仲裁,执行API密钥管理,执行细分授权和/或数据校订,支持保密性和完整性,执行基于风险的认证,执行用于移动终端装置410的基于装置的安全,支持隔离区(DMZ)威胁保护,执行协议和有效负荷仲裁等。例如,负载均衡器422和/或认证/授权系统427可以包括子系统,以用于阻止拒绝服务(Dos)攻击,检测和过滤残缺消息,检测并且阻止SQL、JavaScript、和/或XPath/Xquery注入攻击,执行消息验证以针对恶意内容进行保护(例如,检测消息附件内的病毒,验证XML和JSON数据结构,验证形式参数和查询参数等)。令牌仲裁系统426可以被配置成转换指定的客户端装置410和后端网络服务/应用430之间的认证令牌。安全系统424-427还可以支持编排,并且通过去除操作,例如通过聚合多个后端API或服务并且执行自动中介或组成。

[0094] 另外,在这一示例中,代理服务器420包括消息处理策略428的数据存储区。可以将消息处理策略存储在各种形式的计算机可读介质(诸如,XML、JavaScript或其他类型的可执行软件组件)中。如以下更详细讨论的,消息处理策略428可以用于实施代理服务器420内的安全策略和其他通信管理策略。数据存储区428可以包括个体消息处理策略,该个体消息处理策略可以在用于个体消息的端对端处理流程期间的各个阶段中被取回并且应用于个体消息。如在这一示例中示出的,消息处理策略数据存储区428可以驻留在代理服务器420中,或者可以驻留在可信任的内部计算机网络的后端服务器或安全的第三方服务器等内。

[0095] 如图4中所示,可以在两个或更多个计算机网络之间(例如,在提供网络应用/服务430的第一可信任内部网络和各种不可信任的客户端装置410可以通过其访问内部网络应用/服务430的第二不可信任的外部网络415(例如,互联网)之间)的中介网络装置内实现代理服务器420。在一些实施例中,代理服务器420可以在内部计算机网络的子网内操作,以便提供用于内部计算机网络的初始的安全和通信管理层。例如,安全内部网络460可以包括多个网络服务/应用430,连同各种其他服务器和客户端装置440。代理服务器420和/或附加装置可以是相同内部网络460的部分,但是可以在内部计算机网络的物理子网465内操作,通过防火墙435b与内部计算机网络分开。在一些示例中,代理服务器420可以被实现为在内部

计算机网络460的逻辑子网465(而非物理子网)内执行的代理服务器应用。因此,代理服务器420可以驻留在与防火墙435b和/或后端网络服务/应用430中的一个或多个相同的计算系统上。

[0096] 另外,在一些实施例中,代理服务器420可以在可信任的内部网络460和不可信任的外部网络之间的隔离区(DMZ)网络内操作。DMZ可以被实现为物理子网465,其提供第一层安全和通信管理,与在客户端装置410和客户端装置440和后端网络服务/应用430和后端网络服务/应用450处提供的端点分开。如图4中所示,可以在两个防火墙435a和435b之间实现DMZ。在其他实施例中,可以使用单个防火墙,或者使用将子网465与可信任的内部网络460和不可信任的外部网络二者物理或逻辑分开的其他各种配置的网络装置来实现DMZ。DMZ内的所有计算机服务器和其他装置(诸如,代理服务器420)可以具有到内部网络460内的装置的特定子集(例如,网络应用/服务器430)的受限连接。可以基于特定的主机、端口、协议等来限制这种连接。类似地,当与任何外部不可信任的网络(例如,网络415和装置410)通信时,可以在DMZ内的装置上实施受限连接的策略。在DMZ内操作代理服务器420之外,在某些实施例中,后端网络服务器/应用430中的一个或多个可以在DMZ内进行操作。例如,更容易或更倾向于受到来自外部系统(例如,网络服务器、电子邮件服务器、域名系统(DNS)服务器等)攻击的某些服务器可以被移动到具有代理服务器420的DMZ中。

[0097] 现在参照图5,示出例示使用所选择的消息处理策略来接收和处理消息的过程的流程图。如以下描述的,可以通过计算环境400中的一个或多个组件(诸如,代理服务器420和本文中实现的各种子系统/子组件)来执行该过程中的步骤。另外,在一些实施例中,该过程中的某些步骤可以在客户端装置410、后端网络服务/应用430内执行,和/或通过其他各种中介装置来执行。还应该理解的是,本文中描述的技术(包括接收和分析消息、选择消息处理策略和处理消息)不需要限于上述的特定系统和硬件实现方式,而是可以在包括硬件、软件和网络组件的其他组合的其他硬件和系统环境内执行。

[0098] 在步骤501中,可以由中介计算系统或应用(诸如,代理服务器420)来接收网络消息。如上所述,代理服务器420可以被实现为在可信任的内部网络460和一个或多个不可信任的外部网络之间的中介服务器装置和/或应用。步骤501中接收到的网络消息可能不旨在用于代理服务器420。相反,代理服务器420可以拦截由第一端点装置(例如,客户端装置410)传输的并且旨在用于第二端点装置(例如,托管后端网络服务和/或应用430的计算机服务器)或反之亦然的消息。

[0099] 在一些实施例中,进入或离开内部网络460的所有网络流量可以通过代理服务器420进行路由。在其他情况下,代理服务器420可以被配置成拦截特定类型或协议的网络消息,例如,来自客户端装置410和客户端装置440的对于SOAP、REST或URL资源的HTTP请求以及来自SOAP、REST或URL网络服务/应用430和450回到客户端装置的HTTP响应。因此,步骤501中接收到的网络消息可以是例如而不限于TCP消息、HTTP或HTTPS消息、简单邮件传输协议(SMTP)、用户数据报协议(UDP)消息、和/或Java消息服务(JMS)消息。在一些情况下,网络消息可以对应于从客户端装置410到托管网络服务/应用430的后端计算机服务器的SOAP、REST或网络内容的请求,或者对应于后端网络服务或应用430对来自客户端装置410的SOAP、REST或网络内容请求的响应。另外,网络消息可以对应于从在内部计算机网络460内操作的客户端装置440到提供在外部计算机网络上操作的网络服务/应用的计算机服务器

450的SOAP、REST或网络内容请求，或者对应于来自外部网络服务或应用450对来自内部客户端装置440的SOAP、REST或网络内容请求的响应。

[0100] 在步骤502中，代理服务器420可以分析步骤501中接收的网络消息，以确定消息的所期望目的地，并且还确定在处理网络消息时代理服务器420应该充当正向代理(即，正向代理模式)还是反向代理(即，反向代理模式)。如本文中使用的，网络消息的“所期望目的地”可以是指由传输装置或传输装置的用户指定的消息的目的地。可以通过解析并且分析消息报头和/或消息主体的部分来确定消息的所期望目的地。例如，消息的统一资源标识符(URI)或消息主体内的网络服务或应用的标识符和/或操作标识符可以对应于由内部网络460提供的网络服务/应用或网络内容。在这一示例中，代理服务器420可以基于消息报头和内容来确定消息旨在用于内部网络460内的特定服务器。在另一个示例中，如果消息URI对应于不可信任的网络上的远程服务器，则代理服务器420可以确定消息的所期望目的地是远程服务器，而非内部网络460内的装置。消息内的识别消息的传输方(诸如，源IP地址或主机名称标识符)的信息也可以用于确定消息的所期望目的地。

[0101] 除了确定消息的所期望目的地，代理服务器420可以确定消息是来自客户端装置410或440的请求的部分还是来自网络服务/应用服务器装置430或450的响应的部分，以便确定在处理消息时代理服务器420应该以正向代理模式还是反向代理模式操作。例如，如果接收到的消息是从客户端装置410到网络服务或应用430的请求，则所期望目的地在可信任的内部网络460内并且代理服务器420应该以反向代理模式操作。相反，如果接收到的消息是从内部客户端装置440到外部网络服务450、网络应用450或ULR 450的请求，则所期望目的地在可信任的内部网络之外并且代理服务器420应该以正向代理模式操作。

[0102] 在其他情况下，步骤501中接收到的消息可能不是来自客户端装置410或440的请求，而是可以是来自网络服务器430或450的对于之前请求的响应。例如，如果接收到的消息是来自可信任的内部网络460内的网络服务/应用430或其他服务器对来自客户端装置410的请求的响应，则原始请求的所期望目的地在可信任的内部网络460内并且代理服务器420应该以反向代理模式操作。相反，如果接收到的消息是来自内部网络460外的网络服务/应用450或其他服务器对来自客户端装置440的请求的响应，则原始请求的所期望目的地在内部网络460之外并且代理服务器420应该以正向代理模式操作。

[0103] 在步骤503中，代理服务器420可以确定针对步骤501中接收到的消息的预定处理流程中的当前点。消息处理流程可以是指将由代理服务器420执行的端到端消息处理流程，该流程以由代理服务器420从客户端装置410或440接收消息开始，并且以由代理服务器420向客户端装置410或440传输响应结束。如以下讨论的，确定针对消息的预定处理流中的当前点可以包括识别与消息关联的策略模型，以及确定处理模型内的当前处理位置。

[0104] 在一些实施例中，可以通过策略模型来限定针对消息的预定消息处理流程。策略模型可以包括限定策略集合(例如，安全策略、通信管理策略等)的数据，该策略集合可以被代理服务器420应用以在消息的端到端消息处理流程期间的各个点处理消息。限定消息的端到端处理流程的策略模型和个体消息处理策略这两者可以是各种形式的计算机可读介质(诸如，XML、JavaScript或其他类型的可执行软件组件)。策略模型和/或消息处理策略可以被存储在代理服务器420内，例如存储在数据存储区428内或内部网络460内的其他地方。

