(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113711563 B
(45)授权公告日 2023.06.13

(21)申请号 202080029015.4
(22)申请日 2020.04.06
(65)同一申请的已公布的文献号
申请公布号 CN 113711563 A
(43)申请公布日 2021.11.26
(30)优先权数据
19172926.8 2019.05.07 EP
(85)PCT国际申请进入国家阶段日
2021.10.15
(86)PCT国际申请的申请数据
PCT/IB2020/053254 2020.04.06
(87)PCT国际申请的公布数据
WO2020/225616 EN 2020.11.12
(73)专利权人 国际商业机器公司
地址 美国纽约阿芒克
(72)发明人 M·斯莫尔尼 T·杜尔 M·贝克 J·舍克
(74)专利代理机构 北京市金杜律师事务所 11256
专利代理师 姚杰
(51)Int.Cl.
*H04L 9/40*(2022.01)
*H04L 9/32*(2006.01)
(56)对比文件
US 2014223516 A1,2014.08.07
US 2017223026 A1,2017.08.03
US 2018337784 A1,2018.11.22
CN 109309683 A,2019.02.05
CN 107085681 A,2017.08.22
CN 103716283 A,2014.04.09
审查员 魏慧慧

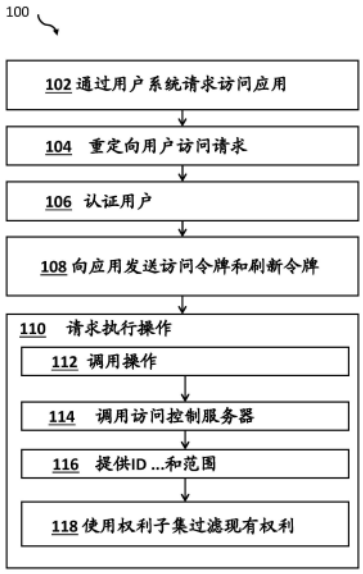
权利要求书2页 说明书13页 附图15页

(54)发明名称

# 基于细粒度令牌的访问控制

(57)摘要

可以提供一种用于在数据处理环境中进行基于令牌的授权的计算机实现的方法。数据处理环境至少包括用户系统、应用、认证服务器和访问控制服务器。方法包括通过用户系统请求访问应用，将用户访问请求重定向到认证服务器，认证用户，其中认证凭证包括对受限权利的请求，其中受限权利表示由访问控制服务器管理的资源的现有权利的子集。方法还包括从认证服务器向应用发送访问令牌，请求执行包括通过提供访问令牌的应用调用该操作的操作，调用访问控制服务器，以及提供包括现有权利的子集的令牌的范围。




CN 113711563 B

1.一种用于数据处理环境中基于令牌的授权的计算机实现的方法,其中数据处理环境至少包括用户系统、应用、认证服务器和访问控制服务器,其中用户系统通过网络连接连接到执行应用的服务器,其中应用提供对操作的访问,其中至少操作可通过其标识符识别,其中该方法包括:

-通过用户系统请求访问应用,

-将用户访问请求重定向到身份验证服务器,

-通过在认证服务器和应用之间交换的认证凭证来认证用户,其中认证凭证包括使用范围的标准语义的对受限权利的请求,其中受限权利表示由访问控制服务器管理的资源的现有权利的子集,

-如果认证成功并且应用已经在认证服务器处注册,则从认证服务器向应用发送访问令牌和刷新令牌,其中访问令牌和刷新令牌包括受限权利,

-请求由用户系统发起的应用执行操作,包括

-通过提供包括受限权利的访问令牌的应用调用操作

-通过操作调用访问控制服务器,

-向访问控制服务器提供用户系统的标识符和包括现有权利子集的令牌的范围,

-访问控制服务器使用权利子集过滤现有权利,导致用户系统对操作的访问决策。

2.根据权利要求1所述的方法,其中标准语义基于OAuth2。

3.根据权利要求1或2所述的方法,其中应用是不安全的组件。

4.根据权利要求1或2所述的方法,其中对包括在认证凭证中受限权利的请求是第一请求,其中第一请求包括现有权利的最大子集。

5.根据权利要求4所述的方法,其中对刷新令牌的后续请求还包括对进一步受限权利的请求。

6.根据权利要求1或2所述的方法,其中操作包括从包括对预定义数据的访问、数据库访问、文件访问、预定义应用编程接口,对网络特定子网的访问的组中选择的至少一个。

7.根据权利要求1或2所述的方法,其中访问令牌具有预定义的有效时间。

8.根据权利要求7所述的方法,其中预定义的有效时间可由特权用户系统更新。

9.一种用于数据处理环境中基于令牌的授权的访问系统,其中数据处理环境至少包括用户系统、应用、认证服务器和访问控制服务器,其中用户系统通过网络连接连接到执行应用的服务器,其中应用提供对操作的访问,其中至少操作可通过其标识符识别,其中该访问系统包括:

-访问模块,适于通过用户系统请求访问应用,

-重定向模块,适于将用户访问请求重定向到身份验证服务器,

-认证服务器,适于通过在认证服务器和应用之间交换的认证凭证来认证用户,其中认证凭证包括使用范围的标准语义的对受限权利的请求,其中受限权利表示由访问控制服务器管理的资源的现有权利的子集,

-发送器,适于如果认证成功并且应用已经在认证服务器处注册,则从认证服务器向应用发送访问令牌和刷新令牌,其中访问令牌和刷新令牌包括受限权利,

-其中用户系统适于请求由用户系统发起的应用执行操作,其中请求执行操作包括

-通过提供包括受限权利的访问令牌的应用调用操作

-通过操作调用访问控制服务器，

-向访问控制服务器提供用户系统的标识符和包括现有权利子集的令牌的范围，

-访问控制服务器使用权利子集过滤现有权利，导致用户系统对操作的访问决策。

10.根据权利要求9所述的系统，其中标准语义基于OAuth2。

11.根据权利要求9或10所述的系统，其中应用是不安全的组件。

12.根据权利要求9或10所述的系统，其中对包括在认证凭证中受限权利的请求是第一请求，其中第一请求包括现有权利的最大子集。

13.根据权利要求12所述的系统，其中对刷新令牌的后续请求还包括对进一步受限权利的请求。

14.根据权利要求9或10所述的系统，其中操作包括从包括对预定义数据的访问、数据库访问、文件访问、预定义应用编程接口，对网络特定子网的访问的组中选择的至少一个。

15.根据权利要求9或10所述的系统，其中访问令牌具有预定义的有效时间。

16.根据权利要求15所述的系统，其中预定义的有效时间可由特权用户系统更新。

17.一种用于数据处理环境中基于令牌的授权的计算机可读存储介质，其中数据处理环境至少包括用户系统、应用、认证服务器和访问控制服务器，其中用户系统通过网络连接连接到执行应用的服务器，其中应用提供对操作的访问，其中至少操作可通过其标识符识别，所述计算机可读存储介质包括具有嵌入其中的程序指令，所述程序指令可由一个或多个计算系统或控制器执行以使得所述一个或多个计算系统：

-通过用户系统请求访问应用，

-将用户访问请求重定向到身份验证服务器，

-通过在认证服务器和应用之间交换的认证凭证来认证用户，其中认证凭证包括使用范围的标准语义的对受限权利的请求，其中受限权利表示由访问控制服务器管理的资源的现有权利的子集，

-如果认证成功并且应用已经在认证服务器处注册，则从认证服务器向应用发送访问令牌和刷新令牌，其中访问令牌和刷新令牌包括受限权利，

-请求由用户系统发起的应用执行操作，包括

-通过提供包括受限权利的访问令牌的应用调用操作

-通过操作调用访问控制服务器，

-向访问控制服务器提供用户系统的标识符和包括现有权利子集的令牌的范围，

-访问控制服务器使用权利子集过滤现有权利，导致用户系统对操作的访问决策。

# 基于细粒度令牌的访问控制

## 技术领域

[0001] 本发明总体上涉及一种用于基于令牌的授权的方法，并且更具体地，涉及一种用于在数据处理环境中进行基于令牌的授权的计算机实现的方法。本发明还涉及一种用于在数据处理环境中进行基于令牌的认证的相关访问系统，以及一种相关的计算机程序产品。

## 背景技术

[0002] 在过去十年中，基于令牌的认证和授权系统已经得到发展。在这些系统中，最终用户通过授权服务器进行身份验证并取回一个临时令牌，该令牌证明该最终用户的授权以及他的身份(可选)。例如，基于Web的用户界面(UI)使用此令牌来安全地调用后端服务。由于没有凭证流向这些服务，基于令牌的系统为不同供应商的服务提供了一个可以利用相同的授权系统的生态系统。对于服务之间更复杂的交互，还可以将令牌从UI通过一个服务传递到另一个下游服务。

[0003] 作为令牌数据的一部分，有效载荷通常携带授权集，称为“范围”。在各种基于令牌的系统中，范围的处理方式非常不同。从标准化的角度来看，只保留了少数几个作用域关键字，其他授权由授权框架决定。这些框架中的许多都采用了一个动作和/或基于角色的模型，这些模型通常描述服务、组件和/或动作。例如，像“database.query.read”这样的函数调用就是这样一个潜在的作用域。

## 发明内容

[0004] 根据本发明的一个方面，一种用于基于令牌的授权的计算机实现的方法可以至少包括用户系统、应用、认证服务器和访问控制服务器。由此，用户系统可以通过网络连接连接到执行应用的服务器，并且应用可以提供对操作的访问。至少该操作可以通过其标识符来识别。

[0005] 该方法可以包括经由用户系统请求访问应用，将用户访问请求重定向到认证服务器，以及通过在认证服务器和应用之间交换的认证凭证来认证用户。由此，认证凭证可以包括使用范围的标准语义的对受限权利的请求，并且受限权利可以表示由访问控制服务器为资源管理或控制的现有权利的子集。

[0006] 该方法还可以包括：如果认证成功并且应用已经在认证服务器处注册，则从认证服务器向应用发送访问令牌和刷新令牌，其中访问令牌和刷新令牌包括受限权利。

[0007] 请求由用户系统发起的应用执行操作可以包括通过提供包含受限权利的访问令牌的应用调用操作，通过操作调用访问控制服务器，向访问控制服务器提供用户系统的标识符和包括现有权利的子集的令牌的范围，访问控制服务器使用权利子集过滤现有权利，从而导致用户系统对操作的访问决策。

[0008] 根据本发明的另一方面，可以提供一种用于数据处理环境中基于令牌的授权的相关访问系统。

[0009] 此外，实施例可以采用相关计算机程序产品的形式，可从提供程序代码的计算机

可用或计算机可读介质访问，以供计算机或任何指令执行系统使用或与计算机或任何指令执行系统结合使用。出于本说明的目的，计算机可用或计算机可读介质可以是任何装置，可以包含用于存储、通信、传播或传输程序以供指令执行系统使用或与指令执行系统装置结合使用的装置，或设备。

## 附图说明

[0010] 应当注意，本发明的实施例是参照不同的主题来描述的。具体地，参考方法类型权利要求描述了一些实施例，而参考设备类型权利要求描述了其他实施例。然而，本领域技术人员将从以上和以下描述中了解到，除非另有说明，除了属于一类主题的特征的任意组合之外，还包括与不同主题相关的特征之间的任意组合。特别地，在方法类型权利要求的特征和设备类型权利要求的特征之间，被认为是在本文件中公开的。

[0011] 以上定义的方面，以及本发明的其他方面，从将在下文中描述的实施例的示例变得显而易见，并且参考实施例的示例进行解释，但本发明不限于此。

[0012] 本发明的优选实施例将仅通过示例的方式并参考以下附图进行描述：

[0013] 图1示出了用于数据处理环境中基于令牌的授权的本发明的计算机实现的方法的实施例的框图。

[0014] 图2示出了基于这里提出的概念的交互模型的框图。

[0015] 图3示出了替代发起交互的实施例的框图。

[0016] 图4示出了与图3相关的不言自明的通信图。

[0017] 图5示出了替代交互的框图，其中客户端检索具有受限权利的资源所有者的访问令牌和/或刷新令牌的。

[0018] 图6示出了第二类型交互600的框图，通过该交互，客户端204调用对不安全组件208的操作。

[0019] 图7示出了与图6相关的不言自明的通信图。

[0020] 图8示出了第三类交互的框图，其中不可信组件与资源服务器1和资源服务器2交互。

[0021] 图9示出了与图8相关的不言自明的通信图。

[0022] 图10示出了第四类型交互的框图，其中资源服务器1和资源服务器2与访问控制服务器交互。

[0023] 图11示出了与图10相关的不言自明的通信图。

[0024] 图12示出了传统访问控制服务器决策引擎的示例性伪代码段。

[0025] 图13示出了这里提出的访问控制服务器决策引擎的示例性伪代码段。

[0026] 图14示出了在数据处理环境中用于基于令牌的授权的访问系统的框图。

[0027] 图15示出了适合于为这里描述的组件执行程序代码的计算系统的实施例。

## 具体实施方式

[0028] 在本说明书的上下文中，可以使用以下约定、术语和/或表达：

[0029] 术语“基于令牌的授权”可以表示一种机制，通过该机制用户可以从他已经证明其认证的认证服务器接收临时数据结构，表示为令牌。这种令牌可以授权用户访问例如其他

