Verteilte Anwendungsumgebungen mit XML-Web-Services

Autor(en): **Müller, Urs**

Objekttyp: **Article**

Zeitschrift: **Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association Suisse des Electriciens, de l'Association des Entreprises électriques suisses**

Band (Jahr): **93 (2002)**

Heft 19

PDF erstellt am: **04.06.2024**

Persistenter Link: https://doi.org/10.5169/seals-855458

Nutzungsbedingungen

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.
Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.
Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

Haftungsausschluss

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

Ein Dienst der *ETH-Bibliothek*
ETH Zürich, Rämistrasse 101, 8092 Zürich, Schweiz, www.library.ethz.ch

http://www.e-periodica.ch

Verteilte Anwendungsumgebungen mit XML-Web-Services

XML-Web-Services sind eine neue Applikationstechnologie mit hohem Potenzial. Auf der Grundlage von standardisierten Spezifikationen wie XML, SOAP, UDDI und WSDL ermöglichen sie es, über Unternehmensgrenzen hinweg Geschäftsprozesse in Form von Services frei verfügbar und nutzbar zu machen – das Internet wird zu einem Ort integrierter, system- und plattformunabhängiger Anwendungen. Das Ziel der gegenwärtigen Entwicklung ist eine globale XML-Web-Services-Architektur mit modularen Infrastrukturprotokollen, mit denen Webanwendungen flexibel und sicher zu gänzlich neuen Umgebungen im Sinne wirtschaftlicher «Ökosysteme» integriert werden können.

Die Verbindung bestehender Legacy-Systeme[1] innerhalb eines Unternehmens, die Integration von Applikationen über die Unternehmensgrenzen hinweg, die nahtlose Zusammenarbeit mit Geschäftspartnern und Kunden – dies sind heute die Grundanforderungen von Unternehmen an ihre IT-Infrastrukturen.

Statt isolierte Probleme zu lösen, sollen integrierte, verteilte Umgebungen geschaffen werden, die vollumfänglich auf den Kundennutzen ausgerichtet sind. XML[2]-Web-Services sind die zukunftsträchtige Antwort auf diese Herausforderungen. Zahlreiche Branchenleader – darunter Schwergewichte wie Microsoft und IBM – arbeiten mit Standardgremien wie W3C[3] oder OASIS[4] an der Weiterentwicklung von gemeinsamen Standards, Entwicklungsumgebungen und konkreten Lösungsangeboten.

Urs Müller

Das Konzept der XML-Web-Services ist äusserst einfach. Auf der Grundlage des offenen Datenformat-Standards XML können Webanwendungen – aber auch bestehende Legacy-Applikationen – zu plattformübergreifenden, systemunabhängigen Services erweitert werden, die auch Dritte ohne zusätzlichen Programmieraufwand nutzen können. Damit wird das Internet zu einem gigantischen Fundus an Anwendungen, aus dem sich Unternehmen beliebig bedienen können, um ihren Mitarbeitern, Partnern und Kunden je nach Bedarf einzelne Applikationen oder auch Kombinationen von mehreren Diensten zur Verfügung zu stellen.

Bereits heute setzen zahlreiche Unternehmen wie etwa Fluggesellschaften, Versicherungen und Behörden XML-Web-Services ein. Die bisherigen Erfahrungen zeigen, dass sich – dank hoher Wiederverwendbarkeit und einer kurzen Time-to-Market – durch den Einsatz von XML-Web-Services signifikante Kosteneinsparungen erzielen lassen. Durch die Integration von Backend-Anwendungen[5] und Applikationen von wichtigen Partnern lässt sich die Effizienz von Geschäftsprozessen wesentlich steigern. Zudem können neue Vertriebskanäle erschlossen werden, indem eigene Geschäftsfunktionalitäten für Partnerunternehmen zugänglich gemacht werden.

