(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 101853433 B
(45)授权公告日 2016.08.03

(21)申请号 201010156040.7

(22)申请日 2003.07.10

(30)优先权数据
10/199,967 2002.07.19 US

(62)分案原申请数据
03821615.9 2003.07.10

(73)专利权人 开放创新网络有限责任公司
地址 美国纽约州

(72)发明人 拉古纳思·萨普拉姆
杰亚拉姆·R·卡西
托德·C·克劳斯
克里斯托弗·克拉尔
约瑟夫·桑菲利波

(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 钱大勇

(51)Int.Cl.
G06Q 10/00(2012.01)
H04L 29/06(2006.01)

(56)对比文件
WO 0059178 A1,2000.10.05,
WO 0011888 A2,2000.03.02,
CN 1233378 A,1999.10.27,

审查员 刘栩

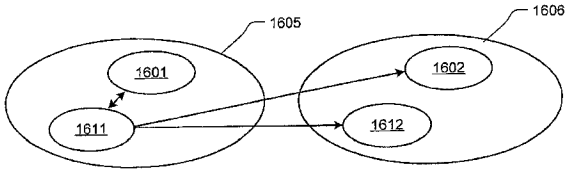
权利要求书3页 说明书19页 附图13页

(54)发明名称

建立用于发送文件的路由的方法

(57)摘要

本发明公开了一种设备和方法，该设备和方法能够以被信和可信方式、在位于具有互异接口的社区(1605、1606)之内的源(1611)和目的地(1612)之间路由电子文档。



CN 101853433 B

1.一种在源和目的地社区之间建立电子商务文档从源到目的地的安全路由的方法，包括：

确定所述源和目的地社区相互可接受的安全策略；

对于SAML服务验证所述源；

从所述SAML服务接收SAML断言；

根据封装协议来将所述SAML断言和电子商务文档打包到封装中；

通过被信连接器将该封装路由到所述目的地；

经由该路由将该封装发送到目的地。

2.如权利要求1所述的方法，还包含：

从所述源接收所述封装；

比较所述SAML断言的电子签名与所述源社区关于目的地社区注册的电子签名；和

根据目的地社区注册表来确定所述源社区被信任来发送电子商务文档。

3.如权利要求1所述的方法，其中，所述路由和发送步骤还包含：

通过至少一个中间社区中的被信连接器来路由所述封装；

在所述中间社区接收所述封装，并且比较所述SAML断言的电子签名与在所述中间社区注册的所述源社区的电子签名；

获取与所述中间社区相对应的中间SAML断言；

将所述中间SAML断言和所述封装打包在被注释的封装中；和

转发所述被注释的封装。

4.如权利要求3所述的方法，其中，在超过一个中间社区中重复权利要求3所述的接收、获取、打包和转发步骤。

5.如权利要求4所述的方法，其中，随同所述被注释的封装传送的所述中间SAML断言的数目为1。

6.如权利要求4所述的方法，还包含：

在所述目的地接收所述封装；

比较与所述中间社区相对应的中间SAML断言的电子签名与在所述目的地社区注册的所述中间社区的电子签名；

比较所述源社区的SAML断言的电子签名与所述源社区关于所述目的地社区注册的电子签名；和

根据目的地社区注册表来确定所述源社区被信任来发送电子商务文档。

7.一种在源和目的地社区之间建立电子商务文档从源到目的地的安全路由的方法，包括：

提供关于所述源或源社区可以使用的所支持的安全协议的描述；

从源接收封装，该封装包括电子商务文档和SAML断言；

比较所述SAML断言的电子签名与所述源社区关于所述目的地社区注册的电子签名；和

根据目的地社区注册表来确定所述源社区被信任来发送电子商务文档。

8.如权利要求7所述的方法，还包含：

经由至少一个中间社区从所述源接收所述封装，所述封装还包括所述中间社区的SAML断言；和

比较所述中间社区的SAML断言的电子签名与所述中间社区关于所述目的地社区注册的电子签名。

9.一种经由封装翻译网关在源和目的地社区之间建立电子商务文档从源到目的地的安全路由的方法,包括:

确定所述源和目的地社区相互可接受的安全策略;

对于第一安全服务验证所述源;

从所述第一安全服务接收第一断言;

通过被信网关计算经由被信连接器到所述目的地的路由;

其中,所述被信网关包含一个或多个服务,用于确认第一安全凭证、解密所述电子商务文档和将其重加密、将第一封装协议翻译成第二封装协议、以及根据所述第二封装协议转发第二安全凭证和所述电子商务文档;

根据所述第一封装协议经由到目的地的路由来转发所述第一断言和所述电子商务文档。

10.如权利要求9所述的方法,还包含:

从所述被信网关接收所述封装;

比较所述第二安全凭证与所述被信网关关于所述目的地社区注册的电子签名;和

根据目的地社区注册表来确定所述被信网关和所述源社区被信任来发送电子商务文档。

11.如权利要求9所述的方法,其中,验证所述源包含:

提供注册入口,该注册入口授权中间连接器对于所述第一安全服务验证所述源;和

从所述源调用所述中间连接器,以便代表所述源来对于所述第一安全服务进行验证。

12.如权利要求10所述的方法,其中,验证所述源包含:

提供注册入口,该注册入口授权中间连接器对于所述第一安全服务验证所述源;和

从所述源调用所述中间连接器,以便代表所述源来对于所述第一安全服务进行验证。

13.如权利要求9所述的方法,其中,所述路由包含在所述源和目的地之间的所述电子商务文档经过的被信连接器的列表。

14.如权利要求10所述的方法,其中,所述路由包含在所述源和目的地之间的所述电子商务文档经过的被信连接器的列表。

15.一种经由封装翻译网关在源和目的地社区之间建立电子商务文档从源到目的地的安全的路由的方法,包括:

提供关于所述源或源社区可以使用的所支持的安全协议的描述;

经由被信连接器从被信网关接收封装,所述封装包括来自所述源社区的电子商务文档和第二安全凭证;

其中,所述被信网关包含一个或多个服务,用于确认来自所述源社区的第一安全凭证、解密所述电子商务文档和将其重加密、将第一封装协议翻译成第二封装协议、以及根据所述第二封装协议转发所述第二安全凭证和所述电子商务文档;

比较所述第二安全凭证的电子签名与所述被信网关关于所述目的地社区注册的电子签名;和

根据目的地社区注册表来确定所述被信网关和所述源社区被信任来发送电子商务文

档。

建立用于发送文件的路由的方法

[0001] 本案是申请日为2003年7月10日、申请号为03821615.9、发明名称为“电子商务社区网络和社区内/间安全路由实现”的发明专利申请的分案申请。

[0002] 版权声明

[0003] 本专利文献的一部分公开包含受版权保护的资料。版权拥有者不反对任何人以它出现在专利和商标局专利文献或记录中那样对该专利文献或该专利公开进行传真复制，但是在其他方面是保留全部版权的。

[0004] 发明背景

[0005] 本发明涉及一种系统和协议，该系统和协议支持社区(community)的参加者之间以及加入到社区网络中的社区的参加者之间的电子商务文档的通信。具体地说，它涉及一种系统和协议，该系统和协议用于在参加者之间路由电子商务文档以及用于确保沿该路由的传输。

[0006] 事务对事务(business-to-business，B2B)和应用对应用(application-to-application，A2A)电子商务正在代替电子数据交换(EDI)的以前协议。由于事务努力改善它们关于B2B和A2A系统的效率，出现了大量不兼容的平台和竞争的标准。在兼容的标准中，存在着有待填补的不兼容。例如，工业上已经定义了简单Web服务是什么。涉及简单Web服务的标准包含UDDI、WSDL、XSDL和SOAP。然而，这些标准并非完全地意味着实际B2B和A2A电子商务的安全性、可靠性、易管理性以及灵活性(choreography)要求。基于处理流(process flow)的会话和合作web服务构成具有合作和复杂web服务的组成部分，它们不是任何综合性的或统一的标准的主体。

[0007] 大量工业倡导者倡导对可应用于B2B和A2A电子商务的标准进行扩展。choreography努力成果包含OASIS的ebXML/BPSS、IBM的WSFL和Microsoft的XLANG。会话努力成果包含OASIS的ebXML/TRP和Microsoft的WS-routing。主导的安全性努力成果是IBM和Microsoft的WS-security，还有OASIS的称为SAML补充安全性努力成果。对于可靠性，有Microsoft的建议、OASIS的ebXML/TRP和IBM的HTTPR。W3C正在解决所有这些领域中的标准化问题。关键的工业参与者已经形成了称为WSI的竞争集团。没有处理流的真正标准，尽管有很多处理流的专用实现(propietary implementation)。对于易管理性，中心地定义信息可能有用，该信息通过规定在电子商务中涉及的策略和实体的能力，授权web服务的互操作性。中心定义(central definition)的一个工业标准是ebXML CPP/CPA合同定义，它是由OASIS公布的。

[0008] 因此，有机会设计用于包含、统一以及填补很多相关Web服务标准之间的不兼容的方法和结构，这些相关Web服务标准包含SOAP、UDDI、ebXML、WSDL、WS-security、SAML和XSDL。总之，端对端服务以及实体之间电子商务文档的安全传递能力是有用的。

发明内容

[0009] 本发明包含一种设备和方法，用于建立社区网络、在具有互异接口的社区之间路由文档，并且以被信(trusted)和可信(trustworthy)方式来进行。本发明的具体方面在权

利要求书、说明书和附图中描述。

附图说明

[0010] 图1描述属于几个社区的几个实体。

[0011] 图2描述具有可选(alternate)通信信道的实体或连接器。

[0012] 图3描述用作在使用相似通信信道的两个连接器之间的集线器的连接器。

[0013] 图4和5描述社区网络中社区之间、跨越社区边界的通信。

[0014] 图6描述中间社区、通信链。

[0015] 图7描述使用中间社区进行翻译服务。

[0016] 图8是支持社区内路由的电子注册表数据的方框图。

[0017] 图9是更详细的电子注册表图。

[0018] 图10是支持社区间路由的电子注册表数据的方框图。

[0019] 图11和12示出了可以编译路由的XML格式。

[0020] 图13和14是社区内路由和社区间路由的高层示意图。

[0021] 图15描述社区内的安全性实现。

[0022] 图16描述社区间的安全性实现。

[0023] 图17描述授权(delegation)安全性服务。

[0024] 图18A和18B示出了将验证(anthentication)授权于网关。在图19A和19B中,将MML和SAML安全协议之间的转换扩展到社区之间的通信。

[0025] 图20到图25中呈现附加安全性使用的情况。图20是方框图(修改页)(91条)。图21示出了使用图20中的设计进行注册服务本地验证。图22中示出了该使用情况的变化,用于注册服务远程验证。图23和24示出了本地和远程验证。图25示出了获取文档服务预订的属性断言(assertion)。

[0026] 图26和27描述社区网络的建立。

具体实施方式

[0027] 以下参照附图进行详细说明。描述优选实施例是为了说明本发明,而不是为了限制其范围,本发明的范围由权利要求书限定。本领域技术人员将意识到关于描述的各种等效变化。

[0028] 图1示出了社区和社区网络。在这些社区中,社区保持本地注册表,该本地注册表包含如用户、公司、服务以及连接器等信息,其中连接器是社区的一部分。社区能够是市场、企业或子企业。社区能够属于一个或多个社区网络。一般地,社区和网络具有一些共同的商业兴趣。在一个或多个社区网络的成员社区之间具有互操作性。网络包含黄金市场网络1、贵金属市场网络2、私人网络3和全球贸易web网络4。在该示例中,黄金市场网络1和贵金属市场网络2包含在全球贸易web网络4中。贵金属市场网络2包含黄金和白银市场14、13。黄金市场消费者能够在白银市场13中进行白银交易,而白银市场消费者能够在黄金市场14中进行交易。一个社区,即PQR企业17,属于黄金市场网络1、私人网络3和全球贸易web网络4;另一个社区,即ABC大提供商18,属于私人网络3。在该示例中,XYZ黄金市场14是交易黄金的市场或社区。企业属于该社区。诸如自身形成社区的PQR企业17之类的企业属于黄金市场网络

1。这些社区是黄金市场网络1和全球贸易web网络4的一部分。小提供商15是黄金市场社区的一部分。其他企业16是社区，这些社区是黄金市场社区网络1的一部分。XYZ黄金市场14和其他黄金市场实体15-17之间的连接表明黄金市场需要企业(社区或其他)之间的所有通信，以便例如收集帐单和商业情报信息，这些企业处理将要通过XYZ黄金市场14路由的黄金交易。PQR企业17是一社区，该社区是黄金市场的一部分，并且还是包含提供商18的本地私人网络的一部分。小提供商15可以是单独的小提供商，它们自身不需要形成社区，而是在黄金市场的注册表中注册其例如用户、组织、服务和变换的元数据。另一方面，ABC大提供商18形成了它自身的私人网络，例如，因为它希望相对于普通公众访问而使元数据、内部事务部门系统和变换保持隐藏，因为它们是以相当高的成本开发出来的。因为PRQ17是ABC18的消费者，它加入私人网络3中。金融服务提供商DEF金融12想要给全球贸易web网络4中的任何一个提供金融服务，这就形成了它自身的社区，并且以全球贸易web根11注册。社区网络提供了社区的全球注册表。全球注册表允许查找社区和确定到该社区或到外部连接器的一个或多个路由，可以通过外部连接器来路由前往该社区的电子商务文档。从一个社区路由到另一个社区的文档，可以直接在两个社区的外部连接器之间路由，或通过一个或多个中间社区来非直接地路由。还能够定义并且在社区注册表中保持涉及社区的交易的商务和安全性规则。一般地，图1示出了实体和社区的混合成员(loyalty)，它创造了电子商务平台间的互操作性的推动力。

