



## [12] 发明专利申请公开说明书

[21] 申请号 03824896.4

[43] 公开日 2005 年 11 月 9 日

[11] 公开号 CN 1695339A

[22] 申请日 2003.8.19 [21] 申请号 03824896.4

[30] 优先权

[32] 2002. 9. 18 [33] US [31] 10/246,592

[86] 国际申请 PCT/US2003/025971 2003. 8. 19

[87] 国际公布 WO2004/027547 英 2004. 4. 1

[85] 进入国家阶段日期 2005. 4. 30

[71] 申请人 JGR 阿奎西申公司

地址 美国特拉华州

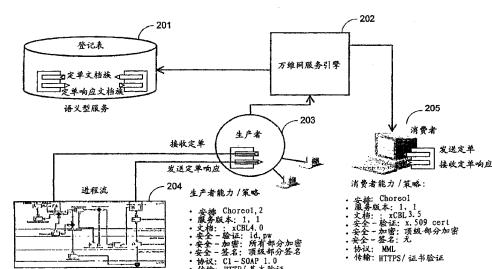
[72] 发明人 杰亚拉姆·R·卡西 拉什米·默西  
西蒙·S·Y·常  
托德·C·克劳斯 海伦·S·尤恩[74] 专利代理机构 北京市柳沈律师事务所  
代理人 郭定辉 黄小临

权利要求书 4 页 说明书 70 页 附图 3 页

[54] 发明名称 用于万维网服务的动态互通性合同

[57] 摘要

本发明涉及机器可读数据结构和数据结构的动态计算以支持互通性。更具体地说，本发明涉及提高互通性的数据结构和数据结构的动态生成这些方面。在权利要求书、说明书和附图中描述了本发明的具体方面。



1. 一种为消费服务和提供服务指定互通性数据的机器可读数据结构，这些服务通过网络可选地利用中间连接器交换文档，该数据结构包括：
- 5       通过服务名指定的服务和中间连接器之间的路线以及有名服务和连接器之间的路线；  
        用于消息交换的安排版本；  
        存档消息、保证消息的可靠交付和要求接收确认从而可以减少接收被拒绝的策略。
- 10      2. 根据权利要求 1 所述的数据结构，还包括在服务之间交换的特定消息的一些部分的签名要求和使用的至少一种签名算法的说明；  
          3. 根据权利要求 1 所述的数据结构，还包括在服务之间交换的特定消息的一些部分的加密要求和使用的至少一种签名算法的说明。  
          4. 根据权利要求 1 所述的数据结构，还包括使用的一个或多个验证过程  
15      的说明。  
          5. 根据权利要求 1 所述的数据结构，还包括：  
        应用于包括在在服务之间交换的特定消息中的文档的一个或多个转换逻辑单元的说明； 和  
        是否应该与文档的转换副本一起包括文档的非转换副本的说明。
- 20      6. 一种为消费服务和提供服务指定互通性数据的机器可读数据结构，这些服务通过网络可选地利用中间连接器交换包括文档的消息，该数据结构包括：  
          在服务之间交换的特定消息的一些部分的签名要求和使用的至少一种签名算法的说明；  
          25     在服务之间交换的特定消息的一些部分的加密要求和使用的至少一种签名算法的说明； 和  
          使用的一个或多个验证过程的说明。  
          7. 根据权利要求 6 所述的数据结构，还包括通过服务名指定的服务和中间连接器之间的路线以及有名服务和连接器之间的路线。
- 30      8. 根据权利要求 6 所述的数据结构，还包括用于消息交换的安排版本。  
          9. 根据权利要求 6 所述的数据结构，还包括存档消息、保证消息的可靠

支付和要求接收确认从而可以减少接收被拒绝的策略。

10. 根据权利要求 6 所述的数据结构，还包括：

应用于包括在在服务之间交换的特定消息中的文档的一个或多个转换逻辑单元的说明；和

5 是否应该与文档的转换副本一起包括文档的非转换副本的说明。

11. 一种为消费服务和提供服务指定互通性数据的机器可读数据结构，这些服务通过网络可选地利用中间连接器交换包括文档的消息，该数据结构包括：

10 应用于包括在在服务之间交换的特定消息中的文档的一个或多个转换逻辑单元的说明；和

是否应该与文档的转换副本一起包括文档的非转换副本的说明。

12. 根据权利要求 11 所述的数据结构，还包括通过服务名指定的服务和中间连接器之间的路线以及有名服务和连接器之间的路线。

13. 根据权利要求 11 所述的数据结构，还包括用于消息交换的安排版本。

14. 根据权利要求 11 所述的数据结构，还包括存档消息、保证消息的可靠交付和要求接收确认从而可以减少接收被拒绝的策略。

15. 根据权利要求 11 所述的数据结构，还包括在服务之间交换的特定消息的一些部分的签名要求和使用的至少一种签名算法的说明；

16. 根据权利要求 11 所述的数据结构，还包括在服务之间交换的特定消息的一些部分的加密要求和使用的至少一种签名算法的说明。

17. 根据权利要求 11 所述的数据结构，还包括使用的一个或多个验证过程的说明。

18. 一种为消费服务和提供服务指定当前互通性数据的机器可读数据结构，这些服务通过网络交换包括文档的消息，该数据结构是通过如下的进程准备的：

响应启动服务之间的消息交换的请求，访问服务的互通性数据；

使服务的互通性数据相交；和

对于产生多于一个可相互接受选项的互通性数据的相交，应用判定规则来选择一个选项。

30 19. 根据权利要求 18 所述的数据结构，其中，判定规则由服务预约。

20. 根据权利要求 18 所述的数据结构，其中，判定规则通过服务的预约

适合于贸易共同体。

21. 根据权利要求 18 所述的数据结构，其中，互通性数据包括如下的一个或多个：

通过服务名指定的服务和中间连接器之间的路线以及有名服务和连接器  
5 之间的路线；

用于消息交换的安排版本；

存档消息、保证消息的可靠交付和要求接收确认从而可以减少接收被拒  
绝的策略；

通过服务名指定的服务和中间连接器之间的路线以及有名服务和连接器  
10 之间的路线；

用于消息交换的安排版本；

存档消息、保证消息的可靠交付和要求接收确认从而可以减少接收被拒  
绝的策略；

应用于包括在在服务之间交换的特定消息中的文档的一个或多个转换逻  
15 辑单元的说明； 和

是否应该与文档的转换副本一起包括文档的非转换副本的说明。

22. 根据权利要求 19 所述的数据结构，其中，互通性数据包括如下的一个或多个：

通过服务名指定的服务和中间连接器之间的路线以及有名服务和连接器  
20 之间的路线；

用于消息交换的安排版本；

存档消息、保证消息的可靠交付和要求接收确认从而可以减少接收被拒  
绝的策略；

通过服务名指定的服务和中间连接器之间的路线以及有名服务和连接器  
25 之间的路线；

用于消息交换的安排版本；

存档消息、保证消息的可靠交付和要求接收确认从而可以减少接收被拒  
绝的策略；

应用于包括在在服务之间交换的特定消息中的文档的一个或多个转换逻  
30 辑单元的说明； 和

是否应该与文档的转换副本一起包括文档的非转换副本的说明。

23. 根据权利要求 18 所述的数据结构，其中，互通性数据包括：  
通过服务名指定的服务和中间连接器之间的路线以及有名服务和连接器之间的路线；  
用于消息交换的安排版本；
- 5 存档消息、保证消息的可靠交付和要求接收确认从而可以减少接收被拒绝的策略；
24. 根据权利要求 18 所述的数据结构，其中，互通性数据包括：  
在服务之间交换的特定消息的一些部分的签名要求和使用的至少一种签名算法的说明；
- 10 在服务之间交换的特定消息的一些部分的加密要求和使用的至少一种签名算法的说明；和  
使用的一个或多个验证过程的说明。
25. 根据权利要求 18 所述的数据结构，其中，互通性数据包括：  
应用于包括在服务之间交换的特定消息中的文档的一个或多个转换逻辑单元的说明；和  
是否应该与文档的转换副本一起包括文档的非转换副本的说明。
26. 一种为消费服务和提供服务指定互通性数据的机器可读数据结构，这些服务通过网络可选地利用中间连接器交换包括文档的消息，该数据结构包括：
- 20 可应用于签名、加密或验证中的一个或多个的一个或多个安全通道，其中，安全通道包括：  
发出安全相关请求的连接器；  
响应安全相关请求的连接器；和  
作为签名、加密或验证中的一个或多个的安全相关请求的说明。
- 25 27. 根据权利要求 26 所述的数据结构，其中，如果安全相关请求是签名、加密、验证或不拒绝中的一个或多个，安全通道可应用于签名、加密、验证或不拒绝中的一个或多个和说明。
28. 根据权利要求 26 所述的数据结构，其中，数据结构是响应启动服务之间的消息交换的请求形成的。
- 30 29. 根据权利要求 27 所述的数据结构，其中，数据结构是响应启动服务之间的消息交换的请求形成的。

## 用于万维网服务的动态互通性合同

### 5 版权通告

本专利文件的公开部分包含受版权保护的内容。版权拥有者不反对任何人一模一样地复制像出现在专利和商标局专利文件或记录中那样的专利文件或专利公开，但是在其它方面仍然保留所有版权。

### 10 参考计算机程序列表附录

计算机程序列表附录附在本说明书之后。计算机程序列表附录包括如下程序节选。

InteroperabilityContract.XSD (总合同的模式)

GeneralContract.XSD (一般信息的模式)

15 RoutingContract.XSD (消息路由的模式)

TransformationContract.XSD (文档转换的模式)

SecurityContractKeyInfo.XDS (用于安全的密钥的模式)

SecurityContract.XDS (通过协商达成的安全合同的模式)

InteroperabilityContract.XML (互通性合同的例子)

20 ComputeSecurityContract.XML (计算安全合同举例)

### 技术领域

本发明涉及机器可读数据结构和数据结构的动态计算以支持互通性。更具体地说，本发明涉及提高互通性的数据结构和数据结构的动态生成这些方面。在权利要求书、说明书和附图中描述了本发明的具体方面。

### 背景技术

商家到商家（B2B）和应用到应用（A2A）电子商务正在取代用于电子数据交换（EDI）的旧协议。随着商家力争借助于B2B和A2A系统来提高它们的效率，出现了大量不兼容平台和竞争标准。在兼容标准之间，仍然需要填补相互之间的空隙。例如，企业定义了简单万维网服务是什么。与简单万

维网服务有关的标准包括 UDDI、WSDL、XSDL 和 SOAP。但是，这些标准没有完全满足实际 B2B 和 A2A 电子商务的安全性、可靠性、可管理性和安排（choreography）要求。特别是安全性将许多选择和配置问题摆在人们面前。人们期望协同万维网服务和它们的安全需要像非万维网商业那样演变。还没有任何随着万维网服务不断演变动态地解决和更新安全选择和配置的综合或统一设备或方法。

许多企业首创扩展了可应用于 B2B 和 A2A 电子商务的标准。在安排方面的成果包括 OASIS 公司所作的 ebXML/BPSS、IBM 公司所作的 WSFL 和 Microsoft 公司所作的 XLANG。在会话方面的成果包括 OASIS 所作的 ebXML/TRP 和 Microsoft 公司的 WS-routing。与 ebXML 首创有关的进一步信息可从网站 <http://www.ebxml.org/specs/index.htm#whitepapers> 获得，在该网络中可找到 ebXML Trading-Parters Team (2001 年 5 月 10 日) 发现的文章“Collaboration-Protocol Profile and Agreement Specification Version 1.0”。关于交互和服务合同强制执行者逻辑性的无歧义规则，也可在美国专利第 6,148,290 号中找到。占优势的安全成果是 IBM 和 Microsoft 公司所作的 WS-security，还存在称为 SAML 的 OASIS 公司所作的辅助安全成果。有关可靠性，存在 Microsoft 公司提出的建议、OASIS 公司所作的 ebXML/TRP 和 IBM 公司所作的 HTTPPR。W3C 正在解决所有这些领域中的标准化问题。Key industry players 形成了称为 WSI 的竞争者联盟。但是，他们还没有解决动态安全协议问题。

在 ebXML CPP 和 CPA 中，对于它们在单个登记表中的服务，各方为它们的互通性规则互通定义了称为 CPP 的简档（profile）。两个简档可以相交，以推出称为 CPA 的默认互通性协议。可选地，双方可以商定称为 CPA 的相互之间的一组特定互通性规则。ebXML CPP 和 CPA 带来的问题包括：它们假设发送方和接收方处在同一登记表中。互通性规则不足以覆盖互通性的许多方面。在使用的时候，它们假设 CPA 的签名副本（由双方签名）保存在登记表中。使维护和修改变得非常麻烦。这正好与动态计算互通性协议不协调。于是，在一个服务调用另一个服务的运行内，不是借助于高速缓存来解决动态计算，而是涉及预下载和本地安装，使得对 CPA 的管理改变既困难又非自动。

于是，出现了开发为贸易伙伴动态确定互通性协议的方法和设备的机会。

## 发明内容

本发明涉及机器可读数据结构和数据结构的动态计算以支持互通性。更具体地说，本发明涉及提高互通性的数据结构和数据结构的动态生成这些方面。在权利要求书、说明书和附图中描述了本发明的具体方面。

## 附图说明

图 1 例示了共同体和共同体的网络，它们是机器可读、动态协商互通性合同可用的一个环境；

图 2 例示了共用同一连接器以支持不同传输/封装协议和技术的多个轴心和辐条机构；和

图 3 例示了当发送器对于安全、转换和其它措施的计算是本地的时，获取接收器信息的可替代实施例。

## 15 具体实施方式

下面参照附图作出详细描述。描述优选实施例是为了举例说明本发明，而不是限制权利要求书所限定的本发明的范围。本领域的普通技术人员应该认识到，可以对如下的描述作出各种各样等效的改变。

图 1 例示了共同体和共同体的网络，它们是机器可读、动态协商互通性合同可用的一个环境。

在这些共同体中，一个共同体维护本地登记表，本地登记表包括诸如共同体组成部分的用户、公司、服务和连接器之类的信息。共同体可以属性一个或多个共同体网络。通常，共同体和网络存在一些共同的商业利益。互通性是一个或多个网络中的成员共同体之间。网络包括黄金市场网络 1、贵金属市场网络 2、专用网络 3 和全局贸易万维网 4。在这个示例中，黄金市场网络 1 和贵金属市场网络 2 包含在全局贸易万维网 4 中。贵金属市场网络 2 包括黄金和白银市场 14 和 13。黄金市场消费者可以在白银市场 13 中买卖白银，和白银市场消费者可以在黄金市场 14 中买卖黄金。一个共同体 - PQR 企业 17 属于黄金市场网络 1、专用网络 3 和全局贸易万维网 4；另一个共同体 - ABC 大供应商 18 属于专用网络 3。在这个示例中，XYZ 黄金部分 14 是买卖黄金的市场或共同体。企业属性这个共同体。像 PQR 企业 17 那样自己形成

一个共同体的企业属于黄金市场网络 1。这些共同体是黄金市场网络 1 和全局贸易万维网 4 的组成部分。小供应商 15 是黄金市场共同体的组成部分。其它企业 16 是黄金市场共同体的组成部分。XYZ 黄金部分 14 和其它黄金市场实体 15-17 之间的连接表示黄金市场要求参与黄金买卖的企业（或共同体）5 之间的所有业务都要经过 XYZ 黄金部分 14，例如，以收集结帐和商业情报信息。PQR 企业 17 是黄金市场的组成部分，和与供应商 18 一起也是局部专用网络的组成部分。小供应商 15 可以是本身不想形成共同体，而是将诸如用户、机构、服务和转换之类它的元数据登记在黄金市场的登记表中的个别小供应商。另一方面，ABC 大供应商 18 形成它自己的专用网络，例如，由于10 它想要保存因开发它们相当昂贵而不允许一般公开访问的元数据、机构内系统和转换。由于 PRQ 17 是 ABC 18 的客户，所以它加入专用网络 3 中。金融服务提供商 DEF 金融部分 12 想要向全局贸易万维网 4 中的任何人提供金融服务，这样就形成它自己的共同体和与全局贸易万维网顶级部分 11 平起平坐。共同体的网络使共同体的全局登记表可供使用。全局登记表允许发现共15 同体和确定到那个共同体，或到可以路由准备到共同体的电子商务文档的外部连接器的一条或多条路线。从一个共同体路由到另一个共同体的文档可以在两个共同体的外部连接器之间直接路由，或者通过一个或多个中间共同体间接路由。在共同体登记表中也可以定义和维护涉及共同体的交易的商业和安全规则。一般说来，图 1 例示了产生对电子商务平台之间的互通性的推动力的实体和共同体的混合忠实性。

连接器是与其它应用程序通信的应用程序的一般术语。连接器可以对等（P2P）地或直接地通过起集线器、网关、外部端口、轴心连接器等作用的其它连接器通信。P2P 通信的连接器能够与使用相同传输/封装协议的其它连接器通信。当试图与不使用相同传输/封装协议的其它连接器通信时，P2P 通信25 的连接器可以有选择地谋取进行翻译服务的其它轴心连接器的帮助。直接通信的连接器根据路由规则，通过轴心连接器通信。在有向图中可以映射连接器之间的路由规则，对于一种或多种传输/封装协议，支持一种或多种轴心和辐条拓扑结构。轴心和辐条拓扑结构分一层或多层使通信沿着辐条指向轴心。这有助于诸如结帐、商业情报收集、跟踪、审计、计费等的集中服务。像图30 2 所建议的那样，多个轴心和辐条机构可以共用同一连接器，以支持不同传输/封装协议和技术。例如，可以要求较强的轴心和辐条机构将 Sonic 用作传

输技术，而不是使用 HTTP 或 HTTPS。可选地，通信路由可以取决于源和目的地是否是同一共同体的组成部分。在子共同体（它可能包括跨越共同体）内，可能不需要集中功能，并且当与其它子共同体中的目的地通信时，在以另外的方式指向其以便与母连接器通信的连接器之间允许 P2P 通信。

5 连接器可能被标记成简单连接器（有时简称为连接器）、集线器（有时称为网关或路由器）或中心连接器。可替代地，可以在功能上描述它们。通过轴心连接器指向简单连接器通信，除了允许它们在相同子共同体中的连接器之间进行 P2P 通信之外。所谓的集线器供明显指向它们或与它们链接的连接器使用。集线器可以提供不止一种功能，于是，在从源到目的地的路线中可能表现得不止一个。集线器转发电子商务文档或消息。集线器还可以翻译支持公用封装协议的传输协议。例如，集线器可以翻译封装协议，并且，根据发送而不是根据接收实现不同的传输协议。中心连接器是集线器的特殊情况，它可以供没有明显指向它们或与它们链接的连接器使用。中心连接器可用于例如根据路由规则从源开始穿过连接器没有通向支持目的地使用的传输/封装协议的任何集线器时，完成翻译功能。  
10  
15