Beispiele für den Einsatz von intelligenten Webdiensten sind

- die skandinavische Fluggesellschaft SAS, die ihren Kunden in einem ersten Schritt mittels XML-Web-Services auf der Basis der .NET-Plattform (Kasten 1) von Microsoft aktuelle Fluginformationen für verschiedene mobile Geräte zur Verfügung stellt. Weitere Funktionen werden nach und nach implementiert.
- das amerikanische Versicherungsunternehmen Unum Provident Direct (UDP), das XML-Web-Services für die interaktive, rasche Kommunikation mit Geschäftspartnern einsetzt. Diese erhalten in Echtzeit Zugriff auf Informationen und Dienstleistungen von UDP, was die Zufriedenheit und das Geschäftspotenzial der Partner markant erhöht.

Die neue Offenheit

Ein XML-Web-Service ist eine Softwarekomponente, die eine Geschäftsfunktion darstellt und von anderen Anwendungen (Client, Server oder anderen XML-Web-Services) über öffentliche Netzwerke genutzt werden kann, wobei allgemein verfügbare Protokolle wie beispielsweise SOAP[6] (Kasten 2) über HTTP[7] zur Anwendung gelangen. Im Sinne einer Komponente repräsentiert ein XML-Web-Service eine Blackbox mit einer bestimmten Funktionalität, die flexibel eingesetzt werden kann, ohne dass deren Implementationsdetails bekannt sein müssen. Damit versprechen XML-Web-Services die lang ersehnte Integration verschiedener Plattformen und Anwendungen, herstellerübergreifend und auf der Basis von Standards. Dies eröffnet gänzlich neue Horizonte in Bereichen wie Enterprise Application Integration[8]

Microsoft .NET und XML-Web-Services

Die Microsoft .NET-Plattform ermöglicht die Entwicklung und Nutzung von XML-basierenden Anwendungen, Prozessen und Websites als Dienste, mit denen Informationen und Funktionen gemeinsam genutzt und kombiniert werden können – auf jeder Plattform und auf jedem Client. Die .NET-Plattform bietet eine umfassende Sammlung von Produkten für sämtliche Aspekte von XML-Web-Services, von der Entwicklung (Tools) über die Verwaltung (Server) und die Nutzung (modulare Dienste und intelligente Clients) bis zur eigentlichen Anwendererfahrung. Mit *Visual Studio .NET* von Microsoft steht – auf der Basis der .NET-Plattform – eine vollständige Entwicklungsumgebung zur Verfügung. *Visual Studio .NET* erlaubt es Entwicklern, sichere und skalierbare Anwendungen und XML-Web-Services in kurzer Zeit, in einer beliebigen Sprache und unter Nutzung der vorhandenen Systeme zu erstellen.

(EAI), bei der Realisierung von Prozessportalen, für unternehmensübergreifende Integrationen oder der Schaffung neuer Prozessstrukturen und Wertschöpfungsketten durch das Zusammenführen von Geschäftslogiken unterschiedlicher Firmen. Mithin entstehen dabei wirtschaftliche «Ökosysteme», die sich durch hohe Anpassungsfähigkeit an Marktveränderungen, Kundennachfrage sowie an neue Technologien auszeichnen.

Solche «Ökosysteme» sind mit herkömmlichen Komponententechnologien, die auf objektspezifischen Protokollen basieren, nur schwer zu realisieren, da diese eine homogene Infrastruktur für Client und Server voraussetzen – was bekanntlich im Web nicht der Fall ist und auch in Unternehmen ein bedeutendes Problem darstellt. Daher gehen XML-Web-Services einen anderen Weg. Aufbauend auf Internetstandards nutzen sie HTTP und XML als den kleinsten gemeinsamen Nenner. Somit kann jedes System, das HTTP und XML unterstützt, auch XML-Web-Services nutzen.

SOAP – das Fundament für Webdienste

XML-Web-Services konzentrieren sich auf das Wesentliche: Ein Client schickt mittels HTTP – oder einem anderen Protokoll – eine mit XML verpackte Nachricht an einen Server und dieser antwortet auf die Anfrage ebenfalls mit einer XML-Nachricht. Im Zentrum dieses Nachrichtenaustauschs steht SOAP, das Kommunikationsprotokoll für den Austausch von strukturierten Daten in einer dezentralen, verteilten Umgebung[9] (Bild 1). SOAP ist äusserst flexibel und kann damit von einfachen Nachrichtensystemen bis hin zu Remote Procedure Calls (RPC) eingesetzt werden (Kasten 2).

