(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 108701182 B
(45)授权公告日 2023.04.21

(21)申请号 201780012003.9
(22)申请日 2017.08.21
(65)同一申请的已公布的文献号
申请公布号 CN 108701182 A
(43)申请公布日 2018.10.23
(30)优先权数据
62/381,866 2016.08.31 US
15/680,362 2017.08.18 US
(85)PCT国际申请进入国家阶段日
2018.08.17
(86)PCT国际申请的申请数据
PCT/US2017/047726 2017.08.21
(87)PCT国际申请的公布数据
WO2018/044604 EN 2018.03.08
(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 G·威尔逊 V·R·米达姆
(74)专利代理机构 中国贸促会专利商标事务所有限公司 11038
专利代理师 邹丹
(51)Int.Cl.
G06F 21/41(2006.01)
G06F 21/60(2006.01)
(56)对比文件
US 2014164318 A1,2014.06.12
US 2014280948 A1,2014.09.18
US 2014090037 A1,2014.03.27
US 2016124742 A1,2016.05.05
US 2016112521 A1,2016.04.21
US 2016134619 A1,2016.05.12
CN 103532981 A,2014.01.22
CN 105631602 A,2016.06.01
审查员 刘永辉

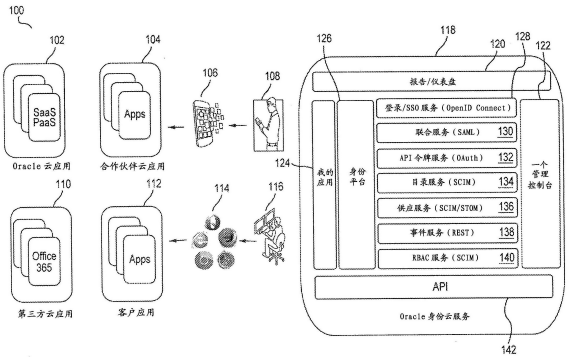
权利要求书3页 说明书31页 附图16页

(54)发明名称

多租户身份云服务的数据管理

(57)摘要

通过由网络门从应用程序接收对资源的请求来提供基于云的身份管理，其中该请求包括对多种资源类型中的资源类型的操作，并且该请求指定多个租户中的租户。实施例基于该请求访问微服务，解析资源类型，并基于元数据来验证资源类型支持该操作。实施例获取与租户相关联的数据提供者，调用数据提供者以执行该操作，并然后返回该资源。



CN 108701182 B

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

由网络门从应用程序接收对资源的请求,其中所述请求包括对多种资源类型中的资源类型的操作,并且所述请求指定多个租户中的租户,所述资源类型包括模式定义,并且所述模式定义包括多个属性和用于每个所述属性的元数据,所述资源类型包括用户或第二应用程序之一;

基于所述请求访问微服务;

解析所述资源类型,所述解析所述资源类型包括确定所述资源类型并检索对应的模式和模式定义;

基于元数据来验证所述资源类型支持所述操作;

获取与所述租户相关联的数据提供者;

调用所述数据提供者以执行所述操作;以及

返回所述资源。

2.根据权利要求1所述的非瞬态计算机可读介质,其中所述元数据包括JavaScript Object Notation(JSON)。

3.根据权利要求1所述的非瞬态计算机可读介质,其中所述验证包括检查有效的属性名称、有效的数据类型和缺少所需的属性。

4.根据权利要求1所述的非瞬态计算机可读介质,所述解析还包括基于所述模式来确定所述资源类型需要哪些属性以执行所述验证。

5.根据权利要求1所述的非瞬态计算机可读介质,其中,对于所述资源类型的多个版本,所述资源类型的至少一个版本包括指示相对于先前版本的弃用属性的标签,并且所述资源类型的至少一个版本包括指示相对于先前资源类型的添加属性的标签,所述提供还包括:

基于所述资源类型的对应标签,使用所述资源类型的所述版本来执行所述操作。

6.根据权利要求1所述的非瞬态计算机可读介质,其中,所述操作包括创建、更新、删除、获取或搜索中的一个。

7.根据权利要求1所述的非瞬态计算机可读介质,其中,所述资源类型是用户,并且所述对应的模式包括密码状态。

8.根据权利要求1所述的非瞬态计算机可读介质,其中,所述数据提供者包括数据库或轻量目录访问协议(LDAP)提供者之一。

9.一种提供基于云的身份管理的方法,所述方法包括:

由网络门从应用程序接收对资源的请求,其中所述请求包括对多种资源类型中的资源类型的操作,并且所述请求指定多个租户中的租户,所述资源类型包括模式定义,并且所述模式定义包括多个属性和用于每个所述属性的元数据,所述资源类型包括用户或第二应用程序之一;

基于所述请求访问微服务;

解析所述资源类型,所述解析所述资源类型包括确定所述资源类型并检索对应的模式和模式定义;

基于元数据来验证所述资源类型支持所述操作;

获取与所述租户相关联的数据提供者；

调用所述数据提供者以执行所述操作；以及

返回所述资源。

10.根据权利要求9所述的方法，其中，所述元数据包括JavaScript Object Notation(JSON)。

11.根据权利要求9所述的方法，其中，所述验证包括检查有效的属性名称、有效的数据类型和缺少所需的属性。

12.根据权利要求9所述的方法，所述解析还包括基于所述模式来确定所述资源类型需要哪些属性以执行所述验证。

13.根据权利要求9所述的方法，其中，对于所述资源类型的多个版本，所述资源类型的至少一个版本包括指示相对于先前版本的弃用属性的标签，并且所述资源类型的至少一个版本包括指示相对于先前资源类型的添加属性的标签，所述方法还包括：

基于所述资源类型的对应标签，使用所述资源类型的所述版本来执行所述操作。

14.根据权利要求9所述的方法，其中，所述操作包括创建、更新、删除、获取或搜索中的一个。

15.根据权利要求9所述的方法，其中，所述资源类型是用户，并且所述对应的模式包括密码状态。

16.根据权利要求9所述的方法，其中，所述数据提供者包括数据库或轻量目录访问协议(LDAP)提供者之一。

17.一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

一个或多个处理器，所述一个或多个处理器：

由网络门从应用程序接收对资源的请求，其中所述请求包括对多种资源类型中的资源类型的操作，并且所述请求指定所述多个租户中的租户，所述资源类型包括模式定义，并且所述模式定义包括多个属性和用于每个所述属性的元数据，所述资源类型包括用户或第二应用程序之一；

基于所述请求访问所述多个微服务中的微服务；

解析所述资源类型，所述解析所述资源类型包括确定所述资源类型并检索对应的模式和模式定义；

基于所述元数据来验证所述资源类型支持所述操作；

获取与所述租户相关联的数据提供者；

调用所述数据提供者以执行所述操作；以及

返回所述资源。

18.根据权利要求17所述的系统，其中，所述元数据包括JavaScript Object Notation(JSON)。

19.根据权利要求17所述的系统，其中，所述验证包括检查有效的属性名称、有效的数据类型和缺少所需的属性。

20.根据权利要求17所述的系统，所述解析还包括基于所述模式来确定所述资源类型

需要哪些属性以执行所述验证。

多租户身份云服务的数据管理

[0001] 相关申请的交叉引用

[0002] 本申请要求于2016年8月31日提交的美国临时专利申请序列第62/381,866号的优先权,该临时专利申请的公开内容以引用的方式并入本文。

技术领域

[0003] 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

[0004] 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

[0005] 实施例通过由网络门(web gate)从应用程序接收对资源的请求来提供基于云的身份管理,其中请求包括对多种资源类型中的资源类型的操作,并且请求还指定多个租户中的租户。实施例基于请求来访问微服务、解析资源类型,并基于元数据来验证所述操作由所述资源类型支持。实施例获取与租户相关联的数据提供者,调用数据提供者执行操作,并然后返回资源。

附图说明

[0006] 图1-5是提供基于云的身份管理的示例实施例的框图。

[0007] 图6是提供实施例的系统视图的框图。

[0008] 图6A是提供实施例的功能视图的框图。

[0009] 图7是实现云门的实施例的框图。

[0010] 图8图示了在一个实施例中实现多个租户的示例系统。

[0011] 图9是实施例的网络视图的框图。

[0012] 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。

[0013] 图11是一个实施例中的SSO功能的消息序列流程。

[0014] 图12图示了一个实施例中的分布式数据网格的实例。

[0015] 图13图示了根据一个实施例的用于身份云服务(“IDCS”)或身份即服务(“IDaaS”)的数据管理器架构。

[0016] 图14图示了由图13的资源数据管理器实现的本发明实施例的功能流程。

[0017] 图15图示了根据一个实施例的自动模式版本控制。

具体实施方式

[0018] 实施例实现定义资源类型和关联模式的元数据。使用元数据来解析对多租户系统中资源执行操作的请求，以确定与执行该操作的租户相关联的数据提供者。

[0019] 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性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、控制和审计跨整个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。

[0020] 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和近实时的任务。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中。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身份和访问管理(“IAM”)平台。

[0021] 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、发送激活等等。身份云服务还支持终端用户例如修改个人简档、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等。

[0022] 访问的统一安全性

[0023] 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性控制扩展到云环境中。例如，如果一个人离开公司，那么实施例确保他们的账户在内部部署和云中都被禁用。

[0024] 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。

[0025] 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。

[0026] 身份云服务

[0027] 实施例提供了作为多租户、云规模的IAM平台的身份云服务(“IDCS”)。IDCS提供认证、授权、审计和联合。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,可以使用IDCS在这样各种服务/应用/系统中提供单点登录(“SSO”)功能。

[0028] 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那个服务。服务是可以被不同客户端用于不同的目的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)。

[0029] 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,术语微服务设想了一种软件架构设计模式,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,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少,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。

[0030] 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,并且允许及早和持续地发布软件。

[0031] 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中的任何其它模块/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”。

[0032] 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。

[0033] 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。

[0034] 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到近实时,具有有保证的递送和处理。实施例从网络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。

[0035] 实施例基于行业标准(例如,OpenID Connect、OAuth2、安全声明标记语言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。该实施例还提供了具有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。

[0036] 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、用户供应连接器等等)。

[0037] 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。

[0038] 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)的孤立(siloed)访问。这种孤立的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IDCS 118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来访问,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等等。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®**。

[0039] 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)以及授权服务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。

[0040] 集成工具

[0041] 一般而言,大型公司通常具有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。

[0042] 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”)、文档等等。

[0043] 应用210、212、214、216、218可以通过不同的渠道被访问,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等等。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,用于将来自云208的认证联合到内部部署AD 204(例如,使用密码232)。

[0044] 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。它是用于在受信任的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协议传送SCIM消息。

[0045] 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。

[0046] 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访问云应用210、212、214、216以及内部部署应用218。

[0047] 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IDM304(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。

[0048] 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；审计等等。OAM具有对SAML的内置支持。如果用户在IDCS 202中有账户，那么OIM连接器302和OAM联合306可以与Oracle IDM 304一起使用，以创建/删除那个账户并且管理来自那个账户的访问。

[0049] 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。

[0050] 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。

[0051] 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。

[0052] 图5是包括与图2-4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等等。

[0053] 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。

[0054] 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)。

[0055] 一个实施例可以提供以下功能：

[0056] • 维护身份存储，以跟踪已被授权的用户帐户、所有权、访问和许可，

[0057] • 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)，

[0058] • 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，

[0059] • 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。

[0060] 除了这些功能之外，实施例还可以提供：

[0061] • 云帐户供应，以管理云应用中的帐户生命周期，

[0062] • 更健壮的多因素认证(“MFA”)集成，

[0063] • 广泛的移动安全能力，以及

[0064] • 动态认证选项。

[0065] 一个实施例提供自适应的认证和MFA。一般而言，密码和挑战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。

[0066] 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身的重要组成部分，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。

[0067] 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，“特权蠕动(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动的“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。迁移到云可以进一步使这种情况复杂，并且该处理可以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、访问历史记录、供应/解除供应和精细粒度赋权。

[0068] 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。

[0069] 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。

[0070] 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甚至知道密码的方式。这个能力还为会话管理和附加功能铺平了道路。

[0071] 一般而言，大多数云服务利用API和管理界面，其为渗透者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提供者身上发现自己具有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。

[0072] API平台

[0073] 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是使得可用于使服务消费者使用由IDCS提供的服务的东西。

[0074] 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")。

[0075] 配置利用必要的端点与外部服务集成的应用以及保持那种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连接配置(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。

[0076] 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”))。

[0077] 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平衡器即服务(“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)。

[0078] 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。

[0079] 平台服务

[0080] 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++”)等等的平台服务。

[0081] OpenID Connect平台服务实现标准的OpenIDConnect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受信任的SSO服务，该服务为本地和联合登录实现用户登录/注销仪式。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 Foundation标准来实现OpenID Connect功能。

[0082] 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”)请求评论(“RFC”)文案6749来实现OAuth功能。

[0083] 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。

[0084] 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可访问的资源暴露。

[0085] 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本身的所有IDCS内部资源都作为基于SCIM的REST API暴露。对身份存储618的访问被隔离到SCIM++API。

[0086] 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标准定义的语言支持附加的IDCS内部资源(例如，密码策略、角色、设置等等)。

[0087] 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”等等。

[0088] 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。

[0089] 基础设施服务

[0090] 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。

[0091] 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。

[0092] 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，因为存在提供那个功能的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。

[0093] IDCS还包括数据存储，这些数据存储是IDCS所需/生成的数据储存库，包括身份存储618(存储用户、组等等)、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。

[0094] 实时和近实时任务

[0095] IDCS将所请求服务的任务分离成同步实时和异步近实时任务，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。

