(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 109639687 B
(45)授权公告日 2021.05.28

(21)申请号 201811551459.5
(22)申请日 2017.09.14
(65)同一申请的已公布的文献号
申请公布号 CN 109639687 A
(43)申请公布日 2019.04.16
(30)优先权数据
62/394,273 2016.09.14 US
62/394,345 2016.09.14 US
(62)分案原申请数据
201780034546.0 2017.09.14
(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 J·V·甘咖韦恩 B·约瑟夫 B·萨恩克萨拉 M·P·尤奇尔
(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 边海梅
(51)Int.Cl.
H04L 29/06(2006.01)
H04L 29/08(2006.01)
(56)对比文件
US 2014245389 A1,2014.08.28
US 9118657 B1,2015.08.25
CN 105659558 A,2016.06.08
CN 104903905 A,2015.09.09
CN 102315945 A,2012.01.11
CN 101166173 A,2008.04.23
CN 105659557 A,2016.06.08
CN 103930897 A,2014.07.16
审查员 王莉

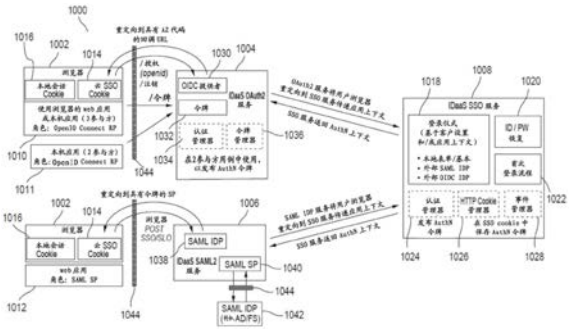
权利要求书3页 说明书48页 附图18页

(54)发明名称

用于提供基于云的身份和访问管理的系统、方法和介质

(57)摘要

本发明涉及用于提供基于云的身份和访问管理的系统、方法和介质。实现单点登录(“SSO”)的基于云的身份和访问管理系统接收对被配置为允许访问应用的身份管理服务的第一请求。实施例将第一请求发送到第一微服务,该第一微服务通过生成令牌来执行身份管理服务。第一微服务至少部分地通过向SSO微服务发送第二请求来生成令牌,该SSO微服务被配置为跨基于不同协议的不同微服务来提供SSO功能。然后,实施例从第一微服务接收令牌并将令牌提供给应用,其中令牌允许访问应用。



CN 109639687 B

1.一种其上存储有指令的非瞬态计算机可读介质,所述指令在由服务器的处理器执行时使得所述处理器提供基于云的身份和访问管理,所述提供包括:

接收对被配置为允许访问应用的身份管理服务的第一请求;

将所述第一请求发送到第一微服务,其中所述第一微服务通过生成令牌来执行所述身份管理服务,其中所述第一微服务至少部分地通过向单点登录SSO微服务发送第二请求来生成所述令牌,其中所述SSO微服务被配置为跨基于不同协议的不同微服务提供SSO功能;

其中所述SSO微服务实现SSO,并生成包括全局状态并用于与不同的微服务进行通信的cookie;

接收所述SSO的单点注销SLO;以及

使用所述cookie迭代地从所述应用注销,其中在从第一协议的应用的每次注销之后,执行到所述SSO微服务的重定向以触发从不同协议的应用的注销。

2.如权利要求1所述的计算机可读介质,其中所述cookie指示登录到所述SSO的应用。

3.如权利要求1所述的计算机可读介质,还包括:

从所述第一微服务接收所述令牌;以及

将所述令牌提供给应用,其中所述令牌允许访问所述应用。

4.如权利要求1所述的计算机可读介质,其中所述第一微服务被配置为基于所述协议提供认证功能。

5.如权利要求4所述的计算机可读介质,其中所述协议是OAuth或安全断言标记语言SAML,其中所述SSO微服务被配置为跨OAuth和SAML两者提供SSO功能。

6.如权利要求5所述的计算机可读介质,其中所述SSO微服务生成被配置为会话cookie的全局会话,并将所述全局会话提供给所述第一微服务。

7.如权利要求6所述的计算机可读介质,其中所述第一微服务将所述全局会话转换成所述令牌。

8.一种提供基于云的身份和访问管理的方法,所述方法由服务器的处理器执行,并且所述方法包括:

接收对被配置为允许访问应用的身份管理服务的第一请求;

将所述第一请求发送到第一微服务,其中所述第一微服务通过生成令牌来执行所述身份管理服务,其中所述第一微服务至少部分地通过向单点登录SSO微服务发送第二请求来生成所述令牌,其中所述SSO微服务被配置为跨基于不同协议的不同微服务提供SSO功能;

其中所述SSO微服务实现SSO,并生成包括全局状态并用于与不同的微服务进行通信的cookie;

接收所述SSO的单点注销SLO;以及

使用所述cookie迭代地从所述应用注销,其中在从第一协议的应用的每次注销之后,执行到所述SSO微服务的重定向以触发从不同协议的应用的注销。

9.如权利要求8所述的方法,其中所述cookie指示登录到所述SSO的应用。

10.如权利要求8所述的方法,还包括:

从所述第一微服务接收所述令牌;以及

将所述令牌提供给应用,其中所述令牌允许访问所述应用。

11.如权利要求8所述的方法,其中所述第一微服务被配置为基于所述协议提供认证功

能。

12.如权利要求11所述的方法,其中所述协议是OAuth或安全断言标记语言SAML,其中所述SSO微服务被配置为跨OAuth和SAML两者提供SSO功能。

13.如权利要求12所述的方法,其中所述SSO微服务生成被配置为会话cookie的全局会话,并将所述全局会话提供给所述第一微服务。

14.如权利要求13所述的方法,其中所述第一微服务将所述全局会话转换成所述令牌。

15.一种用于提供基于云的身份和访问管理的系统,包括:

多个租户;

多个微服务;以及

服务器的一个或多个处理器,所述一个或多个处理器:

接收对被配置为允许访问应用的身份管理服务的第一请求;

将所述第一请求发送到第一微服务,其中所述第一微服务通过生成令牌来执行所述身份管理服务,其中所述第一微服务至少部分地通过向单点登录SSO微服务发送第二请求来生成所述令牌,其中所述SSO微服务被配置为跨基于不同协议的不同微服务提供SSO功能;

其中所述SSO微服务实现SSO,并生成包括全局状态并用于与不同的微服务进行通信的cookie;

接收所述SSO的单点注销SLO;以及

使用所述cookie迭代地从所述应用注销,其中在从第一协议的应用的每次注销之后,执行到所述SSO微服务的重定向以触发从不同协议的应用的注销。

16.如权利要求15所述的系统,其中所述cookie指示登录到所述SSO的应用。

17.如权利要求15所述的系统,还包括:

从所述第一微服务接收所述令牌;以及

将所述令牌提供给应用,其中所述令牌允许访问所述应用。

18.如权利要求15所述的系统,其中所述第一微服务被配置为基于所述协议提供认证功能。

19.如权利要求18所述的系统,其中所述协议是OAuth或安全断言标记语言SAML,其中所述SSO微服务被配置为跨OAuth和SAML两者提供SSO功能。

20.如权利要求19所述的系统,其中所述SSO微服务生成被配置为会话cookie的全局会话,并将所述全局会话提供给所述第一微服务。

21.一种用于提供基于云的身份和访问管理的设备,包括:

处理器;以及

存储器,所述存储器耦合到所述处理器并且所述存储器包括存储在其上的指令,所述指令在由所述处理器执行时,使得所述处理器执行如权利要求8-14中任一项所述的方法。

22.一种用于提供基于云的身份和访问管理的装置,所述装置包括:

用于接收对被配置为允许访问应用的身份管理服务的第一请求的部件;

用于将所述第一请求发送到第一微服务的部件,其中所述第一微服务通过生成令牌来执行所述身份管理服务,其中所述第一微服务至少部分地通过向单点登录SSO微服务发送第二请求来生成所述令牌,其中所述SSO微服务被配置为跨基于不同协议的不同微服务提供SSO功能;

其中所述SSO微服务实现SSO，并生成包括全局状态并用于与不同的微服务进行通信的cookie；

用于接收所述SSO的单点注销SLO的部件；以及

用于使用所述cookie迭代地从所述应用注销的部件，其中在从第一协议的应用的每次注销之后，执行到所述SSO微服务的重定向以触发从不同协议的应用的注销。

23.如权利要求22所述的装置，其中所述cookie指示登录到所述SSO的应用。

24.如权利要求22所述的装置，还包括：

用于从所述第一微服务接收所述令牌的部件；以及

用于将所述令牌提供给应用的部件，其中所述令牌允许访问所述应用。

25.如权利要求22所述的装置，其中所述第一微服务被配置为基于所述协议提供认证功能。

26.如权利要求25所述的装置，其中所述协议是OAuth或安全断言标记语言SAML，其中所述SSO微服务被配置为跨OAuth和SAML两者提供SSO功能。

27.如权利要求26所述的装置，其中所述SSO微服务生成被配置为会话cookie的全局会话，并将所述全局会话提供给所述第一微服务。

28.如权利要求27所述的装置，其中所述第一微服务将所述全局会话转换成所述令牌。

用于提供基于云的身份和访问管理的系统、方法和介质

[0001] 本申请是国际申请日为2017年9月14日、国家申请号为201780034546.0、发明名称为“用于多租户身份和数据安全管理云服务的单点登录和单点注销功能”的进入中国国家阶段的PCT申请的分案申请。

[0002] 相关申请的交叉引用

[0003] 本申请要求于2016年9月14日提交的序列号为62/394,273的美国临时专利申请和于2016年9月14日提交的序列号为62/394,345的美国临时专利申请的优先权。这些申请中的每一个的公开内容通过引用并入本文。

技术领域

[0004] 一个实施例一般涉及身份管理,尤其涉及云系统中的身份管理。

背景技术

[0005] 一般而言,基于云的应用(例如,企业公共云应用、第三方云应用等)的使用正在飞速发展,其中访问来自各种设备(例如,桌面和移动设备)以及各种用户(例如,员工、合作伙伴、客户等)。基于云的应用的丰富多样性和可访问性已经导致身份管理和访问安全性成为中心问题。云环境中的典型安全问题是未经授权的访问、帐户劫持、恶意的内部人员等。因而,需要安全地访问基于云的应用或位于任何地方的应用,而不管应用被何种设备类型或何种用户类型访问。

发明内容

[0006] 一个实施例是实现单点登录(“SSO”)的基于云的身份和访问管理系统。实施例接收对被配置为允许访问应用的身份管理服务的第一请求。实施例将第一请求发送到第一微服务,该第一微服务通过生成令牌来执行身份管理服务。第一微服务至少部分地通过向SSO微服务发送第二请求来生成令牌,该SSO微服务被配置为跨基于不同协议的不同微服务提供SSO功能。然后,实施例从第一微服务接收令牌并将令牌提供给应用,其中令牌允许访问应用。

[0007] 一个实施例是实现单点登录(“SSO”)的基于云的身份和访问管理系统。实施例接收对被配置为允许访问应用的身份管理服务的第一请求。实施例将第一请求发送到第一微服务,其中该第一微服务通过生成令牌来执行身份管理服务。第一微服务至少部分地通过向SSO微服务发送第二请求来生成令牌,其中该SSO微服务被配置为跨基于不同协议的不同微服务提供SSO功能。SSO微服务实现SSO并生成包括全局状态并用于与不同的微服务进行通信的cookie。实施例接收SSO的单点注销(“SLO”),并使用cookie迭代地从应用注销,其中在从第一协议的应用的每次注销之后,执行到SSO微服务的重定向以触发从不同协议的应用的注销。

附图说明

[0008] 图1-图5是提供基于云的身份管理的示例实施例的框图。

[0009] 图6是提供实施例的系统视图的框图。

[0010] 图6A是提供实施例的功能视图的框图。

[0011] 图7是实现云门的实施例的框图。

[0012] 图7A是图示了实现云门的实施例中的主机标识符功能的框图。

[0013] 图8图示了在一个实施例中实现多个租户的示例系统。

[0014] 图9是实施例的网络视图的框图。

[0015] 图10是一个实施例中的单点登录(“SSO”)功能的系统架构视图的框图。

[0016] 图10A图示了一个实施例中的SSO运行时的示例框图。

[0017] 图10B图示了一个实施例中的SSO功能的示例状态和流程图。

[0018] 图11是一个实施例中的SSO功能的消息序列流程。

[0019] 图12图示了一个实施例中的分布式数据网格的实例。

[0020] 图13是根据实施例的身份和访问管理功能的流程图。

[0021] 图14是图示了根据一个实施例的部件的各种功能和cookie传播功能的注销流程。

具体实施方式

[0022] 实施例在身份云服务(“IDCS”)中提供单点登录(“SSO”)功能,该身份云服务提供多租户、云规模、身份和访问管理(“IAM”)平台。在一个实施例中,SSO功能是通过提供全局会话然后基于全局会话生成协议特定的令牌来实现的。实施例还通过使用cookie迭代地从多个应用注销并使用重定向来提供单点注销(“SLO”)功能,从而安全信息不会存储在cookie上。

[0023] 实施例提供实现基于微服务的架构的身份云服务,并提供多租户身份和数据安全管理以及对基于云的应用的安全访问。实施例支持对于混合云部署的安全访问(即,包括公共云和私有云的组合的云部署)。实施例保护云中和内部部署(on-premise)的应用和数据。实施例支持经由web、移动和应用编程接口(“API”)的多信道访问。实施例管理对于不同用户(诸如客户、合作伙伴和员工)的访问。实施例管理、控制和审计跨云以及内部部署的访问。实施例与新的和现有的应用和身份集成。实施例是水平可伸缩的。

[0024] 一个实施例是在无状态中间层环境中实现多个微服务以提供基于云的多租户身份和访问管理服务的系统。在一个实施例中,每个被请求的身份管理服务被分解成实时任务和近实时任务。实时任务由中间层中的微服务处理,而近实时任务被卸载到消息队列。实施例实现由路由层和中间层消耗以强制实施用于访问微服务的安全模型的访问令牌。因而,实施例提供了基于多租户、微服务架构的云规模的IAM平台。

[0025] 一个实施例提供身份云服务,该服务使得组织能够为其新的商业计划快速开发高速、可靠和安全的服务。在一个实施例中,身份云服务提供许多核心服务,每个核心服务解决许多企业面临的独特挑战。在一个实施例中,身份云服务支持管理员例如进行用户的初始登入/导入、导入具有用户成员的组、创建/更新/禁用/启用/删除用户、将用户指派到组/从组中解除指派、创建/更新/删除组、重新设置密码、管理策略、发送激活等。身份云服务还支持最终用户例如修改简档、设置主要/恢复电子邮件、核实电子邮件、解锁其账户、改变密

码、在忘记密码的情况下恢复密码等。

[0026] 访问的统一安全性

[0027] 一个实施例保护云环境中以及内部部署环境中的应用和数据。该实施例保护任何人从任何设备对任何应用的访问。由于两个环境之间的安全性不一致会导致较高的风险，因此该实施例提供跨两种环境的保护。例如，这种不一致会使销售人员即使在已经叛逃到竞争对手之后仍能继续访问其客户关系管理(“CRM”)账户。因而，实施例将在内部部署环境中提供的安全控制扩展到云环境中。例如，如果人员离开公司，则实施例确保他们的账户在内部部署和云中都被禁用。

[0028] 一般而言，用户可以通过许多不同的渠道(诸如web浏览器、台式机、移动电话、平板电脑、智能手表、其它可穿戴设备等)来访问应用和/或数据。因而，一个实施例提供跨所有这些渠道的安全访问。例如，用户可以使用他们的移动电话完成他们在台式机上开始的事务。

[0029] 一个实施例还管理对于各种用户(诸如客户、合作伙伴、员工等)的访问。一般而言，应用和/或数据不仅可以由员工访问，而且可以由客户或第三方访问。虽然许多已知的系统在员工登入(onboarding)时采取安全措施，但是在向客户、第三方、合作伙伴等给予访问时一般不采取相同级别的安全措施，从而导致由未妥善管理的各方造成的安全漏洞的可能。但是，实施例确保为每种类型的用户的访问而不仅仅是员工的访问提供足够的安全措施。

[0030] 身份云服务

[0031] 实施例提供了作为多租户、云规模IAM平台的IDCS。IDCS提供认证、授权、审计和联合(federation)。IDCS管理对公共云上以及内部部署系统上运行的自定义应用和服务的访问。在替代或附加的实施例中，IDCS还可以管理对公共云服务的访问。例如，IDCS可用于跨这样的各种服务/应用/系统来提供SSO功能。

[0032] 实施例基于用于设计、构建和递送云规模的软件服务的多租户、微服务架构。多租户是指让服务的一个物理实现安全地支持多个客户购买该服务。服务是可以被不同客户端出于不同目的重用的软件功能或软件功能集合(诸如检索指定的信息或执行一组操作)，以及控制该软件功能或该软件功能集合的使用的策略(例如，基于请求服务的客户端的身份)。在一个实施例中，服务是使得能够访问一个或多个能力的机制，其中访问是使用规定的接口而提供的并且与由服务描述指定的约束和策略一致地被实施(exercised)。

[0033] 在一个实施例中，微服务是独立可部署的服务。在一个实施例中，术语微服务设想了软件架构设计模式，其中复杂应用由使用语言不可知的API彼此通信的小型独立进程组成。在一个实施例中，微服务是小的、高度解耦的服务，并且每个微服务可以专注于做一个小任务。在一个实施例中，微服务架构样式是将单个应用开发为一套小服务的做法，每个小服务在其自己的进程中运行并与轻量级机制(例如，超文本传输协议(“HTTP”)资源API)通信。在一个实施例中，相对于执行全部或许多相同功能的单件式服务，微服务更容易更换。而且，每个微服务可以被更新而不会对其它微服务产生不利影响。相比之下，对单件式服务的一部分的更新会不期望地或无意地对单件服务的其它部分产生负面影响。在一个实施例中，微服务可以围绕其能力被有益地组织。在一个实施例中，微服务集合中的每一个微服务的启动时间远小于笼统执行那些微服务的所有服务的单个应用的启动时间。在一些实施例

中,这种微服务中的每一个微服务的启动时间是大约一秒或更短,而这种单个应用的启动时间可以是大约一分钟、几分钟或更长。

[0034] 在一个实施例中,微服务架构是指用于面向服务的架构(“SOA”)的专业化(即,系统内任务的分离)和实现做法,以构建灵活的、独立可部署的软件系统。微服务架构中的服务是经网络彼此通信以便履行目标的进程。在一个实施例中,这些服务使用技术不可知的协议。在一个实施例中,服务具有小粒度并使用轻量级协议。在一个实施例中,服务是独立可部署的。通过将系统的功能分配到不同的小型服务中,增强了系统的内聚性并降低了系统的耦合度。这使得更容易随时改变系统以及向系统添加功能和质量。它还允许个体服务的架构通过不断的重构而出现,从而减少了对大型前期设计的需求,并且允许较早地并持续地发布软件。

[0035] 在一个实施例中,在微服务架构中,应用作为服务的集合被开发,并且每个服务运行相应的进程并使用轻量级协议进行通信(例如,用于每个微服务的唯一API)。在微服务架构中,取决于要提供的服务,将软件分解成各个服务/能力可以以不同的粒度级别来执行。服务是运行时部件/进程。每个微服务是可以与其它模块/微服务交谈的自包含模块。每个微服务具有可以被其它微服务联系的未命名的通用端口。在一个实施例中,微服务的未命名的通用端口是微服务按照惯例暴露并允许同一服务内的任何其它模块/微服务与其交谈的标准通信信道(例如,作为常规的超文本传输协议(“HTTP”)端口)。微服务或任何其它自包含的功能模块可以统称为“服务”。

[0036] 实施例提供多租户身份管理服务。实施例基于开放标准,以确保易于与各种应用集成,从而通过基于标准的服务来递送IAM能力。

[0037] 实施例管理用户身份的生命周期,这需要确定和强制实施身份可以访问什么、谁可以被给予这种访问、谁可以管理这种访问等。对于不一定在云中的应用,实施例在云中运行身份管理工作负载并且支持安全功能。由实施例提供的身份管理服务可以从云购买。例如,企业可以从云购买这种服务以管理他们的员工访问他们的应用。

[0038] 实施例提供系统安全性、大规模可伸缩性、最终用户可用性和应用互操作性。实施例解决了云的增长和客户对身份服务的使用。基于微服务的基础解决了水平可伸缩性需求,同时服务的仔细编排解决了功能需求。实现这两个目标需要(尽可能)分解业务逻辑以实现具有最终一致性的无状态性,而大多数不受实时处理影响的操作逻辑通过卸载到高度可伸缩的异步事件管理系统而转移为近实时,具有有保证的递送和处理。实施例从web层到数据层是完全多租户的,以便实现成本效率和系统管理的容易性。

[0039] 实施例基于行业标准(例如,OpenID Connect、OAuth2、安全断言标记语言2(“SAML2”)、用于跨域身份管理的系统(“SCIM”)、具象状态传输(“REST”)等)以便于与各种应用集成。一个实施例提供了云规模API平台并实现了用于弹性可伸缩性的水平可伸缩微服务。该实施例充分利用云原理并提供具有每租户数据分离的多租户架构。该实施例还提供了经由租户自助服务的每租户定制。该实施例经由API可用于与其它身份服务的按需集成,并提供持续的特征发布。

[0040] 一个实施例提供互操作性并充分利用对云中和内部部署的身份管理(“IDM”)功能的投资。该实施例提供从内部部署轻量级目录访问协议(“LDAP”)数据到云数据的自动化身份同步,反之亦然。该实施例在云和企业之间提供SCIM身份总线,并且允许混合云部署的不

同选项(例如,身份联合和/或同步、SSO代理、用户供应连接器等)。

[0041] 因而,一个实施例是在无状态中间层中实现多个微服务以提供基于云的多租户身份和访问管理服务的系统。在一个实施例中,每个被请求的身份管理服务被分解成实时任务和近实时任务。实时任务由中间层中的微服务处理,而近实时任务被卸载到消息队列。实施例实现由路由层消耗以强制实施用于访问微服务的安全模型的令牌。因而,实施例提供了基于多租户、微服务架构的云规模的IAM平台。

[0042] 一般而言,已知的系统提供对由不同环境提供的应用(例如,企业云应用、合作伙伴云应用、第三方云应用和客户应用)的竖井式(siloed)访问。这种竖井式访问可能需要多个密码、不同的密码策略、不同的帐户供应和解除供应方案、全异的审计等。但是,一个实施例实现了IDCS,以在这种应用上提供统一的IAM功能。图1是具有IDCS118的示例实施例的框图100,其提供了用于对用户和应用进行登入的统一身份平台126。该实施例跨各种应用(诸如企业云应用102、合作伙伴云应用104、第三方云应用110和客户应用112)提供无缝的用户体验。应用102、104、110、112可以通过不同的渠道(例如,由移动电话用户108经由移动电话106、由台式计算机用户116经由浏览器114等)来访问。web浏览器(通常称为浏览器)是用于检索、呈现和遍历万维网上的信息资源的软件应用。web浏览器的示例是Mozilla **Firefox®**、Google **Chrome®**、Microsoft Internet **Explorer®**和Apple **Safari®**。

[0043] IDCS 118提供用户的应用、(经由身份平台126)跨设备和应用的统一安全凭证以及(经由管理控制台122)统一的管理方式的统一视图124。IDCS服务可以通过调用IDCS API 142来获得。这种服务可以包括例如登录/SSO服务128(例如,OpenID Connect)、联盟服务130(例如,SAML)、令牌服务132(例如,OAuth)、目录服务134(例如,SCIM)、供应服务136(例如,SCIM或任何经多协议的传输(“AToM”))、事件服务138(例如,REST)以及基于角色的访问控制(“RBAC”)服务140(例如,SCIM)。IDCS 118还可以提供与所提供的服务相关的报告和仪表盘120。

[0044] 集成工具

[0045] 一般而言,大型公司通常在适当的位置具有IAM系统以保护对其内部部署应用的访问。业务实践通常都是围绕内部IAM系统(诸如来自Oracle公司的“Oracle IAM套件”)成熟和标准化的。甚至中小企业通常也会通过简单目录解决方案(诸如Microsoft Active Directory(“AD”))来围绕管理用户访问设计其业务进程。为了启用内部部署集成,实施例提供了允许客户将其应用与IDCS集成的工具。

[0046] 图2是在云环境208中具有IDCS 202的示例实施例的框图200,其提供与内部部署206的AD 204的集成。该实施例提供跨包括内部部署及第三方应用(例如,内部部署应用218和云208中诸如云服务210、云应用212、合作伙伴应用214和客户应用216的各种应用/服务)在内的所有应用的无缝用户体验。云应用212可以包括例如人力资本管理(“HCM”)、CRM、人才获取(例如,来自Oracle公司的Oracle Taleo云服务)、配置价格和报价(“CPQ”)等。云服务210可以包括例如平台即服务(“PaaS”)、Java、数据库、商业智能(“BI”)、文档等。

[0047] 应用210、212、214、216、218可以通过不同的渠道(例如,由移动电话用户220经由移动电话222、由台式计算机用户224经由浏览器226等)来访问。该实施例提供经由云208和企业206之间的SCIM身份总线234从内部部署AD数据到云数据的自动化身份同步。该实施例还提供SAML总线228,用于(例如,使用密码232)将来自云208的认证联合到内部部署AD

204。

[0048] 一般而言,身份总线是用于身份相关的服务的服务总线。服务总线为从一个系统到另一个系统传送消息提供平台。它是用于在可信系统之间交换信息的受控机制,例如,在面向服务的架构(“SOA”)中。身份总线是根据基于标准HTTP的机制(诸如web服务、web服务器代理等)构建的逻辑总线。身份总线中的通信可以根据相应的协议(例如,SCIM、SAML、OpenID Connect等)。例如,SAML总线是两个系统*之间基于HTTP的连接,用于传送用于SAML服务的消息。类似地,SCIM总线用于根据SCIM协议来传送SCIM消息。

[0049] 图2的实施例实现身份(“ID”)桥接器230,其是可以与客户的AD 204一起下载并安装在内部部署206的小二进制文件(例如,1MB尺寸)。ID桥接器230监听来自客户选择的组织单位(“OU”)的用户和组(例如,用户组),并将那些用户同步到云208。在一个实施例中,用户的密码232不同步到云208。客户可以通过将IDCS用户的组映射到在IDCS 208中管理的云应用来管理用户的应用访问。每当用户的组成员资格在内部部署206被改变时,其对应的云应用访问自动改变。

[0050] 例如,从工程转移到销售的员工可以几乎即时访问销售云并失去对开发者云的访问。当这种改变反映在内部部署AD 204中时,云应用访问改变是近实时完成的。类似地,对于离开公司的用户,对在IDCS 208中管理的云应用的访问被撤销。为了完全自动化,客户可以通过例如AD联合服务(“AD/FS”或实现SAML联合的某种其它机制)在内部部署AD 204和IDCS 208之间设置SSO,使得最终用户可以用单个公司密码332取得对云应用210、212、214、216以及内部部署应用218的访问。

[0051] 图3是包括与图2中相同的部件202、206、208、210、212、214、216、218、220、222、224、226、228、234的示例实施例的框图300。但是,在图3的实施例中,IDCS 202提供与内部部署IDM 304(诸如Oracle IDM)的集成。Oracle IDM 304是Oracle公司提供IAM功能的软件套件。该实施例提供跨包括内部部署和第三方应用在内的所有应用的无缝用户体验。该实施例经由云202和企业206之间的SCIM身份总线234从内部部署IDM 304到IDCS 208供应用户身份。该实施例还提供SAML总线228(或OpenID Connect总线),用于将来自云208的认证联合到内部部署206。

[0052] 在图3的实施例中,来自Oracle公司的Oracle身份管理器(“OIM”)连接器302和来自Oracle公司的Oracle访问管理器(“OAM”)联合模块306被实现为Oracle IDM 304的扩展模块。连接器是具有关于如何与系统交谈的物理意识的模块。OIM是被配置为管理用户身份(例如,基于用户应当和不应当访问的内容来管理不同系统中的用户账户)的应用。OAM是提供访问管理功能的安全应用,访问管理功能诸如Web SSO;身份上下文;认证和授权;策略管理;测试;记录;审计等。OAM具有对SAML的内置支持。如果用户在IDCS 202中有账户,则OIM连接器302和OAM联合306可以与Oracle IDM 304一起使用,以创建/删除该账户并且管理来自该账户的访问。

[0053] 图4是包括如图2和3中所示的相同部件202、206、208、210、212、214、216、218、220、222、224、226、234示例实施例的框图400。但是,在图3的实施例中,IDCS 202提供将云身份扩展到内部部署应用218的功能。该实施例提供跨包括内部部署和第三方应用在内的所有应用的身份的无缝视图。在图4的实施例中,SCIM身份总线234用于使IDCS 202中的数据与称为“云高速缓存”402的内部部署LDAP数据同步。下面更详细地公开云高速缓存402。

[0054] 一般而言，被配置为基于LDAP进行通信的应用需要LDAP连接。由于LDAP需要在本地网络上，因此这种应用不能通过URL建立LDAP连接(不像例如连接到Google的“www.google.com”)。在图4的实施例中，基于LDAP的应用218作出到云高速缓存402的连接，并且云高速缓存402建立到IDCS 202的连接并随后在其被请求时从IDCS 202拉出数据。IDCS 202与云高速缓存402之间的通信可以根据SCIM协议来实现。例如，云高速缓存402可以使用SCIM总线234来向IDCS 202发送SCIM请求并作为回报接收对应数据。

[0055] 一般而言，完全实现应用包括构建消费者门户、在外部用户群体上运行营销活动、支持web和移动渠道以及处理用户认证、会话、用户简档、用户组、应用角色、密码策略、自助服务/注册、社会整合、身份联合等。一般而言，应用开发人员不是身份/安全专家。因此，按需身份管理服务是期望的。

[0056] 图5是包括与图2-图4中相同的部件202、220、222、224、226、234、402的示例实施例的框图500。但是，在图5的实施例中，IDCS 202按需提供安全身份管理。该实施例按需提供与IDCS 202的身份服务的集成(例如，基于诸如OpenID Connect、OAuth2、SAML2或SCIM的标准)。应用505(其可以是内部部署的、在公共云中或在私有云中)可以调用IDCS 202中的身份服务API 504。由IDCS 202提供的服务可以包括例如自助服务注册506、密码管理508、用户简档管理510、用户认证512、令牌管理514、社会整合516等。

[0057] 在这个实施例中，SCIM身份总线234用于使IDCS 202中的数据与内部部署的LDAP云高速缓存402中的数据同步。另外，在web服务器/代理(例如，NGINX、Apache等)上运行的“云门(Cloud Gate)”502可以被应用505使用，以从IDCS 202获得用户web SSO和REST API安全性。云门502是通过确保客户端应用提供有效访问令牌和/或用户成功认证来保护到多租户IDCS微服务的访问的部件，以便建立SSO会话。云门502在下面进一步公开。云门502(类似于webgate/webagent的强制实施点)使得在被支持的web服务器后面运行的应用能够参与SSO。

[0058] 一个实施例提供SSO和云SSO功能。在许多组织中，用于内部部署的IAM和IDCS的一般入口点是SSO。云SSO使用户能够用单一用户登录来访问多个云资源。组织常常想要联合他们的内部部署的身份。因而，实施例利用开放标准来允许与现有SSO集成，以保持和扩展投资(例如，直到进行了到身份云服务方法的完全最终过渡为止)。