服务器系统上的预定义系统、应用和/或服务。

[0030] 术语“用户系统”在这里通常可以表示用户操作的设备，如个人计算机、移动设备或由最终用户操作的任何其他计算机化系统。

[0031] 术语“应用”可以表示在从用户系统激活之后提供一种或多种服务和功能的可执行软件代码。通常，应用可以在服务器系统上管理和执行，例如在数据中心和/或在云计算范式下操作的服务器系统。因此，服务器系统通常可以从用户系统远程操作。

[0032] 术语“认证服务器”可以表示可信计算环境—通常由第三方操作用户系统和服务器系统—终端用户针对该环境操作用户系统—例如，通过使用Web浏览器—可以验证自己的身份，以获得访问预定义应用和服务的授权。

[0033] 术语“访问控制服务器”可以表示硬件或软件系统，能够向应用或使用该应用的用户系统授予或拒绝对特定服务的访问。

[0034] 术语“操作”可以表示应用的单个功能，或者一个或多个应用的多个功能和/或服务。操作的类型可以由访问范围定义和/或限制。操作的示例可以是数据库读取、读取指定的数据子集、处理这些数据、组合数据、在内容管理系统中执行搜索以及访问指定的网络地址和更多软件和/或硬件支持的服务和/或功能。

[0035] 术语“认证凭证”可以表示由最终用户(或相关系统)用来授权他访问预定义功能和服务的一组数据。

[0036] 术语“受限权利”可以指这样一个事实，即如果与应用或服务提供实体的所有可用功能相比，用户系统和/或相关的最终用户可能仅具有访问和/或使用受限的功能。因此，他可能无权使用全套功能或服务，而只能使用受限制的子集，例如读取数据但不更改数据，应用也可以这样做。

[0037] 术语“范围”可以表示用户系统或相关最终用户或—通常—系统用户可以访问的服务和/或功能的定义集合。范围可能是资源可用的完整服务和功能集，也可能仅限于一个子集。

[0038] 术语“标准语义”在此可以表示例如像OAuth版本2中的标准化授权系统的一组定义的命令和功能。然而，也可以支持其他版本。

[0039] 术语“OAuth2”可以表示用于访问委托的已知开放标准，通常用作因特网用户授予网站和/或应用访问他们在其他网站上的信息但不向他们提供密码的方式。这种机制被公共互联网站点广泛使用，以允许用户与第三方应用或网站共享有关其帐户的信息。通常，OAuth2(表示标准的第2版)可以代表资源所有者向客户端提供对资源的“安全委托访问”。它可以为资源所有者指定一个流程来授权第三方访问他们的资源—即服务和功能—而不共享他们的凭证(即身份验证凭证)。通常，授权服务器向用户系统颁发的访问令牌用于确认资源所有者的批准。第三方，即具有其相关用户系统的最终用户，然后可以使用访问令牌来访问由服务器托管的受保护资源。

[0040] 此外，OAuth2的优选的基于令牌的实施环境的一些术语—特别是角色—也可以在此处定义：

[0041] 基本上，在OAuth2规范RFC 6749中描述了所使用的角色，比较https://tools.ietf.org/html/rfc6749。

[0042] 资源所有者(例如，最终用户、系统用户)可以是能够授予对受保护资源的访问权

的实体。当资源所有者是个人时，它被称为最终用户。

[0043] 资源服务器(即，服务)可以是托管受保护资源的服务器，能够使用访问令牌接受和响应受保护资源请求。对于这里提出的概念，资源服务器可以连接到可以帮助确定基于传递的令牌的操作是被允许还是被拒绝的访问控制服务器。

[0044] 客户端可以是代表资源所有者并在其授权下发出受保护资源请求的应用。术语“客户端”并不意味着任何实现特征，即，这可以是基于Web的应用、桌面应用或在不同设备上运行的应用。

[0045] 授权服务器可以是在成功验证资源所有者(最终用户)并获得授权之后向客户端发出访问令牌的服务器。授权服务器可以是与资源服务器相同的服务器，也可以是单独的实体。单个授权服务器可以发布多个资源服务器接受的访问令牌。

[0046] 对于这里描述的概念并且除了OAuth2定义的角色之外，不受信任的组件可以代表资源所有者执行脚本。为此，它接收资源所有者的访问令牌并使用它来调用一个或多个资源服务器。

[0047] 除了OAuth2定义的角色之外，访问控制服务器可能旨在通过将来自传递的令牌的属性与先前创建的策略进行比较来支持资源服务器评估访问请求。

[0048] 此外，可以注意到，在此处提出的概念的上下文中，根据OAuth2的单个授权服务器的功能已被拆分为认证服务器和访问控制服务器，用于更细粒度的授权和访问控制.

[0049] 所提出的用于数据处理环境中基于令牌的认证的计算机实现的方法可以提供多种优势和技术效果：

[0050] 所提出的概念可以无缝地集成到使用基于令牌的授权模型的身份和访问管理(IAM)系统中，允许对资源管理策略的细粒度访问控制。根据这样的模型，令牌可能仅标识所执行操作的参与者(即最终用户)，但可以将任何访问控制委托给调用的服务。令牌本身可能仅包含一组最小的授权属性，例如，可能仅指示令牌是否可用于调用服务和/或功能。

[0051] 此外，在某些情况下，令牌可以被移交给潜在的不可信组件以执行操作。为了限制该组件故障的潜在影响，提议的概念公开了一种限制令牌授权的优雅方式，以便接收此类令牌的服务仅允许对特定服务类型、服务实例、资源类型和/或资源实例的列表进行操作。这样的列表可能会混合这些不同的类型。目标服务及其授权组件在“幕后”进行了修改，以有效地满足这些限制。相关的授权限制可能仅由该令牌限定，而其他令牌继续根据该用户的完整权限集授权执行操作。

[0052] 还可以注意到，这里使用的令牌可以具有与时间相关的所有时间定界的有效性。因此，也可以使用刷新令牌，其可以作为与访问和/或认证令牌相同的操作原理的基础。

[0053] 因此，所提出概念的一般概念是通过定义令牌范围的特定语义并通过应用令牌语义范围在不改变现有权利的情况下修改访问控制将令牌对资源所有者的现有权利的子集的访问限制为服务实例和/或资源实例。因此，定义的现有权利可能会继续按原样存在；可能只需要定义更多分隔功能和/或服务的附加子集。

[0054] 因此，当前可能仅限制对由服务名称标识的服务的访问的OAuth2的当前可用访问控制可被增强为目前尚不可用且超出OAuth2的多个附加访问限制，比如

[0055] • 对服务中资源类型的受限访问

[0056] <service-name>.<resource-type>，例如kubernetes.pods，

[0057] • 对服务实例中资源类型的受限访问

[0058] <service-name>:<service-instance>.<resource-type>,例如,

[0059] kubernetes:6271293102982712.pods

[0060] • 对服务中资源的受限访问

[0061] <service-name>.<resource-type>:<resource>,例如,kubernetes.pods:mypod,

[0062] • 对服务实例中资源的受限访问

[0063] <service-name>:<service-instance>.<resource-type>:<resource>,例如,kubernetes:6271293102982712.pods:mypod

[0064] 在传统的OAuth2方法中,只有以下访问限制可用:

[0065] • 对服务的访问权限仅限于<service-name>,例如,kubernetes,

[0066] • 对服务操作的访问权限仅限于<service-name>.<action>,

[0067] 例如,database.read。

[0068] 此外,可以仅以受限的权利创建令牌;因此,潜在被盗令牌的破坏性半径较小。例如,处理令牌的移动应用可能会被同一移动设备上的恶意应用调试或入侵。拥有一组受限的权利可能有助于减少被盗令牌的潜在破坏性半径。

[0069] 即使不鼓励,一些Web应用仍将令牌保留在客户端,即在浏览器的Java脚本客户端中。同样在这里,在跨站点脚本攻击的情况下,具有受限权利的令牌具有较小的潜在破坏性半径。

[0070] 根据该方法的优选实施例,标准语义基于OAuth2。因此,这里提出的概念的使用基础可以通过细化或改进已经更粗粒度的访问控制机制来集成到已经明确定义的协议和功能集合中以实现允许更细粒度控制的访问控制机制访问服务和功能。因此,广泛使用的OAuth安全功能(目前在版本2中可用)得到了增强和改进。可以预期,这里提出的增强功能也可能用于更高版本的OAuth。

[0071] 根据该方法的一个有利实施例,应用可以是不安全的组件,例如,Jupyter Notebooks或类似物。Jupyter Notebooks-或Jupyter文档可能表示由已知的Jupyter Notebook App生成的文档,其中可能包含计算机代码和富文本元素以及段落、方程式、数字、链接等。因此,Jupyter Notebook App可以是可以允许例如,通过网络浏览器在服务器系统上编辑和运行笔记本文档的服务器-客户端应用。

[0072] 根据该方法的一个有用实施例,对受限权利的请求可以在认证凭证中包括第一请求,其中第一请求可以包括现有权利的最大子集。因此,初始授权可以允许访问服务和功能的完整集合并且可以不限于子集或受限权利。可以稍后通过请求具受限制的刷新令牌来定义或请求这样的限制。

[0073] 根据该方法的另一个有用实施例,对刷新令牌(和相关的访问令牌)的后续请求—即,在初始请求之后访问应用和/或功能或服务—还可以包括对以下内容的请求:进一步限制的权利。这样,可以以优雅的方式管理第一次访问和后续访问之间的区别。

[0074] 根据该方法的一个许可实施例,该操作可以包括从包括对预定义数据的访问、数据库访问、文件访问、预定义应用编程接口(例如,某些特定分析或机器学习功能),和/或对网络特定子网的访问的组中选择的至少一个。通常,该操作可能与计算环境中可用的任何可用低级功能、服务或资源有关。

[0075] 根据该方法的一个有利实施例,访问令牌可以具有预定义的有效时间。因此,如果访问继续被授权,则可能必须刷新授权,例如,通过相关的刷新令牌。有效时间可以明显短于默认令牌寿命。

[0076] 根据该方法的另一个有利实施例,预定义的有效时间可以由特权用户系统更新。因此,细粒度访问控制可以像对计算环境中的一组资源、功能和/或服务的任何其他访问控制一样是可管理的。现有的IAM系统可用于此任务。

[0077] 在下文中,将给出附图的详细描述。图中的所有说明都是示意性的。首先,给出了用于数据处理环境中基于令牌的授权的本发明的计算机实现的方法的实施例的框图。之后,将描述进一步的实施例,以及用于数据处理环境中基于令牌的授权的访问系统的实施例。

[0078] 图1示出了用于数据处理环境中基于令牌的授权的计算机实现的方法100的实施例的框图。数据处理环境至少包括用户系统、应用、认证服务器和访问控制服务器。用户系统通过网络连接连接到执行应用的服务器,其中应用提供对操作的访问—例如,数据库访问—其中至少该操作可通过其标识符来识别。

[0079] 该方法包括通过用户系统请求访问应用102—例如,通过Web浏览器—重定向104用户访问请求到认证服务器,以及通过在认证服务器和应用之间交换的认证凭证来认证106用户。因此,根据OAuth2.0术语,认证凭证包括使用范围的标准语义(特别是OAuth2)对受限权限的请求,其中受限权利表示由访问控制服务器为资源管理的现有权利的子集。受限权利所需的数据可以嵌入或成为标准语义的一部分,即所使用的标准协议(即OAuth2)。

[0080] 方法100还包括如果认证成功并且应用已经在认证服务器处注册,则从认证服务器向应用发送108访问令牌和刷新令牌,其中访问令牌和刷新令牌包含受限权利。

[0081] 此外,该方法包括请求,110,由用户系统发起的应用执行操作,其中请求包括调用,112,由提供包括受限权利的访问令牌的应用进行的操作,通过操作调用,114,访问控制服务器,向访问控制服务器提供,116,用户系统的标识符和包括现有权利的子集的令牌的范围,以及使用,118,访问控制服务器过滤现有权利的权利子集,导致用户系统对操作的访问决策。

[0082] 在继续图描述之前,应该解释这里使用的受限权利的句法和语义。要了解首选实现中受限权利的语义和语法,了解云资源名称(CRN)的使用(例如,在IBM Cloud及其身份和访问管理系统内部)是有用的。