本发明的这些方面解决了联合登记表、互通性合同数据结构的成分和互通性合同的动态协商。登记表的范围是共同体。共同体可以是企业、市场或较大分散企业中的子企业。互通的各方可能处在不同共同体中。例如，一方可能处在供应者共同体中，另一方可能处在购买者共同体中。因此，应该使用存储简档和协议的联合方案。关于互通性，本发明胜过 ebXML 和解决电子商务互通性的其它传统手段。互通性合同被推广到包括如下的组合：在服务之间传送消息时要沿着走的路线，和遵从定义的路由规则。例如，一条规则可能阐述到/来自万维网服务的所有消息应该由特定路由器置顶。该路线包括对于譬如 SOAP 和 EDI 封装协议之间的封装转换，通过网关的自动路由。由于在消息中可能存在多个部分，指定基于逐个消息部分的信号发送、验证和加密策略，其中，该策略包括算法、技术（例如，XML 加密、SMIME、PKCS #7）和元素（例如，XML 文档中的 XML 元素）。一个部分一个部分地为包括在消息部分中的文档指定转换规则。例如，是否允许版本互通性的转换，如果这样的话，是否应该附上原文。还可以标识特定转换逻辑。标识要使用的消息交换安排的版本，例如，一种服务可能支持多种版本的安排，因此，服务知道发送器和接收器支持的正确版本是有益的。某些消息传递策略被设  
20  
25  
30

置成譬如是否存档消息、是否使用可靠交付和是否要求不拒绝接受确认。需要桥接的发送和接收消息之间的差异通过封装调整或封装转换来解决。例如，使用的不同封装外延、消息部分顺序的差异和不同的封装协议。路线中提供与连接器的能力相符的各种各样互通性功能的连接器被登记在登记表中。本  
5 领域的普通技术人员应该认识到，在许多有用的子集中可以组合本发明的前面和后面这些方面，且本发明无意局限于包括本发明所有方面的互通性合同。

互通性合同一般通过使发送器和接收器服务的策略和接口相交导出。但是，也可以使用指示应该用于解决发送器和接收器之间的冲突的判定规则的越权。例如，判定规则可以确定发送器取胜、接收器取胜和最严格策略取胜  
10 等。由于互通性合同由可信服务譬如值得信任的共同体顶级方 (root party) 计算和签名，这对于支持服务修改是有用的。互通性合同可以当调用服务时动态地计算，并且可以本地地高速缓存在消息发送站上。

当调用的服务处在一个共同体中和被调用的服务处在另一个共同体中时，可以由从对于两种服务都是本地的共同体登记表中获得发送方和接收方  
15 信息和使其相交的分布式逻辑单元进行合同计算。如果以前没有批准，将在接收方和发送方定义的任何越权 (override) 加上注释，复制给辅助方，加以批准。在创建合同时，由于交叉共同体在线协商进程可以用于解决越权，分布式地计算的所得互通性合同完全相同地结束。服务编写者觉察不到非常复杂，和从登记表中自动推出互通性合同。这极大地简化了服务开发。  
20

应该动态计算互通性合同，以便通过将合同安装在本地机器中来避免主要同步问题。由于高速缓冲存储器可以辅助动态计算，这未必要求对每个消息进行重新计算。通过登记表中有关变化的无效通知或过期策略可以使高速  
25 缓存保持连贯。高速缓存保持者可以预约任何所需通知。一旦从发送服务连接器 (或它的知道如何管理它的代理器) 开始万维网服务，就可以动态地计算互通性合同。计算之后可以将合同附在消息上，以便通信服务之间的中间连接器明白它们在消息交换中的作用。例如，合同可以指定哪个连接器应该进行版本转换、信号发送和加密等。

本发明的这些方面可以推广到多维的互通性。对于真正的端到端消息互通性，存在多维的互通性需要解决。解决任何一维的互通性都推进利用基于  
30 万维网的服务的电子商务。解决互通性组合问题可以产生显著推进。在如下的讨论中，给出了十几维的互通性和在本发明范围内的解决方案。

### 1. 传输协议互通

互通的一维是传输层互通。根据在相关应用中更全面讨论的本发明的这些方面，将允许的和支持的传输协议与使用的封装协议结合在一起。相反，在 ebXML 中，允许的传输协议是 HTTP(S)。对于 MML，允许的传输协议是 5 Sonic。对于 Biztalk，允许的传输协议是 HTTP(S)。在本发明的一个实施例中，允许的传输协议是 HTTP(S)和 Sonic。

对于 Sonic，可靠性被下推到传输层。Sonic 可靠性协议是极好的算法。如果被标准化，长期 HTTP(S)R 可以提供 HTTP(S)层本身上的可靠性。ebXML 和 Biztalk 现在使用的解决方案是在 HTTP(S)上存在封装协议相关措施的可靠性。在未作扩展的情况下，SOAP 不提供对封装层上的可靠性的支持。10

本发明包括对协商受支持协议当中的传输协议的支持。在一个实施例中，这涉及到 HTTP(S)和 Sonic 之间的选择。由于其它传输协议也适用于电子商务，因此，本发明可以使那些附加选项包括在协商中。

### 2. 封装协议互通

15 在本发明的一个实施例中，支持的封装协议是 MML、C1 SOAP、email 和 external SOAP，它允许像 C1 address、conversation and message info、manifest、SAML 和 SOAP 那样的可选扩展与附件任意组合。在企业中用 pure SOAP、SOAP WA、standard WSDL 和 discoverable with UDDI 展示的服务被称为简单万维网服务。但是，虽然与作为简单万维网服务（借助于第三方开发环境和第三方执行环境开发的）的端点互通，C1 SOAP 还支持具有可靠性、20 安全性和参与双向安排的本地万维网服务。用 J2EE CA 或 EJB 展示的机构内系统可以由第三方包装成简单万维网服务。这个实施例可以与它们以及支持的电子邮件协议和 external SOAP 互通。

支持的协议定义允许的传输、可靠性和安全协议。它们还定义在那种协议下阐述服务和各方的方式和将相关消息结合在一起的数据连接。消息路由 25 和调度基于定义的地址。

互通性合同可以支持封装协议确定和转换。这是互通性合同远远超过典型 ebXML CPA 合同的方式之一。此外，互通性合同可以包括有关要沿着走的路线、要完成的转换和在什么地方完成转换、要签名和加密的东西和在什么 30 地方做和使用什么算法、安排的名称和版本、以及发送/接收 TP/服务/服务版本/操作的信息。互通性合同可以用于沿着服务之间的路线导出中间连接

器。参与服务之间的线段是即使端点严格地遵守没有使用本专利的受让人开发的软件的标准，也可以附加数值的所谓“智能互通网络”。

封装协议之间的互通通过网关。同一协议的不同版本可以当作不同协议来对待。路由器知道通过适合互通的一组网关透明地路由消息。目的地连接器中的调度器将入站消息交给适当部件。这种再次调度基于通过目标地址和其它封装字段导出的规则。  
5

封装协议互通的一种变种是我们拥有存在基线和可以使用的多个选项的协议。一个例子是 external SOAP，带有附件、路由、安全和 SAML 等的 SOAP 是可选的。如果发送器指定一组选项和接收器指定另一组，进入网络点将计算互通是否可能，如果可能的话，如何互通。根据规则自动地附加可选块和去掉网络退出点上的无用块。  
10

当我们将来自 SOAP 正文的 XML 数据转换成 MIME 部分中的文档或反过来时，我们认为这是封装互通的一种形式。这样的转换发生在 Biztalk 和 ebXML 之间或在 SOAP 和 ebXML 之间转换的时候。对于发送器将有效载荷放在附件中和接收器期望它在正文中（或反过来）的 SOAP 到 SOAP 互通，  
15 也会发生。

### 3. 安全协议互通

封装协议互通存在的一个问题是在于支持的安全协议通过封装协议来定义和安全协议之间的转换几乎是不可能的。例如，从封装协议 A 支持 XML 签名转换成封装协议 B 支持的 PKCS #7 是不可能的。如果接收服务要求原始签名或对互通加密，网关应该将错误返回给发送器，除非信任网关转换安全协议。一种克服安全协议不兼容的手段是信任网关核实消息中的签名，和解密（加密者使用网关密钥），重新签名和重新加密消息。制定信任方案，从而接收器可以信任网关签名。  
20

企业建议的 SOAP 扩展版是 WS-security (GXA 的组成部分)。本发明的实施例可以支持 WS-security，包括用于 C1 SOAP 的 WS-security。此时，这样的安全扩展是可选的，如果外地方维网服务没有采用 WS-security，可以将签名权限授予进入互通网络点和代表它加密消息（进入点访问用户密钥）。如果进入网络点位于存在外地方维网服务的企业内，就这样工作。  
25

在没有首先得到验证的情况下（除非被调用服务明白地表示不在乎），不应该接受到互通网络的消息。  
30

安全协议互通的一个方面是当发送器和接收器指定不同安全策略和能力时。互通框架必须计算互通是否可能，如果可能的话，如何互通。

#### 4. 不同类型服务之间的互通

通常，除非另有说明，服务被登记在协同登记表中。在这个讨论的背景  
5 下，预期协同万维网服务通过至少一个接口与另一个协同万维网服务交互。

存在所谓的简单万维网服务、高性能万维网服务和协同万维网服务。还  
存在本地万维网服务和外地万维网服务。最后，存在登记服务和未登记服务。  
简单万维网服务不使用签名、加密、可靠消息传送，和不要求来自中心可信  
10 方的验证。也不支持双向安排。换句话说，简单万维网服务的每一次调用都  
独立于简单万维网服务的以前所有调用，并且，没有保留在简单万维网服务  
中的安排背景，和不知道在这种背景下的返回地址，因此，以后可以回答。  
高性能万维网服务可以包括更高的可靠性和安全性。协同万维网服务可以是  
简单的或高性能的，另外还支持双向安排。除了由本申请的受让人预备的那  
些（外地万维网服务）之外的万维网服务都是简单万维网服务。

15 正如本申请和引用的申请通篇所述的那样，本发明的这些方面可以以许  
多种方式推广电子商务的机制。与高性能万维网服务和协同万维网服务一样，  
新的万维网服务可以登记在协同登记表中。可以为可以逐个加入单元的本地  
简单和高性能万维网服务之间的连续性提供支持。高性能万维网服务可以在  
20 登记表中声明它支持什么单元。可以下载新本地简单万维网服务（从 UDDI  
或从 Commerce One 自己的协同登记表）的 WSDL 定义，WSDL 定义标识作为  
进入网络点的 URL 的服务端口。通过进入端口传送的消息将自动从那里路  
由到它们的逻辑目的地。按照本发明路由的消息包括互通性合同或受互通性  
合同支配，互通性合同支配每个跳段将发生什么。本地万维网服务可以调用  
本地或外地简单万维网服务。

25 新的网络可以支持外地简单万维网服务。如果外地万维网服务知道新的  
寻址和消息身份和相关 SOAP 扩展版，它甚至可以作为协同万维网服务参与  
双向安排中。外地万维网服务可以使用新 SOAP 扩展版的组合。它们不需要  
访问共同体登记表或了解互通性合同。本发明可以推广到提供建立外地万维  
网服务的软件和应该使用第三方软件。外地万维网服可以供任何本地万维网  
30 服务或任何其它外地万维网服务通过我们的网络来调用。外地万维网服可以  
使用 external SOAP 或 email。在 email 的情况下，使用电子邮件浏览器的人

员可以“实现”万维网服务和与简单和协同本地或外地万维网服务互通。可以从协同登记表或从 UDDI 下载外地万维网服务的 WSDL 定义。外地万维网服务通过调用进入网络点上的 URL 调用新网络中的万维网服务。协同外地万维网服务通过本地协同万维网服务提供 SOAP 扩展版中进入我们网络点的 5 URL，作为调用的一部分，因此，如果它了解 SOAP 扩展版，以后可以动态地作出回应。

### 5. 网络和位置无关互通

目的地服务部件的位置应该无关紧要，和其中登记服务的市场或企业共同体也应该无关紧要。路由算法应该透明地管理位置透明性和市场或企业共同体透明性。沿着传输和安全机构的道路应该支持在安全方面不打折扣地自动穿越适当的企业和市场防火墙。 10

### 6. 平台无关互通

平台可以包括软件运行的硬件/操作系统和商业服务运行的服务器的开发和执行环境。它还可以涉及软件运行的服务器技术（J2EE 应用服务器、万 15 维网服务器或小程序运行器）。无关的硬件部分可以利用 100% 纯度的 Java 来实现。与开发/执行环境的无关性可以通过支持与外地连接器和服务器的基于严格标准有线层互通来实现。服务器技术无关性可以通过使部件可嵌入和遵从 J2EE 标准来实现。

当销售商供应的部件是平台无关的时，消费者可以利用来自它们优选的 20 喜欢销售商和与它们喜欢的客户机方工具相联系的它们优选开发/执行环境开始服务。这样的服务借助于智能网络加入的互通值仍然可以与销售商开发的服务互通，和利用进程流引擎可以将所有服务组合成具有复合能力的更复杂服务。

轻型商用万维网服务服务器主要根据消息互通部件来部署。轻型（light 25 weight）服务器将是供应商连通、网关编写者和 ISV 市场的目标。协同万维网服务服务器的更完整实施例是轻型版的一个超集。轻型版包括用于文档相关开发的基本开发工具，但主要补充支持用于服务开发的第三方工具。对于协同万维网服务服务器实施例，可以包括用于基于 UI 和文档进程中心自含或组合服务的复杂全面开发环境。

### 30 7. 机构内系统互通

与外地连接器互通的一个方面是与机构内系统的互通。本发明的这些方

面允许机构内系统展示出来，以便从消息传递层和从发现层看来就像数个服务一样。工具箱使机构内系统操作者将他们的接口展示成简单万维网服务，或者将它们的定制适配器包装成万维网服务。定制综合中介器能够将建立的 EAI 技术与新消息传递系统结合在一起或直接构建万维网服务接口。与机构 5 内系统结合的另一个实施例是电子邮件支持。电子邮件服务器可以用于将机构内系统与新网络结合在一起。

将机构内系统展示成万维网服务可能涉及不基于 XML 的专业转换方案。例子是 DB 和 XML 或 XML 和平面文件之间的转换，或 J2EE CA 1.0 记录结构和 XML 之间的转换。所有这些都对下游万维网服务保密和对下游万维网 10 服务开发者是透明的。

#### 8. 服务发现和跨越共同体互通

将来，很有可能贸易伙伴之间的互通将变得更加动态。发现机制可用于在建立商业关系之前，找出要做生意的贸易伙伴。服务和提供服务的贸易伙伴的发现是通过 UDDI 标准完成的。UDDI 支持的更强大工具调用新登记表 15 万维网服务。与本发明有关的发明提供将数据上载到公共 UDDI 登记表或上载到对于一个共同体或一组共同体用作黄页的专用 UDDI 登记表的支持。在 跨越共同体的网络内发现是可能的。

对于跨越共同体的发现，每个共同体可能拥有全局白页共同体的列表或与之相联系的全局黄页登记表。全局白页共同体包含将请求路由到一组被广告的共同体的传输地址。全局黄页登记表包含贸易伙伴和一组被广告的共同体的服务，以及别名和类别。按类别进行搜索。由于互通是双向的，两个共同体可以预约公用全局白页共同体或在它们的共同体登记表内拥有直接到彼此的路由信息。如果两个共同体预约公用黄页登记表，它们可以在彼此中发现对象。通常，黄页登记表寄宿在白页共同体内部。

25 不仅对于发现，而且对于包括角色和特权的贸易伙伴信息、以及用户与机构及其它它们的关系，也支持编程登记表访问接口。此外，存在对获取包括 WSDL 文件、服务接口、转换代码和模式文件的有关互通的技术信息的支持。

#### 9. 登记表版本互通

登记表服务可以被配置成其它万维网服务，并且，可以从所有服务的互 30 通性支持中获益。

#### 10. 文档语义互通

基础设施不关心有效负载的语义。但是，文档语义互通允许使用不同文档的服务享用端到端互通。发送器和接收器必须承认文档语义，譬如，文档族成员和成员之间的转换，以促进互通。对于与机构内系统的互通，文档标准可以包括 Idocs 和 OAGI。

## 5 11. 文档版本互通

服务中接收操作的接口可以定义对一个或多个版本的文档的支持。新版互通系统转换发送文档和要接收的预期文档，并且，设法通过挑选最佳接收版本降低损失。转换发生在发送方签名和加密消息之前。

10 登记表支持文档族中的主要和次要版本。主要版本可以遵从不同模式语言。预计次要版本将可选部分加入基本版本中。

## 12. 模式语言互通

15 有效负载 XML 文档的模式语言通过封装协议来定义。模式语言的例子是 SOX 和 XSDL。这些是描述 XML 文档的模式的语言。一种语言下模式的 XML 事例不同于另一种语言下等效模式的 XML 事例。因此，模式语言事例转换应该得到网关中的转换支持。

网关可以进行有效负载（单元之间的关系）和语义的结构不改变，但语法和包装发生改变的所谓语法转换。将兼容结构转换成完全等效的 XML 标记，或反之。

## 13. 对互通步骤的位置和顺序的依赖

20 从这个讨论中，本领域的普通技术人员将看到，互通性合同是保证在商定位置上和按商定顺序执行互通步骤的一种方式。从发送器到接收器的消息穿过不同连接器为互通执行各种步骤的一系列连接器。在模式语言事例转换、版本转换、封装转换、签名和加密的位置和顺序之间存在相互影响。基础设施适当地对这些转换排序。

## 25 14. 服务版本互通

30 万维网服务通过它们外在如何表现，根据它们的登记表描述在将消息交给它们时定义。随着时间升级服务和改变服务版本是自然的事。服务的新版本可能将操作加入现有消息中，或者，将可选部分加入现有消息中或从现有消息中删除可选部分。也有可能改变支持的一组安排和某部分在消息中的位置。所述的安排互通可以用于使发送器知道他们是否应该调用新操作。另外，使服务的版本号为发送器和接收器所知，因此，它们可以适当地作出响应。

当该组可选部分不同时，或者，当正文部分变成附件或反过来时，基础设施关心互通。

### 15. 安排互通

至少存在两个支持安排的实施例。一个实施例定义进程流和让所有参与者通过这个进程运行它们的消息。在服务中进程流在进程流引擎中运行。另一个实施例支持知道端点服务本身中的安排细节的端点服务之间的直接消息传送。进程流引擎进程借助于其它服务发送和接收消息，因此，可以使它看起来像服务本身。这种抽象非常有用。