XML-Web-Services sind völlig unabhängig vom Transportprotokoll. Die SOAP-Spezifikationen legen fest, wie SOAP-Nachrichten über HTTP – das von der überwiegenden Mehrheit der XML-Web-Services derzeit benutzt wird – gesendet werden. Aber auch andere Transportprotokolle können eingesetzt werden. SOAP-Nachrichten können über SMTP[10], TCP[11] oder ein Instant-Messaging-Protokoll wie Jabber[12] geschickt werden. Diese Offenheit für unterschiedlichste Transportprotokolle ist ein wesentlicher Vorteil, weil HTTP nicht alle Kommunikationsbedürfnisse abdeckt. XML-Web-Services kommen bei Bedarf also auch ohne HTTP aus und lassen sich damit noch flexibler aneinanderkoppeln.

Die Gliederung von SOAP

Jede SOAP-Nachricht ist ein XML-Dokument, das aus dem SOAP-Envelope, dem optionalen SOAP-Header und dem SOAP-Body besteht.

Der Envelope ist das oberste XML-Element der Nachricht und umschliesst sie wie ein Umschlag mit XML-Tags. Er enthält die nötigen Namensräume sowie Attribute und Unterelemente, die ebenfalls durch einen Namensraum qualifiziert werden müssen. Der optionale Header enthält Angaben beispielsweise zu Adressen, Zahlungscodes oder Informationen über eine RPC-Interaktion. Er stellt damit einen flexiblen Erweiterungsmechanismus für Nachrichten zur Verfügung. Solche Erweiterungen können dezentralisiert, modular und ohne vorherige Kenntnisnahme durch den Kommunikationspartner geschehen. Typische Beispiele dafür sind Authentifizierung und Transaktionsmanagement. Im Body findet der Empfänger die eigentlichen Daten in Form einer selbstbeschreibenden Struktur oder als RPC-Schnittstelle. Eine mögliche SOAP-Nachricht ist in Bild 2 dargestellt.

SOAP stellt sicher, dass Programme miteinander reden können. Dies heisst noch nicht, dass sie sich auch verstehen. Die «Web-Service Description Language» (WSDL) legt dafür im Detail fest, welche Mitteilungen mit einem Service ausgetauscht bzw. welche seiner Methoden beansprucht werden können. Darüber hinaus gibt das WSDL-Dokument die Adresse des XML-Web-Service an und liefert weitere Informationen zum verwendeten Übermittlungsprotokoll. Das zu einem XML-Web-Service gehörende

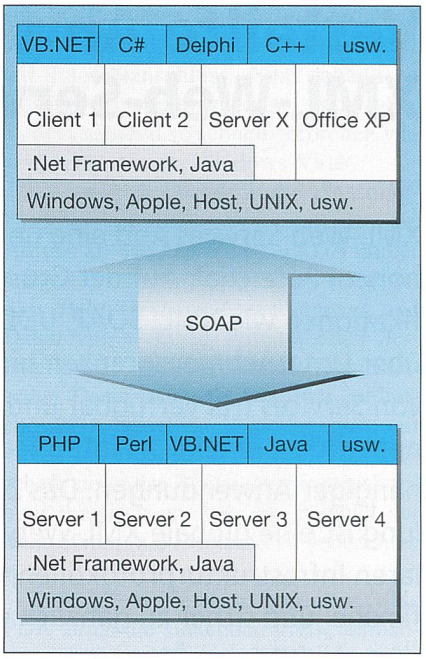

Bild 1 Plattformübergreifende Integration verschiedener Applikationen auf unterschiedlichen Plattformen mit SOAP