[0083] 如今的权利、CRN和IAM的特征如下:例如,云环境中的任何服务和资源都可以通过CRN唯一标识。CRN的基本规范格式是:

[0084] crn:<version>:<cname>:<ctype>:<service-name>:<location>:<scope>:

[0085] <service-instance>:<resource-type>:<resource>

[0086] 参数如表1所述。

[0087] 表1

[0088]

| 字段 | 内容 |
| --- | --- |
| <version> | 目前仅支持 v1 |
| <cname> | 对于公共云: bluemix |
| <ctype> | 对于公共云: public |
| <service-name> | 服务类型的唯一标识名称 |
| <location> | 资源的位置。许多可能的值，全局资源为全局性 |
| <scope> | 通常是格式为 a/<account id>的帐户范围 |
| <service-instance> | 服务实例的标识符(如果可用)。可以为空。 |

[0089]

| | |
| --- | --- |
| <resource-type> | 此服务中的可选资源类型。可以为空。 |
| <resource> | 可选的资源标识符。可以为空。 |

[0090] Cloud IAM的访问控制组件将授权存储在策略中。策略允许个人用户访问云计算环境中的服务或资源。示例策略是：

[0091] 身份:IBMid-2700XYZ

[0092] 角色:管理员

[0093] 目标:crn:v1:bluemix:public:kubernetes::a/1234::::

[0094] 该策略允许具有身份“IBMid-2700XYZ”的用户在帐户“1234”中属于“kubernetes”服务类型的任何服务或资源上执行角色“管理员”涵盖的任何操作。

[0095] 当服务想要决定是否允许某个动作时,该服务向访问控制服务器提供以下信息:身份/账户/所需动作/目标资源/范围。

[0096] 今天,首先检查范围是否包含关键字“ibm”或者范围是否包含在目标资源内使用的服务名称。如果没有,策略检查将返回“拒绝(DENY)”。现在,所有潜在的应用策略都根据帐户和身份进行评估。如果找到匹配的策略,策略检查将返回“允许(PERMIT)”。

[0097] 应当清楚的是,特定标识符仅是示例。可以使用任何其他标识符和/或变量而不离开这里提出的概念的范围。

[0098] 图2示出了基于这里提出的概念的交互模型200的框图。资源所有者202—通常是坐Web浏览器前面的最终用户—与客户端204—通常是向最终用户提供服务的Web应用交互。对于交互式登录体验,资源所有者202被重定向到认证服务器206以对客户端的访问请求进行认证和授权。客户端204—通常是Web应用—与认证服务器206交互以检索或刷新访问令牌。此外,作为这里提出的概念的一部分,客户端204还与不可信组件208交互以执行操作。

[0099] 不可信或不安全组件208与资源服务器1、210和资源服务器2、212交互以执行其操作。资源服务器1、210和资源服务器2、212与访问控制服务器214交互以确定所请求的操作是被允许还是被拒绝。

[0100] 在以下优选实施方式中,示例性的两个资源服务器210、212(可以使用更多)正在使用根据表2的示例性的以下服务类型、服务实例、资源类型和资源实例。

[0101] 表2

[0102]

| 资源服务器1,210 | 资源服务器2,212 |
| --- | --- |
| 服务类型: COS<br>服务实例: 1<br>资源类型: bucket<br>资源实例: 123 | 服务类型: COS<br>服务实例: 2<br>资源类型: bucket<br>资源实例: 456 |

[0103] 为了执行这里提出的概念中的交互,可能需要以下步骤:

[0104] 1.客户端204为具有受限权利(新颖概念的一部分)的资源所有者202(由OAuth2标准覆盖)检索访问令牌和刷新令牌。

[0105] 2.客户端204调用对不可信组件的操作(Web环境中的正常行为)。

[0106] 3.不可信组件208与资源服务器1、210和2、212交互(Web环境中的正常行为)。

[0107] 4.资源服务器1、210和资源服务器2、212与访问控制服务器214交互以决定是否应该允许所请求的操作(新颖概念的一部分)。

[0108] 因为先决条件可以被命名为:资源所有者202被分配了访问控制服务器214上的策略,该策略允许资源所有者202访问资源服务器1、210和资源服务器2、212两者上的资源。这些策略不是在以下交互过程中发生了变化,即这里新提出的概念的一部分。

[0109] 图3示出了替代发起交互的实施例的框图300:客户端204检索具有受限权利的资源所有者202的访问和/或刷新令牌。客户端204通过在OAuth2标准中描述的授权类型之一检索资源所有者202的访问和/或刷新令牌。此外,作为新颖概念的一部分,客户端204已经将权利列表传递给授权服务器206以限制对要生成的访问令牌的潜在访问。在令牌创建期间指定范围列表是OAuth2标准的一部分,但范围的语义和后续交互中的用法是此处新提出的概念的一部分。对于此示例交互,可以假设受限权利列表只是一个元素“COS:1.bucket:123”。

[0110] 图4示出了相关的不言自明的通信图400,示出了具有相应消息流的客户端204、认证服务器206和资源所有者202。

[0111] 图5示出了替代交互500的框图,其中客户端204检索具有受限权利的资源所有者202的访问令牌和/或刷新令牌。客户端204通过在OAuth2标准中描述的授权类型之一检索资源所有者202的访问和/或刷新令牌。在随后的步骤中,客户端204使用refresh_token授权类型来刷新访问和/或刷新令牌并且将受限权利的列表传递到授权服务器206以限制访问令牌和要生成的刷新令牌的潜在访问。在令牌刷新期间指定范围列表是OAuth2标准的一部分,但范围的语义和后续交互中的用法是此处新提出的概念的一部分。对于此示例交互,可以假设受限权利列表只是一个元素“COS:1.bucket:123”。

[0112] 图6示出了客户端204通过调用对不安全组件208的操作的第二类型交互600的框图。在根据图3、4、5的交互之后,存在携带一组受限的权利。在本描述中,可以假设它包括访问资源服务器2、212但不包括访问资源服务器1、210的权利。

[0113] 客户端204调用对不安全或不可信组件208的操作。不可信组件208是例如已经由其他人开发并且未被审查或验证为可信的脚本。不可信组件208的调用包括具有受限权利的访问令牌,这是这里提出的概念的一部分。

[0114] 图7示出了相关的不言自明的通信图700,示出了具有相应消息流和处理的客户端204和不安全组件208。

[0115] 图8示出了第三类交互的框图800,其中不可信组件208与资源服务器1、210和资源服务器2、212交互。作为不可信组件208的执行的一部分,脚本在资源服务器1上开始操作,210提供具有受限权利的访问令牌。作为正常处理的一部分,资源服务器1、210与访问控制服务器214交互以确定由访问令牌表示的资源所有者是否可以执行该操作。资源服务器1、210在访问令牌内提供身份、访问令牌的权利(范围)(COS:1.bucket:123)和目标资源(例如,crn:v1:bluemix:public:COS::a/1234:1:bucket:123)以执行操作,这是此处提出的概念的显式组件。

[0116] 基于访问令牌内的权利和现有策略,访问控制服务器214确定允许操作。与资源服务器2、212的类似交互返回“拒绝(DENY)”,因为访问令牌(COS:1.bucket:123)的权利(范围)与目标资源(crn:v1:blumix:public:COS::a/1234:2:bucket:456)不匹配。

[0117] 图9示出了相关的不言自明的通信图900,示出了不可信组件208以及资源服务器1、210和资源服务器2、212以及各自的消息流和处理。

[0118] 图10示出了第四类交互1000的框图,其中资源服务器1、210和资源服务器2、212与访问控制服务器214交互。不可信组件208开始对资源服务器1、210的操作,然后在资源服务器2、212上提供具有受限权利的访问令牌。再次,作为正常OAuth2处理的一部分,资源服务器210、212与访问控制服务器214交互以确定是否允许执行该操作。资源服务器210、212正在提供访问令牌和目标资源内部的身份和权利(范围)以执行操作。

[0119] 由于令牌内的受限权利不包含资源服务器2、212的服务实例,访问控制服务器214将决定拒绝资源服务器2、212的操作,即使访问内的身份的策略令牌将在系统中具有匹配策略,这也是此处新提出的概念的明确部分。

[0120] 图11示出了相关的不言自明的通信图1100,示出了具有各自消息流和处理的资源服务器1、210、资源服务器2、212和访问控制服务器214。

[0121] 接下来的两个图12、13示出了传统访问控制服务器决策引擎(图12)的伪代码段与这里提出的概念所需的伪代码段(图13)的比较。很容易看出,需要对限制权利的正确细粒度访问控制进行更复杂的确定。但是,与标准的OAuth2访问控制相比,它保证了更灵活和安全的资源管理。

[0122] 出于完整性原因,图14示出了用于数据处理环境中基于令牌的认证的访问系统1400的框图。数据处理环境至少包括用户系统、应用、认证服务器和访问控制服务器(均未示出)。从而,用户系统通过网络连接连接到执行应用的服务器,并且应用提供对操作的访问。至少该操作可以通过其标识符进行识别。还可以注意到,这里讨论的所有组件都连接到连接这些组件的网络。

[0123] 接入系统1400包括:接入模块1402,用于通过用户系统请求接入应用;重定向模块1404,用于将用户接入请求重定向到认证服务器206。认证服务器206,用于对用户进行认证通过在认证服务器和应用之间交换的认证凭证,其中认证凭证包括使用范围的标准语义的对受限权利的请求,其中受限权利表示由访问控制服务器管理的资源的现有权利的子集.

[0124] 访问系统1400还包括发送器,该发送器适用于如果认证成功并且应用已经在认证服务器206处注册,则从认证服务器206向应用发送访问令牌和刷新令牌,其中访问令牌和刷新令牌包括受限制的权利。

[0125] 因此,用户系统适于请求由用户系统发起的应用执行操作,其中请求执行操作包

括通过提供包括受限权利的访问令牌的应用调用操作，通过操作调用访问控制服务器，向访问控制服务器214提供用户系统的标识符和包括现有权利的子集的令牌的范围，访问控制服务器214使用权利子集过滤现有权利，导致用户系统对操作的访问决策。

[0126] 本发明的实施例可以与几乎任何类型的计算机一起实现，而不管平台适合于存储和/或执行程序代码。作为示例，图15示出了适合于执行与所提出的方法相关的程序代码的计算系统1500。实际上，处理环境的任何组件都可以由根据图15的计算系统表示。

[0127] 计算系统1500仅是合适的计算机系统的一个示例，并且不旨在暗示对本文描述的本发明的实施例的使用范围或功能性的任何限制，无论计算机系统1500是否能够被实施和/或执行任何上述功能。在计算机系统1500中，存在与许多其他通用或专用计算系统环境或配置一起操作的组件。可适用于计算机系统/服务器1500的众所周知的计算系统、环境和/或配置的示例包括但不限于个人计算机系统、服务器计算机系统、瘦客户端、胖客户端、手动手持设备或膝上型设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费电子产品、网络PC、小型计算机系统、大型计算机系统和包括任何上述系统或设备的分布式云计算环境等。计算机系统/服务器1500可以在由计算机系统1500执行的计算机系统可执行指令的一般上下文中描述，例如程序模块。通常，程序模块可以包括例程、程序、对象、组件、逻辑、数据结构，等等，执行特定任务或实现特定抽象数据类型。计算机系统/服务器1500可以在分布式云计算环境中实践，其中任务由通过通信网络链接的远程处理设备执行。在分布式云计算环境中，程序模块可以位于本地和远程计算机系统存储介质中，包括内存存储设备。

[0128] 如图所示，计算机系统/服务器1500以通用计算设备的形式示出。计算机系统/服务器1500的组件可以包括但不限于一个或多个处理器或处理单元1502、系统存储器1504和将包括系统存储器1504在内的各种系统组件耦合到处理器1502的总线1506。1506代表多种类型的总线结构中的一种或多种，包括存储器总线或存储器控制器、外围总线、加速图形端口以及使用各种总线架构中的任一种的处理器或本地总线。作为示例而非限制，此类架构包括工业标准架构(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)本地总线和外围组件互连(PCI)总线。计算机系统/服务器1500通常包括各种计算机系统可读介质。这种介质可以是可由计算机系统/服务器1500访问的任何可用介质，并且它包括易失性和非易失性介质、可移动和不可移动介质。