进程流引擎进程应该表现得像服务。想要与进程交互的应用将消息发送给这个服务和从这个服务接收消息。由于这种抽象，进程流引擎进程也可以用于利用进程定义组成更大服务，以便将一组服务结合在一起形成流和展示更大服务。此外，可以使进程流引擎进程用在每个新服务中，因此，可以构建跨越多个进程引擎的分布式进程。这是可能的，因为分布式进程中的每个子进程看起来就像一个服务，和就像一个服务调用另一个服务那样对待一个子进程调用另一个子进程。各种子进程与消息交互和消息携带可用在每个子进程中的进程流前后关系，以便更紧密地合并子进程。

考虑服务之间的双向安排是知道发送 TP/服务/操作的能力，特别是当服务之一不直接支持安排或会话 ID 扩展时。将相关消息与会话 ID 关联的方法是有用的。利用有效负载数据使形成会话的相关消息关联，可以与不支持安排的简单万维网服务进行虚拟会话。进程流引擎包括进行关联的逻辑单元和资源。对于来自没有寻址扩展的外地连接器（通常，机构内系统就是这样）的消息，可以将消息发送给寻找有效负载的固定服务、登记表或本地数据库，以便在继续转发消息之前推出目的地地址。这种能力被称为逻辑路由，并且，根据从中可以推断会话 ID 的、要检查的有效负载中的字段的配置说明，进程流引擎起促进作用。

### 16. 安排变种互通

安排将参与者提供的一组服务类型结合在一起。安排的所有变种形成第一消息基本相同的族。应该存在唯一一个在互通的两个服务之间受到支持的族和可以按偏好排序那个族中的安排。但是，一个服务可能支持涉及不同服务组合的多个安排族。

在安排协商的一个实施例中，当发送安排中的第一消息时，告诉发送器

和接收器由系统挑选的安排变种。然后，这些之间的安排可以不发生改变。然后，它们据此调整它们的处理。如果将一个新服务加入会话中，发送服务可以在起多极安排中不同服务支持的安排变种之间的桥梁作用和强迫使用所选安排之间作出选择。在引用的共同拥有的申请之一中进一步描述了安排协商。  
5 商。

### 17. 隐藏互通的复杂性

由于可以隐蔽地关心复杂问题，按照本发明这些方面交互的服务可能需要知道一点或一点也知道互通。可以配置实现互通的新模块。这些模块关心与通过登记表元数据导出的互通有关的复杂问题。可以提供 API 抽象概念来完全隐藏封装结构和尽可能多地隐藏特定字段语义和语法。所有安全策略都可以包括在互通性合同中，从而简化了服务开发者实现应用的工作。  
10

### 18. 限制互通的机制

直接互通的一道障碍是安全。模型是基础设施验证发送器和服务根据登记表捕获的元数据有可能授权它。这些障碍包括商业规则、预约和隐式服务。  
15 商业规则有时应该限制几个共同体之间或一个共同体内的互通。正如提供者服务策略所指的那样，在互通之前可能要求预约。拥有在共同体外面看不见或只有特定方看得见的隐式服务也是有用的。

图 2 例示了生产者服务和消费者服务之间的动态协商互通性合同的有用性。该图的主要特征包括登记表 201、包括动态确定互通性合同的逻辑单元的万维网服务引擎 202、向内部进程流 204 展示安排化接口的生产者服务 203 和消费者服务 205。图文表示这个例子涉及接收定单的定单接收系统。生产者和消费者服务对于安排、服务版本、文档、安全验证、安全加密、安全签名、封装协议和传输 213 和 215，拥有它们自己的能力和策略。动态协商互通性合同降低了建立或维护服务网所需的成对配置的程度。它为解决参与者设置的策略之间的差异提供了无歧义规则。随着参与服务不断演变，动态协商互通性合同也不断演变。  
20 25

互通性合同的动态协商呈现出与更接近合法合同协商的传统手段的显著偏离。动态协商从生产者服务对它的可用性、能力和策略的描述开始。消费者服务利用诸如 UDDI 之类的发现协议，可以容易地发现产生者服务。生产者和消费者拥有它们的能力和策略的机器可读说明。生产者和消费者识别的一种或多种模式无歧义地定义如何翻译各方的能力和策略和找出相交部分。  
30

取代要求通过协商来解决不同的互通性条款，系统提供与如何解决两种冲突有关的判定规则：对可替代选项的偏好之间的冲突和有关是否将诸如签名和加密之类的安全措施应用于根据动态协商互通性合同交换的消息的特定部分的冲突。有关偏好的判定规则可以是标准规则，譬如，接收器取胜，接收器  
5 取胜，最严格要求取胜，最不严格要求取胜，或应用对双方喜欢的加权考虑。有关是否应用例如安全措施的判定规则是相似的。在与本申请同时提出和特此引用以供参考的“万维网服务之间的安全措施的动态协商（Dynamic Negotiation Of Security Arrangements Between Web Services）”专利申请中进一步讨论了包括越权的这些判定规则。在一些事例中，生产者可能要求在消费者  
10 可以与生产者交互之前进行预约。这可能有助于信用和验证检验等。相交和判定规则的框架使可信软件代理可以动态地协商互通性协议，特别是当生产者已经接受预约时。这种授权动态协商互通性合同的可信软件代理的使用是对生效之前由生产者和消费者两者用密码签名的更传统 CPA 风格互通性协议的明显违背（虽然这种描述是针对生产者和消费者服务展开的，但为了帮助读者  
15 理解，可以平等地应用于两种或更多种服务，与它们作为生产者、消费者、中间人等的角色无关）。

一组模式和示范性互通性合同提供了与本发明这些方面有关的附加细节。

源代码附录中的模式 Interoperability.XSD 可以用于模型化包括本发明几个方面的互通性合同。在这个实施例中，机器可读输出文件是 XML 文档。在其它实施例中，其它数据结构可以用于存储相同信息，例如，在 XML 代码之后模型化的树结构。通过将文件装入诸如 XML Spy TM 之类的综合开发环境（IDE）中可以最好地理解模式 Interoperability.XSD，XML Spy TM 提供模式的几种可替代视窗，包括文档生成视窗。在 Spy 模式设计视窗中观察到，  
20 Interoperability.XSD 部件包括一般合同部分、路由合同部分、转换合同部分、安全合同部分和合同签名。正如下面所讨论的，四个部分每一个都通过引用另一种模式插入。与传统互通性合同不同，信任协商合同的软件代理应用合同签名。不需要各方单独对合同签名。正如在源代码中进一步文档化的那样，  
25 合同签名部分包括 SignedInfoType、SignatureValue、KeyInfo 和 ObjectType。

30 也在源代码附录中的模式 GeneralContract.XSD 可以用于模型化包括本发明几个方面的互通性合同的一般部分。GeneralContract.XSD 部件包括来往信

息、ErrorHandling 和 DeliveryReceiptHandling。部件可选地包括 RequiredMessageParts 和 OptionalMessageParts 以及发送和接收连接器能力。来往信息与所涉及的各方/服务/活动有关。ErrorHandling 部件描述能力和可选地标识发送错误消息的地方。与 ErrorHandling 一样，Delivery-ReceiptHandling 是带有消息的可选地址的能力参数。交付收据用于实现非拒绝。必选消息和可选部分就像命名的那样。在引用的申请中更全面地讨论了在服务版本确定和文档族版本确定中必选部分和可选部分的作用。发送和接收连接器能力列出了连接器的属性和属性的值（譬如，能够签名或加密）。能力是可选的，因为对于非协同请求或对于单向消息它们可能不出现。在源代码中进一步将这些部件文档化。

也在源代码附录中的模式 RoutingContract.XSD 可以用于模型化包括本发明几个方面的互通性合同的路由部分。在 Spy 模式设计视窗中观察到，RoutingContract.XSD 指定一条路线。路线包括路线中的两个或更多个 Route Nodes，包括发送器和接收器。进入节点和从节点退出的信道通过用于到达节点和从节点退出的传输和封装协议来定义。这个信息的对称性使退出和进入信道对于相反的路线起相反的作用。在源代码中进一步将这种模式文档化。在引用的申请中更全面地讨论了路由。

正如在同时提出的申请之一中所阐述的那样，安全措施的协商是通过将发送和接收服务的安全简档用于确定相互同意安全措施的基于计算机进程实现的。最好，无需用户介入，定期协商或潜在地更新这个安全措施。每当交换消息时或基于一些其它周期或偶发事件，譬如，每月、每周、每日、发生影响消息在特定发送器和接收器之间的交换的事件（例如，软件部件出现故障或安全偏好发生改变），并且当以前协商的措施失效时或基于一些其它周期或偶发事件，可以在用户的请求下或无需用户介入协商、更新或有效性检验这种措施。源代码附录中的模式 SecurityContract.XSD 可以用作准备机器可读安全互通性合同文档的模型。在这个实施例中，机器可读文档是 XML 文档。在其它实施例中，其它数据结构可以用于存储相同信息，例如，在 XML 代码之后模型化的树结构。这种模式定义了策略和安全策略的通道。安全通道定义资源和到执行安全算法，譬如，签名、加密和验证算法的资源的路线。它也可以包括不拒绝和验证资源。

下面部分再现了一组计算的安全措施：

```
<SecurityContractICD ... >
  <SecurityPolicies>
    <SignaturePolicies>
      <XMLEdgPolicy PolicyId="P-XMLSignatureRSA-MD5-C14N">
        <SignaturePolicyAlgorithm>...</SignaturePolicyAlgorithm>
        <SignatureAlg...>MD5withRSA</SignatureAlg...>
        <HashFunction>MD5</HashFunction>
        <Canonical ...>...14n-20001026</Canonical ...>
        <Transform>...#RoutingSignatureT...</Transform>
      </XMLEdgPolicy>
    </SignaturePolicies>
    <EncryptionPolicies>
      <XMLEncryptionPolicy PolicyId="P-XMLEncrypt3DES-RSA-2048">
        <EncryptionPolicyAlgorithm>http://www.w3.org/2001/04/xmlenc#</EncryptionPolicyAlgorithm>
        <EncryptionMethod>http://www.w3.org/2001/04/xmlenc#3des-cbc</EncryptionMethod>
        <KeySize>2048</KeySize>
        <KeyEncryptionMethod>http://www.w3.org/2001/04/xmlenc#rsa-1_5</KeyEncryptionMethod>
      </XMLEncryptionPolicy>
    </EncryptionPolicies>
    <EncryptionKeyInfo KeyOwner="x-ccns:commerceone.com:CollaborationParty::sellParty">
      <PublicKeyID>DefaultTestCert</PublicKeyID>
      <X509Data>          <X509Certificate>LS0tLS1... ==</X509Certificate>
      </X509Data>
    </EncryptionKeyInfo>
  </SecurityPolicies>
  <SecurityChannel channelId="CHANNEL1" sourceConnector="x-ccns:cup.commerceone.com:connector::centerSell" targetConnector="x-ccns:cup.commerceone.com:connector::centerSell">
    <Confidential AlgorithmId="P-XMLEncrypt3DES-RSA-2048">
      <PublicKeyName KeyOwner="x-ccns:commerceone.com:CollaborationParty::sellParty">DefaultTestCert</PublicKeyName>
      <MessagePart PartName="Order" isOptional="false"/>
      <MessagePart PartName="Image" isOptional="false"/>
    </Confidential>
  </SecurityChannel>
</SecurityContractICD>
```

```

</Confidential>
</SecurityChannel>
<SecurityChannel channelId="CHANNEL2" sourceConnector="x-
ccns:cup.commerceone.com:connector::buy" targetConnector="x-
ccns:cup.commerceone.com:connector::sell">
    <Integrity AlgorithmId="P-XMLSignatureRSA-MD5-C14N">
        <PublicKeyName KeyOwner="OwnerA">BuyerPublicKey</PublicKeyName>
        <MessagePart PartName="Order" IsOptional="false"/>
    </Integrity>
</SecurityChannel>
</SecurityContractICD>

```

这组安全措施拥有安全策略和安全通道两个主要部分。在本例中，存在一个可应用于整个消息的安全策略和实现安全策略的各个部分的多个安全通道。安全策略部分展示签名策略以及加密策略和加密密钥信息。它也可以展示与验证、授权和不拒绝发送或接收有关的信息。在本实施例中，同一签名和加密策略应用于文档的所有部分。在其它实施例中，多种算法可应用于不同部分或一个部分内的不同单元。为签名、加密和验证选择的算法通过包含选项集的模板抽象化，从而简化算法的选择。所选算法与逻辑单元和资源相联系，因此，不同服务或进程可以用于签名/核实和加密/解密消息的不同部分。

可以在安全策略部分的加密密钥单元中发送公开密钥或证书。安全通道部分描述应用安全策略所涉及的服务或连接器。对于特定策略，通道部分标识要求有助于应用安全策略（例如，发送服务请求加密）的源连接器和应用安全策略或起应用安全策略的逻辑单元和资源的中介作用的目标连接器。对于特定安全策略，譬如，签名、加密、验证、授权或不拒绝，在安全通道部分中提供执行安全策略所需的特定信息。

图 3 例示了当发送器对于安全、转换和其它措施的计算是本地的时，获取接收器信息的可替代实施例。在该图中，标出了本地登记表 331 和远程登记表 332。在本例中，发送器是本地的和接收器是远程的。在本地登记表 331 中发送器数据是当前的和完整的。收集 (321) 发送器信息和使它可用于计算 (311) 安全措施的逻辑单元和资源。接收器数据可能是当前的和完整的，例如，如果接收器处在与发送器相同的共同体中和存在全共同体的登记表，或者，如果最近已经获得和本地高速缓冲了接收器信息。取决于可以找出 (331 或 332) 接收器信息的地方，调用进程 322 或 323，以收集接收器信息和使它

可用于计算安全措施的逻辑单元。得出一组安全措施 301。

可能需要协调两种类型的偏好。可以陈述为共同体特有偏好和服务特有偏好两者。一种偏好是在算法模板之间。在选项 B 和 D 之间作出选择的判定规则可能要考虑消息传送服务偏好之一或两者。例如，可能从匹配当中选择倾向于签名的接收服务偏好 (D) 和倾向于加密的发送服务偏好 (B)。对这两种偏好加以考虑，可以选择最严格的 (B) 或最不严格的 (D)。在另一个实施例中，各自服务可能加权它们的偏好或对它们的偏好打分，组合权重或分数可以用于考虑这两种偏好。第二种偏好是是否签名或加密一部分消息。判定表可以用于实现与是否签名或加密一部分消息有关的那种偏好协调。此外，可以使判定偏向于接受偏好，而不是签名，或接受接收器偏好，或正好相反。可以用于实现可能判定规则的一些判定表如下：

		发送器偏好		
		签名 必选	签名 可选	无签名
接收器偏好	签名 必选	签名	签名	错误
	签名 可选	签名	不签名	不签名
	无签名	错误	不签名	不签名

		发送器		
		加密 必选	加密 可选	无加密
接收器	加密 必选	加密	加密	错误
	加密 可选	加密	不加密	不加密
	无加密	错误	不加密	不加密

		发送器		
		签名 必选	签名 可选	无签名
接收器	签名 必选	签名	签名	签名
	签名 可选	签名	不签名	不签名
	无签名	不签名	不签名	不签名

		发送器		
		加密 必选	加密 可选	无加密
接收器	加密 必选	加密	加密	加密
	加密 可选	加密	不加密	不加密
	无加密	不加密	不加密	不加密

这些安全判定规则的格式同等效力地应用于其它偏好协商。在一些特殊情况下，譬如，转换，正如在引用的申请中所述的那样，可以应用信息损失或信息精度的度量。

5 也在源代码附录中的模式 TransformationContract.XSD 可以用于模型化包括本发明几个方面的互通性合同的文档转换部分。在 Spy 模式设计视窗中观察到，TransformationContract.XSD 部件指定一个或多个文档，以转换和可选地指定响应文档。DocumentToTransformType 包括源文档 ID 和部分名和接收器附件偏好标志。它可选地包括附件部分 ID 和描述如何实现转换的一个或多个转换图。在源代码中进一步描述了这种模式，特别是转换图。在引用的申请中更全面地讨论了文档转换。  
10

在源代码附录中，在 InteroperabilityContract.XML 中提供了计算互通性合同的部分例子。这个例子包括一般、路由和转换合同部分。有关安全合同部分的例子，请看上文。对于本领域的普通技术人员来说，例子大都显而易见，  
15 特别是可用的伴随模式。一些重点如下。一般合同部分将此标识成支配协同

交互的合同。对于不拒绝、错误管理和其它使用，将消息存档。在编译总（或，可配置、特定）商业情报信息过程中，允许实用程序对由这个合同支配的消息加以考虑。对于 buyParty ConsumerOrderManagement send-Order 活动，给出来源地址。历史 DDID 号或地址进一步标识发送服务。对于 sellParty providerOrderManagement 进程定购活动，给出接收地址。发送器利用到指定地址的 C1 SOAP 1.0 封装协议接受异步错误消息。发送器可以要求接收器可以异步生成的交付收据。必选的消息部分或文档是 Order 和 Image。可选地，可以包括一些 XMLPart。对于签名、加密、存档、消息封装、载货单类型和交付收据类型，列举发送和接收连接器能力。在源代码附近中，示范性一般合同部分是例子的组成部分。

除了一般合同部分之外，还有路由合同部分和转换合同部分。示范性路由合同部分如下：

```
<RoutingContract>
    <route:RouteNode preICDComputation="false" connector="x-
gtw:cup.commerceone.com:connector::default" isNative="true" connectorFunction="service-send">
        <route:EntryChannel envelopeProtocol="C1 SOAP 1.0"
transportSupportedMessageType="both"
transportPhysicalAddress="icdtest.commerceone.com::SOAP_buyspicenutmeg"
transportProtocol="SONIC" transportNative="true" transportReliable="true"/>
        <route:ExitChannel envelopeProtocol="C1 SOAP 1.0"
transportSupportedMessageType="both"
transportPhysicalAddress="icdtest.commerceone.com::SOAP_buyspicenutmeg"
transportProtocol="SONIC" transportNative="true" transportReliable="true"/>
    </route:RouteNode>
    <route:RouteNode preICDComputation="false" connector="x-
gtw:cup.commerceone.com:connector::default" isNative="true" connectorFunction="service-receive">
        <route:EntryChannel envelopeProtocol="C1 SOAP 1.0"
transportSupportedMessageType="both"
transportPhysicalAddress="icdtest.commerceone.com::SOAP_buyspicenutmeg"
transportProtocol="SONIC" transportNative="true" transportReliable="true"/>
        <route:ExitChannel envelopeProtocol="C1 SOAP 1.0"
transportSupportedMessageType="both"
transportPhysicalAddress="icdtest.commerceone.com::SOAP_buyspicenutmeg"
transportProtocol="SONIC" transportNative="true" transportReliable="true"/>
    </route:RouteNode>
</RoutingContract>
```