Remote Procedure Calls mit SOAP

SOAP stellt einen einfachen Mechanismus zur Verfügung, um auf verteilten Rechnern Methoden aufzurufen (Kapslung und Austausch von Remote Procedure Calls, RPC). Dazu wurde in den SOAP-Spezifikationen eine einheitliche Darstellung von RPC definiert mit dem Ziel, auf der Basis von Standards wie HTTP und XML ein einheitliches und erweiterbares Protokoll für Umgebungen zu schaffen, in denen herkömmliche RPC-Mechanismen nicht geeignet sind oder scheitern. Wird HTTP als Transportprotokoll genutzt, fügt sich ein RPC in das Request/Response-Schema ein. Der RPC wird als HTTP-Request übermittelt, das Ergebnis der Berechnung als HTTP-Response. Um eine entfernte Methode aufzurufen, werden Informationen wie der URI des Zielobjekts, der Methodenname sowie eventuell vorhandene Parameter der Methode veröffentlicht. Zusätzliche Informationen können im SOAP-Header transportiert werden. RPC über SOAP haben gegenüber herkömmlichen RPC-Lösungen mit Object Request Broker (ORB) Vor- und Nachteile. ORB können zwar mit den versendeten Daten mit höherer Performance umgehen, da die Datenstrukturen nicht serialisiert werden müssen. Anderseits muss die Kommunikationsgegenstelle ein äusserst komplexes Protokoll beherrschen, das den Einsatz teurer Middleware-Komponenten erfordert. Zudem werden Datenpakete von ORB von Firewalls leicht herausgefiltert, was den Einsatz von ORB in Business-to-Business-Internetumgebungen erschwert. SOAP- und XML-Web-Services sind damit die ideale Lösung, wenn sehr unterschiedliche System über das Internet miteinander kommunizieren wollen und die Geschwindigkeit der Datenübertragung keine vorrangige Rolle spielt. Zudem kann eine auf Standards basierende SOAP-Lösung wiederverwendet werden. Und nicht zuletzt erleichtert es die Kommunikation zwischen zwei Partnern, wenn man sich an ein öffentliches Protokoll halten kann und sich die beiden nicht gegenseitig von den Vorteilen der jeweiligen proprietären Lösungen überzeugen müssen.

WSDL-Dokument macht also alle nötigen Angaben für den Aufruf einer Funktion eines Webbdienstes.

Mit der «Universal Discovery Description and Integration[13)]» (UDDI) steht zudem eine Art «Gelbe Seiten» für XML-Web-Services zur Verfügung. UDDI – ursprünglich von Microsoft, IBM und Ariba gegründet – ist heute ein unabhängiges Konsortium für die Registrierung bzw. für die Suche von Web-Diensten. Die UDDI-Registratur listet die Dienste eines Unternehmens in einem Standardprofil auf, das es anderen Firmen erlaubt, entsprechende Services aufzufinden, Abkommen zu definieren und XML-Nachrichten auszutauschen, mit denen Geschäftstransaktionen ermöglicht werden. Mit UDDI verfügen Unternehmen über eine auch für kommerzielle Zwecke nutzbare Plattform für ihre XML-Web-Services.

```
<?xml version="1.0" encoding="UTF-8" ?>
<env:Envelope
xmlns:env="http://www.w3.org/2001/09/soap-envelope">
  <env:Header>
    <n:alertcontrol
xmlns:n="http://example.org/alertcontrol">
      <n:priority>1</n:priority>
      <n:expires>2001-06-22T14:00:00-05:00</n:expi-
res>
    </n:alertcontrol>
  </env:Header>
  <env:Body>
    <m:alert xmlns:m="http://example.org/alert">
      <m:msg>Pick up Mary at school at 2pm</m:msg>
    </m:alert>
  </env:Body>
</env:Envelope>
```

Bild 2 Beispiel einer SOAP-Nachricht

Globale XML-Architektur für Webdienste

Die bisherigen Funktionalitäten der XML-Web-Services sind erst der Anfang der Entwicklung. Marktkenner sind sich einig, dass den Komponenten für das Internet die Zukunft gehört. Daher arbeiten die Hersteller daran, die Standards der intelligenten Webdienste für komplexe Geschäftsbedürfnisse zu erweitern. Im Visier sind Themen wie Sicherheit, Routing, verlässliches Messaging und Transaktionsmanagement[14)].