[0096] 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。

[0097] 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执行。

[0098] 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将消息在作为存储的队列628中排队。

[0099] 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。

[0100] 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。

[0101] IDCS使用实时高速缓存结构来增强系统性能和用户体验。高速缓存本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。

[0102] 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。

[0103] 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。

[0104] 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，那么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。

[0105] 在一个实施例中，IDCS可以被用于用户登录。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。

[0106] 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。

[0107] 无状态功能

[0108] 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。

[0109] 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。

[0110] 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。

[0111] IDCS网络层、中间层和数据层可以各自独立和分开地进行伸缩。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。

[0112] IDCS功能视图

[0113] 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。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。数据存储是IDCS所需/生成的数据储存库，包括身份存储698b、全局配置700b、消息存储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(未示出)、高速缓存存储(未示出)等等。

[0114] 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658b和私人帐户管理660b。

[0115] 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包括作业调度器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。

[0116] 在一个实施例中，IDCS为IDCS服务的消费者提供各种UI602b，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。

[0117] 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。

[0118] 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。

[0119] 云门

[0120] 在一个实施例中，IDCS在web层中实现“云门”。云门是web服务器插件，它使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。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,其提供用于基于OAuth保护HTTP资源的web策略强制实施点(“PEP”)。

[0121] 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”)集成的策略强制实施点(“PEP”),同时保护对应用的REST API资源和web浏览器714的访问。在一些实施例中,PDP在OAuth和/或OpenID Connect微服务704上实现。例如,当用户浏览器706向IDCS发送用于用户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的进一步的凭证)。在图7的实施例中,云门702可以既充当PEP又充当PDP,因为它具有本地策略。

[0122] 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、正则表达式等等)来维护关于应用的受保护和不受保护的资源的配置信息。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。

[0123] 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,那么云门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。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/观众等等)并且为用户710发布本地会话cookie。它充当会话管理器716,以保护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。

[0124] 云门户702还充当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。

[0125] 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、web应用、JavaScript等等)呈现访问令牌(由IDCS发布)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、观众等等)之前验证访问令牌。原始访问令牌未经修改就被传递。

[0126] 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。

[0127] 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流程中提供多租户。

[0128] 租赁类型

[0129] 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。

[0130] 这三种租赁的分离和标识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。

[0131] 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。

[0132] 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。

[0133] 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。

[0134] 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”中的用户可以需要在由“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己请求令牌。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。

[0135] 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”将自己识别为存在于“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。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”)。

[0136] 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，而多租赁是服务的业务功能中的不同维度。图8图示了实施例中实现多租赁的示例系统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是通过https://tenant3/oauth/token请求的OAuth微服务，用于获得令牌。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，那么使用来自不同租赁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。

[0137] 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。

[0138] 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。

[0139] 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，谁是那个连接的所有者、谁的凭证正在验证中、是数据库用户、还是系统凭证等)。

[0140] 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中的连接来提供那个数据。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代理用户上下文中(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。

[0141] 以下提供了用于实现代理用户812的示例功能：

[0142] dbOperation＝<准备要执行的DB命令>

[0143] dbConnection＝getDBConnectionFromPool()

[0144] dbConnection.setProxyUser(resourceTenant)

[0145] result＝dbConnection.executeOperation(dbOperation)

[0146] 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。

[0147] 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。

[0148] 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。

[0149] 在一个实施例中,身份令牌包括至少声称/声明,其指示用户租户名字(即,用户所在的地方)。与授权令牌联系的“声称”(如安全领域中普通技术人员所使用的)是主体关于其自己或另一个主体作出的声明。例如,声明可以是关于名字、身份、密钥、组、特权或能力。声称是由提供者发布的,并且它们被给赋予一个或多个值,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”)。

[0150] 在一个实施例中,访问令牌包括至少指示在对访问令牌作出请求时的资源租户名字(例如,客户)的声称/声明、指示用户租户名字的声称、指示作出请求的OAuth客户端的名字的声称,以及指示客户端租户名字的声称。在一个实施例中,访问令牌可以根据以下JSON功能来实现:

[0151]

```
{
...
 “tok_type”： “AT”，
  “user_id”： “测试用户”，
  “user_tenantname”： “<身份租户的值>”
  “tenant”： “<资源租户的值>”
  “client_id”： “测试客户端”，
  “client_tenantname”： “<客户端租户的值>”
...
}
```

[0152] 在一个实施例中，客户端断言令牌包括至少指示客户端租户名字的声称以及指示作出请求的OAuth客户端的名字的声称。

[0153] 本文描述的令牌和/或多种租赁可以在除IDCS以外的任何其它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。

[0154] 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。

[0155] 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。

[0156] 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公共云负载平衡外部虚拟IP地址(“VIP”)906(或其它类似技术)、公共云web路由层908以及IDCS负载平衡内部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。

[0157] 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

他们运行的应用的杰出的安全合规性。

[0158] 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。

[0159] 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”。

[0160] 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。

[0161] 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。

[0162] 云访问控制-SSO

[0163] 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。

[0164] 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/证书等等)。

[0165] 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。

[0166] 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OpenIDConnect,它是用于执行基于web的认证的轻量级协议。

[0167] 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。

[0168] 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。在同一浏览器内的第二应用上向用户提供SSO。因而,浏览器是状态或用户代理并且维护cookie。

[0169] 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Auth)、外部SAML IDP、外部OIDC IDP等等进行配置。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。

[0170] 在一个实施例中，OAuth微服务1004和SSO微服务1008之间的交互是基于浏览器重定向，使得SSO微服务1008使用HTML表单挑战用户、验证凭证并发布会话cookie。

[0171] 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。

[0172] 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。

[0173] 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。

[0174] 在一个实施例中，例如，SAML微服务1006可以充当SAML SP 1040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之间的交互是基于浏览器重定向，使得SSO微服务1008使用HTML表单挑战用户、验证凭证并发布会话cookie。

[0175] 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。

[0176] 登录/注销流程

[0177] 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，那么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)。

[0178] 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，应用可以做什么等等)，并将浏览器1102重定向到SSO微服务1112以便登录。

[0179] 如果用户具有有效的SSO会话，那么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),那么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。

[0180] 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,那么重定向到IDP,以针对经配置的IDP进行认证。

[0181] 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(由观众令牌声称来指示)。云门1104从OAuth2微服务1110接收令牌。

[0182] 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,认证过程就重复。

[0183] 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。

[0184] 当用户访问另一个OAuth2集成的连接应用时,重复相同的处理。

[0185] 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)。

[0186] 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),那么在SAML微服务和SSO微服务1112之间开始相似的处理。

[0187] 云高速缓存

[0188] 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。

[0189] 为了支持基于LDAP的应用，云高速缓存将LDAP通信翻译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。

[0190] 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之间进行翻译，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。

[0191] 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。

[0192] 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)的自身的身份管理服务。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。

[0193] 分布式数据网格

[0194] 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，如在例如美国专利公开No.2016/0092540中所公开的，其公开内容通过引用结合于此。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。

[0195] 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提供了点对点(即，没有中央管理器)、存储器内的分布式高速缓存。

[0196] 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。

[0197] 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)。

[0198] 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到96个物理服务器。

[0199] 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。

[0200] 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。

[0201] 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接口,或者通过网络或其它任何方式进行远程接口。

[0202] 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。

[0203] 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。

[0204] 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的云身份管理模块以及本文公开的所有其它功能。

[0205] 客户端可以访问网络服务,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 Server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。

[0206] 数据管理-元数据驱动框架

[0207] 在实施例中,需要管理大量不同类型的数据/资源,称为“资源类型”。资源类型的实例包括用户、组、应用程序(“app”)、配置、设置、密码策略等。资源类型是使用本发明的实施例在云中管理的管理实体。

[0208] 每种资源类型都有定义其行为的各种配置。一种配置是模式(即可以与该类型的资源相关联的数据),包括主/核心模式和扩展模式。例如,如果资源类型是“用户”,则每个用户都可以具有与其相关联的4-5个模式。核心模式包括用户名(例如,名、姓)、电话号码、地址以及该用户的任何相关简档信息。扩展模式可能包括密码状态(例如,用户密码已过期、超过密码尝试的最大次数等)或用户状态(例如,活跃或非活跃)。这些被视为“模式属性”。同样,另一种资源类型(如应用程序)也有自己的模式,该模式定义了资源如何被组织以及其数据如何相关。

[0209] 在一个实施例中,每个模式都具有属性列表。每个属性都有一组元数据,该组元数据定义了该属性的行为(例如,它是读/写的、它的可变性(即,可被改变/更改的能力)、它是否是必需的、是否可以搜索)。对于每种资源类型,都可以有资源类型定义和模式定义。资源类型定义可以包括:(1)模式列表—核心和可选扩展模式;(2)支持的操作-创建、替换、更新、删除、获取、搜索、后期搜索;(3)数据提供者类型:LDAP提供者、数据库(“DB”)提供者、通知提供者等。模式定义可以包括属性列表和用于每个属性的描述其特性和行为的元数据——数据类型、可变性(只读、读写、不可变、只写)、返回(总是、请求、从不)、目标属性名称、去除空格(trimspace)、最大长度(maxlength)等。因此,元数据补充了资源类型的定义、与之关联的模式以及用于每个属性的元数据。

[0210] 在一个实例实施例中,有175种不同的资源类型。如果需要软件代码来管理这些资源类型中的每一种资源类型和每个特定模式,则需要大量特定于每种资源类型和模式的代

码。例如,对于为特定属性(例如获取用户名、设置用户名、获取名字、获取姓氏)提供getter和setter的“POJO”(普通旧Java对象)资源类型,所有资源类型都需要大量代码。

[0211] 相反,在本发明的实施例中,单个统一代码/模块对于任何资源类型执行所有功能(例如,创建、更新、修改、删除、搜索等)。首先,实施例确定资源类型,然后查找模式和模式定义,以确定属性和属性定义。实施例随后可以作出运行时决定,诸如需要该资源类型的哪些属性用于其模式,以便执行验证。在执行将数据库中的数据保留在LDAP中时,实施例知道资源类型的属性是什么,因此实施例可以为SQL更新/插入正确地构造有效负载。对于每个资源,资源类型和模式定义包含配置的提供者类型和目标属性映射。如果提供者类型为LDAP,则调用LDAP提供者,以便按照资源类型和模式定义中的配置,使用目录树下的正确对象类,将数据保留在LDAP中。如果提供者类型是DB提供者,则提供者将生成SQL,以便按照资源类型和模式定义中的配置,将DB中的数据保留在右列的表中/从右列的表中获取DB中的数据。

[0212] 因此,一个实施例完全是数据驱动的,因此如果开发人员想要添加新的资源类型,开发人员只需要添加新资源类型和模式的JSON定义,而不是编写新代码。实施例被认为是数据驱动模型,因为实施例完全是元数据驱动的。实施例从运行时代码执行数据的完全抽象。因此,实施例不需要为每种资源类型编写、维护或测试单独的代码。

[0213] 在一个实施例中,出于性能原因,在服务启动时高速缓存所有元数据。此外,实施例可以使用元数据来驱动文档,并且可以基于资源类型和模式定义,自动生成包括REST API的外部文档。

[0214] 图13图示了根据一个实施例的用于IDCS或身份即服务(“IDaaS”)的数据管理器架构。该架构包括资源管理器1301(或“ResourceManager”)、SCIM/REST层1310、API层1311、数据提供者(“DP”)层1312和数据存储层1313。

[0215] 资源管理器1301是IDCS/IDaaS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定义的任何资源类型。在一个实施例中,资源管理器1301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、规范值等的基于模式的验证。它还处理设置默认值、创建/更新日期、创建/更新者、保护敏感属性、映射到目标属性、授权检查和事件发布。API层1311中的资源类型特定管理器扩展资源管理器1301,以处理任何资源类型特定操作或扩展CRUDQ操作的公共逻辑。资源管理器1301通过数据提供者层1313中的数据提供者(它们直接与数据存储特定接口集成)与数据存储层1313中的数据存储集成。因为资源管理器1301是元数据驱动的,所以它支持运行时模式定制和数据提供者(DataProvider)配置改变,而不影响资源管理器1301或资源类型管理器1311。

[0216] 资源类型(或“ResourceType”)是由IDCS管理的资源类型。实例包括用户(User)、组(Group)、应用程序(Application)、令牌(Token)、密钥(Key)等。在SCIM中,每个资源类型都是顶级端点(例如,/用户(/Users)、/组(/Groups))。每种资源类型都有资源类型定义(或“ResourceTypeDef”)。资源类型定义是描述给定资源类型的元数据。资源类型定义定义了资源类型名称、端点、主模式URI和扩展模式URI(如果有的话)。此外,它还定义了若干IDCS特定的元数据。每个资源类型定义都是JSON的blob。资源类型可以是预播种的,也可以在运行时创建。任何资源类型定义都可以配置为内部的,意味着不能通过SCIM REST 1310来访问它,也可以配置为外部的,意味着能够通过SCIM REST GET/ResourceType和GET/

Schemas/<ResourceType>来发现它。

[0217] 模式定义(或“SchemaDef”)是描述整个或部分资源类型(例如“设备(Device)”)的内容的属性定义(或“AttributeDefs”)的集合。SCIM定义核心元数据,而根据实施例,IDCS扩展元数据。具体而言,模式定义描述了该资源类型的每个属性和子属性。每个模式定义都是JSON的blob。模式定义可以是预播种的,也可以在运行时创建。