[0059] 一个实施例可以提供以下功能：

[0060] • 维护身份存储库，以跟踪已被授权的用户帐户、所有权、访问和许可，

[0061] • 与工作流程集成，以促进应用访问所需的各种审批(例如，管理、IT、人力资源、法律和合规性)，

[0062] • 为选择性设备(例如，移动和个人计算机(“PC”))提供SaaS用户帐户，以访问包含许多私有和公共云资源的用户门户，

[0063] • 促进周期性管理鉴证(attestation)审查，以符合法规和当前的工作职责。

[0064] 除了这些功能之外，实施例还可以提供：

[0065] • 云帐户供应，以管理云应用中的帐户生命周期，

[0066] • 更健壮的多因素认证(“MFA”)集成，

[0067] • 广泛的移动安全能力，以及

[0068] • 动态认证选项。

[0069] 一个实施例提供自适应的认证和MFA。一般而言，密码和质询问题被认为是不够的，并且易于受诸如网络钓鱼等常见攻击。如今的大多数商业实体都在寻求某种形式的MFA以降低风险。但是，为了被成功部署，解决方案需要由最终用户轻松供应、维护和理解，因为最终用户通常会抵制干扰其数字体验的任何事情。公司正在寻找安全地结合自带设备(“BYOD”)、社交身份、远程用户、客户和承包商的方式，同时使MFA成为无缝用户访问体验中几乎透明的部件。在MFA部署中，诸如OAuth和OpenID Connect等行业标准对于确保现有多因素解决方案的集成和新型自适应认证技术的结合至关重要。因而，实施例将动态(或自适应)认证定义为在用户会话已经发起之后证明身份的可用信息(即，IP地址、位置、一天中的时间和生物测定)的评估。通过适当的标准(例如，开放认证(“OATH”)和快速身份在线(“FIDO”))集成和可扩展身份管理框架，实施例提供了可以在IT组织中被容易地采用、升级和集成的MFA解决方案，作为端到端安全IAM部署的一部分。在考虑MFA和自适应策略时，组织必须在内部部署和云资源上实现一致的策略，这在混合IDCS和内部部署IAM环境中需要系统之间的集成。

[0070] 一个实施例提供用户供应和证实(certification)。一般而言，IAM解决方案的基本功能是启用和支持整个用户供应生命周期。这包括为用户提供适合于他们在组织内的身份和角色的应用访问、证实他们具有正确的持续访问许可(例如，当他们的角色或在其角色内使用的任务或应用随时间变化时)，并且在他们离开组织时迅速地解除供应。这是重要的，不仅为了满足各种合规要求，而且还因为不适当的内部人员访问是安全漏洞和攻击的主要来源。身份云解决方案中的自动化用户供应能力不仅可以作为自己的重要组成部分，而且也可以作为混合IAM解决方案的一部分，借此，对于当公司缩小规模、扩大规模、合并或指望将现有系统与IaaS/PaaS/SaaS环境集成在一起时的过渡，IDCS供应可以提供比内部部署解决方案更大的灵活性。IDCS方法可以节省一次性升级的时间和精力，并确保必要的部门、分区和系统之间的适当集成。伸缩这项技术的需求常常在企业中悄然发现，并且跨企业立即递送可伸缩的IDCS能力的能力可以提供灵活性、成本和控制方面的好处。

[0071] 一般而言，随着她/他的工作改变，员工随着年限被授予附加的特权(即，“特权蠕变(creep)”)。受到轻度监管的公司一般缺乏要求管理人员定期审计员工的特权(例如，访问网络、服务器、应用和数据)以中止或减慢导致超级特权帐户的特权蠕变的“鉴证”处理。因而，一个实施例可以提供定期进行(至少一年一次)的鉴证处理。另外，随着并购，对这些工具和服务的需求将以指数形式增长，因为用户在SaaS系统上、内部部署、跨不同部门和/或正在被解除供应或重新分配。迁移到云会进一步使这种情况混乱，并且事情会迅速升级到超出现有的、常常是人工管理的证实方法。因而，一个实施例使这些功能自动化，并将复杂的分析应用于用户简档、访问历史、供应/解除供应和精细粒度赋权。

[0072] 一个实施例提供身份分析。一般而言，能够将身份分析与IAM引擎集成以进行全面证实和鉴证对于确保组织的风险简档至关重要。正确部署的身份分析可以要求全面的内部策略强制实施。在积极主动的治理、风险和合规性(“GRC”)企业环境中，非常需要跨云和内部部署提供统一的单个管理视图的身份分析，并且可以帮助提供闭环处理以降低风险和满足合规性法规。因而，一个实施例提供客户端能够容易定制的身份分析，以适应管理者、主管人员和审计人员所需的用于报告和分析的具体行业需求和政府法规。

[0073] 一个实施例提供自助服务和访问请求功能，以改进最终用户的体验和效率并降低

服务台呼叫的成本。一般而言，虽然许多公司为员工部署了内部部署的自助访问请求，但是许多公司并没有在正式的公司范围之外充分扩展这些系统。除了员工使用，积极的数字客户体验增加商业信誉并最终有助于增加收入，并且公司不仅可以节省客户服务台呼叫和成本，而且还可以改进客户满意度。因而，一个实施例提供基于开放标准并且在必要时无缝地与现有访问控制软件和MFA机制集成的身份云服务环境。SaaS递送模式节省了以前投入于系统升级和维护的时间和精力，从而使专业IT人员能够专注于更核心的业务应用。

[0074] 一个实施例提供特权账户管理(“PAM”)。一般而言，无论是使用SaaS、PaaS、IaaS还是内部部署应用，每个组织都容易受到具有超级用户访问凭证的内部人员(诸如系统管理员、主管人员、人事主管、承包商、系统集成商等)对未经授权的特权账户的滥用的攻击。而且，外部威胁通常首先突破低级用户帐户，以最终到达并利用企业系统内的特权用户访问控制。因而，一个实施例提供PAM，以防止这种未经授权的内部人员账户使用。PAM解决方案的主要组成部分是可以以各种方式递送的密码保险库(valult)，例如作为安装在企业服务器上的软件、作为同样企业服务器上的虚拟设备、作为打包的硬件/软件设备，或者作为云服务的一部分。PAM功能类似于用于存储保存在信封中并定期改变的密码的物理保险箱，具有用于签入和签出的清单。一个实施例允许密码校验(password checkout)以及设置时间限制、强制周期性改变、自动跟踪校验以及报告所有活动。一个实施例提供直接连接到所请求的资源而无需用户知道密码的方式。这个能力还为会话管理和附加功能铺平了道路。

[0075] 一般而言，大多数云服务利用API和管理界面，其为渗入者(infiltrator)提供了绕过安全性的机会。因而，一个实施例在PAM实践中考虑这些漏洞，因为向云的移动给PAM带来了新的挑战。现在许多中小型企业都在管理他们自己的SaaS系统(例如，Office 365)，而更大型的企业越来越多地拥有各自的业务单元，这些业务单元分别启动(spinning up)自己的SaaS和IaaS服务。这些客户在身份云服务解决方案或其IaaS/PaaS提供者身上发现自身具有PAM能力，但在在处理这个责任方面经验不足。而且，在一些情况下，许多不同的地理位置分散的业务单元正试图将针对相同SaaS应用的管理责任隔离。因而，一个实施例允许客户在这些情况下将现有的PAM链接到身份云服务的整体身份框架中，并且朝着更大的安全性和与确保伸缩到如业务需求规定的云负载要求的合规性移动。

[0076] API平台

[0077] 实施例提供了将能力的集合作为服务来暴露的API平台。API被聚合到微服务中，并且每个微服务暴露API中的一个或多个。即，每个微服务可以暴露不同类型的API。在一个实施例中，每个微服务仅通过其API来进行通信。在一个实施例中，每个API可以是微服务。在一个实施例中，基于要由服务提供的目标能力(例如，OAuth、SAML、管理员等)将多个API聚合到该服务中。因此，类似的API不作为分开的运行时进程来被暴露。API使得可用于使服务消费者使用由IDCS提供的服务。

[0078] 一般而言，在IDCS的web环境中，URL包括三部分：主机、微服务和资源(例如，主机/微服务/资源)。在一个实施例中，微服务的特征在于具有特定的URL前缀，例如“主机/oauth/v1”，其中实际的微服务是“oauth/v1”，并且在“oauth/v1”下有多个API，例如请求令牌的API：“主机/oauth/v1/令牌”、认证用户的API：“主机/oauth/v1/授权”等。即，URL实现微服务，并且URL的资源部分实现API。因而，在同一微服务下聚合了多个API。在一个实施例中，URL的主机部分标识租户(例如https://tenant3.identity.oraclecloud.com:/oauth/

v1/token")。

[0079] 配置利用必要的端点与外部服务集成的应用以及保持该配置最新通常是一个挑战。为了应对这一挑战,实施例在众所周知的位置暴露公开的发现API,从那里应用可以发现关于它们为了消费IDCS API而需要的IDCS的信息。在一个实施例中,支持两个发现文档:IDCS配置(其包括IDCS、SAML、SCIM、OAuth和OpenID Connect配置,例如在<IDCS-URL>/.well-know/idcs-configuration)和行业标准OpenID Connect配置(例如,<IDCS-URL>/.well-known/openid-configuration)。应用可以通过配置有单个IDCS URL来检索发现文档。

[0080] 图6是在一个实施例中提供IDCS的系统视图600的框图。在图6中,各种应用/服务602中的任何一个可以对IDCS API进行HTTP调用,以使用IDCS服务。这种应用/服务602的示例是web应用、本机应用(例如,被构建为在具体操作系统上运行的应用,诸如Windows应用、iOS应用、Android应用等)、web服务、客户应用、合作伙伴应用或者由公共云提供的任何服务(诸如软件即服务("SaaS")、PaaS和基础设施即服务("IaaS"))。

[0081] 在一个实施例中,需要IDCS服务的应用/服务602的HTTP请求经过Oracle公共云BIG-IP应用604和IDCS BIG-IP应用606(或类似技术,诸如负载均衡器,或者实现适当的安全规则以保护流量的被称为云负载均衡器即服务("LBaaS")的部件)。但是,请求可以以任何方式被接收。在IDCS BIG-IP应用606(或者如适用的,诸如负载均衡器或云LBaaS的类似技术),云供应引擎608执行租户和服务编排。在一个实施例中,云供应引擎608管理与正在登入到云中的新租户或由客户购买的新服务实例相关联的内部安全工件(artifact)。

[0082] 然后,HTTP请求由实现安全门(即,云门)的IDCS web路由层610接收,并提供服务路由以及微服务注册和发现612。取决于所请求的服务,HTTP请求被转发到IDCS中间层614中的IDCS微服务。IDCS微服务处理外部和内部HTTP请求。IDCS微服务实现平台服务和基础设施服务。IDCS平台服务是分开部署的基于Java的运行时服务,其实现了IDCS的业务。IDCS基础设施服务是分开部署的运行时服务,其为IDCS提供基础设施支持。IDCS还包括基础设施库,基础设施库是作为由IDCS服务和共享库使用的共享库打包的公共代码。基础设施服务和库提供平台服务用于实现其功能所需的支持能力。

[0083] 平台服务

[0084] 在一个实施例中,IDCS支持标准认证协议,因此IDCS微服务包括诸如OpenID Connect、OAuth、SAML2、用于跨域身份管理的系统++("SCIM++")等的平台服务。

[0085] OpenID Connect平台服务实现标准的OpenID Connect登录/注销流程。交互式的基于web的本机应用充分利用标准的基于浏览器的OpenID Connect流来请求用户认证,从而接收是传达用户经认证的身份的JavaScript对象标记("JSON")Web令牌("JWT")的标准身份令牌。在内部,运行时认证模型是无状态的,从而以主机HTTP cookie(包括JWT身份令牌)的形式维护用户的认证/会话状态。经由OpenID Connect协议发起的认证交互被委托给可信SSO服务,该服务为本地和联合登录实现用户登录/注销仪式(ceremony)。下面参考图10和11公开这个功能的更多细节。在一个实施例中,根据例如OpenID基础(Foundation)标准来实现OpenID Connect功能。

[0086] OAuth2平台服务提供令牌授权服务。它提供了丰富的API基础设施,用于创建和验证传达进行API调用的用户权限的访问令牌。它支持一系列有用的令牌授予类型,从而使客

户能够安全地将客户端连接到他们的服务。它实现了标准的双参与者(2-legged)和三参与者(3-legged)OAuth2令牌授予类型。支持OpenID Connect(“OIDC”)使得兼容的应用(OIDC中继方(“RP”))与作为身份提供者(OIDC OpenID提供者(“OP”))的IDCS集成。类似地,将IDCS作为OIDC RP与社交OIDC OP(例如,Facebook、Google等)集成使得客户能够允许对应用进行基于社交身份策略的访问。在一个实施例中,根据例如互联网工程任务组(“IETF”)请求注解(“RFC”)6749来实现OAuth功能。

[0087] SAML2平台服务提供身份联合服务。它使得客户能够基于SAML身份提供者(“IDP”)和SAML服务提供者(“SP”)关系模型与其合作伙伴建立联合协议。在一个实施例中,SAML2平台服务实现标准SAML2浏览器POST登录和注销简档。在一个实施例中,根据例如IETF、RFC 7522来实现SAML功能。

[0088] SCIM是用于自动化身份域或信息技术(“IT”)系统之间的用户身份信息交换的开放标准,如由例如IETF RFC 7642、7643、7644提供的。SCIM++平台服务提供身份管理服务,并使客户能够访问IDCS的IDP特征。管理服务暴露覆盖身份生命周期、密码管理、组管理等的无状态REST接口(即,API)集合,从而将这些工件作为web可访问的资源来暴露。

[0089] 所有IDCS配置工件都是资源,并且管理服务的API允许管理IDCS资源(例如,用户、角色、密码策略、应用、SAML/OIDC身份提供者、SAML服务提供者、密钥、证实、通知模板等)。管理服务充分利用和扩展SCIM标准,以便为所有IDCS资源上的创建、读取、更新、删除和查询(“CRUDQ”)操作实现基于模式的REST API。此外,用于管理和配置IDCS自身的所有IDCS内部资源都作为基于SCIM的REST API暴露。对身份存储库618的访问被隔离到SCIM++API。

[0090] 在一个实施例中,例如,SCIM标准被实现,以管理如由SCIM规范定义的用户和组资源,而SCIM++被配置为使用由SCIM标准定义的语言来支持附加的IDCS内部资源(例如,密码策略、角色、设置等)。

[0091] 管理服务在需要的地方利用标准SCIM 2.0核心模式和模式扩展来支持SCIM 2.0标准端点。此外,管理服务还支持若干个SCIM 2.0兼容端点扩展,以管理其它IDCS资源,例如用户、组、应用、设置等。管理服务还支持远程过程调用样式(“RPC样式”)REST接口集合,其不执行CRUDQ操作,而是代替地提供功能服务,例如“UserPasswordGenerator”、“serPasswordValidator”等。

[0092] IDCS管理API使用OAuth2协议进行认证和授权。IDCS支持常见的OAuth2场景,诸如用于web服务器、移动和JavaScript应用的场景。对IDCS API的访问受访问令牌的保护。为了访问IDCS管理API,应用需要通过IDCS管理控制台将应用注册为OAuth2客户端或IDCS应用(在这种情况下,OAuth2客户端是自动创建的)并被授予期望的IDCS管理角色。在进行IDCS管理API调用时,应用首先从IDCS OAuth2服务请求访问令牌。在获取令牌之后,应用通过将访问令牌包括在HTTP授权报头中来将访问令牌发送给IDCS API。应用可以直接使用IDCS管理REST API,或者使用IDCS Java客户端API库。

[0093] 基础设施服务

[0094] IDCS基础设施服务支持IDCS平台服务的功能。这些运行时服务包括:事件处理服务(用于异步处理用户通知、应用预订和数据库审计);作业调度程序服务(用于调度和执行作业,例如,立即执行或在配置的时间执行不需要用户干预的长时间运行的任务);高速缓存管理服务;存储管理服务(用于与公共云存储服务集成);报告服务(用于生成报告和仪表

盘);SSO服务(用于管理内部用户认证和SSO);用户界面("UI")服务(用于托管不同类型的UI客户端);以及服务管理器服务。服务管理器是Oracle公共云和IDCS之间的内部接口。服务管理器管理由Oracle公共云发布的命令,其中命令需要由IDCS实现。例如,当客户在云商店购买东西之前先注册账户时,云会向IDCS发送要求创建租户的请求。在这种情况下,服务管理器实现了云预期IDCS支持的特定于云的操作。

[0095] IDCS微服务可以通过网络接口(即,HTTP请求)调用另一个IDCS微服务。

[0096] 在一个实施例中,IDCS还可以提供允许使用数据库模式的模式服务(或持久性服务)。模式服务允许将管理数据库模式的责任委托给IDCS。因而,IDCS的用户不需要管理数据库,因为存在提供该功能的IDCS服务。例如,用户可以使用数据库以每个租户为基础来持久化模式,并且当数据库中没有更多的空间时,模式服务将管理获得另一个数据库和增加空间的功能,使得用户不需要必须自己管理数据库。

[0097] IDCS还包括数据存储库,这些数据存储库是IDCS所需/生成的数据储存库,包括身份存储库618(存储用户、组等)、全局数据库620(存储由IDCS使用以配置自身的配置数据)、操作模式622(提供每个租户的模式分离并且以每个客户为基础来存储客户数据)、审计模式624(存储审计数据)、高速缓存集群626(存储高速缓存的对象,以加速性能)等。所有内部和外部IDCS客户经基于标准的协议与身份服务进行集成。这使得能够使用域名系统("DNS")来解决将请求路由到何处,并且使得消费应用与对身份服务的内部实现的理解解耦。

[0098] 实时任务和近实时任务

[0099] IDCS将所请求服务的任务分离成同步实时任务和异步近实时任务,其中实时任务仅包括用户继续前进所需的操作。在一个实施例中,实时任务是以最小延迟执行的任务,而近实时任务是在后台执行、无需用户等待它的任务。在一个实施例中,实时任务是基本上没有延迟或以可忽略的延迟被执行的任务,并且对于用户而言几乎是瞬间执行的。

[0100] 实时任务执行具体身份服务的主要业务功能。例如,当请求登录服务时,应用发送消息以认证用户的凭证,并作为回报获得会话cookie。用户体验到的就是登录系统。但是,结合用户的登录可以执行若干其它任务,诸如验证用户是谁、审计、发送通知等。因而,验证凭证是实时执行的任务,使得用户被给予开始会话的HTTP cookie,但是与通知(例如,发送电子邮件以通知创建账户)、审计(例如,跟踪/记录)等相关的任务是可以异步执行的近实时任务,这样用户可以以最少的延迟继续前进。

[0101] 当接收到针对微服务的HTTP请求时,对应的实时任务由中间层中的微服务执行,而剩余的近实时任务(诸如不必实时处理的操作逻辑/事件)被卸载到消息队列628,消息队列628支持具有有保证的递送和处理的高度可伸缩的异步事件管理系统630。因而,某些行为被从前端推送到后端,以使IDCS能够通过减少响应时间中的等待时间来向客户提供高级服务。例如,登录处理可以包括凭证的验证、日志报告的提交、最后登录时间的更新等,但是这些任务可以被卸载到消息队列并且近实时地而不是实时地执行。

[0102] 在一个示例中,系统可能需要注册或创建新的用户。系统调用IDCS SCIM API,以创建用户。最终的结果是,当用户在身份存储库618中被创建时,用户得到包括重置他们的密码的链接的通知电子邮件。当IDCS接收到注册或创建新用户的请求时,对应的微服务查看操作数据库(位于图6中的全局数据库620中)中的配置数据,并确定"创建用户"操作被标

记有“创建用户“事件,这在配置数据中被识别为异步操作。微服务返回给客户端并且指示用户的创建成功完成,但是通知电子邮件的实际发送被推迟并被推送到后端。为了这样做,微服务使用消息传送API 616将消息在作为存储库的队列628中排队。

[0103] 为了从队列628中出队,作为基础设施微服务的消息传送微服务在后台连续地运行并且扫描队列628,以在队列628中查找事件。队列628中的事件由事件订户630处理,诸如审计、用户通知、应用订阅、数据分析等。取决于由事件指示的任务,事件订户630可以与例如审计模式624、用户通知服务634、身份事件订户632等通信。例如,当消息传送微服务在队列628中发现“创建用户”事件时,它执行对应的通知逻辑并向用户发送对应的电子邮件。

[0104] 在一个实施例中,队列628将由微服务614公布的操作事件以及由管理IDCS资源的API 616公布的资源事件排队。

[0105] IDCS使用实时高速缓存结构来增强系统性能和用户体验。高速缓存自身也可以作为微服务提供。IDCS实现弹性高速缓存集群626,其随着IDCS所支持的客户数量的伸缩而增长。高速缓存集群626可以利用下面更详细公开的分布式数据网格来实现。在一个实施例中,只写资源绕过高速缓存。

[0106] 在一个实施例中,IDCS运行时部件向公共云监视模块636公布健康和操作度量,公共云监视模块636从公共云(诸如来自Oracle公司的Oracle公共云)收集这种度量。

[0107] 在一个实施例中,可以使用IDCS来创建用户。例如,客户端应用602可以发布创建用户的REST API调用。管理服务(614中的平台服务)将调用委托给用户管理器(614中的基础设施库/服务),该用户管理器进而在ID存储库618中的特定于租户的ID存储条带中创建用户。在“用户创建成功”时,用户管理器在审计模式624下审计对审计表的操作,并向消息队列628公布“identity.user.create.success”事件。身份订户632拾取事件并向新创建的用户发送“欢迎”电子邮件,包括新创建的登录细节信息。

[0108] 在一个实施例中,可以使用IDCS向用户授予角色,从而导致用户供应动作。例如,客户端应用602可以发布授予用户角色的REST API调用。管理服务(614中的平台服务)将该调用委托给角色管理器(614中的基础设施库/服务),该角色管理器在ID存储库618中特定于租户的ID存储条带状授予用户角色。在“角色授予成功”时,角色管理器在审计模式624下审计对审计表的操作,并向消息队列628公布“identity.user.role.grant.success”事件。身份订户632拾取事件并评估供应授权策略。如果在角色被授予时存在活动的应用授予,则供应订户执行某种验证、发起帐户创建、调出目标系统、在目标系统上创建帐户并将帐户创建标记为成功。这些功能中的每一个可以导致对应事件的公布,诸如“prov.account.create.initiate”、“prov.target.create.initiate”、“prov.target.create.success”或“prov.account.create.success”。这些事件可以具有其自己的聚合过去N天内在目标系统中创建的帐户数量的业务度量。

[0109] 在一个实施例中,IDCS可以用于用户登录。例如,客户端应用602可以使用所支持的认证流之一来请求用户的登录。IDCS对用户进行认证,并且在成功时在审计模式624下审计审计表中的操作。在失败时,IDCS在审计模式624下审计失败,并在消息队列628中公布“login.user.login.failure”事件。登录订户拾取事件、更新其用于用户的度量,并确定是否需要执行对用户的访问历史的附加分析。

[0110] 因而,通过实现“控制反转”功能(例如,改变执行流程以在稍后时间安排操作的执

行,使得操作在另一个系统的控制下),实施例使得附加的事件队列以及订户能够被动态添加,以便在部署到更广泛的用户基础之前针对小的用户样本测试新功能,或者处理针对具体的内部或外部客户的具体事件。

[0111] 无状态功能

[0112] IDCS微服务是无状态的,这意味着微服务本身不维持状态。"状态"是指应用为了执行其功能而使用的数据。IDCS通过将所有状态持久化到IDCS数据层中特定于租户的储存库中来提供多租户功能。中间层(即,处理请求的代码)不具有与应用代码存储在相同位置的数据。因而,IDCS在水平和垂直方面都具有高度的可伸缩性。

[0113] 垂直伸缩(或扩大/缩小)意味着向系统中的单个节点添加资源(或从中移除资源),通常涉及将CPU或存储器添加到单个计算机。垂直可伸缩性允许应用扩大到其硬件的极限。水平伸缩(或扩大/缩小)意味着向系统中添加更多节点(或从中移除节点),诸如将新计算机添加到分布式软件应用。水平可伸缩性允许应用几乎无限地伸缩,仅受网络提供的带宽量限制。

[0114] IDCS的中间层的无状态使得仅通过增加更多的CPU就可以水平伸缩,并且执行应用的工作的IDCS部件不需要具有运行特定应用的指定物理基础设施。IDCS中间层的无状态使得IDCS具有高度的可用性,即使在向大量客户/租户提供身份服务时也是如此。每次通过IDCS应用/服务都只关注CPU使用情况来执行应用事务本身,而不使用硬件来存储数据。伸缩是通过在应用运行时添加更多的片(slice)来完成的,而用于事务的数据存储在持久层上,在那里在需要时可以添加更多的副本。

[0115] IDCS web层、中间层和数据层可以各自独立地和分开地进行伸缩。web层可以伸缩,以处理更多的HTTP请求。中间层可以伸缩,以支持更多的服务功能。数据层可以伸缩以支持更多的租户。

[0116] IDCS功能视图

[0117] 图6A是一个实施例中的IDCS的功能视图的示例框图600b。在框图600b中,IDCS功能堆栈包括服务、共享库和数据存储库。服务包括IDCS平台服务640b、IDCS高级服务650b和IDCS基础设施服务662b。在一个实施例中,IDCS平台服务640b和IDCS高级服务650b是分开部署的、实现IDCS的业务的、基于Java的运行时服务,而IDCS基础设施服务662b是分开部署的、为IDCS提供基础设施支持的运行时服务。共享库包括IDCS基础设施库680b,该IDCS基础设施库680b是作为由IDCS服务和共享库使用的共享库被打包的公共代码。数据存储库是IDCS所需/生成的数据储存库,包括身份存储库698b、全局配置700b、消息存储库702b、全局租户704b、个性化设置706b、资源708b、用户瞬态数据710b、系统瞬态数据712b、每租户模式(受管理的ExaData)714b、操作存储库(未示出)、高速缓存存储库(未示出)等。

[0118] 在一个实施例中,IDCS平台服务640b包括例如OpenID Connect服务642b、OAuth2服务644b、SAML2服务646b和SCIM++服务648b。在一个实施例中,IDCS高级服务包括例如云SSO和治理(governance)652b、企业治理654b、AuthN中介656b、联合中介(broker)658b和私人帐户管理660b。

[0119] IDCS基础设施服务662b和IDCS基础设施库680b提供IDCS平台服务640b完成其工作所需的支持能力。在一个实施例中,IDCS基础设施服务662b包括作业调度程序664b、UI 666b、SSO 668b、报告670b、高速缓存672b、存储装置674b、服务管理器676b(公共云控制)和

事件处理器678b(用户通知、应用订阅、审计、数据分析)。在一个实施例中,IDCS基础设施库680b包括数据管理器API 682b、事件API 684b、存储API 686b、认证API 688b、授权API 690b、cookie API 692b、密钥API 694b和凭证API 696b。在一个实施例中,云计算服务602b(内部Nimbula)支持IDCS基础设施服务662b和IDCS基础设施库680b的功能。

[0120] 在一个实施例中,IDCS为IDCS服务的消费者提供各种UI 602b,诸如客户最终用户UI 604b、客户管理UI 606b、DevOps管理UI 608b和登录UI 610b。在一个实施例中,IDCS允许应用(例如,客户应用614b、合作伙伴应用616b和云应用618b)的集成612b和固件集成620b。在一个实施例中,各种环境可以与IDCS集成,以支持其访问控制需求。这种集成可以由例如身份桥622b(提供AD集成、WNA和SCIM连接器)、Apache代理624b或MSFT代理626b提供。

[0121] 在一个实施例中,内部和外部IDCS消费者经基于标准的协议628b(诸如OpenID Connect 630b、OAuth2 632b、SAML2 634b、SCIM 636b和REST/HTTP 638b)与IDCS的身份服务集成。这使得能够使用域名系统(“DNS”)来解决将请求路由到何处,并且使得消费应用与理解身份服务的内部实现解耦。

[0122] 图6A中的IDCS功能视图还包括公共云基础设施服务,其提供IDCS为了用户通知(云通知服务718b)、文件存储(云存储服务716b)以及用于DevOps的度量/警告(云监视服务(EM)722b和云度量服务(Graphite)720b)而依赖的常见功能。

[0123] 云门

[0124] 在一个实施例中,IDCS在web层中实现“云门”。在一个实施例中,云门是通过web服务器插件来实现的,web服务器插件使web应用能够将用户SSO外部化到身份管理系统(例如,IDCS),类似于与企业IDM堆栈一起工作的WebGate或WebAgent技术。在一个实施例中,云门充当保护对IDCS API的访问的安全守卫。在一个实施例中,云门由web/代理服务器插件实现,该插件提供用于基于OAuth保护HTTP资源的web策略实施点(“PEP”)。在一个实施例中,云门还执行认证(例如,OAuth认证)。

[0125] 在一个实施例中,可以部署云门来保护单租户应用或多租户应用,并且可以将云门部署在代理模式前端不同类型的应用中。在一个实施例中,云门通过使用<租户>(表示什么租户拥有云门实例)和<主机标识符>(表示正在访问什么应用)相应的默认配置来支持单租户和多租户应用。在一个实施例中,本地配置为这些工件指定默认值,并且可以通过由路由层提供的HTTP报头在每个请求的基础上被重写。例如,对于多租户应用,<租户>定义了云门的租赁,用于在联系IDCS服务器时构建IDCS服务URL,并且可以在每个请求的基础上被可选的X-RESOURCE-IDENTITY-DOMAIN-NAME报头重写,该报头包括<租户-名称>(例如,“acme”)。进一步地,对于多租户应用,<主机标识符>定义cookie域,并且用于管理本地会话cookie,并且可以在每个请求的基础上被由客户端或路由层提供的可选“主机”报头(例如“acme.pcs1.dc4.oraclecloud.com”)重写。

[0126] 在一个实施例中,为每个租户部署云门的一个实例。然而,在替代实施例中,例如,在提供复制和故障转移的高可用性服务器集群中,可能存在为租户部署的云门的多个实例。在一个实施例中,云门提供了多租户模式,该模式(例如当“基础设施”云门位于IDCS自身前面(而不是租户的应用的前面)时)允许单个云门实例支持多个租户,并且在将来自多个租户的请求传递给IDCS服务之前处理这些请求。

[0127] 在一个实施例中，云门通过根据传递给它的请求的报头值确定特定租户来支持多租户。云门替代了各种URL中的租赁，并使用URL查找特定于应用/主机的策略。云门将这些策略应用于请求，并使用该信息将请求传递给IDCS的适当部件。

[0128] 在一个实施例中，传入请求中指出的租户不被替换(即，传入请求不被修改)，而是用于云门自身向IDCS服务作出的请求(例如，策略查找、凭证验证等)中。在一个实施例中，为云门供应了IDCS服务的URL，并且这些URL可以具有租户的占位符(例如，“http://%tenant%.idcs.oracle.com”)或者可以指出云门的租户。在该实施例中，占位符或云门的租户被传入请求中指出的租户替换，使得云门使用传入请求中所指定的租户向IDCS服务作出请求。

[0129] 例如，云门可能正在保护企业云中的备审(docketing)应用。该应用可以属于正在被请求的IDCS服务(例如，可以是IDCS开发的应用，例如租户专用或用户专用管理控制台)，或者可以是IDCS的租户开发的。对应于IDCS的相应租户的多个客户可以使用该备审应用。当客户需要访问该应用时，它向IDCS发送请求，并且在请求的报头中指出客户的租赁。云门根据该请求推断租赁，并查找需要针对该租赁应用的特定策略。然后，云门将策略应用于该请求，并且如果策略允许请求通过，则云门断言租赁，并允许请求到达备审应用。在一个实施例中，云门通过向请求添加具有租户名称的报头来断言租赁，然后将请求转发到源服务器。在一个实施例中，云门可以以这种方式仅断言一些租户，并且其他租户的报头可能已经存在于请求中或者可能已经由先前的负载均衡服务器添加。