Unter diesen Themen ist «Sicherheit» derzeit ein wichtiges Anliegen der Anwender. Sie verlangen eine nahtlose Sicherheitsarchitektur, die unkompliziert über Unternehmensgrenzen hinweg implementiert werden kann. Es gibt aber auch noch offene Fragen, was die Skalierbarkeit neuer Geschäftsmodelle und -systeme auf der Basis von XML-Web-Services betrifft. Entwickler benötigen für skalierbare Lösungen eine Möglichkeit, Nachrichtenpfade zu spezifizieren und dynamisch zu konfigurieren. Für geschäftskritische Anwendungen ist zudem die Verlässlichkeit des Nachrichtenaustauschs bedeutsam.

Mit der «Global XML Architecture» (GXA) von Microsoft wurden und werden die Grundlagen gelegt, um diese Anforderungen zu erfüllen. GXA ist ein Protokoll-Framework auf Infrastrukturebene, mit der ein konsistentes Modell zur Schaffung von verschiedenen Teilprotokollen für XML-Web-Services und Webanwendungen realisiert werden kann. Es definiert verschiedene Infrastrukturprotokolle – beispielsweise für Sicherheit und Verlässlichkeit der Kommunikation – die mit einer breiten Palette von Anwendungsprotokollen kombiniert werden können.

GXA nutzt die Ausbaufähigkeit von SOAP und erweitert dessen Nachrichtenverarbeitungsmodell. Seine Spezifikationen sind kurz und knapp, sie bieten eine fokussierte, konkrete Lösung für Bereiche wie Sicherheit, Routing, Attachements und Transaktionen. Derartige Protokoll-Bausteine können modular zusammengefügt werden. Dies erlaubt es, orthogonale GXA-Protokolle – wie beispielsweise Web-Service-Security (WS-Security) und Web-Service-Routing (WS-Routing) – gemeinsam in einer Nachricht zu kombinieren. Sie können aber auch einzeln eingesetzt werden. Zudem handelt es sich bei GXA-Protokollen um generische Lösungen, die, anders als bei herkömmlichen Umgebungen, für verschiedenste Einsatzgebiete geeignet sind – von Supply-Chain-Management bis hin zu Terminverwaltungsprogrammen. Sämtliche GXA-Protokolle basieren auf SOAP und können daher in Form des XML-Datenmodels spezifiziert werden, das in der Industrie breit unterstützt wird. Wie bei SOAP gibt es daher auch bei den weiterführenden GXA-Protokollen keine Einschränkungen hinsichtlich Plattformen, Entwicklungsumgebungen und Transportprotokoll.

GXA-Protokolle für Routing und Sicherheit

Diese Offenheit bedeutet auch, dass es sich bei allen bisherigen und künftigen Protokollen um offene Standards handelt, die in Zusammenarbeit mit Normierungsgremien wie dem World-Wide-Web-Konsortium (W3C) entstehen. GXA ermöglicht bereits heute ein virtuelles Netzwerk über beliebige Netzwerke und Protokolle hinweg und baut auf dem hierarchischen

XML haucht EDI neues Leben ein

Die Idee des Datenaustauschs zwischen Unternehmen ist nicht neu. Während über 20 Jahren war «Electronic Data Interchange» (EDI) das vorherrschende Format für den Informationsaustausch zwischen Unternehmen. Allerdings mit einigen Nachteilen hinsichtlich Flexibilität. Heute bieten neue XML-Formate ein viel offeneres, flexibles Austauschmedium für Geschäftstransaktionen. Das heisst nicht, dass damit EDI obsolet würde. Im Gegenteil: die technischen Möglichkeiten von XML bieten ein Framework, in dem die substanziellen Investitionen in Geschäftsprozesse auf der Basis von EDI weiterhin genutzt und auf breiter Basis zugänglich gemacht werden können. Die META Group[19)] geht davon aus, dass bis 2003 rund 70% des EDI-Datenverkehr internetbasiert stattfindet. Dies zeigt, dass XML und die darauf basierenden Web-Services nicht in erster Linie bestehende Umgebungen ablösen sollen. Sie ermöglichen es vielmehr, die verschiedenen Welten zu neuen Darstellungen von Geschäftsprozessen zusammenzuführen.