[0129] 系统存储器1504可以包括易失性存储器形式的计算机系统可读介质，例如随机存取存储器(RAM)1508和/或高速缓存存储器1510。计算机系统/服务器1500还可以包括其他可移动/不可移动，易失性/非易失性计算机系统存储介质。仅作为示例，可以提供存储系统1512用于从不可移动、非易失性磁介质(未示出并且通常称为“硬盘驱动器”)读取和写入。尽管未示出，用于读取和写入可移动非易失性磁盘(例如“软盘”)的磁盘驱动器，以及用于读取或写入可移动非易失性光盘的光盘驱动器可以提供诸如CD-ROM、DVD-ROM或其他光学介质的盘。在这种情况下，每个都可以通过一个或多个数据媒体接口连接到总线1506。如下文将进一步描绘和描述的，存储器1504可以包括至少一个程序产品，该程序产品具有一组(例如，至少一个)程序模块，该程序模块被配置为执行本发明的实施例的功能。

[0130] 具有一组(至少一个)程序模块1516的程序/实用程序可以作为示例而非限制地存储在存储器1504中，以及操作系统、一个或多个应用程序、其他程序模块和程序数据。操作

系统、一个或多个应用程序、其他程序模块和程序数据或它们的某种组合中的每一个可以包括联网环境的实现。如本文所述，程序模块1516通常执行本发明的实施例的功能和/或方法。

[0131] 计算机系统/服务器1500还可以与一个或多个外部设备1518通信，例如键盘、定点设备、显示器1520等；使用户能够与计算机系统/服务器1500交互的一个或多个设备；和/或使计算机系统/服务器1500能够与一个或多个其他计算设备通信的任何设备(例如，网卡、调制解调器等)。这种通信可以通过输入/输出(I/O)接口1514发生。然而，计算机系统/服务器1500可以与一个或多个网络通信，例如局域网(LAN)、通用广域网(WAN)、和/或经由网络适配器1522的公共网络(例如，因特网)。如图所示，网络适配器1522可以经由总线1506与计算机系统/服务器1500的其他组件通信。应当理解，尽管未示出，其他硬件和/或软件组件可以与计算机系统/服务器1500结合使用。示例包括但不限于：微代码、设备驱动程序、冗余处理单元、外部磁盘驱动器阵列、RAID系统、磁带驱动器和数据归档存储系统等。

[0132] 本发明的各种实施例的描述是出于说明的目的而呈现的，但并非旨在穷举或限于所公开的实施例。在不脱离所描述的实施例的范围和精神的情况下，许多修改和变化对于本领域的普通技术人员来说将是显而易见的。选择此处使用的术语以最好地解释实施例的原理、实际应用或对市场中发现的技术的技术改进，或者使本领域的其他普通技术人员能够理解此处公开的实施例。

[0133] 本发明可以是系统、方法和/或计算机程序产品。计算机程序产品可以包括计算机可读存储介质，其上载有用于使处理器实现本发明的各个方面的计算机可读程序指令。

[0134] 介质可以是用于传播介质的电、磁、光、电磁、红外或半导体系统。计算机可读介质的示例可以包括半导体或固态存储器、磁带、可移动计算机磁盘、随机存取存储器(RAM)、只读存储器(ROM)、硬磁盘和光盘。光盘的当前示例包括光盘只读存储器(CD-ROM)、光盘读/写(CD-R/W)、DVD和蓝光光盘。

[0135] 计算机可读存储介质可以是可以保持和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质例如可以是——但不限于——电存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或者上述的任意合适的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括：便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式压缩盘只读存储器(CD-ROM)、数字多功能盘(DVD)、记忆棒、软盘、机械编码设备、例如其上存储有指令的打孔卡或凹槽内凸起结构、以及上述的任意合适的组合。这里所使用的计算机可读存储介质不被解释为瞬时信号本身，诸如无线电波或者其他自由传播的电磁波、通过波导或其他传输媒介传播的电磁波(例如，通过光纤电缆的光脉冲)、或者通过电线传输的电信号。

[0136] 这里所描述的计算机可读程序指令可以从计算机可读存储介质下载到各个计算/处理设备，或者通过网络、例如因特网、局域网、广域网和/或无线网下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光纤传输、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或者网络接口从网络接收计算机可读程序指令，并转发该计算机可读程序指令，以供存储在各个计算/处理设备中的计算机可读存储介质中。

[0137] 用于执行本发明操作的计算机程序指令可以是汇编指令、指令集架构(ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、集成电路配置数据或者以一种或多种编程语言的任意组合编写的源代码或目标代码，所述编程语言包括面向对象的编程语言—诸如Smalltalk、C++等，以及过程式编程语言—诸如“C”语言或类似的编程语言。计算机可读程序指令可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络—包括局域网(LAN)或广域网(WAN)—连接到用户计算机，或者，可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。在一些实施例中，通过利用计算机可读程序指令的状态信息来个性化定制电子电路，例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA)，该电子电路可以执行计算机可读程序指令，从而实现本发明的各个方面。

[0138] 这里参照根据本发明实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述了本发明的各个方面。应当理解，流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合，都可以由计算机可读程序指令实现。

[0139] 这些计算机可读程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器，从而生产出一种机器，使得这些指令在通过计算机或其它可编程数据处理装置的处理器执行时，产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中，这些指令使得计算机、可编程数据处理装置和/或其他设备以特定方式工作，从而，存储有指令的计算机可读介质则包括一个制造品，其包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的各个方面的指令。

[0140] 也可以把计算机可读程序指令加载到计算机、其它可编程数据处理装置、或其它设备上，使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤，以产生计算机实现的过程，从而使得在计算机、其它可编程数据处理装置、或其它设备上执行的指令实现流程图和/或框图中的一个或多个方框中规定的功能/动作。

[0141] 附图中的流程图和框图显示了根据本发明的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分，所述模块、程序段或指令的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。在有些作为替换的实现中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或动作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0142] 此处使用的术语仅用于描述特定实施例的目的，并不旨在限制本发明。如本文所用，单数形式“一个”和“这个”、“那个”也旨在包括复数形式，除非上下文另有明确指示。将进一步理解，术语“包含”和/或“包括”，当在本说明书中使用时，指定所述特征、整数、步骤、操作、元素和/或组件的存在，但不排除存在或添加一个或多个其他特征、整数、步骤、操作、元素、组件和/或它们的组。

[0143] 权利要求中的所有装置或步骤加功能元件的相应结构、材料、动作和等效物旨在包括用于与如具体要求保护的其他要求保护的元件组合执行功能的任何结构、材料或动作。本发明的描述是为了说明和描述的目的而呈现的，但并不旨在穷举或限制所公开形式的本发明。在不脱离本发明的范围的情况下，基于本文的教导，对本领域普通技术人员而言，许多修改和变化将是显而易见的。描述该主题是为了解释本发明的原理和实际应用，并使本领域的其他普通技术人员能够理解本发明的各种实施例以及适合于预期的特定用途的各种修改。

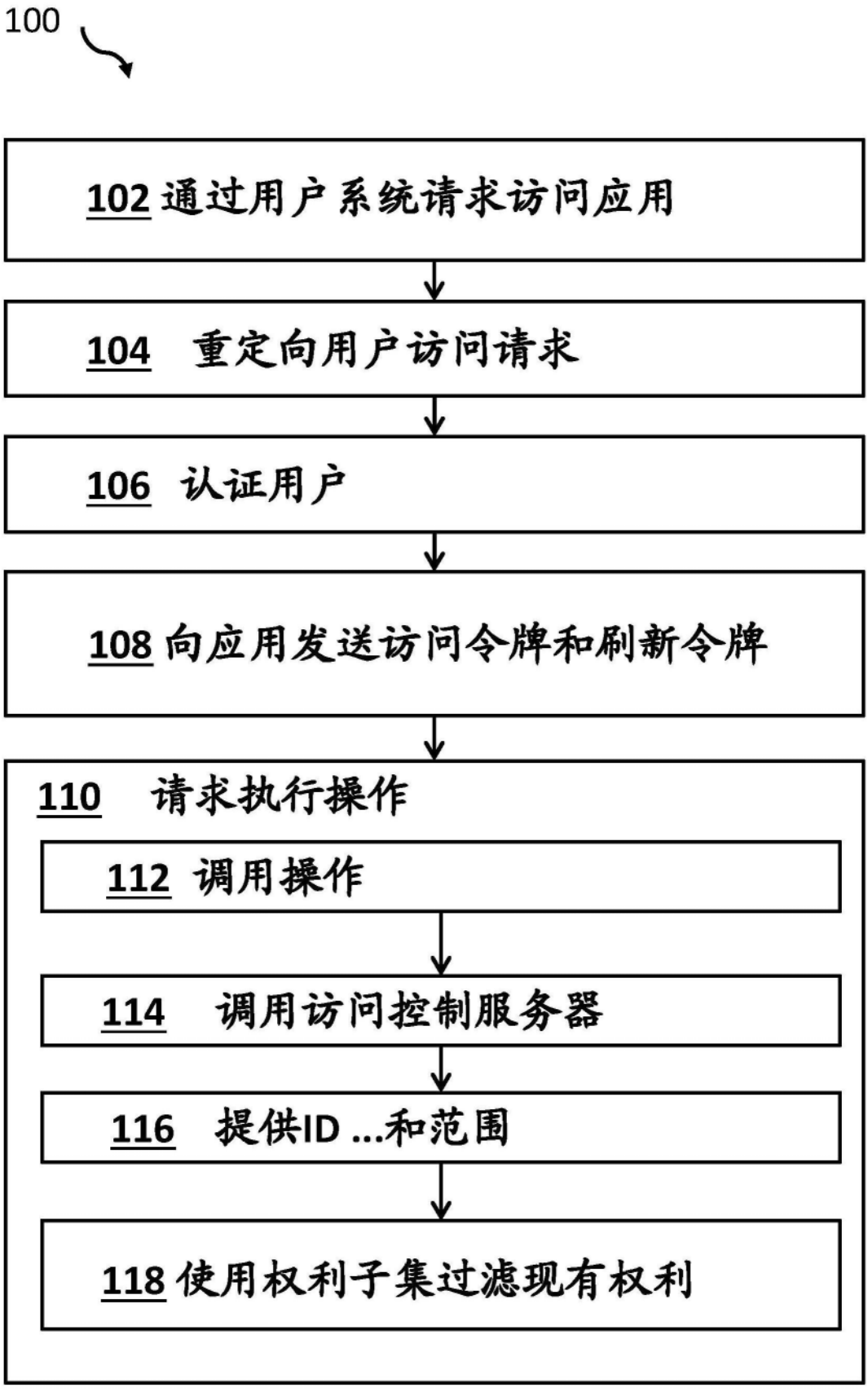

100
102 通过用户系统请求访问应用
104 重定向用户访问请求
106 认证用户
108 向应用发送访问令牌和刷新令牌
110 请求执行操作
112 调用操作
114 调用访问控制服务器
116 提供ID ...和范围
118 使用权利子集过滤现有权利