[0105] 如上所述，策略模型可以限定代理服务器420在消息的端到端处理流程中的各个

点可以应用于消息的消息处理策略集合。在一些实施例中，在步骤503中，代理服务器420可以依赖于步骤501中接收到的消息的特性来应用不同的策略模型。例如，代理服务器420所取回并应用的特定策略模型可以依赖于步骤502中执行的消息的所期望目的地和正向或反向代理模式的确定。另外，代理服务器420所取回并应用的策略模型可以依赖于用于传输消息的网络协议和/或消息的请求类型或客户端类型。例如，对于REST请求、SOAP请求、网络内容(URL)请求等可以使用不同的策略模型。

[0106] 简要参照图6A和图6B，示出均以XML实现的策略模型的两个示例。图6A示出用于虚拟应用的示例策略模型。因此，可以取回示例的策略模型600a并且将其用于反向代理使用情况的消息处理。相反，图6B示出用于代理应用的示例策略模型，并且因此可以取回示例的策略模型600b并且将其用于正向代理使用情况下的消息处理。如每个示例中示出的，策略模型可以包括处理流程内的各个点(也可以被称为“断言”)的标签或标识符，以及用于针对处理点/断言中的每个的一个或多个策略标识符。例如，示例策略模型600a识别在接收到请求时将执行的两种策略(在“on-request”标签内)、执行消息变换的策略(在“message-transformation”标签内)、和在调用后端网络服务时将执行的策略(在“invoke”标签内)。示例的策略模型600b识别在接收到请求时将执行的策略(在“on-request”标签内)和在调用后端网络服务时执行的策略(“invoke-proxy”标签内)。

[0107] 在一些实施例中，代理服务器420可以对于代理应用(即，在正向代理模式下)在服务级(或URL级)应用策略，而对于虚拟应用(即，在反向代理模式下)代理服务器420可以在服务级和/或操作级(或方法级)应用策略。因此，当调用可信任的内部网络460内的后端网络服务/应用430时，代理服务器420可以在它可以实施在策略模型内识别的策略之前首先确定操作(针对SOAP)或方法(针对REST和URL)。

[0108] 在识别与步骤501中接收到的消息关联的策略模型(或限定处理流程的其他数据)之后，代理服务器420可以按照策略或处理流程来确定消息处理中的当前点。可以通过消息本身的特性以及基于关于消息的早前处理的之前存储的数据来确定消息处理流程中的当前点。如上所述，预定处理流程可以针对消息应用端到端处理，从由客户端装置410或440的初始请求，到向客户端装置410或440传输回的响应。因此，确定步骤501中接收到的消息是来自客户端装置的初始请求、来自客户端装置的另外的数据传输(例如，与请求相关的认证凭证或另外的数据)、来自后端网络服务/应用的响应、还是来自后端服务器或装置另外的数据传输(例如，来自单点登陆或令牌翻译服务的数据)可以允许代理服务器420确定端到端消息处理流程内消息处理的当前点。另外，代理服务器420可以存储与对该消息或其他相关消息执行的之前处理相关的数据(诸如，之前消息变换的结果、服务的调用、所遭遇的处理错误)，以便确定代理服务器420应该向消息应用的下一个消息处理策略。

[0109] 以下的段落包括策略模型或其他消息处理流程内的可能点(也可以被称为“断言”)的许多示例，在这些点处可以应用消息处理策略。应该理解的是，这些示例只是例示性的，并且不需要是排他性的列表。此外，在各种其他实施例中，可以改变本文中描述的断言名称(例如，OnRequest、OnInvoke、OnResponse、OnError、MessageTransformation等)以及用于断言和策略的XML结构和标签名称。

[0110] 步骤503中确定策略模型或其他预定消息处理流程内的当前点的第一示例可以包括确定步骤501中接收到的消息对应于来自外部计算机网络中的客户端装置410的请求。在

消息的端到端处理流程开始处的这一点可以被称为“OnRequest”断言或类似物。如以下更详细讨论的，OnRequest断言可以包括对可以被应用以便保护虚拟服务、代理服务和/或网络应用的策略的引用。例如，OnRequest断言可以包括代表代理服务器420应该针对从客户端装置410接收的新网络服务/应用/内容请求而实施的安全策略的URL或其他标识符。OnRequest断言还可以指的是其他策略和/或可以包含其他断言。在一些情况下，OnRequest断言可以只以反向代理模式操作，也就是说，可以只处置来自外部客户端装置410对于内部网络资源430的请求。在这种情况下，可以通过可以应用不同消息处理策略的不同断言来处置来自内部客户端装置440对于外部网络资源450的请求。

[0111] 步骤503中可以发生的当前消息处理点的另一个确定可以包括在从外部客户端装置410接收到请求之后，确定代理服务器420应该向内部计算机网络460中的后端网络应用或网络服务430传输请求。消息的端到端处理流程内的这一点可以被称为“OnInvoke”断言或类似物。与OnRequest断言类似，在一些实施例中，OnInvoke断言可以只在反向代理使用情况中应用，在该反向代理使用情况中，从外部客户端装置410接收初始请求以调用内部网络460内的后端网络服务/应用430。OnInvoke断言可以包括代表代理服务器420在端到端处理流程中的该点期间应该实施的策略的URI或其他标识符。多个策略标识符(或引用)可以例如通过使用多个XML“PolicyURI”XML元素而包括在OnInvoke断言内。另外，OnInvoke断言可以使用客户端的资源模式来唯一地识别客户端细节。可以由代理服务器420在运行时基于OnInvoke断言内配置的值确定用于OnInvoke断言的客户端类型(例如，REST客户端、SOAP客户端、URL/网络客户端等)。OnInvoke断言还可以是指其他策略和/或可以包含其他断言。

[0112] 确定当前消息处理点的另一个示例可以包括在从外部客户端装置410接收到请求之后以及在调用后端网络服务/应用430之后，确定代理服务器420应该向外部客户端装置410传输响应。消息的端到端处理流程内的这一点可以被称为“OnResponse”断言或类似物。与OnRequest和OnInvoke断言类似，在一些实施例中，OnResponse断言可以只在反向代理使用情况中应用，在该代理使用情况中，从外部客户端装置410接收初始请求以调用内部网络460内的后端网络服务/应用430。OnResponse断言可以包括代表代理服务器420在端到端处理流程中的这一点期间应该实施的策略的URI或其他标识符。多个策略标识符(或引用)可以包括在OnResponse内，并且OnResponse断言还可以是指其他策略和/或可以包含其他断言。

[0113] 步骤503中可以发生的当前消息处理点的另一个确定可以包括确定代理服务器420应该向外部网络服务或应用450传输来自内部客户端装置440的请求。消息的端到端处理流程内的这一点可以被称为“OnProxyInvoke”断言或类似物。不同于以上讨论的“OnInvoke”示例断言，OnProxyInvoke断言可以只在正向代理使用情况中应用，在该正向代理使用情况中，从内部客户端装置440接收初始请求以调用不可信任的外部网络内的后端网络服务/应用450。OnProxyInvoke断言可以包括代表代理服务器420在端到端处理流程中的这一点期间应该实施的策略的URI或其他标识符。多个策略标识符(或引用)可以例如通过使用多个XML“PolicyURI”XML元素而包括在OnProxyInvoke断言内。例如，可以由代理服务器420在运行时基于运行时间变元确定用于OnProxyInvoke断言的客户端类型(例如，REST客户端、SOAP客户端、URL/网络客户端等)。OnProxyInvoke断言还可以是指其他策略和/或可以包含其他断言。

[0114] 确定当前消息处理点的另一个示例可以包括在端到端处理流程期间的某一点确定代理服务器420应该将消息从一种消息类型变换成另一种类型。消息的端到端处理流程内的这一点可以被称为“MessageTransformation”断言或类似物。例如，代理服务器420可以接收具有第一消息类型的消息(例如，REST请求)，并且可以分析该消息以确定该消息是旨在用于只接受第二消息类型的后端服务或应用(例如，后端SOAP服务)。在进行此确定之后，代理服务器420可以对消息执行适宜的MessageTransformation断言，之后向所期望目的地传输变换后的消息。代理服务器420可以支持的变换策略的示例可以包括而不限于XML到JavaScript对象表示(JSON)和JSON到XML策略、XML到SOAP和SOAP到XML策略、和JSON到SOAP和SOAP到JSON策略。在各种实施例中可以支持其他熟知介质类型之间的变换。代理服务器420可以在进行后端服务虚拟化时自动地附连适宜的变换策略，并且可以使用安装在代理服务器420处或计算环境400中其他地方的一个或多个翻译框架来执行变换。在一些实施例中，MessageTransformation断言可以只在反向代理模式中操作，也就是说，可以只对于来自外部客户端装置410对内部网络资源430的请求的转换和用于返回客户端装置410的响应支持该断言。在其他实施例中，对于正向代理和反向代理使用情况二者可以支持MessageTransformation断言。

[0115] 确定当前消息处理点的另一个示例可以包括确定已经在针对消息的端到端处理流程期间的某一点处发生了错误。消息的端到端处理流程内的这一点可以被称为“OnError”断言或类似物。触发针对消息的OnError断言(例如，触发与消息关联的OnError断言中识别的一个或多个策略的执行)的错误可以是代理服务器420完成的处理内发生的错误和/或代理服务器420从后端计算机服务器或装置接收的错误。例如，代理服务器420可以从消息的处理流程期间调用的后端计算机服务器(诸如，授权服务、令牌翻译服务、或后端网络服务/应用430或450)接收错误指示。另外，代理服务器420可以在执行消息处理任务时识别或产生错误。触发OnError断言中的策略的消息可以是在代理服务器420完成的消息处理内发生的错误(诸如，解析或验证消息时的错误、或当执行消息变换策略时的错误)。因此，不同于处理流程内的可以应用特定消息处理策略的点(也被称为“断言”)的之前的一些示例，OnError断言可以是条件性的。也就是说，在消息的端到端处理流程期间，代理服务器420可以依赖于在该处理期间可能发生的错误的数量和类型，可以将来自OnError断言的策略应用一次、多个次或根本不应用。在各种不同实施例中，OnError断言可以在正向代理使用情况、反向代理使用情况、或这二者中应用。