[0029] 连接器是与其他应用程序通信的应用程序的通用术语。连接器可以通过充当集线器、网关、外部端口、中央连接器等的其他连接器、基于定向进行通信或者基于对等(P2P)进行通信。基于P2P通信的连接器能够与使用相同传输/封装(envelope)协议的其他连接器通信。基于P2P通信的连接器可选地可以借助于执行交易服务的其他集线器连接器，当试图与不使用相同的传输/封装协议的连接器通信时，基于定向通信的连接器根据路由规则、通过集线器连接器通信。连接器间的路由规则能够映射在定向图中，支持一个或多个传输/封装协议的一个或多个集线器和轮辐拓扑(spoke topology)。在一个或多个层(tier)中，集线器和轮辐拓扑将通信以轮辐形式导向集线器。这便利于集中服务，例如帐单、商务情报收集、跟踪、审计、记帐或其他。多个集线器和轮辐机构可以覆盖相同的连接器，来支持不同的传输/封装协议和技术，如图2所示。例如，可以需要更强健的集线器和轮辐机构来使用Sonic作为传输技术，而不使用HTTP或HTTPS。可选地，可以根据源和目的地是否是相同社区的一部分来进行通信路由。在子社区(它可以包含整个社区)之内，可能不需要集中功能，并且允许在连接器间进行P2P通信，然而，当与其他子社区中的目的地通信时，引导这些连接器与父连接器进行通信。

[0030] 连接器可以分为简单连接器(有时简单地称为连接器)、集线器(有时称为网关或路由器)或者中心连接器。可替换地，它们可以被以功能性进行描述。简单连接器用于经由集线器连接器通信，当允许它们在相同子社区的连接器间进行P2P通信时除外。所谓的集线器由明确地指向或连接到它们的连接器使用。集线器可以提供多种功能，因此可以在从源到目的地的路由中出现超过一次。集线器转发电子商务文档或消息。集线器还可以在支持公共封装协议的传输协议之间进行翻译。例如，集线器可以翻译封装协议，并且当传输时而不是接收时，还可以实现不同的传输协议。中心连接器是集线器的特殊情况，它们能够由没有明确地指向或连接到它们的连接器使用。中心连接器是有用的，例如，当根据路由规则从

源开始经过连接器不会通向支持由目的地使用的传输/封装协议的任何集线器时，中心连接器用于执行翻译功能。

[0031] 图2示出了三个连接器：简单连接器201和一对集线器202-203，它们中的一个已被称为网关203。连接器201受路由规则约束，当传输/封装协议是SOAP/HTTP 204时，将通信导向集线器202，而当使用MML/Sonic 205时，将通信导向集线器203。实际上，子201具有两个父202-203。相关的父依赖于所使用的通信协议204-205。所示出的对于通信协议的定向路由的覆盖还可以由传输安全性协议进一步覆盖，使得通过父的路由依赖于传输/封装/传输安全性协议三者组成的一组。可替换地，正如本文所使用，传输/封装协议可以指支持封装和传输两者的简单的统一协议。当前，传输和封装协议是不同的，但是使用术语传输/封装协议旨在包含将来的任何统一协议。

[0032] 图3-7示出了源A和目的地B的不同关系。在图3中，源301和目的地302处于相同的社区中。它们两者都通过路由规则指向集线器303，该路由规则通过MML/Sonic 304、305将通信导向集线器303。在图4中，源401和目的地402处于由社区边界403分隔开的不同社区中。当使用SOAP/HTTPS协议406、407时，源和目的地都分别指向通过集线器404和405的通信。因为社区之间的通信经由外部连接器，所以该示例中的集线器还是注册为可访问其他社区的外部连接器。在图5中，源501和目的地502再次被社区边界503分隔开来。源501经由ebXML/HTTPS协议507与集线器504通信。可以将集线器505、506看作集线器504和连接器502没有明确地指向或连接到的中心连接器。假设该目的地502使用MML/HTTPS协议。集线器504没有翻译能力，但是集线器505、506有翻译能力。集线器的公共协议是SOAP/某传输协议。各集线器将ebXML/HTTPS翻译成SOAP/某传输协议(505)，并且将MML/HTTPS翻译成相同的SOAP/某传输协议(506)。如果集线器执行两种翻译功能，则它们可以在从源到目的地的该路由中出现两次。正如所示，集线器505、506还是外部连接器，因为它们穿过社区边界503进行通信。如果没有社区边界503，则集线器就不是外部连接器。

[0033] 在图6-7中，引入中间社区来提供服务。这些服务是网关和商务情报数据收集。中间社区一般是市场，这些市场使用各种协议、通过网关、提供到企业的连接。它们还能够充当企业的被信中介，以便彼此交互。它们还能够给它们的消费者提供商务情报数据。在图6中，服务桥接网络中的两个社区。如下的实现是有可能的，即该实现支持属于多个网络的中介，并且充当允许桥接的两个网络的成员的网络之间的桥接器。例如，在图6中，源和目的地社区601和602可以不属于相同的社区网络。假设社区603属于与源601共同的一个社区网络，以及属于与目的地602共同的另一个社区网络，则社区603能够在源和目的地的通信中充当被信中介。在图示的这种情况下，社区边界608还是社区网络边界。这是简单的情况，因为由源601和目的地602使用的协议606、607相同。任何集线器603-605都不需要翻译。在图7中，需要翻译，因为连接器701使用ebXML/HTTPS协议706，而连接器702使用MML/Sonic协议707。在该图中，集线器711、712和713属于由社区边界708从源社区和目的地社区分隔开的中间社区。全部三个集线器都涉及从源到目的地的传输。集线器711是外部连接器，它从集线器704接收电子商务文档。第一次翻译是由集线器712执行的，从ebXML/HTTPS翻译到soap/HTTPS。这仍然不是目的地所需的协议组合，所以文档被转发给集线器713。集线器713执行从soap/HTTPS到MML/sonic的进一步翻译，MML/sonic是目的地的协议组合。文档被转发到集线器705。

[0034] 如图3-7所示的传递消息所需的路由，由包含路由信息和路由算法的注册表来支持。图8是包含路由信息的部分注册表的简化示意图。连接器801是该数据结构的中心特征。连接器具有例如封装翻译、传输翻译、外部可视能力、对等路由、子社区中的成员资格以及到相同子社区中的其他连接器的对等路由之类的能力802。因此，连接器801和能力802之间的关系811支持零个或多个能力。一个或多个链接803将连接器801连接到协议804和其他连接器。连接器801和特定传输/封装协议804之间的关系816支持一个或多个协议。从连接器801通过812、链接803通过815到协议804，关系是一对一的。也就是说，从连接器801到特定传输/封装协议804存在不超过一个的出站链接813，除了在图8中未示出的、根据安全考虑而进一步区分的传输协议的上述情况以外。出站链接813和入站链接814与路由规则相对应。而且，出站链接813表达了如下路由规则，即根据特定传输/封装协议815、804而被传送的消息需要被转发到另一个连接器。链接803与子和父连接器801相关联。入站链接814将连接器801标识为由子连接器根据特定传输/封装协议通信的目的地，它由应用于该子连接器的路由规则来表达。对于传输/封装协议804，存在传输信息806和封装格式信息805两者。至于使对象结构正规化的问题，该传输和封装信息806、805可以被链接818、817到协议对804，而不是相同对象的一部分。在上述情况下，当路由依赖于传输/封装/安全性三者组成的一组时，可以进一步将所谓的信道对象引入到连接器801和协议804之间，以便进一步对数据结构正规化。

[0035] 图9提供了描述了连接器的部分注册表的替换图。连接器901具有各种属性。它具有名称，该名称可以是社区名称和单独的连接器名称的结合。它具有描述和统一资源指示符(URI)。标志指示该连接器是否是中心连接器或外部连接器。外部连接器是用户定义的连接器。在社区之内，子社区中的成员资格反映在属性peerToPeerGroup中。该属性可以是包含管理范围或子社区的名称的字符串。连接器901的能力包含传输翻译922、封装翻译923和路由器动作924。连接器901可以具有超过一个的传输翻译能力922。在当前实施例中，传输与特定封装协议相关联。假设翻译在传输1到传输2之间是双向的。标志指示两个传输协议是否是软件实现自身拥有的。一种软件实现提供HTTP、HTTPS和sonic传输协议的自动(native)传输支持。其他传输协议，例如FTP，可以是用户实现的。连接器901还可以具有超过一个的封装翻译能力923。再次假设翻译是双向的。连接器还可以具有路由器能力924。路由器能力指集线器连接器转发从其他连接器接收到的消息的能力。在本实施例中，路由与特定封装协议相关联。连接器901还与传输/封装协议904相关联。支持特定封装协议和协议版本的传输规范。可以使用具有特定封装协议的各种传输类型。物理地址与特定传输类型相关联。可选地，传输安全性可以与特定传输类型相关联。传输规范904可以反映传输/封装对或传输/封装/安全性三者。标志根据传输规范来指示是否可以忽略路由规则，以及是否可以基于对等原则、在该连接器与其为相同子社区的成员的其他连接器之间引导通信。路由规则对应于允许的路由925。图9中示出的注册机构能够通过在连接器901和传输规范904之间引入信道对象而被正常化，以便利用封装协议或传输/封装协议来组成传输规范。

[0036] 图10提供了支持社区之间的路由的注册表的部分的高层示意图。目标1001是源试图到达的目的地。该目标可以是最终目的地，或者是接近该目的地的负责转发到该目的地的点。目标与社区相关联，社区提供地址，例如URL，能够在该地址处访问社区注册表。当前实施例支持一个中间社区1003的指定，通过中间社区1003将消息转发到目标1001。目标与

目的地连接器1004以及一个或多个传输/封装协议1005相关联。如图8所示，传输/封装协议与封装格式1006和传输1007相关联。

[0037] 这里是使用该方案的XML文件示例，其中简单集线器和轮辐拓扑中有两个应用。

```
<?xml version="1.0" encoding="UTF-8"?>
<Registry xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\design\routing\registry.xsd">
  <Connector uuid="A">
    <TransportSpec uuid="A01">
      <Parent>Hub01</Parent>
      <EnvelopeProtocol version="C1">SOAP</EnvelopeProtocol>
      <TransportType>Sonic</TransportType>
      <PhysicalAddress>String</PhysicalAddress>
    </TransportSpec>
  </Connector>
  <Connector uuid="Hub">
    <TransportSpec uuid="Hub01">
      <Parent>None</Parent>
      <Child>A01</Child>
      <Child>B01</Child>
      <EnvelopeProtocol version="C1">SOAP</EnvelopeProtocol>
```

[0038]

```
      <TransportType>Sonic</TransportType>
      <PhysicalAddress>String</PhysicalAddress>
    </TransportSpec>
    <Capability>
      <Hub>
        <EnvelopeProtocol version="C1">SOAP</EnvelopeProtocol>
      </Hub>
    </Capability>
  </Connector>
  <Connector uuid="B">
    <TransportSpec uuid="B01">
      <Parent>Hub01</Parent>
      <EnvelopeProtocol version="C1">SOAP</EnvelopeProtocol>
      <TransportType>Sonic</TransportType>
      <PhysicalAddress>String</PhysicalAddress>
    </TransportSpec>
  </Connector>
</Registry>
```

[0039] 该注册表数据对应于图3中的变化，其中源和目的地连接器301、302利用SOAP/Sonic协议304、305。在该示例入口中，集线器303名称为“集线器01”。数据一般地遵守图9的结构。

[0040] 路由方框的简单格式能够用于在社区内路由和在社区间路由两者。图11和图12示出了能使用XML来表达路由的格式，随后是示例。在图11和图12中，路由1101与具有两个或多个连接器1103/1203的1102相关联。连接器1203与复杂数据类型1204相关联，复杂数据类型1204包含名称1205、封装类型1206、本地或外部传输标志1207、连接器功能指定1208以及传输协议1209。传输协议1209还与传输地址相关联，该图中没有示出。路由1101定义了从源到目的地将要越过的连接器的列表。连接器1103/1203提供沿路由的单一功能。名称1205可以是惟一名称，例如发出授权前缀/连接器类型/社区名称/本地名称的结合。封装类型1206是到达该连接器时的封装类型，例如SOAP、ebXML或MML。“Is Native”标志1207指示该传输类型是否是软件系统固有的或者是否被支持为用户实现的扩展。连接器功能1208标识在该

节点将要执行的功能。传输1209标识用于到达该节点的传输。该节点的传输/封装协议对应于传输1209和封装类型1206的组合。以下是一般遵守该数据结构的XML代码示例：

```
<?xml version="1.0" encoding="UTF-8"?>
<Route xmlns="http://commerceone.com/wse/routing"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://commerceone.com/wse/routing
D:\design\routing\route-block.xsd">
   <Connector>
      <Name>
         BUY:C:buySpice:nutmeg
      </Name>
      <EnvelopeType>ebXML</EnvelopeType>
      <isNative>true</isNative>
      <ConnectorFunction>service-send</ConnectorFunction>
      <Transport type="HTTPS" mode="sync" reliable="false">
         <Address>http://buyer.com/buy/nutmeg</Address>
      </Transport>
   </Connector>
   <Connector>
      <Name>
         BUY:C:buySpice:gw1
      </Name>
      <EnvelopeType>ebXML</EnvelopeType>
      <isNative>true</isNative>
      <ConnectorFunction>envelope-gateway</ConnectorFunction>
      <Transport type="HTTPS" mode="sync" reliable="false">
         <Address>http://gateway.seller.com/external</Address>
      </Transport>
   </Connector>
   <Connector>
      <Name>
         BUY:C:buySpice:exotic
      </Name>
      <EnvelopeType>SOAP-C1</EnvelopeType>
      <isNative>true</isNative>
      <ConnectorFunction>service-receive</ConnectorFunction>
      <Transport type="Sonic" mode="async" reliable="true">
         <Address>SonicCluster1:SellApp</Address>
      </Transport>
   </Connector>
</Route>
```

[0041]

[0042] 该示例示出了与图3的变化相对应的路由，其中源301使用ebXML/HTTPS协议304来与集线器303通信。源连接器301是固有连接器，名称为nutmeg。集线器连接器303也是固有连接器，名称为GW1，它执行封装网关功能。目的地302使用SOAP/Sonic协议305来与集线器303通信。目的地302是固有连接器，名称为exotic。所有三个连接器属于社区BUY:C:buySpice。由源和目的地使用的传输协议是软件实现固有的，所以集线器303将封装和传输协议翻译作为单一功能来执行。因此，它在该示例路由编码中仅仅出现一次。