图1

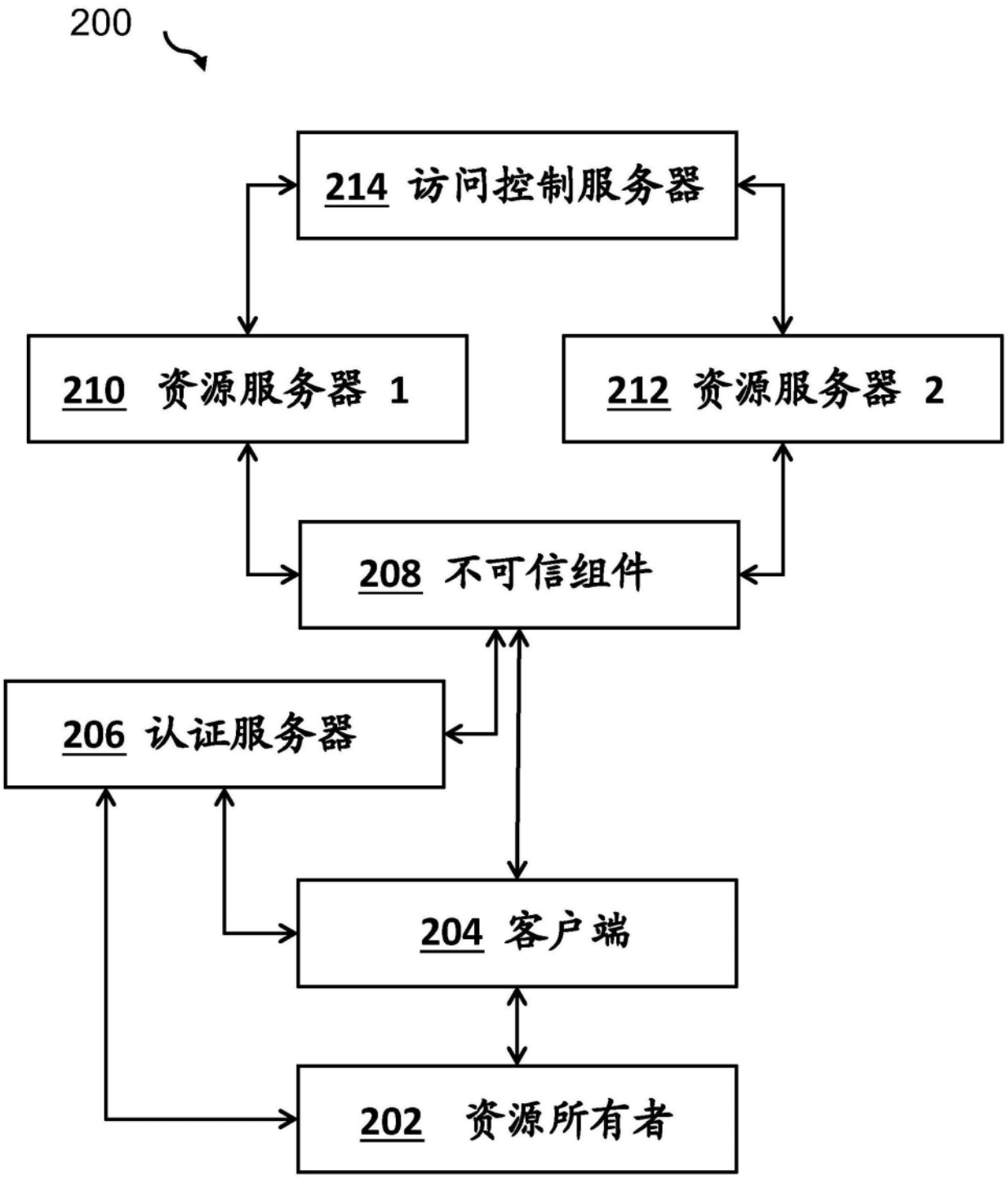

200
214 访问控制服务器
210 资源服务器 1
212 资源服务器 2
208 不可信组件
206 认证服务器
204 客户端
202 资源所有者

图2

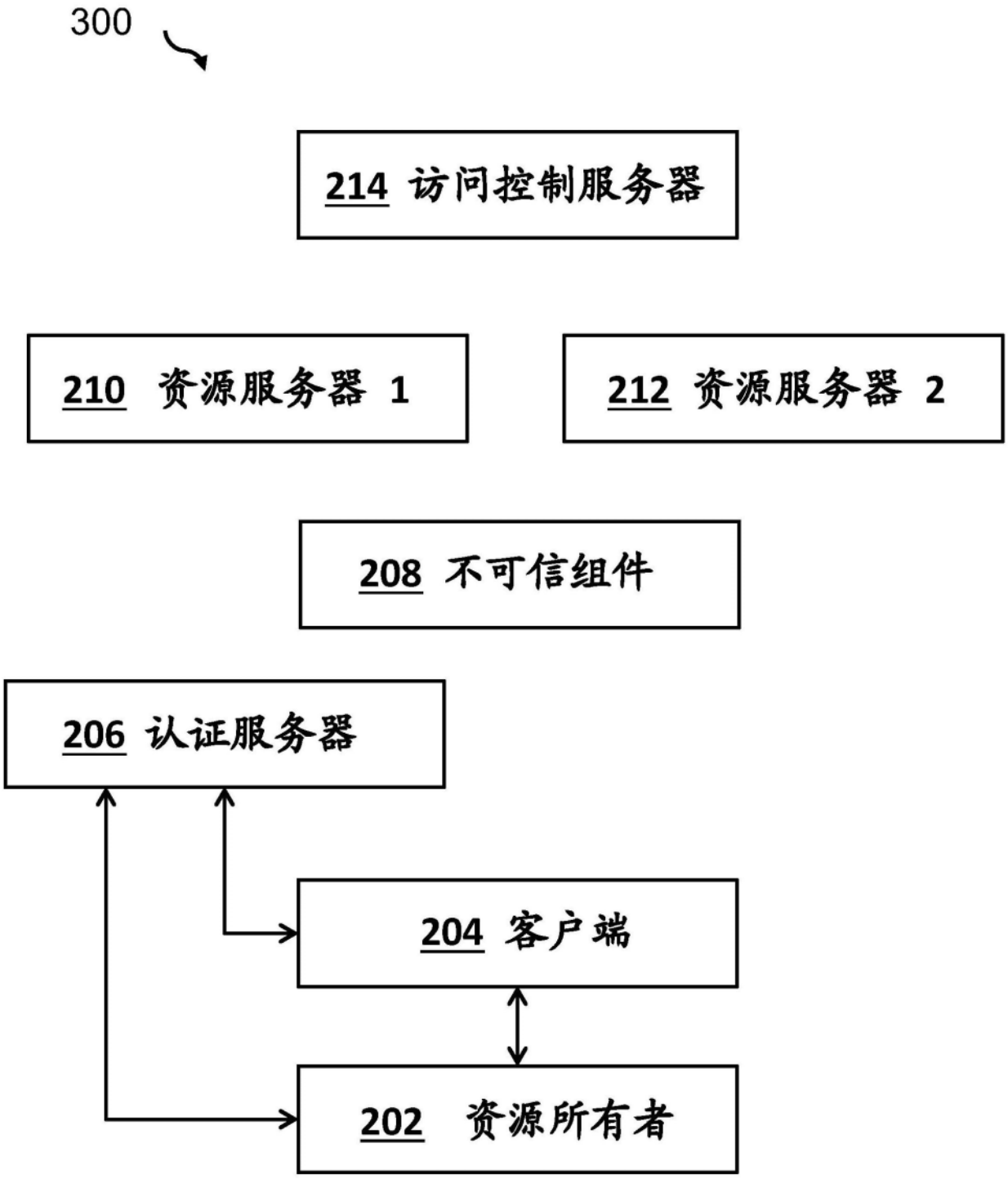

300
214 访问控制服务器
210 资源服务器 1
212 资源服务器 2
208 不可信组件
206 认证服务器
204 客户端
202 资源所有者

图3

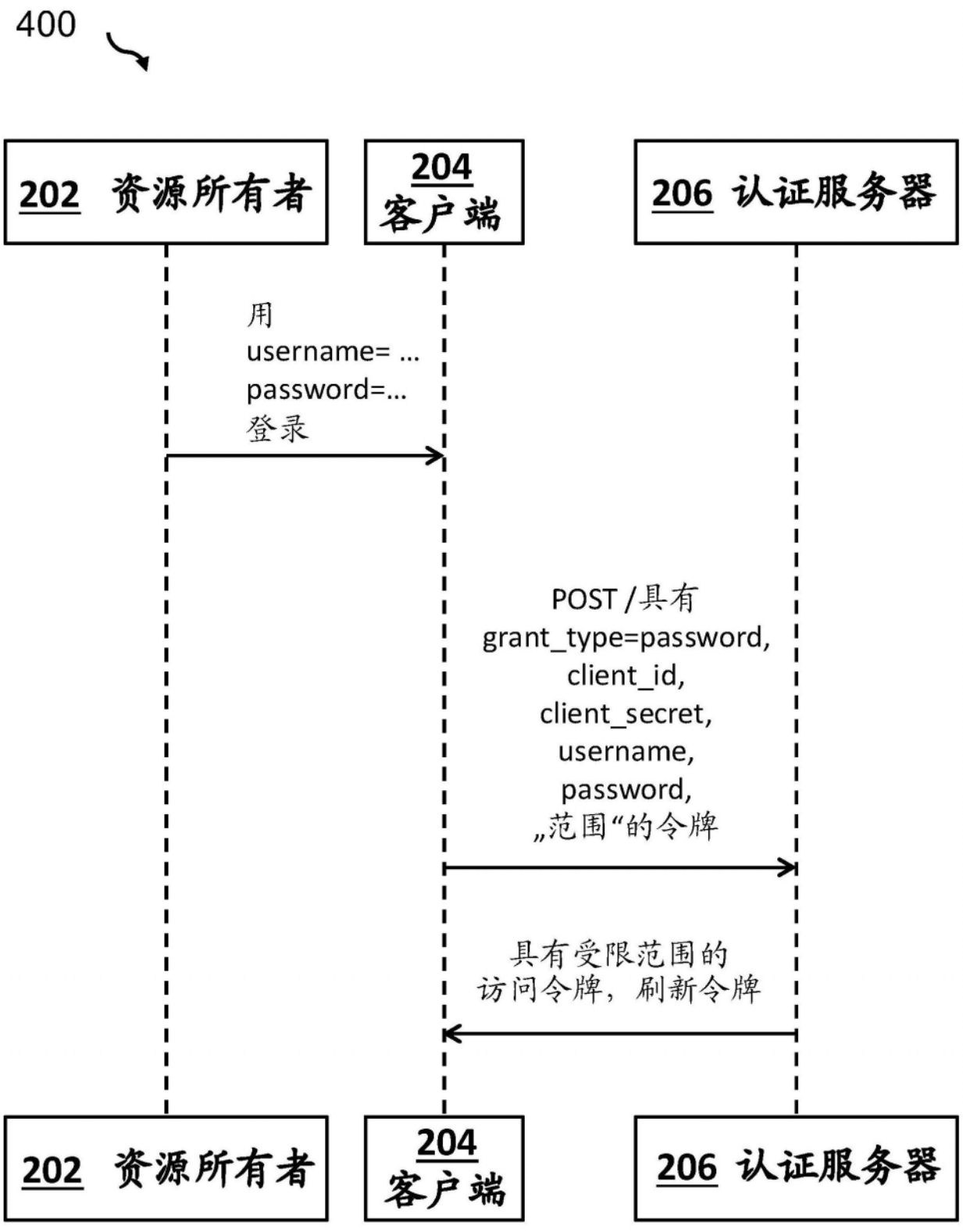

400
202 资源所有者
204 客户端
206 认证服务器
用
username= ...
password=...
登录
POST /具有
grant_type=password,
client_id,
client_secret,
username,
password,
„范围“的令牌
具有受限范围的
访问令牌，刷新令牌
202 资源所有者
204 客户端
206 认证服务器

图4

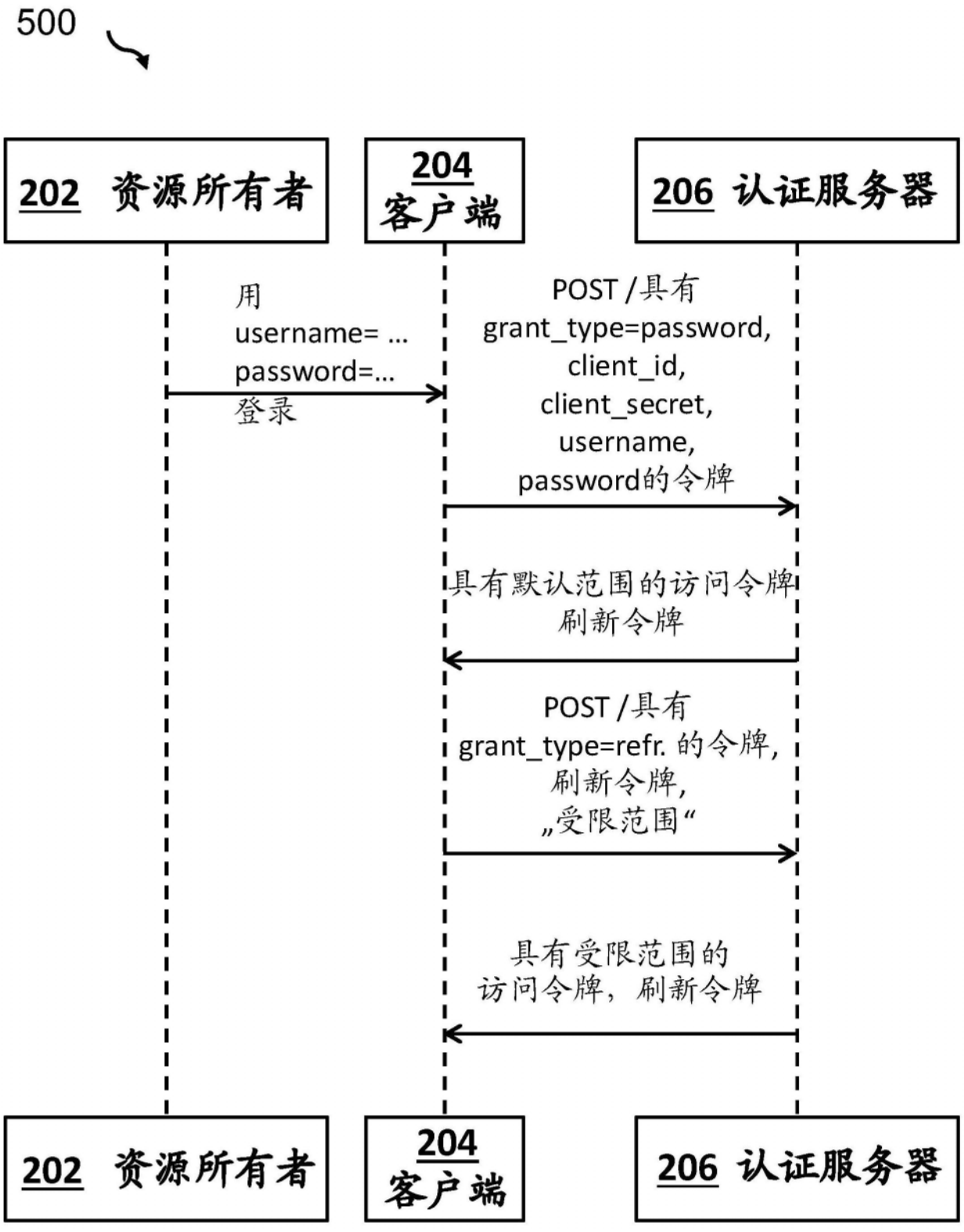

500
202 资源所有者
204 客户端
206 认证服务器
用 username= ... password=... 登录
POST /具有 grant_type=password, client_id, client_secret, username, password的令牌
具有默认范围的访问令牌 刷新令牌
POST /具有 grant_type=refr. 的令牌, 刷新令牌, „受限范围“
具有受限范围的 访问令牌，刷新令牌
202 资源所有者
204 客户端
206 认证服务器