[0116] 在步骤504中，可以由代理服务器420选择和取回用于处理步骤501中接收到的消息的一个或多个特定策略。如以上讨论的，由代理服务器420选择并且应用于消息的特定策略可以包括安全策略以及任何其他类型的通信管理策略。例如而不限于，这些策略可以执行与认证、授权、审计、单点登陆、安全策略实施、密码管理和分配、安全通信、安全数据存储和安全数据共享等等相关的功能。

[0117] 在步骤504中，策略可以由代理服务器420如下选择：首先取回与消息关联的(一个或多个)端到端处理流程(例如，策略模型)，并且随后使用步骤503中确定的端到端处理流程内的当前点(例如，断言)以识别将在端到端流程中的当前点处应用于消息的特定策略。例如，如果步骤501中接收到的消息是来自外部客户端装置410对于网络服务/应用430的请求，并且如果示例的策略模型600a用于控制这些消息的端到端处理，则代理服务器420可以

取回在策略模型600a的“on-request”标签内识别的任何策略。在此情形下，在策略模型600a的“on-request”标签内找到两个策略标识符，每个标识符被包含在“PolicayReference URI”标签内。因此，在这一示例中，代理服务器420可以在步骤504中选择这两个策略用于在步骤505中处理消息。

[0118] 又如，如果步骤501中接收到的消息是来自内部客户端装置440的访问外部网络服务/应用450的请求，并且如果示例策略模型600b用于控制这些消息的端到端处理，则代理服务器420可以取回策略模型600b的“on-request”标签内识别的策略。可替代地，如果已经应用了“on-request”策略并且代理服务器420准备向外部网络服务/应用450传输请求，则代理服务器420可以取回策略模型600b的“invoke-proxy”标签内识别的策略。

[0119] 在步骤505中，代理服务器420可以使用在步骤504中选择的策略来处理消息。如以上讨论的，代理服务器420可以通过从针对消息的预定端到端处理流程中识别URI或其他策略标识符来确定将应用于消息的适宜策略。在示例策略模型600a和600b中，可以在与端到端处理流程中的当前点对应的断言的“PolicyReferenceURI”标签内找到将应用的策略的URI。这些策略的URI可以引用策略的存储位置。在其他示例中，策略标识符不需要被表示为URI，但是可以包括其他标识数据(诸如，API或服务标识符、函数名称、方法名称和/或操作名称等)。在任何情况下，策略标识符可以识别用于消息处理策略的存储位置或者其他访问信息。这些策略本身可以被存储在各种形式的计算机可读介质(诸如，XML、JavaScript、或其他类型的可执行软件组件)中。

[0120] 消息处理策略可以被存储在位于计算环境400内的各种不同服务器或装置中的数据存储区(诸如，数据库和/或基于文件的存储系统)中。例如，某些策略(诸如，消息变换策略、消息节流策略、负载均衡策略和可能相对不变并且没有安全数据的其他策略)可以被本地存储在代理服务器420内(例如，消息处理策略数据存储区428内)。其他策略(诸如，用户认证/授权策略和可能频繁改变或者可能包括安全数据的其他策略)可以被存储在可信任的内部计算机网络460的安全服务器或存储系统内。在其他情况下，某些策略可以被存储在外部网络中的安全第三方服务器或客户端装置410上。代理服务器420可以被配置成在步骤505中从这各个位置中的任何一个取回策略并且应用策略。

[0121] 在步骤506中，在步骤505中使用各种安全策略和/或其他通信管理策略处理消息之后，代理服务器420可以向消息的所期望目的地传输处理后的消息。如之前讨论的，在步骤502中可以通过解析并且分析消息报头和/或消息主体的部分来确定所期望目的地。消息(诸如，对网络服务/应用430的请求、或对内部客户端装置440的响应或其他传输)的所期望目的地可以在内部网络460内。可替代地，消息(诸如，对外部网络服务/应用450的请求、或对外部客户端装置410的响应或其他传输)的所期望目的地可以在外部网络内。

[0122] 如以上讨论的，可以通过针对消息的预定端到端处理流程连同端到端流程内的针对消息的当前处理点的确定来确定用于在代理服务器420内处理消息的特定策略的选择和应用。以上介绍的策略模型可以限定代理服务器420将在消息的端到端处理流程中的各个点向消息应用的消息处理策略的集合。例如，示例的策略模型600a和600b分别限定用于虚拟应用(即，反向代理使用情况)和代理应用(即，正向代理使用情况)的端到端处理流程。这些策略模型识别消息的端到端处理流程内的各个点(或断言)，并且包括在识别的各处理点或断言处要向消息应用的特定策略。

[0123] 在一些实施例中，可以使用策略模板的集合来创建用于限定端到端处理流程的策略模型和其他技术。例如，简要参照图7A至图7D，示出与四个不同断言对应的四个示例策略模板。图7A示出示例的“On Request”策略模板；图7B示出示例的“Invoke”策略模板；图7C示出示例的“Invoke Proxy”策略模板；并且图7D示出示例的“On Respones”策略模板。图7A至图7D中的每个策略模板包括“PolicyReference URI”标签，但是在这些模板中URI被留为空。因此，可以使用这些模板来创建端到端处理流程的策略模型(诸如，策略模型600a和600b)。例如，可以复制图7A至图7D中的模板中的一个或多个并且可以将适宜的策略URI插入每个模板副本中。然后可以将定制的模板添加到适宜的策略模型以限定在端到端处理流程期间可以执行的策略。

[0124] 在限定在端到端处理流程期间将执行的断言和策略之外，策略模型(和其他形式的预定端到端处理流程)还可以限定可以执行或不可以执行某些策略的条件。在一些实施例中，策略模型可以包含逻辑指令的集合，逻辑指令的集合用于实现用于执行在策略模型中引用的每个策略的条件。例如，策略模型可以包括指示代理服务器420应该针对某些消息类型(例如，SOAP、REST、或URL消息)而不针对其他消息类型执行某个策略的条件。另外，如以上讨论的，在一些情况下，策略模型可以选择性地在服务/应用级和/或操作/方法级应用策略，并且因此特定策略的应用可以不仅依赖于正被调用的后端网络应用/服务430，而且依赖于正在应用/服务430内调用的特定操作或方法。在各种另外的实施例中，一些策略模型可以包括指示代理服务器420应该针对一些用户执行某个策略而针对其他用户不执行、针对一些客户端装置类型执行而针对其他类型不执行，针对一些后端网络服务/应用执行而针对其他后端网络服务/应用不执行、和/或针对与消息相关的任何其他特性执行某个策略的条件。

[0125] 现在参照图8，示出从外部客户端装置410发送到内部SOAP网络服务430的REST请求的端到端处理流程的示例示图。在这一示例中，可以由代理服务器420连同如上所述的计算环境400中的各种其他组件来执行处理流程的执行。在这一示例中，初始消息是客户端装置410旨在用于内部计算机网络460中的后端网络服务430的REST请求，并且因此代理服务器420可以以反向代理模式操作。

[0126] 如上讨论的，在这一示例中，可以通过限定特定处理点(或断言)和应该由代理服务器420在端到端处理流程期间的每个处理点处执行的特定策略的预定策略模型来控制端到端处理流程图800。在这一示例中，在步骤801处，从客户端装置410接收REST请求。在步骤802中，代理服务器420可以执行在控制这一请求的处理的策略模型内识别的一个或多个“On Request”策略。在这一示例中，“On Request”策略包括在步骤803中访问认证/授权服务，以认证从客户端装置410接收的用户凭证和/或确认用户访问所请求的后端网络服务430的授权许可。在步骤804中，代理服务器420确定所请求的服务需要SOAP输入，因此在步骤805中执行“Message Transformation”策略以将REST请求转换成SOAP请求。在步骤806中，在步骤807中向后端SOAP网络服务传输SOAP请求之前，代理服务器420执行可以实现各种安全和通信管理功能的“Invoke Service”策略。在步骤808中，在从后端SOAP网络服务430接收到SOAP响应之后，代理服务器420可以再次确定向客户端410的输出应该是REST输出并且因此可以在步骤809中执行另一个“Message Transformation”策略以将SOAP响应变换成REST响应。在步骤810中，在步骤811中向客户端装置传输SOAP请求之前，代理服务器

420执行可以实现各种另外的安全和通信管理功能的“On Response”策略。

[0127] 现在参照图9，示出对从内部客户端装置440发送到外部网络服务或应用450的对于网络资源的请求的端到端处理流程的另一个示例示图。与在前一示例中一样，在这一示例中，可以通过代理服务器420连同如上所述的计算环境400中的各种其他组件来执行处理流程的执行。在这一示例中，初始消息是内部计算机网络460中的客户端装置410对外部网络服务或应用450的请求，因此代理服务器420可以以正向代理模式操作。

[0128] 如上讨论的，在这一示例中，可以通过限定特定处理点(或断言)和限定应该由代理服务器420在端到端处理流程期间的每个处理点处执行的特定策略的预定策略模型来控制端到端处理流程图900。在这一示例中，在步骤901处，从客户端装置440接收网络请求。在步骤902中，代理服务器420可以执行在控制这一请求的处理的策略模型内识别的一个或多个“On Request”策略。在执行“On Request”策略之后，在步骤904中向外部网络服务或应用450传输请求之前，代理服务器420可以在步骤903中执行一个或多个“On Invoke”策略以实现各种安全和通信管理功能。在这一示例中，代理服务器420识别在端到端处理流程期间已经发生的错误(诸如，从外部网络服务或应用450接收的错误或在代理服务器420所执行的处理内发生的错误)。因此，在步骤905中，代理服务器420可以执行一个或多个“On Error”策略来实现各种安全功能、分析和错误处置。在这种情况下，在向外部网络服务或应用450重新发送回请求之前，“On Error”策略可以指示代理服务器420执行另外的消息处理。因此，在已经应用“On Error”策略之后，代理服务器420可以在步骤906中重新执行“On Invoke”策略，然后在步骤907中向外部网络服务或应用450重新传输请求。在步骤908中，在从后端网络服务或应用450接收到响应之后，在步骤909中向内部客户端装置440传输请求之前，代理服务器420可以执行可以实现各种另外的安全和通信管理功能的“On Request”策略。