这个样本例示了如上所述的模式的应用。类似地，例示转换模式的应用的示范性转换合同如下：

```

<TransformationContract>
  <xform:DocumentToTransform>
    <xform:SourceDocID>publicid:com.commerceone.schemas:PurchaseOrder:3.5</xform:SourceDocID>
      <xform:PartName>PurchaseOrder</xform:PartName>
      <xform:Attachment>false</xform:Attachment>
      <xform:TransformationMap>
        <xform:Connector>x-gtw::lion-z-01.lion.commerceone.com::connector::buyspicenutmeg</xform:Connector>
        <xform:StartDoc>
          <xform:DocURI>publicid:com.commerceone.schemas:PurchaseOrder:3.5</xform:DocURI>
            <xform:DocName>PurchaseOrder</xform:DocName>
            <xform:Namespace>publicid:com.commerceone.schemas</xform:Namespace>
              <xform:Version>3.5</xform:Version>
            </xform:StartDoc>
            <xform:EndDoc>
          <xform:DocURI>publicid:com.commerceone.schemas:PurchaseOrder:4.0</xform:DocURI>
            <xform:DocName>PurchaseOrder</xform:DocName>
            <xform:Namespace>publicid:com.commerceone.schemas</xform:Namespace>
              <xform:Version>4.0</xform:Version>
            </xform:EndDoc>
            <xform:CommunityID>exostar</xform:CommunityID>
            <xform:TransformationMapURI>urn:x-commerceone:transformation:1</xform:TransformationMapURI>
          </xform:TransformationMap>
        </xform:DocumentToTransform>
      </TransformationContract>
    
```

本领域的普通技术人员可以从前面的描述中明显看出，根据本发明的这  
5 些方面和部件可以构造出各种各样的系统和方法。一个实施例是指定互通性  
数据的机器可读数据结构。这种机器可读数据结构可用的环境是消费服务和  
提供或生产服务之间的互通。这些服务通过网络可选地利用中间连接器交换  
文档。机器可读数据结构可以组合如下有效数据单元的两个或更多个：通过  
服务名指定的服务和中间连接器之间的路线；用于消息交换的安排版本；存  
10 档消息、保证消息的可靠交付和要求接收确认的策略；特定消息的一些部分

的签名要求和使用的至少一种签名算法的说明；特定消息的一些部分的加密要求和使用的至少一种签名算法的说明；使用的一个或多个验证过程的说明；应用于包括在特定消息中的文档的一个或多个转换逻辑单元的说明；和是否应该与文档的转换副本一起包括文档的非转换副本的说明。在从属权利要求中指定的组合并不意味着是唯一的。特此清楚地描述了两个或更多个上面有用数据单元的置换。

本发明的进一步实施例是指定通过进程准备的当前互通性数据的机器可读数据结构。这种机器可读数据结构可用的环境是消费服务和提供或生产服务之间的互通。这些服务通过网络交换文档。可选地，这些服务可以使用中间连接器。与静态互通合同，譬如，由双方签名的合同不同，这种机器可读数据结构由一个进程响应启用服务之间的交换消息的请求创建。该处理包括访问服务的互通性数据，使服务的互通性数据相交，和对于产生多于一个可相互接受选项的互通性数据的相交，应用判定规则来选择一个选项。这种机器可读数据结构可以包括在前面实施例中描述的有用数据单元的任何置换。  
使用的判定规则可以由交换消息的服务预约或通过服务的预约可以适合于贸易共同体。贯穿本申请描述的任何判定规则可以用作本实施例的进一步方面。

本发明的另一个实施例是指定一个或多个安全通道的机器可读数据结构。这种机器可读数据结构可用的环境是消费服务和提供或生产服务之间的互通。这些服务通过网络交换文档。可选地，这些服务可以使用中间连接器。  
安全通道应用于签名、加密或验证中的一个或多个。它们也可应用于授权或不拒绝，或这些安全相关任务的任何组合。安全通道本身包括发出安全相关请求的连接器和响应安全相关请求的连接器的说明和安全相关请求的说明。安全相关请求可以包括一个或多个上面列出的安全相关任务。这种包括安全通道的数据结构可以响应启动服务之间的消息交换的请求形成。

虽然通过参照上面详述的优选实施例和例子公开了本发明，但应该明白，这些例子的用途是说明性的，而不是限制性的。在所述的实施例中暗示了计算机辅助处理。于是，本发明可以用计算机辅助处理的方法、包括实现这些方法的逻辑单元的系统、利用逻辑单元实现这些方法的媒体、利用逻辑单元实现这些方法的数据流或计算机可访问处理服务具体化。可以设想，本领域的普通技术人员可以容易地作出修改和组合，这些修改和组合在本发明的范围和所附权利要求书的范围内。

计算机程序列表附录:

### InteroperabilityContract.xsd

```

<?xml version="1.0" encoding="UTF-8" ?>
- <!--
edited with XML Spy v4.3 U (http://www.xmlspy.com) by Rashmi Murthy (Commerce One)
-->
- <xs:schema
  targetNamespace="publicid:com.commerceone:schemas/soapextension/c
  ontract/v1_0/InteroperabilityContract.xsd"
  xmlns:security="publicid:com.commerceone:schemas/soapextension/cont
  ract/security/v1_0/SecurityContract.xsd"
  xmlns:xform="publicid:com.commerceone:schemas/soapextension/contr
  act/transformation/v1_0/TransformationContract.xsd"
  xmlns:route="publicid:com.commerceone:schemas/soapextension/contra
  ct/routing/v1_0/RoutingContract.xsd"
  xmlns:general="publicid:com.commerceone:schemas/soapextension/cont
  ract/general/v1_0/GeneralContract.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:icd="publicid:com.commerceone:schemas/soapextension/contract
  /v1_0/InteroperabilityContract.xsd" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import
    namespace="publicid:com.commerceone:schemas/soapextension/co
    ntract/general/v1_0/GeneralContract.xsd"
    schemaLocation="http://schemas.commerceone.com/schemas/soape
    xtension/contract/general/v1_0/GeneralContract.xsd" />
  <xs:import
    namespace="publicid:com.commerceone:schemas/soapextension/co
    ntract/routing/v1_0/RoutingContract.xsd"
    schemaLocation="http://schemas.commerceone.com/schemas/soape
    xtension/contract/routing/v1_0/RoutingContract.xsd" />
  <xs:import
    namespace="publicid:com.commerceone:schemas/soapextension/co
    ntract/transformation/v1_0/TransformationContract.xsd"
    schemaLocation="http://schemas.commerceone.com/schemas/soape
    xtension/contract/transformation/v1_0/TransformationContract.x
    sd" />
  <xs:import
    namespace="publicid:com.commerceone:schemas/soapextension/co
    ntract/security/v1_0/SecurityContract.xsd"
    schemaLocation="http://schemas.commerceone.com/schemas/soape
    xtension/contract/security/v1_0/SecurityContract.xsd" />
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-
    core-schema.xsd" />
- <xs:element name="InteroperabilityContract">
  - <xs:annotation>
    - <xs:documentation>Container for ICD blocks</xs:documentation>
  </xs:annotation>
  - <xs:complexType>
    - <xs:sequence>

```

```

- <xs:element name="GeneralContract"
  type="general:GeneralContractType">
- <xs:annotation>
  <xs:documentation>General contract sub-block of
    ICD. This contains all general contract
    information</xs:documentation>
</xs:annotation>
</xs:element>
- <xs:element name="RoutingContract"
  type="route:RouteType">
- <xs:annotation>
  <xs:documentation>Routing contract sub-block of
    ICD. Contains the end-to-end
    route</xs:documentation>
</xs:annotation>
</xs:element>
- <xs:element name="TransformationContract"
  type="xform:TransformationContractType"
  minOccurs="0">
- <xs:annotation>
  <xs:documentation>Transformation contract sub-
    block of ICD. Contains transformation
    information required for version
    interoperability</xs:documentation>
</xs:annotation>
</xs:element>
- <xs:element name="SecurityContract"
  type="security:SecurityContractType" minOccurs="0">
- <xs:annotation>
  <xs:documentation>Security contract sub-block.
    Contains security information needed to satisfy
    security constraints between the sending and
    receiving parties</xs:documentation>
</xs:annotation>
</xs:element>
- <xs:element name="ContractSignature"
  type="ds:SignatureType" minOccurs="0">
- <xs:annotation>
  <xs:documentation>Signature for this
    contract</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
<xs:anyAttribute namespace="#other" processContents="lax" />
</xs:complexType>
</xs:element>
</xs:schema>

```

## GeneralContract.XSD

```

<?xml version="1.0" encoding="UTF-8" ?>
- <!--
edited with XML Spy v4.3 U (http://www.xmlspy.com) by Rashmi Murthy (Commerce One)
-->
- <xs:schema
    targetNamespace="publicid:com.commerceone:schemas/soapextension/c
    ontract/general/v1_0/GeneralContract.xsd"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:gen="publicid:com.commerceone:schemas/soapextension/contrac
    t/general/v1_0/GeneralContract.xsd" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
- <xs:element name="GeneralContract" type="gen:GeneralContractType">
-   <xs:annotation>
      <xs:documentation>General information of the
          InteroperabilityContract</xs:documentation>
    </xs:annotation>
- </xs:element>
- <xs:complexType name="ServiceActivityType">
-   <xs:annotation>
      <xs:documentation>Description of service and
          activity</xs:documentation>
    </xs:annotation>
-   <xs:sequence>
-     <xs:element name="Service">
-       <xs:annotation>
          <xs:documentation>URI of the service
              definition</xs:documentation>
        </xs:annotation>
-       <xs:complexType>
-         <xs:simpleContent>
-           <xs:extension base="xs:anyURI">
-             <xs:attribute name="Version" type="xs:string"
-               use="optional" />
-             <xs:attribute name="EnvelopeProtocol"
-               type="xs:string" use="optional" />
-           </xs:extension>
-           <xs:simpleContent>
-             </xs:complexType>
-           </xs:element>
-         <xs:element name="Activity" type="xs:string">
-           <xs:annotation>
              <xs:documentation>Activity name</xs:documentation>
            </xs:annotation>
-           </xs:element>
-         </xs:sequence>
-         <xs:attribute name="SoapAction" type="xs:string" use="optional" />
-       </xs:complexType>
-     <xs:complexType name="GeneralContractType">
-       <xs:sequence>
-         <xs:element name="From">
-           <xs:annotation>

```

```

<xs:documentation>Sending
    party/service/activity</xs:documentation>
</xs:annotation>
<xs:complexType>
    <xs:sequence>
        <xs:element name="FromAddress"
            type="gen:FromAddressType" minOccurs="0" />
        <xs:element name="SenderDDID" type="xs:string"
            minOccurs="0">
            <xs:annotation>
                <xs:documentation>DDID of the sender.
                    This will not be present if the sender is
                    a virtual CP or if the mode is
                    client/server</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="To">
    <xs:annotation>
        <xs:documentation>Receiving
            party/service/activity</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="ToAddress"
                type="gen:ToAddressType" />
            <xs:element name="ReceiverDDID"
                type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>DDID of the receiver.
                        DDID of the sender. This will not be
                        present if the receiver is a virtual
                        CP</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ErrorHandling">
    <xs:annotation>
        <xs:documentation>This is a capability parameter in
            activity definition</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="SendAsyncErrorResponseTo"
                type="gen:ServiceActivityType" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>service/activity in
                        From party to which the async error
                        response should be
                        sent</xs:documentation>

```

```

</xs:annotation>
</xs:element>
</xs:sequence>


<xs:documentation>Indicates whether the sender accepts async error response. This only applies to one-way messages</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>


<xs:documentation>This is a capability parameter in activity definition</xs:documentation>
</xs:annotation>
</xs:complexType>



<xs:documentation>service/activity in From party to which the delivery receipt should be sent</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>


<xs:documentation>This applies to only one-way messages</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="IsSignatureRequiredBySender" type="xs:boolean" use="optional" />
<xs:attribute name="IsAsyncDeliveryReceiptAcceptedBySender" type="xs:boolean" use="optional" />


<xs:documentation>Indicates whether the receiver can generate a delivery receipt as required by the sender. If set to false, gateway will generate the delivery receipt on behalf of the receiving connector</xs:documentation>
</xs:annotation>
</xs:attribute>

```

```

        </xs:complexType>
    </xs:element>
    -<xs:element name="RequiredMessagePart"
        type="gen:MessagePartInfo" maxOccurs="unbounded">
        -<xs:annotation>
            <xs:documentation>Contains information collected
                from the registry for all the required message
                parts</xs:documentation>
        </xs:annotation>
    </xs:element>
    -<xs:element name="OptionalMessagePart"
        type="gen:MessagePartInfo" minOccurs="0"
        maxOccurs="unbounded">
        -<xs:annotation>
            <xs:documentation>Contains information collected
                from the registry for all the optional message
                parts</xs:documentation>
        </xs:annotation>
    </xs:element>
    -<xs:element name="SendingConnectorCapabilities"
        type="gen:ConnectorCapabilitiesType" minOccurs="0">
        -<xs:annotation>
            <xs:documentation>Describes the list of attributes and
                their associated values for the send side connector.
                This will not be present for non-collaborative
                request and oneway messages</xs:documentation>
        </xs:annotation>
    </xs:element>
    -<xs:element name="ReceivingConnectorCapabilities"
        type="gen:ConnectorCapabilitiesType" minOccurs="0">
        -<xs:annotation>
            <xs:documentation>Describes the list of attributes and
                their associated values for the receive side
                connector. This will not be present for non-
                collaborative response message</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
-<xs:attribute name="ChoreographyID" type="xs:anyURI"
    use="optional">
    -<xs:annotation>
        <xs:documentation>Choreography which the service is
            associated with. This only applies to Collaborative
            Interactions</xs:documentation>
    </xs:annotation>
</xs:attribute>
-<xs:attribute name="MessageType" use="required">
    -<xs:annotation>
        <xs:documentation>Indicates if the message is request,
            response or oneway</xs:documentation>
    </xs:annotation>
    -<xs:simpleType>
        -<xs:restriction base="xs:string">
            <xs:enumeration value="REQUEST" />
            <xs:enumeration value="RESPONSE" />

```

```

        <xs:enumeration value="ONEWAY" />
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
- <xs:attribute name="CollaborativeInteraction" type="xs:boolean"
    use="required">
    - <xs:annotation>
        <xs:documentation>Indicates whether it is a collaborative
            or non-collaborative messaging
            paradigm</xs:documentation>
    </xs:annotation>
</xs:attribute>
- <xs:attribute name="ICDTimeToLive" type="xs:long" use="required">
    - <xs:annotation>
        <xs:documentation>Time duration after which the cached
            version of this ICD expires. This value is set in the
            config file</xs:documentation>
    </xs:annotation>
</xs:attribute>
- <xs:attribute name="MessageTimeToLive" type="xs:long"
    use="required">
    - <xs:annotation>
        <xs:documentation>Time duration after which the message
            will be dropped. This value is set in the activity
            definition</xs:documentation>
    </xs:annotation>
</xs:attribute>
- <xs:attribute name="MessageArchived" type="xs:boolean"
    use="required">
    - <xs:annotation>
        <xs:documentation>Indicates whether the message should
            be archived. This is a capability parameter in activity
            definition</xs:documentation>
    </xs:annotation>
</xs:attribute>
- <xs:attribute name="BusinessIntelligence" type="xs:boolean"
    use="required">
    - <xs:annotation>
        <xs:documentation>Indicates whether the message is
            available for Business Intelligence purposes. This is a
            capability parameter in activity
            definition</xs:documentation>
    </xs:annotation>
</xs:attribute>
- <xs:attribute name="ContractID" type="xs:string" use="required">
    - <xs:annotation>
        <xs:documentation>This contract's ID</xs:documentation>
    </xs:annotation>
</xs:attribute>
- <xs:attribute name="QualityOfService" use="required">
    - <xs:simpleType>
        - <xs:restriction base="xs:string">
            <xs:enumeration value="EXACTLYONCE" />
            <xs:enumeration value="BESTEFFORT" />

```

```

        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>




<xs:documentation>URI of the collaboration party. This will not be present if the sender is an unregistered foreign party</xs:documentation>




<xs:documentation>Sending service and activity. This will not be present if the from party is not present. Also, it will not be present if the message is the request part of a request/response message in a non-collaborative messaging paradigm</xs:documentation>








<xs:documentation>URI of the collaboration party</xs:documentation>




<xs:documentation>Receiving service and activity. This will not be present if the message is the response part of a request/response in a non-collaborative messaging paradigm</xs:documentation>








<xs:documentation>List of attributes</xs:documentation>





```

```

    = <xs:element name="Value" type="xs:string"
      minOccurs="0" maxOccurs="unbounded">
      = <xs:annotation>
        <xs:documentation>Values for each
          attribute</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="Name" type="xs:string" />
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
= <xs:complexType name="MessagePartInfo">
  = <xs:attribute name="PartName" type="xs:string" use="required">
    = <xs:annotation>
      <xs:documentation>Name of the document part. An
        example would be PurchaseOrder</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  = <xs:attribute name="DocIDRequired" type="xs:boolean"
    use="required">
    = <xs:annotation>
      <xs:documentation>Document ID of the part. This
        information is present in the input ICD
        Request</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  = <xs:attribute name="Location" type="xs:string" use="required">
    = <xs:annotation>
      <xs:documentation>Location of the part in the message.
        Possible values are SOAP body, attachment and
        external</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  = <xs:attribute name="MimeType" type="xs:string" use="optional">
    = <xs:annotation>
      <xs:documentation>Specifies the MIME
        type</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  = <xs:attribute name="Root" type="xs:boolean" use="required">
    = <xs:annotation>
      <xs:documentation>Indicates if this is the root
        part</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  = <xs:attribute name="XML" type="xs:boolean" use="required">
    = <xs:annotation>
      <xs:documentation>Indicates if this part is an XML
        message</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>

```

</xs:schema>

## RoutingContract.XSD

```

<?xml version="1.0" encoding="UTF-8" ?>
- <!--
edited with XML Spy v4.4 U (http://www.xmlspy.com) by Todd Klaus (Commerce One)
-->
= <xsd:schema
  targetNamespace="publicid:com.commerceone:schemas/soapextension/c
  ontract/routing/v1_0/RoutingContract.xsd"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:route="publicid:com.commerceone:schemas/soapextension/contra
  ct/routing/v1_0/RoutingContract.xsd" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
- <!--
imports
-->
- <!--
elements and types
-->
= <xsd:element name="Route" type="route:RouteType">
  = <xsd:annotation>
    <xsd:documentation>Routing element in the
      ICD</xsd:documentation>
  </xsd:annotation>
</xsd:element>
= <xsd:complexType name="RouteType">
  = <xsd:annotation>
    <xsd:documentation>Defines the list of nodes to be traversed
      from sender to receiver</xsd:documentation>
  </xsd:annotation>
  = <xsd:sequence>
    = <xsd:element name="RouteNode" type="route:RouteNodeType"
      minOccurs="2" maxOccurs="unbounded">
      = <xsd:annotation>
        <xsd:documentation>Nodes in the route. There must be
          at least two nodes in the route (sender and
          receiver)</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
= <xsd:complexType name="RouteNodeType">
  = <xsd:annotation>
    <xsd:documentation>Defines a node in the
      route</xsd:documentation>
  </xsd:annotation>
  = <xsd:sequence>
    = <xsd:element name="EntryChannel" type="route:ChannelType">
      = <xsd:annotation>
        <xsd:documentation>Transport and envelope protocol
          used to reach this node. Becomes ExitChannel when
          route is reversed.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>