图5

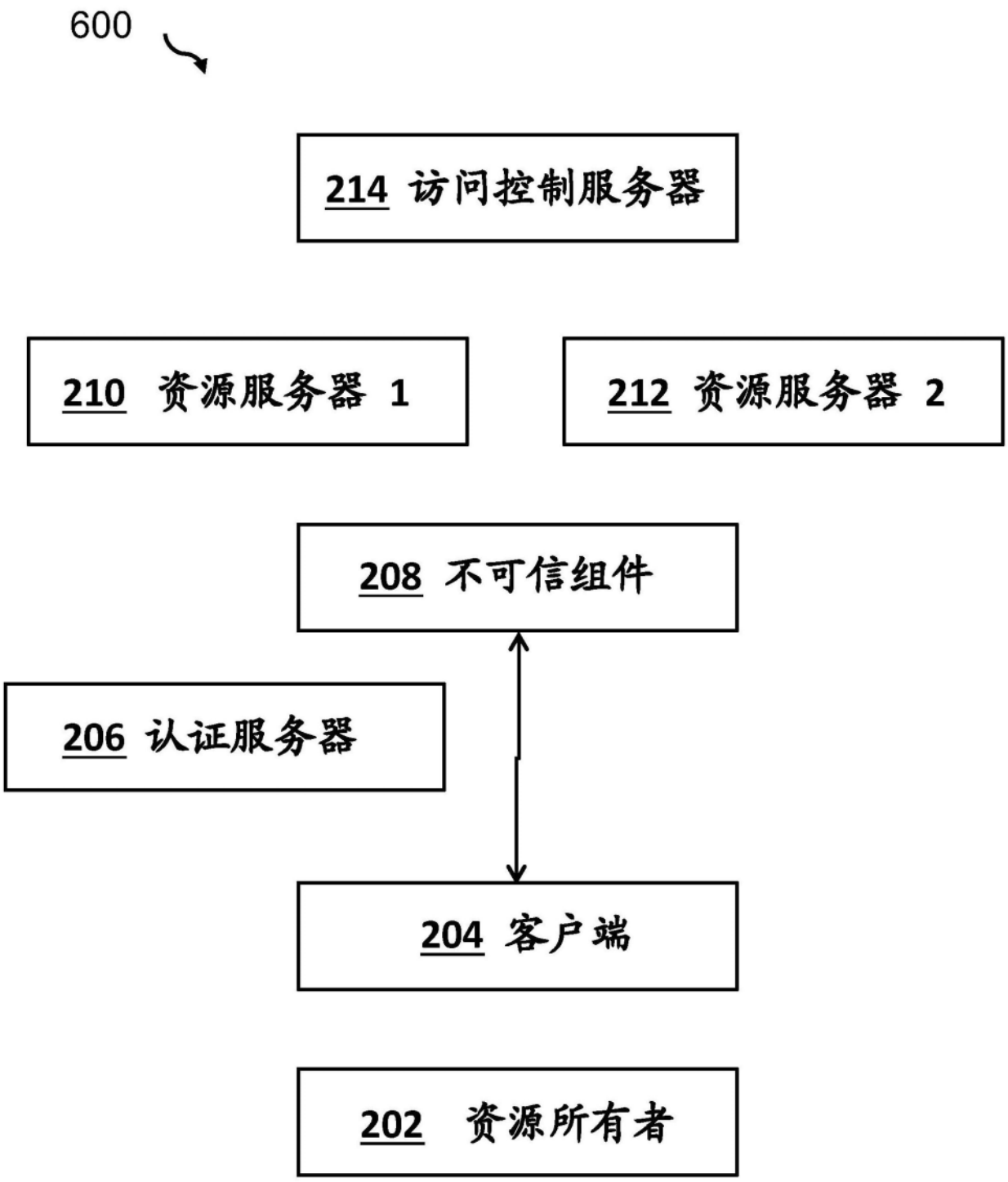

600
214 访问控制服务器
210 资源服务器 1
212 资源服务器 2
208 不可信组件
206 认证服务器
204 客户端
202 资源所有者

图6

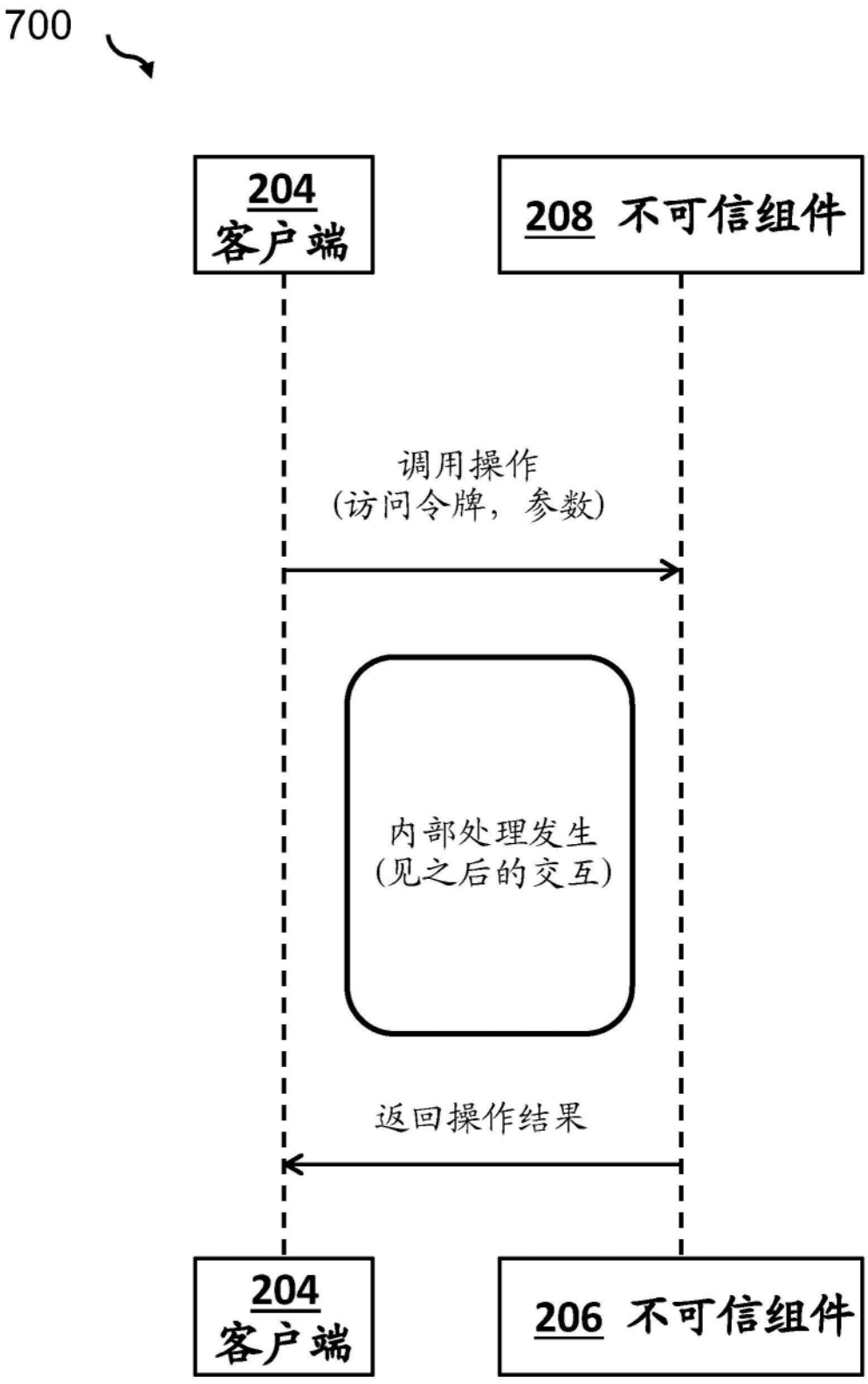

700
204
客户端
208 不可信组件
调用操作
(访问令牌，参数)
内部处理发生
(见之后的交互)
返回操作结果
204
客户端
206 不可信组件

图7

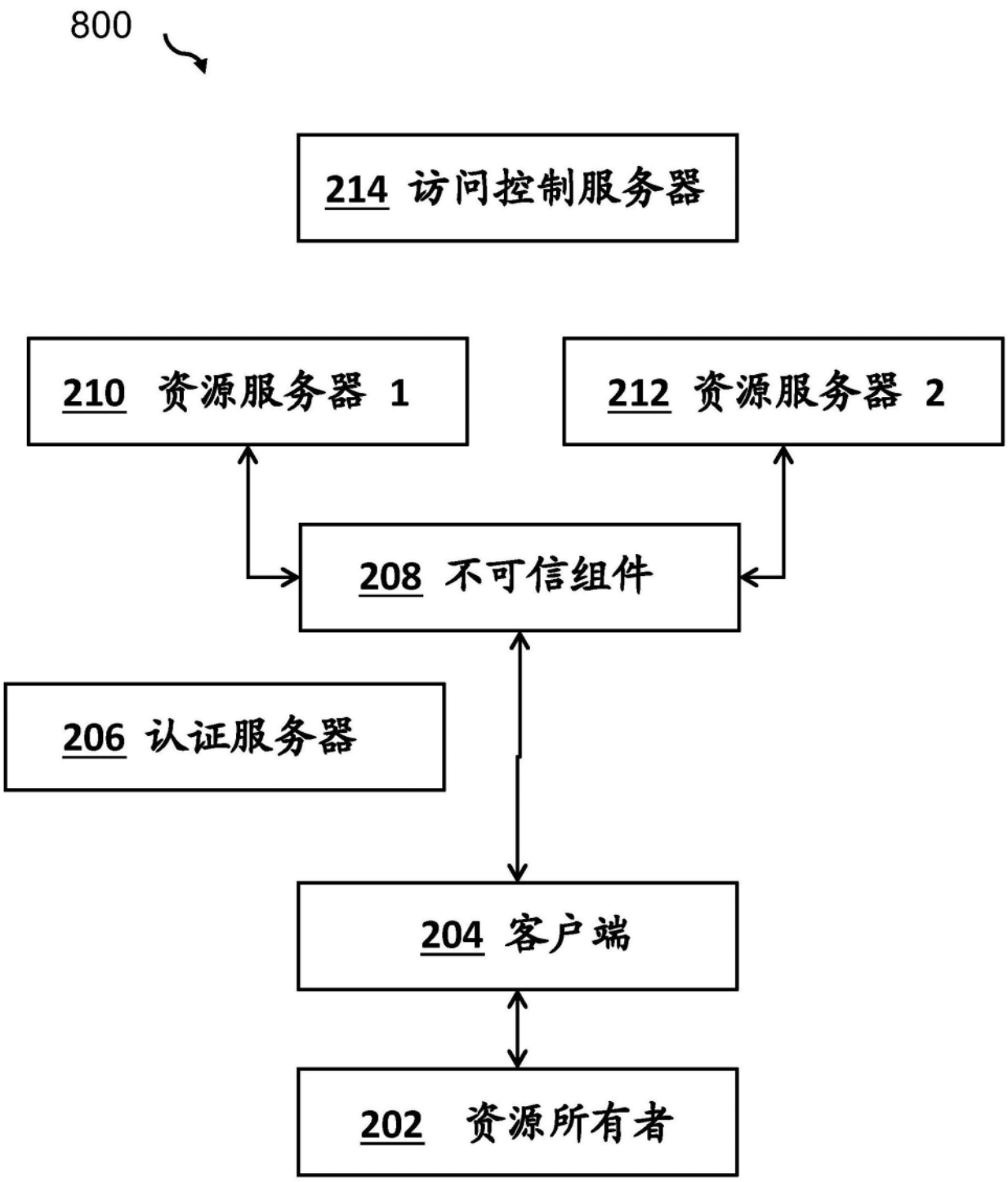

800
214 访问控制服务器
210 资源服务器 1
212 资源服务器 2
208 不可信组件
206 认证服务器
204 客户端
202 资源所有者

图8

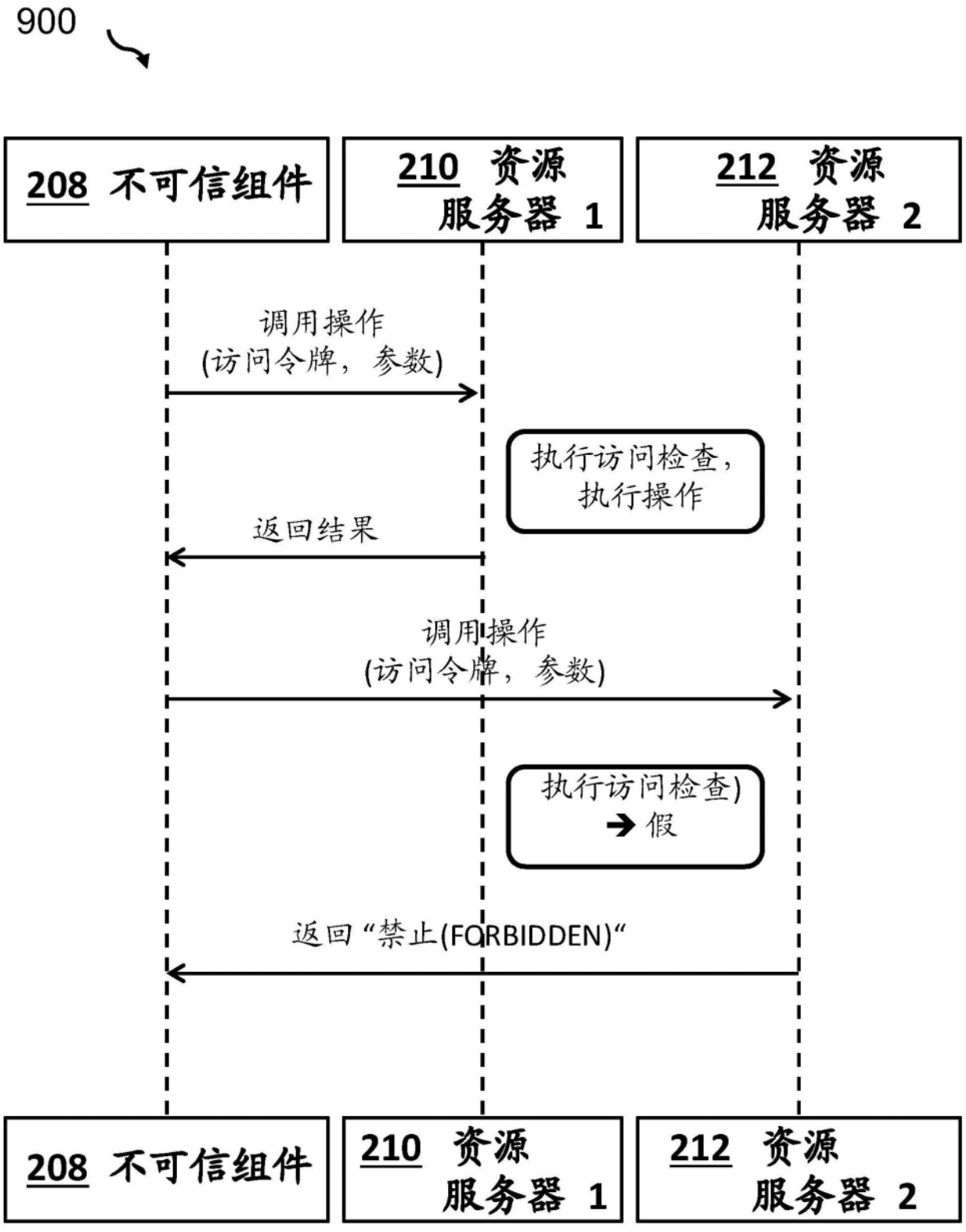

900
208 不可信组件
210 资源服务器 1
212 资源服务器 2
调用操作
(访问令牌，参数)
执行访问检查，
执行操作