[0043] 图13和14是社区内路由和社区间路由的高层示意图。路由包含源、目的地和它们之间的一系列连接器。对于安全路由，路由还包含一个或多个安全区域，使得通信总保持安全。在图13中，可以是服务的源和目的地二者被映射到连接器1301。本地注册表1302包含关于连接器的信息，来自该连接器的部分路径列表能够被构造成起始于源和目的地连接器。构造电子商务文档经过的连接器的部分路径列表包含几个子步骤。注册表中的信息包含连接器间通信的路由规则，它能够表示为定向图。构造部分路径列表包含从子到父连接器进行的穿越。在穿越中的每一个连接器处，对于下一个顶点可用的替换传输/封装协议被看作是分离的部分路径。穿越从源或目的地的部分路径的完成对应于到达连接器，该连接器不具有使用该穿越的特定传输/封装协议的可用的父连接器。可替换地，穿越部分路径的完成可以对应于到达使用该穿越的特定传输/封装协议进行对等通信的连接器。如果将标志设置为允许连接器忽略到相同的子社区中其他连接器的路由，则确定连接器是否进行对等通信可能需要考虑该连接器所属于的子社区。通过从子到父连接器进行穿越，建立了对于源和目的地连接器的部分路径列表，每一个列表中包含一个或多个部分路径。源和目的地部分路径列表被链接1304。当各源和目的地部分路径列表共享连接器，并且该共享连接器的传输/封装协议与两个列表中的相同，或者该共享连接器具有在各列表的各传输/封装协议之间进行翻译的翻译能力时，各源和目的地部分路径列表能够通过共享连接器被链接。各列表还能够在各列表中的相似连接器之间被链接，相似链接器是支持相同的传输/封装协议的连接器。如上所述，对于该路由方法的扩展是使用传输/封装/安全性三者组成的一组作为构造部分路径列表的基础。链接各列表的另一个替换方案是通过一个或多个中心连接器来链接，该中心的连接器具有在出现在各部分路径列表中的传输/封装/安全性协议之间进行翻译的能力。为了使用中心连接器，部分路径列表可以从父连接器被扩展到没有明确地链接到父连接器的中心连接器。

[0044] 图14应用路由交叉社区。路由交叉社区包含社区之间的跳跃以及社区之内的连接器到连接器的内部跳跃。可以是服务的源和目的地被映射到连接器1401。各源和目的地连接器属于社区。反过来，源和目的地社区属于社区网络。一个或多个社区网络从注册表1403被标识1402为链接源和目的地社区。如果源和目的地社区在社区网络中共享成员资格，则各社区的外部连接器可以使用公共传输/封装协议被直接链接。如果源和目的地社区没有公共传输/封装协议，则一个或多个中间社区可以被添加到路由中来执行翻译服务。源和目的地的外部连接器被链接到一个或多个中间社区的相似连接器。在一些实现中，中间社区的数量可以被限制为1、2、3或一些其他少于5或10的小数目，以便简化路由。当已经确定了潜在地通过中间社区的源和目的地社区的外部连接器之间的路由时，就计算每一个参加社区之内的社区内路由。在源和目的地社区中，利用一个或多个注册表1405中的数据，计算1404从源和目的地连接器到各外部连接器的路由。对于中间社区，计算从进入到离开外部连接器的路由。通过组合社区内和社区间路由1406来规定完整的路由，以便产生从源到目的地的的连接器到连接器路由。路由交叉社区可以利用存储在本地或全球注册表中的预先计算的路由。预先计算的路由可以指定路由消息经过的中间社区。中间社区可以提供翻译、记帐、商务情报或其他服务。当计算了用于从源到目的地发送消息的路由、以便节省路由供以后使用时，它可能是有效率的。可替换地，可以有效地利用空闲CPU周期，来预先计算社区网络中到其他社区的路由，或者到可以通过当前社区网络中的社区到达的任何非当前社区

网络的社区的路由。

[0045] 由于可能会遭遇到很多偷窃，电子商务文档的路由最好遵循安全性和被信路由。如果没有加密，有线上的文档处于危险之中。源或目的地可能受到危害，并且暴露被推测为维持保密状态的信息。源和目的地之间的连接器可能会恶意活动：它们可能抛弃、延迟或重发文档，记录它们接收到的并且暴露保密信息的文档，或者修改文档。负有翻译责任的连接器在翻译期间可能恶意地或者简单地改变文档的语义。这些已知的问题制造了提供如下方法和设备的机会，该方法和设备用于电子商务文档的安全和被信通信。

[0046] 图15到19描述安全和被信通信的一些使用情况。源和目的地之间的信任是通过先前的契约而建立的。例如，作为交易伙伴的公司同意并且可以以合作契约的形式注册它们的契约。该合作契约可以包含使用的文档类型以及相互认可安全性重点，例如签名和加密。安全性契约可以在交易伙伴之间达成，或者可以由各交易伙伴所属于的社区通过。达成契约的方式对于被信通信的安全性来说不是重要的。

[0047] 当沿从源到目的地的路径上的连接器作为被信任的连接器而被列在注册表中时，它们是被信任的。集线器被信任，以便根据路由沿路由中的下一个跳来发送文档，并且保护它对于内容的任何了解或者甚至对该文档沿路由的传输的了解。当它的功能仅仅是转发该文档时，集线器无需关心文档是否被加密或签名。为了支持加密通信，例如虚拟私人网络通信或HTTPS通信，集线器可以拥有密钥和证书，如实现PKI或其他安全模式那样。

[0048] 提供封装翻译服务的连接器，即所谓的网关，提出了更多复杂的信任问题。从一种格式到另一种格式的文档翻译，需要集线器或连接器能够解密它所接收到的东西，并且对签名进行确认。翻译之后，网关重新签名并且加密它翻译的东西。为了完成所有这些，由网关接收到的消息应当使用网关的密钥被加密，而不是目的地的密钥。最好是参照支持验证签名的注册表，网关必须能够验证所接收的消息上的签名。如果它接收到的消息是签名的，当网关已经翻译了文档时，它就重新签名该文档。如果不需要进一步的翻译，它还使用与沿路由的下一个网关或者目的地相对应的密钥来加密该消息。该网关可以具有多个密钥对或证书，能够将不同的密钥对用于加密、签名以及用于安全、虚拟私人网络连接。

[0049] 从源到目的地、通过网关进行的验证是顺序的。第一个网关检验源的签名，并且重新签名该文档。该网关必须信任该源，后续的网关必须信任它们从其接收消息的网关。该链中的每一个网关都信任它前面的网关，验证它接收的签名，并且施加它自己的签名。在建立该安全链的过程中，例如，SAML断言能够被接受并且由网关施加。网关需要保持翻译文档的扩展存档，以便解决任何发生的争端。文档应当被存档两次，一次是解密之后作为初始文档，最好包含接收的签名(如果有的话)，而另一次是翻译之后，加密之前，最好显示出网关的签名。

[0050] 总之，当源和目的地彼此信任，源和目的地信任执行翻译的网关，并且该路由中的每一个连接器信任该路由中前面的和后续的连接器时，就可以建立路由中的信任。前面讨论了源和目的地之间的信任。在电子商务中，它对于在交换电子文档之前彼此信任的伙伴是有意义的。源和目的地应当信任网关，因为网关执行翻译功能。被信网关的列表可以保持在注册表中。如果源或目的地缺少对于执行翻译服务的网关连接器的信任，就不应当使用该网关。虽然不是全部的交易伙伴都对此非常敏感，但是意识到安全问题的交易伙伴，例如国防工业参加者，则认为知道可能能够读取文档内容的每一个和全部的连接器是有利的。

对于仅仅转发文档而不解密或翻译它们的那些连接器，可以放松信任要求。

[0051] 例如，考虑特定的源和目的地：源使用MML封装，目的地使用ebXML。一种可能的路由将是从源到第一个网关，使用MML；从MML到SOAP的封装转换；从第一个网关转发到第二个网关；从SOAP到ebXML的翻译；以及从第二个网关到目的地的转发。在这种情况下，该源必须信任网关和目的地。该目的地必须信任该源和网关。如果源或目的地不信任执行翻译的中间网关，则该路由是不可接受的。

[0052] 网关还应当信任前面一个的源或网关以及紧接着的网关或目的地。这可以被称为传递信任。该信任链中执行翻译的每一个元素都想肯定它正在与它能够信任的元素进行交互。接着前面的示例，假设路由从源穿越到网关1，继续到网关2，和到网关3，接着到目的地。沿该路由，第一个网关必须信任源和第二个网关。第二个网关必须信任第一个和第三个网关。第三个网关必须信任第二个网关和目的地。实际上，网关3信任网关1，因为它信任网关2，而网关2反过来信任网关1。当传递信任是由源和目的地之间的明确信任所组合时，认为传递信任沿该种路由是充分的。当中间社区提供翻译网关时，可以采用一种替换的、简化的信任模式。那么，信任提供翻译服务的中间社区是充分的，而无需信任由中间社区执行翻译服务所使用的每一个和全部网关。只有当涉及中间社区时，才认为可以应用这种简化的信任。例如，源和目的地处于相同的社区中，该源和目的地将彼此信任，但是可能不信任执行文档翻译的网关。在社区之内，应当明确信任网关。在社区网络之内，可以信任中间社区来执行确定的翻译服务，而无需信任中间社区的翻译机制的每一个和全部组成部分。

[0053] 可以使用很多不同的验证和信任模式。三种目前使用的用于验证的安全模式包含：用户名和口令；SAML验证断言；和X.509证书。这些和其他安全模式的操作被集体地称为获得安全凭证(credential)。这些安全凭证可以由作为分离的处理或例程或代码段而运行的服务器提供。对于用户名和口令验证，接收方确认用户名和口令。用户名和口令的通信应当经由加密信道。

[0054] SAML断言更复杂。源和目的地之间的信任部分是目的地信任产生SAML断言的管理局。该信任可以是社区范围水平的或者是关于由特定源使用的特定管理局。一般地，被信SAML验证管理局和其他被信管理局的证书被保持在注册表中，以便由需要检验该断言的方面访问。

[0055] 在操作中，从SAML客户机可以访问SAML服务。SAML服务从SAML客户机接收请求，并且回应该客户机。例如，可以使用SOAP封装进行该通信。SAML服务可以支持验证和属性请求。响应于验证请求，它可以产生安全凭证。由SAML服务产生的断言一般将被使用SAML服务的签名密钥进行签名。对于基本验证，SAML服务可以从SOAP封装的应用凭证头块中提取交易伙伴或用户id和口令。该服务调用注册表访问应用程序接口，来获取与交易伙伴或用户id相对应的口令。它比较接收的和注册口令。如果它们匹配，则产生断言并且签名。如果它们不匹配，则它报告错误。在替换实施例中，在被存储到注册表中以前，可以对口令进行散列处理。如果如此，则比较散列数据。

[0056] 对于X.509验证，信任应当基于发放该证书的证书管理局。被信证书管理局的列表可以基于证书管理局的身份标识或证书管理局身份标识与通信方(correspondent)身份标识的组合。被信证书管理局的列表，例如VeriSign三级证书，可以包含不管它们来自哪个社区都被信任的证书。可替换地，被信证书管理局的列表可以是对于远程社区特定的。于是，

Boeing证书可能仅仅在它们来自Boeing社区时才被信任。VeriSign证书可能对于多个社区被信任。

[0057] SAML服务可以响应于X.509验证证书。当接收到X.509证书时，它可以从SOAP封装的应用凭证头块中提取该证书。它可以检索存储在附件中的处理信息。如果附件为空，它将报告验证失败。它将比较从附件的处理信息字段中提取的客户机证书与来自凭证头块中的证书。如果证书不匹配，它就报告验证失败。如果它们匹配，它就调用注册表访问应用接口，来从该注册表中获取交易伙伴或用户id证书。它比较处理信息和注册证书。如果它们不匹配，它就报告验证失败。如果它们匹配，它就产生断言并且可对其签名。该断言被返回给客户机。

[0058] 信任机制还可以用于实施电子商务文档的不同级别的分类。例如，关键的军用零部件可能需要一组特殊的极机密的安全凭证，而物资供应所(commissary)商品可以使用常规断言来处理。特殊路由规则可以应用于特殊断言。例如，极机密的文档可能仅仅通过特殊的被授权翻译极机密文档的被信网关来进行路由。特殊的被信网关可以服从附加安全措施，例如加强的监视、特别健壮的加密、专用传输介质等等。

[0059] 假定源和目的地之间信任，安全和被信通信机制可以依赖于如下因素，例如源和目的地是否处于相同的社区，它们是否与安全服务器直接交互或者通过代理进行交互，以及两者是否都使用相同的安全机制。图15描述一种社区内的安全性实现。在该社区内，全部连接器，包含源1511和目的地1512彼此信任。在该示例中，安全服务器1501是该社区本地的SAML服务器。源1511从SAML服务器1501获取用于验证的安全凭证，并且还可以获取管理局的属性断言。至少将安全凭证和电子商务文档打包在封装中，并且沿社区内的路由将其从源发送到目的地。该路由可以包含图中未示出的连接器。

[0060] 图16描述第一个社区1605中的源1611和第二个社区1606中的目的地1612。网络概念需要源和目的地具有关于它们与其交换文档的其他社区的信息。当SAML被用作安全机制时，关于其他社区的信息包含其他社区中的被信任提供安全服务的SAML服务器的身份标识。其他社区中的SAML服务器被明确列表为受本地社区信任。在图15中，每一个社区1605、1606包含SAML服务器1601、1602以及源1611或目的地1612。当源发送文档给目的地时，该源请求其SAML服务器1601提供SAML断言。该源还查询目的地的SAML服务器1602以便检索验证信息。来自两个SAML服务器的断言连同电子商务文档被发送到目的地。作为性能增强，源所属于的社区或源的安全服务能够保存从第二个社区1606检索到的SAML断言或者本地和外部SAML断言，以便以后使用，从而改善了系统性能。替换SAML机制，当使用PKI机制时，给源发放证书的被信证书管理局的证书必须在目的地社区(1606)中注册。对于用户名和口令安全性，有效的用户名和口令组合在目的地社区和网关中注册。