[0129] 如以上示例例示的，本文中描述的各种实施例可以支持动态策略模型，在该动态策略模型中，可以在DMZ或其他逻辑或物理子网内在消息的整个端到端处理流程中的各种不同处理点处应用不同安全策略和其他通信管理策略。这一动态策略模型框架可以用于建立和实现另外的安全以阻止来自恶意外部计算系统的攻击，并且可以实现在最后一英里安全基础设施内(例如，在后端网络服务/应用430内)可能不可能或不优选的另外类型的安全策略。另外，可以使用本文中描述的动态策略模型来实现鲁棒的认证和授权系统(诸如，令牌翻译和/或单点登陆访问控制系统)。例如，客户端装置410可以经由用户名/密码或其他用户凭证认证，并且预定的端到端处理流程可以在代理服务器420内执行，该代理服务器从内部网络460内的可信任的认证/授权服务执行令牌取回和验证，以便取回或生成不同类型的各种不同的访问令牌(例如，Kerberos令牌、SPNEGO令牌、用户名令牌、NTLM令牌、SAML令牌等)。因此，在用户提供有效凭证的一个集合并且被成功认证和授权之后，代理服务器420内的各种策略模型可以用于通过取回或生成针对后续被用户访问的各种不同后端网络服务/应用430的对应令牌类型来实现单点登陆访问控制系统。

[0130] 根据本申请的实施例，提供了一种包括处理单元和通信单元的系统。这种系统可以由硬件、软件或硬件和软件的组合来实现，以执行本发明的原理。本领域的技术人员理解的是，可以由上述组件(诸如，图3中示出的组件)来实现处理单元和通信单元。同时，本领域的技术人员理解的是，处理单元和通信单元可以被组合和分离成子单元以实现如上所述的

本发明的原理。因此，本文中的描述可以支持本文中描述的功能单元的任何可能的组合或分离或其他限定。

[0131] 在以上实施例的示例中，处理单元和通信单元可以协作以执行以下操作：接收第一消息，其中，该系统被配置成在内部计算机网络的子网内操作，该系统将内部计算机网络的网络应用或服务的集合暴露于外部计算机网络；确定针对第一消息的所期望目的地；基于针对第一消息的所期望目的地，确定该系统应该充当正向代理还是反向代理；确定针对所述第一消息的预定处理流程中的当前点；从用于处理消息的多个策略中选择用于处理第一消息的策略，其中，该选择基于预定处理流程中的当前点或确定该系统应该充当正向代理还是反向代理中的至少一个；按照所选择的策略来处理第一消息；以及在处理第一消息之后，向所期望目的地传输第一消息。

[0132] 在另一个示例中，处理单元和通信单元可以协作以进一步执行以下操作：确定第一消息调用内部计算机网络的简单对象访问协议(SOAP)虚拟服务内的一个或多个SOAP操作或者是一个或多个SOAP操作的部分；基于所确定的SOAP操作和SOAP虚拟服务来选择用于处理第一消息的策略；以及在按照所选择的策略处理第一消息之后，使用第一消息内的数据来调用所确定的一个或多个SOAP操作。

[0133] 在又一个示例中，处理单元和通信单元可以协作以进一步执行以下操作：确定第一消息对应于与内部计算机网络的表征状态转移(REST)虚拟服务或虚拟网络应用关联的一种或多种超文本传输协议(HTTP)方法；基于所确定的HTTP方法和REST虚拟服务或虚拟网络应用来选择用于处理第一消息的策略；以及在按照所选择的策略处理了第一消息之后，使用第一消息内的数据来调用所确定的一种或多种HTTP方法。

[0134] 在以上描述中，出于例示目的以特定次序描述了方法。应该理解的是，在替代实施例中，可以用与所描述的不同的次序来执行方法。还应该理解的是，上述方法可以由硬件组件来执行，或者体现在机器可执行指令序列中，该机器可执行指令序列可以用于致使机器(诸如，用指令编程的通用或专用处理器或逻辑电路)执行这些方法。这些机器可执行指令可以被存储在一个或多个机器可读取介质或存储器装置(诸如，CD-ROM或其他类型的光盘、软盘、ROM、RAM、EPROM、EEPROM、磁卡或光学卡、闪存、或适于存储电子指令的其他类型的机器可读介质或存储器装置)上。另选地，这些方法可以通过硬件和软件的组合来执行。

[0135] 虽然在本文中详细描述了本发明的例示和目前优选的实施例，但要理解的是，可以以其他方式不同地实施和采用本发明构思，并且所附权利要求书旨在被理解为包括除了受现有技术限制外的这些变型。

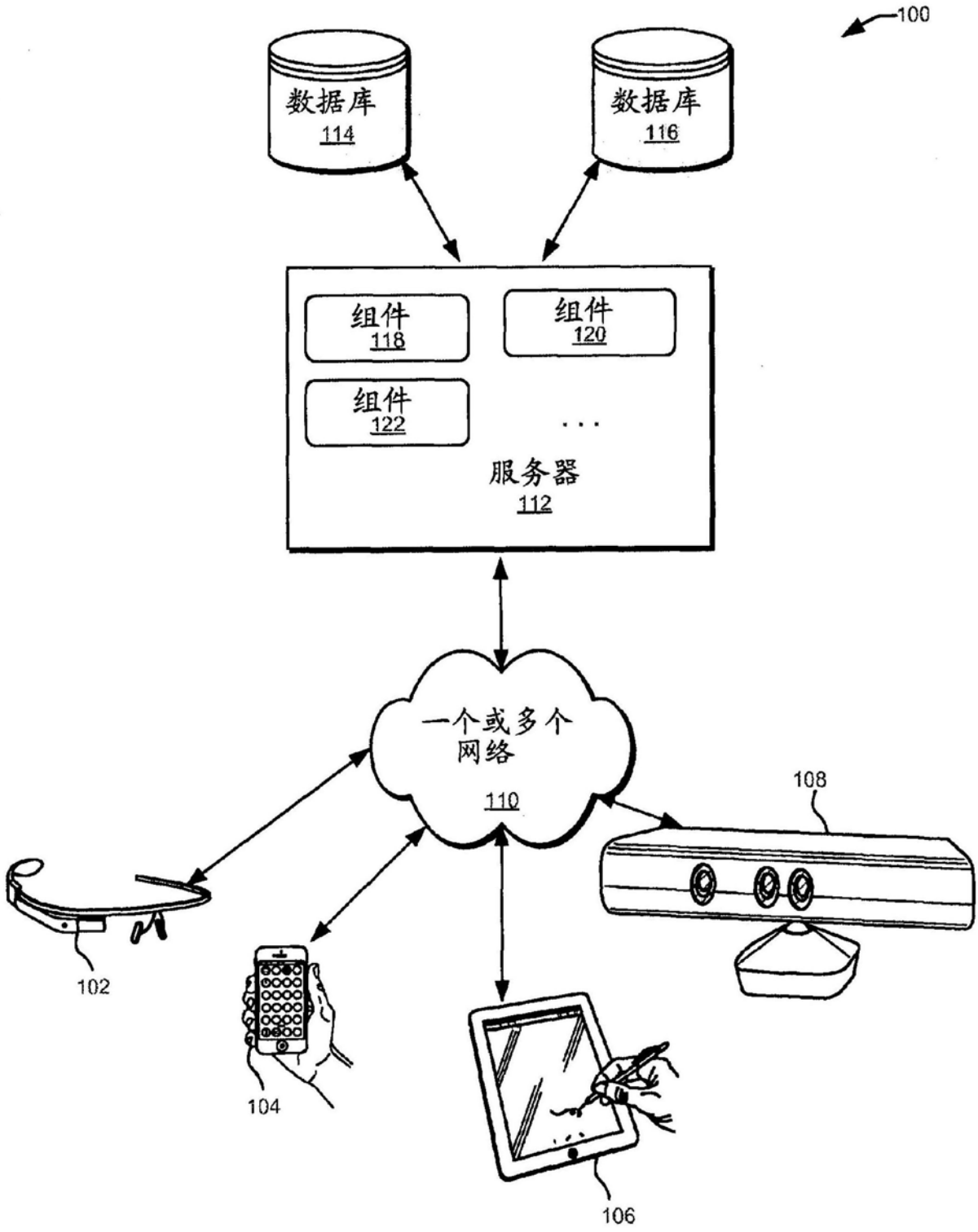

100
数据库
114
数据库
116
组件
118
组件
120
组件
122
...
服务器
112
一个或多个
网络
110
108
102
104
106