[0218] 属性定义定义名称和元数据,例如类型(例如字符串、二进制)、基数(单、多、复杂)、可变性(只读、读写等))、可退回性、可搜索性等。属性名称应该是“驼峰式”(例如,camelCased),并且在定义它们的模式中应该是唯一的。在一个实施例中,属性数据类型可以是以下各项之一:字符串;布尔型;十进制;整数;日期时间;二进制;引用;或者复合。

[0219] 单一属性是包含0..1值的资源属性(例如,“displayName”)。多值属性是包含0..n值的资源属性(例如,“emails”)。简单属性是单一或多值属性,其值是一个原语(例如“字符串”)。复合属性是单一或多值属性,其值是一个或多个简单属性(例如“地址”)的组合,并具有子属性(例如“街道地址”、“地点”、“邮政编码”和“国家”)。子属性是包含在复合属性中的简单属性。

[0220] 资源是包含一个或多个属性的IDCS管理的工件的实例。在SCIM中,资源是可以被读取和操作的对象,例如特定的用户、组或令牌。每个资源都有一个全局唯一标识符,并包含符合相应资源类型定义的模式的属性值。

[0221] 在一个实施例中,在API层1311中,为每个资源类型实现Java类,作为资源类型管理器。例如,用户管理器管理用户。组管理器管理组。每个管理器暴露适用于管理该资源类型的对象的接口。每个资源类型管理器扩展一个公共抽象资源管理器,该管理器实现创建、替换、更新、删除、获取和搜索资源的方法。如果需要,每个资源类型管理器都可以为抽象资源管理器实现的每个方法实现自定义验证。此外,每个资源类型管理器都可以根据需要扩展这些方法。例如,用户管理器暴露用户资源特有的方法,包括启用、禁用、锁定、解锁、更改密码。组管理器暴露用户管理器不暴露的方法,例如授予或撤销用户成员资格。

[0222] 在一个实施例中,基于JSR-330标准注释的HK2将用于API层1311。每个资源类型管理器的自定义Java接口都将使用@Contract来注释,并且其impls将使用@Service来注释。这将确保资源类型管理器类将被放入在资源类型管理器和<ResourceType>managerlmpl下通告的服务注册表中,并且能够通过服务定位器(ServiceLocator)来请求。

[0223] 在一个实施例中,资源管理器1301是无状态的公共Java类,其定义了用于查询和管理任何资源类型的资源的一组API。资源管理器的接口将使用@Contract来注释,<ResourceType>Manager的impls将使用@Service来注释,以确保它们将被放入服务注册表,并能够通过服务定位器来请求。

[0224] 抽象资源管理器(或“AbstractResourceManager”)实现资源管理器接口,提供每个<ResourceType>Manager继承的公共行为。例如,AbstractResourceManager检查授权,基于ResourceTypeDef执行验证,并发出创建、替换、更新和删除操作的事件。此外,AbstractResourceManager方法还将回调<ResourceType>Manager,以启用自定义验证。

[0225] 数据提供者层1312是资源管理器1301下面的可插拔层。它对底层数据存储实现每个操作。例如,JDBC数据提供者使用JDBC来与数据库进行对话。JNDI数据提供者使用JNDI来与目录服务进行对话。数据提供者将根据每个请求的租户ID来在数据存储之间进行切换。

[0226] 一个实施例最初支持两种类型的数据存储:JDBC和JNDI。其他实施例支持其他数据存储,例如NoSQL。数据存储是租户特定的,可以是资源类型特定的。例如,事件可以存储在与用于存储应用程序的JDBC数据库分离的JDBC数据库中。

[0227] 如上所述,实施例是数据驱动的,因为公共资源管理器1301不仅处理验证、创建以及错误处理和异常,而不管资源类型如何,而且还基于元数据来生成事件。事件在消息服务中排队,由后端处理程序进行处理,并接受审核、生成通知等。

[0228] 图14图示了图13的资源数据管理器实现的本发明实施例的功能流程。功能包括:

[0229] (1)在1401,解析资源类型。

[0230] (2)在1402,验证资源类型支持该操作(即,验证用户试图做什么)。例如,有些资源类型支持创建、更新和删除,而有些仅支持获取和搜索。支持由元数据决定。

[0231] (3)在1403,通过进行授权检查来授权,以确定是否被授权执行操作。

[0232] (4)在1404、1405,如果需要(即,某些资源类型需要定制),例如定制验证、定制预处理、定制后处理、定制事件生成,则实现对资源管理器的回调。回调是从公共流调用的,并且是新颖的,因为它允许每种资源类型注入特定于资源类型的行为。

[0233] (5)在1406,基于请求,获得数据提供者。对于给定的资源类型,不管租户是谁,都只有一个数据提供者。基于租户,数据提供者建立到正确数据库模式(对于存储在DB中的资源)的连接,或者指向正确的目录树(对于利用LDAP存储的资源)。在一个实施例中使用“Getdataprovider”Java方法。

[0234] (6)在1406,调用数据提供者来执行操作(例如,在图14的实例中,操作是“创建”)。

[0235] (7)在1407,在创建操作之后,对资源管理器进行回调,以确定是否需要任何后处理(例如,在结果中注入附加属性,修改数据)。向资源管理器提供插件回调。

[0236] (8)在1408,发布事件。

[0237] (9)在1409,返回在POST响应中返回的资源。

[0238] 在一个实施例中,在图14的流程中元数据的主要用途是对照为该资源类型定义的模式来验证伴随Post请求1409而来的有效负载。这是基于高速缓存的资源数据动态完成的。但是,资源类型元数据不仅用于验证,因为在请求开始时加载的元数据可以用于资源管理器处理的每个阶段,包括:验证1402(检查有效的属性名称、数据类型、缺少所需的属性等)、授权1407、数据提供者(Dataprovider)操作1430(目标属性映射、表或对象类映射等)、后处理1409(基于元数据中的返回属性特性来过滤掉数据)、发布事件1408(要发布的事件在元数据中)。

[0239] 实施例确定请求来自哪里以及谁是资源管理器的用户。用户可以是最终用户、另一应用程序、内部IDCS组件等。通常,客户端发出请求——HTTP请求的形式的API请求,然后将该请求移交给资源管理器,以便基于资源元数据进行处理。

[0240] 在一个实施例中,资源管理器1301是被称为“管理服务”的微服务,并且处理对所有资源类型的管理。它在生成事件时与消息服务(即微服务)进行对话。再次参考图13,资源管理器1301表示管理服务,而整个图13(除了底部的数据库)是管理服务微服务。

[0241] 一个实施例支持IDCS的高速缓存一致性。例如,对于获取请求:get/user/ID,实施例将首先查询高速缓存数据提供者以查看用户ID(User ID)是否被高速缓存。如果被高速缓存,则数据将从高速缓存中返回。如果没有被高速缓存,则从数据存储中检索数据,并在

返回途中将数据添加到一致性高速缓存中。

[0242] 基于租户,实施例使用元数据来确定从哪个DB(例如,图13的LDAP数据分区“DP”1305)读取。实施例基于租户在数据层执行DB切换。

[0243] 实施例可由需要管理数据的任何服务使用。使用实施例的客户端可以包括UI服务控制台、导入作业或正在IDCS中更新数据的任何事物。

[0244] 数据管理-多租户

[0245] 一个实施例在数据层实现多租户支持。资源管理器1301处理请求,然后为该请求请求适当的数据源。处理是租户驱动的,与元数据功能分离。该功能为实施例提供了基于租户在数据存储之间进行运行时切换的能力,这有助于安全性(即租户数据的隔离)。

[0246] 其他已知的身份管理器可能没有多个租户。相反,一些已知的身份管理器系统针对每个租户的每个表,使用数据库中的不同列来执行条带化。在这些解决方案中,一个表将多个租户中的所有租户的数据混合在一起,这可能不安全。

[0247] 相反,在一个实施例中,为每个租户使用不同的数据库,而不是条带化。实施例根据请求执行到适当数据源的运行时切换。

[0248] 数据管理-自动模式版本控制

[0249] 在一个实施例中,在资源类型(例如,“用户”资源类型)的生命周期中,版本1可以具有用户的模式。在资源类型的后续版本2中,可能需要添加或删除属性,因此可能需要复制表示具有版本1的所有属性的模式和具有版本2的所有属性的版本2模式的模式。可能需要在每个新版本中不断复制模式。

[0250] 然而,相反,在一个实施例中,不是复制模式,而是允许模式属性本身被标记为“自版本……添加”属性或“自版本……弃用”属性。

[0251] 例如,一个实施例可以具有称为“红色”的资源类型的版本1属性。在版本2中,不再需要红色。在单个用户模式中,实施例将红色属性标记为版本2已弃用,但也可以向版本2添加3个新属性:A、B和C。对于这三个新属性,为版本2添加了一个标记。

[0252] 在运行时,当用户提出请求时,请求可以包括它们希望使用的模式版本(例如,版本1用户、版本2用户等)。在运行时,实施例评估模式(元数据驱动)以确定模式版本1包括什么、模式版本2包括什么等。

[0253] 在一个示例用例中:请求获取模式版本2的用户。请求基于添加和弃用的属性来获取用户。因此,请求将返回添加的属性,而不是返回已弃用的属性。相比之下,版本1不会添加或弃用任何标记。

[0254] 实施例完全是元数据驱动的。这允许通过同一资源管理器服务同时支持每个用户的多个版本。实施例可以支持零停机时间的模式改变。

[0255] 图15图示了根据一个实施例的自动模式版本控制。如图15所示,用户模式的版本1包括属性“名称”和“类型”(在1501)。版本2包括“name”和“costcenter”(在1502)。在此示例中,属性“Type”已被弃用,因为它不包括在版本2中。

[0256] 当发出获取用户的请求时,默认情况下,在一个实施例中总是检索模式的最新版本(即,图15中的版本2)。但是,请求有效负载可以请求版本1。在这种情况下,实施例将获得特定于该模式的数据子集。实施例也适用于高速缓存。

[0257] 大多数已知的解决方案对每个版本都有单独的模式定义。相反,实施例仅具有单

个模式定义和定义版本之间的改变的元数据。一个好处是支持零停机时间。

[0258] 如所公开的，实施例实现定义资源类型和关联模式的元数据。使用元数据来解析对多租户系统中的资源执行操作的请求，以确定与执行该操作的租户相关联的数据提供者。

[0259] 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。

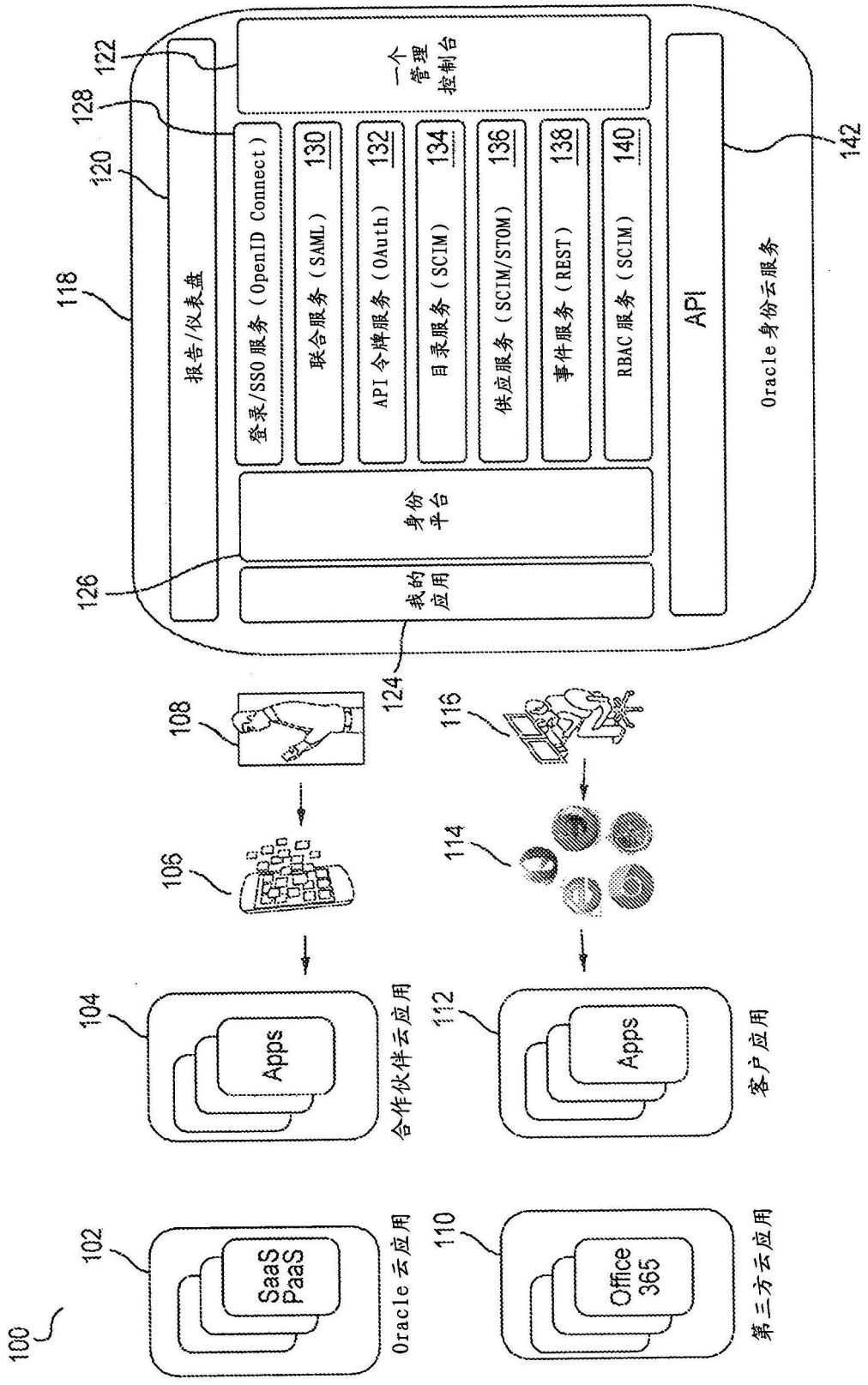

100
102
SaaS
PaaS
Oracle 云应用
104
Apps
合作伙伴云应用
106
108
110
Office
365
第三方云应用
112
Apps
客户应用
114
116
118
120
122
一个管理控制台
128
登录/SSO 服务（OpenID Connect）
130
联合服务（SAML）
132
API 令牌服务（OAuth）
134
目录服务（SCIM）
136
供应服务（SCIM/STOM）
138
事件服务（REST）
140
RBAC 服务（SCIM）
报告/仪表盘
身份平台
我的应用
126
124
API
142
Oracle 身份云服务