[0130] 在一个实施例中，策略规定是否允许对通过请求端点标识的特定资源的访问，以及允许什么特定的访问方法取得对资源的访问。例如，在一个实施例中，“/admin/v1/HTTPAuthenticator”是用于验证基于浏览器的HTTP基本认证的用户凭证的端点，在这种情况下资源是凭证验证服务。例如，在一个实施例中，租户的公开访问策略可以指出该租户中的任何人都可以访问资源(例如，“/ui/v1/sign”(该资源是登录页面的公开可见端点)、“/ui/v1/pwdmustchange”和“/ui/v1/resetpwd”，用于密码更改和重置页面等)。当租户的公开访问策略指出该租户中的任何人都可以访问资源时，云门允许来自该租赁的访问该资源的任何请求通过。另一个策略可能要求用户需要提供用户名和密码以便(例如，基于基本认证)访问资源，并且对照身份对用户名和密码进行验证。进一步策略可能需要基于OAuth的基于令牌的认证，在这种情况下，资源需要用户或编程应用提供OAuth访问令牌来取得对后端中资源的访问。例如，用户可以发起OpenID流程以取得OAuth访问令牌来取得对资源的访问，或者客户端应用可以通过提前取得令牌并且然后提供令牌及其凭证来取得对资源的访问，从而获得对资源的编程访问。

[0131] 在一个实施例中，在由云门管理的文件中指定策略。在一个实施例中，文件存储在云门本地。在替代或附加实施例中，文件可以存储在IDCS内的服务器侧(例如，在策略存储库中)。

[0132] 图7是实现云门702的实施例的框图700，云门702在web服务器712中运行并充当被配置为使用开放标准(例如，OAuth2，OpenID Connect等)与IDCS策略决定点(“PDP”)集成的策略实施点(“PEP”)，同时保护对应用的web浏览器和REST API资源714的访问。在一些实施例中，PDP在IDCS的OAuth和/或OpenID Connect微服务704处实现。

[0133] 在图7的实施例中，程序或应用客户端708可以尝试访问受云门702保护的资源

714,或者最终用户710可以在其计算机上使用web浏览器706来访问受云门702保护的资源714。在一个实施例中,云门702可以实现用于保护资源714的不同功能,这取决于最终用户710或编程应用客户端708是否正在访问资源714。

[0134] 在一个实施例中,当用户浏览器706向IDCS发送用户710的登录的请求时,对应的IDCS PDP验证凭证,然后决定凭证是否充足(例如,是否请求诸如第二密码之类的进一步的凭证)。在图7的实施例中,云门702既可以充当PEP,也可以充当PDP,因为它具有本地会话。在一个实施例中,云门通过处理策略实施和确定必须如何(或是否)认证用户来充当PEP,而由PDP处理实际认证。在一个实施例中,如果(例如经由“公开”策略)不需要认证,或者如果存在云门的会话cookie并且可以执行验证cookie来代替验证凭证,则云门可以充当PDP。

[0135] 例如,在一个实施例中,为了访问资源714,用户710可以在浏览器706中键入相应的URL。浏览器706将用户710带到web服务器712。然后,云门702加载相应的策略。如果策略指出资源714受到保护,则云门702发起任何所需的认证处理。在一个实施例中,云门702不直接质询(challenge)用户(即,它不负责任何UI),但是一旦用户被质询以便进行认证,云门702就验证用户凭证(例如,用于基本认证(Basic Auth))。例如,当用户710输入他们的用户名和密码时,云门702将用户名和密码发送到IDCS中的对应微服务704(例如,OAuth2)以认证用户710。如果认证成功执行,则云门702允许用户710访问资源714。

[0136] 在一个实施例中,云门702充当HTTP基本认证认证器,从而针对IDCS验证HTTP基本认证凭证。这种行为在无会话和基于会话(本地会话cookie)模式下均受支持。在这种情况下,不创建服务器侧的IDCS会话。对于HTTP基本认证,云门702将HTTP“未授权”状态代码返回给浏览器,作为响应,浏览器显示提示输入用户名/密码的对话框。这些凭证被提交给云门702,云门702针对IDCS对它们进行认证。

[0137] 在一个实施例中,对于根据OAuth的认证,云门702返回HTTP重定向状态代码,该HTTP重定向状态代码将浏览器引导到OAuth微服务,该微服务与SSO微服务一起处理实际登录流程,包括在浏览器中显示登录页面和认证用户提供的凭证。在基于web浏览器的用户访问期间,云门702充当发起用户认证流程的OIDC RP 718。如果用户710没有有效的本地用户会话,则云门702将用户重定向到SSO微服务并且与SSO微服务一起参与OIDC“授权码”流程。该流程以递送JWT作为身份令牌结束。云门708验证JWT(例如,查看签名、到期、目的地/受众等)并且为用户710发布本地会话cookie。它充当会话管理器716,该会话管理器716保护web浏览器访问受保护的资源以及发布、更新并验证本地会话cookie。它还提供了用于移除其本地会话cookie的注销URL。

[0138] 例如,在一个实施例中,为了访问受云门702保护的资源714,编程应用客户端708向web服务器712作出对应的请求,并且应用客户端708与该请求一起发送OAuth令牌。云门702接收令牌并连接到OAuth服务704以对照OAuth 2服务704验证OAuth令牌。一旦令牌被验证,云门702就允许应用客户端708访问受保护资源714。

[0139] 在一个实施例中,应用客户端708可以在(例如,由管理员执行的)预注册处理中取得令牌。在预注册处理中,应用客户端708在IDCS中将自身注册为被允许访问受保护资源714的可信应用。令牌可以由IDCS OAuth微服务生成。

[0140] 作为一次部署的一部分,云门702作为OAuth2客户端向IDCS注册,从而使其能够针对IDCS请求OIDC和OAuth2操作。其后,它根据请求匹配规则(如何匹配URL,例如,通配符、正

则表达式等)来维护关于应用的受保护资源和不受保护资源的配置信息。可以部署云门702以保护具有不同安全策略的不同应用,并且受保护的应用可以是多租户的。

[0141] 在REST API客户端708的编程访问期间,云门702可以充当应用的受保护的REST API 714的OAuth2资源服务器/过滤器720。它检查具有授权报头和访问令牌的请求的存在。当客户端708(例如,移动装置、web应用、JavaScript等)呈现(由IDCS发布的)访问令牌以与受保护的REST API 714一起使用时,云门702在允许访问API(例如,签名、到期、受众等)之前验证访问令牌。原始访问令牌未经修改就被传递。

[0142] 图7A是图示云门702B如何处理主机标识符的示例框图700B。每个主机标识符表示与受云门702B保护的应用相关联的逻辑web域。在图7A的示例实施例中,路由层即服务("RTaaS")704B接收的URL的通用格式是:

[0143] https://<tenant>.<service>.<dc>.oraclecloud.com/<resource>

[0144] 在起始处标识租赁(<tenant>),然后标识服务(<service>)、数据中心(<dc>)、根域(<oraclecloud.com>),在结尾处标识资源API端点(<resource>)。在一个实施例中,在请求处理期间的运行时,云门702B使用由客户端或路由层(即RTaaS 704B)经由"主机"报头提供的主机标识符。如果没有来自路由层的主机标识符可用,或者没有找到匹配,则云门702B从其自己的"<默认主机标识符>"构建主机标识符。在一个实施例中,IDCS路由层可以为应用使用多于一个的主机标识符,并且在相应主机标识符的上下文中处理通过云门702B的每个请求,包括对本地会话cookie的范围界定和对访问策略文件的解析。例如,当云门702B接收到URL时,它查看对应的主机标识符并确定需要应用什么策略。

[0145] 在一个实施例中,RTaaS 704B具有多个物理web代理,每个代理具有在RTaaS IDCS帐户708B中定义的物理云门702B。每个web代理及其对应的云门702B支持多个虚拟主机(例如,1至30000之间数量的虚拟主机)。每个虚拟主机代表由RTaaS 704B指派的每个服务实例706B的web层。关于每个请求,RTaaS 704B设置具有虚拟主机的名称的报头(例如,"coke.docs.dc4.oraclecloud.com"、"pepsi.mcs.dc4.oraclecloud.com"、"intel.pcs.dc4.oraclecloud.com"等)。在一个实施例中,云门702B处的会话cookie被限定为虚拟主机的范围,并且实现了适当的重新进入功能。例如,云门702B可以在"redirect_uri"端点714B上接收AZ代码以设置cookie,并且可以在"logout_uri"端点712B上接收清除cookie的请求。

[0146] 在一个实施例中,每个物理云门实例暴露在为该云门实例注册的OAuth简档中定义的对应的固定"redirect_uri"端点714B和对应的固定"logout_uri"端点712B。URI用于云门702B和OIDC之间的交互。在一个实施例中,OIDC针对请求源自的云门的已知重定向URI来验证"redirect_uri"参数。在一个实施例中,就比较字符串值而言,匹配不必"精确",但是无论匹配算法如何,匹配都必须满足它是云门的相同物理实例的条件。在一个实施例中,"logout_uri"保存在IDCS 710B中作为各个云门的注册OAuth简档的一部分,并且该云门在其任何交互中不将"logout_uri"传递给服务器。当完全构建时,假设"redirect_uri"和"logout_uri"包括租户模板,并且在被相应的云门702B和OIDC使用之前被转化。

[0147] 在一个实施例中,云门702B和IDCS服务器710B之间的交互将物理云门702B和主机标识符的组合视为一个逻辑OAuth RP。在由云门702B向IDCS OIDC微服务发起的操作中,以及由IDCS OIDC微服务向云门702B发起的操作中,主机标识符经由"X-HOST-IDENTIFIER-

NAME”查询字符串参数被发送。当云门702B调用登录/注销请求时，云门702B从其从客户端或路由层接收到的主机标识符报头发送主机标识符查询字符串参数。当跟踪IDCS OIDC微服务所创建的请求cookie中的RP时，主机标识符查询字符串参数的值对于每个RP保存在请求cookie中。当IDCS OIDC微服务调用RP的“redirect_uri”和“logout_uri”时，IDCS OIDC微服务从保存在请求cookie中的主机标识符值发送主机标识符查询字符串参数。

[0148] 一般而言，OAuth用于生成客户端身份传播令牌(例如，指示客户端是谁)或者用户身份传播令牌(例如，指示用户是谁)。在这些实施例中，云门中OAuth的实现基于JWT，JWT定义用于web令牌的格式，如由例如IETF、RFC 7519提供的。

[0149] 当用户登录时，发布JWT。JWT由IDCS签署，并支持IDCS中的多租户功能。云门验证由IDCS发布的JWT以允许IDCS中的多租户功能。因而，IDCS在物理结构中以及在支撑安全模型的逻辑业务处理中提供多租赁。

[0150] 租赁类型

[0151] IDCS指定三种类型的租赁：客户租赁、客户端租赁和用户租赁。客户或资源租赁指定IDCS的客户是谁(即，正在为谁执行工作)。客户端租赁指定哪个客户端应用在试图访问数据(即，哪个应用正在做工作)。用户租赁指定哪个用户在使用该应用来访问数据(即，由谁来执行工作)。例如，当专业服务公司为仓储俱乐部提供系统集成功能并使用IDCS为仓储俱乐部系统提供身份管理时，用户租赁与专业服务公司对应，客户端租赁是用于提供系统集成功能的应用，并且客户租赁是仓储俱乐部。

[0152] 这三种租赁的分离和识别在基于云的服务中启用多租户功能。一般而言，对于安装在内部部署的物理机器上的内部部署软件，由于用户需要在机器上物理地登录，因此不需要指定三种不同的租赁。但是，在基于云的服务结构中，实施例使用令牌来确定谁在使用什么应用来访问哪些资源。这三种租赁由令牌编码，由云门强制实施并由中间层的业务服务使用。在一个实施例中，OAuth服务器生成令牌。在各种实施例中，令牌可以与除OAuth以外的任何安全协议结合使用。

[0153] 解耦用户、客户端和资源租赁为IDCS提供的服务的用户提供了实质性的业务优势。例如，它允许服务提供者了解业务(例如，医疗保健业务)的需求和他们的身份管理问题，以购买IDCS提供的服务、开发消费IDCS的服务的自己的后端应用，并向目标业务提供后端应用。因而，服务提供者可以扩展IDCS的服务，以提供其期望的能力并将那里能力提供给某些目标业务。服务提供者不需要构建和运行软件来提供身份服务，而是可以代替地扩展和定制IDCS的服务以适应目标业务的需求。

[0154] 一些已知的系统仅解决了客户租赁这单一的租赁。但是，这种系统在处理由用户(诸如客户用户、客户的合作伙伴、客户的客户端、客户端本身或者客户委托访问的客户端)的组合进行的访问时是不足够的。在实施例中定义和强制实施多种租赁促进对这各种用户的身份管理功能。

[0155] 在一个实施例中，IDCS的一个实体不同时属于多个租户；它只属于一个租户，而“租赁”是工件存活的地方。一般而言，存在实现某些功能的多个部件，并且这些部件可以属于租户，或者它们也可以属于基础设施。当基础设施需要代表租户行事时，它代表租户与实体服务进行交互。在那种情况下，基础设施本身具有它自己的租赁，并且客户具有其自己的租赁。在提交请求时，请求中可以涉及多种租赁。

[0156] 例如,属于“租户1”的客户端可以执行用于获得让“租户2”指定“租户3”中的用户的令牌的请求。作为另一个示例,存在于“租户1”中的用户可能需要在由“租户2”拥有的应用中执行动作。因此,用户需要去“租户2”的资源名称空间并为他们自身请求令牌。因而,授权的委托通过识别“谁”可以对“谁”做“什么”来完成。作为又一个示例,为第一组织(“租户1”)工作的第一用户可以允许为第二组织(“租户2”)工作的第二用户有权访问由第三组织(“租户3”)托管的文档。

[0157] 在一个示例中,“租户1”中的客户端可以请求让“租户2”中的用户访问“租户3”中的应用的访问令牌。客户端可以通过转到“http://tenant3/oauth/token”来调用对令牌的OAuth请求。客户端通过在请求中包括“客户端断言”将自身识别为存在于“租户1”中的客户端。客户端断言包括客户端ID(例如,“客户端1”)和客户端租赁“租户1”。作为“租户1”中的“客户端1”,该客户端有权调用对“租户3”上的令牌的请求,并且客户端想要用于“租户2”中的用户的令牌。因而,“用户断言”也作为同一个HTTP请求的一部分被传递,以将“租户2”识别为用户的租赁,并且请求的报头将“租户3”识别为资源/应用的租赁。因此,请求会识别客户端的租赁、用户的租赁和资源的租赁。所生成的访问令牌将在作为应用租赁(“租户3”)的目标租赁的上下文中发布,并将包括用户租赁(“租户2”)。因此,访问令牌识别资源/应用的租赁和用户的租赁。

[0158] 在一个实施例中,在数据层中,每个租户被实现为分离的条带。从数据管理的角度来看,工件存在于租户中。从服务的角度来看,服务知道如何与不同的租户共事,而多个租户是服务业务功能的不同维度。图8图示了在一个实施例中实现多个租赁的实例系统800。系统800包括客户端802,客户端802请求由理解如何与数据库806中的数据共事的微服务804提供的服务。数据库806包括多个租户808,并且每个租户包括对应租赁的工件。在一个实施例中,微服务804是由“租户1”中的客户端通过https://tenant3/oauth/token请求的OAuth微服务,用于获取“租户2”中的用户访问“租户3”中的应用的令牌,并且数据库806存储客户端(“租户1”)的租赁、用户(“租户2”)的租赁以及资源/应用(“租户3”)的租赁的数据。在微服务804中使用来自数据库806的数据执行核实客户端802的请求是合法的OAuth微服务的功能,并且如果是合法的,则使用来自不同租赁808的数据来构造令牌。因而,系统800是多租户的,因为通过不仅支持进入每个租赁的服务,而且还支持代表不同租户行事的服务,它可以在跨租户环境中工作。

[0159] 系统800是有利的,因为微服务804在物理上与数据库806中的数据解耦,并且通过在更靠近客户端的位置复制数据,微服务804可以作为本地服务被提供给客户端并且系统800可以管理服务的可用性并在全球提供。

[0160] 在一个实施例中,微服务804是无状态的,这意味着运行微服务804的机器不维护将服务指向任何具体租户的任何标记。代替地,例如,可以在进入的请求的URL的主机部分上标记租赁。那种租赁指向数据库806中的租户808之一。当支持大量租户(例如,数百万租户)时,微服务804不能具有到数据库806的相同数量的连接,而是代替地使用连接池810,其在数据库用户的上下文中提供到数据库806的实际物理连接。一般而言,通过向底层驱动器或提供者供给连接字符串来构建连接,连接字符串被用于寻址具体的数据库或服务器并提供实例和用户认证凭证(例如,“Server=sql_box;Database=Common;User ID=uid;Pwd=Password;”)。一旦连接已经建立,它就可以被打开和关闭,并且可以设置特性(例如,命

令超时长度,或事务(如果存在的话))。连接字符串包括由数据提供者的数据访问接口规定的键-值对的集合。连接池是所维护的数据库连接的高速缓存,使得在将来需要对数据库的请求时可以重用连接。在连接池中,在创建连接之后,它被放在池中并被再次使用,使得不必建立新连接。例如,当微服务804和数据库808之间需要10个连接时,在连接池810中将存在10个打开的连接,全部在数据库用户的上下文中(例如,与具体的数据库用户相关联,例如,谁是该连接的所有者、谁的凭证正在验证中、是否为数据库用户、是否为系统凭证等)。

[0161] 连接池810中的连接是为可以访问任何东西的系统用户创建的。因此,为了正确处理由代表租户处理请求的微服务804进行的审计和特权,在与指派给具体租户的模式所有者相关联的“代理用户”812的上下文中执行数据库操作。这个模式所有者只能访问为其创建模式的租赁,并且租赁的价值是模式所有者的价值。当请求数据库806中的数据时,微服务804使用连接池810中的连接来提供该数据。因而,多租赁是通过使无状态的、弹性的中间层服务在特定于租户的数据存储库绑定的上下文中(例如,与其相关联)处理传入的请求来实现的,其中特定于租户的数据存储库绑定是以每个请求为基础的在与资源租赁相关联的数据存储库代理用户的上下文中(例如,与其相关联)创建的数据连接之上建立的,并且数据库可以独立于服务进行伸缩。

[0162] 以下提供了用于实现代理用户812的示例功能:

[0163] dbOperation=<准备要执行的DB命令>

[0164] dbConnection=getDBConnectionFromPool()

[0165] dbConnection.setProxyUser(resourceTenant)

[0166] result=dbConnection.executeOperation(dbOperation)

[0167] 在这个功能中,微服务804将在从连接池810拉出的连接上的“代理用户”设置设置为“租户”,并且在使用连接池810中的数据库连接的同时在租户的上下文中执行数据库操作。

[0168] 当将每个表分条以针对不同租户配置同一个数据库中的不同列时,一个表可以包括混合在一起的所有租户的数据。相反,一个实施例提供租户驱动的数据层。该实施例不为不同租户将同一个数据库分条,而是代替地为每个租户提供不同的物理数据库。例如,多租赁可以通过使用可插拔数据库(例如,来自Oracle公司的Oracle数据库12c)来实现,其中每个租户被分配分离的分区。在数据层,资源管理器处理请求,然后请求用于该请求的数据源(与元数据分离)。该实施例依据请求执行到相应数据源/存储库的运行时切换。通过将每个租户的数据与其他租户隔离,该实施例提供了改进的数据安全性。

[0169] 在一个实施例中,各种令牌编码不同的租赁。URL令牌可以识别请求服务的应用的租赁。身份令牌可以编码将被认证的用户的身份。访问令牌可以识别多个租赁。例如,访问令牌可以对作为这种访问的目标的租赁(例如,应用租赁)以及被给予访问的用户的用户租赁进行编码。客户端断言令牌可以识别客户端ID和客户端租赁。用户断言令牌可以识别用户和用户租赁。

[0170] 在一个实施例中,身份令牌至少包括“声称(claim)”[如安全领域的普通技术人员所使用的],其指示用户租户名称(即,用户所在的地方)。与授权令牌联系的“声称”是一个主体关于其自身或另一个主体作出的声明(statement)。例如,声明可以是关于名称、身份、密钥、组、特权或能力的。声称是由提供者发布的,并且它们被给给予一个或多个值,然后打

包在由发布者发布的安全令牌中,通常被称为安全令牌服务(“STS”)。

[0171] 在一个实施例中,访问令牌至少包括指示在对访问令牌作出请求时的资源租户名称(例如,客户)的声称/声明、指示用户租户名称的声称/声明、指示作出请求的OAuth客户端的名称的声称/声明,以及指示客户端租户名称的声称/声明。

[0172] 在一个实施例中,访问令牌可以根据以下JSON功能来实现:

[0173] {

[0174] …

[0175] “tok_type”:“AT”,

[0176] “user_id”:“测试用户”,

[0177] “user_tenantname”:“<身份租户的值>”

[0178] “tenant”:“<资源租户的值>”

[0179] “client_id”:“测试客户端”,

[0180] “client_tenantname”:“<客户端租户的值>”

[0181] …

[0182] }

[0183] 在一个实施例中,客户端断言令牌至少包括指示客户端租户名称的声称/声明以及指示作出请求的OAuth客户端的名称的声称/声明。

[0184] 本文描述的令牌和/或多种租赁可以在除IDCS以外的任何多租户的基于云的服务中实现。例如,本文描述的令牌和/或多租赁可以在SaaS或企业资源规划(“ERP”)服务中实现。

[0185] 图9是一个实施例中的IDCS的网络视图900的框图。图9图示了在一个实施例中在应用“区”904之间执行的网络交互。基于所需的保护级别和到各种其它系统(例如,SSL区、无SSL区等)的连接的实现,将应用分成区。一些应用区提供需要从IDCS内部进行访问的服务,而一些应用区提供需要从IDCS外部进行访问的服务,并且一些应用区域是开放访问。因而,针对每种区强制实施相应的保护级别。

[0186] 在图9的实施例中,服务到服务通信是使用HTTP请求来执行的。在一个实施例中,IDCS使用本文描述的访问令牌不仅提供服务,而且还保护对IDCS自身的访问以及在IDCS自身内的访问。在一个实施例中,IDCS微服务通过REST性的接口暴露并由本文所述的令牌保护。

[0187] 在图9的实施例中,各种应用/服务902中的任何一个都可以对IDCS API进行HTTP调用以使用IDCS服务。在一个实施例中,应用/服务902的HTTP请求经历Oracle公共云负载均衡外部虚拟IP地址(“VIP”)906(或其它类似技术)、公共云web路由层908以及IDCS负载均衡内部VIP应用910(或其它类似的技术),以被IDCS web路由层912接收。IDCS web路由层912接收来自IDCS的外部或内部的请求,并且跨IDCS平台服务层914或IDCS基础设施服务层916路由它们。IDCS平台服务层914包括从IDCS外部调用的IDCS微服务,诸如OpenID Connect、OAuth、SAML、SCIM等。IDCS基础设施服务层916包括支持从IDCS的内部调用的微服务,以支持其它IDCS微服务的功能。IDCS基础设施微服务的示例是UI、SSO、报告、高速缓存、作业调度程序、服务管理器、用于制作密钥的功能等。IDCS高速缓存层926支持用于IDCS平台服务层914和IDCS基础设施服务层916的高速缓存功能。

[0188] 通过强制实施在外部访问IDCS和IDCS内部的安全性,IDCS的客户可以被提供对于他们运行的应用的卓越的安全合规性。

[0189] 在图9的实施例中,除了基于结构化查询语言(“SQL”)通信的数据层918和基于LDAP通信的ID存储层920之外,OAuth协议也被用于保护IDCS内的IDCS部件(例如,微服务)之间的通信,并且用于保护来自IDCS外部的访问的相同令牌也被用于IDCS内的安全性。即,web路由层912使用相同的令牌和协议来处理它接收到的请求,而不管请求是从IDCS外部还是从IDCS内部接收到的。因而,IDCS为保护整个系统提供了单一一致的安全模型,由此允许卓越的安全合规性,因为在系统中实现的安全模型越少,系统越安全。

[0190] 在IDCS云环境中,应用通过进行网络调用来进行通信。网络调用可以基于适用的网络协议,诸如HTTP、传输控制协议(“TCP”)、用户数据报协议(“UDP”)等。例如,通过将应用Y作为HTTP统一资源定位符(“URL”)暴露,应用X可以基于HTTP与应用Y通信。在一个实施例中,Y是暴露各自与能力对应的多个资源的IDCS微服务。当X(例如,另一个IDCS微服务)需要调用Y时,它构造包括Y和需要被调用的资源/能力的URL(例如https:/host/Y/resource),并进行对应的REST调用,该REST调用经过web路由层912并被定向到Y。

[0191] 在一个实施例中,IDCS之外的调用者可以不需要知道Y在哪里,但是web路由层912需要知道应用“Y”在哪里运行。在一个实施例中,IDCS实现发现功能(由OAuth服务的API实现),以确定每个应用在哪里运行,因此不需要静态路由信息的可用性。

[0192] 在一个实施例中,企业管理器(“EM”)922提供将内部部署的和基于云的管理扩展到IDCS的“单一虚拟管理平台(single pane of glass)”。在一个实施例中,是来自Chef Software公司的配置管理工具的“Chef”服务器924为各种IDCS层提供配置管理功能。在一个实施例中,服务部署基础设施和/或持久性存储模块928可以将OAuth2HTTP消息发送到IDCS web路由层912,以进行租户生命周期管理操作、公共云生命周期管理操作或其它操作。在一个实施例中,IDCS基础设施服务层916可以将ID/密码HTTP消息发送到公共云通知服务930或公共云存储服务932。

[0193] 云访问控制-SSO

[0194] 一个实施例支持用于实现云规模的SSO服务的轻量级云标准。轻量级云标准的示例是HTTP、REST以及通过浏览器提供访问的任何标准(因为web浏览器是轻量级的)。相反,SOAP是重量级云标准的示例,其需要更多的管理、配置和工具来构建客户端。该实施例对于应用使用OpenID Connect语义来针对IDCS请求用户认证。该实施例使用轻量级的基于HTTP cookie的用户会话跟踪来在IDCS处跟踪用户的活动会话,而无需有状态的服务器端会话支持。该实施例对于应用使用基于JWT的身份令牌,以在将经认证的身份映射回其自己的本地会话时使用。该实施例支持与联合的身份管理系统的集成,并且暴露用于企业部署的SAML IDP支持以针对IDCS请求用户认证。

[0195] 图10是一个实施例中的IDCS中的SSO功能的系统架构视图的框图1000。该实施例使得客户端应用能够充分利用基于标准的web协议来发起用户认证流程。需要将SSO与云系统集成的应用可以位于企业数据中心中、远程合作伙伴数据中心中,或者甚至由客户内部部署操作。在一个实施例中,不同的IDCS平台服务实现SSO的业务,诸如OpenID Connect,用于处理来自所连接的本机应用(即,利用OpenID Connect与IDCS集成的应用)的登录/注销请求;SAML IDP服务,用于处理来自所连接的应用的基于浏览器的登录/注销请求;SAML SP

服务,用于编排针对外部SAML IDP的用户认证;以及内部IDCS SSO服务,用于编排包括本地或联合登录流程的最终用户登录仪式以及用于管理IDCS主机会话cookie。一般而言,HTTP可以带表单或不带表单工作。当它带表单工作时,表单就会在浏览器中被看到。当它不带表单工作时,它作为客户端到服务器的通信。OpenID Connect和SAML都需要能够呈现表单,这可以通过存在浏览器来实现,或者通过充当浏览器起作用的应用虚拟执行。在一个实施例中,通过IDCS实现用户认证/SSO的应用客户端需要作为OAuth2客户端在IDCS中注册,并且需要获得客户端标识符和凭证(例如,ID/密码、ID/证书等)。

[0196] 图10的示例实施例包括共同提供登录能力的三个部件/微服务,包括两个平台微服务:OAuth2 1004和SAML2 1006,以及一个基础设施微服务:SSO 1008。在图10的实施例中,IDCS提供“身份元系统”,其中在不同类型的应用(诸如需要三参与方OAhth流程并充当OpenID Connect中继方(“RP”,将其用户认证功能外包给IDP的应用)的基于浏览器的web或本机应用1010、需要双参与方OAuth流程并充当OpenID Connect RP的本机应用1011以及充当SAML SP的web应用1012)上提供SSO服务1008。

[0197] 一般而言,身份元系统是用于数字身份的可互操作架构,从而允许基于多种底层技术、实现和提供者来采用数字身份的集合。LDAP、SAML和OAuth是提供身份能力并且可以是用于构建应用的基础的不同安全标准的示例,并且身份元系统可以被配置为在这种应用之上提供统一的安全系统。LDAP安全模型指定用于处理身份的具体机制,并严格保护系统的所有次通过。开发SAML,以允许一组应用安全地与属于不同安全域中的不同组织的另一组应用交换信息。由于这两个应用之间没有信任,因此开发了SAML,以允许一个应用对另一个不属于同一组织的应用进行认证。OAuth提供了用于执行基于web的认证的轻量级协议的OpenIDConnect。

[0198] 在图10的实施例中,当OpenID应用1010连接到IDCS中的OpenID服务器时,其“通道”请求SSO服务。类似地,当SAML应用1012连接到IDCS中的SAML服务器时,其“通道”也请求SSO服务。在IDCS中,相应的微服务(例如,OpenID微服务1004和SAML微服务1006)将处理每个应用,并且这些微服务从SSO微服务1008请求SSO能力。通过针对每个协议添加微服务,然后对于SSO能力使用SSO微服务1008,可以扩展这个架构,以支持任意数量的其它安全协议。SSO微服务1008发布会话(即,提供SSO cookie 1014)并且是架构中具有发布会话的权限的唯一系统。IDCS会话通过浏览器1002使用SSO cookie 1014来实现。浏览器1002还使用本地会话cookie 1016来管理其本地会话。

[0199] 在一个实施例中,例如在浏览器内,用户可以使用基于SAML的第一应用并且登录,并且随后使用由诸如OAuth之类的不同协议构建的第二应用。用户被提供有同一浏览器内的第二应用上的SSO。因而,浏览器是状态或用户代理并且维护cookie。

[0200] 在一个实施例中,SSO微服务1008提供登录仪式1018、ID/密码恢复1020、首次登录流程1022、认证管理器1024、HTTP cookie管理器1026以及事件管理器1028。登录仪式1018实现基于客户设置和/或应用上下文的SSO功能,并且可以根据本地表单(即,基本认证)、外部SAML IDP、外部OIDC IDP等进行配置。使用ID/密码恢复1020来恢复用户的ID和/或密码。当用户第一次登录时(即,SSO会话尚不存在时),实现首次登录流程1022。认证管理器1024在成功认证时发布认证令牌。HTTP cookie管理器1026将认证令牌保存在SSO cookie中。事件管理器1028公布与SSO功能相关的事件。

[0201] 在一个实施例中，OAuth微服务1004和SSO微服务1008之间的交互基于浏览器重定向，使得SSO微服务1008使用HTML表单来质询用户、验证凭证并发布会话cookie。

[0202] 在一个实施例中，例如，OAuth微服务1004可以从浏览器1002接收授权请求，以根据三参与方OAuth流程来认证应用的用户。OAuth微服务1004然后充当OIDC提供者1030、将浏览器1002重定向到SSO微服务1008，并沿着应用上下文传递。取决于用户是否具有有效的SSO会话，SSO微服务1008验证现有会话或者执行登录仪式。在成功认证或验证后，SSO微服务1008将认证上下文返回到OAuth微服务1004。然后OAuth微服务1004用授权(“AZ”)代码将浏览器1002重定向到回调URL。浏览器1002将AZ代码发送到OAuth微服务1004，以请求所需的令牌1032。浏览器1002还在HTTP授权报头中包括其客户端凭证(当在IDCS中注册为OAuth2客户端时获得的)。作为回报，OAuth微服务1004向浏览器1002提供所需的令牌1032。在一个实施例中，提供给浏览器1002的令牌1032包括由IDCS OAuth2服务器签署的JW身份和访问令牌。这个功能的进一步细节在下面参考图11公开。