图1

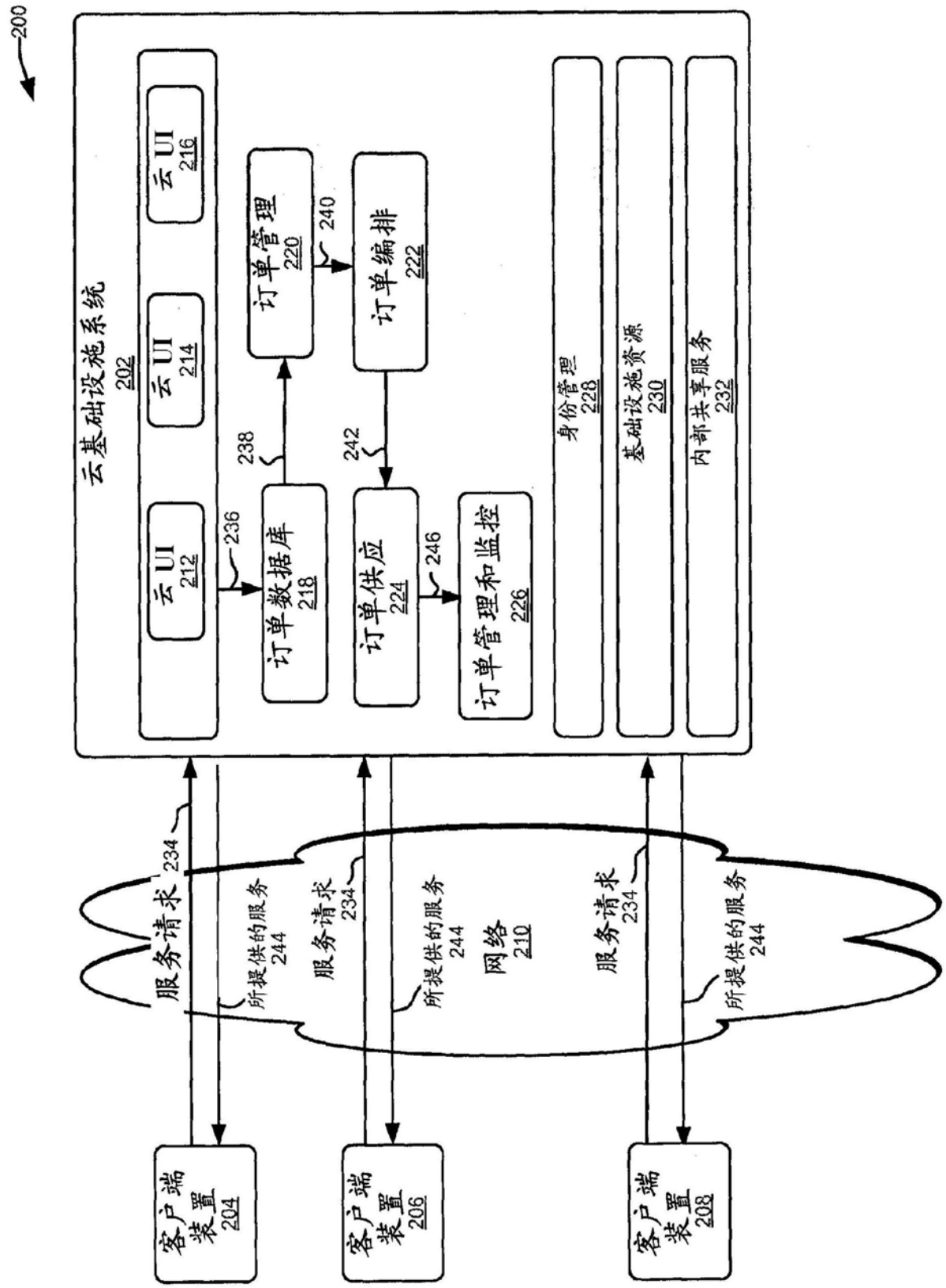

200
云基础设施系统
202
云 UI
212
云 UI
214
云 UI
216
订单数据库
218
订单管理
220
订单编排
222
订单供应
224
订单管理和监控
226
身份管理
228
基础设施资源
230
内部共享服务
232
236
238
240
242
246
服务请求
234
所提供的服务
244
网络
210
客户端
装置
204
客户端
装置
206
客户端
装置
208

图2

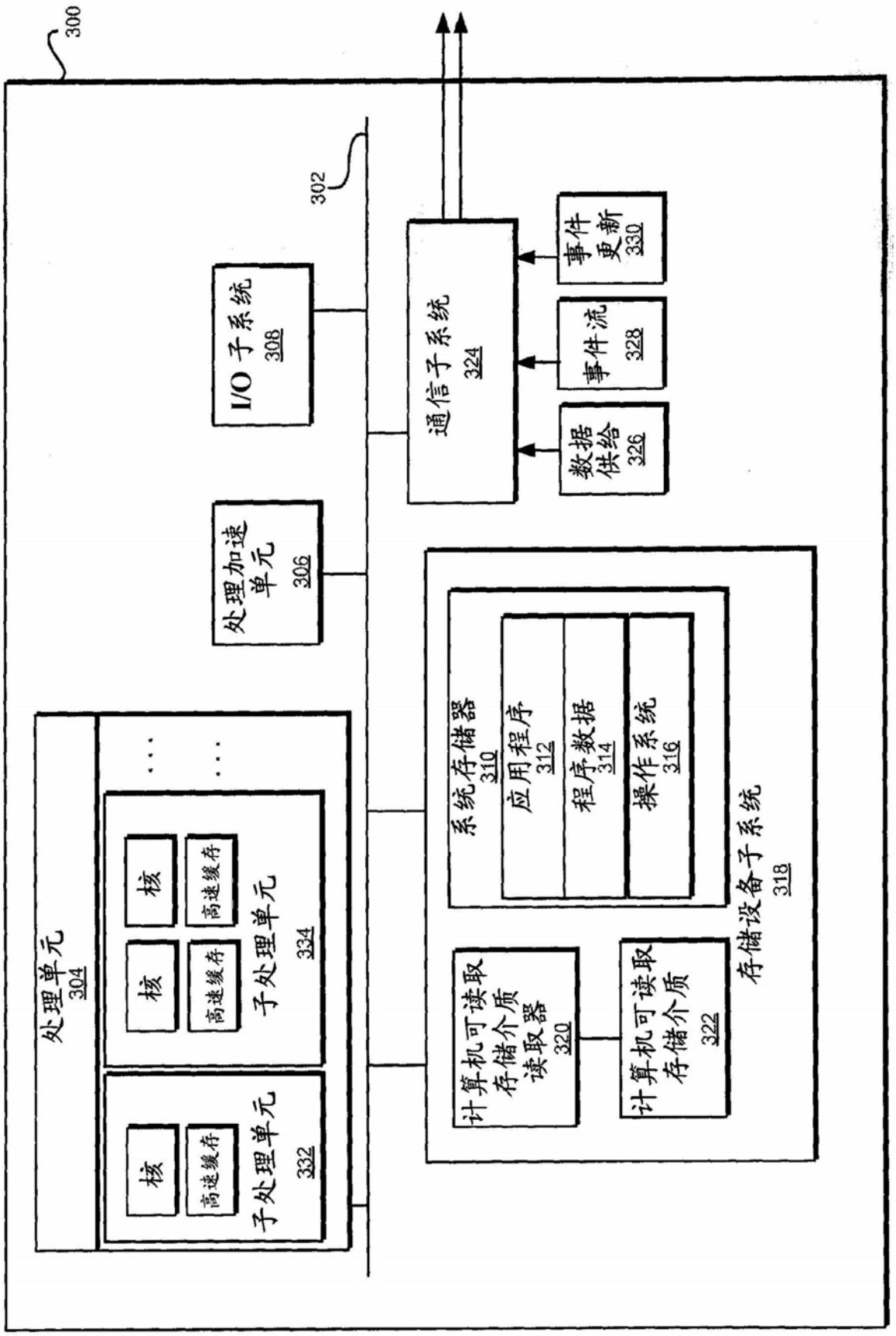

300
302
处理单元
304
核
高速缓存
子处理单元
332
核
高速缓存
核
高速缓存
子处理单元
334
I/O 子系统
308
处理加速单元
306
通信子系统
324
事件更新
330
事件流
328
数据供给
326
系统存储器
310
应用程序
312
程序数据
314
操作系统
316
存储设备子系统
318
计算机可读取存储介质读取器
320
计算机可读取存储介质
322