[0061] 图17描述安全服务的授权。在该图中，连接器1711提供安全服务，并且充当源1721和1722的代理。可以将源1721和1722注册为依赖代理1711提供安全服务。当代理1711与SAML服务1701通信时，它标识它自己以及源，它代表该源来请求安全断言。接着，SAML服务器给代表源的主机发出安全断言，并且还可以发出属性断言。该主机至少将安全断言和电子商务文档转发给目的地。安全断言可以伴随公开主机在该安全处理中涉及的忠告声明。

[0062] 图18A和18B示出了将验证授权给网关1812。在图18A中，连接器1821、1822处于大社区1805的子社区1806中。这些连接器1821、1822利用MML协议与网关1812进行通信。它们

将MPID、用户id和口令传递给网关。处理信息包含相同的信息。网关进行注册表查找，来定位它接收到的MPID、用户id和口令。如果在注册表中找到这三者，该网关就创建凭证对象作为该MML封装的附件。该网关将授权验证请求发送到SAML服务。SAML服务使用网关的凭证进行验证。SAML服务返回验证断言，该验证断言可以包含忠告部分。该验证断言是网关1812所转发给目的地1811的一部分。

[0063] 在图18B中，反转通过网关1812进行的通信。源1811与子社区1806中的使用MML协议的连接器1821、1822通信。源1811从SAML服务器1801获取SAML安全凭证，并且将其连同电子商务文档一起转发到网关1812。网关将SOAP封装转换成MML封装。网关使用他自己的MPID、用户id和口令来创建安全凭证，并且将其附着到MML封装中。MML封装被发送到使用MML的子社区1806。

[0064] 在图19A和19B中，将MML和SAML安全协议之间的翻译扩展到社区1905、1906之间的通信。源1922位于使用MML的社区中，而目的地1911位于使用SOAP的社区中。源交易伙伴1922依靠入口路由器1921将安全凭证和电子商务文档转发到处理封装和安全变换的网关1912。路由器1921将安全凭证传递到网关1912。该安全凭证在MML封装内部。该网关从MML封装的安全凭证中提取CPID，并且将授权验证请求发送到SAML服务。验证是使用网关的凭证进行的。SAML服务返回验证断言，如上所述。验证断言和翻译的电子商务文档被转发到目的地911。在图19B中，反转通过网关1912进行的通信。源1911与使用MML协议的子社区1906中的目的地1922通信。源1911从SAML服务器1901获取SAML安全凭证，并且将其连同电子商务文档一起转发到网关1912。该网关将SOAP封装转换成MML封装。该网关使用它自己的MPID、用户id和口令来创建安全凭证，并且将其附着到MML封装中。该MML封装被发送到使用MML的子社区1906。

[0065] 为了便利上述授权情形，可以如下扩展SAML协议方案。

[0066]

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://schemas.commerceone.com/wse/security/delegation"
xmlns:samlp="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-
protocol-27.xsd" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:c1del="http://schemas.commerceone.com/wse/security/delegation"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="0.1">
    <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="xmldsig-core-schema.xsd"/>
    <xsd:import namespace="http://www.oasis-open.org/committees/security/docs/draft-
sstc-schema-protocol-27.xsd" schemaLocation="draft-sstc-schema-protocol-27.xsd"/>
    <!--
        //DelegateFor - 代表正在为其进行验证请求的TP或服务的ID
        //CommunityID - 定义TP或服务的社区ID
        //CommunityType - 社区类型（例如 MOE4x)
        //Description - 仅仅是信息

    -->
    <xsd:complexType name="DelegationType">
        <xsd:complexContent>
            <xsd:extension base="samlp:AuthenticationQueryType">
                <xsd:attribute name="DelegateFor" type="xsd:string"/>
                <xsd:attribute name="CommunityID" type="xsd:string"/>
                <xsd:attribute name="CommunityType" type="xsd:string"/>
                <xsd:attribute name="Description" type="xsd:string"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
</xsd:schema>
```

[0067] 符合该方案的授权验证请求如下：

[0068]

```
<?xml version="1.0" encoding="UTF-8"?>
<Request xmlns="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-
protocol-27.xsd" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:saml="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-
assertion-27.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:c1del="http://schemas.commerceone.com/wse/security/delegation"
xsi:schemaLocation="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-
protocol-27.xsd
C:\XMLSPY~2.3\draft-sstc-schema-protocol-27.xsd"
xsi:schemaLocation="http://schemas.commerceone.com/wse/security/delegation
C:\XMLSPY~2.3\sec-delegation-ext.xsd" RequestID="1fgtTGzMXSqpN++/LcFpBmZWrQg="
MajorVersion="1" MinorVersion="0" IssueInstant="2001-09-11T09:30:47-05:00">
        <RespondWith>AuthenticationStatement</RespondWith>
        <!-- SAML服务将签名块看作一小块 -->
        <ds:Signature Id="ID01">
            <!--    数字签名    -->
        </ds:Signature>

        <AuthenticationQuery xsi:type="c1del:DelegationType" DelegateFor="TPxxx">
            <saml:Subject>
                <saml:NameIdentifier Name="unique-string-that-identifies-the-TP"/>
                <saml:SubjectConfirmation>
                    <!--      对于基本验证     -->
                    <saml:ConfirmationMethod>http://www.oasis-
open.org/committies/security/docs/draft-sstc-core-27/password
                    <!-- 对于基于 X509 证书的验证
                    <saml:ConfirmationMethod>http://www.oasis-
open.org/committies/security/docs/draft-sstc-core-27/X509
                    -->
                    </saml:ConfirmationMethod>
                </saml:SubjectConfirmation>
            </saml:Subject>
        </AuthenticationQuery>
</Request>
```

[0069] 注意，上面的DelegateFor和NameIdentifier指相同的实体，为其执行授权验证的参加者。包含授权的对于该请求的示例响应如下：

[0070]

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- 由XML Spy 4.2 U产生的示例XML文件 (http://www.xmlspy.com)-->
<Response xmlns="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-
protocol-27.xsd" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:saml="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-
assertion-27.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-
protocol-27.xsd
C:\XMLSPY~2.3\draft-sstc-schema-protocol-27.xsd" ResponseID="String"
InResponseTo="1fgtTGzMXSqpN++/LcFpBmZWrQg=" MajorVersion="1" MinorVersion="0"
IssueInstant="2001-09-11T09:30:47-05:00">
```

```
<Status>
    <StatusCode Value="samlp:Success">
        <SubStatusCode Value="q:name"/>
    </StatusCode>
    <StatusMessage>String</StatusMessage>
    <StatusDetail/>
</Status>
<saml:Assertion MajorVersion="1" MinorVersion="0"
AssertionID="+1UyxJDBUza+ao+LqMrE98wmhAI=" Issuer="String" IssueInstant="2001-09-
11T09:30:47-05:00">
    <saml:Conditions NotBefore="2001-09-11T09:30:47-05:00"
NotOnOrAfter="2001-09-11T09:30:47-05:00"/>
    <saml:Advice>
        <saml:Subject>
            <saml:NameIdentifier Name="ID-of-the-delegate"/>
        </saml:Subject>
        <saml:AdviceElement xsi:type="xsd:string" value="some
description"/>
    </saml:Advice>
    <saml:AuthenticationStatement>
[0071]      <saml:Subject>
                <saml:NameIdentifier Name="unique-string-that-identifies-
the-TP"/>
                <saml:SubjectConfirmation>
                    <!--    对于基本验证    -->
                    <saml:ConfirmationMethod>http://www.oasis-
open.org/committies/security/docs/draft-sstc-core-27/password
                    <!-- 对于基于 X509 证书的验证
                    <saml:ConfirmationMethod>http://www.oasis-
open.org/committies/security/docs/draft-sstc-core-27/X509
                    -->
                    </saml:ConfirmationMethod>
                </saml:SubjectConfirmation>
            </saml:Subject>
        </saml:AuthenticationStatement>
        <ds:Signature Id="ID11">

        </ds:Signature>
    </saml:Assertion>
</Response>
```

[0072] 当SAML服务通过授权创建了断言时，它使用上面的<Advice>块来披露该授权。没有授权，响应将是相似的，除了没有<Advice>块以外。

[0073] 在一个实施例中，断言方案不允许单一的请求消息包含验证和属性请求两者。在该实施例中，SAML客户机提出请求：首先是验证请求，接着是属性请求。属性请求跟随验证。示例的属性请求如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- 由 XML Spy v4.2 U 产生的示例XML文件 (http://www.xmlspy.com)-->
<!-- 属性请求 -->
<Request xmlns="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-
protocol-27.xsd" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:saml="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-
assertion-27.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-
protocol-27.xsd
C:\XMLSPY~2.3\draft-sstc-schema-protocol-27.xsd"
```

[0074]