[0203] 在一个实施例中，例如，OAuth微服务1004可以从本机应用1011接收授权请求，以根据双参与方OAuth流程来认证用户。在这种情况下，OAuth微服务1004中的认证管理器1034执行对应的认证(例如，基于从客户端1011接收到的ID/密码)，并且令牌管理器1036在成功认证后发布对应的访问令牌。

[0204] 在一个实施例中，例如，SAML微服务1006可以从浏览器接收SSO POST请求，以认证充当SAML SP的web应用1012的用户。然后，SAML微服务1006充当SAML IDP 1038，将浏览器1002重定向到SSO微服务1008，并沿着应用上下文传递。取决于用户是否具有有效的SSO会话，SSO微服务1008验证现有会话或者执行登录仪式。在成功认证或验证后，SSO微服务1008将认证上下文返回给SAML微服务1006。然后SAML微服务用所需的令牌重定向到SP。

[0205] 在一个实施例中，例如，SAML微服务1006可以充当SAML SP1040并前往远程SAML IDP 1042(例如，活动目录联合服务(“ADFS”))。一个实施例实现标准SAML/AD流程。在一个实施例中，SAML微服务1006和SSO微服务1008之间的交互基于浏览器重定向，使得SSO微服务1008使用HTML表单来质询用户、验证凭证并发布会话cookie。

[0206] 在一个实施例中，通过防火墙1044执行IDCS内的部件(例如，1004、1006、1008)与IDCS外的部件(例如，1002、1011、1042)之间的交互。

[0207] SSO服务

[0208] 在一个实施例中，在IDCS的云环境中，当租户被供应并且需要对其应用的保护时，IDCS提供身份功能，诸如认证、授权、审计等。当用户需要使用该应用时，IDCS提供用户身份的完全安全性，并且生成特定类型的令牌，并为该应用建立或保护用户访问。

[0209] 在一个实施例中，SSO服务用于保护不同类型的客户端/应用，其中对每个特定应用的访问由标准协议(诸如OAuth或SAML)保护。协议规定了令牌格式、需要什么令牌以及生成特定令牌需要什么浏览器语义。一旦接收到令牌，应用可以使用基于令牌的声称来给予对特定用户的访问。在一个实施例中，令牌要求用户是谁、行为者是谁、范围是什么以及被访问的受众是什么。在一个实施例中，令牌对于OAuth和SAML是不同的，并且包括用户在认证之后建立的用户声称/声明的语义。

[0210] 通常，基于标准的协议(诸如OAuth和SAML)未规定或提供任何指出如何生成令牌或如何认证用户的规范。相反，它们仅需要对用户进行认证，并规定应该获得哪种令牌。例

如,对于OAuth或SAML令牌,协议仅规定了令牌的语义,但没有规定或指定应该如何对用户进行认证、用户如何登录、注销等。

[0211] 然而,在一个实施例中,IDCS SSO服务定义了如何认证特定用户(例如,认证是否是单因素、双因素、生物特征认证等)。在该实施例中,登录策略或访问策略指出如何对用户进行认证以访问应用。在一个实施例中,在执行认证之后,IDCS将认证转换成IDCS会话或“全局会话”(例如,被配置为会话cookie或会话令牌)。全局会话是具有特定语义/格式的云感知会话,并被配置用于IDCS多租户环境。取决于要为应用执行的特定身份任务,IDCS可以将全局会话转换为适用令牌(例如,OAuth令牌、SAML令牌等),并将令牌提供给应用。

[0212] 在一个实施例中,SSO服务是可用于任何协议(例如,OAuth、SAML、社交令牌、社交服务等,或任何新开发的规范)的公共控制器系统。在一个实施例中,当SSO服务接收到请求时,将请求参数归一化。因此,SSO服务可以保持通用,并为不同类型的令牌服务。SSO服务不需要知道特定的协议语义(例如,特定于OAuth的语义或特定于SAML的语义)。

[0213] 在一个实施例中,SSO服务将完整的登录仪式定义为公共逻辑。登录仪式规定用户如何登录、哪些因素用于进行认证、系统如何执行因素认证、用户和IDCS SSO服务之间的编排如何发生等。SSO服务还定义注销仪式,该仪式规定如何从全局会话注销以及如何复制到不同的协议。SSO服务提供协议驱动的应用注销。

[0214] 在一个实施例中,完整的登录仪式发生在SSO服务和UI交互之间。登录仪式取决于用户用于登录的因素(例如,用户ID和密码、基于证书的认证、基于移动因素的认证等)。

[0215] 一个实施例将SSO服务实现为公共控制器,并且还将认证模块插件实现为每个登录仪式的后端认证器。例如,对于基于用户ID和密码的认证,该实施例实现了前往包括用户ID和密码的身份存储库的插件。然后,插件对到该存储库的用户凭证进行认证。作为另一示例,如果用户提供用户凭证的证书(诸如X.509证书),则SSO服务确定适当的认证模块,在这种情况下该模块是证书模块。证书模块前往证书存储库,并对照证书存储库验证所提供的证书。在一个实施例中,用户可以提供移动认证因素,诸如短消息服务(“SMS”)或基于时间的一次性密码(“TOTP”)。

[0216] 在一些实施例中,SSO服务可以确定前往多因素认证插件来认证用户。SSO服务编排对每个插件的要求。在各种实施例中,插件可能需要与用户进行一次交互,诸如一个用户ID和密码,或者可能需要与用户进行多次交互,诸如多个页面。

[0217] 因而,SSO服务为不同类型的用户认证编排整个登录仪式。一旦用户被认证,SSO服务就建立全局会话。在一个实施例中,全局会话填充有各种用户信息,例如用户ID、用户偏好、用户区域、用户时区、向其认证的因素或模块、认证时间等。

[0218] 在一个实施例中,在IDCS的云架构中,服务被配置为无状态的和水平可伸缩的,并且在服务器中不维护任何状态。因而,由于服务无法维护客户端会话的会话状态,因此作为全局会话的整个会话都放在客户端cookie中。即,该实施例将全局会话配置为客户端cookie,并且客户端会话是其中保持整个全局会话的cookie。在一个实施例中,cookie是请求报头的一部分,并且每次请求以“302”返回到浏览器,并且重定向返回到另一个服务,cookie被重放(即,报头被重放)。

[0219] 在一个实施例中,使用协议缓冲器来存储全局会话的内容,以最小化其尺寸,使得全局会话可以被包括在cookie中。一个实施例提供用于保持全局会话、优化其尺寸以及将

cookie或客户端会话转换成特定的基于协议的令牌的语义。

[0220] 在一个实施例中，当用户已经登录到系统(因此在系统中具有全局会话)并且正在利用同一浏览器会话访问另一应用时，调用第一流程。在该流程中，协议层(例如，图10中的1004和1006)不确定用户是否具有有效会话，因为他们不理解全局会话。相反，它们请求SSO服务来确定用户是否有有效的全局会话。SSO服务任何进入其中该服务与有效会话取决于协议而编排不同的流程的模式。SSO服务基于某些参数来确定会话的有效性，并且如果用户具有有效的全局会话，则SSO服务生成特定于协议的令牌并给予访问权限。

[0221] 在一个实施例中，当会话的有效性不足并且需要断言用户并且需要本地身份时，调用第二流程。

[0222] 在一个实施例中，调用第三流程用于第一次登录。当(使用相应的认证模块)对用户进行认证时，并且在执行编排并且SSO服务将要创建会话之后，该实施例确保用户存在于本地IDCS用户存储库中。用户需要具有本地身份存在(例如，如果用户已经被远程认证，则可能不是密码，但至少是一些最少量简档信息)。类似地，在存在现有有效会话的流程中，SSO服务在本地存储库中断言用户，并确保用户存在于本地IDCS身份存储库中。

[0223] 一个实施例为联合系统提供一组流程。联合提供跨域SSO，可能包括SP、IDP或其他行为者和实体。例如，SP和IDP可能在不同的域上(例如，cookie域、身份域等)，并且会话可能需要使用联合协议(诸如SAML2)从一个域转移到另一个域。在一个域中，该实施例可以创建不能被另一个域读取的本地会话或全局会话。因此，在转到另一个域(也必须是可信域)之前，该实施例将全局会话转换为SAML令牌，并将SAML令牌转移到另一个域中，在该域中可以对其进行验证和断言。

[0224] 在一个实施例中，当表现为SAML SP或IDP时，SSO服务编排了不同的流程。一般而言，当用户不在本地时，用户被认证为远程IDP。例如，用户可能具有使用云服务的云帐户，但可能是不希望将其用户身份复制到云存储库中的内部部署用户或现有企业用户。由于用户身份是内部部署的，所以内部部署成为远程IDP。在这种情况下，本地IDCS SSO服务表现为SP并联系远程IDP。

[0225] 当用户进入充当SP的SSO服务时，SSO服务使用联合协议(诸如SAML)前往远程IDP。认证在远程IDP处执行，并在远程IDP上生成SAML令牌。远程IDP将SAML令牌发送到SSO服务。IDCS SAML服务提取SAML令牌，从SAML令牌中提取所需的属性/声明(例如，认证声明、授权声明、名称、ID等)，以特定格式将该信息归一化，然后将其传递给SSO服务。IDCS SAML服务不执行断言。SSO服务不需要编排用户登录仪式，因为远程认证已经由远程IDP内部部署执行。相反，SSO服务以归一化的属性格式获取归一化的SAML状态，并在本地系统中断言该状态(断言这些属性)。

[0226] 一个实施例为联合提供不同的选项/标志或配置，例如，用户是否必须存在于本地，即使它是远程认证的，是否忽略用户在本地系统中的存在等。在此基础上，SSO服务在本地断言用户或者仅断言声称，然后创建全局会话。然后，SSO服务将请求交还给协议层(即SAML服务)。

[0227] 在一个实施例中，整个IDCS系统中的全局会话是要生成的任何协议令牌的真实来源。SSO服务基于其编排的不同类型的流程(诸如新的认证、现有会话、用于联合的远程认证、用于OAuth的远程认证或社交登录等)而生成全局会话。然后，IDCS系统可以为不同的协

议生成令牌并将令牌交还给应用,并且基于令牌,客户端可以访问租户应用。

[0228] 归一化状态

[0229] 一个实施例通过为不同协议(例如SAML和OAuth)提供归一化参数来支持全局会话,同时OAuth和SAML服务中的每一个服务接收它们相应的令牌。例如,SAML服务接收SAML断言,SAML断言是带有报头和主体的XML块。报头定义令牌的类型或用于生成认证的安全机制的类型,例如可以是X.509、用户ID/密码等。在一个实施例中,SAML令牌是根据SAML2规范的。SAML服务提取SAML令牌,读取SAML令牌,并验证SAML令牌的每个部分。令牌的主体可以包括SAML断言,诸如名称ID、认证声明(指出远程IDP使用的认证的类型)等。

[0230] 令牌还包括授权声明(指出用于授权的属性)和认证已经转移了一些属性之后的属性声明(指出远程IDP)。在一个实施例中,远程IDP具有用户的真实身份存储库,并使用该身份存储库来认证用户,并基于属性声明要求而发回SAML令牌中的一些属性。需要发送的属性是基于IDP和SP之间建立的SAML配置来确定的。在一个实施例中,在实体成为SP的IDP或IDP的SP之前,两个实体都必须是可信实体,并且存在作为先决条件执行的元数据交换。它们还共享它们的证书(例如,公钥/证书)以建立彼此需要的信任和配置。当SP获取SAML断言时,属性声明基于SP已经向IDP规定的配置声明(例如,在生成SAML令牌后将向SP发送哪些属性)。

[0231] 在一个实施例中,SAML服务只知道SAML语义。它可以打开SAML令牌、验证签名,并且如果它是加密的,则基于签名和加密算法对其进行解密。这些算法也是SP和IDP之间的配置的一部分。在一个实施例中,所有这些步骤都在协议层执行,其中SAML服务从令牌中获取属性。在一个实施例中,SAML服务将属性归一化为键/值对(key/value pair)(例如,属性名称和值)。在该实施例中,键/值对配置是不基于任何特定协议的通用格式。SSO服务将属性的归一化状态作为键/值对接收。在一个实施例中,在协议级声称中对OAuth令牌执行相同的归一化。也就是说,OAuth声称也以相同的键/值对格式被归一化。

[0232] 在一个实施例中,协议可以给出作为认证协议的一部分的属性、需求属性和/或在建立全局会话之后规定在执行认证/验证之后SSO服务应该返回到哪里(例如,基于协议服务的返回URL,在该返回URL上进行回调)。例如,对于SAML,由SSO服务接收属性,而对于OAuth,SSO服务提供属性,因为它对用户进行了认证。例如,在一个实施例中,SAML协议可以从远程实体(例如,远程IDP)接收属性,或者指出它需要返回属性。在OAuth的情况下,协议规定在建立全局会话后,协议需要一些属性返回来为最终应用生成基于OAuth的声称。在一个实施例中,OAuth或SAML可能需要SSO服务前往某些端点/URL。这些信息作为来自两种协议的状态信息回到SSO服务。

[0233] 在一个实施例中,SSO服务和去往/来自协议层(例如,SAML和OAuth服务)的通信是通过“重定向”(通过HTTP 303)进行的,并且它们共享加密cookie中的公共状态。加密的cookie存储协议信息和归一化格式,该格式被配置为简洁并且适合不同的浏览器限制(例如,报头尺寸、cookie尺寸等)。

[0234] 请求状态

[0235] 在实施例的云架构中,服务是无状态的。因而,一个实施例提供了流经不同的层来彼此交互的共同状态。例如,当协议层在不存在SSO会话时联系(HTTP重定向到)SSO服务以进行认证时,当协议层联系SSO服务以进行会话验证和与现有会话的用户断言等时,该实施

例在各种流程的持续时间内(以加密主机cookie的形式)保持客户端的状态。

[0236] 在一个实施例中,公共状态包括客户端会话和客户端状态。该实施例提供了“请求状态”,并将其作为报头包含在“请求cookie”中。请求cookie不是SSO cookie或认证会话,而是跨不同服务(例如,协议服务、SSO服务等)维护的预认证状态。它还包括执行上下文(例如,诸如审计、诊断等的状态)。归一化状态(例如,属性、返回URL、不同流程所需的内容等)也在请求状态下被捕获并放入请求cookie中。在一个实施例中,该cookie被加密,并且加密密钥被安全地存储并定期滚动(例如,每12小时滚动一次),使得它不会被劫持。

[0237] 在一个实施例中,存储在请求状态中的状态信息不包括诸如密码的用户安全数据。在一个实施例中,即使在协议之间的交互期间,也只有声称被存储在请求状态中,并且签名被附加到声称中。

[0238] 在一个实施例中,在登录处理(例如,登录仪式或不同的流程)中使用的请求状态也可以在注销处理中使用,因为该状态包括关于用户已经访问并且需要针对协议返回或注销的不同端点的信息、每个协议的归一化状态、声称、在身份系统中断言用户所需的属性等。

[0239] 在一个实施例中,为了在各种流程的持续时间内(以加密主机cookie的形式)保持客户端的状态,实现了以下表示(representation):

[0240] • “UserRequestData”,这是每当协议层向SSO服务发起新的请求用于创建/验证现有会话时创建的;

[0241] • “UserSessionData”,这是由SSO服务在完成对来自协议层的请求的处理后创建的(一旦创建了该表示,“UserRequestData”就被删除,并且所有相关数据保存在该表示中);

[0242] • “AbstractUserData”,这是“UserRequestData”和“UserSessionData”的父级,包括两者使用的公共状态数据结构。

[0243] 表1提供了在一个实施例中在“UserRequestData”表示中包括的功能。

[0244] 表1:“UserRequestData”表示中包括的示例功能

[0245]

字符串 ssoRequestRemotePartnerID	由OAuth/SAML-IDP服务器设置的OAuth客户端ID或SAML SP ID,用于标识发起请求的协议
字符串 ssoRequestRemotePartnerName	与用于全局审计的合作伙伴ID相对应的远程合作伙伴的名称
字符串 ssoRequestProtectedRequestedURL	由最终用户/客户端访问的、触发来自协议层的认证请求的URL

字符串ssoRequestReferrer	由协议层向发起请求的引用者（referrer）设置。它还用于标识特殊情况，诸如“碎玻璃流程（break-glass flow）”（例如，为了在使用和完成单个远程IDP时，通过使用管理控制台来避免无休止的循环）
字符串ssoRequestCallbackURL	由协议层设置为SSO服务器在处理请求后必须恢复到的URL。它也可以由SAML-SP服务器在IDP发起的SSO流程中设置，并指向返回IDP发送的URL/中继状态
字符串 logoutDoneURL	注销完成后，SSO 服务将把用户重定向到的 URL，可以为空（null）

[0246]

[0247] 在一个实施例中，“UserSessionData”表示包括“byte[]authContext”，其是用户会话数据的简明二进制表示，例如由“AuthContext”对象定义的属性。协议缓冲器库用于以二进制格式有效地表示数据。使用滚动的私钥(例如每12小时滚动一次)对其进行加密。表2提供了在一个实施例中在“AuthContext”表示中所包括的功能。

[0248] 表2：“AuthContext”表示中包括的示例功能

[0249]

字符串userId	登录到用户全局唯一 ID（“GUID”）
字符串userEmailId	用户的电子邮件 ID 主体
字符串sessionId	当前会话 ID
字符串userDisplayName	用户的显示名称
字符串tenantName	用户的租赁
Map<String, String>声称	用户的认证声称（例如来自 IDCS（本地）IDP 或远程 IDP 的用户属性）

[0250]

long（长整型） sessionExpirationTime	会话到期时间
long sessionCreationTime	会话创建时间
字符串userLoginId	本地认证情况下的用户登录属性值
字符串userMappingAttr	本地认证情况下的用户登录属性名称

[0251] 表3提供了在一个实施例中在“AbstractUserData”表示中所包括的示例功能。

[0252] 表3：“AbstractUserData”表示中包括的示例功能

[0253]

Map<Integer, byte[]> componentData	保存抽象语法符号标记1（“ASN.1”）或每个部件的数据的协议缓冲器编码的序列化二进制表示，范围为请求或会话
Date（日期）generationTime	服务器处的请求或会话cookie设置时间
字符串tenantName	会话的用户租赁

[0254] 在一个实施例中，对应于“Map<Integer,byte[]>componentData”的部件可以是例如SSO、OAuth2和SAML服务器。数据是二进制的，因此可以被加密并且可以从部件数据表示中抽象出私钥管理。作为“UserSessionData”的一部分，它保存会话的持续期间向其发布令牌/断言的每个客户端或合作伙伴的跟踪数据。这也有助于全局注销。作为“UserRequestData”的一部分，它保存部件级请求范围的数据。

[0255] 在一个实施例中，SSO服务实现资源管理器和SSO运行时。资源管理器是IDCS的公共数据访问层。它是元数据驱动的，因此它可以管理在符合SCIM的资源类型和模式定义中所定义的任何资源类型。资源管理器处理所有资源类型的所有公共逻辑，包括对属性、数据类型、所需值、典型(canonical)值等的基于模式的验证。它还处理设置默认值、设置“创建/更新日期”值、设置“创建者/更新者”值、保护敏感属性、映射到目标属性、授权检查和事件发布。资源类型数据管理器的外部接口是HTTP/REST(部署在IDCS管理服务器中，并且由“serviceUp”命令提出)，其内部实现是通过Java接口实现的。在一个实施例中，基于Java特定请求(“JSR”)330，使用百千字节内核(“HK2”)来注入实现。资源类型有效负载遵循由SCIM 2.0概述的资源模型。例如，“/Applications”和“/Credentials”是特定的资源类型。

[0256] 能够通过“/ResourceTypes”和“/Schemas”发现资源定义和模式。资源类型数据管理器管理每个租户的操作数据(例如，用户、组、令牌、资源类型、模式等)、租户设置(跨服务租户特定设置和服务特定租户特定设置)和全局配置(跨服务全局配置和服务特定全局配置)。

[0257] SSO运行时

[0258] 图10A图示了一个实施例中的SSO运行时的示例框图1000A，其中SSO服务1002A实现了本地和联合登录的用户登录/注销仪式。云SSO集成架构使用标准的OpenID Connect用户认证流程进行基于浏览器的用户登录。因为云登录基于OpenID Connect，所以OpenID提供者(“OP”，在图1000A中示为OAuth OP 1004A)是登录请求到达SSO 1002A的位置。内部地，运行时认证模型是无状态的，从而以主机HTTP cookie的形式维护用户的SSO状态。

[0259] 用户登录请求流程如下：用户访问基于浏览器的应用或移动/本地应用；连接应用(OAuth RP 1006A或应用的强制实施点)检测到用户没有本地应用会话，并要求用户被认证；连接应用(OAuth RP 1006A或应用的强制实施点)发起针对OAuth2服务的OpenID Connect登录流程(范围＝“openid简档”的三参与方AZ授予流程)；并且OAuth2服务(OAuth OP 1004A)构建应用上下文并重定向到SSO服务1002A。

[0260] 在一个实施例中，OAuth OP 1004A和SSO 1002A之间的服务等级协议(“SLA”)包括

以下内容：

[0261] ●IDCS_REQUEST-IDCS_REQUEST HTTP cookie用于维护SSO 1002A和通道部件之间的各种请求/响应流程之间的状态。

[0262] ○CallbackURL:这是由OAuth OP 1004A在重定向到SSO 1002A之前设置的。SSO 1002A在处理后重定向到该URL。

[0263] ○ErrorCode:这是SSO 1002A如何就发生的任何错误返回与OAuth OP 1004A进行通信。

[0264] ○ErrorMessage:本地化SSO错误消息。

[0265] ●IDCS_SESSION

[0266] ○如果认证状态为“成功”,则SSO 1002A创建具有认证的IDCS_SESSION,并将用户重定向到IDCS_REQUEST中的/callBackUrl。

[0267] ○在有效的IDCS_SESSION已经存在的情况下,SSO 1002A断言从IDCS_SESSION cookie中取出的用户。

[0268] 一个实施例还可以实现持久cookie,该持久cookie可以存储用户偏好,诸如用户场所(locale)。该cookie可以用于将错误消息转换为用户特定的场所。

[0269] 在一个实施例中,根据IDCS微服务架构,UI服务1008A(登录应用/注销)被实现为单独的服务,而非SSO 1002A的一部分。UI服务1008A和SSO服务1002A经由在图1000A中的REST层1010A中实现的REST端点彼此进行交互,并且状态信息作为JSON有效负载在SSO服务1002A和REST层1010A之间传递。

[0270] 在一个实施例中,SSO服务1002A包括REST层1010A、SSO控制器1012A、凭证收集器1014A、认证管理器1016A(用于插入在不同模块中并在每个模块上调用、认证和验证)、会话管理器1018A、SSO事件管理器1020A、SSO客户端API集成器1022A、错误处理1024A和断言器(基于不同身份进行断言,未在图1000A中示出)。在一个实施例中,当认证请求在“/sso”到达SSO 1002A的GET方法时,该请求包括由OAuth OP 1004A(对应于OAuth RP 1006A)或SAML IDP 1005A(对应于SAML SP 1007A)作为IDCS_REQUEST HTTP cookie提交给SSO 1002A的请求和状态数据。IDCS_REQUEST cookie被解密和解析,并且参数和其他HTTP报头在请求上下文对象中被捕获,并被传递给SSO控制器1012A。SSO 1002A通过重定向到登录页面(由UI服务1008A实现)来收集用户的凭证,并且登录页面向SSO 1002A的POST方法提交凭证。状态信息作为JSON有效负载在它们之间传递。

[0271] 在一个实施例中,SSO控制器1012A是根据请求的状态确定要采取什么动作的类。可能存在指示凭证收集、凭证验证、第一次用户、忘记密码、密码重置等的各种状态。SSO控制器1012A根据各种条件将流程编排到其他部件。

[0272] 一个实施例支持两种形式的认证:基本(Basic)和表单形式。(Form)在凭证收集器1014A处收集所需的凭证,诸如用户名、密码和租户名称,并传递以进行认证。这个类准备要被发送到登录UI的JSON请求,并解析从登录应用接收到的响应。

[0273] 认证管理器1016A是验证IDCS用户ID/密码和/或发布认证令牌的类。可能存在各种认证结果,诸如认证成功、无效凭证、禁用用户、密码已过期、密码需要重置等。正确的错误代码作为SSO响应的一部分被返回。认证管理器1016A与用户/身份管理器1030A(管理服务中的资源管理器)的“/UserAuthenticator”REST端点进行对话以执行对凭证的实际验

证。在一个实施例中，在认证管理器1016A处创建用户的认证上下文1017A(全局会话)和认证声称。如果认证上下文1017A已经存在于请求中，则通过与相同的用户认证服务进行对话而非验证凭证来断言令牌。

[0274] 在一个实施例中，会话管理器1018A与Cookie管理器1028A进行通信并管理SSO cookie，包括在连接应用上存储认证上下文和元数据。Cookie管理器1028A是与所有其他层共享的公共层，并生成和优化全局会话状态，使其成为客户端会话cookie。在成功认证或验证后，SSO服务1002A使用包括用户的认证令牌的新创建/更新的SSO主机HTTP cookie重定向回OAuth服务。SSO服务1002A从cookie中移除特定于请求的数据。以下是用于从HTTP小服务程序(Servlet)请求中读取cookie数据的示例功能：

[0275] CookieManager cm＝CookieManagerFactory.createInstance(request, response)；

[0276] UserSessionData sessionData＝cm.getSessionData()；

[0277] String myAuthnToken＝sessionData.getAuthnToken()；

[0278] byte[]mySamlData＝

[0279] sessionData.getComponentData(CookieManagerComponents.SAML)；

[0280] sessionData.setAuthnToken("4141FB74579EAB131ECF6369....")；

[0281] cm.setSessionData(sessionData)；

[0282] cm.commit()；

[0283] 在一个实施例中，HTTP请求中的cookie数据的示例如下：

[0284] 两个cookie:

[0285] 条目(Entry)#1:

[0286] 名称:"ORA_OCIS_1"

[0287] 值:"624790624790578067acbe05a58a589c5595346236e57a5c373b83e3758ac38"

[0288] 条目#2:

[0289] 名称:"ORA_OCIS_2"

[0290] 值:"6246bef44242bf6246bef44242bf6246bef44242bf6246bef44242bf～4～6246bef44242bf"

[0291] 在一个实施例中，SSO事件管理器1020A与事件管理器1032A进行通信，并公布与SSO流程相关联的事件。一个实施例定义SSO事件及其目标是什么(例如，审计、分析、通知、度量等)。以下是对应的示例功能：

[0292] ssoservice.authentication.success

[0293] ssoservice.authentication.failure

[0294] ssoservice.server.startup

[0295] ...

[0296] import oracle.idaas.common.event.*;

[0297] Event event＝new

[0298] EventImpl.Builder(SystemEventId.SYSTEM_ERROR_RUNTIME)

[0299] .ecid("123-456-7890").rid("relationship id")

[0300] .eventValue("event value").serviceName("service name")

[0301] .actorId("actor id").actorName("actor name")

[0302] .tenantName("test tenant").message("test message")

[0303] .eventData("name3","value3").build();

[0304] eventManager.publish(event);

[0305] 在一个实施例中,SSO客户端API集成器1022A与资源管理器1026A进行通信,并且特定于租户的SSO设置存储在管理服务器的资源管理器1026A中并可用。经由REST请求,在SSO运行时服务器中读取SSO设置,诸如AuthN方案、会话超时等。因为读取这些设置涉及网络调用,所以只要有回调机制来获得这些设置的任何更改的通知,就可以为每个租户缓存这些设置。以下是资源管理器1026A的功能的示例:

[0306] String url＝"http://"+hostName+":"+port;

[0307] IdaasClient client＝IdaasClient.login(tenantName,username,password);

[0308] client.setTargetURI(URI.create(url));

[0309] SSOSettingsService service＝

[0310] client.getService(SSOSettingsService.class);

[0311] SSOSettings ssoSettings＝service.get("SSOSettings",null);

[0312] ssoSettings.getCookieSessionTimeout()

[0313] ssoSettings.getMaxLoginAttempts()

[0314] 在一个实施例中,UI服务1008A为消费SSO服务1002A的最终用户提供交互点,并提供诸如以下操作:

[0315] ●从最终用户收集凭证用于认证、密码管理以及通过基于JSON的rest调用来调用相应的SSO服务端点;

[0316] ●处理用SSO服务1002A发起的JSON rest调用的各种错误场景;

[0317] ●通过SSO服务1002A发起的重定向来处理SSO服务1002A引起的以及在与协议层交互期间遇到的各种错误。

[0318] 在一个实施例中,SSO控制器1012A根据以下流程为上述情况提供UI服务1008A。

[0319] 在协议三参与方最终用户流程中:

[0320] ●最终用户访问受RP 1006A保护的应用(例如,特定于协议的网关守护代理);

[0321] ●RP 1006A触发协议交互,并且协议层重定向到端点“/sso/v1/user/login”;

[0322] ●在IDAAS_REQ cookie中设置上下文后,端点在SSO服务1002A上返回;

[0323] ●SSO服务1002A读取IDAAS_REQ cookie中的租户报头,并基于租户配置,通过重定向到端点“/member/v1/signin”来调用远程IDP 1036A或登录UI服务1008A;

[0324] ●UI服务1008A收集凭证,并经由JSON请求将它们提交给端点“SSO/v1/user/login”上的post方法;

[0325] ●SSO服务1002A验证凭证。在失败时,用分级(例如,2级:主要和次要)错误代码来响应UI服务1008A。在密码重置时,下一个操作(“nextOp”)和相应的错误代码被发送到UI服务1008A。在成功时,如果启用了post登录,则相应的nextOp被发送到UI服务1008A。在成功时,如果禁用了post登录,则JSON响应被返回,其中nextOp作为“重定向”,有效负载中带有“redirectUrl”。

[0326] “redirectUrl”是从IDAAS_REQ cookie获得的协议层的回调URL。

[0327] 在任一个成功情况下,在响应时IDCS_SSO cookie都会被创建,并在用户浏览器中进行设置;

[0328] ●如果启用了post登录,并且操作由UI完成,在异议或同意的情况下,控制通过JSON请求返回到SSO服务1002A;

[0329] ●SSO服务1002A用同意/异议更新IDAAS_REQ cookie,并发送带有作为“重定向”的nextOp并且有效负载中带有“redirectUrl”的响应,;

[0330] ●协议层读取IDCS_SSO Cookie并生成特定于协议的令牌以返回到RP 1006A;

[0331] ●RP 1006A设置读取令牌的特定于应用的cookie,然后用户被授予访问权限,并可以使用该cookie来访问应用。

[0332] 在来自云门的基本认证流程中:

[0333] ●最终用户/客户端使用用户名和密码授权报头来访问受云门保护的资源。

[0334] ●云门验证资源受基本认证保护,并使用post HTTP请求和有效负载中的凭证数据来调用“/sso/v1/user/login”处的SSO端点。

[0335] ●缺少IDCS_REQUEST并且存在属性:“RPID”、“Referrer”是将SSO控制器流程分叉到不同处理程序:“BasicAuthHandler.java”以处理基本认证的条件。

[0336] ●“BasicAuthHandler”仅执行对JSON有效负载中的凭证的验证,并以成功/失败信息进行响应,而不使用认证上下文来设置IDCS_SSO cookie。

[0337] ●它还处理用于认证的SSO事件,并使用JSON响应来传播进行凭证验证期间的错误代码。

[0338] 在从UI自动登录(首次使用)和密码更改流程中:

[0339] ●租户管理员从管理员UI为用户和他们的电子邮件进行配置,并向用户发送包括带有限期令牌的URL的电子邮件。

[0340] ●用户点击链接以使用令牌来访问UI服务1008A。UI服务1008A用身份服务来验证令牌并收集凭证。

[0341] ●UI服务1008A将凭证发布到身份服务以更新到后端。

[0342] ●UI服务1008A还向SSO控制器发布端点“/sso/v1/user/login”,用于自动登录,其中在JSON有效负载中设置了“redirectURL”属性,带有操作/op:“cred_submit”,没有IDCS_REQ cookie。这是由“CredSubmitHandler”处理的。

[0343] ●SSO控制器1012A按照以下优先级顺序来获取重定向URL:

[0344] 1)来自IDCS_REQUEST的回调URL;

[0345] 2)仅当IDCS_REQUEST cookie不存在时,JSON有效负载中的“redirectURL”属性;

[0346] 3)如果“IDCS_REQUEST”cookie或“redirectURL”属性都不存在,则在成功认证时使用“我的服务”URL以进行重定向。

[0347] SSO状态和流程

[0348] 图10B图示了一个实施例中SSO功能的示例状态和流程图1000B。图10B的实施例提供了每个身份操作(诸如忘记密码、帐户日志、来自SAML SP和IDP的请求等)的SSO控制器引擎流程图的示例。在该实施例中,SSO服务1002a具有模块化配置,其中模块使用全局状态彼此交互以及与协议层(例如,OpenID connect、SAML等)交互。该实施例提供了以安全方式传送状态的新机制。状态的大小被有效地配置,使得状态占用较小的空间,并且操作执行得更

快。该实施例以自适应的方式进行配置以确保它能够被增强到多因素认证、逐步认证等。该实施例提供了自适应配置，以便可以容易地添加附加的认证插件，并且可以提供附加的逐步认证。该实施例还可以使用相同的部件/模块来执行其他功能，诸如基于策略的授权检查等。

[0349] 在一个实施例中，协议层(例如，OAuth、SAML等)和SSO服务1002a之间的交互通过HTTP执行。它们通过请求上下文中的某些状态差异(在cookie中解析)进行通信，并且它们(基于该状态中的固定属性)总是知道返回到哪里。例如，当SSO服务1002a获取控制并基于它从其获取请求的协议确定去哪里时，该协议已经将最终URL或返回URL设置为该状态，并且SSO服务1002a基于在认证之后或成功或失败之后SSO服务1002a发现了什么来跟随该状态。类似地，当协议获取一些关于例如令牌请求的信息时，它会跟踪请求cookie中的指定部件。

[0350] 在一个实施例中，状态包括不同的映射对象，包括每个协议的部件数据。例如，当OAuth获取授权请求时，它包括指出从谁那里获取请求的跟踪信息等。在一个实施例中，OAuth跟踪特定部件数据中的信息并初始化请求cookie。用于在请求cookie中写入的关键字(key)在部件(例如，OAuth、SAML、SSO等)之间共享，并且这些部件之间有关于例如应该写什么数据和写在哪里、什么时候应该写数据等的共同的理解。例如，当实施例必须记录“n”次往返时，对于应该在请求cookie中放入多少数据以及如何放入有共同的理解，因此cookie对下一个部件是尽可能可用的。状态作为加密和安全的cookie流过。当在将数据作为加密的cookie流动方面存在安全顾虑时，一个实施例将状态配置为加密的有效负载post而非在“302”上进入的请求cookie。实施例作为HTTP post进行接口(port)。

[0351] 例如，在一个实施例中，RP 1016a可以从浏览器接收访问受保护资源的请求。RP 1016a向IDCS OP/IDP/SAML SP 1020a发送302/authorize或302SSO/SLO，IDCS OP/IDP/SAML SP 1020a进而向SSO控制器1004a发送带有IDCS_REQUEST应用上下文的302/sso/v1/user login。

[0352] IDCS登录流程是SSO、OP、SAML IDP、SAML SP和登录应用(凭证收集和密码管理JS客户端)之间的一系列请求/响应流程以提供Web单点登录。IDCS_REQUEST http cookie用于维护SSO和通道部件之间的各种请求/响应流程之间的状态。IDCS_REQUEST cookie的关键字将在SSO和通道部件之间共享。登录应用和SSO服务将经由REST端点相互交互。状态信息将作为JSON有效负载在它们之间传递。

[0353] 请求行为

[0354] IDCS SSO服务1002a处理以下不同类型的请求：(1)来自IDCS OP/IDP服务1020a的请求；(2)来自IDCS SP服务1020a的请求；(3)来自登录应用的Ajax HTTP POST请求，从登录应用1042b提交认证凭证；(4)成功完成密码管理操作之后，来自登录应用1042b的Ajax HTTP POST请求；以及(5)成功完成Post登录操作之后，来自登录应用1042b的Ajax HTTP POST请求。

[0355] 来自OP/IDP服务1020a的请求将具有由OP/IDP提交给SSO的请求+状态数据作为IDCS_Request http cookie。该cookie将被保存为IDCS主机cookie，并带有加密数据等。cookie的关键字将在OP/IDP/SSO之间共享。关键字将是在租户创建操作期间初始化租户数据时创建的命名合理的关键字资源，如下所示：

[0356] 1.RP ID－由RP/SP表示的连接应用的ID；

[0357] 2.Referrer(引用者)－用户浏览器被重定向到OP/IDP时的HTTP引用者；

[0358] 3.回调URL－指回OP/IDP登录响应流程的URL。

[0359] 对于针对来自OP/IDP的请求的响应行为：

[0360] 1.SSO将使用登录应用状态来更新IDCS_REQUEST cookie,并重定向到登录应用以显示具有所有配置的登录选项的动态登录；

[0361] 2.如果联合SSO是唯一的选择,则SSO自身将重定向到IDCSSP服务,以发起与远程IDP的联合SSO；

[0362] 3.在IDCS_SESSION有效的情况下,SSO将断言从IDCS_SESSION cookie中提取的用户。

[0363] a.断言用户失败后,SSO将使用断言失败详细信息来更新IDCS_REQUEST cookie、移除IDCS_SESSION cookie、并且重定向到登录应用。登录应用将基于登录应用状态来显示带有自定义错误的登录页面。

[0364] b.断言成功后,(如果不存在)SSO将使用通道注销详细信息来更新IDCS_SESSION、将IDCS请求Cookie的内容设置为IDCS_REQ query(查询)/POST参数、移除IDCS请求cookie、并且重定向到存储在IDCS_Request cookie中的IDCS通道/callbackUrl。

[0365] 对于来自SP服务的请求,SP服务：

[0366] 1.从IDCS请求Cookie中读取和移除SAML-SP状态；

[0367] 2.处理SAML断言并将其映射到本地用户；

[0368] 3.将联合SSO操作的结果(状态,userID...)设置为OCIS_REQ_SP query/POST参数；

[0369] 4.使用IDCS_REQ query/POST参数重定向到SSO服务。

[0370] 对于针对来自SP的请求的响应行为,在有效IDCS_REQUEST具有SP状态的情况下,SSO将检查SP状态：

[0371] 1.如果SP状态的现状为“失败”,则SSO将使用断言失败详细信息来更新IDCS_REQUEST cookie,并重定向至登录应用。登录应用将基于登录应用状态来显示带有自定义错误的登录页面。

[0372] 2.如果SP状态的现状为“成功”,则SSO将创建IDCS_SESSION,其中认证上下文检查它是否是IDP发起的联合,并且post登录未启用,然后：

[0373] a.idcsSSOInitiated:包含该流程是否由SSO服务启动(布尔型)

[0374] b.如果为假(false),则SSO服务将需要检查ssoResponseFedSSOReturnURL

[0375] i.如果ssoResponseFedSSOReturnURL为空,则SSO服务将移除IDCS请求Cookie并重定向到MyConsole

[0376] ii.否则,SSO服务将移除IDCS请求Cookie并重定向到ssoResponseFedSSOReturnURL

[0377] c.如果为真(true),SSO服务将重定向到callbackURL,因为联合SSO流程由IDCS SSO服务启动。

[0378] 3.如果启用post登录,则SSO将把用户重定向到登录应用,其中登录应用状态使用指示它是post登录流程的标志来更新。

[0379] 对于来自登录应用的提交认证凭证的请求，登录应用收集凭证并将凭证发布到SSO服务REST端点。凭证作为JSON有效负载进行发送。

[0380] 对于针对来自登录应用的提交认证凭证的请求的响应行为，SSO将对发布的认证凭证进行认证，并且进行以下检查：

[0381] 1.如果认证状态为“成功”，则SSO将使用认证来创建IDCS_SESSION，并将用户重定向到IDCS_REQUEST中的/callBackUrl，或者基于特定于租户的配置向登录应用发送Ajax响应。

[0382] 2.如果认证状态为“失败”并且原因是“密码过期”，则SSO将发送Ajax响应，其中JSON有效负载具有密码管理状态以允许登录应用启动密码重置操作。状态包含用户身份信息和指示计划的操作是“对于过期密码的密码重置”的标志。

[0383] 3.对于所有其他认证状态失败的原因，SSO将发送Ajax响应，其中JSON有效负载具有登录应用状态。登录应用将使用登录应用状态来显示自定义错误消息。

[0384] 对于成功完成密码管理操作后来自登录应用的请求，登录应用收集新密码并且使用配置的特定于租户的策略规则验证来该密码。密码重置成功完成之后，登录应用会检查IDCS_REQUEST cookie的可用性。如果IDCS_REQUEST cookie可用，则它将Ajax POST请求发送到SSO服务REST端点，其中JSON有效负载具有密码管理状态。如果IDCS_REQUEST cookie不存在，则登录应用将重定向到IDCS管理控制台URL。

[0385] 对于针对成功完成密码管理操作后来自登录应用的请求的响应行为，SSO将在密码管理状态下断言用户信息：

[0386] 1.如果断言导致失败，则SSO将向登录应用状态发送Ajax响应，其中失败详细信息作为JSON有效负载。

[0387] 2.如果断言导致成功，则SSO将创建IDCS_SESSION，如果不post登录，则将用户重定向到IDCS_REQUEST中的/callBackUrl，或者将其中登录应用状态为JSON有效负载的Ajax响应发送到登录应用以显示post登录页面。

[0388] 对于成功完成post登录操作之后来自登录应用的请求，登录应用显示post登录页面并处理post登录相关操作。post登录操作成功完成后，登录应用发送Ajax POST请求，其中带有指示成功的post登录操作的标志的登录应用状态作为JSON有效负载。如果IDCS_REQUEST cookie不存在，则登录应用将重定向到IDCS管理控制台URL。

[0389] 对于针对成功完成Post登录操作之后来自登录应用的请求的响应行为，SSO将检查登录应用在登录应用状态下是否设置了Post登录标志。如果已设置，则：

[0390] 1.idcsSSOInitiated:包含该流程是否由SSO服务启动(布尔型)

[0391] A.如果为假(false)，则SSO服务将需要检查ssoResponseFedSSOReturnURL

[0392] i.如果ssoResponseFedSSOReturnURL为空，则SSO服务将移除IDCS请求Cookie并重定向到MyConsole

[0393] ii.否则，SSO服务将移除IDCS请求Cookie并重定向到ssoResponseFedSSOReturnURL

[0394] B.如果为真(true)，SSO服务将重定向到callbackURL，因为联合SSO流程由IDCS SSO服务启动。

[0395] 如所公开的，实施例提供SSO服务，该服务规定如何认证用户以及如何执行登录仪

式。登录可以基于不同类型的认证因素配置(例如,一个因素、两个因素、生物特征等)。该实施例是模块化的(不是单片代码),并且包括控制器层,并且每个特定认证都被实现为插件(例如,可以用于向LDAP认证的用户ID密码的插件模块、用于向证书存储库认证的X.509认证的插件模块、基于SMS的移动因素的移动认证的插件、基于QR码的注册的插件等)。在一个实施例中,SSO系统包括服务提供商接口("SPI")层,认证模块可以插入其中。该实施例编排了UI提交要验证的凭证的流程。该实施例执行SSO功能,而不管协议层是OpenID connect、SAML还是OAuth等。该实施例提供SSO服务作为登录和注销流程的整个IDCS访问的中央控制器,并使用丰富的、优化的和归一化的状态来使服务对所有协议不可知。一个实施例提供向不同的身份存储库(例如,远程IDP、本地身份存储库等)断言用户的功能。

[0396] 单点注销("SLO")

[0397] 一个实施例提供了全局注销过程,该过程对于用于保护每个应用的协议是不可知的。在一个实施例中,注销过程可以触发从每个协议端点注销,并且还触发从每个协议正在保护的实际应用注销。

[0398] 例如,在一个实施例中,当存在若干应用受OAuth保护时,OAuth服务到达SSO服务以获得整个全局会话和全局状态。全局状态保持为映射,并包括来自每个协议的归一化状态。对于每个协议,全局状态保存要返回的端点和协议保护的已访问URL。SSO注销不仅可以触发协议注销,还可以触发从每个应用注销、注销图像或注销端点。

[0399] 在一个实施例中,在每次注销之后,控制返回到SSO服务(即,请求流程返回到SSO服务),使得该实施例可以提供新颖的中央注销控制器。例如,注销可以从应用端点开始,然后转到相应的协议。然后,协议进入SSO服务,并且SSO服务触发从协议注销以及从受协议保护的所有受访应用注销。一旦完成上述步骤,控制就再次回到SSO服务。

[0400] 然后,SSO服务确定在全局状态中是否还有SSO服务需要用以触发注销以便完成全局注销的其他任何东西。例如,如果完成了对于OAuth的注销,则SSO服务将确定是否对于另一个协议尚未完成注销。下一个协议可能是例如SAML。所有这些信息都处于归一化全局状态。

[0401] 如果需要注销的下一个协议是SAML,则SSO服务以类似的方式触发SAML注销(例如,协议注销、实际被访问的状态协议或应用的注销等)。完成SAML注销时,控制再次回到SSO服务,并且SSO服务确定要触发的下一个协议注销。因而,SSO服务持续触发对于所有协议和所有被访问应用的注销(如在全局状态中标识的)以完成全局注销。

[0402] 一个实施例在全局状态下安全地存储归一化注销端点信息。一个实施例不存储所有注销重定向URL,而替代地存储指向具有访问权限的每个客户端的指针。一般而言,OAuth和SAML都包括为个体应用存储注销重定向URL的规定。当OAuth应用访问OAuth服务时,对应的客户端ID被跟踪并保存在SSO会话中相应的特定部件中。类似地,当SAML应用访问SAML服务时,相应的SP ID被跟踪并保存在SSO会话中相应的特定部件中。当全局注销URL(例如,通过访问SSO全局注销URL、应用注销URL中的一个等)被触发时,控制总是会回到SSO服务。然后,SSO服务移除全局会话、更新会话cookie,同时将到期时间设置为当前时间以便它不再被使用、并且使得每个部件注销。

[0403] 一个实施例确保控制在每次注销后返回到SSO服务。为此,当触发注销时,该实施例确定已经作为会话的一部分被访问或跟踪的应用列表、将注销URL构建为嵌入在HTML页

面中的图像源、并将最后一个图像源配置为SSO的重定向URL。一旦这个虚拟(dummy)HTML页面作为响应被发送给用户浏览器并且HTML页面被加载,图像源就被异步加载。因为每个图像源都是注销URL,所以会触发注销,并且响应包括设置的cookie调用,这些调用取消设置每个个体应用的cookie。由于最后一个图像的一部分是SSO的重定向URL,因此本实施例异步调用SSO服务,并且请求流程返回到SSO服务。

[0404] 在一个实施例中,一旦SSO服务发送该HTML页面以及已经对其触发注销的部件列表,该SSO服务就移除这些部件。因此,下一次控制返回到SSO服务时,它知道已经注销了什么以及还有什么需要注销,并且该过程继续直到列表为空为止。

[0405] 因此,一个实施例通过将不同应用的不同注销端点组织成图像、将其嵌入到HTML页面中、将协议重定向到该页面以便加载所有图像并遵循注销URL、并且还嵌入图像以返回到SSO服务,以触发全局注销。不同应用的不同注销端点可以基于不同的协议,例如SAML、OAuth等。

[0406] 例如,在一个实施例中,当最终用户具有受OAuth保护的若干应用时,SSO服务建立全局会话。在访问这些应用的同时,用户还可能具有受到基于同一全局会话的SAML令牌的保护的若干其他应用。一旦用户从这些应用(SAML或OAuth)中的任何一个注销并希望全局注销,则无论其协议如何,用户都将从所有应用注销。

[0407] 在一个实施例中,对于SAML,可能存在远程IDP认证。IDP是可能希望调用其自己注销的外部一方。在这种情况下,该实施例实现了在请求cookie中指示SAML调用的注销的信息。因而,该实施例提供了能够调用远程认证注销的功能。

[0408] 在一个实施例中,SSO cookie是仅针对IDCS主机的安全主机cookie,并且是事实的来源。除了IDCS部件,没有其他实体接收主机cookie。相反,相应的令牌被生成并被提供给IDCS外部的应用。

[0409] 登录/注销流程

[0410] 图11是在一个实施例中由IDCS提供的SSO功能的消息序列流程1100。当用户使用浏览器1102访问客户端1106(例如,基于浏览器的应用或移动/本机应用)时,云门1104充当应用强制实施点并执行在本地策略文本文件中定义的策略。如果云门1104检测到用户没有本地应用会话,则需要对用户进行认证。为了这样做,云门1104将浏览器1102重定向到OAuth2微服务1110,以发起针对OAuth2微服务1110的OpenID Connect登录流程(范围=“openid简档”的三参与方AZ授予流程)。

[0411] 浏览器1102的请求遍历IDCS路由层web服务1108和云门1104并到达OAuth2微服务1110。OAuth2微服务1110构造应用上下文(即,描述应用的元数据,例如,连接应用的身份、客户端ID、配置,应用可以做什么等),并将浏览器1102重定向到SSO微服务1112以便登录。

[0412] 如果用户具有有效的SSO会话,则SSO微服务1112验证现有会话而不开始登录仪式。如果用户不具有有效的SSO会话(即,不存在会话cookie),则SSO微服务1112根据客户的登录偏好(例如,显示有品牌的登录页面)来发起用户登录仪式。为了这样做,SSO微服务1112将浏览器1102重定向到以JavaScript实现的登录应用服务1114。登录应用服务1114在浏览器1102中提供登录页面。浏览器1102将REST POST发送到包括登录凭证的SSO微服务1112。SSO微服务1112生成访问令牌并将其发送到REST POST中的云门1104。云门1104将认证信息发送到管理SCIM微服务1116,以验证用户的密码。管理SCIM微服务1116确定成功的

认证,并向SSO微服务1112发送对应的消息。

[0413] 在一个实施例中,在登录仪式期间,登录页面不显示同意页面,因为“登录”操作不需要进一步的同意。相反,在登录页面上声明隐私政策,以通知用户关于向应用暴露的某些简档属性。在登录仪式期间,SSO微服务1112尊重客户的IDP偏好,并且如果被配置,则重定向到IDP,以针对经配置的IDP进行认证。

[0414] 在成功认证或验证后,SSO微服务1112利用包含用户的认证令牌的新创建/更新的SSO主机HTTP cookie(例如,在由“HOSTURL”指示的主机的上下文中创建的cookie)将浏览器1102重定向回到OAuth2微服务1110。OAuth2微服务1110将AZ代码(例如,OAuth概念)返回给浏览器1102并重定向到云门1104。浏览器1102将AZ代码发送到云门1104,并且云门1104将REST POST发送到OAuth2微服务1110,以请求访问令牌和身份令牌。这两个令牌都被限定到OAuth微服务1110(由受众令牌声称来指示)。云门1104从OAuth2微服务1110接收令牌。

[0415] 云门1104使用身份令牌将用户的经认证的身份映射到其内部账户表示,并且它可以将这个映射保存在其自己的HTTP cookie中。然后,云门1104将浏览器1102重定向到客户端1106。然后浏览器1102到达客户端1106,并从客户端1106接收对应的响应。从这个时候开始,浏览器1102可以无缝地访问应用(即,客户端1106),只要该应用的本地cookie是有效的就可以。一旦本地cookie失效,认证处理就重复。

[0416] 云门1104还使用在请求中接收的访问令牌来从OAuth2微服务1110或SCIM微服务获得“userinfo”。访问令牌足以访问用于“profile(简档)”作用域允许的属性的“userinfo”资源。经由SCIM微服务访问“/me”资源也是足够的。在一个实施例中,默认情况下,接收到的访问令牌仅适用于在“profile”范围下所允许的用户简档属性。访问其它简档属性是基于由云门1104发布的AZ授予登录请求中提交的附加(可选)范围来授权的。

[0417] 当用户访问另一个OAuth2集成的连接应用时,重复相同的处理。

[0418] 在一个实施例中,SSO集成架构使用相似的OpenID Connect用户认证流程来进行基于浏览器的用户注销。在一个实施例中,具有现有应用会话的用户访问云门1104,以发起注销。可替代地地,用户可能已经在IDCS侧发起了注销。云门1104终止特定于该应用的用户会话,并且发起针对OAuth2微服务1110的OAuth2OpenID提供者(“OP”)注销请求。OAuth2微服务1110重定向到杀死用户的主机SSO cookie的SSO微服务1112。SSO微服务1112针对如在用户的SSO cookie中被跟踪的已知注销端点发起重定向的集合(OAuth2OP和SAML IDP)。

[0419] 在一个实施例中,如果云门1104使用SAML协议来请求用户认证(例如,登录),则在SAML微服务和SSO微服务1112之间开始相似的处理。

[0420] 云高速缓存

[0421] 一个实施例提供被称为云高速缓存的服务/能力。在IDCS中提供云高速缓存,以支持与基于LDAP的应用(例如,电子邮件服务器、日历服务器、一些业务应用等)的通信,因为IDCS不根据LDAP进行通信,而此类应用被配置为仅基于LDAP进行通信。通常,云目录经由REST API暴露,并且不根据LDAP协议进行通信。一般而言,管理跨公司防火墙的LDAP连接需要特殊的配置,这些配置难以建立和管理。

[0422] 为了支持基于LDAP的应用,云高速缓存将LDAP通信转化为适合与云系统进行通信的协议。一般而言,基于LDAP的应用经由LDAP使用数据库。应用可以可替代地被配置为经由诸如SQL之类的不同协议来使用数据库。但是,LDAP在树结构中提供资源的分层表示,而SQL

将数据表示为表和字段。因而,LDAP可能更适合搜索功能,而SQL可能更适合于事务功能。

[0423] 在一个实施例中,由IDCS提供的服务可以在基于LDAP的应用中使用,以例如认证应用的用户(即,身份服务)或者为该应用强制实施安全策略(即,安全服务)。在一个实施例中,与IDCS的接口是通过防火墙并基于HTTP(例如,REST)的。通常,公司防火墙不允许访问内部LDAP通信,即使通信实现了安全套接字层("SSL"),并且不允许通过防火墙暴露TCP端口。但是,云高速缓存在LDAP和HTTP之间进行转化,以允许基于LDAP的应用到达由IDCS提供的服务,并且防火墙将对HTTP开放。

[0424] 一般而言,LDAP目录可以在一系列业务(诸如营销和开发)中使用,并且定义用户、组、工作等。在一个示例中,营销和开发业务可以具有不同的有针对性的客户,并且对于每个客户,可以有其自己的应用、用户、组、工作等。可以运行LDAP高速缓存目录的一系列业务的另一个示例是无线服务提供者。在这种情况下,由无线服务提供者的用户进行的每个呼叫都针对LDAP目录来认证用户的设备,并且LDAP目录中的对应信息中的一些可以与计费系统同步。在这些示例中,LDAP提供了在运行时在物理上隔离正在被搜索的内容的功能。

[0425] 在一个示例中,无线服务提供者可以在使用由IDCS提供的服务以支持短期营销活动的同时处理用于其核心业务(例如,常规呼叫)的自己的身份管理服务。在这种情况下,当云高速缓存具有它针对云运行的单个用户集合和单个组集合时,云高速缓存将"扁平化"LDAP。在一个实施例中,可以在IDCS中实现任何数量的云高速缓存。

[0426] 分布式数据网格

[0427] 在一个实施例中,IDCS中的高速缓存集群是基于分布式数据网格来实现的,如在例如编号为2016/0092540的美国专利公开中所公开的,其公开内容通过引用并入本文。分布式数据网格是一种系统,其中计算机服务器的集合在一个或多个集群中一起工作,以管理分布式或集群环境中的信息和相关操作(诸如计算)。分布式数据网格可以被用于管理跨服务器共享的应用对象和数据。分布式数据网格提供低响应时间、高吞吐量、可预测的可伸缩性、持续可用性和信息可靠性。在特定的示例中,分布式数据网格(诸如来自Oracle公司的Oracle Coherence数据网格)存储存储器中的信息,以实现更高的性能,并且在保持那种信息的副本跨多个服务器同步时采用冗余,从而在服务器发生故障的情况下确保系统的弹性和数据的持续可用性。

[0428] 在一个实施例中,IDCS实现诸如Coherence之类的分布式数据网格,使得每个微服务可以请求访问共享的高速缓存对象而不被阻塞。Coherence是专有的基于Java的存储器内数据网格,其被设计为具有比传统的关系型数据库管理系统更好的可靠性、可伸缩性和性能。Coherence提供了对等(Peer to Peer)(即,没有中央管理器)、存储器内的分布式高速缓存。

[0429] 图12图示了分布式数据网格1200的示例,其存储数据并向客户端1250提供数据访问并实现本发明的实施例。"数据网格集群"或"分布式数据网格"是包括多个计算机服务器(例如,1220a、1220b、1220c和1220d)的系统,这些多个计算机服务器在一个或多个集群(例如,1200a、1200b、1200c)中一起工作,以在分布式或集群环境中存储和管理信息和相关操作(诸如计算)。虽然分布式数据网格1200被示为在集群1200a中包括四个服务器1220a、1220b、1220c、1220d,具有五个数据节点1230a、1230b、1230c、1230d和1230e,但是分布式数据网格1200可以包括任意数量的集群以及每个集群中任意数量的服务器和/或节点。在实

施例中，分布式数据网格1200实现本发明。

[0430] 如图12中所示，分布式数据网格通过在一起工作的多个服务器(例如，1220a、1220b、1220c和1220d)上分布数据来提供数据存储和管理能力。数据网格集群的每个服务器可以是常规的计算机系统，诸如像具有一到两个处理器插槽和每个处理器插槽两到四个CPU核心的“商品x86”服务器硬件平台。每个服务器(例如，1220a、1220b、1220c和1220d)被配置有一个或多个CPU、网络接口卡(“NIC”)和存储器，其中存储器包括例如至少4GB的RAM，高达64GB RAM或更多。服务器1220a被示为具有CPU 1222a、存储器1224a和NIC 1226a(这些元件在其它服务器1220b、1220c、1220d中也存在，但未示出)。可选地，每个服务器也可以提供有闪存(例如，SSD 1228a)，以提供外溢(spillover)存储容量。在被提供时，SSD容量优选地是RAM的尺寸的十倍。数据网格集群1200a中的服务器(例如，1220a、1220b、1220c、1220d)使用高带宽NIC(例如，PCI-X或PCIe)连接到高性能网络交换机1220(例如，千兆以太网或更好)。

[0431] 集群1200a优选地包含最少四个物理服务器，以避免在故障期间丢失数据的可能性，但是典型的安装具有多得多的服务器。每个集群中存在的服务器数量越多，故障转移和故障回复的效率越高，并且服务器故障对集群的影响更小。为了最小化服务器之间的通信时间，每个数据网格集群理想地限于提供服务器之间的单跳通信的单个交换机1202。因此，集群可以受到交换机1202上的端口数量的限制。因此，典型的集群将包括4至96个物理服务器。

[0432] 在分布式数据网格1200的大多数广域网(“WAN”)配置中，WAN中的每个数据中心具有独立但互连的数据网格集群(例如，1200a、1200b和1200c)。WAN可以例如包括比图12中所示的多得多的集群。此外，通过使用互连但独立的集群(例如，1200a、1200b、1200c)和/或在彼此远离的数据中心中定位互连但独立的集群，分布式数据网格可以保护到客户端1250的数据和服务，防止由于自然灾害、火灾、洪水、延长的掉电等造成的一个集群中的所有服务器的同时丢失。

[0433] 一个或多个节点(例如，1230a、1230b、1230c、1230d和1230e)在集群1200a的每个服务器(例如，1220a、1220b、1220c、1220d)上操作。在分布式数据网格中，节点可以是例如软件应用、虚拟机等，并且服务器可以包括节点在其上操作的操作系统、管理程序等(未示出)。在Oracle Coherence数据网格中，每个节点都是Java虚拟机(“JVM”)。取决于服务器上可用的CPU处理能力和存储器，可以在每个服务器上提供多个JVM/节点。可以根据分布式数据网格的需要添加、开始、停止和删除JVM/节点。运行Oracle Coherence的JVM在被开始时自动加入集群。加入集群的JVM/节点被称为集群成员或集群节点。

[0434] 每个客户端或服务器包括用于传送信息的总线或其它通信机制，以及耦合到总线以用于处理信息的处理器。处理器可以是任何类型的通用或专用处理器。每个客户端或服务器还可以包括用于存储要由处理器执行的信息和指令的存储器。存储器可以由随机存取存储器(“RAM”)、只读存储器(“ROM”)、诸如磁盘或光盘的静态存储器或任何其它类型的计算机可读介质的任意组合组成。每个客户端或服务器还可以包括通信设备，诸如网络接口卡，以提供对网络的访问。因此，用户可以直接与每个客户端或服务器进行对接，或者通过网络或其它任何方式进行远程对接。

[0435] 计算机可读介质可以是可由处理器访问的任何可用介质，并且包括易失性和非易

失性介质、可移动和不可移动介质以及通信介质。通信介质可以包括计算机可读指令、数据结构、程序模块或者经调制的数据信号(诸如载波或其它传输机制)中的其它数据,并且包括任何信息传递介质。

[0436] 处理器还可以经由总线耦合到显示器,诸如液晶显示器(“LCD”)。键盘和光标控制设备(诸如计算机鼠标)也可以耦合到总线,以使得用户能够与每个客户端或服务器进行交互。

[0437] 在一个实施例中,存储器存储在由处理器执行时提供功能的软件模块。这些模块包括为每个客户端或服务器提供操作系统功能的操作系统。这些模块还可以包括用于提供云身份管理功能以及本文公开的所有其它功能的云身份管理模块。

[0438] 客户端可以访问web服务,诸如云服务。在一个实施例中,web服务可以在来自Oracle公司的webLogic服务器上实现。在其它实施例中,可以使用web服务的其它实现。web服务访问存储云数据的数据库。

[0439] 如所公开的,实施例实现了基于微服务的架构,以提供基于云的多租户IAM服务。在一个实施例中,每个请求的身份管理服务都被分成实时任务和近实时任务,实时任务由中间层中的微服务处理,而近实时任务被卸载到消息队列。因而,实施例提供了云规模的IAM平台。

[0440] 图13是根据实施例的基于云的IAM功能的流程图1300。在一个实施例中,图13的流程图的功能由存储在存储器或其他计算机可读或有形介质中的软件来实现,并由处理器执行。在其他实施例中,功能可以(例如,通过使用专用集成电路(“ASIC”)、可编程门阵列(“PGA”)、现场可编程门阵列(“FPGA”)等)由硬件来执行,或者由硬件和软件的任意组合来执行。

[0441] 在1302处,接收对被配置为允许访问应用的身份管理服务的第一请求。例如,在一个实施例中,第一请求由诸如本文所描述的云门(例如参考图6中的web路由层610和/或图7中的云门702)的安全门接收。在一个实施例中,第一请求包括对标识身份管理服务的API和被配置为执行身份管理服务的微服务的调用。在一个实施例中,微服务是能够与其他模块/微服务进行通信的自包含模块,并且每个微服务具有可以被其它微服务联系的未命名的通用端口。例如,在一个实施例中,各种应用/服务602可以对IDCS API进行HTTP调用以使用IDCS微服务614,如图6所示。在一个实施例中,微服务是运行时部件/进程。