图3

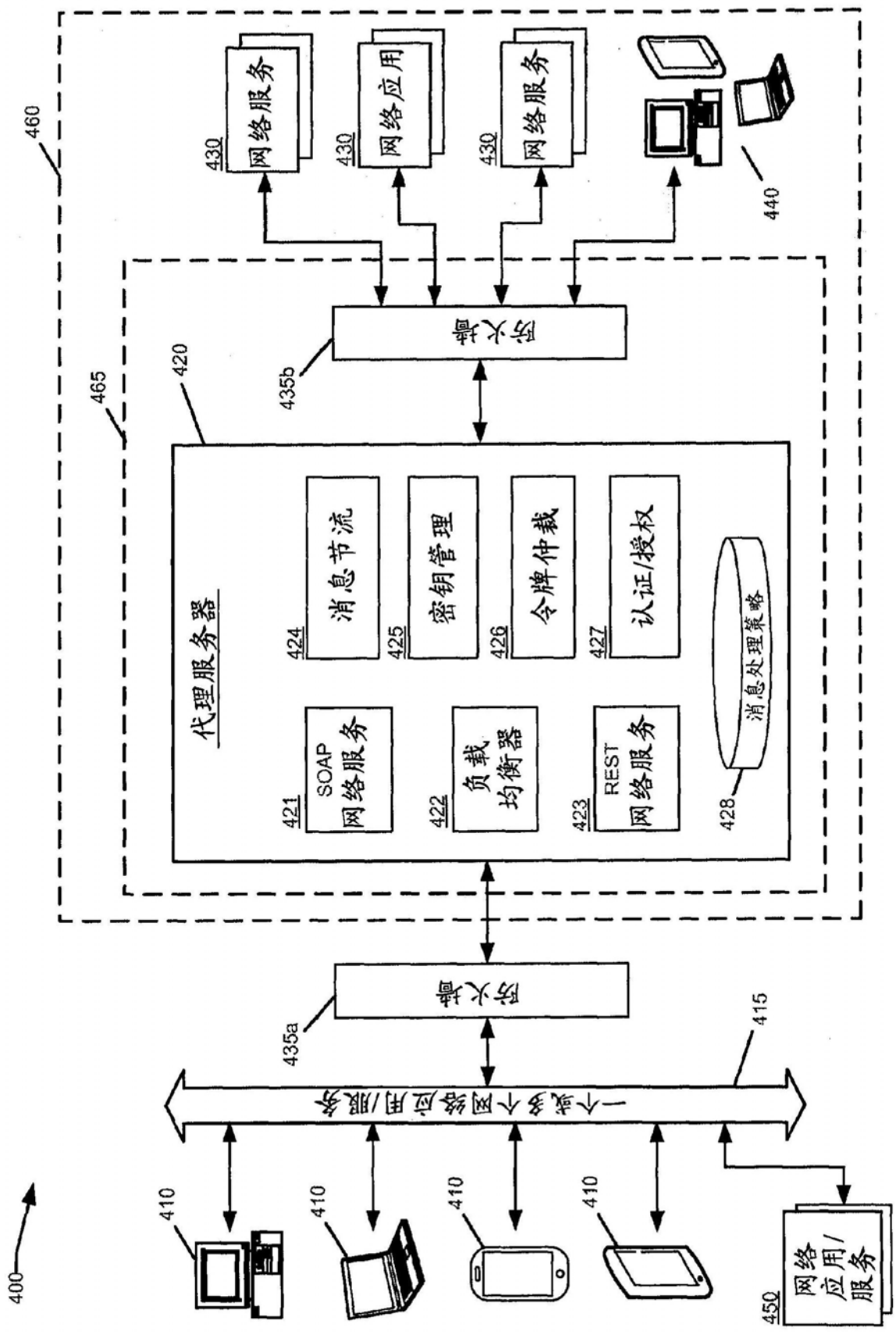

400
460
465
420
代理服务器
SOAP 网络服务
421
负载均衡器
422
REST 网络服务
423
消息节流
424
密钥管理
425
令牌仲裁
426
认证/授权
427
消息处理策略
428
安全性
435a
安全性
435b
网络服务
430
网络应用
430
网络服务
430
440
415
网络应用/服务
410
410
410
410
网络应用/服务
450

图4

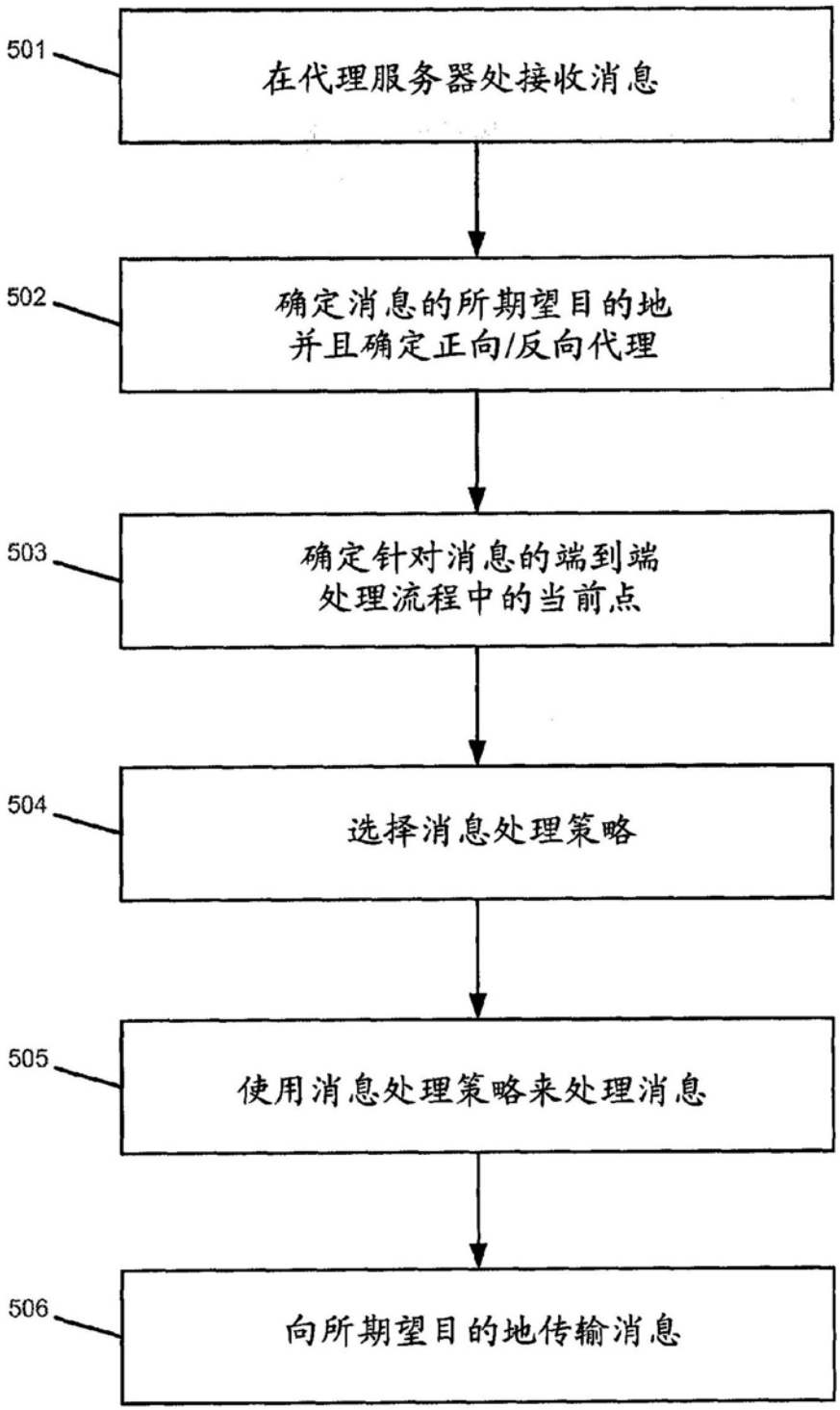

501
在代理服务器处接收消息
502
确定消息的所期望目的地
并且确定正向/反向代理
503
确定针对消息的端到端
处理流程中的当前点
504
选择消息处理策略
505
使用消息处理策略来处理消息
506
向所期望目的地传输消息