图1

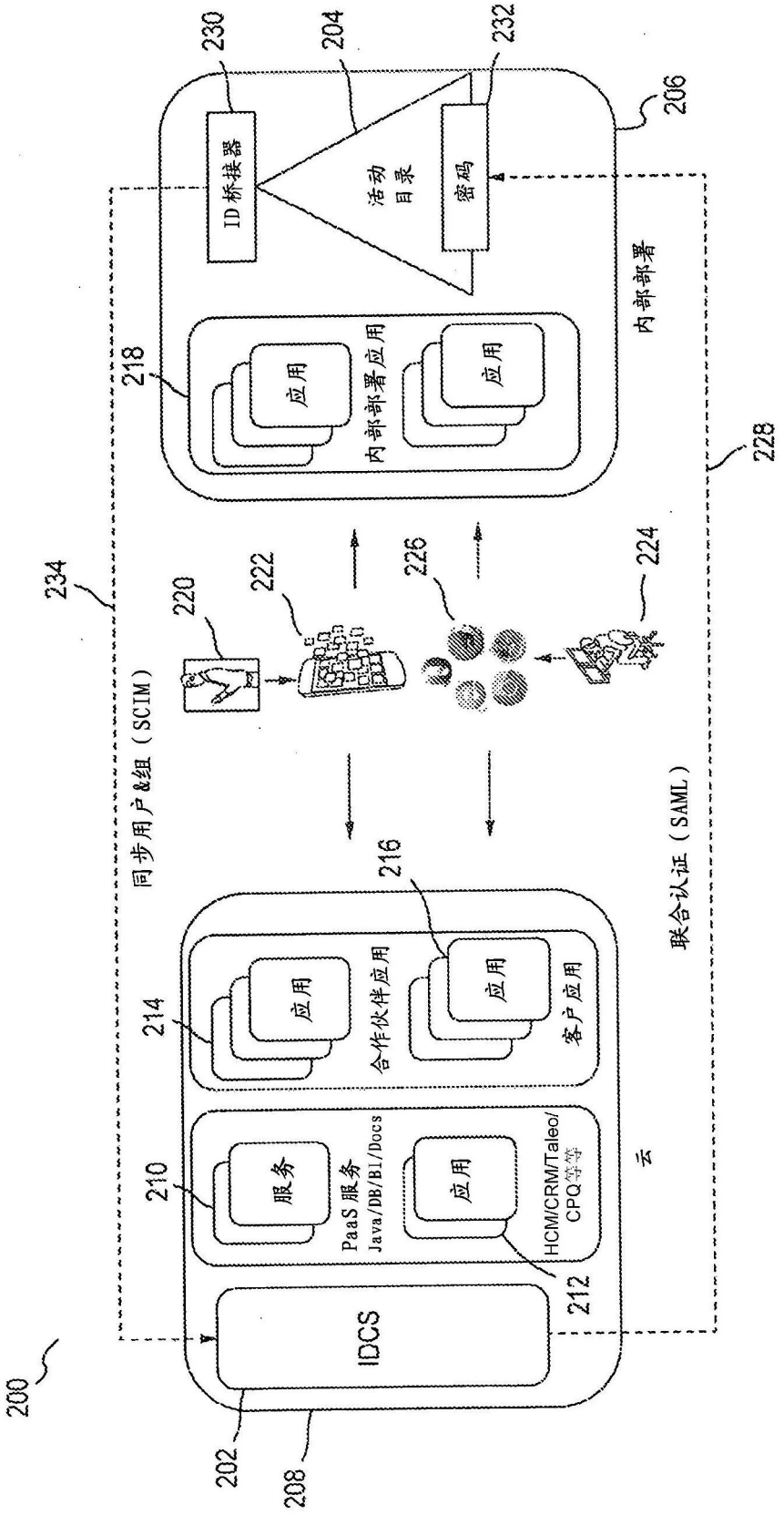

200
202
208
IDCS
212
PaaS 服务
Java/DB/BI/Docs
服务
210
应用
HCM/CRM/Taleo/
CPQ等等
214
合作伙伴应用
应用
216
客户应用
应用
云
同步用户&组（SCIM）
联合认证（SAML）
234
220
222
226
224
228
218
内部部署应用
应用
应用
内部部署
206
ID桥接器
230
活动
目录
204
密码
232

图2

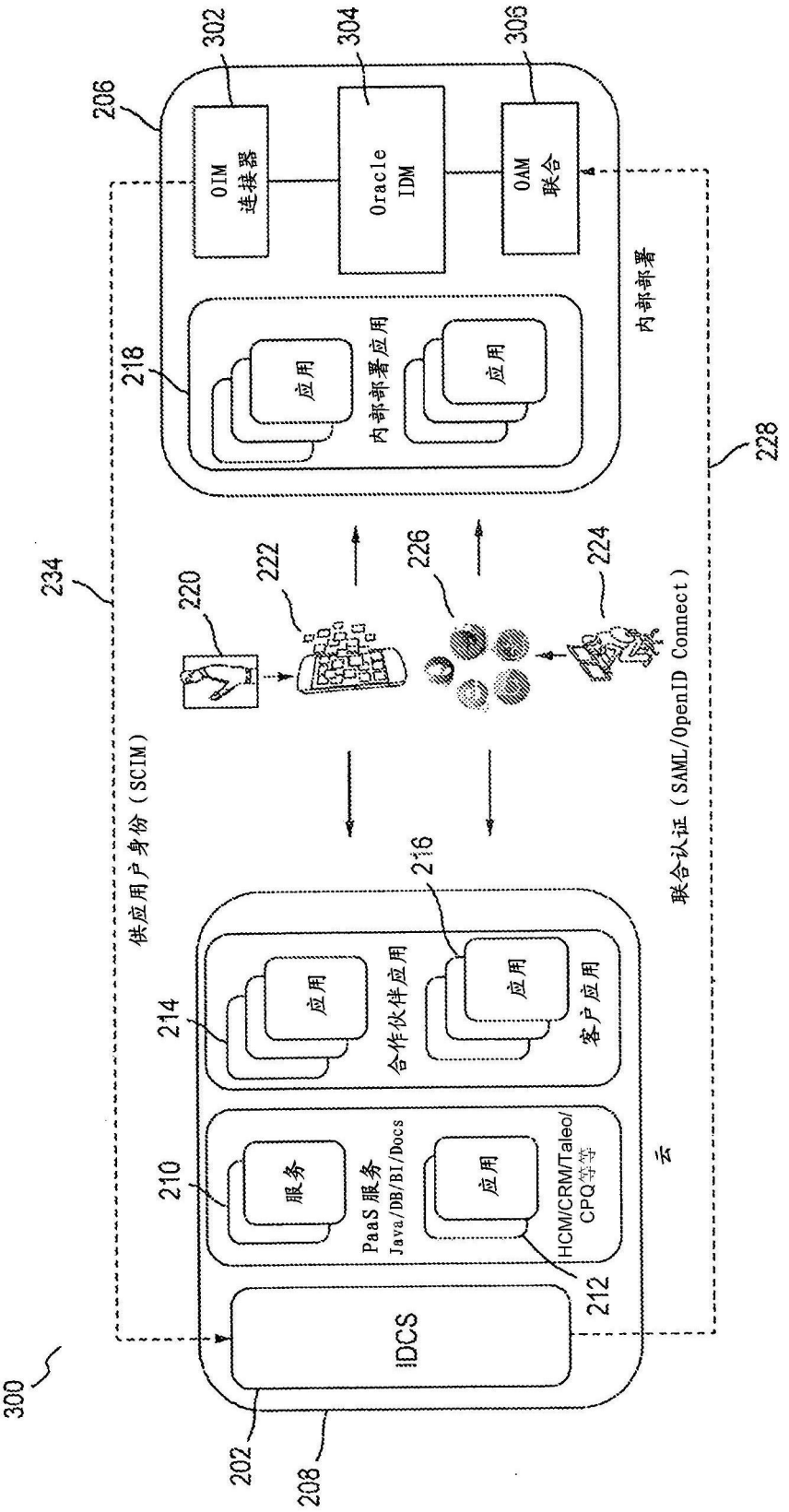

300
202
208
IDCS
212
云
HCM/CRM/Taleo/
CPQ等等
应用
PaaS 服务
Java/DB/BI/Docs
服务
210
客户应用
合作伙伴应用
应用
应用
214
216
供应用户身份（SCIM）
联合认证（SAML/OpenID Connect）
220
222
226
224
234
228
内部部署
内部部署应用
应用
应用
218
206
OIM
连接器
Oracle
IDM
OAM
联合
302
304
306

图3

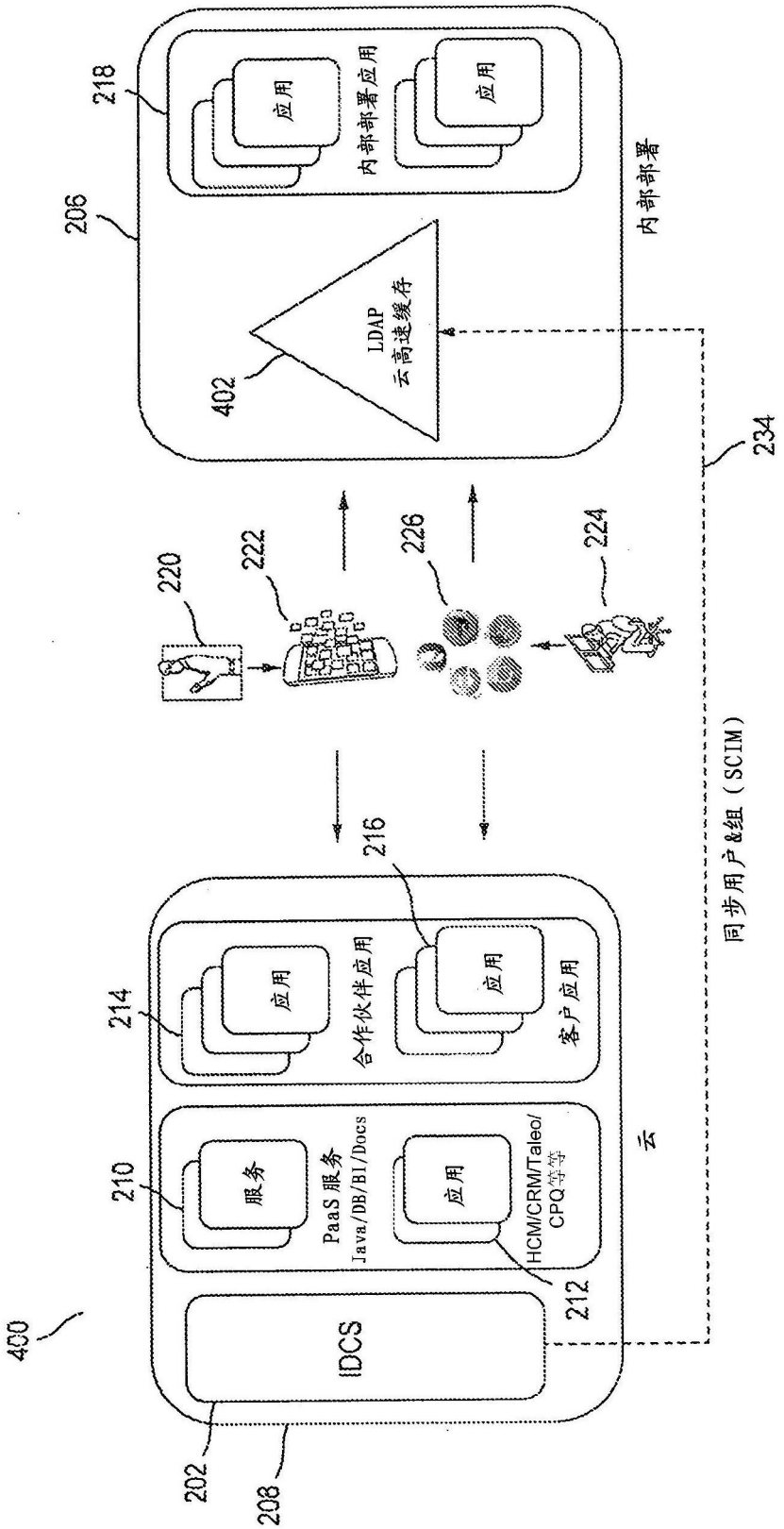

400
IDCS
202
208
212
云
服务
PaaS 服务
Java/DB/BI/Docs
应用
HCM/CRM/Taleo/
CPQ等等
210
应用
合作伙伴应用
应用
客户应用
214
216
206
218
应用
内部部署应用
应用
内部部署
LDAP
云高速缓存
402
220
222
226
224
234
同步用户&组（SCIM）