```

```

- <xsd:element name="ExitChannel" type="route:ChannelType">
  - <xsd:annotation>
    <xsd:documentation>Transport and envelope protocol  

      used to exit this node. Becomes EntryChannel when  

      route is reversed.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
- <xsd:attribute name="connector" type="xsd:string" use="required">
  - <xsd:annotation>
    <xsd:documentation>GTW unique name consisting of  

      issuing authority prefix, type (always connector here),  

      community name, and local name</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="isNative" type="xsd:boolean" use="required">
  - <xsd:annotation>
    <xsd:documentation>Indicates whether this connector is  

      running C1 software (CWSP 6.0+)</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="connectorFunction" use="required">
  - <xsd:annotation>
    <xsd:documentation>Specifies the role this connector plays  

      in the route at the specified node.</xsd:documentation>
  </xsd:annotation>
  - <xsd:simpleType>
    - <xsd:restriction base="xsd:string">
      <xsd:enumeration value="service-send" />
      <xsd:enumeration value="service-receive" />
      <xsd:enumeration value="hub" />
      <xsd:enumeration value="envelope-gateway" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
- <xsd:attribute name="preICDComputation" type="xsd:boolean"  

  use="required">
  - <xsd:annotation>
    <xsd:documentation>Indicates whether this node should  

      have already been traversed by the time the ICD  

      request was made (i.e., it is prior to the current  

      connector/envelope protocol)</xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>
- <xsd:complexType name="ChannelType">
  - <xsd:annotation>
    <xsd:documentation>Defines the transport information needed  

      to reach the associated node</xsd:documentation>
  </xsd:annotation>
  - <xsd:attribute name="envelopeProtocol" type="xsd:string"  

    use="required">
    - <xsd:annotation>

```

---

```

<xsd:documentation>Envelope protocol and version
    associated with this channel</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="transportSupportedMessageType"
    use="required">
- <xsd:annotation>
    <xsd:documentation>Message type supported by this
        channel. One of oneway, request-response, or
        both</xsd:documentation>
</xsd:annotation>
- <xsd:simpleType>
- <xsd:restriction base="xsd:string">
    <xsd:enumeration value="oneway" />
    <xsd:enumeration value="request-response" />
    <xsd:enumeration value="both" />
</xsd:restriction>
</xsd:simpleType>
</xsd:attribute>
- <xsd:attribute name="transportPhysicalAddress" type="xsd:string"
    use="required">
- <xsd:annotation>
    <xsd:documentation>transport-specific address (URL,
        node:queue name, etc)</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="transportProtocol" type="xsd:string"
    use="required">
- <xsd:annotation>
    <xsd:documentation>Transport type (HTTPS, Sonic,
        etc.)</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="transportReliable" type="xsd:boolean"
    use="required">
- <xsd:annotation>
    <xsd:documentation>Indicates whether this transport is
        reliable.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="transportNative" type="xsd:boolean"
    use="required">
- <xsd:annotation>
    <xsd:documentation>Indicates whether this is a natively
        supported transport. If false, it is handled by a
        transport gateway.</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
</xsd:schema>

```

## TransformationContract.XSD

```

<?xml version="1.0" encoding="UTF-8"?>
- <!--
edited with XML Spy v4.4 U (http://www.xmlspy.com) by Helen Yuen (Commerce One)
-->
- <!--
Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema
-->
- <schema
  targetNamespace="publicId:com.commerceone:schemas/soapextension/c
  ontract/transformation/v1_0/TransformationContract.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tpc="publicId:com.commerceone:schemas/soapextension/contract
  /transformation/v1_0/TransformationContract.xsd"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  version="1.0">
- <!--
import namespaces
-->
- <!--
global elements
-->
- <element name="TransformationContract"
  type="tpc:TransformationContractType">
  - <annotation>
    <documentation>Transformation Contract Block of the
      ICD</documentation>
  </annotation>
  </element>
- <complexType name="DocInfoType">
  - <sequence>
    <element name="DocURI" type="xs:anyURI" />
    <element name="DocName" type="xs:string" />
    <element name="Namespace" type="xs:anyURI" />
    <element name="Version" type="xs:string" />
  </sequence>
  </complexType>
- <complexType name="TransformationContractType">
  - <sequence>
    - <element name="DocumentToTransform"
      type="tpc:DocumentToTransformType"
      maxOccurs="unbounded">
      - <annotation>
        <documentation>Source Document transformation
          information</documentation>
      </annotation>
    </element>
    <element name="ResponseDoc" type="tpc:ResponseDocType"
      minOccurs="0" />
  </sequence>
  </complexType>
- <complexType name="TransformationMapType">

```

```

    -<sequence>
        -<element name="Connector" type="xs:anyURI">
            -<annotation>
                <documentation>Connector GTW name. Specify the
                    location where the transformation will
                    occur.</documentation>
            </annotation>
        </element>
        <element name="StartDoc" type="tpc:DocInfoType" />
        <element name="EndDoc" type="tpc:DocInfoType" />
    -<element name="CommunityID" type="xs:string">
        -<annotation>
            <documentation>Community ID of where the
                transformation maps are located.</documentation>
        </annotation>
    </element>
    <element name="TransformationMapURI" type="xs:anyURI" />
</sequence>
</complexType>
-<complexType name="ResponseDocType">
    -<sequence>
        <element name="DocIdURI" type="xs:anyURI" />
        <element name="ColumnNum" type="xs:int" />
    </sequence>
</complexType>
-<complexType name="DocumentToTransformType">
    -<sequence>
        -<element name="SourceDocID" type="xs:anyURI">
            -<annotation>
                <documentation>Source Document ID</documentation>
            </annotation>
        </element>
        -<element name="PartName" type="xs:string">
            -<annotation>
                <documentation>Source Document
                    PartID</documentation>
            </annotation>
        </element>
        -<element name="Attachment" type="xs:boolean">
            -<annotation>
                <documentation>Receiver attachment preference
                    flag</documentation>
            </annotation>
        </element>
        -<element name="AttachmentPartID" type="xs:string"
            minOccurs="0">
            -<annotation>
                <documentation>Attachment Part ID</documentation>
            </annotation>
        </element>
        -<element name="TransformationMap"
            type="tpc:TransformationMapType" minOccurs="0"
            maxOccurs="unbounded">
            -<annotation>

```

```
<documentation>Transformation
    instructions</documentation>
</annotation>
</element>
</sequence>
</complexType>
</schema>
```

## SecurityContractKeyInfo.XSD

```

<?xml version="1.0" encoding="UTF-8" ?>
- <!--
edited with XML Spy v4.4 U (http://www.xmlspy.com) by Symon Chang (Commerce One)
-->
- <xsschema
  targetNamespace="publicid:com.commerceone:schemas/soapextension/c
  ontract/security/v1_0/SecurityContract.xsd"
  xmlns:sicd="publicid:com.commerceone:schemas/soapextension/contrac
  t/security/v1_0/SecurityContract.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  version="1.0">
- <xssimpleType name="CollaborationPartyID">
- <xssannotation>
  <xsddocumentation>This is the Collaboration Partner's
  ID</xsddocumentation>
</xssannotation>
<xssrestriction base="xs:string" />
</xssimpleType>
- <xssimpleType name="KeyUsageTypes">
- <xssannotation>
  <xsddocumentation>Key is used for signature, encryption,
  and/or authentication.</xsddocumentation>
</xssannotation>
- <xssrestriction base="xs:NMTOKENS">
  <xsesenumeration value="AUTHENTICATION" />
  <xsesenumeration value="ENCRYPTION" />
  <xsesenumeration value="SIGNATURE" />
  <xsesenumeration value="SSL" />
</xssrestriction>
</xssimpleType>
- <xssimpleType name="KeyAlgorithmTypes">
- <xssannotation>
  <xsddocumentation>Key is RSA or DSA type of
  key.</xsddocumentation>
</xssannotation>
- <xssrestriction base="xs:NMTOKENS">
  <xsesenumeration value="RSA" />
  <xsesenumeration value="DSA" />
</xssrestriction>
</xssimpleType>
- <xssimpleType name="AuthenticateModeTypes">
- <xssannotation>
  <xsddocumentation>The location of where the authentication
  takes place. NONE means neither source nor target
  connector will perform the authentication. This may be the
  case of letting foreign connector to perform the
  authentication.</xsddocumentation>
</xssannotation>
- <xssrestriction base="xs:NMTOKEN">
  <xsesenumeration value="SOURCE" />
  <xsesenumeration value="TARGET" />

```

```

<xs:enumeration value="NONE" />
</xs:restriction>
</xs:simpleType>
= <xs:element name="PublicKey" type="sicd:PublicKeyType">
  = <xs:annotation>
    <xs:documentation>The Public Key record. Each public key will have partyID, KeyInfo, description and usages.</xs:documentation>
  </xs:annotation>
</xs:element>
= <xs:element name="EncryptionKeyInfo">
  = <xs:annotation>
    <xs:documentation>The KeyInfo that has both PublicKeyID and X509Data for encryption.</xs:documentation>
  </xs:annotation>
</xs:element>
= <xs:complexType>
  = <xs:complexContent>
    = <xs:extension base="sicd:KeyInfoType">
      <xs:attribute name="KeyOwner" type="sicd:CollaberationPartyID" use="optional" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:element>
= <xs:complexType name="PublicKeyType">
  = <xs:annotation>
    <xs:documentation>The Public Key record, including PartyID, KeyInfo, Usages and Description.</xs:documentation>
  </xs:annotation>
  = <xs:sequence>
    <xs:element ref="sicd:PartyID" />
    = <xs:element ref="sicd:EncryptionKeyInfo">
      = <xs:annotation>
        <xs:documentation>The KeyInfo block that has KeyID and X509 Data.</xs:documentation>
      </xs:annotation>
    </xs:element>
    = <xs:element ref="sicd:KeyTypeUsage" maxOccurs="4">
      = <xs:annotation>
        <xs:documentation>Key is used for signature, encryption, and/or authentication.</xs:documentation>
      </xs:annotation>
    </xs:element>
  = <xs:element name="KeyAlgorithm" type="sicd:KeyAlgorithmTypes" minOccurs="0">
    = <xs:annotation>
      <xs:documentation>The Key Is RSA or DSA key</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element ref="sicd:Description" minOccurs="0" />
= <xs:element name="Location" type="xs:string" minOccurs="0">
  = <xs:annotation>

```

```

<xss:documentation>The connector ID that key the
    Private Key.</xss:documentation>
</xss:annotation>
</xss:element>
</xss:sequence>
</xss:complexType>
= <xss:element name="PartyID" type="sicd:CollaborationPartyID">
    = <xss:annotation>
        <xss:documentation>Trading partner ID or Collaboration Partner
            ID in UUID format.</xss:documentation>
        </xss:annotation>
    </xss:element>
= <xss:element name="Description" type="xs:string">
    = <xss:annotation>
        <xss:documentation>The description of the
            key</xss:documentation>
        </xss:annotation>
    </xss:element>
= <xss:element name="KeyTypeUsage" type="sicd:KeyUsageTypes">
    = <xss:annotation>
        <xss:documentation>Key is used for signature, encryption,
            and/or authentication.</xss:documentation>
        </xss:annotation>
    </xss:element>
= <xss:element name="KeyInfo">
    = <xss:annotation>
        <xss:documentation>The KeyInfo object is from the XMLDSig
            ds:KeyInfo object. However, within SICD we only use Public
            Key ID field.</xss:documentation>
        </xss:annotation>
    = <xss:complexType>
        = <xss:sequence>
            = <xss:element ref="sicd:PublicKeyID" />
        </xss:sequence>
    </xss:complexType>
</xss:element>
= <xss:element name="PublicKeyID" type="xs:string">
    = <xss:annotation>
        <xss:documentation>The Public Key ID is a unique key ID (UUID
            or from XMKS server).</xss:documentation>
        </xss:annotation>
    </xss:element>
= <xss:element name="PublicKeyName" type="sicd:PublicKeyNameType">
    = <xss:annotation>
        <xss:documentation>The Name of the Public Key. It is same as
            the PublicKeyID but has owner name as the optional
            attribute.</xss:documentation>
        </xss:annotation>
    </xss:element>
= <xss:complexType name="PublicKeyNameType">
    = <xss:simpleContent>
        = <xss:extension base="xs:string">
            <xss:attribute name="KeyOwner"
                type="sicd:CollaborationPartyID" use="optional" />

```

```

        </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    -<xs:complexType name="KeyInfoType">
        -<xs:annotation>
            <xs:documentation>This is for Encryption. The KeyInfo object is
            from the XMLDSig ds:KeyInfo object. However, within SICD
            we only use Public Key ID and X509 Certificate two
            fields.</xs:documentation>
        </xs:annotation>
        -<xs:sequence>
            <xs:element ref="sicd:PublicKeyID" />
            -<xs:element name="X509Data" minOccurs="0">
                -<xs:complexType>
                    -<xs:sequence>
                        <xs:element name="X509Certificate"
                            type="xs:base64Binary" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    -<!--
    Policy Types
    -->
    -<xs:complexType name="Abstract_PolicyType" abstract="true">
        -<xs:annotation>
            <xs:documentation>This is the abstract policy for all security
            policy related algorithm. The ID is the Template Name for
            the Algorithm.</xs:documentation>
        </xs:annotation>
        <xs:attribute name="PolicyId" type="xs:string" use="optional" />
    </xs:complexType>
    -<xs:complexType name="Abstract_CredentialPolicyType"
        abstract="true">
        -<xs:annotation>
            <xs:documentation>This is the abstract policy for authentication
            credential policy algorithm.</xs:documentation>
        </xs:annotation>
    </xs:complexType>
    -<xs:complexContent>
        -<xs:extension base="sicd:Abstract_PolicyType">
            -<xs:sequence>
                <xs:element name="CredentialPolicyAlgorithm"
                    type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
-<xs:element name="AuthenticateImplementation" type="xs:string">
    -<xs:annotation>
        <xs:documentation>Optional for different implementation, such
        as SAML, SecureID, or Kerberos.</xs:documentation>
    </xs:annotation>
</xs:element>

```

```

    -<xs:element name="AuthenticateMode"
      type="sicd:AuthenticateModeTypes">
    -<xs:annotation>
      <xs:documentation>The location of where the authentication
      takes place. It can be either SOURCE connector or TARGET
      connector. SOURCE means the sender's local connectors will
      perform SAML Single Sign-On type of authentication.
      TARGET means the connector on the receiving end will
      perform the authentication. NONE means neither source nor
      target connector will perform the authentication. This may
      be the case of letting foreign connector to perform the
      authentication.</xs:documentation>
    </xs:annotation>
  </xs:element>
-<xs:complexType name="AuthenticationCredentialPolicyType">
  -<xs:annotation>
    <xs:documentation>This authentication and credential policy
      will work for Basic and X509.</xs:documentation>
  </xs:annotation>
  -<xs:complexContent>
    -<xs:extension base="sicd:Abstract_CredentialPolicyType">
      -<xs:sequence minOccurs="0">
        <xs:element ref="sicd:AuthenticateMode" />
        <xs:element ref="sicd:AuthenticateImplementation"
          minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
-<xs:complexType name="AnonymousCredentialPolicyType">
  -<xs:annotation>
    <xs:documentation>This is an anonymous credential policy type
      that has no credential.</xs:documentation>
  </xs:annotation>
  -<xs:complexContent>
    -<xs:restriction base="sicd:Abstract_CredentialPolicyType">
      -<xs:sequence>
        <xs:element name="CredentialPolicyAlgorithm"
          type="xs:string" fixed="Anonymous" />
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
-<xs:complexType name="BasicCredentialPolicyType">
  -<xs:annotation>
    <xs:documentation>This is a basic credential policy type that
      uses ID and password as credential.</xs:documentation>
  </xs:annotation>
  -<xs:complexContent>
    -<xs:extension base="sicd:AuthenticationCredentialPolicyType"
      />
  </xs:complexContent>
</xs:complexType>
-<xs:complexType name="X509CredentialPolicyType">

```

---

```

    -<xs:annotation>
        <xs:documentation>This is a X509 credential policy type.</xs:documentation>
    </xs:annotation>
    -<xs:complexContent>
        <xs:extension base="sicd:AuthenticationCredentialPolicyType" />
    </xs:complexContent>
</xs:complexType>
-<xs:complexType name="BASE64_BINARYCredentialPolicyType">
    -<xs:annotation>
        <xs:documentation>This is a BASE64_BINARY_CREDENTIAL policy type.</xs:documentation>
    </xs:annotation>
    -<xs:complexContent>
        -<xs:extension base="sicd:AuthenticationCredentialPolicyType">
            -<xs:sequence>
                <xs:element name="valueType" type="xs:QName" />
                <xs:element name="encodingType" type="xs:QName" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
-<xs:complexType name="Abstract_EncryptionPolicyType"
    abstract="true"
    -<xs:annotation>
        <xs:documentation>This is the abstract policy for Encryption policy algorithm.</xs:documentation>
    </xs:annotation>
    -<xs:complexContent>
        -<xs:extension base="sicd:Abstract_PolicyType">
            -<xs:sequence>
                <xs:element name="EncryptionPolicyAlgorithm" type="xs:string" />
                <xs:element name="EncryptionMethod" type="xs:string" />
                <xs:element ref="sicd:KeySize" />
                <xs:element ref="sicd:SymmetryKeySize" minOccurs="0" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
-<xs:complexType name="EncryptionPolicyType">
    -<xs:annotation>
        <xs:documentation>This encryption policy will work for both XMLEnc and PKCS#7.</xs:documentation>
    </xs:annotation>
    -<xs:complexContent>
        -<xs:extension base="sicd:Abstract_EncryptionPolicyType">
            -<xs:sequence>
                <xs:element name="KeyEncryptionMethod" type="xs:string" minOccurs="0" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

---