[0442] 在1304处,第一请求被发送到微服务,其中微服务通过生成令牌来执行身份管理服务,其中微服务至少部分地通过向单次登录微服务发送第二请求来生成令牌,并且其中单次登录微服务被配置为跨基于不同协议的不同微服务来提供单次登录功能。

[0443] 在1306处,从微服务接收令牌,并且在1308处,令牌被提供给应用,其中令牌允许访问应用。

[0444] 图14是图示了根据一个实施例的部件的各种功能和cookie传播功能的注销流程。图14中涉及的部件或层包括最终用户客户端(或依赖方(party)1(“RP1”))1401、OIDC层1402、SSO层1403和SAML层1404。图14涉及注销和清除cookie。

[0445] 在1420处,客户端1401通常通过注销URL触发注销。在1421处,OIDC 1402将客户端1401的返回URL存储在cookie(例如,IDCS_REQ cookie)中,该cookie存储在客户端1401的对应浏览器上。返回URL可以是客户端1401的最终成功注销登陆(landing)页面。

[0446] 在1429处,清除认证上下文。在1430处,从SSO Cookie的部件数据映射中清除SSO部件条目。在1431处,确定部件数据映射中是否存在OAuth条目。

[0447] 如果在1431处为“是”,则HTML 303重定向到OIDC 1402层,并且在1423处读取OAuth条目值并检索clientID。

[0448] 在1424处,提取对应于clientID的注销URL。在一个实施例中,客户端API用于查询特定clientID的注销URL。在1425处,从SSO Cookie的部件数据映射中清除OAuth部件。在1427处,使用图像标签生成HTML响应,其中src属性设置为RP的注销URL。最后一个图像标签将具有被设置为SSO服务注销URL的src属性。在1428处,当执行具有被设置为SSO服务端点的src属性的最后一个图像标签的同时,呈现HTML响应。新的HTTP GET请求将被发送到SSO服务注销端点。HTML 302重定向将功能移动到SSO层1403,其中功能在1429处继续。

[0449] 如果在1431处为“否”,则在1432处确定部件数据映射中是否存在SAML条目。如果在1432处为“是”,则在1433处确定在cookie中SAML注销标志(“logoutSAMLInvoke”)是否被设置为真。如果在1433处为“否”,则在1434处从cookie读取注销重定向URL值。如果在1432处为“否”,则也执行1434。

[0450] 在1435处,确定注销重定向URL值是否为空。如果在1435处为“是”,则在1436处从SSOsettings提取租户的注销登陆页面。如果在1435处为“否”,或者在1436之后,在1437处确定是否跳过SAML注销。

[0451] 如果在1437处为“是”,则在1438处,HTML 302重定向至注销重定向URL(如果存在的话),或者从SSOsettings重定向至租户注销登陆页面。重定向将指向客户端1401的注销登陆页面。然后,在1426处,在客户端1401呈现注销页面。如果在1437处为“否”,则在1439处移除未设置的cookie,然后功能移动到1438处。

[0452] 如果在1433处为“是”,则在1442处,向SAML 1404层的HTML 300重定向执行SAML注销并触发连接的SP的注销。在1441处,SAML条目然后从SSO cookie的部件数据映射中被移除,并且功能移动到SSO层1403至1429。

[0453] 在1442处示出了具有查询参数的从SAML-SP到SSO服务的重定向URL的示例(具有成功的SAML断言),例如可以是:http://HOST.us.oracle.com:8990/sso/v1/user/login?OCIS_REQ_SP＝kERRx9mNY1SWxvcv5ohVeLjwL0pJAI5yLBNhZRNkK88xD。SSO将插入了“OCIS_REQ”的请求参数发送到协议/通道部件。从SAML到SSO,,它是如上所示的“OCIS_REQ_SP”。当IDCS被配置为使用内部IdP对用户进行认证时,该重定向会发生。在IdP成功质询用户后,IdP使用SAML断言以将用户重定向到IDCS SAML SP。IDCS SAML SP验证SAML断言并将用户映射到IDCS用户记录。IDCS SAML SP将SAML的部件数据保存在IDCS会话cookie“ORA_OCIS”中。然后,IDCS SAML SP将用户重定向到SSO,并在OCIS_REQ_SP query/post参数中提供用户数据作为从SAML到SSO的302/POSTFORM重定向的一部分。从SSO向SAML IdP或OAuth IdP通信时使用OCIS_REQ query/form参数,而从SAML SP向SSO通信时使用OCIS_REQ_SP query/form参数。

[0454] 以下是根据一个实施例由SAML SP服务执行的步骤,其由SSO服务用来发起与联合SAML身份提供者的交互。SSO服务将确定如何对用户进行认证。一种可能性是使用联合SSO对用户进行认证:换言之,将用户认证委派给远程IdP,可以是内部IdP,也可以是客户使用的云服务IdP。

[0455] 当SSO服务将用户重定向到联合服务时,SSO模块必须能够指示:

[0456] ●哪个SAML IdP应该用于联合SSO。

[0457] 当联合服务将用户返回到SSO部件时,必须提供以下信息:

[0458] ●UserID;

[0459] ●认证时间;

[0460] ●到期时间;

[0461] ●联合消息属性;

[0462] ●IdP合作伙伴名称。

[0463] 在认证请求期间的运行时,SSO服务将:

[0464] ●通过使用Cookie管理器,将SAML服务所请求的数据存储在请求Cookie中

[0465] ○SSORequestFedSSOIdP:远程IDP的ID。

[0466] ●将用户重定向到SAML服务认证URL(http://tenant-hostname/fed/v1/user/request/login)。

[0467] 在认证请求期间的运行时,SAML服务将:

[0468] ●读取请求Cookie以提取和移除由SSO服务设置的数据

[0469] ○SSORequestFedSSOIdP:远程IDP的ID。

[0470] ●读取请求Cookie以提取而不移除下列数据

[0471] ○SSOECID:OAuth/SAML-IdP服务设置的ECID。

[0472] SAML将使用该数据来记录事件。

[0473] ●使用远程IdP来启动联合SSO;

[0474] ●验证传入的SAML断言并将其映射到SAML用户;

[0475] ●在Session Cookie中写入以存储联合数据;

[0476] ●不要触及现有的请求cookie;

[0477] ●联合SSO操作后的SAML-SP将创建描述联合SSO操作的结果的SOSExternalIdPResponset:

[0478] ○ssoResponseFedSSOStatus:指出状态的字符串

[0479] ■SUCCESS意味着成功;

[0480] ■任何其他字符串都是SAML错误代码(例如:error.samlsrv.sp.sso.bindingUnknown)。

[0481] ○ssoResponseFedSSOUserID:如果操作成功,则此字段将包含用户GUID(字符串)。

[0482] ○ssoResponseFedSSOIdP:如果操作成功,则此字段将包含IdP合作伙伴名称(字符串)。

[0483] ○ssoResponseFedSSONameID:如果操作成功,则此字段将包含SAML断言NameID(字符串)。

[0484] ○ssoResponseFedSSOReturnURL:如果操作成功,并且如果流程未由SSO服务发起,则此字段将包含用户需要被重定向的URL(字符串)。

[0485] ○ssoResponseFedSSOAssertionAttributes:如果操作成功,则此字段将包含SAML断言属性(映射)。

[0486] ○ssoResponseFedSSOTimeout:如果操作成功,则此字段将包含IdP会话超时(长整型)。

[0487] ○ssoECID:包含ECID。

[0488] ○idcsSSOInitiated:包含该流程是否由SSO服务启动(布尔型)

[0489] ■如果为假(false),则SSO服务将需要检查ssoResponseFedSSOReturnURL

[0490] ●如果ssoResponseFedSSOReturnURL为空,则SSO服务将移除IDCS请求Cookie并重定向到MyConsole;

[0491] ●否则,SSO服务将移除IDCS请求Cookie并重定向到ssoResponseFedSSOReturnURL;

[0492] ■如果为真(true),则SSO服务将重定向到callbackURL,因为联合SSO流程由IDCS SSO服务启动。

[0493] ●例如,从SAML-SP到SSO服务的重定向URL将是:"http://HOST.us.oracle.com:8990/sso/v1/user/login?OCIS_REQ_SP＝kERRx9mNY1SWxvcv5ohVeLjwL0pJAI5yLBNhZRNkK88xD..."在认证响应期间的运行时,SSO服务将:

[0494] ●SSO服务将检索查询参数OCIS_REQ_SP的值,并对其进行解码,并由SAML服务读取数据集

[0495] ○SSOResponseFedSSOStatus:SUCCESS指示成功,FAILURE将带来错误代码(例如,error.samlsrv.sp.sso.assertion.noUserReturnedViaNameID)SSOResponseFedSSOUserID:在成功的情况下的IDCS UserID;

[0496] ○SSOResponseFedSSOIdP:远程IdP的ID;

[0497] ○SSOResponseFedSSOTimeout:IdP会话超时的时间(秒)(设置为-1);

[0498] ○SSOResponseFedSSONameID:在SAML断言NameID字段中使用的标识符;

[0499] ○SSOResponseFedSSOReturnURL:SSO服务创建IDCS会话后用户应该被重定向到的URL。此参数仅在IdP启动的SSO流程期间被设置;

[0500] ○SSOResponseFedSSOAssertionAttributes:SAML SSO断言中包含的属性列表。

[0501] ●读取请求Cookie以提取并移除它:

[0502] ○SSOECID:OAuth/SAML-IdP服务设置的ECID。它将被SSO用于记录事件;

[0503] ○rpId,rpName。

[0504] ●为用户创建IDCS会话并将数据存储在会话Cookie中

[0505] ●将用户重定向到:

[0506] ●idcsSSOInitiated:包含该流程是否由SSO服务启动(布尔型)

[0507] ○如果为假(false),则SSO服务将需要检查ssoResponseFedSSOReturnURL;

[0508] ■如果ssoResponseFedSSOReturnURL为空,则SSO服务将移除IDCS请求Cookie并重定向到MyConsole;

[0509] ■否则,SSO服务将移除IDCS请求Cookie并重定向到ssoResponseFedSSOReturnURL。

[0510] ○如果为真(true),则SSO服务将重定向到callbackURL,因为联合SSO流程由IDCS SSO服务启动。

[0511] 在一个实施例中,用于执行SLO的SSO功能/步骤如下:

[0512] 1.从会话cookie中移除用户会话,即AuthNContext。

[0513] 2.对于部件数据中的OAuth条目(如果存在的话),重定向到OAuth注销端点。

[0514] 3.OAuth/OIDC从客户端id提取RP的注销URL,从部件数据中清除OAuth条目,产生具有不同RP注销URL的HTML源(异步过处理),最后返回到SSO注销(最后一个img src是SSO注销)。

[0515] 4.SSO检查部件数据中的SAML条目,如果存在,则检查IDCS_REQ cookie中的“不调用SAML标志”,如果设置为true或不存在,则通过重定向委派到SAML/logout,SAML完成其SP注销流程、清除部件数据中的SAML条目并返回SSO/logout。

[0516] 5.SAML服务启动SAML 2.0注销协议操作,并将用户重定向到各个远程联合伙伴进行注销。

[0517] 6.一旦完成,SAML服务就从IDCS会话Cookie中删除SAML数据。

[0518] 7.SAML服务将用户重定向到SSO服务(来自SSO服务的URL TBD)。

[0519] 8.SSO注销获取“返回URL”、清除会话cookie(如果存在的话,则不得移除SAML部件数据)、并且最后重定向到“返回URL”。如果“返回URL”不存在,则重定向到特定于租户的注销登陆页面。

[0520] 在一个实施例中,微服务被配置为基于协议提供认证功能。在一个实施例中,协议是OAuth或SAML,并且单点登录微服务被配置为跨OAuth和SAML提供单点登录功能。在一个实施例中,单点登录微服务生成被配置为会话cookie的全局会话,并向微服务提供全局会话。在一个实施例中,微服务将全局会话转换为令牌。在一个实施例中,单点登录微服务将第二请求的请求参数归一化。

[0521] 在一个实施例中,应用被配置为由包括该租赁在内的多个租赁使用。在一个实施例中,微服务是无状态的,并从数据库检索数据以执行身份管理服务。在一个实施例中,数据库和微服务被配置为彼此独立地伸缩。在一个实施例中,数据库包括分布式数据网格。

[0522] 如所公开的,实施例提供了多租户、云规模的IAM,其通过实现全局会话并将全局会话转换为适当的协议令牌来跨各种协议提供SSO功能。进一步地,实施例使用cookie和重定向来为所有应用执行SLO功能。

[0523] 本文具体地示出和/或描述了若干实施例。但是,应当认识到的是,在不背离本发明的精神和预期范围的情况下,所公开的实施例的修改和变化被上述教导涵盖并在所附权利要求的范围内。

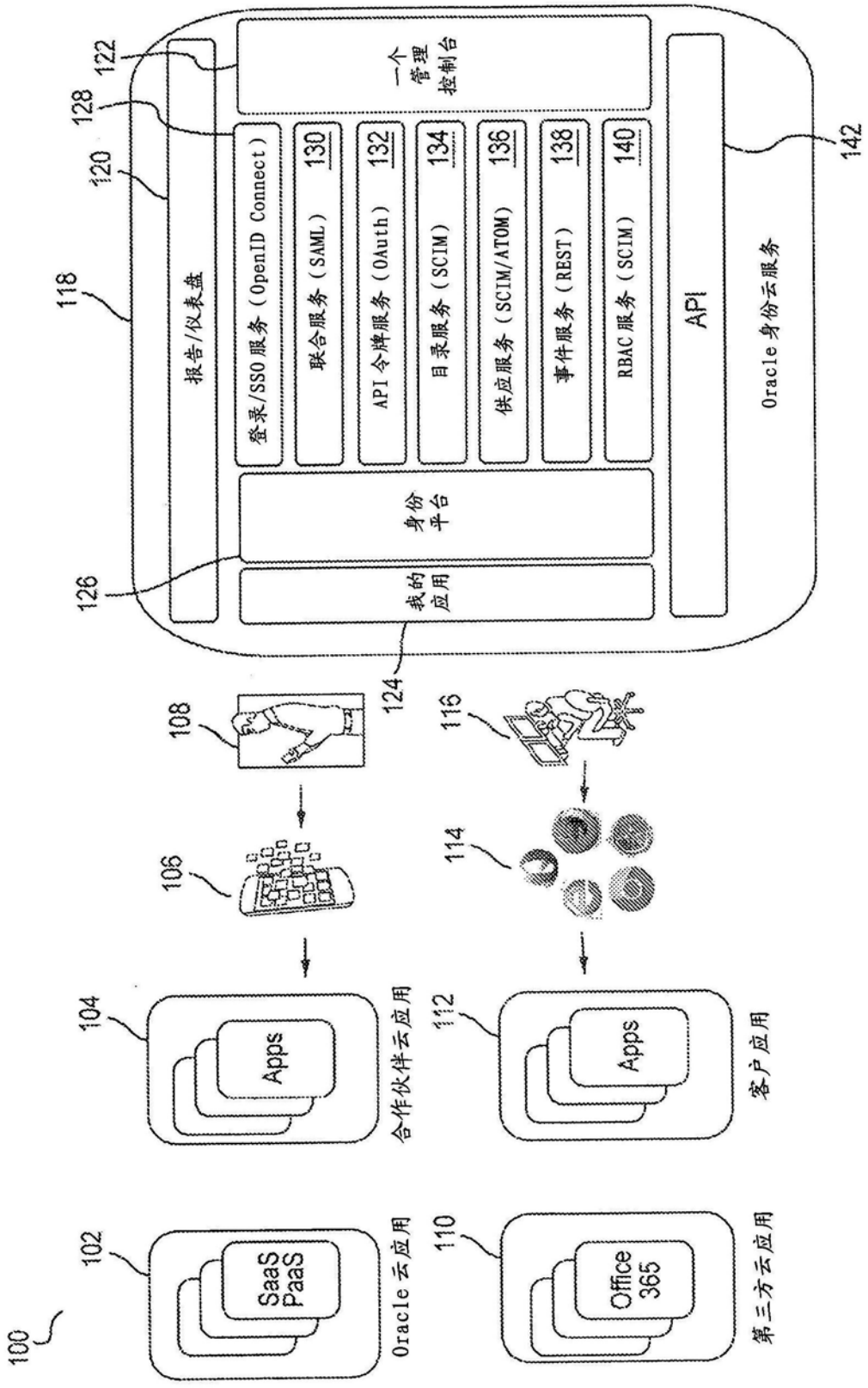

100
102
SaaS
PaaS
Oracle 云应用
104
Apps
合作伙伴云应用
106
108
110
Office
365
第三方云应用
112
Apps
客户应用
114
116
118
120
122
128
126
124
报告/仪表盘
登录/SSO 服务（OpenID Connect）
联合服务（SAML）
130
API 令牌服务（OAuth）
132
目录服务（SCIM）
134
供应服务（SCIM/ATOM）
136
事件服务（REST）
138
RBAC 服务（SCIM）
140
一个管理控制台
身份平台
我的应用
API
142
Oracle 身份云服务

图1

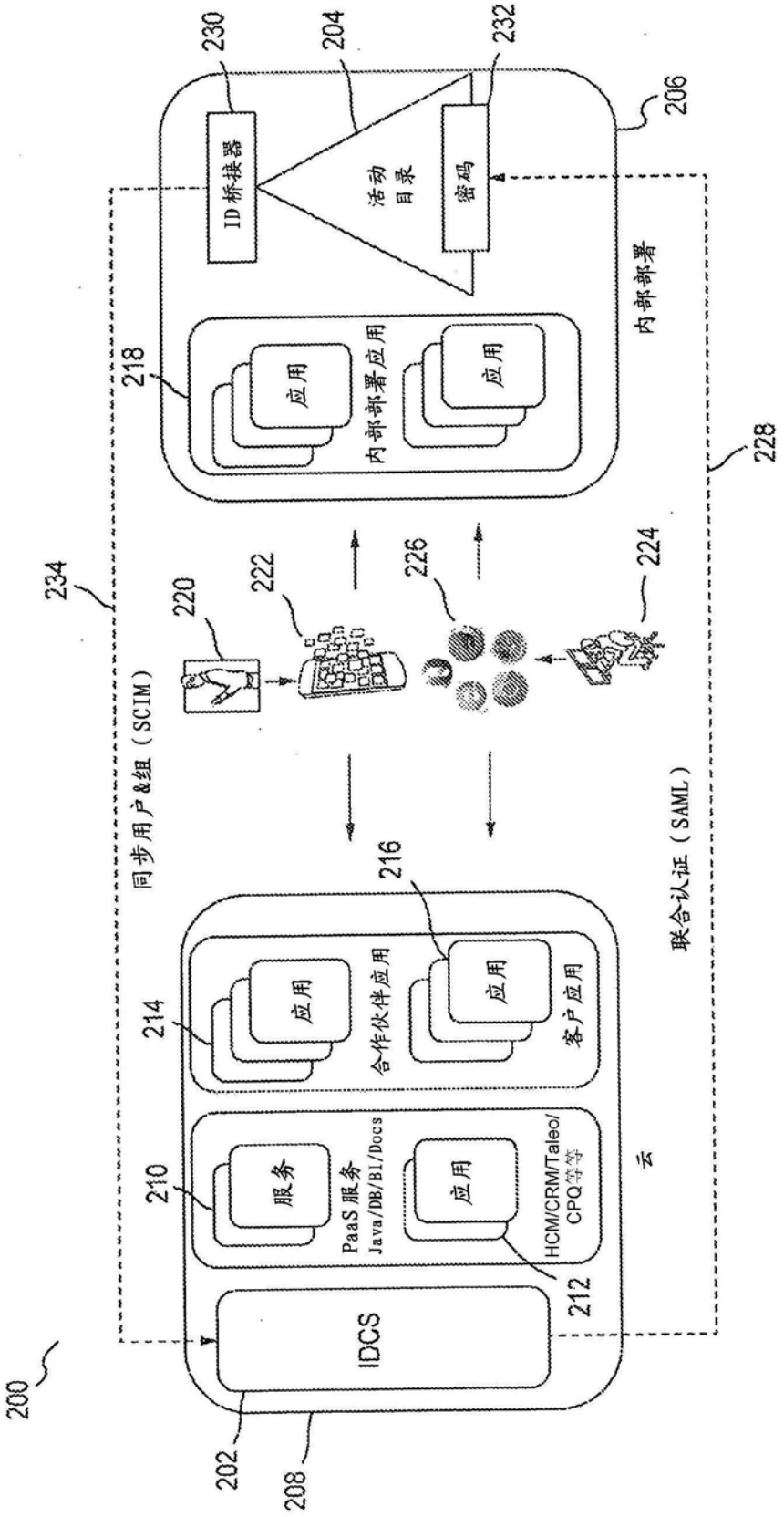

200
202
208
IDCS
212
210
服务
PaaS 服务
Java/DB/BI/Docs
应用
HCM/CRM/Taleo/
CPQ等等
214
应用
合作伙伴应用
216
应用
客户应用
云
218
应用
内部部署应用
应用
内部部署
206
230
ID 桥接器
204
活动
目录
232
密码
228
联合认证（SAML）
234
同步用户&组（SCIM）
220
222
226
224

图2

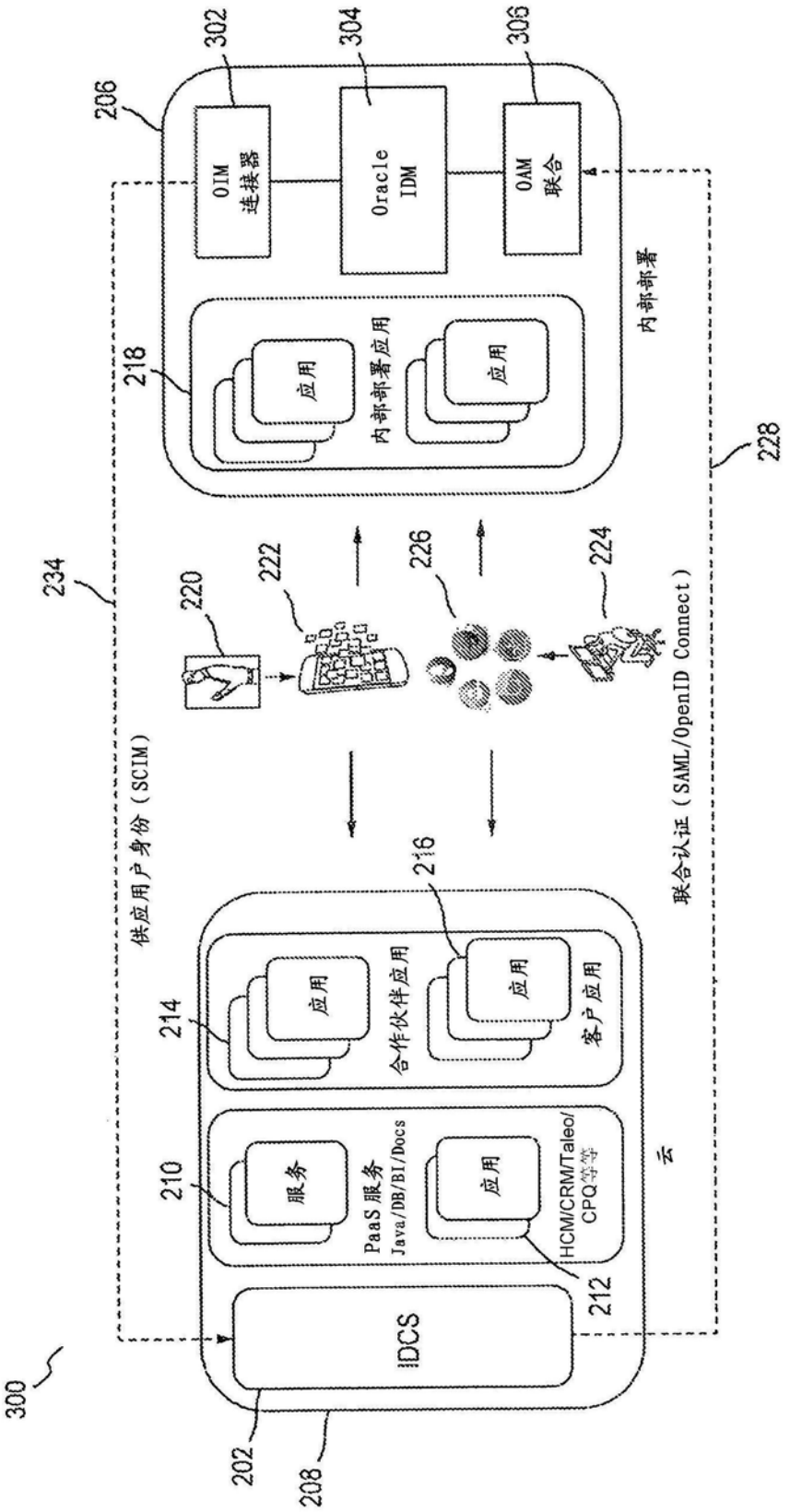

300
202
208
IDCS
212
HCM/CRM/Taleo/
CPQ等等
应用
PaaS 服务
Java/DB/BI/Docs
服务
210
应用
合作伙伴应用
客户应用
214
216
云
供应用户身份（SCIM）
联合认证（SAML/OpenID Connect）
234
220
222
226
224
228
218
应用
内部部署应用
应用
内部部署
206
OIM
连接器
302
Oracle
IDM
304
OAM
联合
306

图3

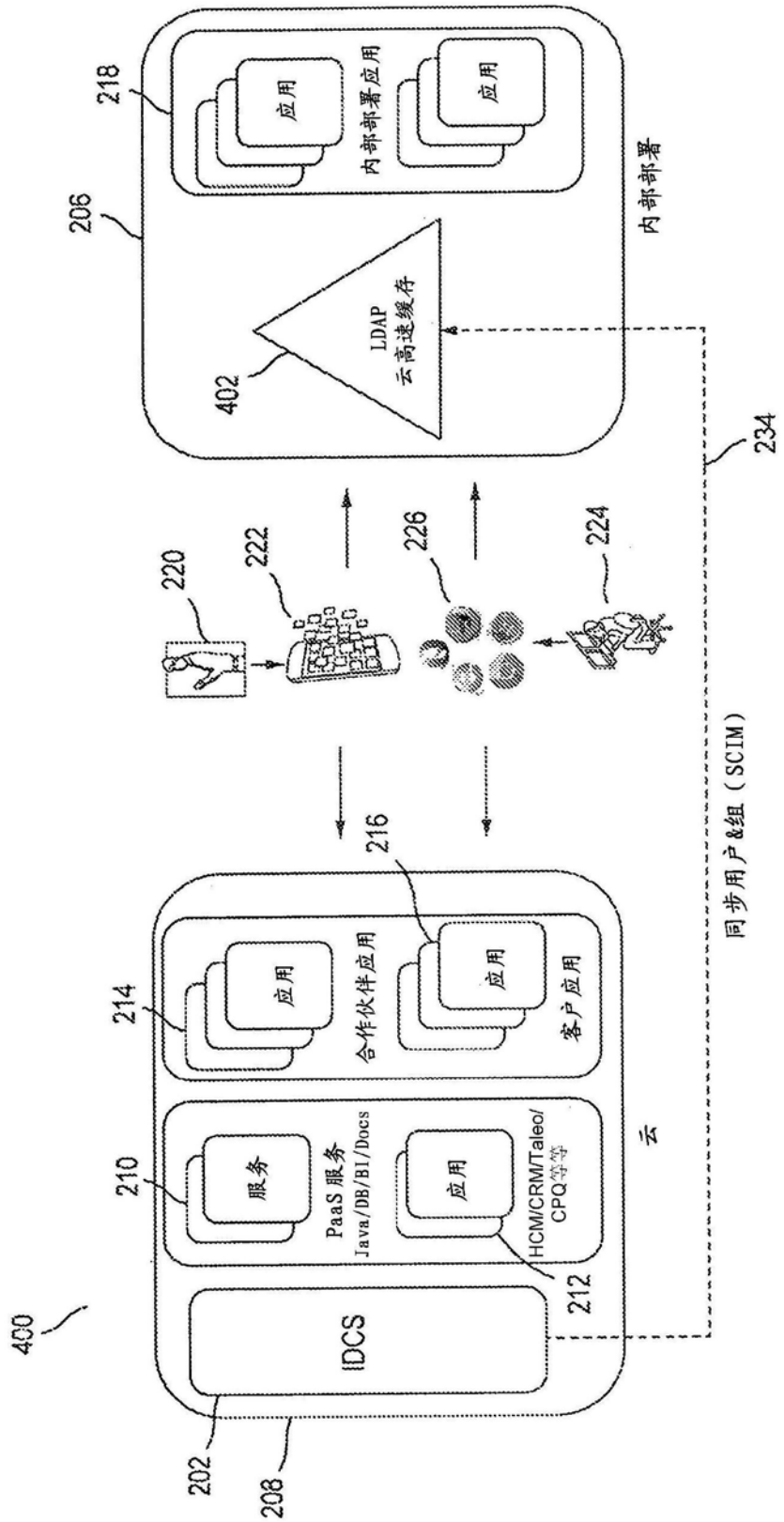

400
IDCS
202
208
212
云
PaaS 服务
Java/DB/BI/Docs
服务
210
应用
HCM/CRM/Taleo/
CPQ等等
214
合作伙伴应用
应用
216
客户应用
应用
218
内部部署应用
应用
应用
内部部署
206
402
LDAP
云高速缓存
234
同步用户&组（SCIM）
220
222
226
224

图4

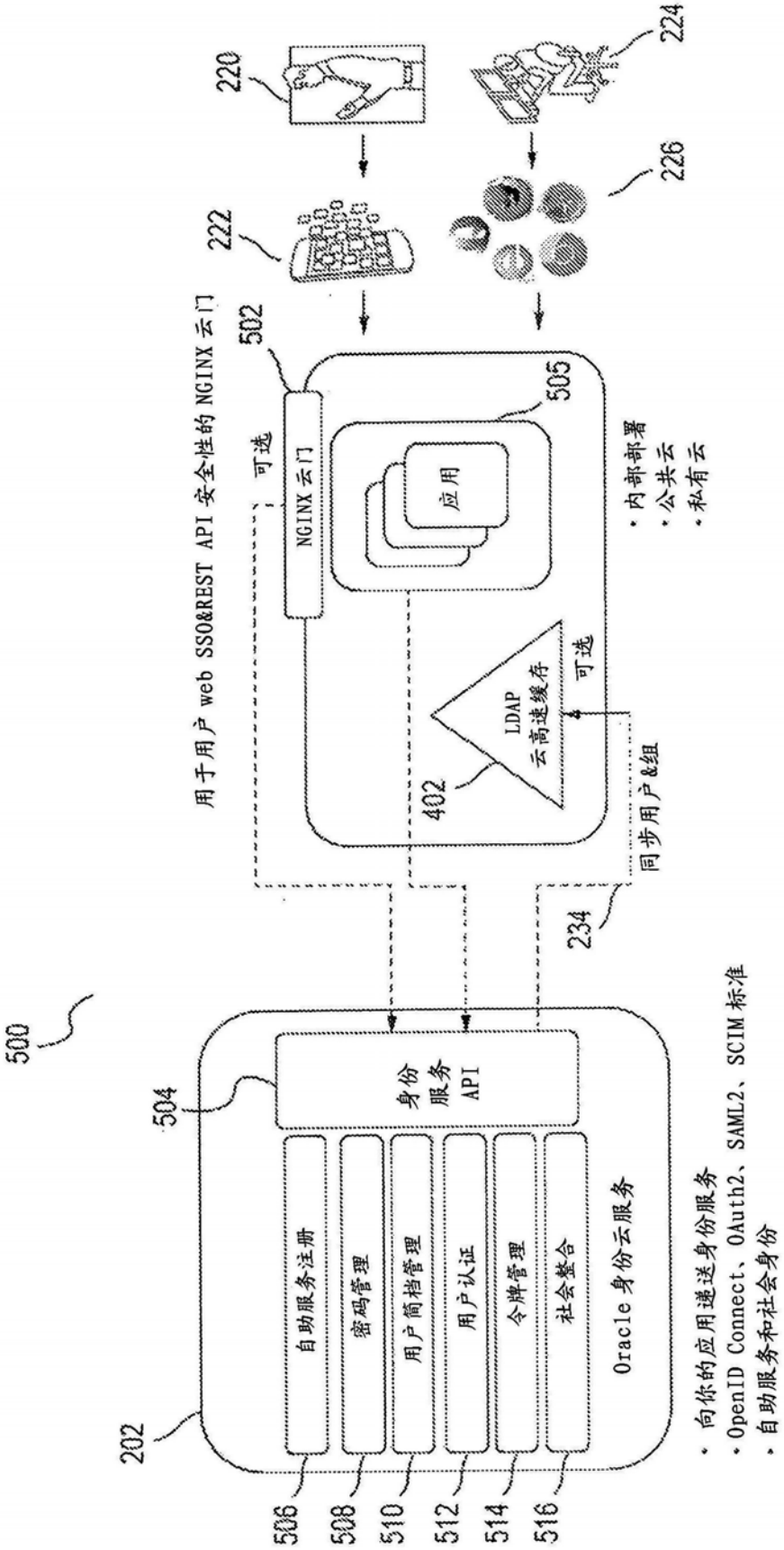

500
202
504
身份服务API
自助服务注册
密码管理
用户简档管理
用户认证
令牌管理
社会整合
Oracle身份云服务
506
508
510
512
514
516
用于用户web SSO&REST API安全性的NGINX云门
502
可选
NGINX云门
505
应用
402
LDAP云高速缓存
可选
234
同步用户&组
·内部部署
·公共云
·私有云
220
222
224
226
·向你的应用递送身份服务
·OpenID Connect、OAuth2、SAML2、SCIM标准
·自助服务和社会身份