返回结果
调用操作
(访问令牌，参数)
执行访问检查)
➔ 假
返回“禁止(FORBIDDEN)“
208 不可信组件
210 资源服务器 1
212 资源服务器 2

图9

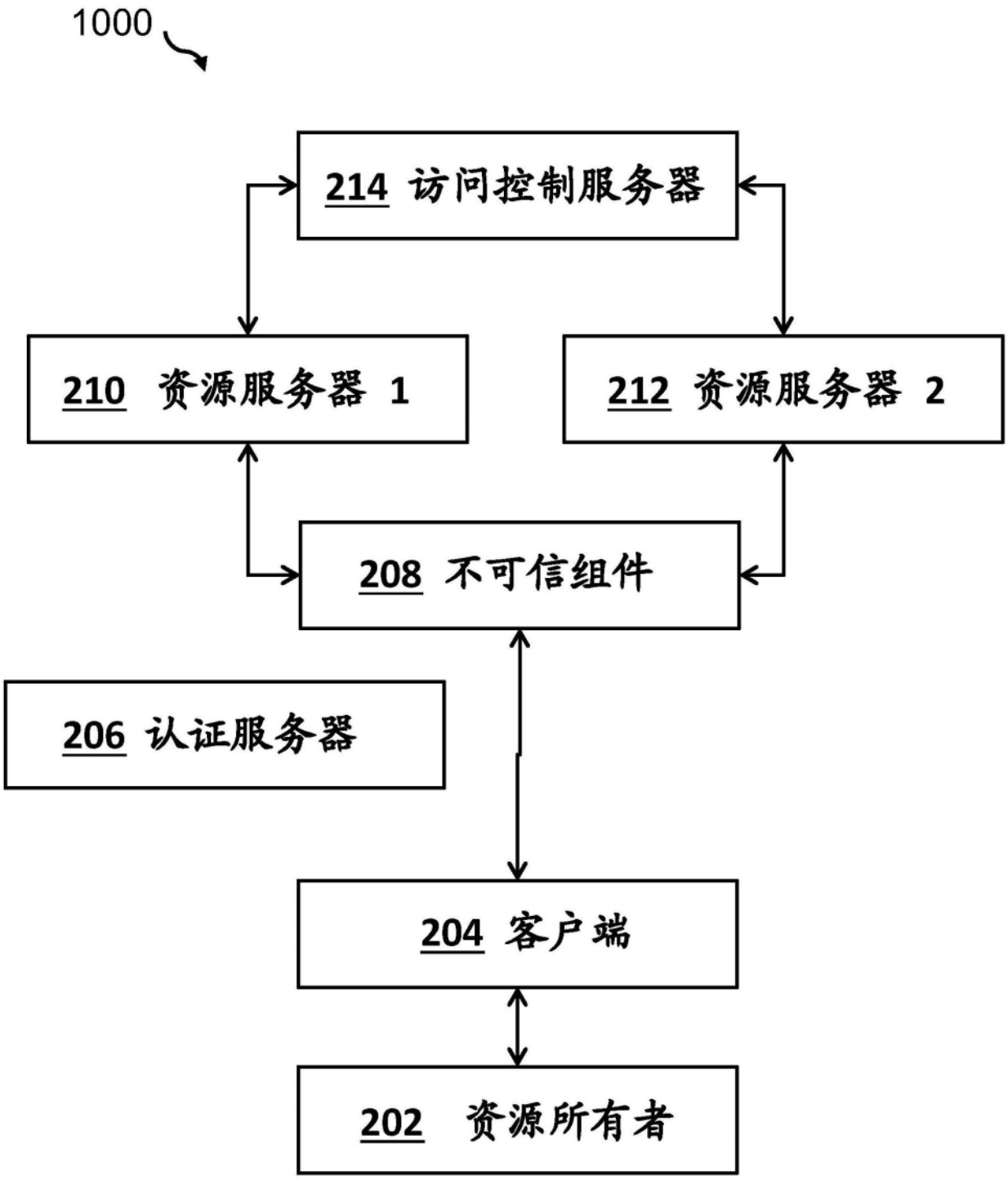

1000
214 访问控制服务器
210 资源服务器 1
212 资源服务器 2
208 不可信组件
206 认证服务器
204 客户端
202 资源所有者

图10

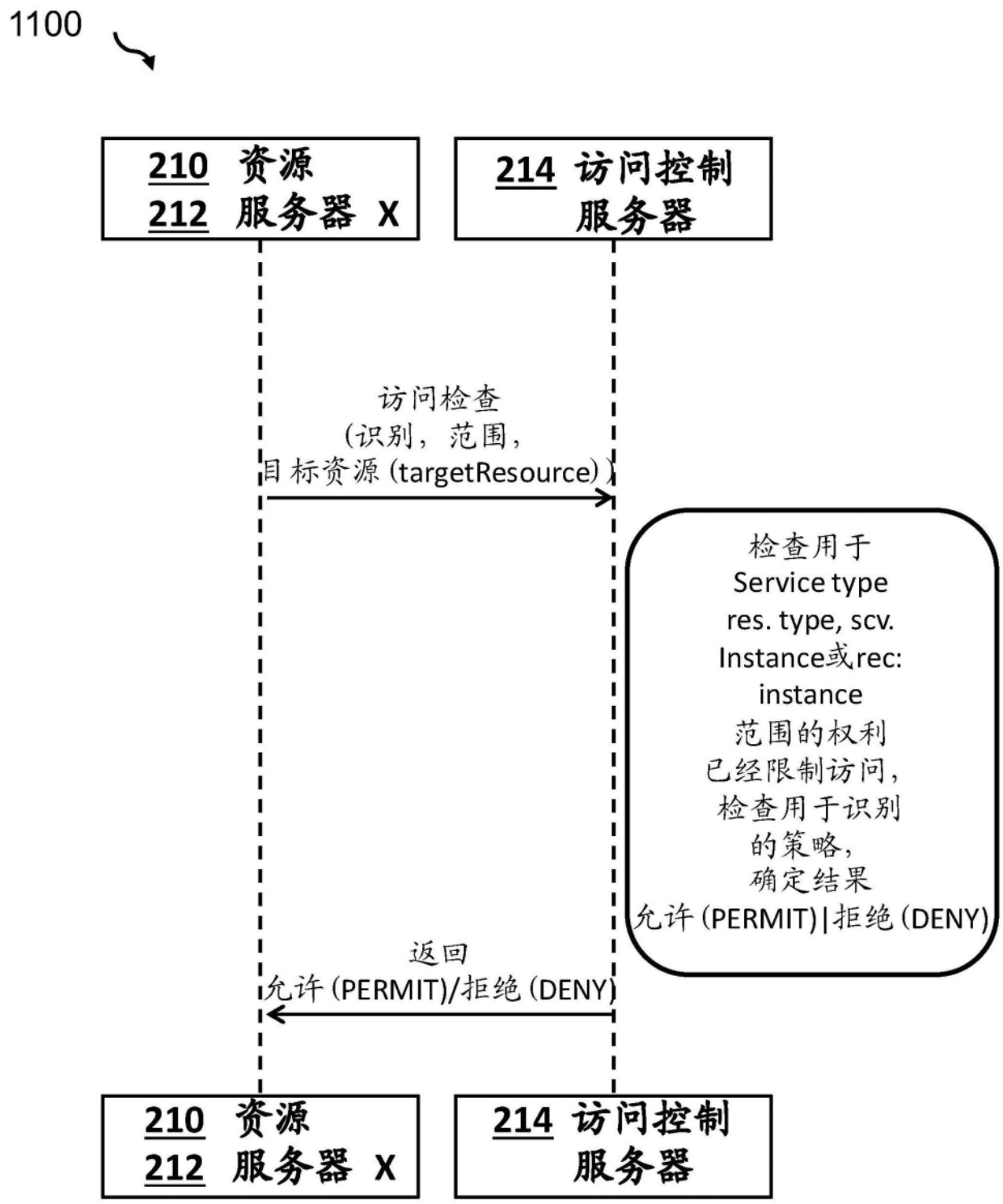

1100
210 资源
212 服务器 X
214 访问控制
服务器
访问检查
(识别，范围，
目标资源(targetResource))
检查用于
Service type
res. type, scv.
Instance或rec:
instance
范围的权利
已经限制访问，
检查用于识别
的策略，
确定结果
允许(PERMIT)|拒绝(DENY)
返回
允许(PERMIT)/拒绝(DENY)
210 资源
212 服务器 X
214 访问控制
服务器