图4

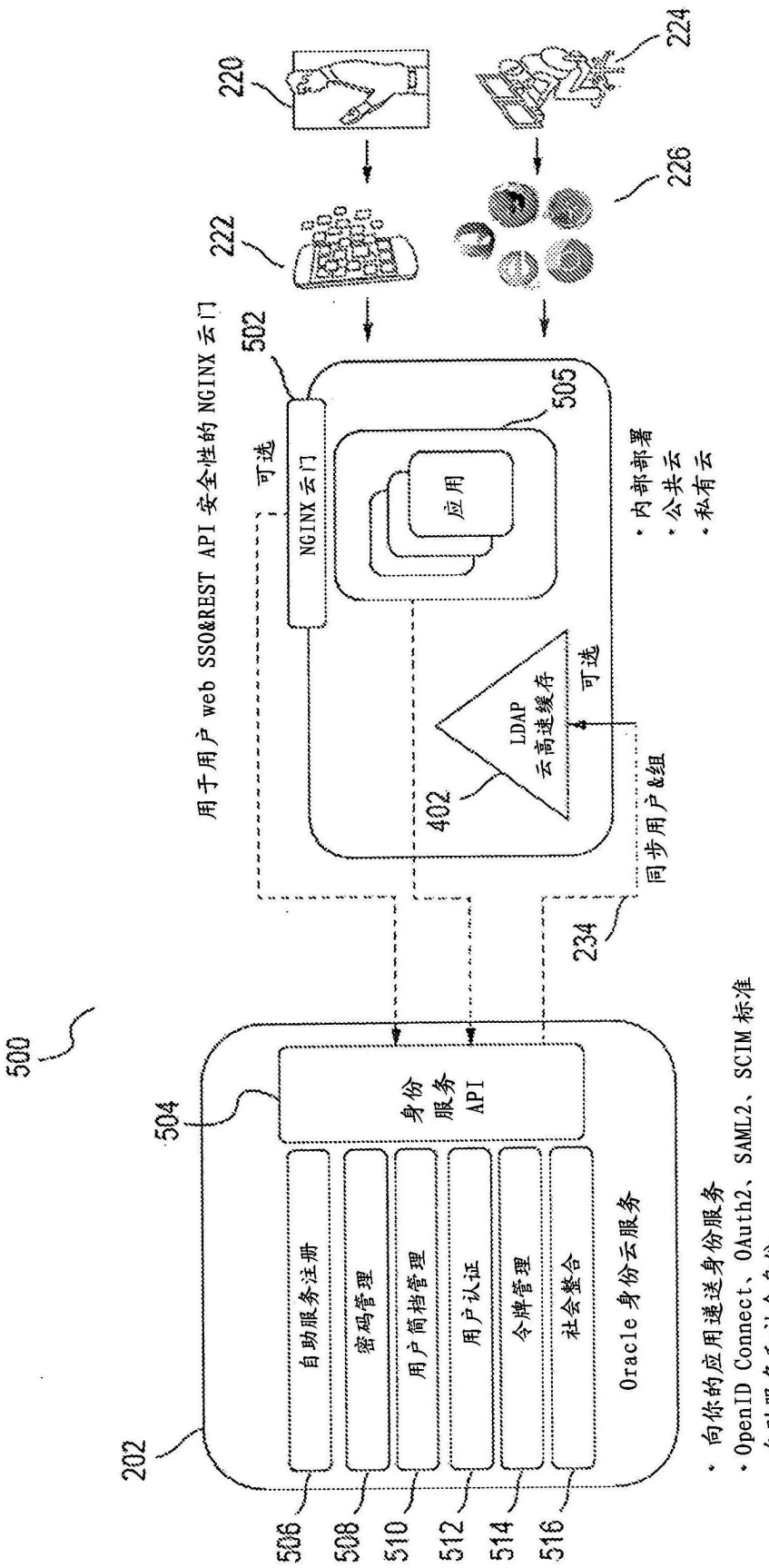

500
202
504
身份服务API
自助服务注册
密码管理
用户简档管理
用户认证
令牌管理
社会整合
Oracle 身份云服务
506
508
510
512
514
516
• 向你的应用递送身份服务
• OpenID Connect、OAuth2、SAML2、SCIM 标准
• 自助服务和社会身份
用于用户 web SSO&REST API 安全性的 NGINX 云门
502
可选
NGINX 云门
505
应用
402
LDAP
云高速缓存
可选
同步用户&组
234
220
222
224
226
• 内部部署
• 公共云
• 私有云

图5

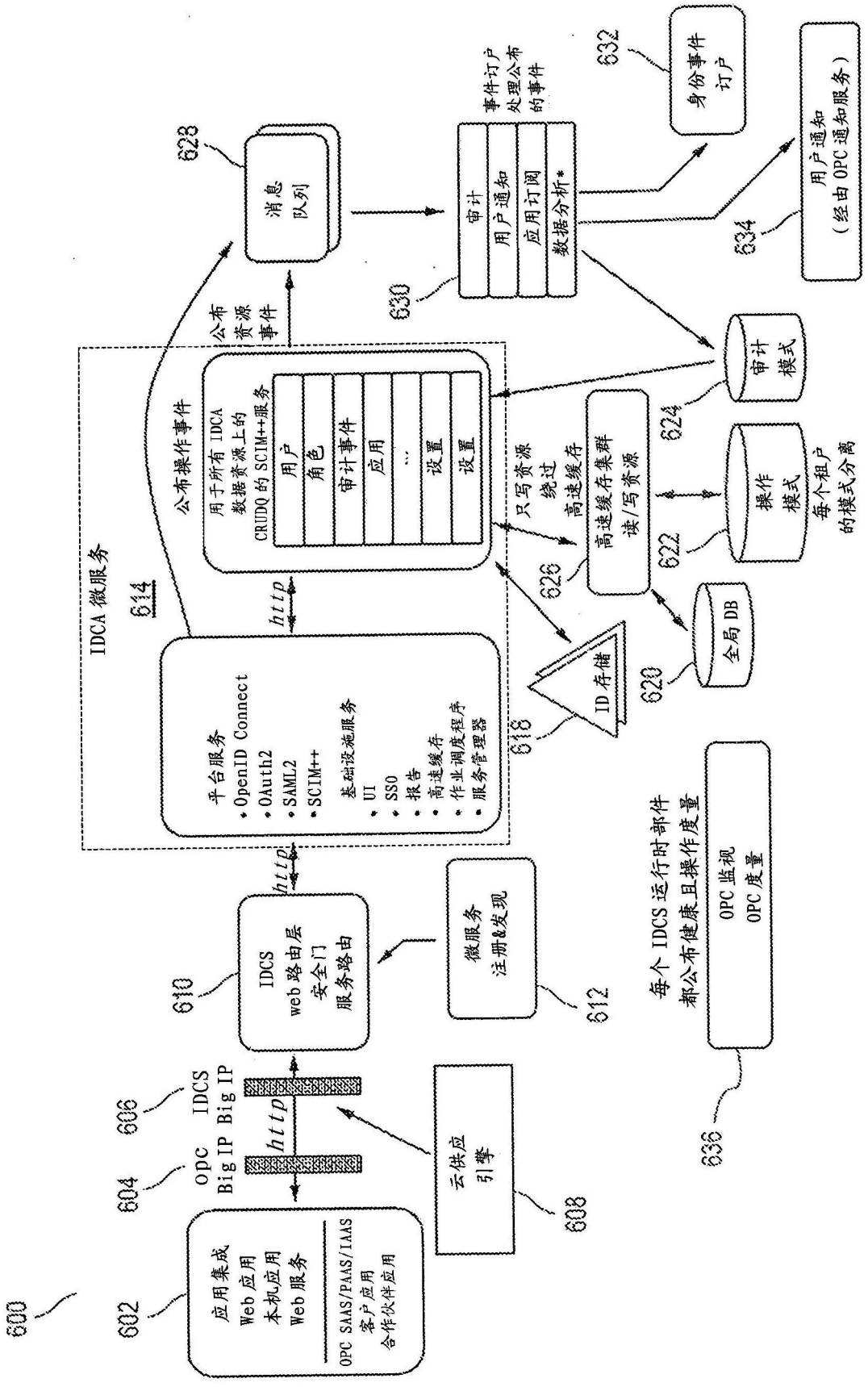

600
602
应用集成
Web 应用
本机应用
Web 服务
OPC SAAS/PAAS/IAAS
客户应用
合作伙伴应用
604
OPC
Big IP
606
IDCS
Big IP
http
云供应
引擎
608
610
IDCS
web 路由层
安全门
服务路由
612
微服务
注册&发现
614
IDCA 微服务
平台服务
• OpenID Connect
• OAuth2
• SAML2
• SCIM++
基础设施服务
• UI
• SSO
• 报告
• 高速缓存
• 作业调度程序
• 服务管理器
http
公布操作事件
用于所有 IDCA
数据资源上的
CRUDQ 的 SCIM++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
620
全局 DB
622
操作
模式
每个租户
的模式分离
624
审计
模式
626
只写资源
绕过
高速缓存
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

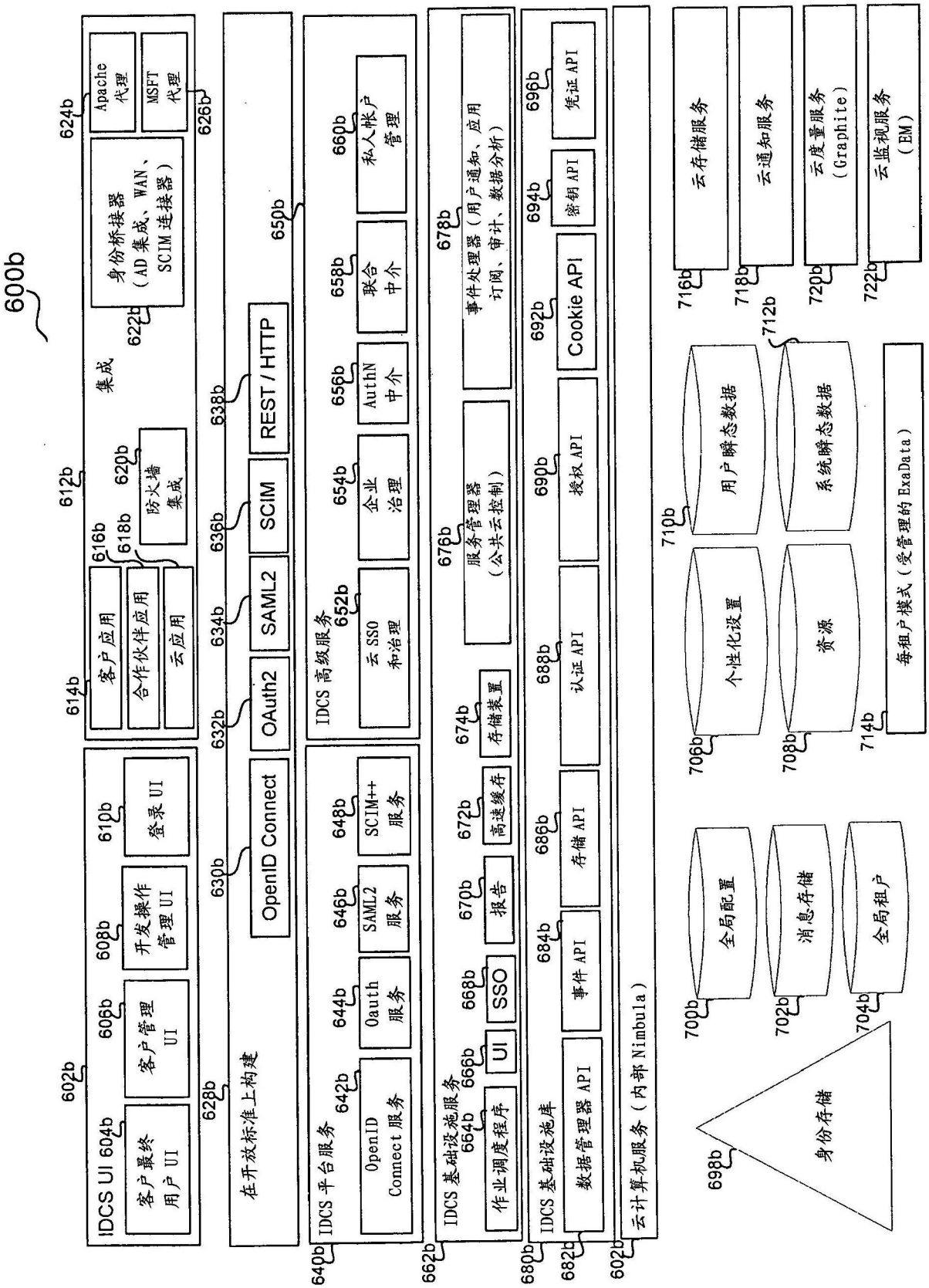

600b
602b
IDCS UI 604b
客户最终用户 UI
606b
客户管理 UI
608b
开发操作管理 UI
610b
登录 UI
612b
614b
客户应用
合作伙伴应用
云应用
616b
618b
620b
防火墙集成
集成
622b
身份桥接器（AD 集成、WAN、SCIM 连接器）
624b
Apache 代理
MSFT 代理
626b
628b
在开放标准上构建
630b
OpenID Connect
632b
OAuth2
634b
SAML2
636b
SCIM
638b
REST / HTTP
650b
640b
IDCS 平台服务
642b
OpenID Connect 服务
644b
Oauth 服务
646b
SAML2 服务
648b
SCIM++ 服务
652b
IDCS 高级服务
654b
云 SSO 和治理
656b
企业治理
658b
AuthN 中介
660b
联合中介
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
700b
全局配置
702b
消息存储
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