图5

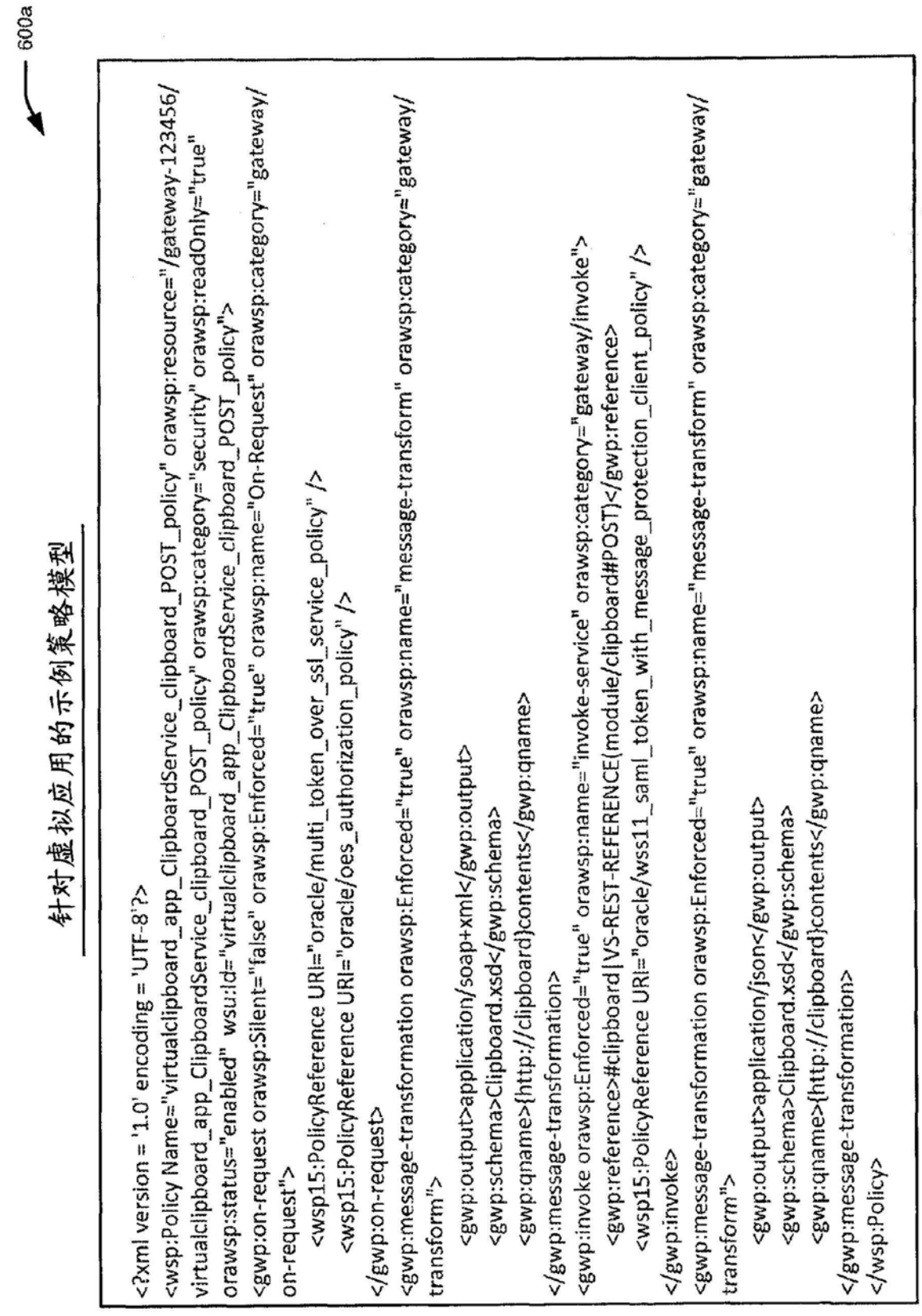

600a
针对虚拟应用的示例策略模型
<?xml version = '1.0' encoding = 'UTF-8'?>
<wsp:Policy Name="virtualclipboard_app_ClipboardService_clipboard_POST_policy" orawsp:resource="/gateway-123456/virtualclipboard_app_ClipboardService_clipboard_POST_policy" orawsp:category="security" orawsp:readOnly="true" orawsp:status="enabled" wsu:Id="virtualclipboard_app_ClipboardService_clipboard_POST_policy">
<gwp:on-request orawsp:Silent="false" orawsp:Enforced="true" orawsp:name="On-Request" orawsp:category="gateway/on-request">
<wsp15:PolicyReference URI="oracle/multi_token_over_ssl_service_policy" />
<wsp15:PolicyReference URI="oracle/oes_authorization_policy" />
</gwp:on-request>
<gwp:message-transformation orawsp:Enforced="true" orawsp:name="message-transform" orawsp:category="gateway/transform">
<gwp:output>application/soap+xml</gwp:output>
<gwp:schema>Clipboard.xsd</gwp:schema>
<gwp:qname>{http://clipboard}contents</gwp:qname>
</gwp:message-transformation>
<gwp:invoke orawsp:Enforced="true" orawsp:name="invoke-service" orawsp:category="gateway/invoke">
<gwp:reference>#clipboard|VS-REST-REFERENCE(module/clipboard#POST)</gwp:reference>
<wsp15:PolicyReference URI="oracle/wss11_saml_token_with_message_protection_client_policy" />
</gwp:invoke>
<gwp:message-transformation orawsp:Enforced="true" orawsp:name="message-transform" orawsp:category="gateway/transform">
<gwp:output>application/json</gwp:output>
<gwp:schema>Clipboard.xsd</gwp:schema>
<gwp:qname>{http://clipboard}contents</gwp:qname>
</gwp:message-transformation>
</wsp:Policy>

图6A

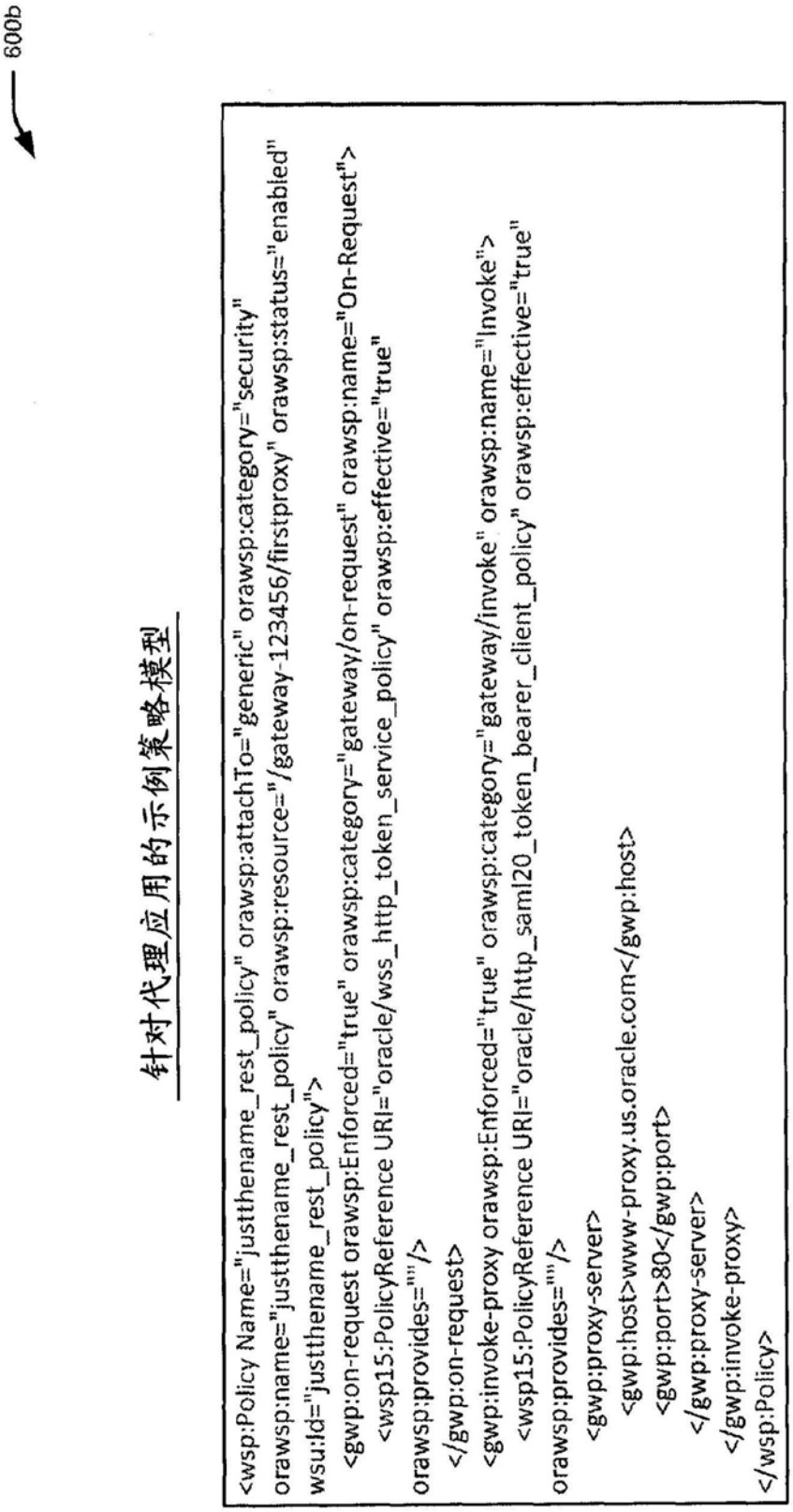

600b
针对代理应用的示例策略模型
<wsp:Policy Name="justthename_rest_policy" orawsp:attachTo="generic" orawsp:category="security"
orawsp:name="justthename_rest_policy" orawsp:resource="/gateway-123456/firstproxy" orawsp:status="enabled"
wsu:Id="justthename_rest_policy">
<gwp:on-request orawsp:Enforced="true" orawsp:category="gateway/on-request" orawsp:name="On-Request">
<wsp15:PolicyReference URI="oracle/wss_http_token_service_policy" orawsp:effective="true"
orawsp:provides=""/>
</gwp:on-request>
<gwp:invoke-proxy orawsp:Enforced="true" orawsp:category="gateway/invoke" orawsp:name="Invoke">
<wsp15:PolicyReference URI="oracle/http_saml20_token_bearer_client_policy" orawsp:effective="true"
orawsp:provides=""/>
<gwp:proxy-server>
<gwp:host>www-proxy.us.oracle.com</gwp:host>
<gwp:port>80</gwp:port>
</gwp:proxy-server>
</gwp:invoke-proxy>
</wsp:Policy>

图6B

示例 “ON REQUEST” 消息处理策略模板

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<orawsp:Template
orawsp:id="on_request_template"
orawsp:description="i18n:oracle.idm.gateway.common.resources.artifactsdescription.GatewayArtifactsDescriptio
nBundle_oracle/on_request_template_ATDescKey"
orawsp:displayName="i18n:oracle.idm.gateway.common.resources.artifactsdescription.GatewayArtifactsDescripti
onBundle_oracle/on_request_template_ATDispNameKey"
orawsp:readOnly="true"
orawsp:attachTo="generic"
xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"
orawsp:name="oracle/on_request_template"
orawsp:category="gateway"
xmlns:gwp="http://schemas.oracle.com/gw-policy">
    <gwp:on-request orawsp:name="On-Request" orawsp:Silent="true" orawsp:Enforced="true"
        orawsp:category="gateway/on-request"/>
    <wsp15:PolicyReference URI="" />
</orawsp:Template>
```

图7A

示例“INVOKE”消息处理策略模板

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<orawsp:Template
orawsp:Id="invoke_template"
orawsp:description="i18n:oracle.idm.gateway.common.resources.artifactsdescription.GatewayArtifactsDescriptio
nBundle_oracle/invoke_template_ATDescKey"
orawsp:displayName="i18n:oracle.idm.gateway.common.resources.artifactsdescription.GatewayArtifactsDescripti
onBundle_oracle/invoke_template_ATDispNameKey"
        orawsp:readOnly="true"
        orawsp:attachTo="generic"
        xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"
        orawsp:name="oracle/invoke_template"
        orawsp:category="gateway"
        xmlns:gwp="http://schemas.oracle.com/gw-policy">
    <gwp:invoke orawsp:name="invoke" orawsp:Silent="true" orawsp:Enforced="true"
            orawsp:category="gateway/invoke"/>
        <wsp15:PolicyReference URI="" />
</orawsp:Template>
```