```
RequestID="1fgtTGzMXSqpN++/LcFpBmZWrQg=" MajorVersion="1" MinorVersion="0"
IssueInstant="2001-09-11T09:30:47-05:00">
    <RespondWith>AttributeStatement</RespondWith>
    <AttributeQuery>
        <saml:Subject>
            <saml:NameIdentifier Name="unique-string-that-identifies-this-TP"/>
        </saml:Subject>
        <saml:AttributeDesignator AttributeName="attribute-name-string"
AttributeNamespace="attribute-name-space-string"/>
    </AttributeQuery>
</Request>
```

[0075] 属性断言响应跟随该请求。

[0076] 图20到25中示出了安全使用情况的其他细节。图20是简单的客户-服务设计的方框图。在该设计中，注册服务验证和费用授权基于CP级别身份标识。注册管理者验证和授权基于用户级别身份标识。注册服务使用SAML进行验证。它直接调用提供商应用程序接口来确定特权；它基于经过验证的CP身份标识来进行它自己的授权。SAML客户机2012访问存储的SAML客户机数据2011。SAML客户机2012与SAML客户机交换机2013通信，该交换机2013在本地和远程情况之间进行交换。在本地情况下，与SAML服务2016直接通信。在远程情况下，组成部分2014、2015处理交换机和远程SAML服务之间的通信。凭证和SAML断言可以经由HTTPS或另一个安全协议在组成部分之间交换。SAML服务2016与服务和用户管理提供商2017通信，服务和用户管理提供商2017可以适应于基于证书的用户标识、基于用户id和口令的验证、或其他验证方案。服务和用户管理提供商2017与支持公共对象架构(COF)2018的模块通信，该COF 2018反过来访问注册表2019。

[0077] 图21示出了使用图20中的设计的注册服务本地验证。该图中的编号与图20相同，除了增加注册客户机2101、注册服务2105、COF 2106和注册表2107以外。在这种使用情况下，开始调用注册客户机2101。注册客户机2101调用SAML客户机2112来获取验证。它可以使用上述的任何验证方案。SAML客户机可以访问存储的SAML客户机数据2111。它可能能够根

据存储的数据来确认验证请求，并且返回有效的断言。如果不是，它继续进行远程验证。SAML客户机通过组件2114、2115与SAML服务2116通信。SAML服务执行验证，产生安全凭证，可以签名，并且将该安全凭证返回该SAML客户机2112。图22示出了这种使用情况下对于注册服务远程验证的变化。再次地，开始调用注册客户机2201。注册客户机确定需要进行的远程调用。它调用组件2202，来通过到对应的组件2204的通信信道-例如https连接2203-进行通信，并且还与SAML客户机2212通信。处理如上所述继续进行。

[0078] 图23和24分别示出了本地和远程授权。在图23中，注册客户机2301直接调用注册服务2305，传递如图21所示所获取的安全凭证。注册服务用安全凭证调用SAML客户机2312，以便确认该安全凭证，并且获得经过验证的CPID。利用该经过验证的CPID，注册服务2305调用用户管理提供商数字(numeral)2317，以便获取对于经过验证的CPID的特权。通过COF 2318，从注册表2319获取对于CP的特权。注册服务数字2305执行该特权。在图24中，将本地授权过程扩展到远程授权。作为验证，注册客户机2401通过组件2402、2403通信，在这种情况下通过组件到达注册服务2405。