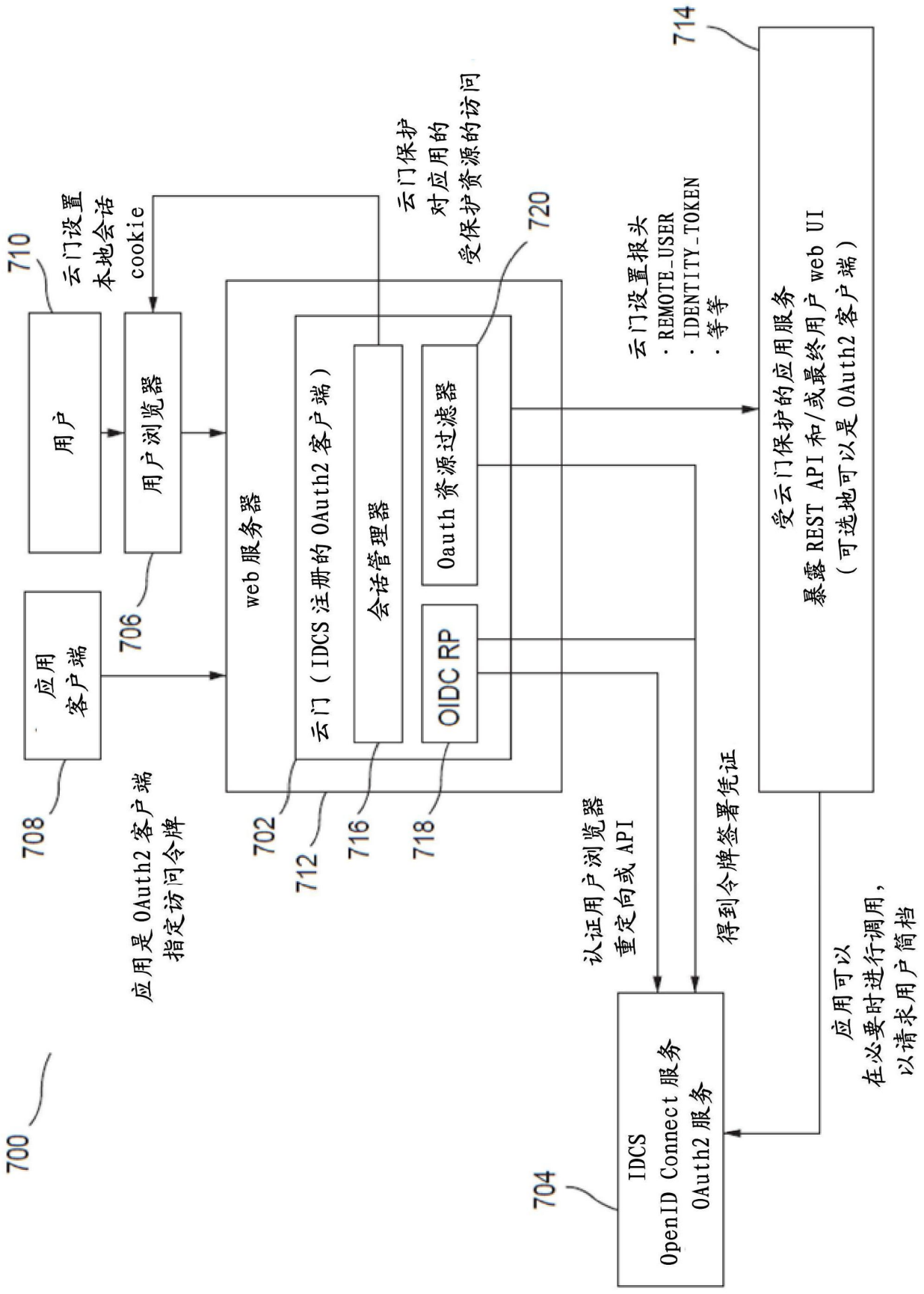

700
710
用户
云门设置
本地会话
cookie
用户浏览器
706
应用
客户端
708
应用是 OAuth2 客户端
指定访问令牌
web 服务器
云门（IDCS 注册的 OAuth2 客户端）
会话管理器
OIDC RP
Oauth 资源过滤器
720
云门保护
对应用的
受保护资源的访问
702
712
716
718
云门设置报头
· REMOTE_USER
· IDENTITY_TOKEN
· 等等
714
受云门保护的应用服务
暴露 REST API 和/或最终用户 web UI
（可选地可以是 OAuth2 客户端）
认证用户浏览器
重定向或 API
得到令牌签署凭证
应用可以
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

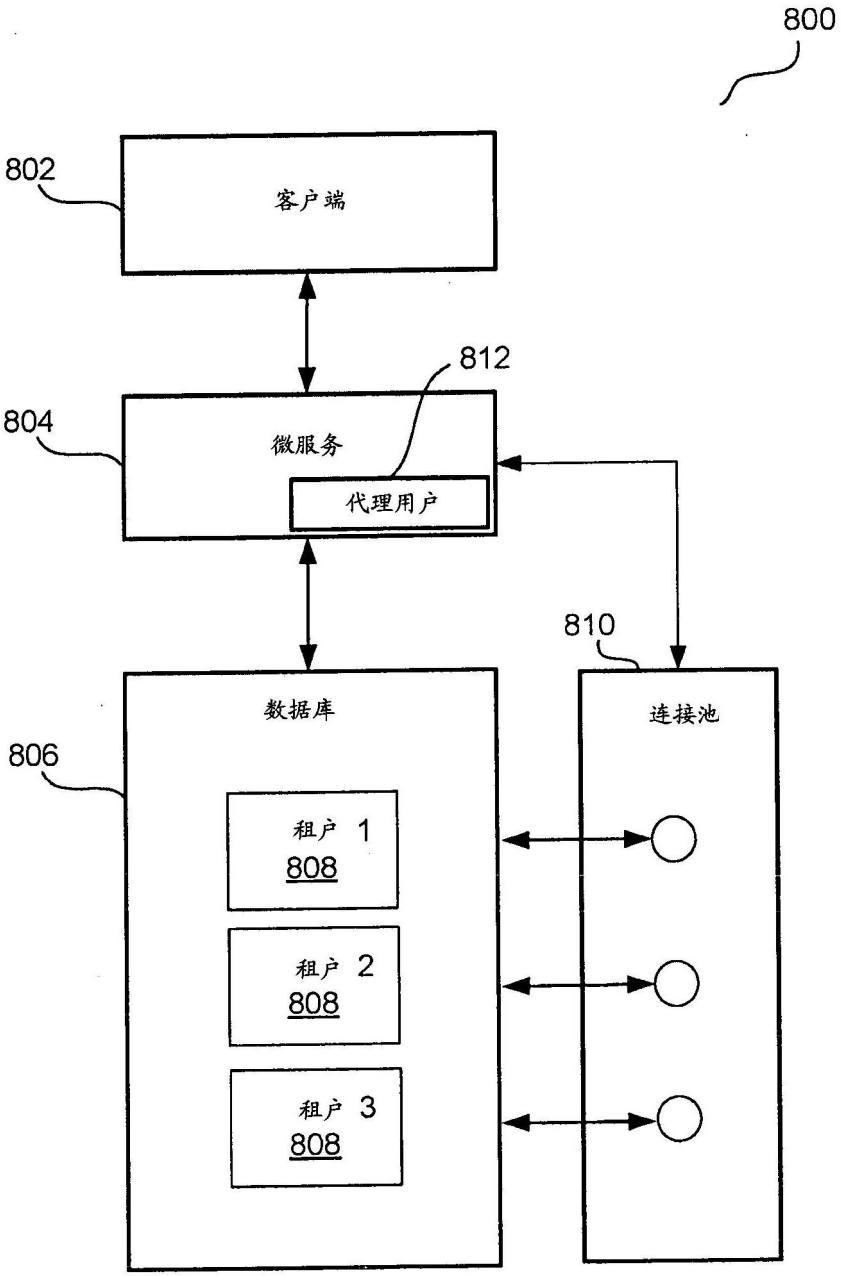

800
802
客户端
812
804
微服务
代理用户
810
数据库
连接池
806
租户 1
808
租户 2
808
租户 3
808

图8

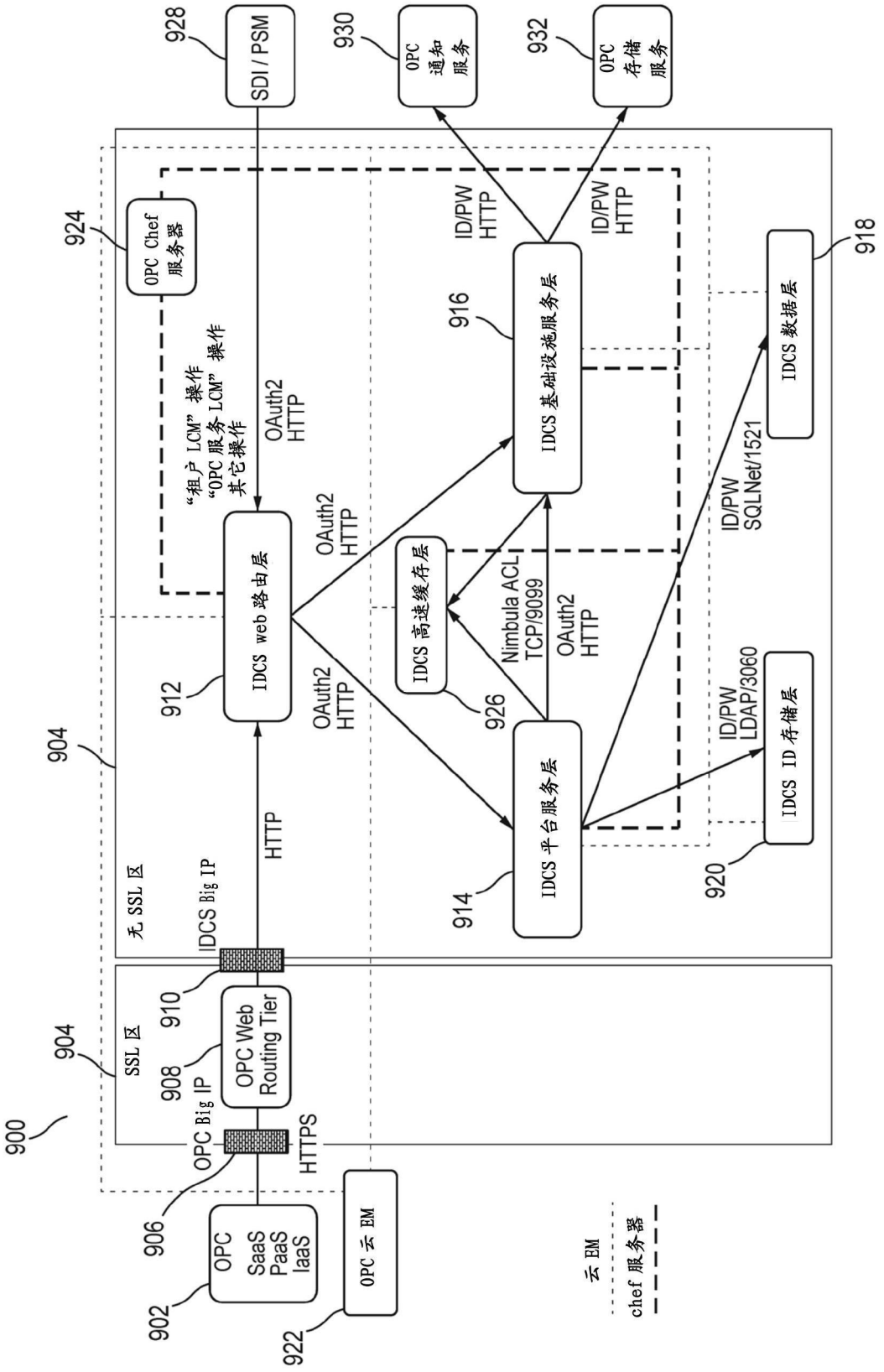

900
902
OPC SaaS PaaS IaaS
904
906
OPC Big IP
HTTPS
908
OPC Web Routing Tier
910
IDCS Big IP
SSL 区
904
无 SSL 区
912
IDCS web 路由层
HTTP
914
IDCS 平台服务层
916
IDCS 基础设施服务层
918
IDCS 数据层
920
IDCS ID 存储层
922
OPC 云 EM
924
OPC Chef 服务器
926
IDCS 高速缓存层
928
SDI / PSM
930
OPC 通知服务
932
OPC 存储服务
“租户 LCM”操作
“OPC 服务 LCM”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