```

        </xs:extension>
    </xs:complexContent>
</xs:complexType>
- <xs:element name="KeySize">
- <xs:annotation>
    <xs:documentation>This is the asymmetry encryption or
        symmetry key size, depends which algorithm is used. For an
        asymmetry case, this will be the asymmetry key size, and
        the symmetry key size is defined on the SymmetryKeySize
        field.</xs:documentation>
</xs:annotation>
- <xs:simpleType>
- <xs:restriction base="xs:short">
    <xs:minInclusive value="56" />
    <xs:maxExclusive value="4096" />
</xs:restriction>
</xs:simpleType>
</xs:element>
- <xs:element name="SymmetryKeySize">
- <xs:annotation>
    <xs:documentation>This is the symmetry encryption key size, if
        the asymmetry algorithm is used.</xs:documentation>
</xs:annotation>
- <xs:simpleType>
- <xs:restriction base="xs:short">
    <xs:minInclusive value="56" />
    <xs:maxExclusive value="4096" />
</xs:restriction>
</xs:simpleType>
</xs:element>
- <xs:complexType name="XMLEncryptionPolicyType">
- <xs:annotation>
    <xs:documentation>This will work for any encryption policy
        type.</xs:documentation>
</xs:annotation>
- <xs:complexContent>
- <xs:extension base="sicd:Abstract_EncryptionPolicyType">
- <xs:sequence>
    <xs:element name="KeyEncryptionMethod"
        type="xs:string"
        default="http://www.w3.org/2001/04/xmlenc#rsa
        -1_5" />
    <xs:element name="DecryptionTransform"
        type="xs:string" minOccurs="0" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
- <xs:complexType name="Abstract_SignaturePolicyType" abstract="true">
- <xs:annotation>
    <xs:documentation>This is the abstract policy for Digital
        Signature policy algorithm.</xs:documentation>
</xs:annotation>
- <xs:complexContent>

```

```

- <xs:extension base="sicd:Abstract_PolicyType">
  - <xs:sequence>
    <xs:element name="SignaturePolicyAlgorithm"
      type="xs:string" />
    <xs:element name="SignatureAlgorithm"
      type="xs:string" />
    <xs:element name="HashFunction" type="xs:string" />
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
- <xs:complexType name="SignaturePolicyType">
  - <xs:annotation>
    <xs:documentation>This will work for any digital signature policy type.</xs:documentation>
  </xs:annotation>
  - <xs:complexContent>
    <xs:extension base="sicd:Abstract_SignaturePolicyType" />
  </xs:complexContent>
</xs:complexType>
- <xs:complexType name="XMLEDsigPolicyType">
  - <xs:annotation>
    <xs:documentation>This is for XMLDSIG policy.</xs:documentation>
  </xs:annotation>
  - <xs:complexContent>
    - <xs:extension base="sicd:SignaturePolicyType">
      - <xs:sequence>
        <xs:element name="CanonicalizationMethod"
          type="xs:string" minOccurs="0" />
        <xs:element name="Transform" type="xs:string"
          minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
- <!--
Message Part
-->
- <xs:complexType name="PartElementType">
  - <xs:annotation>
    <xs:documentation>Xpath is used to define the element within the part of the message.</xs:documentation>
  </xs:annotation>
  - <xs:simpleContent>
    - <xs:extension base="xs:string">
      <xs:attribute name="Type" type="xs:anyURI" use="optional"
        />
      <xs:attribute name="BlockId" type="xs:short"
        use="optional" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
- <xs:complexType name="MessagePartsType">

```

```
    -<xs:annotation>
        <xs:documentation>The part within a message. URI is used to
            define the part.</xs:documentation>
    </xs:annotation>
    -<xs:sequence>
        -<xs:element name="PartElement" type="sicd:PartElementType">
            minOccurs="0" maxOccurs="unbounded">
            -<xs:annotation>
                <xs:documentation>The element within the part. It is
                    only apply to XML type of message
                    part.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="PartName" type="xs:string" use="required" />
    <xs:attribute name="Type" type="xs:anyURI" use="optional" />
    <xs:attribute name="AlgorithmId" type="xs:string" use="optional" />
    <xs:attribute name="BlockId" type="xs:short" use="optional" />
</xs:complexType>
-<xs:element name="MessagePart" type="sicd:MessagePartsType">
    -<xs:annotation>
        <xs:documentation>The part within the message. The
            AlgorithmId is for this part. If the AlgorithmId is not
            defined, then parent's AlgorithmId will be
            used.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:schema>
```

## SecurityContract.XSD

```

<?xml version="1.0" encoding="UTF-8" ?>
- <!--
edited with XML Spy v4.4 U (http://www.xmlspy.com) by Symon Chang (Commerce One)
-->
= <!--
  Security Interop Contract Document

  Created by: Symon Chang
  Copyright 2002 Commerce One, Inc.

-->
= <xs:schema
  targetNamespace="publicid:com.commerceone:schemas/soapextension/c
  ontract/security/v1_0/SecurityContract.xsd"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sicd="publicid:com.commerceone:schemas/soapextension/contrac
  t/security/v1_0/SecurityContract.xsd"
  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  version="1.0">
- <!--
imports
-->
= <!--

  <xs:import
    namespace="publicid:com.commerceone:schemas/soapextension/contract/v1_0/Inter
    operabilityContract.xsd"
    schemaLocation="http://schemas.commerceone.com/schemas/soapextension/contract/
    v1_0/InteroperabilityContract.xsd"/>

-->
<xs:import namespace="urn:oasis:names:tc:SAML:1.0:assertion"
  schemaLocation="http://www.oasis-
  open.org/committees/security/docs/cs-sstc-schema-assertion-
  01.xsd" />
- <!--
includes
-->
<xs:include schemaLocation="SecurityContractKeyInfo.xsd" />
- <!--
Schema for Security Policies
-->
- <!--
top element
-->
= <xs:element name="SecurityContractICD"
  type="sicd:SecurityContractType">
= <xs:annotation>
  <xs:documentation>The Security Interop Contract agreement. It
defines Policies and channels for security
policies.</xs:documentation>
</xs:annotation>

```

```

</xs:element>
- <!--
Schema for Security Policies
-->
- <!--
Define Credential Policies
-->
- <xs:element name="BasicCredentialPolicy"
  type="sicd:BasicCredentialPolicyType">
  <xs:annotation>
    <xs:documentation>The credential and authentication algorithm
policy for ID and Password.</xs:documentation>
  </xs:annotation>
</xs:element>
- <xs:element name="X509CredentialPolicy"
  type="sicd:X509CredentialPolicyType">
  <xs:annotation>
    <xs:documentation>The credential and authentication algorithm
policy for X.509 Certificate.</xs:documentation>
  </xs:annotation>
</xs:element>
- <xs:element name="AnonymousCredentialPolicy"
  type="sicd:AnonymousCredentialPolicyType">
  <xs:annotation>
    <xs:documentation>The credential and authentication algorithm
policy for no credential.</xs:documentation>
  </xs:annotation>
</xs:element>
- <xs:element name="BASE64_BINARYCredentialPolicy"
  type="sicd:BASE64_BINARYCredentialPolicyType">
  <xs:annotation>
    <xs:documentation>The credential and authentication algorithm
policy for BASE64_BINARY_CREDENTIAL</xs:documentation>
  </xs:annotation>
</xs:element>
- <xs:element name="AuthenticationPolicies">
  <xs:annotation>
    <xs:documentation>The abstraction for credential and
authentication algorithm policy.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="sicd:BasicCredentialPolicy" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="sicd:X509CredentialPolicy" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="sicd:BASE64_BINARYCredentialPolicy"
        minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="sicd:AnonymousCredentialPolicy"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
- <!--
Define Encryption Policies
-->

```

```

-->
- <xs:element name="EncryptionPolicy"
  type="sicd:EncryptionPolicyType">
  - <xs:annotation>
    <xs:documentation>The encryption algorithm and policy, such
      as PCSK#7, or S/MIME.</xs:documentation>
    </xs:annotation>
  </xs:element>
- <xs:element name="XMLEncryptionPolicy"
  type="sicd:XMLEncryptionPolicyType">
  - <xs:annotation>
    <xs:documentation>The encryption algorithm and policy for
      XMLEnc.</xs:documentation>
    </xs:annotation>
  </xs:element>
- <xs:element name="EncryptionPolicies">
  - <xs:annotation>
    <xs:documentation>The group of encryption algorithms and
      policies for XMLEnc, PCSK#7, or S/MIME. The PolicyID will
      be the TemplateID in the Registry. This ID will be used in
      the Channel Section as AlgorithmID to identify which
      encryption policy algorithm will be used.</xs:documentation>
    </xs:annotation>
  - <xs:complexType>
    - <xs:sequence>
      <xs:element ref="sicd:XMLEncryptionPolicy" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="sicd:EncryptionPolicy" minOccurs="0"
        maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
- <!--
Digital Signature Policy
-->
- <xs:element name="XMLDSigPolicy" type="sicd:XMLDSigPolicyType">
  - <xs:annotation>
    <xs:documentation>The signature algorithm and policy for
      XMLDSig.</xs:documentation>
  </xs:annotation>
</xs:element>
- <xs:element name="SignaturePolicy" type="sicd:SignaturePolicyType">
  - <xs:annotation>
    <xs:documentation>The signature algorithm and policy for
      XMLDSig, PCSK#7 or S/MIME.</xs:documentation>
  </xs:annotation>
</xs:element>
- <xs:element name="SignaturePolicies">
  - <xs:annotation>
    <xs:documentation>The group of digital signature algorithms
      and policies for XMLDSig, PCSK#7, or S/MIME. The Policy ID
      will be the TemplateID In the Registry. This Policy ID will be
      used in the Channel Section as AlgorithmID to identify
      which sinature policy algorithm will be
      used.</xs:documentation>
  </xs:annotation>
</xs:element>

```

```

</xs:annotation>


<xs:element ref="sicd:XMLDSigPolicy" minOccurs="0"
    maxOccurs="unbounded" />
<xs:element ref="sicd:SignaturePolicy" minOccurs="0"
    maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
- <!--
Non-repudiation
-->

    <xs:documentation>The non-repudiationl algorithm and policy
        that use daigital signature.</xs:documentation>

    <xs:documentation>The policy and algorithm for non-
        repudiation of origin.</xs:documentation>

    <xs:documentation>The non-repudiationl algorithm and policy
        that use daigital signature.</xs:documentation>

    <xs:documentation>The policy and algorithm for non-
        repudiation of receipt.</xs:documentation>

    <xs:documentation>The security Policies section. It defines all
        policy related security policies.</xs:documentation>


```

```

<xs:element ref="sicd:NonRepudiationPolicies"
    minOccurs="0" maxOccurs="unbounded" />
<xs:element ref="sicd:NonRepudiationReceiptPolicies"
    minOccurs="0" maxOccurs="unbounded" />
<xs:element ref="sicd:EncryptionKeyInfo" minOccurs="0"
    maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
- <!--
Schema for Channel
-->
- <xs:complexType name="KeyAlgorithmType">
  - <xs:annotation>
    <xs:documentation>The root for Integraty and Confidential
    blocks. All these two types of block within the Security
    channel have to have PublicKeyID and AlgorithmId, so does
    the signing and encryption policy within the Credential
    block.</xs:documentation>
  </xs:annotation>
  - <xs:sequence>
    <xs:element ref="sicd:PublicKeyName" />
  </xs:sequence>
  <xs:attribute name="AlgorithmId" type="xs:string" use="optional" />
</xs:complexType>
- <xs:complexType name="KeyMessagePartsType">
  - <xs:annotation>
    <xs:documentation>The root for parts in a message. It also
    define the KeyInfo and the algorithm policy for all
    parts.</xs:documentation>
  </xs:annotation>
  - <xs:complexContent>
    - <xs:extension base="sicd:KeyAlgorithmType">
      - <xs:sequence minOccurs="0">
        <xs:element ref="sicd:MessagePart" minOccurs="0"
            maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="SequenceID" type="xs:short"
          use="optional" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
- <xs:element name="Credential">
  - <xs:annotation>
    <xs:documentation>The credentail and authentication polocy.
    Note that the CredentialEncryptionAlgorithm is here. This is
    due to authentication will be preformed before the
    decryption at inbound.</xs:documentation>
  </xs:annotation>
  - <xs:complexType>
    - <xs:sequence minOccurs="0">
      - <xs:choice minOccurs="0">
        - <xs:element name="PartyID"
            type="sicd:CollaberationPartyID" minOccurs="0">

```

```

    = <xs:annotation>
        <xs:documentation>The party ID that is used
            for Basic credential.</xs:documentation>
        </xs:annotation>
    </xs:element>
    = <xs:element ref="sicd:PublicKeyName" minOccurs="0">
        = <xs:annotation>
            <xs:documentation>The key that is used for
                X.509 credential.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:choice>
    = <xs:element name="CredentialEncryptionAlgorithm"
        type="sicd:KeyAlgorithmType" minOccurs="0">
        = <xs:annotation>
            <xs:documentation>The Encryption Algorithm that
                is used to encrypt the credential. This will only
                be used when the Authentication mode is
                TARGET.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="AlgorithmId" type="xs:string"
        use="required" />
    <xs:attribute name="SequenceID" type="xs:short"
        use="optional" />
    <xs:attribute name="DelegationFlag" type="xs:boolean"
        use="optional" default="false" />
</xs:complexType>
</xs:element>
= <xs:element name="Confidential">
    = <xs:annotation>
        <xs:documentation>The encryption security policy. The
            AlgorithmId will be the templateID from the Registry. If the
            AlgorithmId is defined and no message parts, then the
            whole message will be encrypted. In this case, if there are
            Non-XML parts, then the NonXMLAlgorithmID will be
            defined, too.</xs:documentation>
    </xs:annotation>
    = <xs:complexType>
        = <xs:complexContent>
            = <xs:extension base="sicd:KeyMessagePartsType">
                <xs:attribute name="NonXMLAlgorithmId"
                    type="xs:string" use="optional" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:element>
= <xs:element name="Integrity">
    = <xs:annotation>
        <xs:documentation>The digital signature security policy. The
            AlgorithmId will be the templateID from the Registry. If the
            AlgorithmID is defined, and no message parts then the
            whole message will be signed.</xs:documentation>
    </xs:annotation>

```

```

    = <xs:complexType>
    = <xs:complexContent>
        = <xs:extension base="sicd:KeyMessagePartsType">
            = <xs:sequence minOccurs="0">
                = <xs:element name="HeaderSignatureAlgorithm"
                    type="sicd:KeyAlgorithmType" minOccurs="0">
                    = <xs:annotation>
                        <xs:documentation>The Signature Algorithm that is used to sign the header credential.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="NonXMLAlgorithmId"
                type="xs:string" use="optional" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
</xs:element>
= <xs:element name="NonRepudiation">
    = <xs:annotation>
        <xs:documentation>The non-repudiation of origin policy.</xs:documentation>
    </xs:annotation>
    = <xs:complexType>
        = <xs:sequence>
            = <xs:element name="NROSignPart"
                type="sicd:KeyMessagePartsType" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
= <xs:element name="NonRepudiationReceipt">
    = <xs:annotation>
        <xs:documentation>The non-repudiation of receipt policy.</xs:documentation>
    </xs:annotation>
    = <xs:complexType>
        = <xs:sequence>
            = <xs:element name="NRRSignPart"
                type="sicd:KeyMessagePartsType" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
= <xs:element name="Authorization">
    = <xs:annotation>
        <xs:documentation>The SAML attribute assertion for the sending CP that will be pass to the receiving service. This will be shown in the end-to-end security channel.</xs:documentation>
    </xs:annotation>
    = <xs:complexType>
        = <xs:simpleContent>
            = <xs:extension base="xs:string">

```

```

<xs:attribute name="RequireSubscription"
    type="xs:boolean" use="optional" default="false" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
- <!--
saml:AttributeStatementType">
-->
</xs:element>


<xs:documentation>This will be the container for those piggy
back security related objects.</xs:documentation>



<xs:element name="PiggybackObject" type="xs:anyType"
    minOccurs="0" maxOccurs="unbounded" />

</xs:complexType>
</xs:element>


<xs:documentation>The Security Channel defines the from
connector and to connector, and what to do within the
channel, such as authentication, encryption and digital
signature.</xs:documentation>



<xs:element ref="sicd:Credential" minOccurs="0" />
<xs:element ref="sicd:Confidential" minOccurs="0" />
<xs:element ref="sicd:Integrity" minOccurs="0" />
<xs:element ref="sicd:Authorization" minOccurs="0">
    <xs:annotation>
        <xs:documentation>The SAML attribute assertion
        for the sending CP that will be pass to the
        receiving service. This will be shown in the end-
        to-end security channel.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element ref="sicd:NonRepudiation" minOccurs="0" />
<xs:element ref="sicd:NonRepudiationReceipt"
    minOccurs="0" />
<xs:element ref="sicd:SecurityContainer" minOccurs="0">
    <xs:annotation>
        <xs:documentation>This will be the container for
        those piggy back security related
        objects.</xs:documentation>
    </xs:annotation>
</xs:element>

<xs:attribute name="channelId" type="xs:string" use="optional"
/>

```

```
<xs:attribute name="sourceConnector" type="xs:string"
    use="required" />
<xs:attribute name="targetConnector" type="xs:string"
    use="required" />
</xs:complexType>
</xs:element>
<xs:complexType name="SecurityContractType">
    <xs:sequence>
        <xs:element ref="sicd:SecurityPolicies" />
        <xs:element ref="sicd:SecurityChannel"
            maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>
</xs:schema>
```

## InteroperabilityContract.XML

```

<?xml version="1.0" ?>
- <!--
edited with XML Spy v4.3 U (http://www.xmlspy.com) by Ernest Beffel (same)
-->
= <InteroperabilityContract
  xmlns="publicid:com.commerceone:schemas/soapextension/contract/v1
_0/InteroperabilityContract.xsd"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:general="publicid:com.commerceone:schemas/soapextension/cont
ract/general/v1_0/GeneralContract.xsd"
  xmlns:route="publicid:com.commerceone:schemas/soapextension/contra
ct/routing/v1_0/RoutingContract.xsd"
  xmlns:security="publicid:com.commerceone:schemas/soapextension/cont
ract/security/v1_0/SecurityContract.xsd"
  xmlns:xform="publicid:com.commerceone:schemas/soapextension/contr
act/transformation/v1_0/TransformationContract.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="publicid:com.commerceone:schemas/soapextension/
contrar/v1_0/InteroperabilityContract.xsd
http://schemas.commerceone.com/schemas/soapextension/contract/v1\_0/InteroperabilityContract.xsd">
= <GeneralContract ChoreographyID="ccns:orderManagement"
  MessageType="ONEWAY" CollaborativeInteraction="true"
  ICDTimeToLive="123456" MessageTimeToLive="2147483647"
  MessageArchived="true" BusinessIntelligence="true" ContractID="x-
  ccns:commerceone.com:CollaborationParty::buyParty-x-
  ccns:commerceone.com:CollaborationParty::sellParty"
  QualityOfService="EXACTLYONCE">
= <general:From>
= <general:FromAddress>
= <general:Party>x-
  ccns:commerceone.com:CollaborationParty::buyParty</
  general:Party>
= <general:ServiceActivity>
  <general:Service Version="1.0" EnvelopeProtocol="C1
    SOAP
    1.0">A:consumerOrderManagement</general:Service
  >
  <general:Activity>sendOrder</general:Activity>
</general:ServiceActivity>
</general:FromAddress>
<general:SenderDDID>9f76db48-784d-1000-b0d5-
0a0a02030002</general:SenderDDID>
</general:From>
= <general:To>
= <general:ToAddress>
= <general:Party>x-
  ccns:commerceone.com:CollaborationParty::sellParty</
  general:Party>
= <general:ServiceActivity>
  <general:Service Version="1.0" EnvelopeProtocol="C1
    SOAP
  >
```