[0079] 图25示出了获取对于文档服务预订的属性断言。组件2502和2503被进一步细化为包含ICD客户机2511、2521和授权模块2512、2522。ICD客户机2511调用与发送端注册表2541相关联的发送端ICD服务2531。从发送端，到划分线2500的左边，ICD服务2531调用接收端ICD服务2532。安全计算器调用SAML客户机2533来获得属性断言。SAML客户机调用本地的接收端SAML服务2534。SAML服务调用服务提供商应用接口2535来获取发送者信息，并且创建属性断言。服务提供商可以访问注册表2542。属性断言被传递回到发送端ICD服务2531，并且被打包到ICD安全块2511中。授权模块2512将属性断言放置在封装头中，例如SOAP封装头中。将该封装发送到组件2503，其中接收端的授权模块2522读取属性断言，并且执行预订。

[0080] 图26和27描述社区网络的建立。在图26中，两个社区2601、2605参加社区网络。社区的操作者设置它们的社区，包含外部端口2602、2606和本地注册表2603、2607。本地注册表标识外部端口。社区的操作者进行操作配置2611，操作配置2611可以是常规合同或电子合同。提供操作配置以便社区形成或加入网络。社区将具有暴露给网络中其他社区的一个或多个服务。发现服务和推进服务允许网络中的社区使用其他社区中的服务，而无需重复注册。社区的操作者还建立网络搜索注册表，有时称为全球黄页目录2614。当网络操作时，社区可以将服务信息从它们的本地注册表推进到全球黄页，或者全球黄页可以从参加的社区中拉取信息。全球黄页可以轮询参加的社区或等待来自可以拉取信息的社区的通知。社区操纵者可以将特定服务标记为本地社区可见、特定网络可见、全部网络可见、或者特定的其他社区可见。全球黄页中的服务信息列表应当与来自视野内的标记服务相对应。这些全球黄页可以被实现为基于UDDI的注册表。它可以是社区外部的，或者由网络中的一个社区提供。在社区的网络中，可能存在一个或多个全球黄页。社区的操作者交换关于每一个其他社区的外部端口细节和URL 2612的信息。该信息使能在外部端口2602、2606之间进行通信。社区的操作者还交换涉及安全的信息，例如安全凭证或SAML证书2613。利用拥有的所交换的信息，各社区的操作者用连接到外部社区所需的信息来装载它们的注册表。例如，它们可以注册外部社区的外部端口和安全信息。它们还可以注册与外部社区端口相对应的它们自己的外部端口。社区的操作者还建立全球搜索注册表，有时被称为白页2615。它可以是社区外部的，或者由网络中的一个社区提供。当全球搜索注册表由一个社区提供时，它用作社区

地址服务器。要成为该社区地址服务器工作于其中的社区网络的一部分的单独社区，能够通过与社区地址服务器交换设置信息2611、2612、2613而加入到网络中，并且能够通过在它们的本地注册表中注册社区地址服务器的端口、来从社区地址服务器获取其他外部社区的设置信息，例如端口、安全凭证等，而不是通过特殊地设置网络中每一个和全部社区的注册入口。可以通过社区地址服务器来建立互操作性。遵守社区网络中的规则和惯例，并且潜在地服从彼此做生意的社区中的双边契约2611。

[0081] 图27描述超过两个社区的网络，其中一个社区提供社区地址服务器。社区2702保持社区地址服务器2704。社区2701和2703与主社区交换信息，并且变得能够发现它们中的每一个提供的服务、允许的协议，以便彼此通信。根据网络的规则，它们的通信可以是直接的、由中间社区作为中介的、或者被强制经过中间社区。在超过两个社区的网络中，可以在参加的社区之间建立直接的私有链接，并且将其记录在各社区注册表中，或者社区地址服务器可以用于建立连接和信任。较少使用的情况是建立超过两个社区的网络而没有社区地址服务器，所有的成员基于点对点进行相互注册。这种使用情况不会受益于新成员流水线式地加入到网络中。

[0082] 虽然已参照上面详细描述的优选实施例和示例公开了本发明，应该理解，这些示例旨在进行说明而不用于限制意义。上述实施例包含了计算机辅助处理。因此，可以将本发明实施为计算机辅助处理的方法、包含实施该方法的逻辑的系统、嵌入了执行该方法的逻辑的介质、附加了执行该方法的逻辑的数据流、或计算机可访问的处理服务。可以预料到，本领域技术人员将容易地进行修改和组合，这种修改和组合将落在本发明的实质和所附权利要求的范围内。

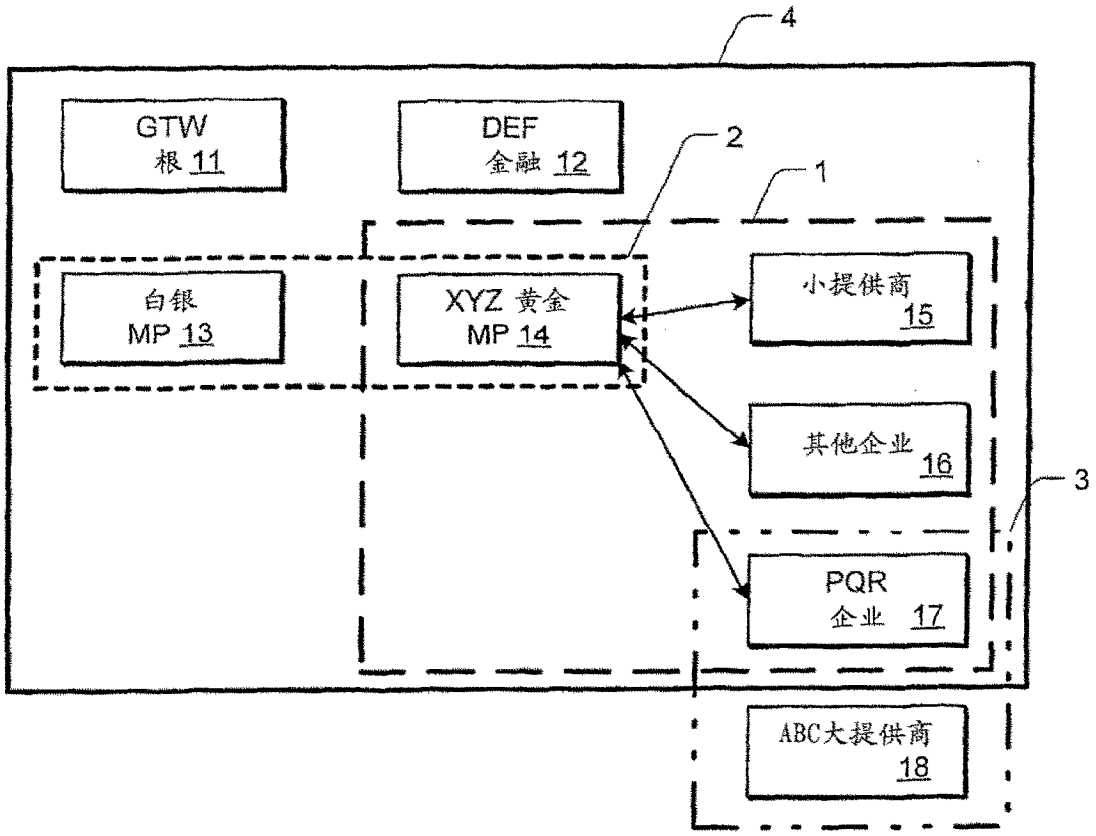