图9

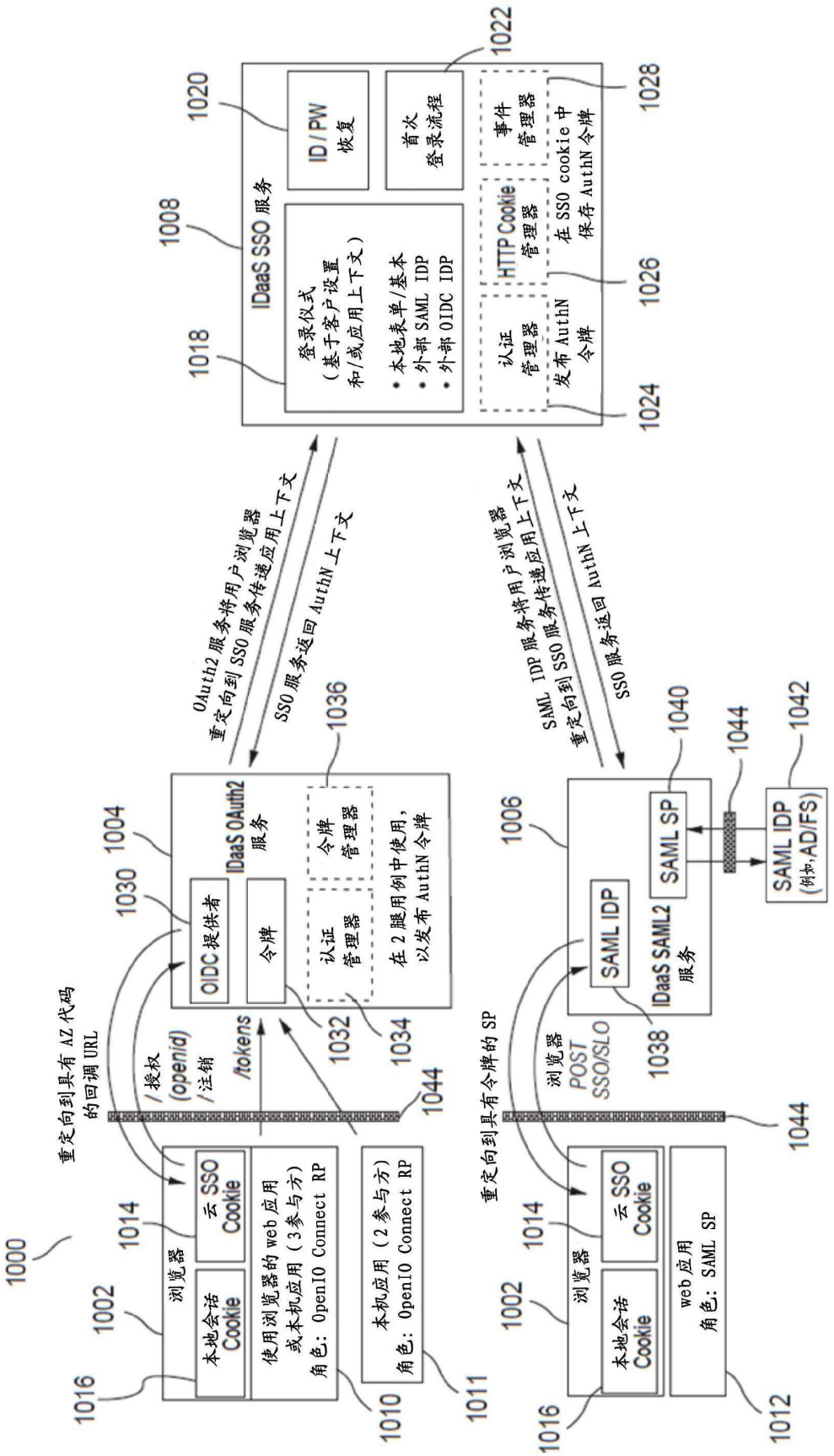

1000
1002
1004
1006
1008
1010
1011
1012
1014
1016
1018
1020
1022
1024
1026
1028
1030
1032
1034
1036
1038
1040
1042
1044
浏览器
本地会话 Cookie
云 SSO Cookie
使用浏览器的 web 应用或本机应用（3参与方）
角色：OpenIO Connect RP
本机应用（2参与方）
角色：OpenIO Connect RP
重定向到具有 AZ 代码的回调 URL
/授权 (openid) /注销
/tokens
IDaaS OAuth2 服务
OIDC 提供者
令牌
认证管理器
令牌管理器
在 2 腿用例中使用，以发布 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
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
认证管理器
发布 AuthN 令牌
HTTP Cookie 管理器
事件管理器
在 SSO cookie 中保存 AuthN 令牌
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)
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
web 应用
角色：SAML SP

图10

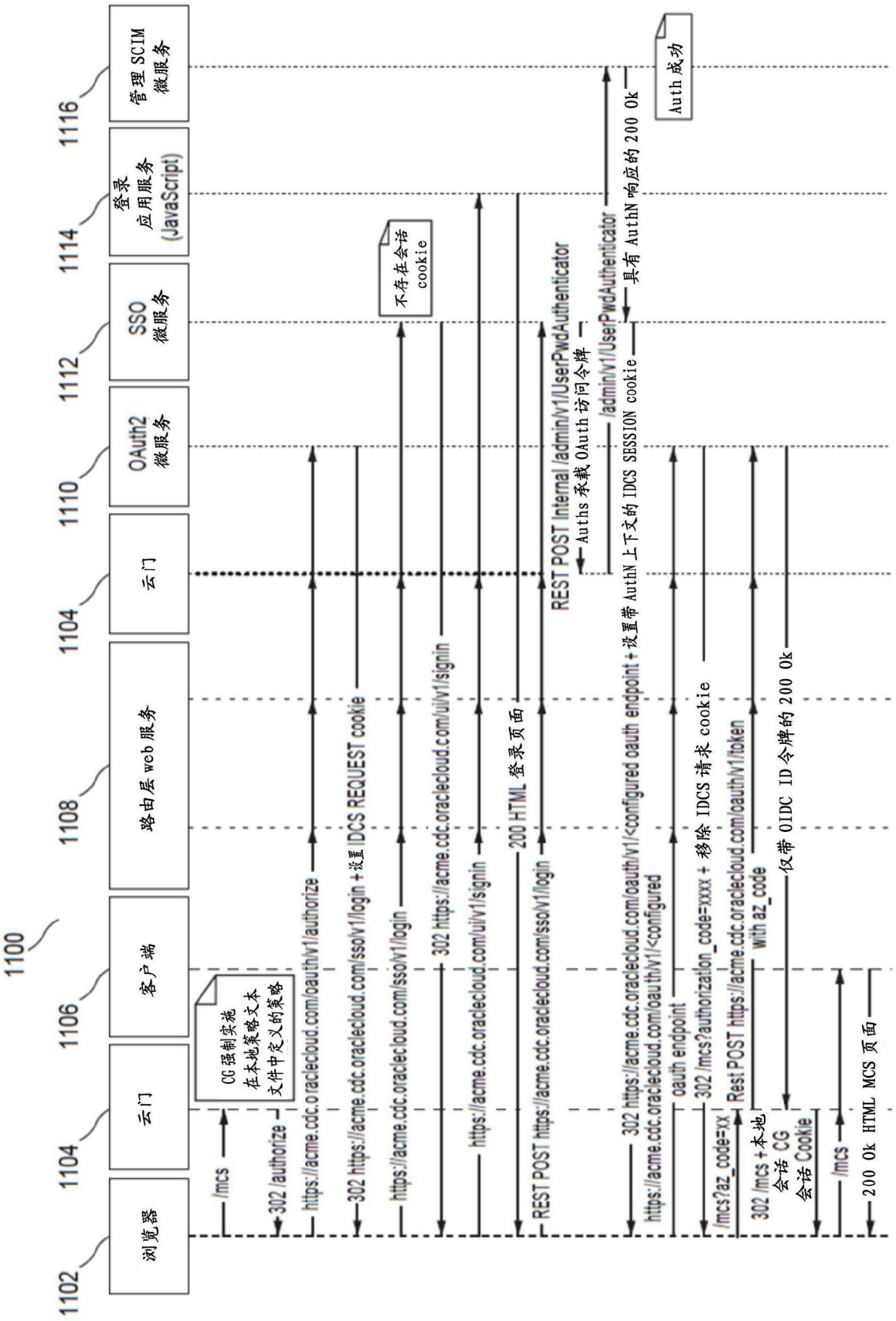

1100
1102
浏览器
1104
云门
1106
客户端
1108
路由层 web 服务
1104
云门
1110
OAuth2 微服务
1112
SSO 微服务
1114
登录 应用服务 (JavaScript)
1116
管理 SCIM 微服务
/mcs
CG强制实施 在本地策略文本 文件中定义的策略
302 /authorize
https://acme.cdc.oraclecloud.com/oauth/v1/authorize
302 https://acme.cdc.oraclecloud.com/sso/v1/login +设置 IDCS REQUEST cookie
https://acme.cdc.oraclecloud.com/sso/v1/login
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/<configured oauth endpoint +设置带 AuthN 上下文的 IDCS SESSION cookie
https://acme.cdc.oraclecloud.com/oauth/v1/<configured
oauth endpoint
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
with az_code
仅带 OIDC ID 令牌的 200 Ok
/mcs
200 Ok HTML MCS 页面

图11

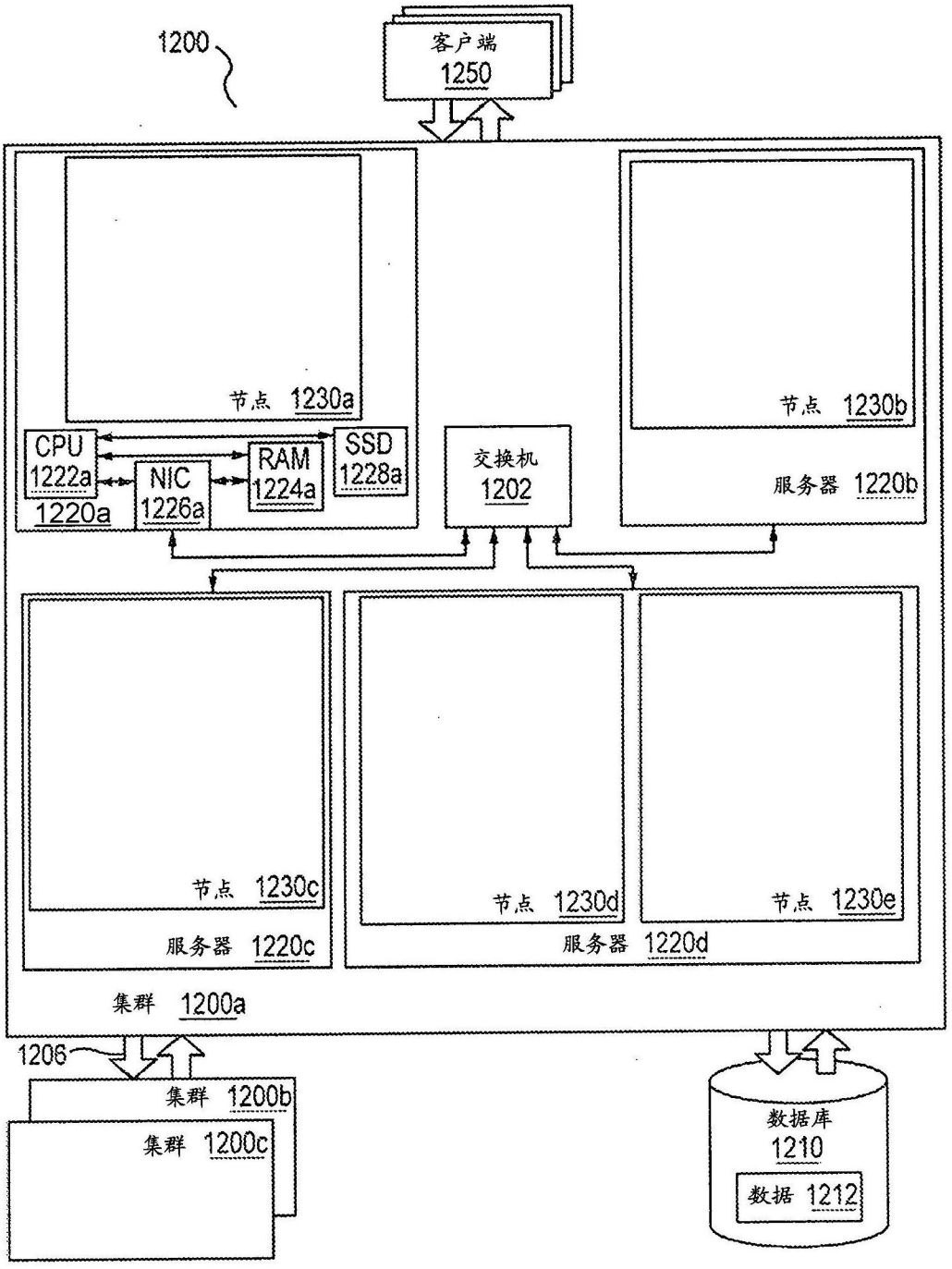

1200
客户端
1250
节点 1230a
CPU
1222a
NIC
1226a
RAM
1224a
SSD
1228a
1220a
交换机
1202
节点 1230b
服务器 1220b
节点 1230c
服务器 1220c
节点 1230d
节点 1230e
服务器 1220d
集群 1200a
1206
集群 1200b
集群 1200c
数据库
1210
数据 1212