图7B

示例 “INVOKE PROXY” 消息处理策略模板

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<orawsp:Template
orawsp:id="invoke_proxy_template"
orawsp:description="i18n:oracle.idm.gateway.common.resources.artifactsdescription.GatewayArtifactsDescriptio
nBundle_oracle/invoke_proxy_template_ATDescKey"
orawsp:displayName="i18n:oracle.idm.gateway.common.resources.artifactsdescription.GatewayArtifactsDescripti
onBundle_oracle/invoke_proxy_template_ATDispNameKey"
        orawsp:readOnly="true"
        orawsp:attachTo="generic"
        xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"
        orawsp:name="oracle/invoke_proxy_template"
        orawsp:category="gateway"
        xmlns:gwp="http://schemas.oracle.com/gw-policy">
    <gwp:invoke-proxy orawsp:name="Invoke-Proxy" orawsp:Silent="true" orawsp:Enforced="true"
                orawsp:category="gateway/invoke"/>
        <wsp15:PolicyReference URI="" />
</orawsp:Template>
```

图7C

示例“ON RESPONSE”消息处理策略模板

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<orawsp:Template
orawsp:id="on_response_template"
orawsp:description="i18n:oracle.idm.gateway.common.resources.artifactsdescription.GatewayArtifactsDescriptio
nBundle_oracle/on_response_template_ATDescKey"
orawsp:displayName="i18n:oracle.idm.gateway.common.resources.artifactsdescription.GatewayArtifactsDescripti
onBundle_oracle/on_response_template_ATDispNameKey"
orawsp:readOnly="true"
orawsp:attachTo="generic"
xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"
orawsp:name="oracle/on_response_template"
orawsp:category="gateway"
xmlns:gwp="http://schemas.oracle.com/gw-policy">
    <gwp:on-response orawsp:name="On-Response" orawsp:Silent="true" orawsp:Enforced="true"
        orawsp:category="gateway/on-response"/>
    <wsp15:PolicyReference URI="" />
</orawsp:Template>
```

图7D

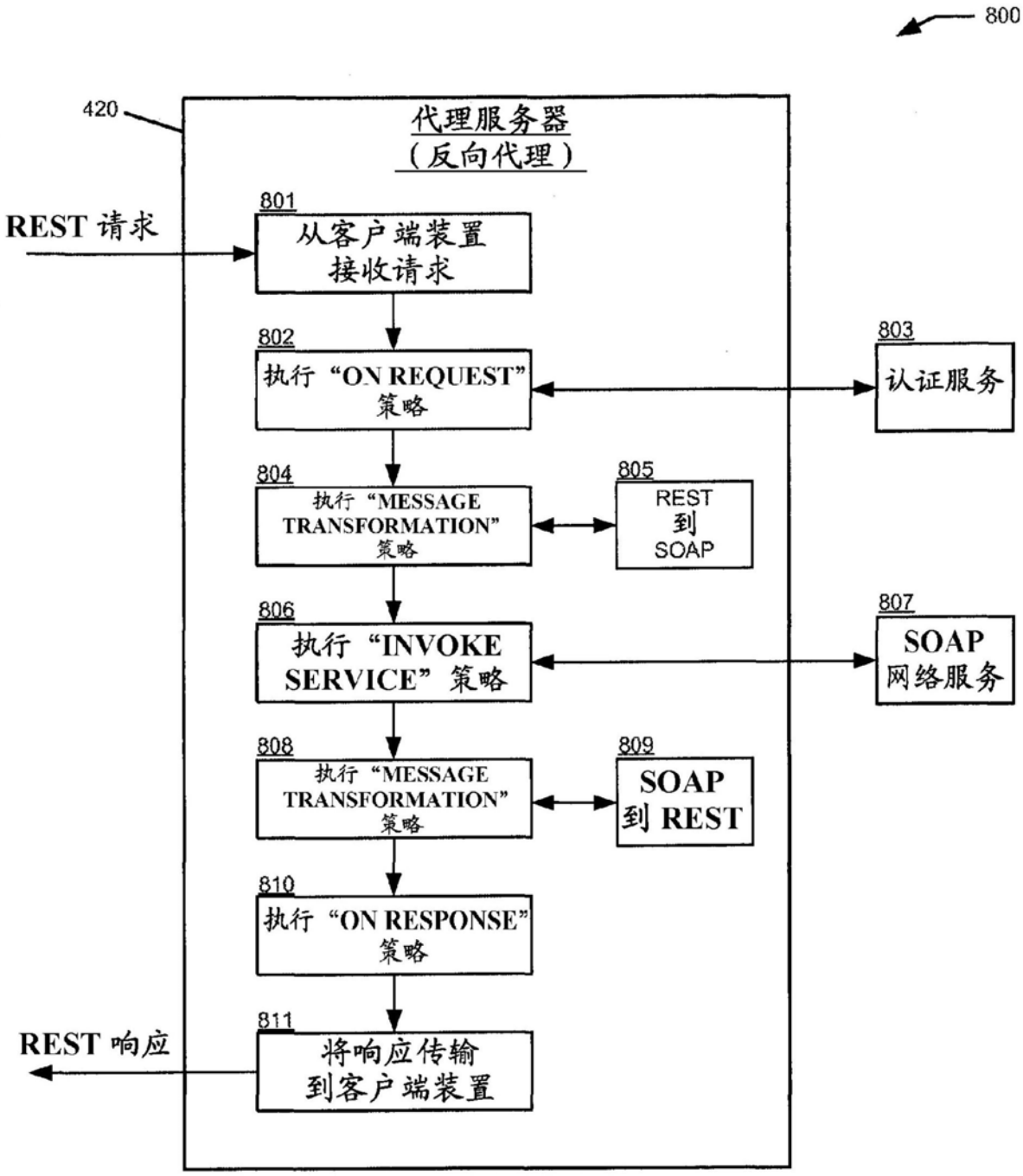

800
420
代理服务器
（反向代理）
REST 请求
801
从客户端装置
接收请求
802
执行 “ON REQUEST”
策略
803
认证服务
804
执行 “MESSAGE
TRANSFORMATION”
策略
805
REST
到
SOAP
806
执行 “INVOKE
SERVICE” 策略
807
SOAP
网络服务
808
执行 “MESSAGE
TRANSFORMATION”
策略
809
SOAP
到 REST
810
执行 “ON RESPONSE”
策略
811
将响应传输
到客户端装置
REST 响应

图8

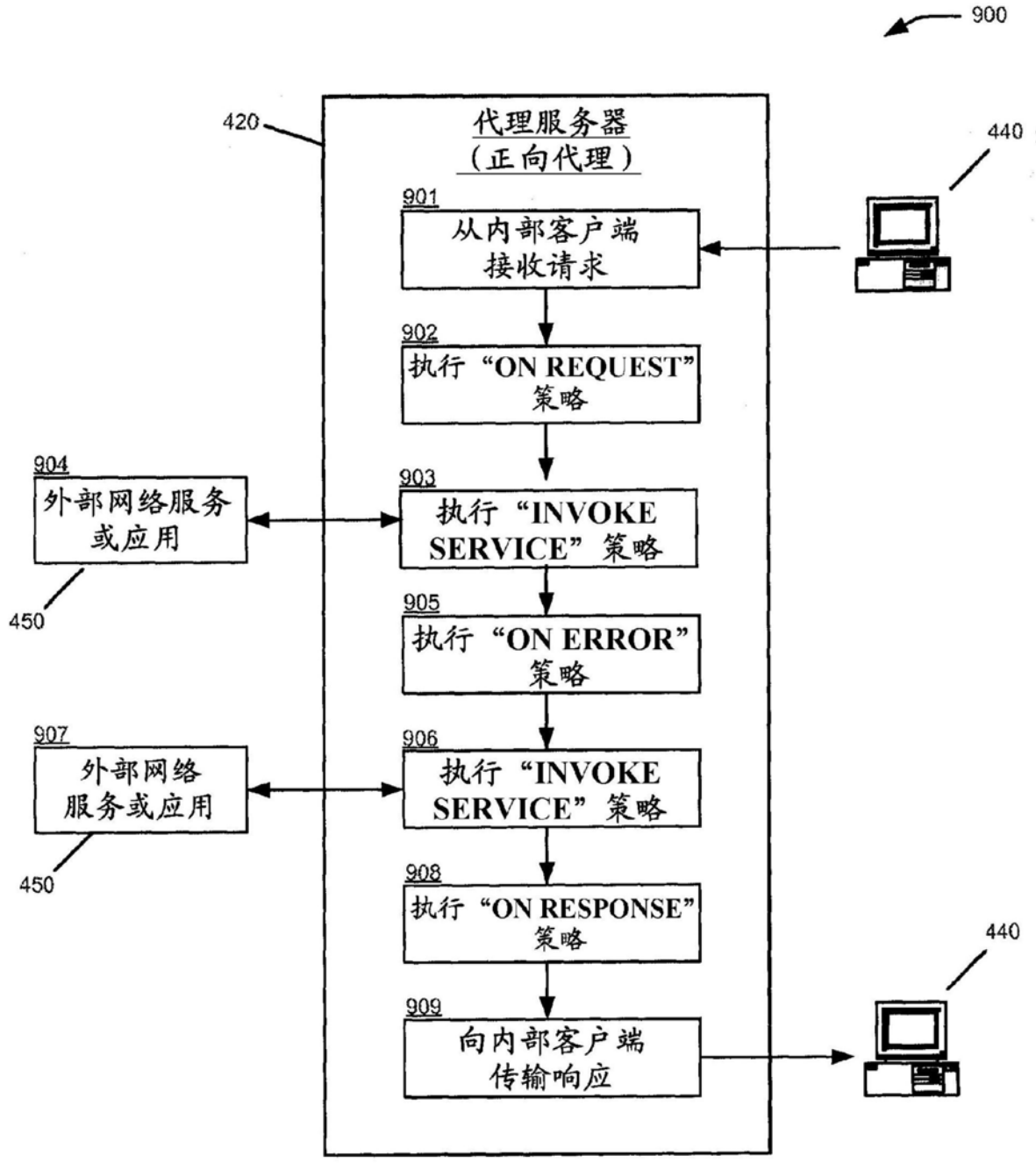

900
420
代理服务器
（正向代理）
440
901
从内部客户端
接收请求
902
执行“ON REQUEST”
策略
904
外部网络服务
或应用
903
执行“INVOKE
SERVICE”策略
450
905
执行“ON ERROR”
策略
907
外部网络
服务或应用
906
执行“INVOKE
SERVICE”策略
450
908
执行“ON RESPONSE”
策略
440
909
向内部客户端
传输响应

图9