4
GTW
根 11
DEF
金融 12
2
1
白银
MP 13
XYZ 黄金
MP 14
小提供商
15
其他企业
16
3
PQR
企业 17
ABC大提供商
18

图1

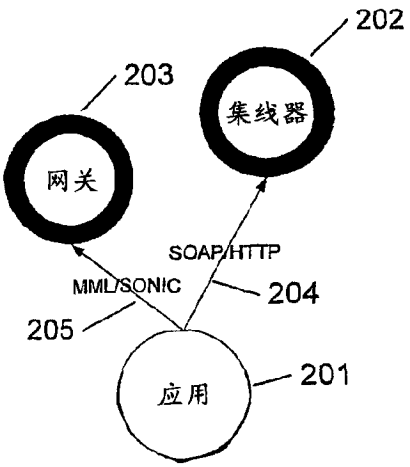

202
203
集线器
网关
SOAP/HTTP
MML/SONIC
204
205
应用
201

图2

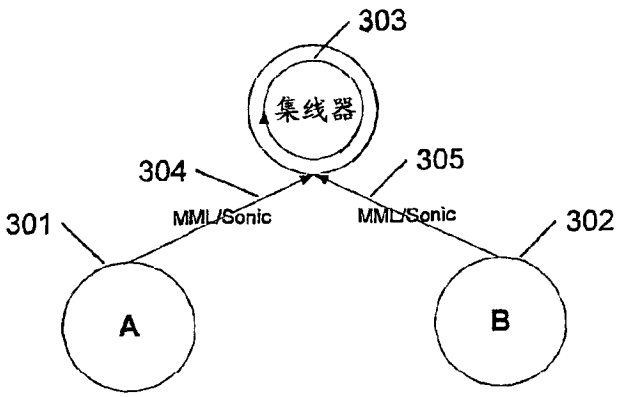

303
集线器
304
305
MML/Sonic
MML/Sonic
301
302
A
B

图3

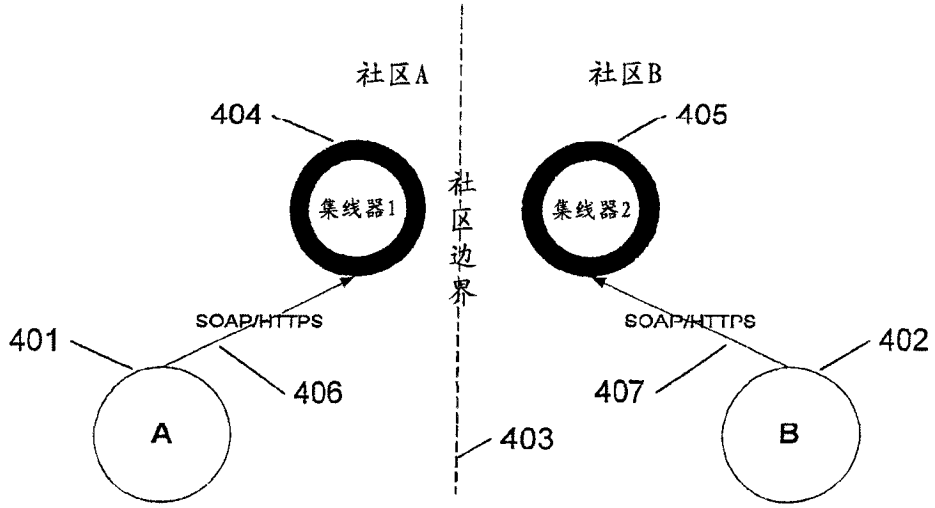

社区A
社区B
404
405
集线器1
集线器2
社区边界
SOAP/HTTPS
SOAP/HTTPS
401
402
406
407
A
B
403

图4

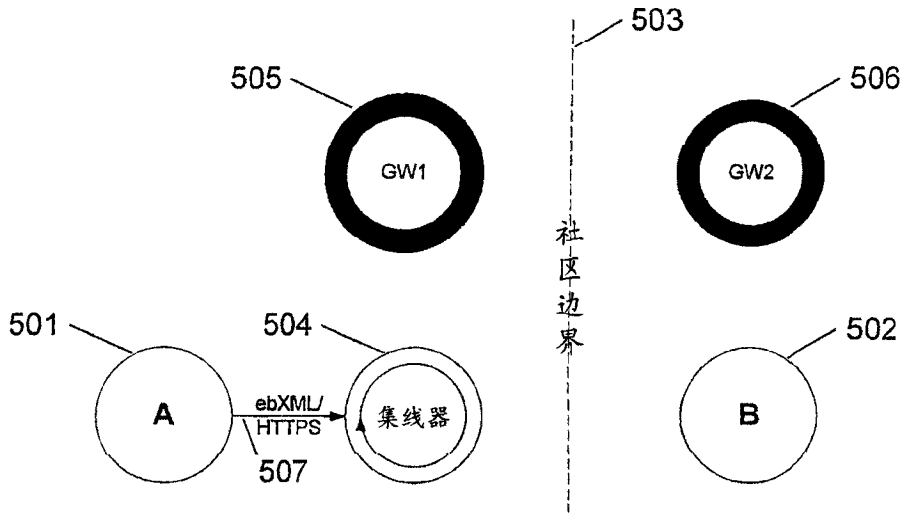

503
505
506
GW1
GW2
社区边界
501
504
502
A
ebXML/
HTTPS
集线器
B
507

图5

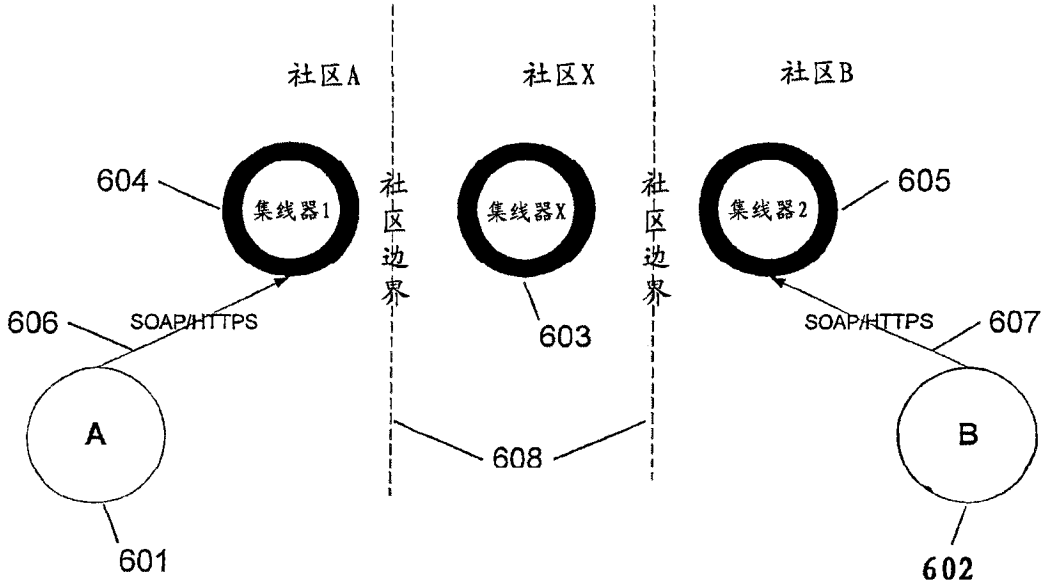

社区A
社区X
社区B
604
集线器1
社区边界
集线器X
社区边界
集线器2
605
606
SOAP/HTTPS
603
SOAP/HTTPS
607
A
608
B
601
602

图6

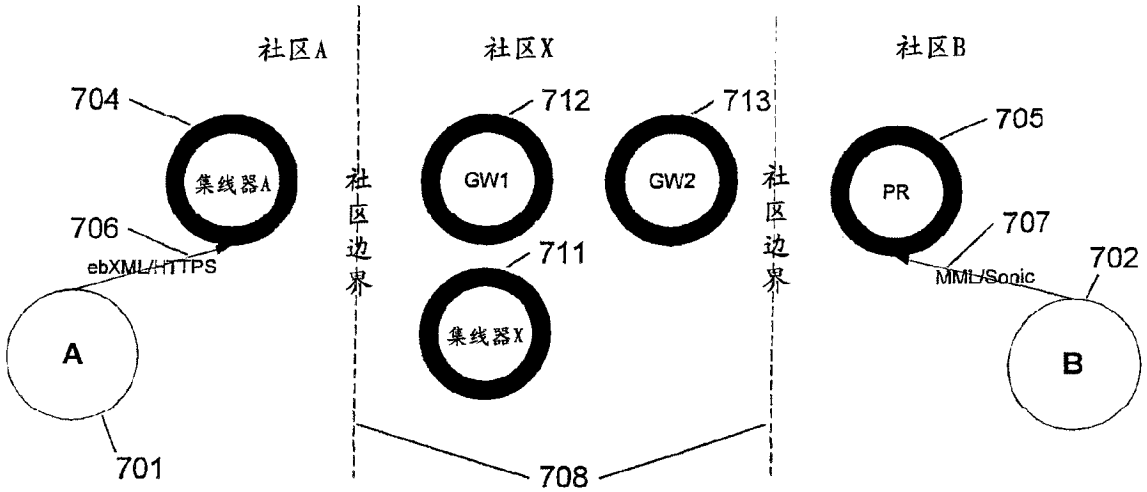

社区A
社区X
社区B
704
712
713
705
集线器A
社区边界
GW1
GW2
社区边界
PR
706
711
707
ebXML/HTTPS
MML/Sonic
702
A
集线器X
B
701
708

图7

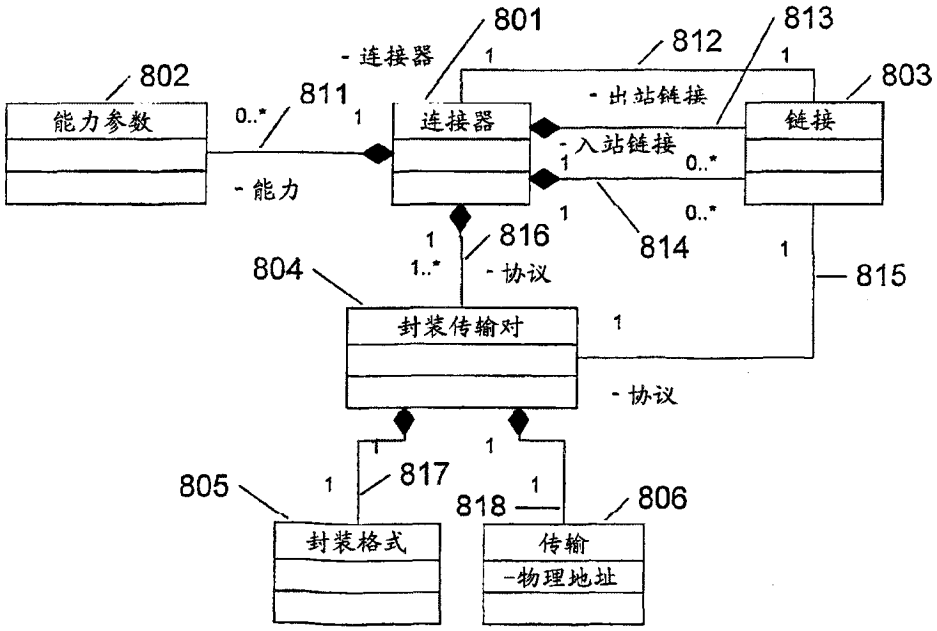

801
- 连接器
1
812
813
1
802
811
803
能力参数
0..*
1
连接器
- 出站链接
链接
- 入站链接
1
0..*
- 能力
1
0..*
1
816
814
1
1..*
- 协议
804
815
封装传输对
1
- 协议
1
1
805
1
817
818
1
806
封装格式
传输
-物理地址

图8

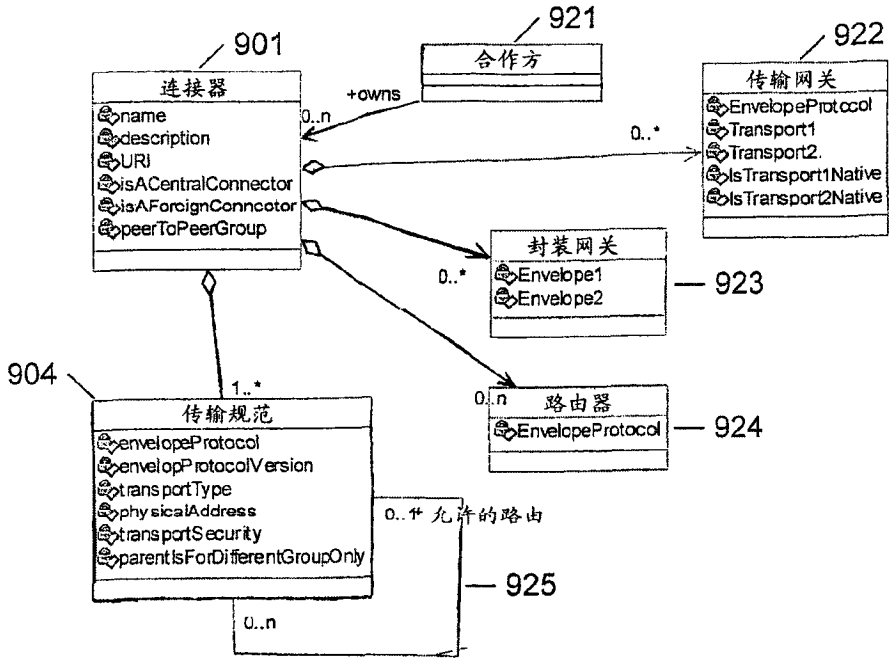

901
921
922
连接器
合作方
传输网关
+owns
0..n
name
description
URI
isACentralConnector
isAForeignConnector
peerToPeerGroup
EnvelopeProtocol
Transport1
Transport2
IsTransport1Native
IsTransport2Native
0..*
0..*
封装网关
Envelope1
Envelope2
923
904
1..*
0..n
路由器
EnvelopeProtocol
924
传输规范
envelopeProtocol
envelopProtocolVersion
transportType
physicalAddress
transportSecurity
parentIsForDifferentGroupOnly
0..n
允许的路由
925
0..n

图9

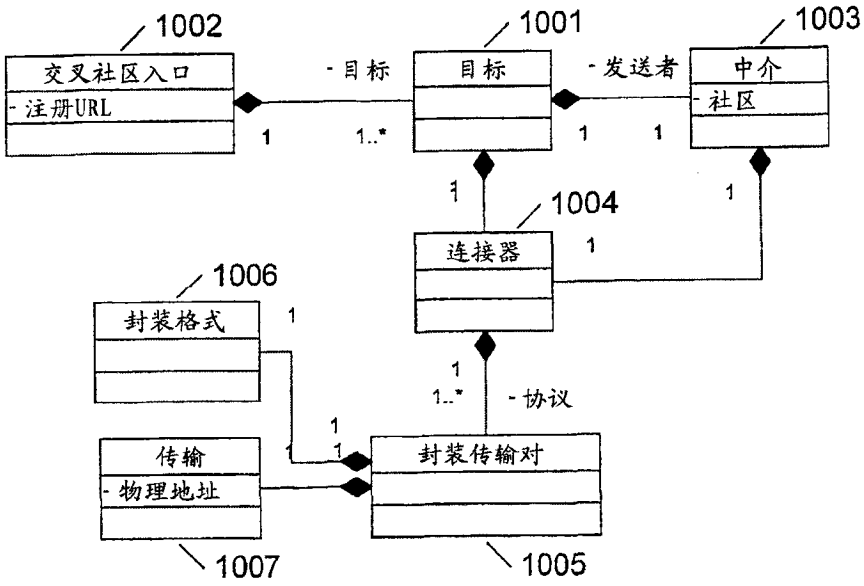

1002
交叉社区入口
- 注册URL
- 目标
1
1..*
1001
目标
- 发送者
1
1
1003
中介
- 社区
1
1
1004
连接器
1
1
1006
封装格式
1
1
1..*
- 协议
传输
- 物理地址
1
1
封装传输对
1007
1005

图10

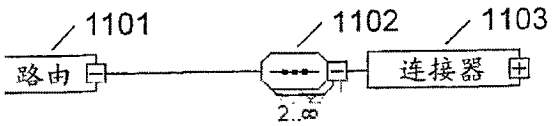

1101
路由
1102
2..∞
1103
连接器

图11

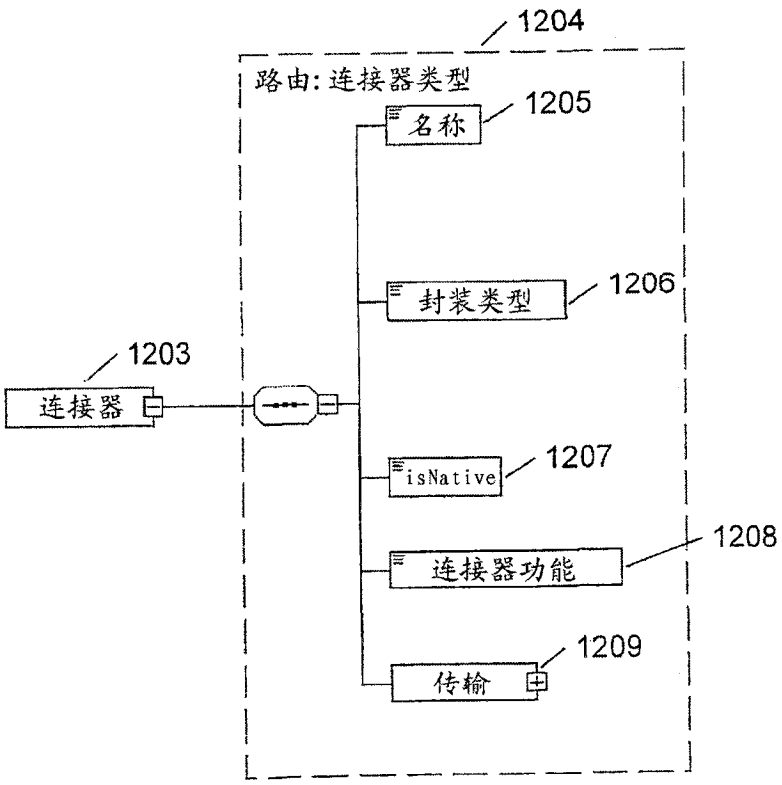

1204
路由:连接器类型
1205