图12

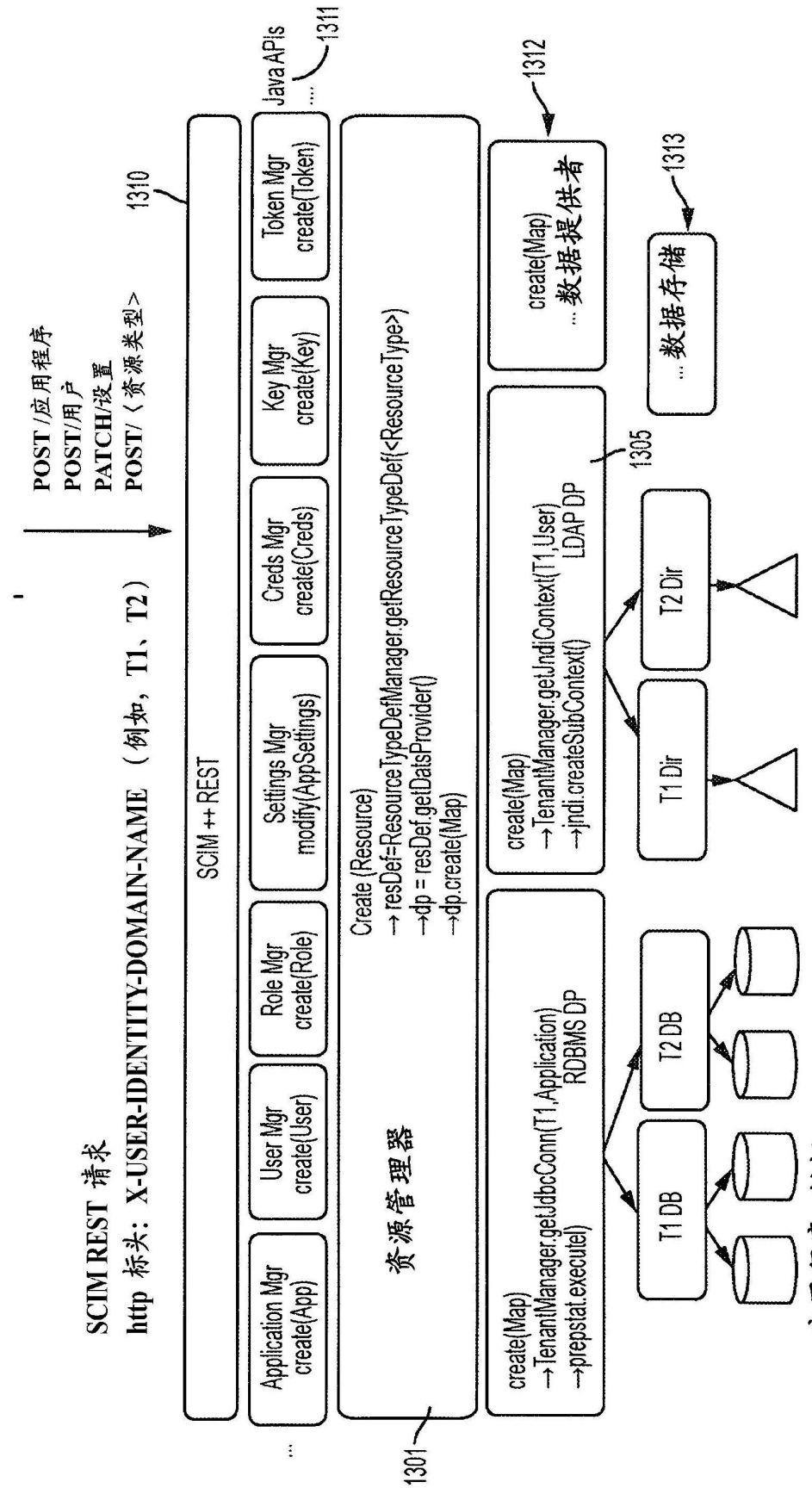

SCIM REST 请求
http 标头：X-USER-IDENTITY-DOMAIN-NAME （例如，T1、T2）
POST /应用程序
POST/用户
PATCH/设置
POST/〈资源类型>
1310
SCIM ++ REST
Application Mgr create(App)
User Mgr create(User)
Role Mgr create(Role)
Settings Mgr modify(AppSettings)
Creds Mgr create(Creds)
Key Mgr create(Key)
Token Mgr create(Token)
Java APIs
1311
资源管理器
Create (Resource)
→ resDef=ResourceTypeDefManager.getResourceTypeDef(<ResourceType>)
→dp = resDef.getDataProvider()
→dp.create(Map)
1301
create(Map)
→TenantManager.getJdbcConn(T1,Application)
→prepstat.execute()
RDBMS DP
create(Map)
→TenantManager.getLdapContext(T1,User)
→jndi.createSubContext()
LDAP DP
1305
create(Map)
...数据提供者
1312
T1 DB
T2 DB
T1 Dir
T2 Dir
...数据存储
1313
应用程序 凭据

图13

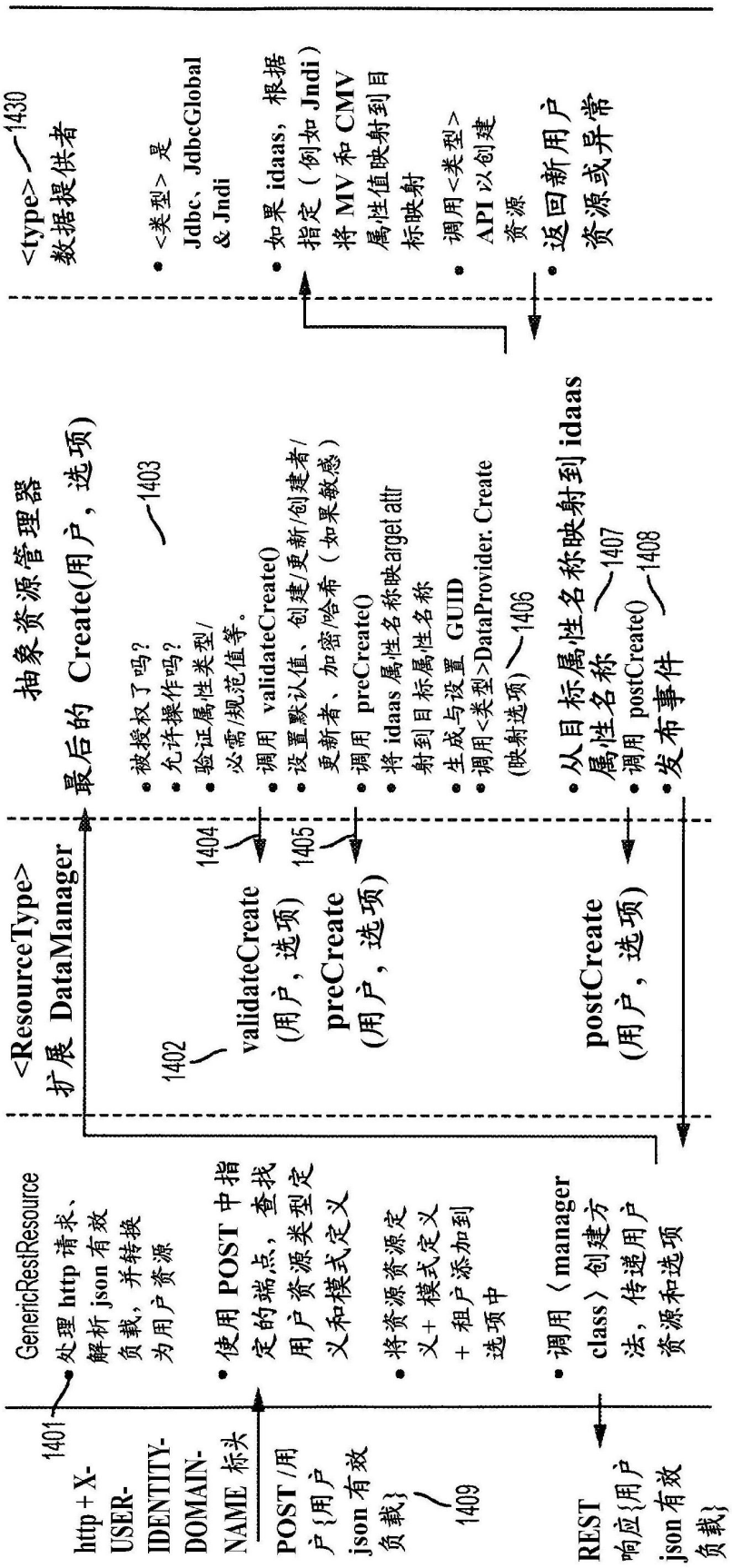

GenericResiResource
处理 http 请求、解析 json 有效负载，并转换为用户资源
使用 POST 中指定的端点，查找用户资源类型定义和模式定义
将资源资源定义 + 模式定义 + 租户添加到选项中
调用〈manager class〉创建方法，传递用户资源和选项
1401
http + X-USER-IDENTITY-DOMAIN-NAME 标头
POST /用户{用户 json 有效负载}
1409
REST 响应{用户 json 有效负载}
<ResourceType> 扩展 DataManager
1402
validateCreate (用户，选项)
1404
preCreate (用户，选项)
1405
postCreate (用户，选项)
抽象资源管理器
最后的 Create(用户，选项)
1403
被授权了吗?
允许操作吗?
验证属性类型/必需/规范值等。
调用 validateCreate()
设置默认值、创建/更新/创建者/更新者、加密/哈希（如果敏感）
调用 preCreate()
将 idaas 属性名称映射到目标属性名称
生成与设置 GUID
调用<类型>DataProvider. Create (映射选项)
1406
从目标属性名称映射到 idaas 属性名称
1407
调用 postCreate()
1408
发布事件
<type>
1430
数据提供者
<类型> 是 Jdbc、JdbcGlobal & Jndi
如果 idaas，根据指定（例如 Jndi）将 MV 和 CMV 属性值映射到目标映射
调用<类型> API 以创建资源
返回新用户资源或异常

图14

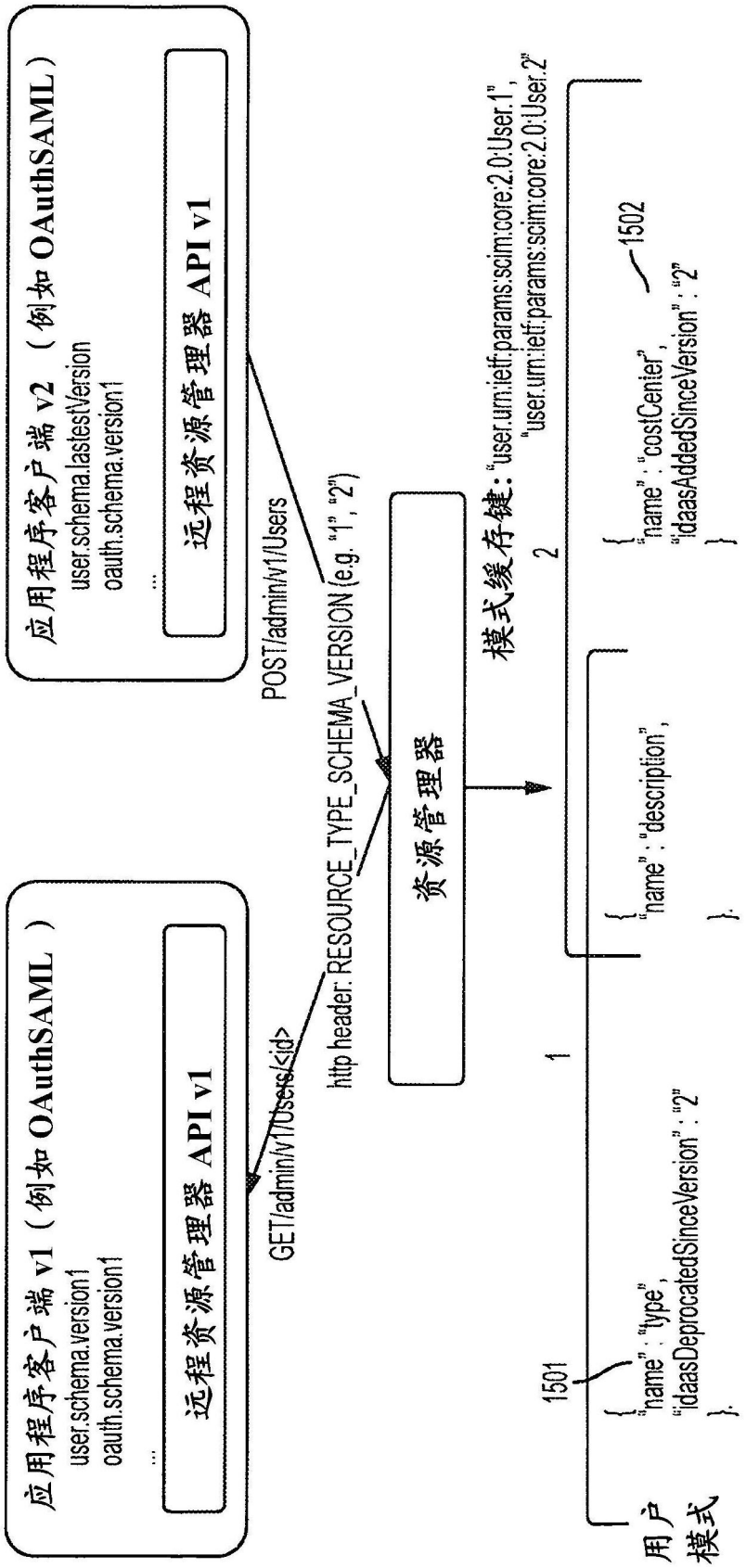

应用程序客户端 v1（例如 OAuthSAML）
user.schema.version1
oauth.schema.version1
...
远程资源管理器 API v1
应用程序客户端 v2（例如 OAuthSAML）
user.schema.lastestVersion
oauth.schema.version1
...
远程资源管理器 API v1
GET/admin/v1/Users/<id>
POST/admin/v1/Users
http header: RESOURCE_TYPE_SCHEMA_VERSION (e.g. "1", "2")
资源管理器
模式缓存键："user.urn:ietf:params:scim:core:2.0:User.1",
"user.urn:ietf:params:scim:core:2.0:User.2"
1
2
1501
1502
用户模式
{ "name": "type", "idaasDeprocatedSinceVersion": "2" }
{ "name": "description", }
{ "name": "costCenter", "idaasAddedSinceVersion": "2" }

图15