```

    1.0">A:providerOrderManagement</general:Service
    >
    <general:Activity>processOrder</general:Activity>
    </general:ServiceActivity>
    </general:ToAddress>
    <general:ReceiverDDID>9f76db48-784d-1000-b0d5-
        0a0a02030001</general:ReceiverDDID>
    </general:To>
    ⊂ <general:ErrorHandling SenderAcceptsAsyncError="true">
        ⊂ <general:SendAsyncResultResponseTo>
            <general:Service Version="1.0" EnvelopeProtocol="C1 SOAP
                1.0">A:consumerOrderManagement</general:Service>
            <general:Activity>sendOrder</general:Activity>
            </general:SendAsyncResultResponseTo>
        </general:ErrorHandling>
    ⊂ <general:DeliveryReceiptHandling SenderRequiresDeliveryReceipt="true"
        IsAsyncDeliveryReceiptAcceptedBySender="true"
        ReceiverCanGenerateAsyncDeliveryReceipt="true">
        ⊂ <general:SendAsyncResultDeliveryReceiptTo>
            <general:Service Version="1.0" EnvelopeProtocol="C1 SOAP
                1.0">A:consumerOrderManagement</general:Service>
            <general:Activity>DeliveryReceiptConsumer</general:Activity>
            </general:SendAsyncResultDeliveryReceiptTo>
        </general:DeliveryReceiptHandling>
    <general:RequiredMessagePart PartName="Order" DocIDRequired="true"
        Location="attachment" MimeType="text/xml" Root="true"
        XML="false" />
    <general:RequiredMessagePart PartName="Image"
        DocIDRequired="false" Location="attachment"
        MimeType="image/jpeg" Root="false" XML="false" />
    <general:OptionalMessagePart PartName="someXMLPart"
        DocIDRequired="false" Location="soapbody" MimeType="text/xml"
        Root="false" XML="false" />
    ⊂ <general:SendingConnectorCapabilities>
        ⊂ <general:Attribute
            Name="Messaging.SupportDeliveryReceiptRequest">
            <general:Value>None</general:Value>
        </general:Attribute>
        ⊂ <general:Attribute Name="Messaging.ConversationData">
            <general:Value>rrn:org.soapextensions:schemas/highpe
                rformancesoap/conversationdata/v1_0/ConversationD
                ata</general:Value>
        </general:Attribute>
        ⊂ <general:Attribute Name="Messaging.AddressInfo">
            <general:Value>rrn:org.soapextensions:schemas/highpe
                rformancesoap/addressinfo/v1_0/AddressInfo</genera
                l:Value>
        </general:Attribute>
        ⊂ <general:Attribute Name="Messaging.MessageIdentity">
            <general:Value>rrn:org.soapextensions:schemas/highpe

```

```
    rformancesoap/messageidentity/v1_0/MessageIdentit
    y</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Archiving.Archiving">
    <general:Value>Yes</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.MessageTimeData">

    <general:Value>rrn:org.soapextensions:schemas/highpe
    rformancesoap/messagetimedata/v1_0/MessageTimeD
    ata</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.Privacy">

    <general:Value>http://schemas.xmlsoap.org/ws/2002/
    04/secext</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.Credential">
    <general:Value>None</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.SecurityAssertion">

    <general:Value>http://schemas.xmlsoap.org/ws/2002/
    04/secext</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.Integrity">

    <general:Value>http://schemas.xmlsoap.org/ws/2002/
    04/secext</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.Manifest">

    <general:Value>rrn:org.soapextensions:schemas:highper
    formancesoap/manifest/v1_0/Manifest</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Transformation.Transformation">
    <general:Value>Yes</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.Reliability">
    <general:Value>None</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.ReturnAddress">
    <general:Value>None</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.MessageEnvelope">
    <general:Value>SOAP WA 1.2</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Archiving.Mining">
    <general:Value>No</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Security.Encryption">
    <general:Value>Message Receiver</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Security.Signing">
    <general:Value>Message Sender</general:Value>
```

```
</general:Attribute>
- <general:Attribute Name="Messaging.TestMode">

    <general:Value>rrn:org.soapextensions:schemas/highpe
rformancesoap/testmode/v1_0/TestMode</general:Valu
e>
</general:Attribute>
- <general:Attribute Name="Messaging.Body">
    <general:Value>Optional</general:Value>
</general:Attribute>
- <general:Attribute Name="Messaging.ContractData">

    <general:Value>rrn:org.soapextensions:schemas/highpe
rformancesoap/contractdata/v1_0/ContractData</gene
ral:Value>
</general:Attribute>
- <general:Attribute Name="Messaging.ReturnDocument">

    <general:Value>rrn:org.soapextensions:schemas/highpe
rformancesoap/returndocument/v1_0/ReturnDocumen
t</general:Value>
</general:Attribute>
- <general:Attribute Name="Messaging.GenerateDeliveryReceipt">
    <general:Value>Yes</general:Value>
</general:Attribute>
</general:SendingConnectorCapabilities>
- <general:ReceivingConnectorCapabilities>
- <general:Attribute
    Name="Messaging.SupportDeliveryReceiptRequest">
    <general:Value>None</general:Value>
</general:Attribute>
- <general:Attribute Name="Messaging.ConversationData">

    <general:Value>rrn:org.soapextensions:schemas/highpe
rformancesoap/conversationdata/v1_0/ConversationD
ata</general:Value>
</general:Attribute>
- <general:Attribute Name="Messaging.AddressInfo">

    <general:Value>rrn:org.soapextensions:schemas/highpe
rformancesoap/addressinfo/v1_0/AddressInfo</genera
l:Value>
</general:Attribute>
- <general:Attribute Name="Messaging.MessageIdentity">

    <general:Value>rrn:org.soapextensions:schemas/highpe
rformancesoap/messageidentity/v1_0/MessageIdentit
y</general:Value>
</general:Attribute>
- <general:Attribute Name="Archiving.Archiving">
    <general:Value>Yes</general:Value>
</general:Attribute>
- <general:Attribute Name="Messaging.MessageTimeData">

    <general:Value>rrn:org.soapextensions:schemas/highpe
```

```
  rformancesoap/messagetimedata/v1_0/MessageTimeD
    ata</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.Privacy">

    <general:Value>http://schemas.xmlsoap.org/ws/2002/
      04/secext</general:Value>
    </general:Attribute>
  -<general:Attribute Name="Messaging.Credential">
    <general:Value>None</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.SecurityAssertion">

    <general:Value>http://schemas.xmlsoap.org/ws/2002/
      04/secext</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.Integrity">

    <general:Value>http://schemas.xmlsoap.org/ws/2002/
      04/secext</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.Manifest">

    <general:Value>rrn:org.soapextensions:schemas:highper
      formancesoap/manifest/v1_0/Manifest</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Transformation.Transformation">
    <general:Value>Yes</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.Reliability">
    <general:Value>None</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.ReturnAddress">
    <general:Value>None</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.MessageEnvelope">
    <general:Value>SOAP WA 1.2</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Archiving.Mining">
    <general:Value>No</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Security.Encryption">
    <general:Value>Message Receiver</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Security.Signing">
    <general:Value>Message Sender</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.TestMode">

    <general:Value>rrn:org.soapextensions:schemas/highpe
      rformancesoap/testmode/v1_0/TestMode</general:Value>
  </general:Attribute>
  -<general:Attribute Name="Messaging.Body">
    <general:Value>Optional</general:Value>
```

```

</general:Attribute>
   _ <general:Attribute Name="Messaging.ContractData">

    <general:Value>rrn:org.soapextensions:schemas/highpe
      rformancesoap/contractdata/v1_0/ContractData</gene
      ral:Value>
    </general:Attribute>
   _ <general:Attribute Name="Messaging.ReturnDocument">

    <general:Value>rrn:org.soapextensions:schemas/highpe
      rformancesoap/returndocument/v1_0/ReturnDocumen
      t</general:Value>
    </general:Attribute>
   _ <general:Attribute Name="Messaging.GenerateDeliveryReceipt">
    <general:Value>Yes</general:Value>
    </general:Attribute>
  </general:ReceivingConnectorCapabilities>
</GeneralContract>
   _ <RoutingContract>
     _ <route:RouteNode connector="x-
      ccns:cup.commerceone.com:connector::buy isNative="true"
      connectorFunction="service-send" preICDComputation="true">
        <route:EntryChannel envelopeProtocol="C1 SOAP 1.0"
          transportSupportedMessageType="both"
          transportPhysicalAddress="https://uranus.cup.commerceone.c
          om:8433/buy/soap" transportProtocol="https, basic
          authentication" transportReliable="true" transportNative="true"
          />
        <route:ExitChannel envelopeProtocol="C1 SOAP 1.0"
          transportSupportedMessageType="both"
          transportPhysicalAddress="https://uranus.cup.commerceone.c
          om:8433/buy/soap" transportProtocol="https, basic
          authentication" transportReliable="true" transportNative="true"
          />
      </route:RouteNode>
     _ <route:RouteNode connector="x-
      ccns:cup.commerceone.com:connector::sell isNative="true"
      connectorFunction="service-receive" preICDComputation="false">
        <route:EntryChannel envelopeProtocol="C1 SOAP 1.0"
          transportSupportedMessageType="both"
          transportPhysicalAddress="https://saturn.cup.commerceone.c
          om:8433/sell/soap" transportProtocol="https, basic
          authentication" transportReliable="true" transportNative="true"
          />
        <route:ExitChannel envelopeProtocol="C1 SOAP 1.0"
          transportSupportedMessageType="both"
          transportPhysicalAddress="https://saturn.cup.commerceone.c
          om:8433/sell/soap" transportProtocol="https, basic
          authentication" transportReliable="true" transportNative="true"
          />
      </route:RouteNode>
   </RoutingContract>
   _ <TransformationContract>
     _ <xform:DocumentToTransform>

```

```

<xform:SourceDocID>x-
  ccns:docid::rrn:org.xcbl:schemas/xcbl/v3_5/xcbI35.xsd:ord
  er:3.5</xform:SourceDocID>
<xform:PartName>Order</xform:PartName>
<xform:Attachment>false</xform:Attachment>
= <xform:TransformationMap>
  <xform:Connector>x-
    ccns:cup.commerceone.com:connector::buy</xform:Con
    nector>
= <xform:StartDoc>
  <xform:DocURI>x-
    ccns:docid::rrn:org.xcbl:schemas/xcbl/v3_5/xcbI3
    5.xsd:order:3.5</xform:DocURI>
  <xform:DocName>Order</xform:DocName>

  <xform:Namespace>rrn:org.xcbl:schemas/xcbl/v3_5
  /xcbl35.xsd</xform:Namespace>
  <xform:Version>3.5</xform:Version>
</xform:StartDoc>
= <xform:EndDoc>
  <xform:DocURI>x-
    ccns:docid::rrn:org.xcbl:schemas/xcbl/v4_0/order
    management/v1_0/ordermanagement.xsd:order:4.
    0.1.0</xform:DocURI>
  <xform:DocName>Order</xform:DocName>

  <xform:Namespace>rrn:org.xcbl:schemas/xcbl/v4_0
  /ordermanagement/v1_0/ordermanagement.xsd</
  xform:Namespace>
  <xform:Version>4.0.1.0</xform:Version>
</xform:EndDoc>

  <xform:CommunityID>commerceone.com</xform:Commun
  ityID>
<xform:TransformationMapURI>x-
  ccns:transformationMap:OrderxcbI35ToxcbI4010</xform
  :TransformationMapURI>
</xform:TransformationMap>
</xform:DocumentToTransform>
</TransformationContract>
= <SecurityContract>
  <security:SecurityPolicies>
    <security:AuthenticationPolicies>
      <security:BasicCredentialPolicy PolicyId="P-
        AuthenBasicSource">

        <security:CredentialPolicyAlgorithm>Basic</security:Cre
        dentialPolicyAlgorithm>

        <security:AuthenticateMode>SOURCE</security:Authent
        icateMode>
      </security:BasicCredentialPolicy>
    </security:AuthenticationPolicies>
= <security:SignaturePolicies>
  <security:XMLsigPolicy PolicyId="P-XMLSsignatureRSA-MD5-
    EXC14N">

```

```

<security:SignaturePolicyAlgorithm>http://www.w3.org/2000/09/xmldsig#</security:SignaturePolicyAlgorithm>

<security:SignatureAlgorithm>MD5withRSA</security:SignatureAlgorithm>
<security:HashFunction>MD5</security:HashFunction>

<security:CanonicalizationMethod>http://www.w3.org/2001/10/xml-exc-c14n#</security:CanonicalizationMethod>

<security:Transform>http://msdn.microsoft.com/ws/2002/01/Security#RoutingSignatureTransform</security:Transform>
</security:XMLEncPolicy>
</security:SignaturePolicies>
- <security:EncryptionPolicies>
  - <security:XMLEncryptionPolicy PolicyId="P-XMLEncryptAES-128-RSA-2048">

    <security:EncryptionPolicyAlgorithm>http://www.w3.org/2001/04/xmlenc#</security:EncryptionPolicyAlgorithm>

    <security:EncryptionMethod>http://www.w3.org/2001/04/xmlenc#aes128-cbc</security:EncryptionMethod>
    <security:KeySize>2048</security:KeySize>

    <security:SymmetryKeySize>128</security:SymmetryKeySize>

    <security:KeyEncryptionMethod>http://www.w3.org/2001/04/xmlenc#rsa-1_5</security:KeyEncryptionMethod>
    </security:XMLEncryptionPolicy>
  </security:EncryptionPolicies>
- <security:EncryptionKeyInfo KeyOwner="x-ccns:commerceone.com:CollaborationParty::sellParty">

  <security:PublicKeyID>DefaultTestCert</security:PublicKeyID>
  - <security:X509Data>

    <security:X509Certificate>LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tTUIjREZEQ0NBZnlnQXdJQkFnSUVQT0ZQSVRBTkJn a3Foa2lHOXcwQkFRVUZBREI2TVFzd0NRWURWUvFHRXdKVIV6RVZNQk1HQTFVRUNoTU1RMjI0YldWeVkyVWdUMjVsTVMwd0t3WURWUVFMRXISVWFHbH pJRUSCSUdseklHWnZjaUIwWiOMGFXNW5JSEIxY25CdmMyVnpJRzI1YkhreEpUQWpCZ05WQkFNvUhFTnZiVzFsY21ObElFOXVaU0JvWlhOMEIFTkJJRkp2YjNRZ0i6RXdIaGNOTURJd05URTBNVGmxTxpNM1doY05NRE13TIRFME1UWTFNek0zV2pCb01S</security:X509Certificate>
  </security:X509Data>
</security:EncryptionKeyInfo>

```

```

Z3dGZ1IEViFREV3OUVZWFPwWkNCVvPYTjBJREi3T
URJeEVqQVFcZ05WQkFjVENVTjFjR1Z5ZEEds
dWJ6RVVNQkIHQTFVRUN4TUxSVzVuYVc1bFpYSnBi
bWN4RIRBVEJnTlZCQW9UREVOdmJXMWxjbU5s
SUU5dVpURUxNQWtHQTFVRUJoTUNWVk13Z1o4d0
RRWUpLb1pJaHZjTkFRRUJCUUFEZ1kwQU1JR0pB
b0dCQU5nc2pTQlxjcFp2QnVDQ2lTHR3RGFkaFZEM
GNLRXJuQ3M2azg5UEhSUGJSMFdYOHBDUzBy
ZWxIMkcyADmxNU5vNGkzQVNIdHzhYmdHeiRVFnI
R2EzcWtNYmVLNDZTSGxtTkJOTUp2YUkvMmZV
QIBxdkkzejILTVJSTGh3eUhCMEdFNmUvSzdnVGZkSU
o0MUJobTzSmcwYzJqZ041cWt1d3FZQkV4
eWN1MUFnTUJBQUDqT0RBMc1DY0dBMVVkRVFRZ0
1CNkJIRzE1VkdWemRFVnRZV2xzUUDOdmJXMWpa
WEpqWlc5dVpTNWpIMB3Q3dZRFZSMFBCQVFEQWd
YZ01BMEdDU3FHU0lIM0RRRUJCUVVBQTRJQkFR
Q0UrNEVaUWZYZWpmVnBsbXEzZnFtUjJZSGZhczErc
XA0MUg4UWRmNmRESXB1YKZ2OUxochorYkc2
c2hWQlptMVpYVXphaHI6N2Q3Z2U3V0MxR2FZVjFH
YldFTXJMUKzkeXM2c1VIQkZNbHZuNkZPRjNq
OHdMY3JuN2FFN3pRMEMwa2U5LzVVNVBHTnlaZWV
aUGNLNTIKM0hPdWpzbXUvaENPVW1OOXZVM2M3
MHVjMmhRaE96aExJQ0VIQ2VTRDFCd2hEMXNkdXZ
mNnVOanAzUGp2eUpCaklTeDVxY2UwS25oQmxp
cDR3ejRNTWxpdEtTdkFXSElqR1Bvb0w0N01ac3I4N
3RLamJHaTgxcWJrQ3hiYlZ1dEloYmkzZDRn
aW1Ockc1RXJ0dUUxNmwwRW9GUkJLU2VRTXd2cFd
GULLiN2YreWtKVGE5ZVRLaWF4R2hOcDR4dnc5
LS0tLS1FTkQgQ0VSVEIGSUNBVEUtLS0tLQ==</secu
rity:X509Certificate>
</security:X509Data>
</security:EncryptionKeyInfo>
</security:SecurityPolicies>
- <security:SecurityChannel channelId="CHANNEL1" sourceConnector="x-
  ccns:cup.commerceone.com:connector::buy" targetConnector="x-
  ccns:cup.commerceone.com:connector::sell">
- <security:Credential AlgorithmId="P-AuthenBasicSource"
  SequenceID="4">
  <security:PartyID>x-
    ccns:commerceone.com:CollaborationParty::buyParty</
  security:PartyID>
  </security:Credential>
- <security:Confidential AlgorithmId="P-XMLEncryptAES-128-RSA-
  2048">
  <security:PublicKeyName KeyOwner="x-
    ccns:commerceone.com:CollaborationParty::sellParty">
    DefaultTestCert</security:PublicKeyName>
  <security:MessagePart PartName="Order" isOptional="false" />
  <security:MessagePart PartName="Image" isOptional="false" />
  </security:Confidential>
- <security:Integrity AlgorithmId="P-XMLSignatureRSA-MD5-
  EXC14N">
  <security:PublicKeyName KeyOwner="x-
    ccns:commerceone.com:CollaborationParty::buyParty">
    DefaultTestCert</security:PublicKeyName>

```