名称
1206
封装类型
1203
连接器
1207
isNative
1208
连接器功能
1209
传输

图12

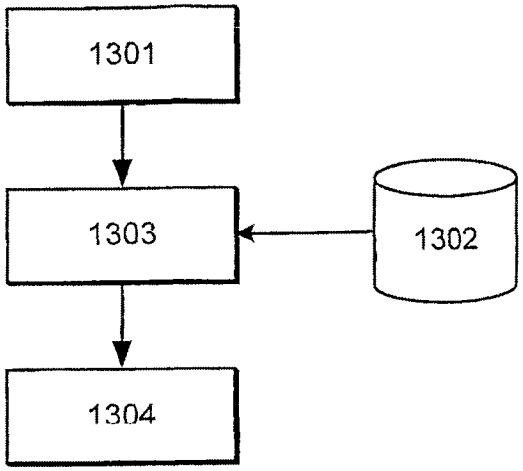

1301
1303
1302
1304

图13

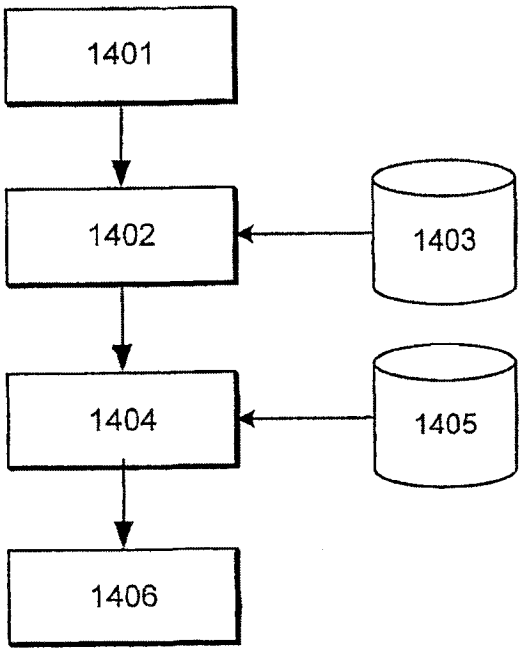

1401
1402
1403
1404
1405
1406

图14

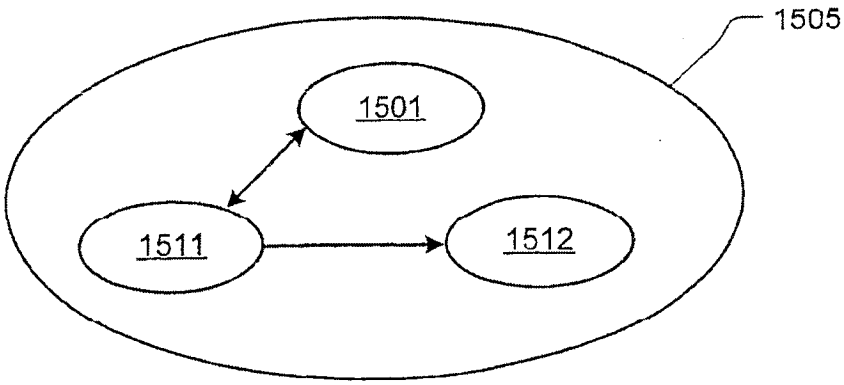

1505
1501
1511
1512

图15

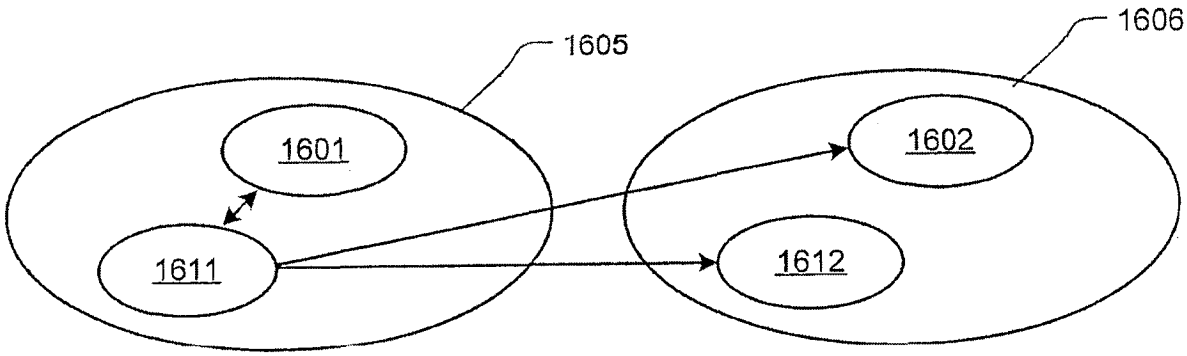

1605
1606
1601
1602
1611
1612

图16

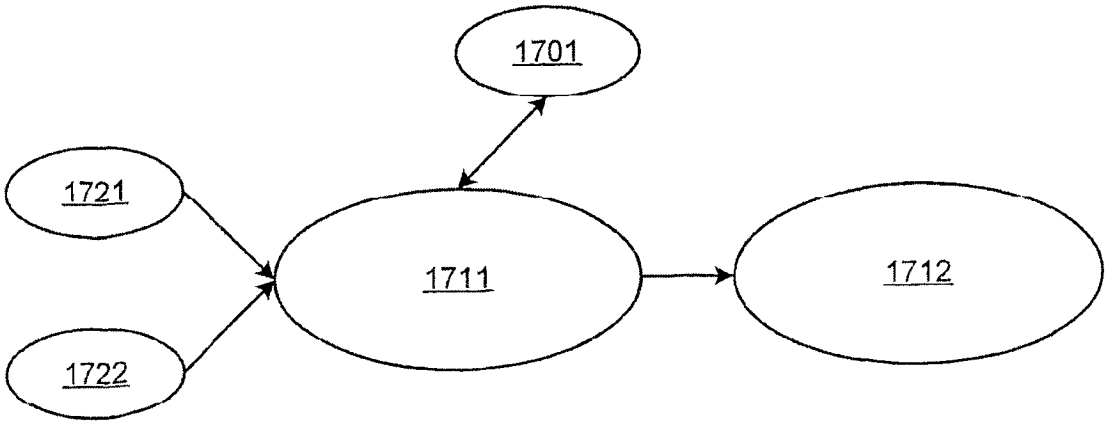

1701
1721
1711
1712
1722

图17

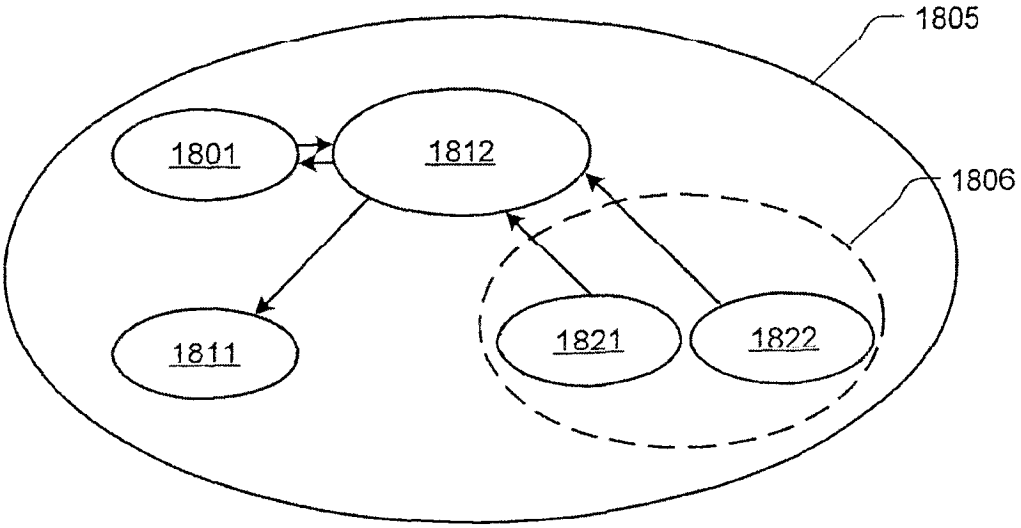


图18A

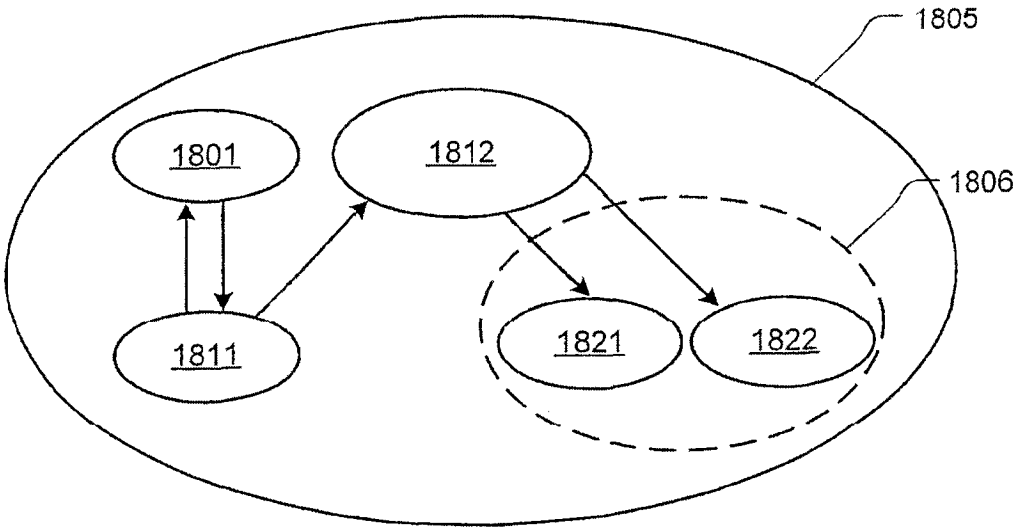


图18B

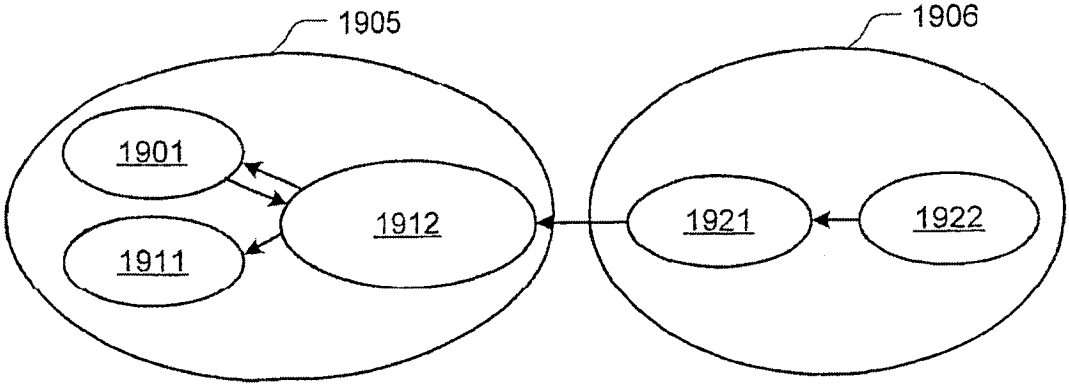


图19A

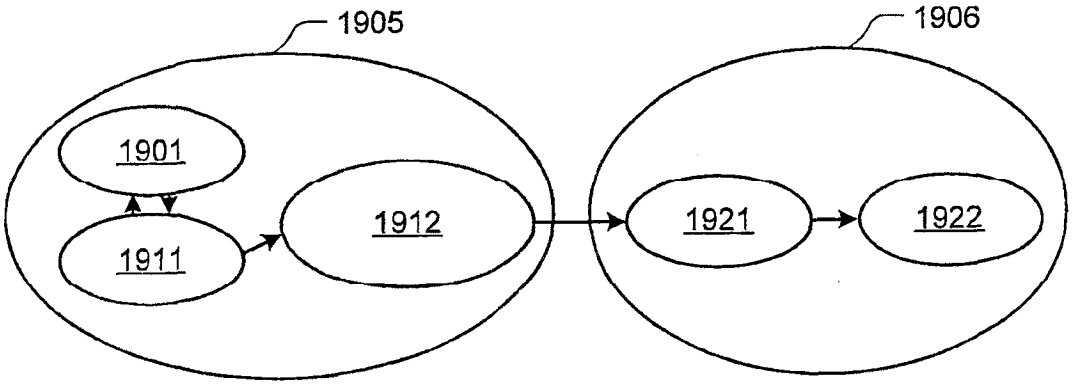

1905
1906
1901
1911
1912
1921
1922

图19B

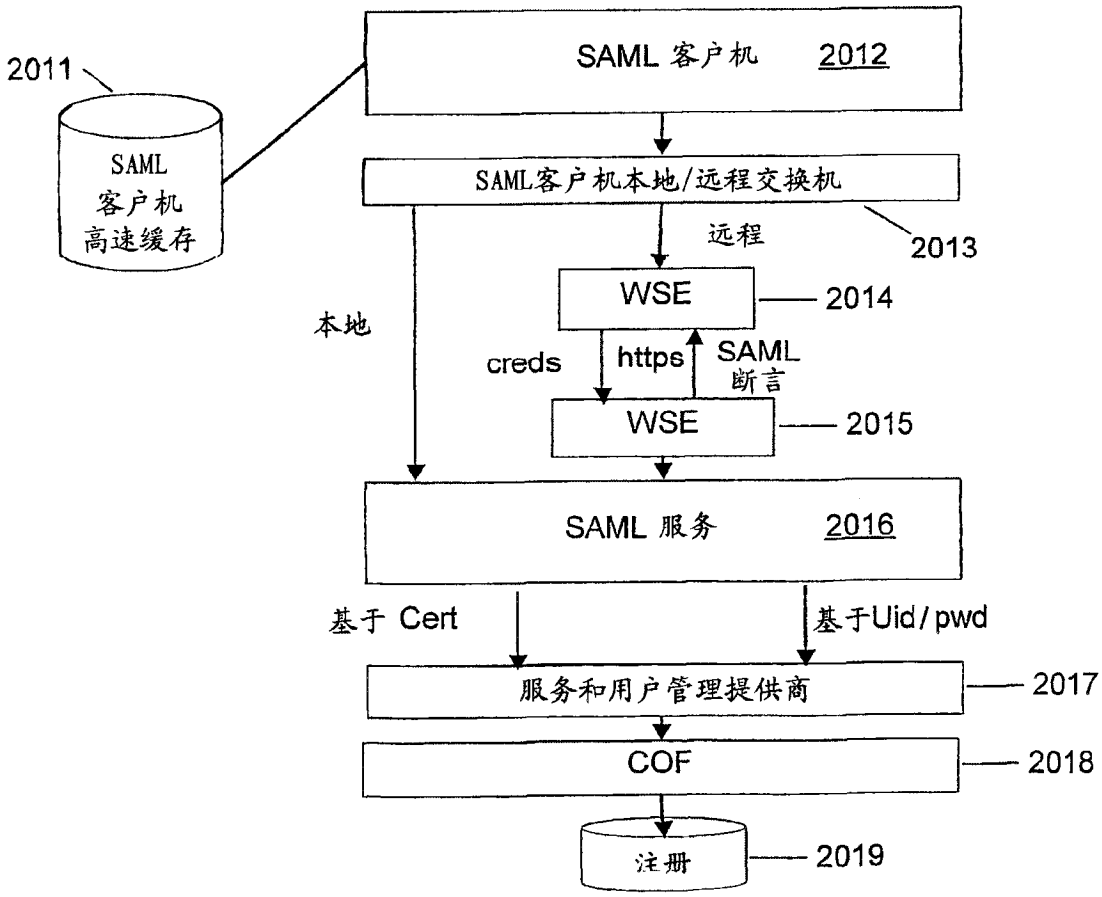

2011
SAML
客户机
高速缓存
SAML 客户机 2012
SAML客户机本地/远程交换机
远程
2013
WSE
2014
本地
creds
https
SAML
断言
WSE
2015
SAML 服务 2016
基于 Cert
基于Uid/pwd
服务和用户管理提供商
2017
COF
2018
注册
2019

图20

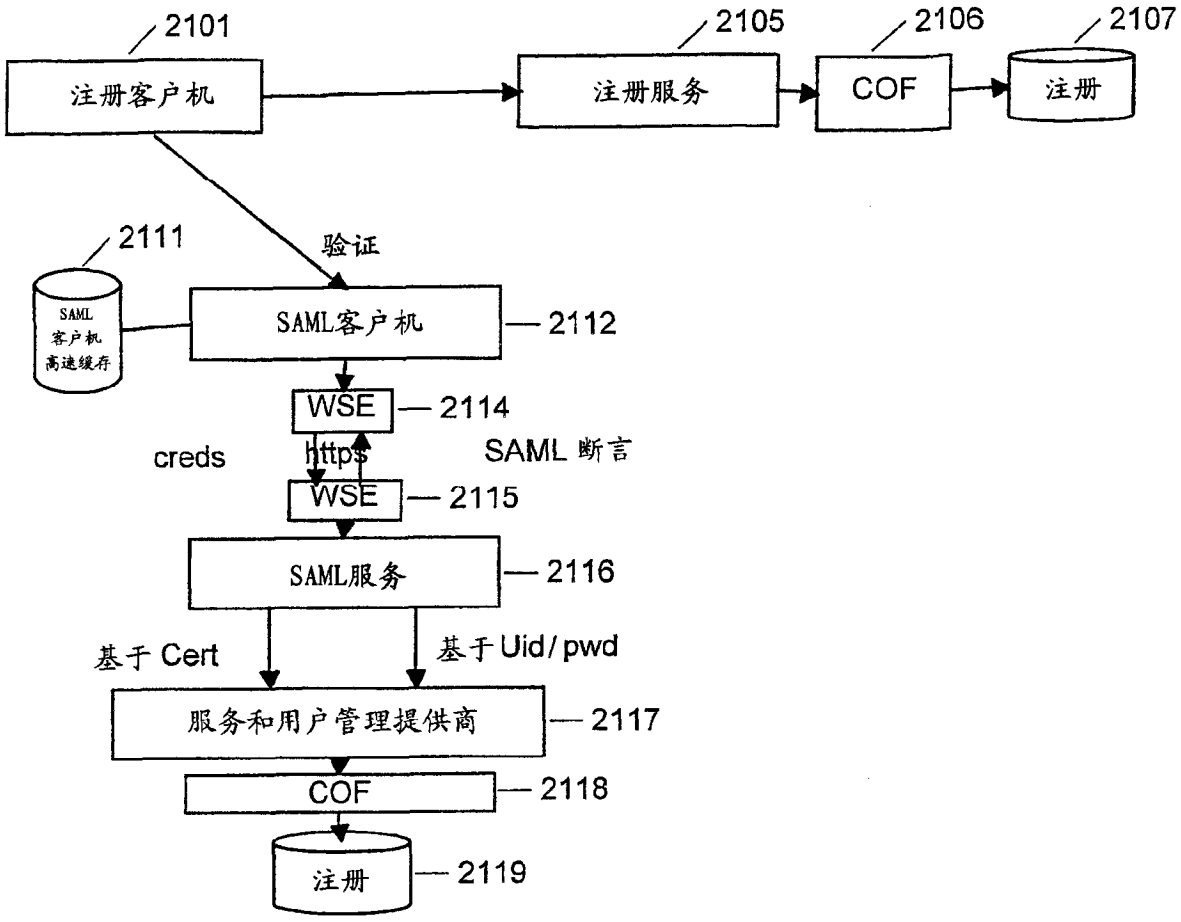

2101
注册客户机
2105
注册服务
2106
COF
2107
注册
验证
2111
SAML 客户机高速缓存
SAML客户机
2112
WSE
2114
creds
https
SAML 断言
WSE
2115
SAML服务
2116
基于 Cert
基于 Uid/pwd
服务和用户管理提供商
2117
COF
2118
注册
2119

图21

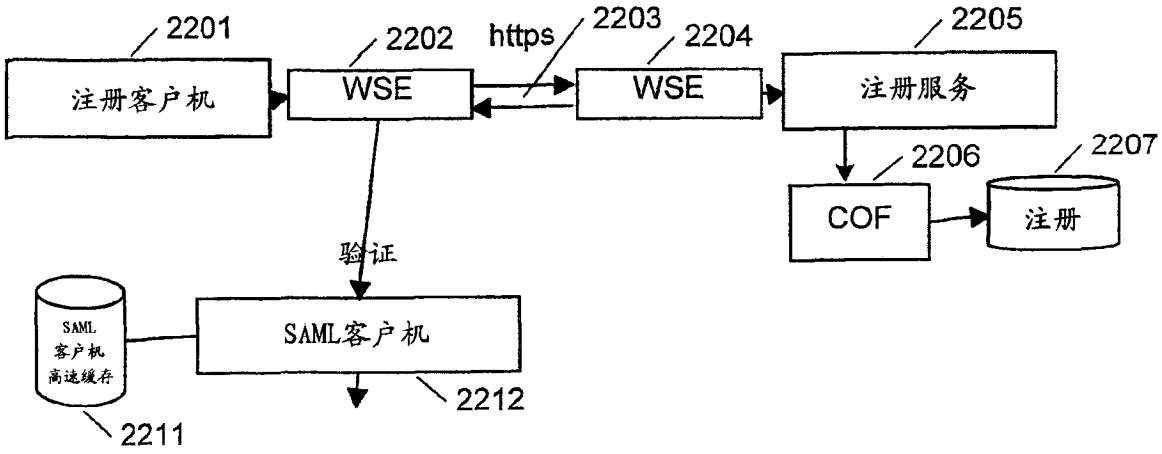

2201
注册客户机
2202
WSE
https
2203
2204
WSE
2205
注册服务
2206
COF
2207
注册
验证
SAML 客户机高速缓存
SAML客户机
2212
2211

图22

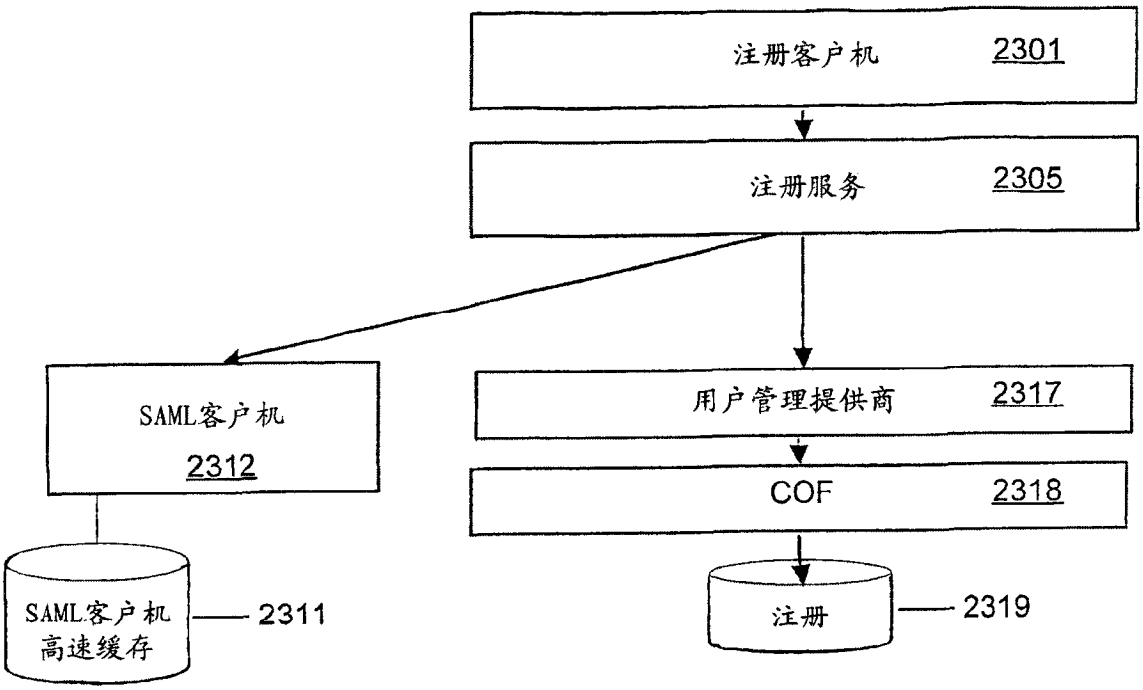

注册客户机
2301
注册服务
2305
SAML客户机
2312
用户管理提供商
2317
COF
2318
SAML客户机
高速缓存
2311
注册
2319

图23

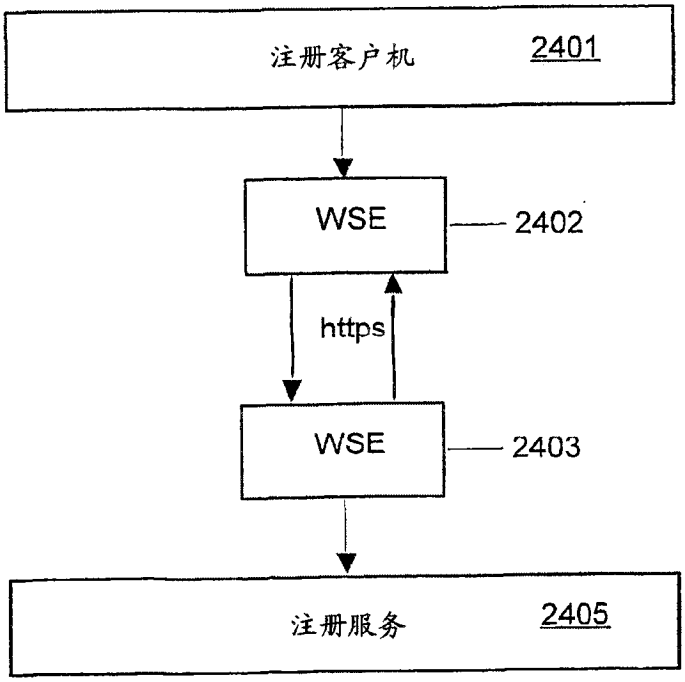

注册客户机
2401
WSE
2402
https
WSE
2403
注册服务
2405

图24

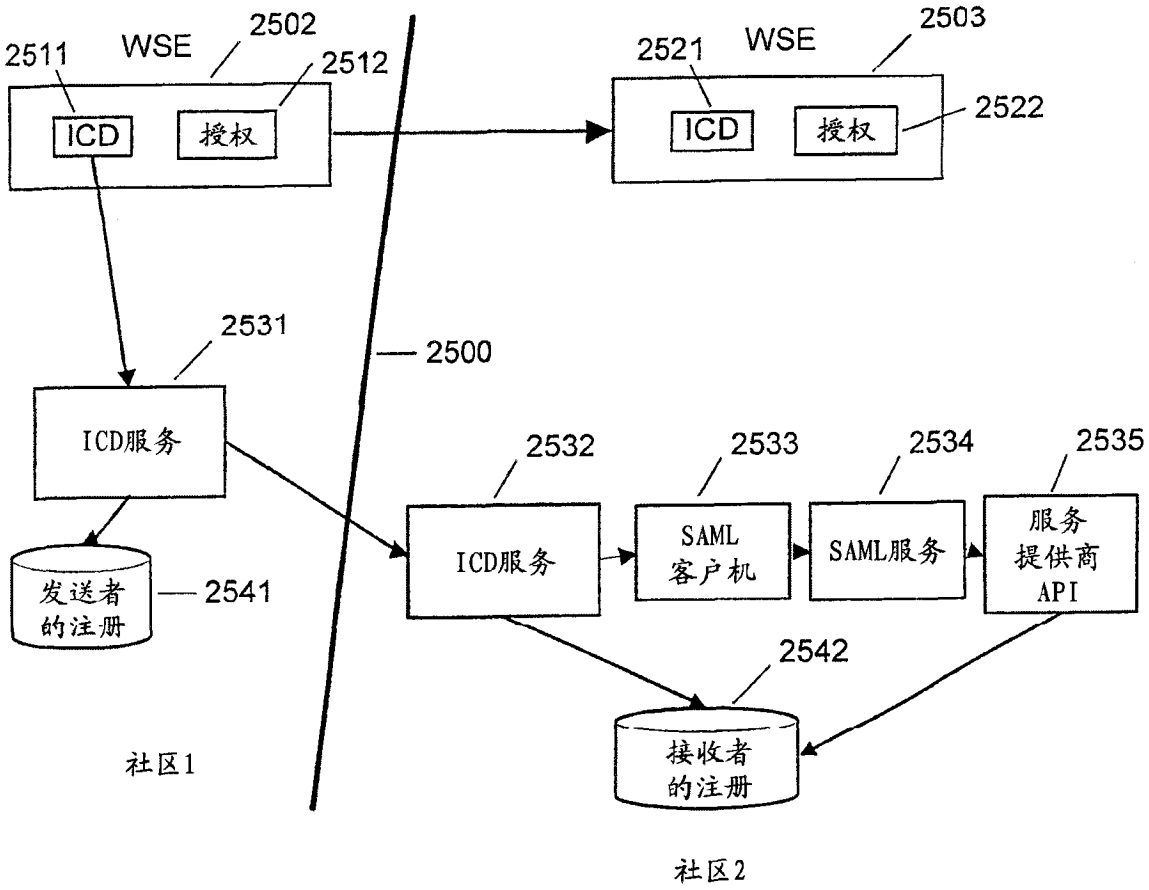

2511
WSE
2502
2512
ICD
授权
2521
WSE
2503
ICD
授权
2522
2531
2500
ICD服务
2532
2533
2534
2535
ICD服务
SAML
客户机
SAML服务
服务
提供商
API
发送者
的注册
2541
2542
接收者
的注册
社区1
社区2

图25

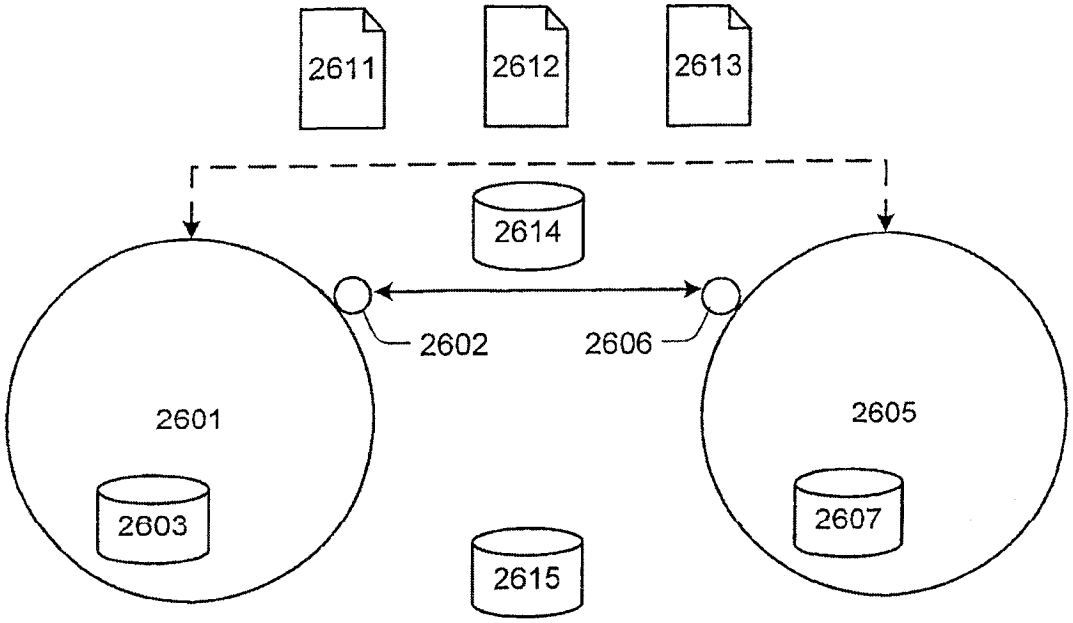

2611
2612
2613
2614
2602
2606
2601
2605
2603
2607
2615

图26

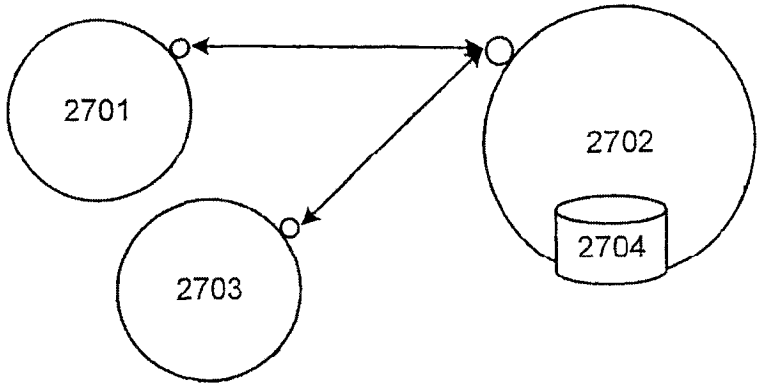


图27