图11

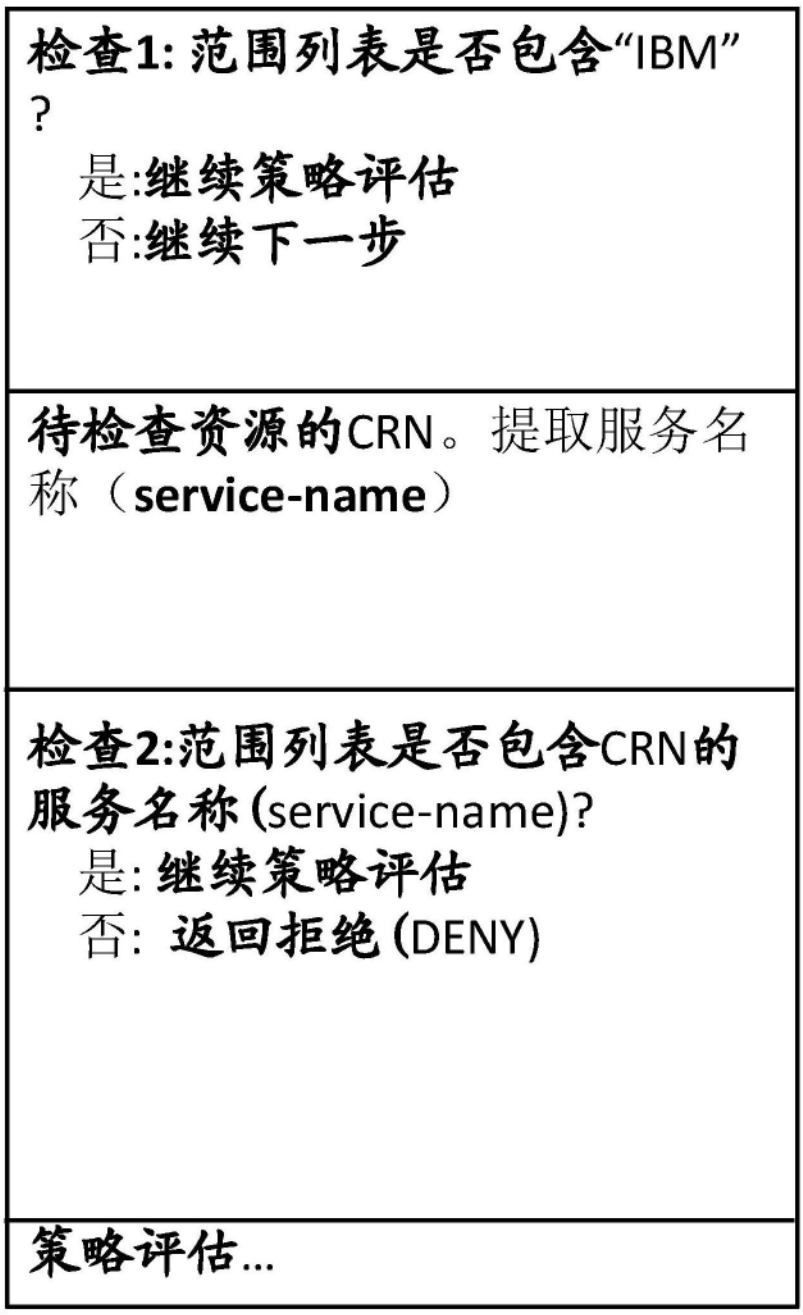

检查1: 范围列表是否包含“IBM”?
是:继续策略评估
否:继续下一步
待检查资源的CRN。提取服务名称（service-name）
检查2:范围列表是否包含CRN的服务名称(service-name)?
是: 继续策略评估
否: 返回拒绝(DENY)
策略评估...

图12

**检查1: 范围列表是否包含"IBM"?**
是:**继续策略评估**
否:**继续下一步**

**解析待检查资源的CRN。有关CRN语法的详细信息，请参见"受限权利的语法和语义"部分。**
如果可用提取: **crn-service-name, crn-service-instance, crn-resource-type, crn-resource**

**检查2: 对于范围的列表中的每一个范围，执行:**
**有关范围语法的详细信息可以在"受限权利的语法和语义"部分找到。**
**将范围分割到 scope-service-name, scope-service-instance, scope-resource-type, scope-resource；个别部分可以是空的。**

**检查**
(**crn-service-name** matches **scope-service-name**)
AND
( (**crn-service-instance** not available and **scope-service-name** not available) OR
(**crn-service-instance** available and **scope-service-name** not available) OR
(**crn-service-instance** available and matches **scope-service-name**))
AND
((**crn-resource-type** not available and **scope-resource-type** not available) OR
(**crn-resource-type** available and **scope-resource-type** not available) OR
(**crn-resource-type** available and matches **scope-resource-type**))
AND
((**crn-resource** not available and **scope-resource** not available) OR
(**crn-resource** available and **scope-resource** not available) OR
(**crn-resource** available and matches **scope-resource**))

**实际范围是否匹配该检查?**
是:**继续策略评估**
否:**继续下一范围**
**没有找到匹配范围?返回拒绝** (DENY)

**策略评估...**

图13

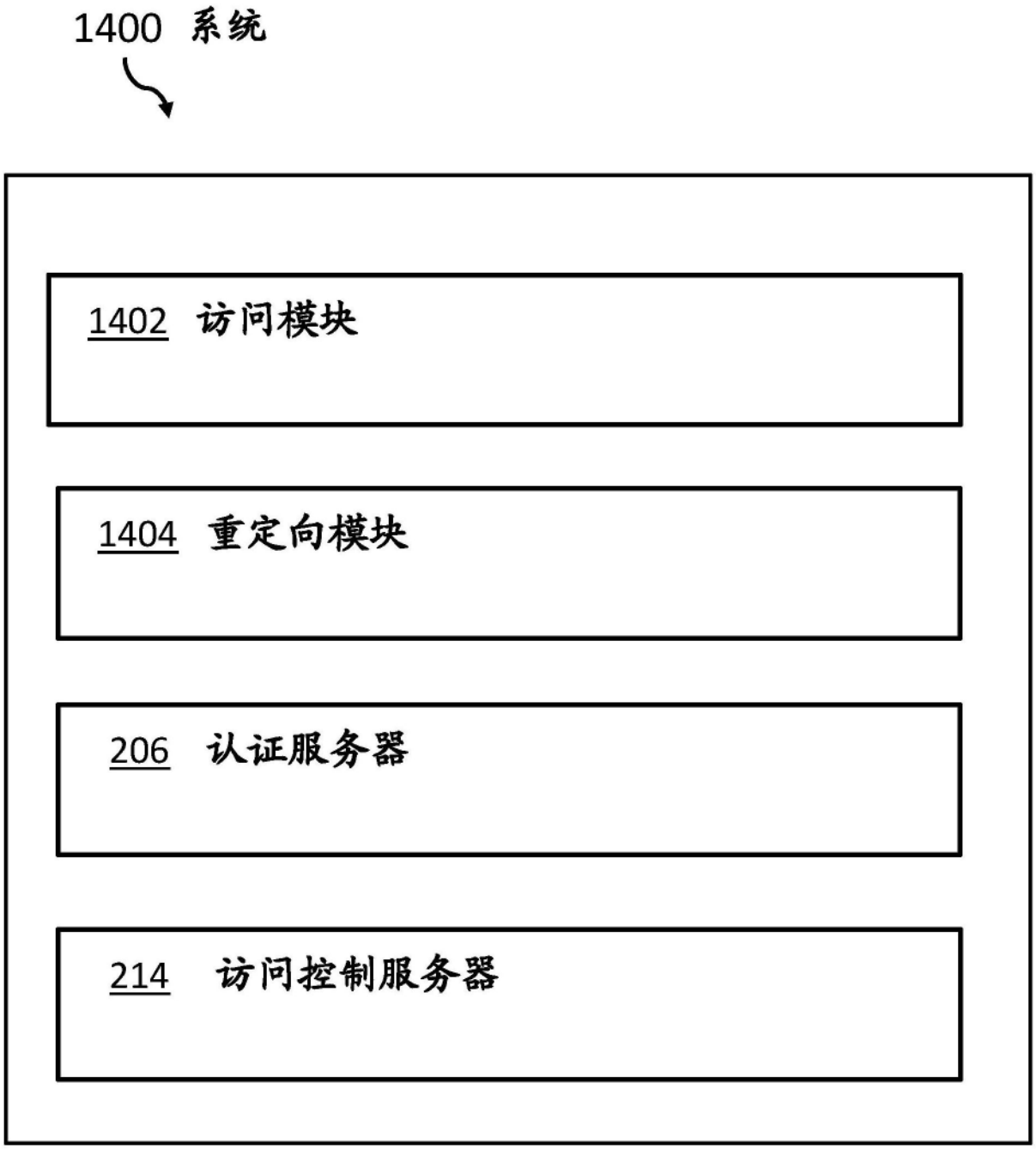



图14

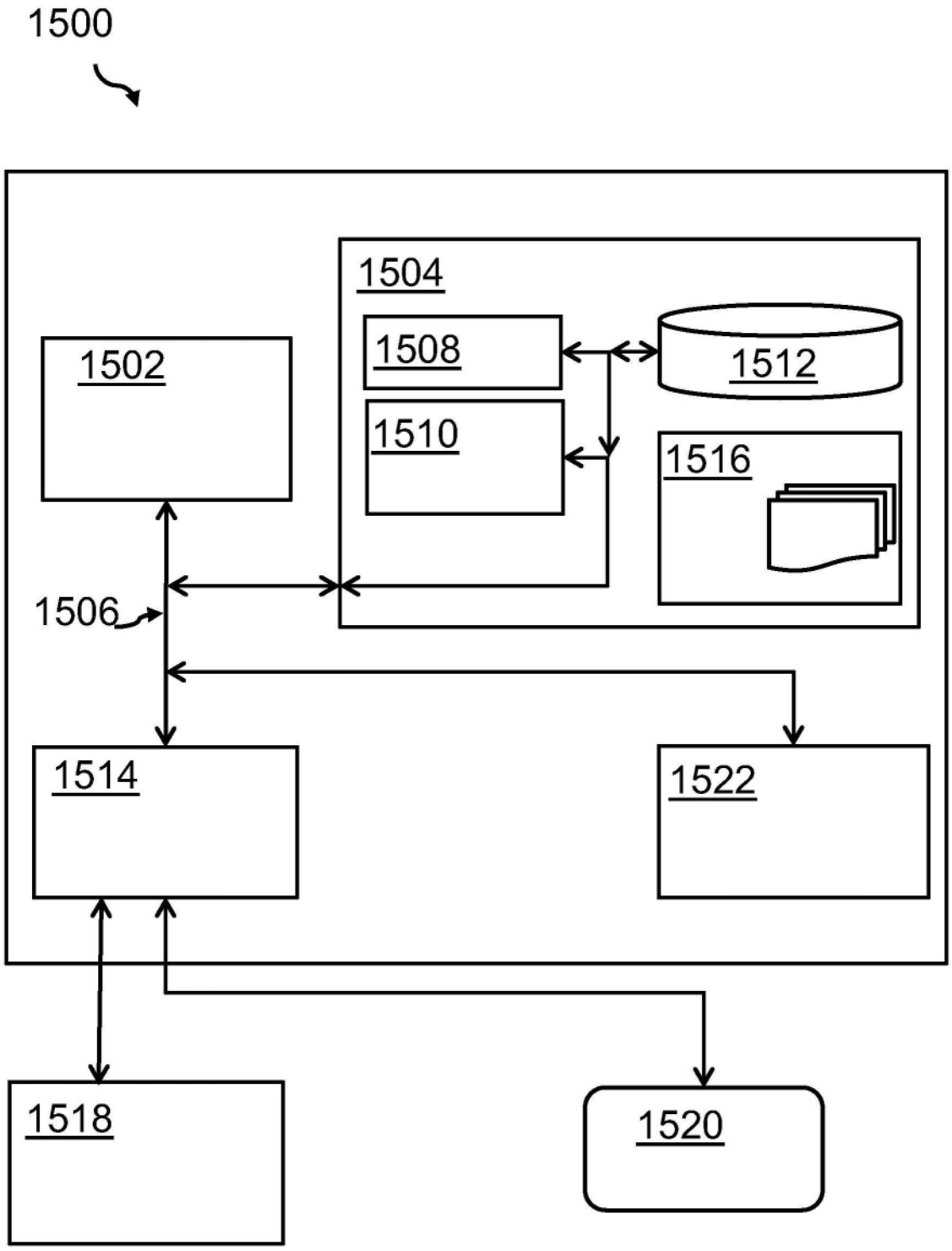



图15