图5

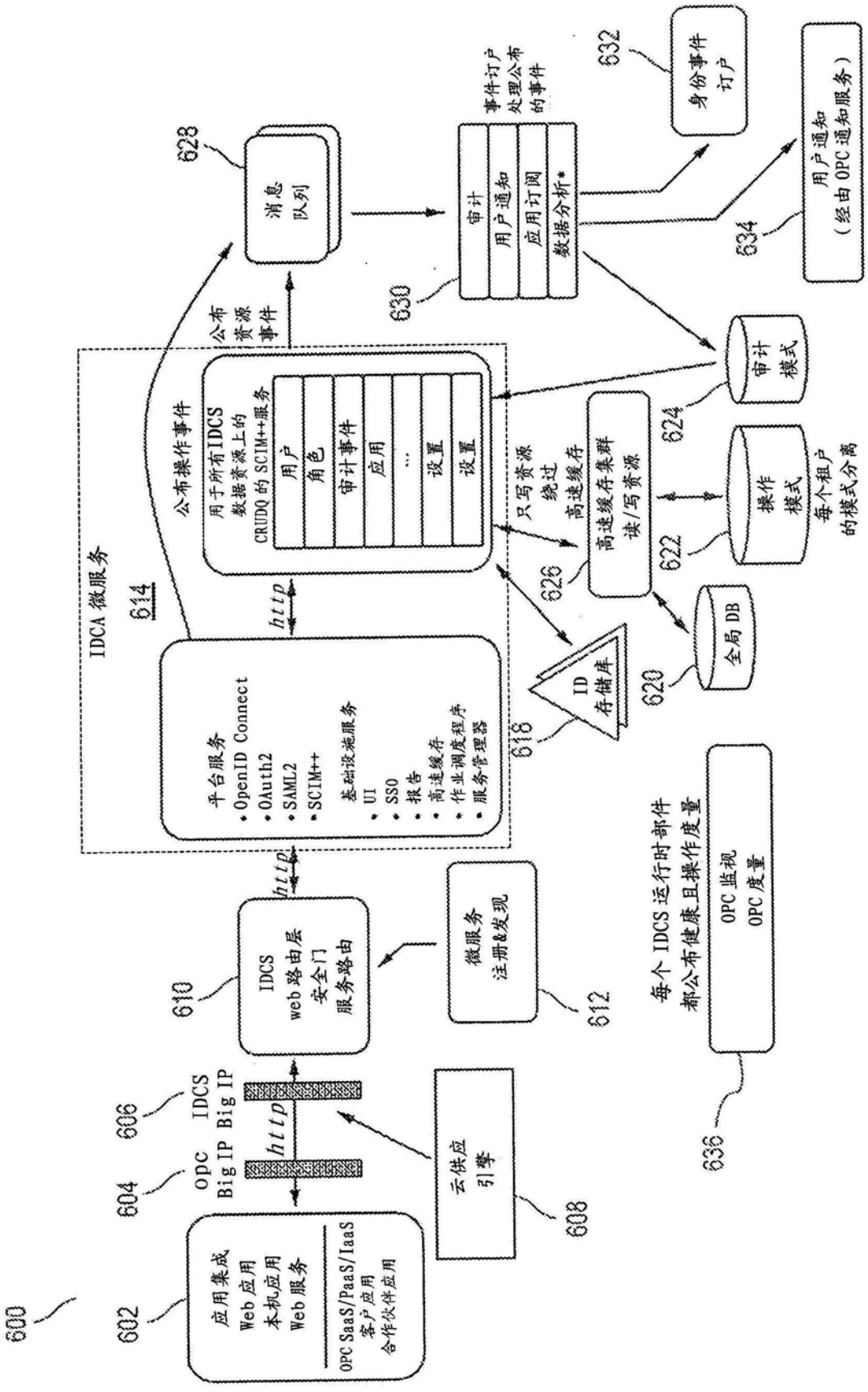

600
602
应用集成
Web 应用
本机应用
Web 服务
OPC SaaS/PaaS/IaaS
客户应用
合作伙伴应用
604
OPC Big IP
606
IDCS Big IP
http
608
云供应引擎
610
IDCS web 路由层 安全门 服务路由
612
微服务 注册&发现
614
IDCA 微服务
平台服务
• OpenID Connect
• OAuth2
• SAML2
• SCIM++
基础设施服务
• UI
• SSO
• 报告
• 高速缓存
• 作业调度程序
• 服务管理器
http
公布操作事件
用于所有IDCS 数据资源上的 CRUDQ 的 SCIM++服务
用户
角色
审计事件
应用
...
设置
设置
公布资源事件
618
ID 存储库
620
全局 DB
622
操作模式
每个租户的模式分离
624
审计模式
只写资源绕过高速缓存
626
高速缓存集群 读/写资源
623
消息队列
630
审计
用户通知
应用订阅
数据分析*
事件订户处理公布的事件
632
身份事件订户
634
用户通知 (经由 OPC 通知服务)
每个 IDCS 运行时部件都公布健康且操作度量
OPC 监视
OPC 度量
636

图6

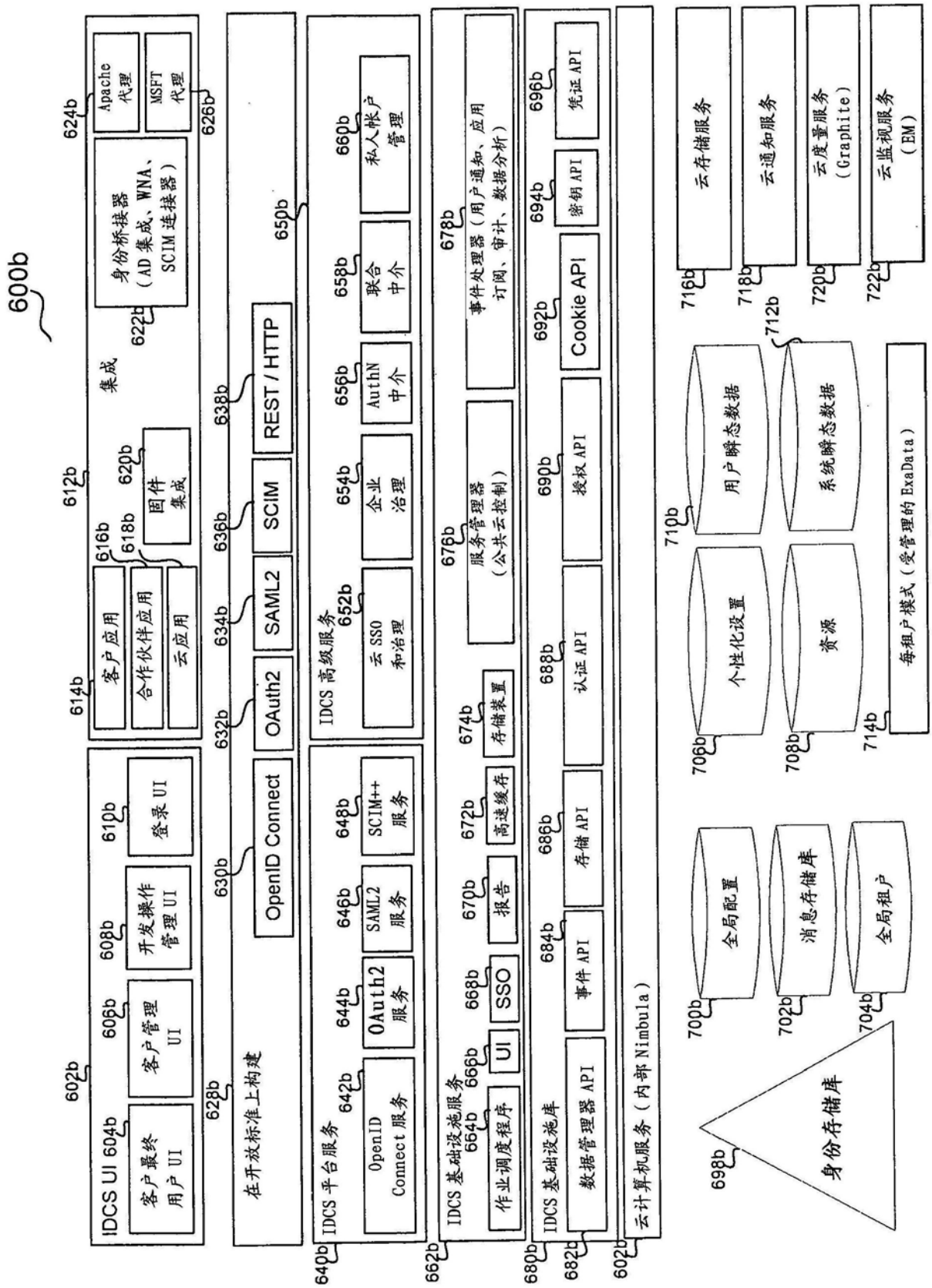

600b
602b
IDCS UI 604b
客户最终用户 UI
606b
客户管理 UI
608b
开发操作管理 UI
610b
登录 UI
612b
614b
客户应用
合作伙伴应用
云应用
616b
618b
620b
固件集成
集成
622b
身份桥接器（AD 集成、WNA、SCIM 连接器）
624b
Apache 代理
626b
MSFT 代理
628b
在开放标准上构建
630b
OpenID Connect
632b
OAuth2
634b
SAML2
636b
SCIM
638b
REST / HTTP
650b
640b
IDCS 平台服务
642b
OpenID Connect 服务
644b
OAuth2 服务
646b
SAML2 服务
648b
SCIM++ 服务
652b
IDCS 高级服务
云 SSO 和治理
654b
企业治理
656b
AuthN 中介
658b
联合中介
660b
私人帐户管理
662b
IDCS 基础设施服务
664b
作业调度程序
666b
UI
668b
SSO
670b
报告
672b
高速缓存
674b
存储装置
676b
服务管理器（公共云控制）
678b
事件处理器（用户通知、应用订阅、审计、数据分析）
680b
IDCS 基础设施库
682b
数据管理器 API
684b
事件 API
686b
存储 API
688b
认证 API
690b
授权 API
692b
Cookie API
694b
密钥 API
696b
凭证 API
602b
云计算机服务（内部 Nimbula）
698b
身份存储库
700b
全局配置
702b
消息存储库
704b
全局租户
706b
个性化设置
708b
资源
710b
用户瞬态数据
712b
系统瞬态数据
714b
每租户模式（受管理的 ExaData）
716b
云存储服务
718b
云通知服务
720b
云度量服务（Graphite）
722b
云监视服务（EM）

图6A

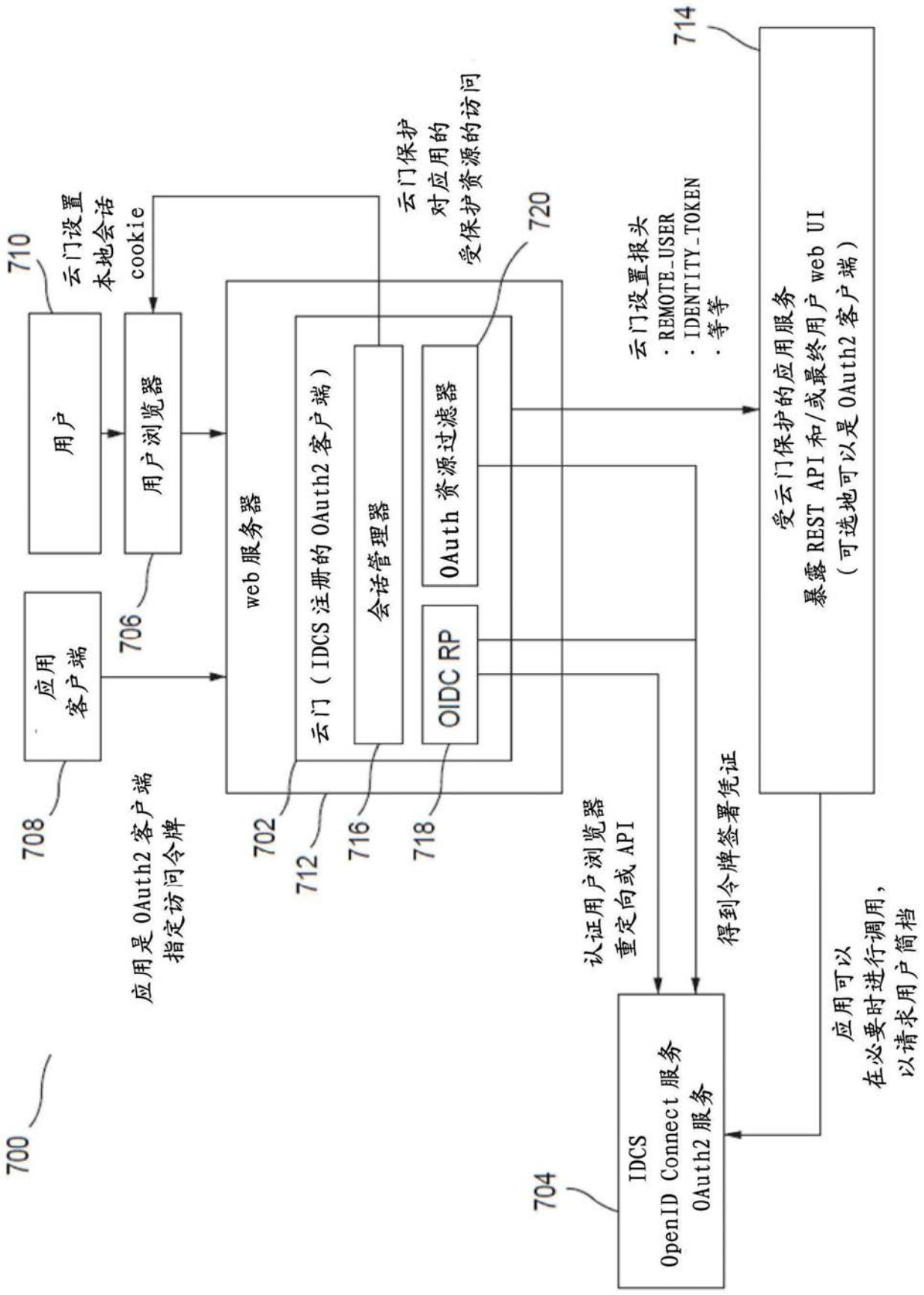

700
710
用户
云门设置
本地会话
cookie
用户浏览器
706
应用
客户端
708
应用是 OAuth2 客户端
指定访问令牌
web 服务器
云门（IDCS 注册的 OAuth2 客户端）
会话管理器
OIDC RP
OAuth 资源过滤器
720
云门保护
对应用的
受保护资源的访问
702
712
716
718
云门设置报头
· REMOTE-USER
· IDENTITY-TOKEN
· 等等
714
受云门保护的应用服务
暴露 REST API 和/或最终用户 web UI
（可选地可以是 OAuth2 客户端）
认证用户浏览器
重定向或 API
得到令牌签署凭证
应用可以
在必要时进行调用，
以请求用户简档
704
IDCS
OpenID Connect 服务
OAuth2 服务

图7

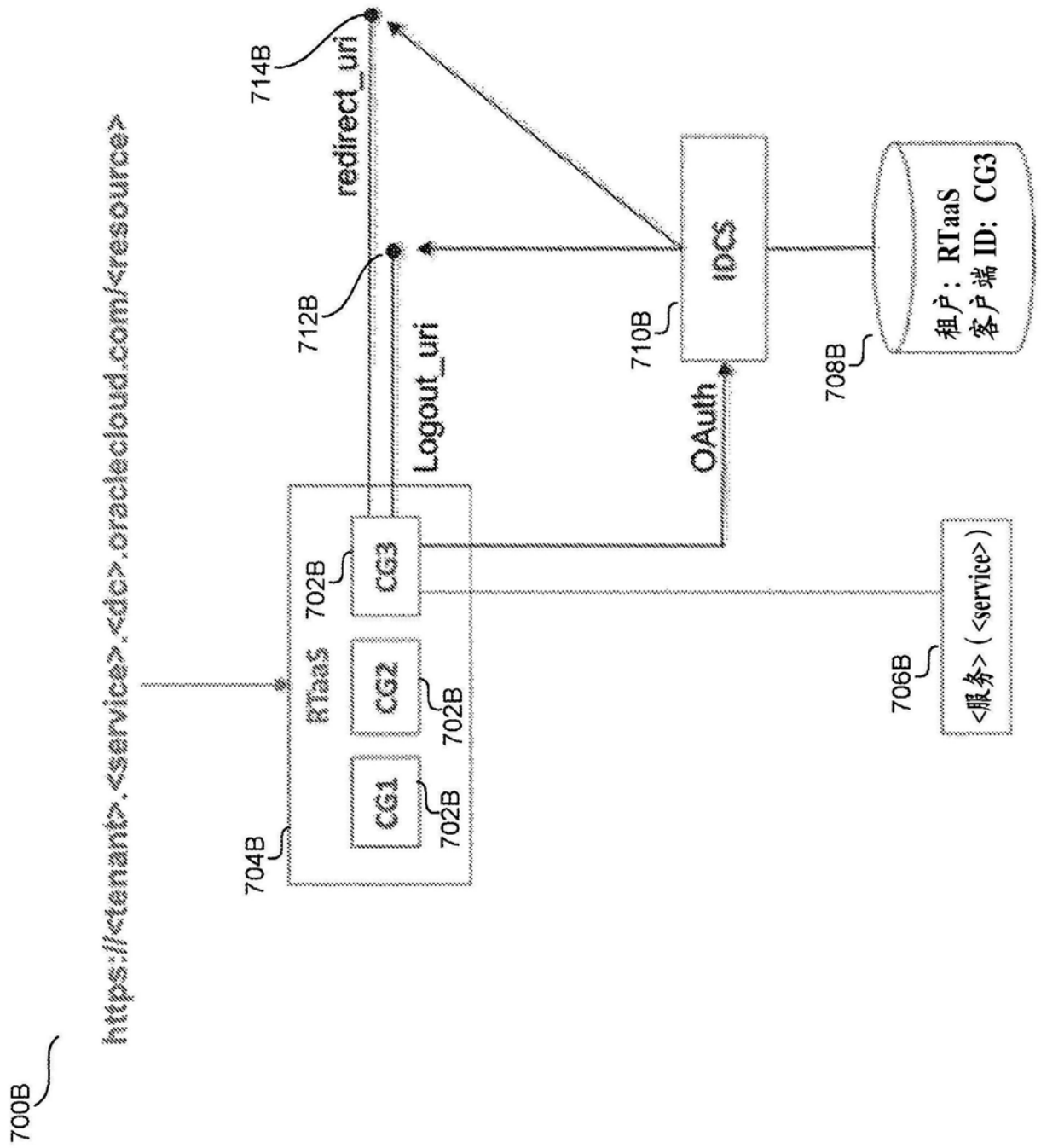

700B
https://<tenant>.<service>.<dc>.oraclecloud.com/<resource>
714B
redirect_uri
712B
Logout_uri
710B
IDCS
OAuth
708B
租户：RTaaS
客户端 ID：CG3
704B
RTaaS
702B
CG3
CG2
CG1
706B
<服务>（<service>）

图7A

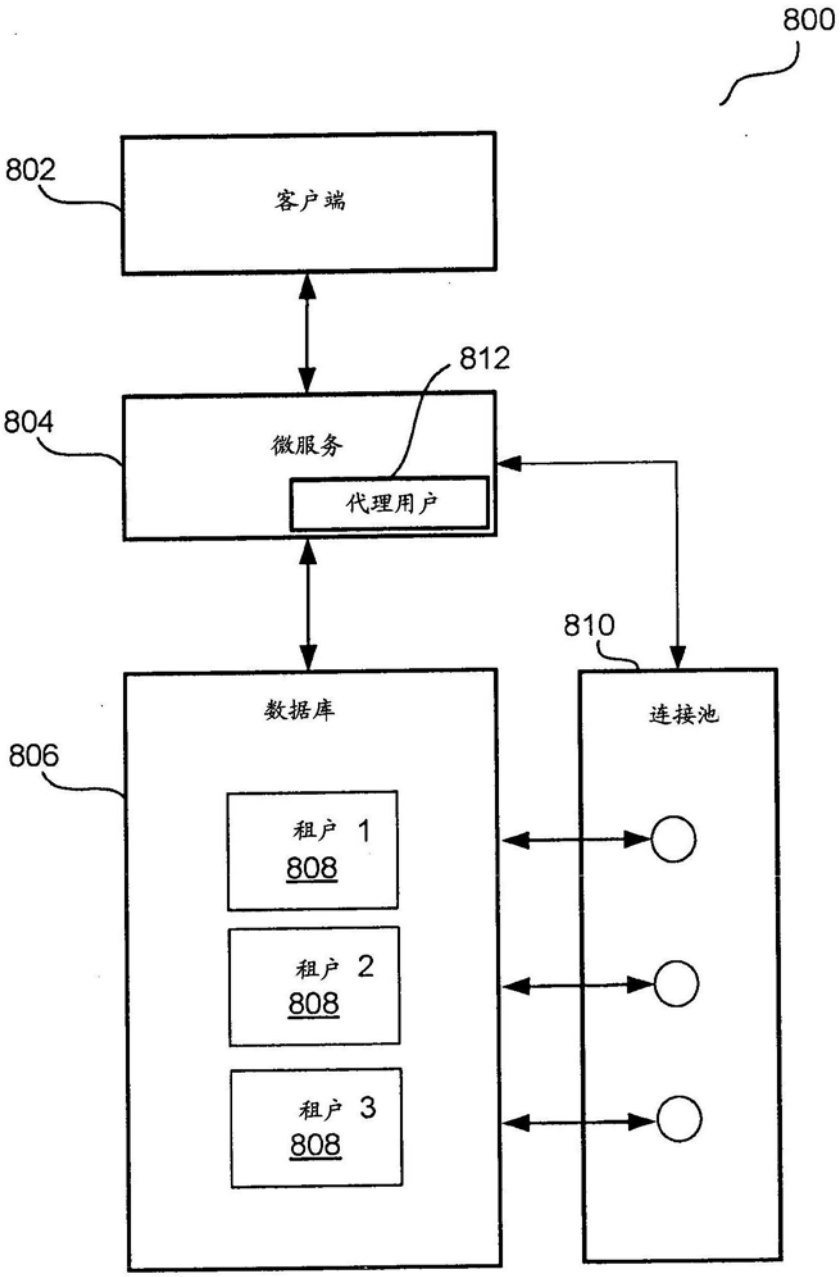

800
802
客户端
812
804
微服务
代理用户
810
数据库
连接池
806
租户 1
808
租户 2
808
租户 3
808

图8

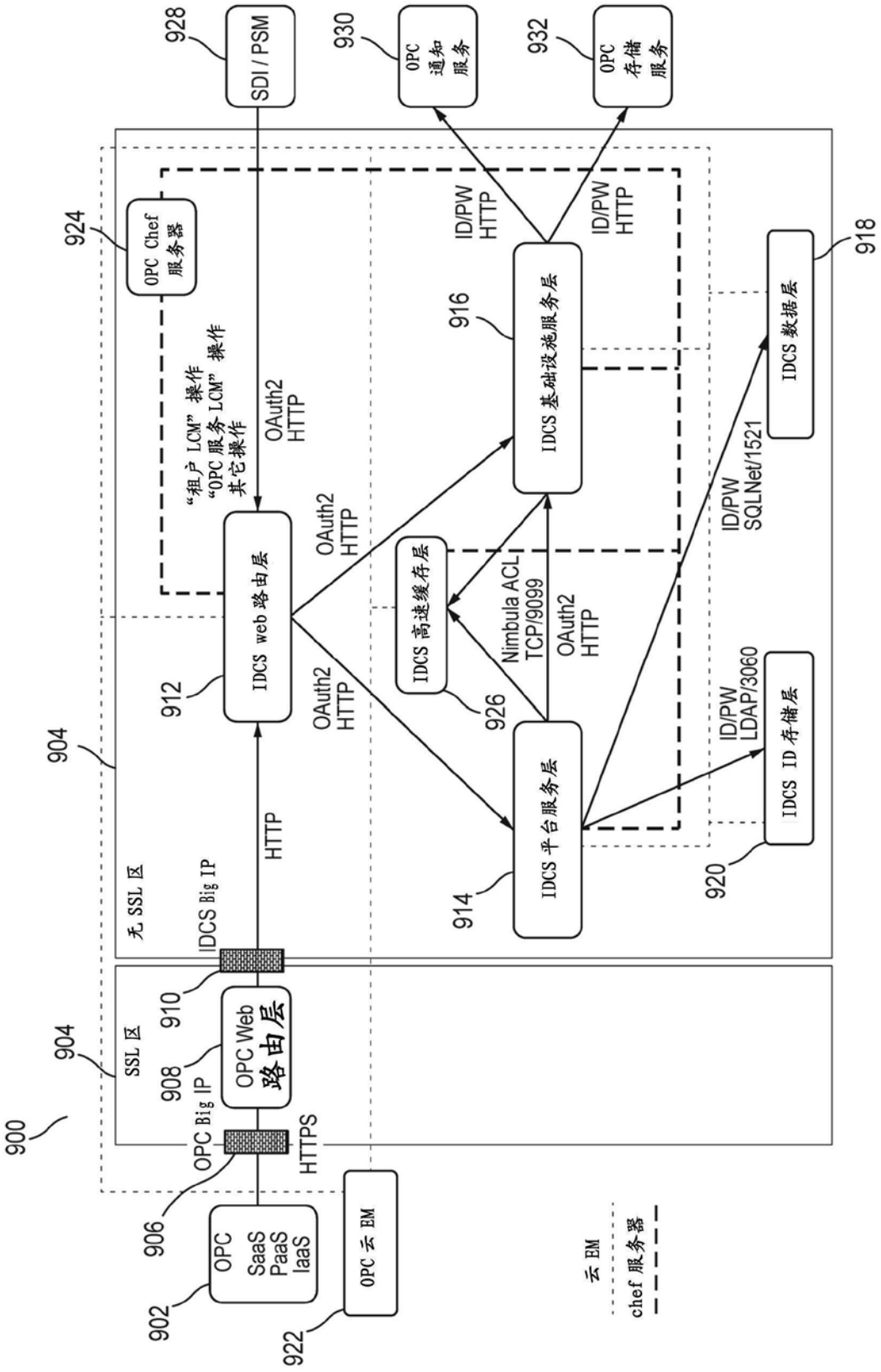

900
902
SaaS
PaaS
IaaS
OPC
906
OPC Big IP
HTTPS
904
SSL 区
908
OPC Web 路由层
910
IDCS Big IP
HTTP
904
无 SSL 区
912
IDCS web 路由层
914
IDCS 平台服务层
916
IDCS 基础设施服务层
918
IDCS 数据层
920
IDCS ID 存储层
922
OPC 云 EM
924
OPC Chef 服务器
926
IDCS 高速缓存层
928
SDI / PSM
930
OPC 通知服务
932
OPC 存储服务
"租户 LCM" 操作
"OPC 服务 LCM" 操作
其它操作
OAuth2
HTTP
ID/PW
HTTP
ID/PW
SQLNet/1521
ID/PW
LDAP/3060
Nimbula ACL
TCP/9099
云 EM
chef 服务器

图9

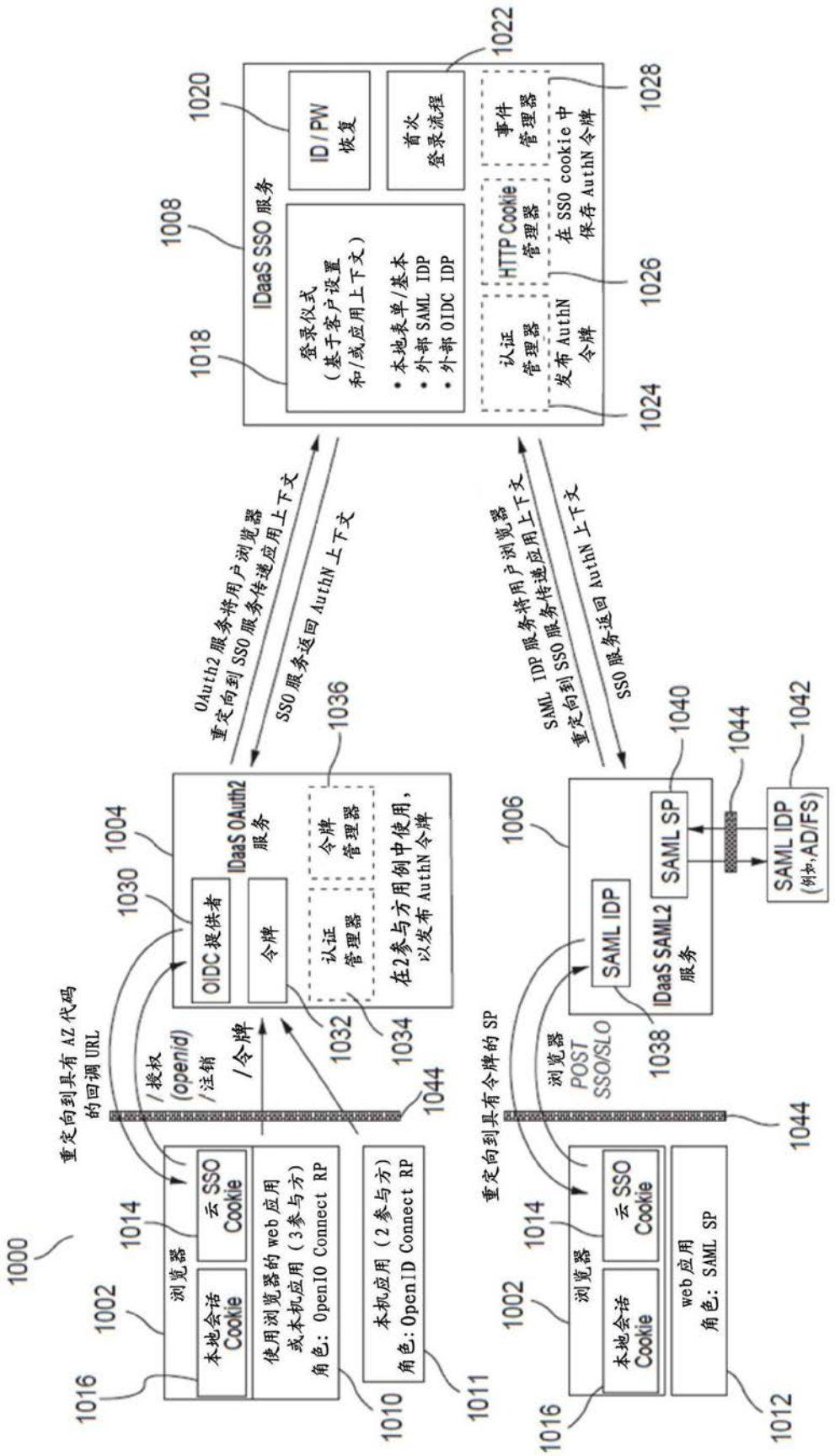

1000
1002
1004
1006
1008
1010
1011
1012
1014
1016
1018
1020
1022
1024
1026
1028
1030
1032
1034
1036
1038
1040
1042
1044
浏览器
本地会话 Cookie
云 SSO Cookie
使用浏览器的 web 应用或本机应用（3参与方）
角色：OpenIO Connect RP
本机应用（2参与方）
角色：OpenID Connect RP
web 应用
角色：SAML SP
重定向到具有 AZ 代码的回调 URL
/授权 (openid)
/注销
/令牌
IDaaS OAuth2 服务
OIDC 提供者
令牌
认证管理器
令牌管理器
在2参与方用例中使用，以发布 AuthN 令牌
IDaaS SSO 服务
登录仪式（基于客户设置和/或应用上下文）
• 本地表单/基本
• 外部 SAML IDP
• 外部 OIDC IDP
ID/PW 恢复
首次登录流程
HTTP Cookie 管理器
事件管理器
认证管理器
发布 AuthN 令牌
在 SSO cookie 中保存 AuthN 令牌
OAuth2 服务将用户浏览器重定向到 SSO 服务传递应用上下文
SSO 服务返回 AuthN 上下文
SAML IDP 服务将用户浏览器重定向到 SSO 服务传递应用上下文
SSO 服务返回 AuthN 上下文
重定向到具有令牌的 SP
浏览器 POST SSO/SLO
SAML IDP
IDaaS SAML2 服务
SAML SP
SAML IDP (例如,AD/FS)