```
<security:MessagePart PartName="Order" isOptional="false" />
</security:Integrity>
</security:SecurityChannel>
</SecurityContract>
</InteroperabilityContract>
```

### ComputeSecurityContract.XML

```

<?xml version="1.0" ?>
= <prefix_0:SecurityContractICD
  xmlns:prefix_0="publicid:com.commerceone:schemas/soapextension/contract/security/v1_0/SecurityContract.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
= <prefix_0:SecurityPolicies>
=   <prefix_0:AuthenticationPolicies>
=     <prefix_0:X509CredentialPolicy PolicyId="P-AuthenX.509Source">
        <prefix_0:CredentialPolicyAlgorithm>X.509v3</prefix_0:CredentialPolicyAlgorithm>
        <prefix_0:AuthenticateMode>SOURCE</prefix_0:AuthenticateMode>
      </prefix_0:X509CredentialPolicy>
    </prefix_0:AuthenticationPolicies>
=   <prefix_0:SignaturePolicies>
=     <prefix_0:XMLEdgPolicy PolicyId="P-XMLSigPolicy-RSA-MD5-C14N">
        <prefix_0:SignaturePolicyAlgorithm>http://www.w3.org/2000/09/xmldsig#</prefix_0:SignaturePolicyAlgorithm>
        <prefix_0:SignatureAlgorithm>MD5withRSA</prefix_0:SignatureAlgorithm>
        <prefix_0:HashFunction>MD5</prefix_0:HashFunction>
        <prefix_0:CanonicalizationMethod>http://www.w3.org/TR/2000/CR-xml-c14n-20001026</prefix_0:CanonicalizationMethod>
        <prefix_0:Transform>http://msdn.microsoft.com/ws/2002/01/Security#RoutingSignatureTransform</prefix_0:Transform>
      </prefix_0:XMLEdgPolicy>
    </prefix_0:SignaturePolicies>
=   <prefix_0:EncryptionPolicies>
=     <prefix_0:XMLEncrypt3DES-RSA-2048">
        <prefix_0:EncryptionPolicyAlgorithm>http://www.w3.org/2001/04/xmlenc#</prefix_0:EncryptionPolicyAlgorithm>
        <prefix_0:EncryptionMethod>http://www.w3.org/2001/04/xmlenc#3des-cbc</prefix_0:EncryptionMethod>
        <prefix_0:KeySize>2048</prefix_0:KeySize>
        <prefix_0:KeyEncryptionMethod>http://www.w3.org/2001/04/xmlenc#rsa-1_5</prefix_0:KeyEncryptionMethod>
      </prefix_0:XMLEncrypt3DES-RSA-2048">
    </prefix_0:EncryptionPolicies>
=   <prefix_0:EncryptionKeyInfo KeyOwner="x-ccns:commerceone.com:CollaborationParty::sellParty">
      <prefix_0:PublicKeyID>DefaultTestCert</prefix_0:PublicKeyID>
    </prefix_0:EncryptionKeyInfo>
  </prefix_0:SecurityPolicies>
</prefix_0:SecurityContractICD>

```

```

_- <prefix_0:X509Data>

    <prefix_0:X509Certificate>LS0tLS1CRUDJTbDRVJUSUZJQ
    0fURS0tLS0tTUIJREZEQ0NBZnlnQXdJQkFnSUVQT0ZQSV
    RBTkJn
    a3Foa2lHOXcwQkFRVUZBREI2TVFzd0NRWURWUVFHRX
    dKViV6RVZNQk1HQTFRUNoTU1RMjI0YIdW
    eVkyVWdUMjVsTVMwd0t3WURWUVFMRXISVWFhbHpJR
    U5CSUdsekIHWNZjaUIwWiOMGFXNWSJSEIx
    Y25CdmMyVnpJRzi1YkhreEpUQWpCZ05WQkFNVuUhFTnZ
    iVzFsY21ObElFOXVaU0JVWlhOMEIFTkJJ
    Rkp2YjNRZ0i6RXdIaGNOTURJd05URTBNVGMxTXpNM1d
    oY05NRE13TIRFME1UWTFNek0zV2pCb01S
    Z3dGZ1IEVIFRREV3OUVZWFpwWkNCVvPYTjBJREI3TURJ
    eEVqQVFCZ05WQkFjVENVTjFjR1Z5ZEds
    dWJ6RVVNQkIHQTFRUN4TUXSVzVuYYC1bFpYSnBibWN
    4RIRBVEnTlZCQW9UREVOdmJXMWxjbU5s
    SUU5dVpURUxNQWtHQTFVRUJoTUNWVki3Z1o4d0RRW
    UpLb1pJaHZjTkFRRUJCUUFEZ1kwQU1JR0pB
    b0dCQU5nc2pTQlxjcFp2QnVDQ2ITTHR3RGFkaFZEMGNL
    RXJuQ3M2azg5UEhSUGJSMFdYOHBDUzBy
    ZWxIMkcyadMxNU5vNGkzQVNidH2hYmdHellRVFNiR2Ez
    cWtNYmVLNDZTSGxtTkJOTUp2YUkvMmZV
    QIBxdkkzejILTvjSTGh3eUhCMEdFNmUvSzdnVGZkSu0M
    UJobTZzSmcwYzJqZ041cWt1d3FZQkV4
    eWN1MUFnTUJBQUdqT0RBMc1DY0dBMVVkRVFRZ01CN
    kJIRzE1VkdWemRFVnRZV2xzUUdOdmJXMWpa
    WEpqWlc5dVpTNWpiMjb3Q3dZRFZSMFBCQVFEQWdYZ0
    1BMEdu3FHU0liiMORRUJCUVVBQTRJQkFR
    Q0UrNEVaUWZYZWpmVnBsbXEzZnFtUjjZSGZhczErcXA0
    MUg4UWRmNmRESXBYkZ2OUxocnorYkc2
    c2hWQlptMVpYVXphaHI6N2Q3Z2U3V0MxR2FZVjFHYldF
    TXJMUK2keXM2c1VIQkZNbHzuNkZPRjNq
    OHdMY3JuN2FFN3pRMEMwa2U5LzVVNVBHTnlaZWVaUG
    NLNTIKM0hPdWpbzXUvaENPVW1OOXZVM2M3
    MHVjMMhRaE96aExJQ0VlQ2VTRDFCd2hEMXNkdXZmNn
    VOanAzUGp2eUpCakITeDVxY2UwS25oQmxp
    cDR3ejRNTWxpdeTdkFXSElqR1Bvb0w0N01ac3I4N3RLa
    mJHaTgxcWJrQ3hiYlZ1dEloYmkzZDRn
    aW1Ockc1RXJ0dUUxNmwvRW9GUkJLU2VRTXd2cFdGUll
    iN2YreWtKVGE5ZVRLaWF4R2hOcDR4dnc5
    LS0tLS1FTkQgQ0VSVEIGSUNBVEUtLS0tLQ==</prefix_0:
    X509Certificate>
    </prefix_0:X509Data>
</prefix_0:EncryptionKeyInfo>
</prefix_0:SecurityPolicies>
_- <prefix_0:SecurityChannel channelId="CHANNEL1" sourceConnector="x-
    ccns:cup.commerceone.com:connector::buy" targetConnector="x-
    ccns:cup.commerceone.com:connector::sell">
    - <prefix_0:Credential AlgorithmId="P-AuthenX.509Source"
        SequenceID="4" DelegationFlag="false">
        <prefix_0:PublicKeyName>BuyerPublicKey</prefix_0:PublicKeyN
        ame>
    </prefix_0:Credential>
    - <prefix_0:Integrity AlgorithmId="P-XMLSignatureRSA-MD5-C14N">
```

```
<prefix_0:PublicKeyName  
    KeyOwner="OwnerA">BuyerPublicKey</prefix_0:PublicKeyNam  
e>  
<prefix_0:MessagePart PartName="Order" IsOptional="false" />  
</prefix_0:Integrity>  
</prefix_0:SecurityChannel>  
= <prefix_0:SecurityChannel channelId="CHANNEL2" sourceConnector="x-  
    ccns:cup.commerceone.com:connector::centerSell"  
    targetConnector="x-  
    ccns:cup.commerceone.com:connector::centerSell">  
= <prefix_0:Confidential AlgorithmId="P-XMLEncrypt3DES-RSA-2048">  
    <prefix_0:PublicKeyName KeyOwner="x-  
        ccns:commerceone.com:CollaborationParty::sellParty">Defa  
        ultTestCert</prefix_0:PublicKeyName>  
    <prefix_0:MessagePart PartName="Order" IsOptional="false" />  
    <prefix_0:MessagePart PartName="Image" IsOptional="false" />  
    </prefix_0:Confidential>  
</prefix_0:SecurityChannel>  
</prefix_0:SecurityContractICD>
```

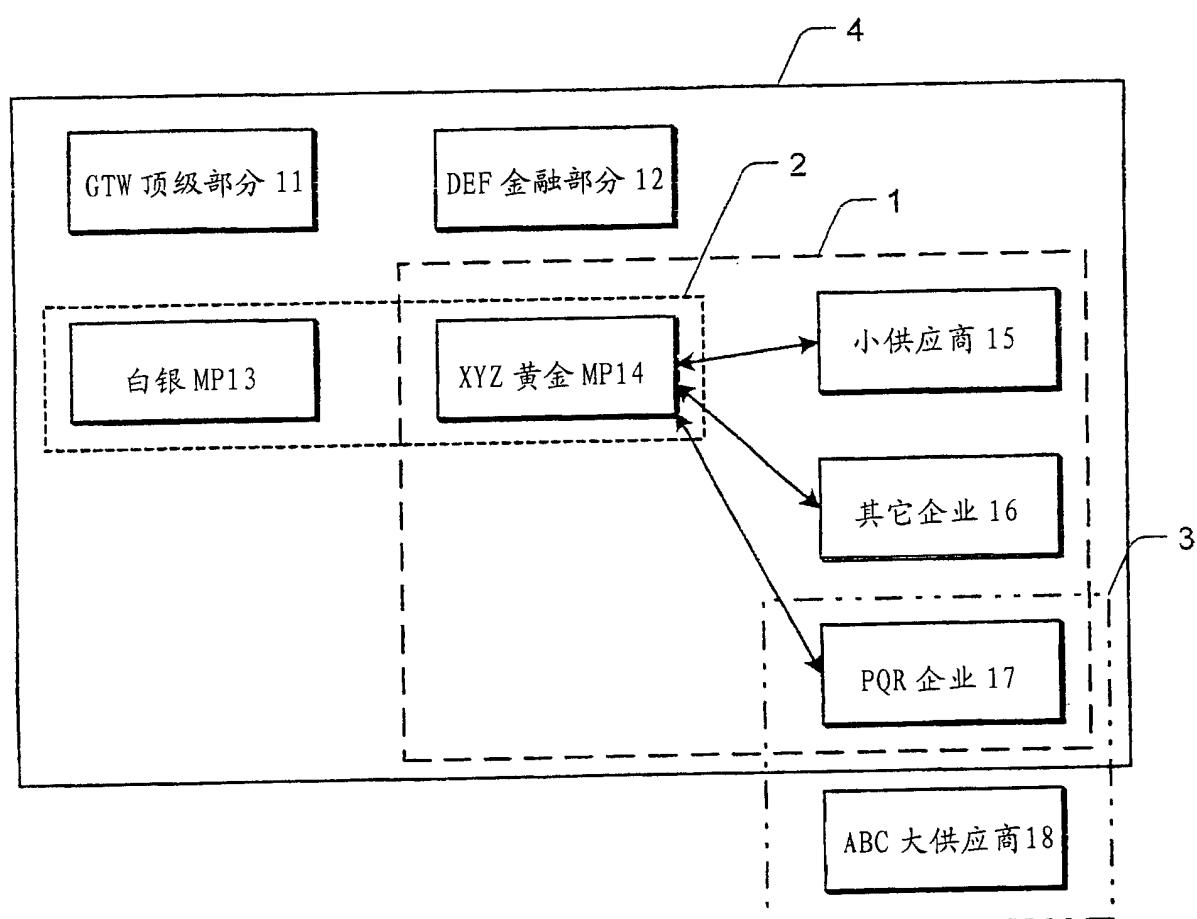


图 1

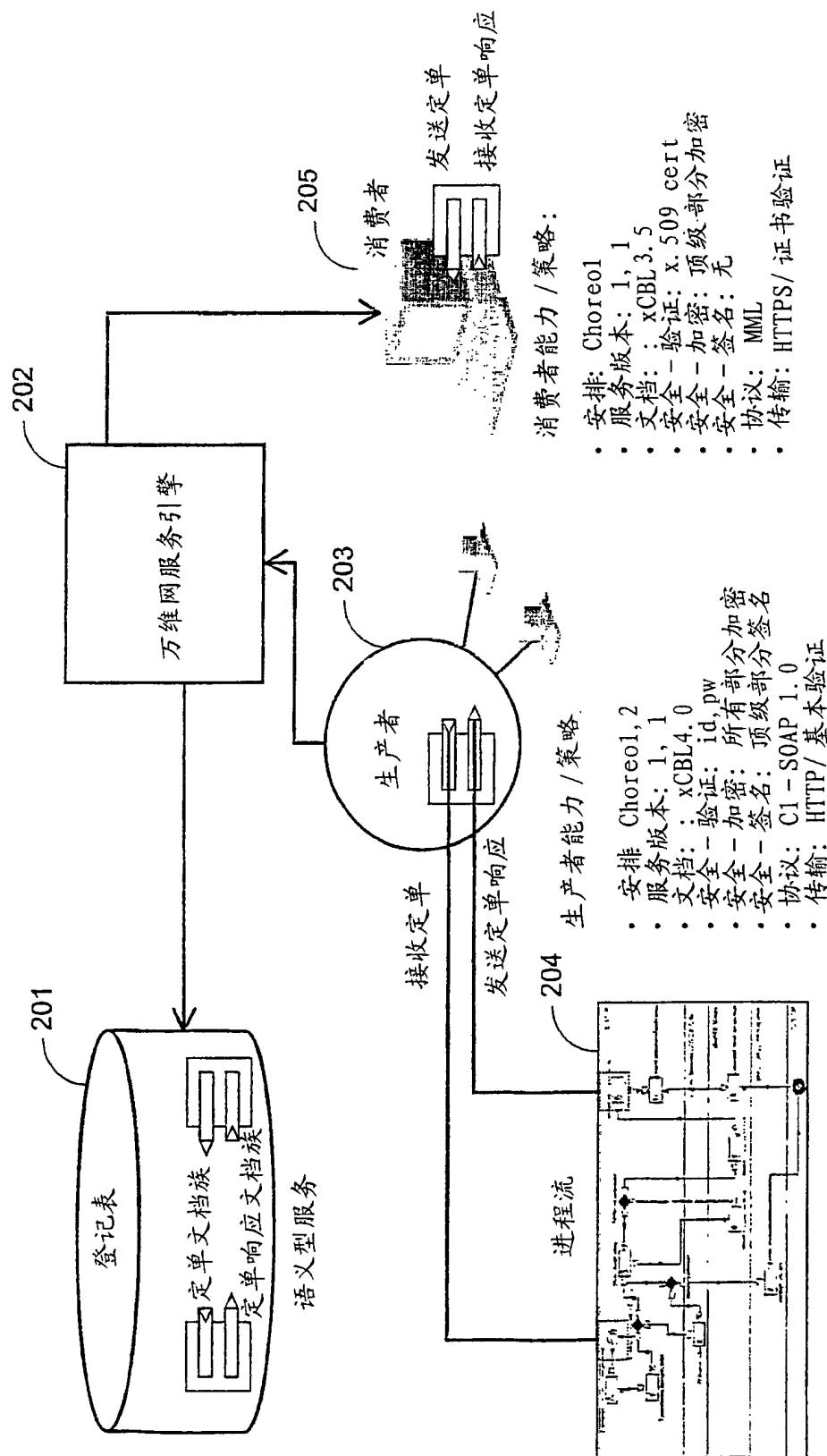


图 2

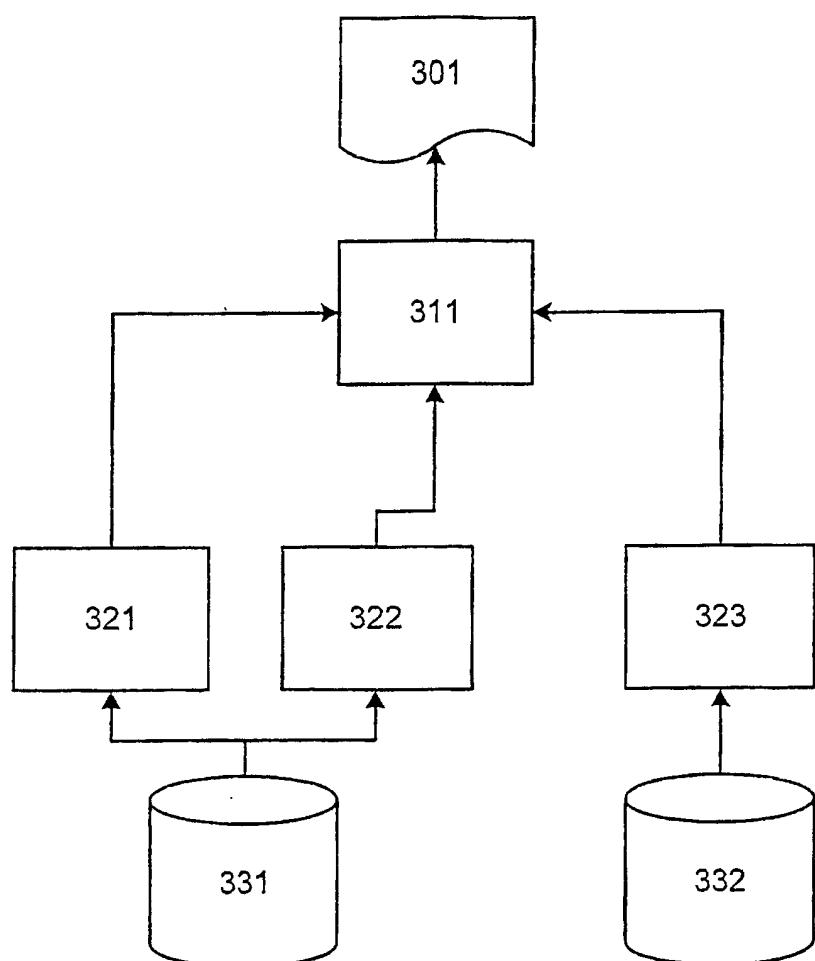


图 3