图10

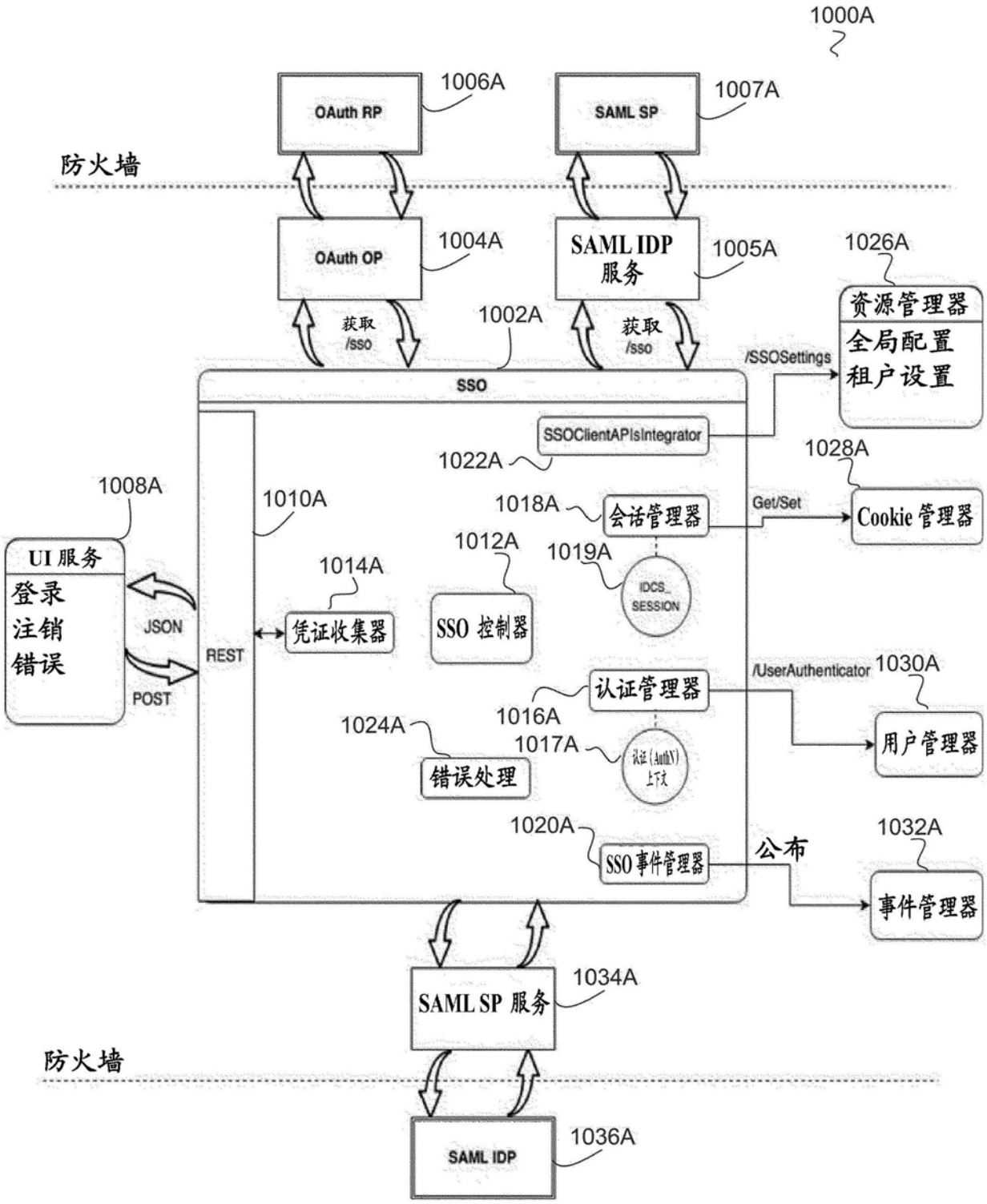

1000A
OAuth RP
1006A
SAML SP
1007A
防火墙
OAuth OP
1004A
SAML IDP
服务
1005A
1026A
资源管理器
全局配置
租户设置
获取
/sso
1002A
获取
/sso
/SSOSettings
SSO
SSOClientAPIsIntegrator
1022A
1028A
1008A
1010A
1018A
会话管理器
Get/Set
Cookie 管理器
UI 服务
登录
注销
错误
1012A
1019A
IDCS_
SESSION
1014A
JSON
凭证收集器
SSO 控制器
REST
POST
/UserAuthenticator
1030A
认证管理器
1016A
1024A
1017A
用户管理器
错误处理
1020A
1032A
SSO 事件管理器
公布
事件管理器
1034A
SAML SP 服务
防火墙
1036A
SAML IDP

图10A

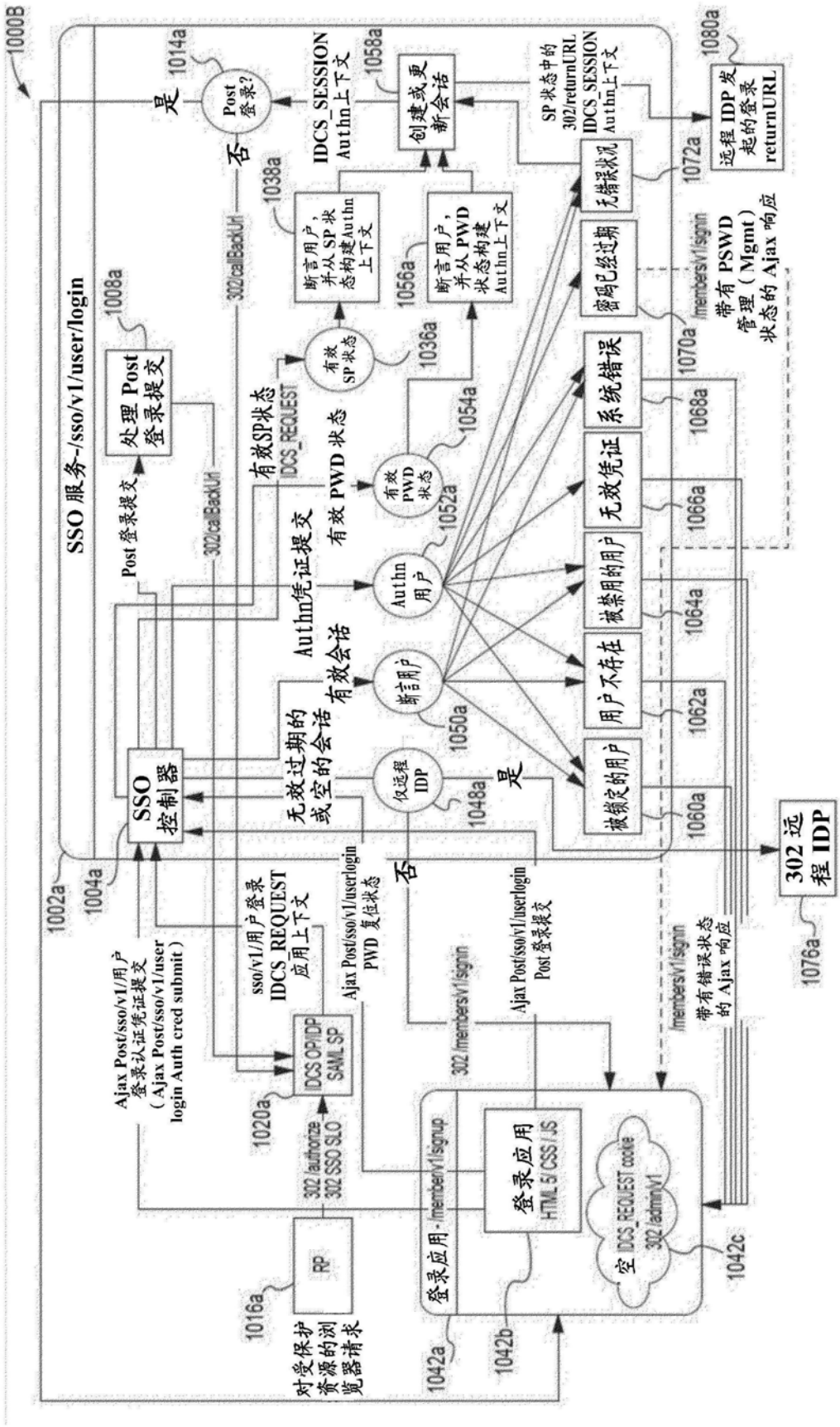

1000B
SSO 服务-/sso/v1/user/login
1002a
1004a
SSO 控制器
Ajax Post/sso/v1/用户登录认证凭证提交 (Ajax Post/sso/v1/user login Auth cred submit)
sso/v1/用户登录 IDCS_REQUEST 应用上下文
Ajax Post/sso/v1/userlogin PWD 复位状态
Ajax Post/sso/v1/userlogin Post 登录提交
302 /members/v1/signin
/members/v1/signin 带有错误状态的 Ajax 响应
302 远程 IDP
1076a
1020a
DCS OP/IDP SAML SP
302 /authorize 302 SSO SLO
1016a
RP
对受保护资源的浏览器请求
登录应用 - /member/v1/signup
登录应用 HTML 5/ CSS / JS
空 IDCS_REQUEST cookie 302 /admin/v1
1042a
1042b
1042c
Post 登录提交
处理 Post 登录提交
1008a
302/callBackUrl
1014a
Post 登录?
是
否
1038a
断言用户，并从 SP 状态构建 Authn 上下文
IDCS_SESSION Authn上下文
1058a
创建或更新会话
1056a
断言用户，并从 PWD 状态构建 Authn上下文
1036a
有效 SP 状态
有效SP状态
IDCS_REQUEST
有效 PWD 状态
1052a
1054a
Authn凭证提交
Authn 用户
有效会话
1050a
断言用户
无效过期的或空的会话
仅远程 IDP
1048a
是
否
被锁定的用户
用户不存在
被禁用的用户
无效凭证
系统错误
密码已经过期
无错误状况
1060a
1062a
1064a
1066a
1068a
1070a
1072a
SP 状态中的 302/returnURL
IDCS_SESSION Authn上下文
带有 PSWD 管理（Mgmt）状态的 Ajax 响应
1080a
远程 IDP 发起的登录 returnURL

图10B

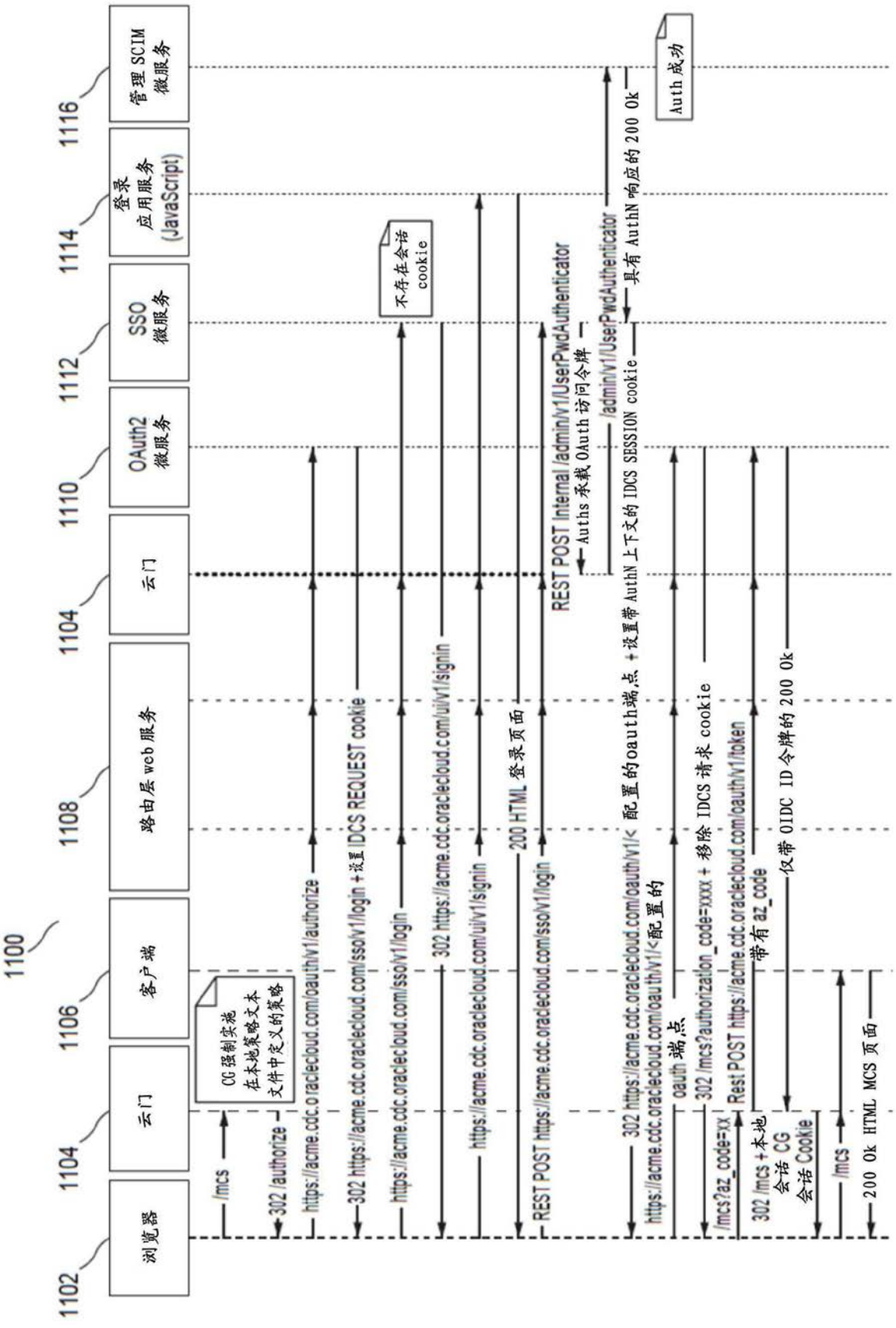

1100
1102
浏览器
1104
云门
1106
客户端
1108
路由层 web 服务
1104
云门
1110
OAuth2 微服务
1112
SSO 微服务
1114
登录 应用服务 (JavaScript)
1116
管理 SCIM 微服务
/mcs
CG强制实施 在本地策略文本 文件中定义的策略
302 /authorize
https://acme.cdc.oraclecloud.com/oauth/v1/authorize
302 https://acme.cdc.oraclecloud.com/sso/v1/login +设置 IDCS REQUEST cookie
https://acme.cdc.oraclecloud.com/sso/v1/login
不存在会话 cookie
302 https://acme.cdc.oraclecloud.com/ui/v1/signin
https://acme.cdc.oraclecloud.com/ui/v1/signin
200 HTML 登录页面
REST POST https://acme.cdc.oraclecloud.com/sso/v1/login
REST POST Internal /admin/v1/UserPwdAuthenticator
Auths 承载 OAuth 访问令牌
/admin/v1/UserPwdAuthenticator
具有 AuthN 响应的 200 Ok
Auth 成功
302 https://acme.cdc.oraclecloud.com/oauth/v1/< 配置的oauth端点, +设置带 AuthN 上下文的 IDCS SESSION cookie
https://acme.cdc.oraclecloud.com/oauth/v1/<配置的 oauth 端点,
302 /mcs?authorization_code=xxxx + 移除 IDCS 请求 cookie
/mcs?az_code=xx
Rest POST https://acme.cdc.oraclecloud.com/oauth/v1/token
302 /mcs + 本地 会话 CG 会话 Cookie
带有 az_code
仅带 OIDC ID 令牌的 200 Ok
/mcs
200 Ok HTML MCS 页面

图11

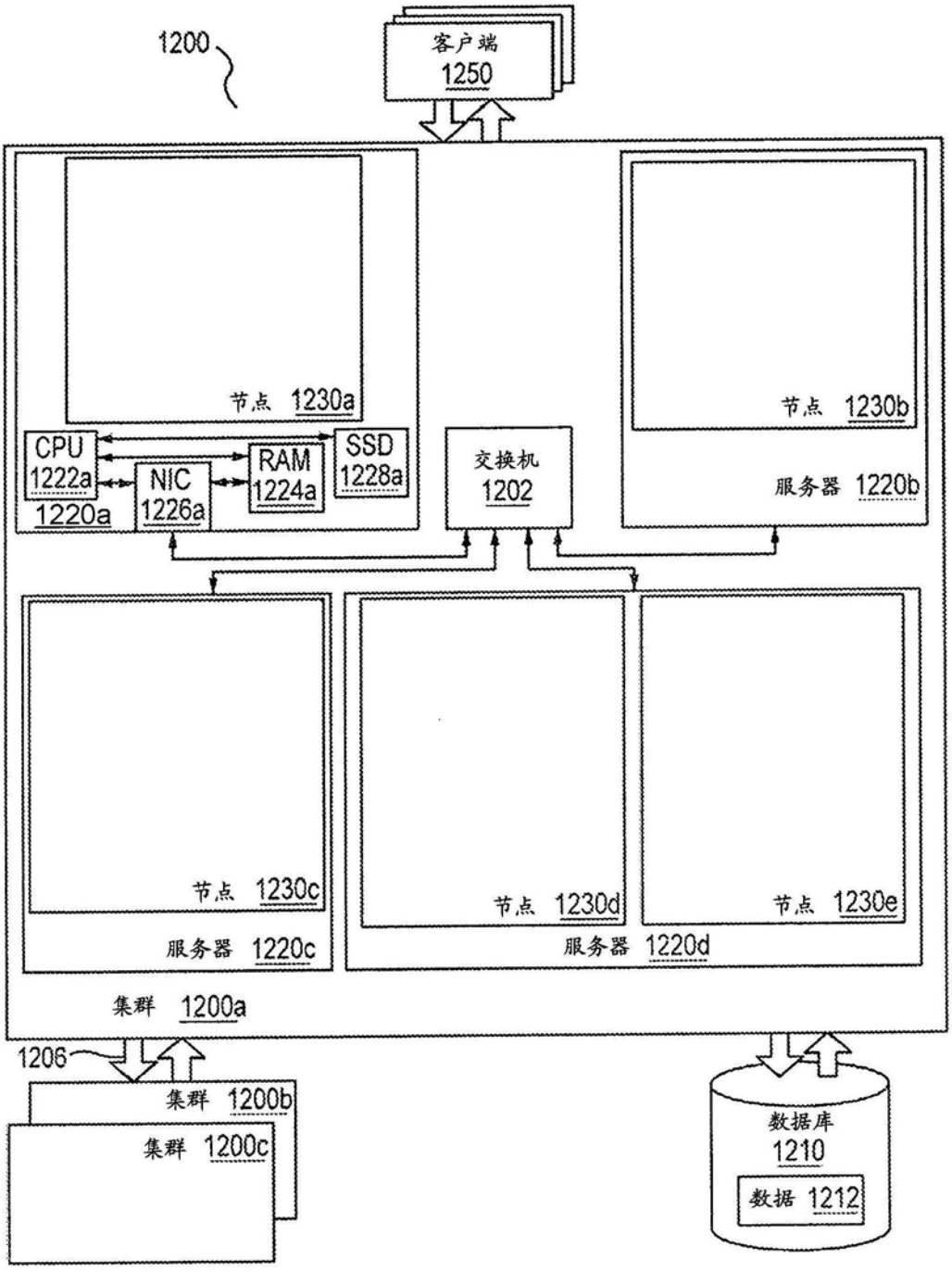

1200
客户端
1250
节点 1230a
CPU
1222a
NIC
1226a
RAM
1224a
SSD
1228a
1220a
交换机
1202
节点 1230b
服务器 1220b
节点 1230c
服务器 1220c
节点 1230d
节点 1230e
服务器 1220d
集群 1200a
1206
集群 1200b
集群 1200c
数据库
1210
数据 1212

图12

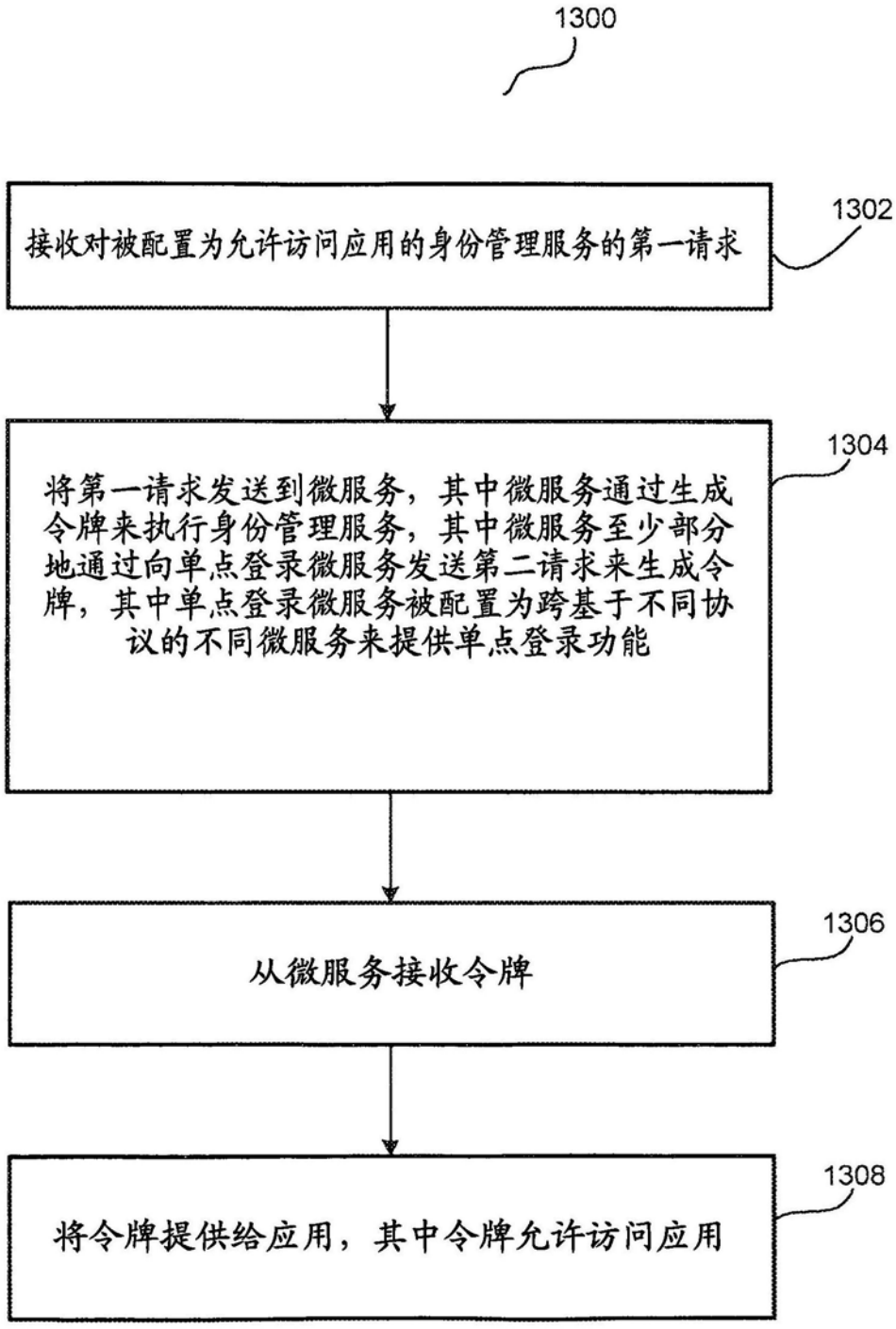

1300
接收对被配置为允许访问应用的身份管理服务的第一请求
1302
将第一请求发送到微服务，其中微服务通过生成令牌来执行身份管理服务，其中微服务至少部分地通过向单点登录微服务发送第二请求来生成令牌，其中单点登录微服务被配置为跨基于不同协议的不同微服务来提供单点登录功能
1304
从微服务接收令牌
1306
将令牌提供给应用，其中令牌允许访问应用
1308

图13

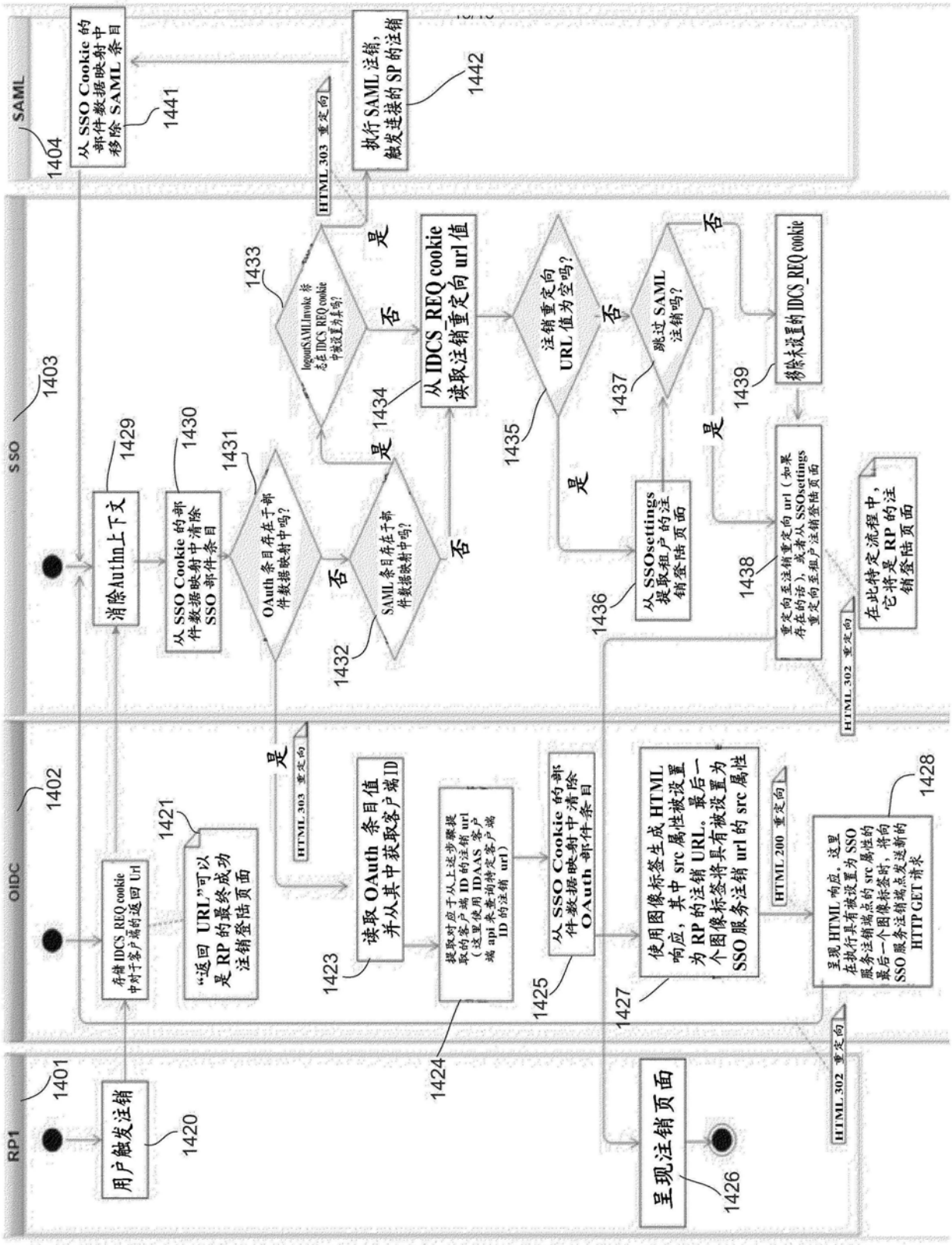

RP1
OIDC
SSO
SAML
用户触发注销
呈现注销页面
存储 IDCS REQ cookie 中对于客户端的返回 Url
"返回 URL"可以是 RP 的最终成功注销登陆页面
读取 OAuth 条目值并从其中获取客户端ID
提取对应于从上述步骤提取的客户端 ID 的注销 url（这里使用 IDAAS 客户端 api 来查询特定客户端 ID 的注销 url）
从 SSO Cookie 的部件数据映射中清除 OAuth 部件条目
使用图像标签生成 HTML 响应，其中 src 属性被设置为 RP 的注销 URL。最后一个图像标签将具有被设置为 SSO 服务注销 url 的 src 属性
呈现 HTML 响应。这里在执行具有被设置为 SSO 服务注销端点的 src 属性的最后一个图像标签时，将向 SSO 服务注销端点发送新的 HTTP GET 请求
消除Authn上下文
从 SSO Cookie 的部件数据映射中清除 SSO 部件条目
OAuth 条目存在于部件数据映射中吗?
SAML 条目存在于部件数据映射中吗?
logoutSAMLInvoke 标志在 IDCS REQ cookie 中被设置为真吗?
从 IDCS_REQ cookie 读取注销重定向 url 值
注销重定向 URL 值为空吗?
跳过 SAML 注销吗?
从 SSOsettings 提取租户的注销登陆页面
重定向至注销重定向 url（如果存在的话），或者从 SSOsettings 重定向至租户注销登陆页面
移除未设置的 IDCS_REQ cookie
在此特定流程中，它将是 RP 的注销登陆页面
从 SSO Cookie 的部件数据映射中移除 SAML 条目
执行 SAML 注销，触发连接的 SP 的注销
HTTP 303 重定向
HTTP 200 重定向
HTTP 302 重定向
是
否
1401
1402
1403
1404
1420
1421
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1441
1442

图14