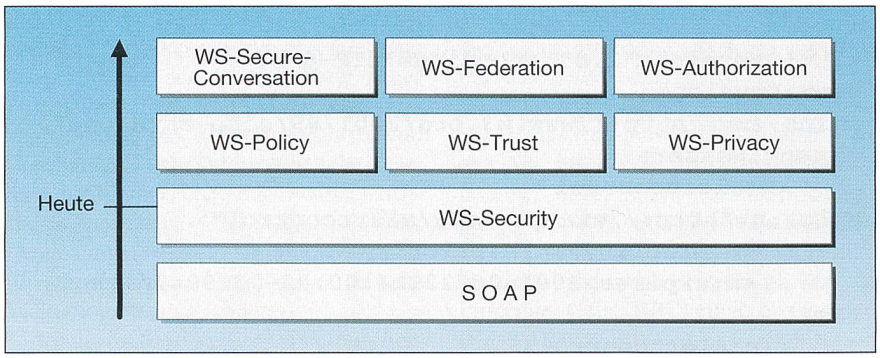

Bild 3 WS-Security
Mit WS-Security ist das Fundament für kommende Sicherheitsprotokolle im Rahmen der Global XML-Architektur gelegt

«Uniform Resource Identifiers»-Mechanismus (URI-Mechanismus) auf. Mit dem Protokoll WS-Routing wird die protokollspezifische Adressierung und das entsprechende Routing hinter einem einheitlichen, URI-basierten Adressierungsschema abgeschirmt. Mit WS-Routing können zwei Endpunkte miteinander kommunizieren, auch wenn kein gemeinsames Transportprotokoll verfügbar ist. Mit WS-Referral, einer einfachen SOAP-Erweiterung, kann zudem das Routing zwischen verschiedenen SOAP-Knoten dynamisch konfiguriert werden.

WS-Security ist das grundlegende Sicherheitsprotokoll von GXA. Es bietet auf der Grundlage von SOAP einen Mechanismus für die Signierung, Versiegelung und Verschlüsselung des Inhalts der SOAP-Nachricht (Bild 3). Dazu definiert WS-Security einen SOAP-Header, der den Security-Token des Senders übermittelt. WS-Security verlangt kein bestimmtes Format für den Sicherheitstoken. Vielmehr werden spezifische Formate durch ein Erweiterungselement im Security Header unterstützt. Damit können weit verbreitete Sicherheitsprotokolle wie X.509[15] oder Kerberos[16] eingesetzt werden. Die Signierung einer Nachricht stellt sicher, dass der Sender auch tatsächlich im Besitz des Sicherheitstokens ist und dass der signierte Nachrichtenteil nicht verändert wurde. Beim Versiegeln der Nachricht wird der Inhalt verschlüsselt und zwar mit einem Schlüssel, der ausschliesslich von Sender und Empfänger verstanden wird. WS-Security ist lediglich das Grundkonzept für Signierung und Verschlüsselung von Nachrichten. Der eigentliche Algorithmus hingegen wird von extern definierten Protokollen wie XML-Signature[17] und XML-Encryption[18] bezogen. Dies erlaubt es WS-Security, neue Algorithmen rasch und einfach einzubinden.

WS-Security ist damit eine Basisspezifikation mit der minimalen Funktionalität, um sichere Nachrichten zu senden. Microsoft und IBM haben im April 2002 eine Roadmap[18] vorgelegt, welche kommende Protokolle im Bereich Sicherheit umreisst. Dazu gehören:

- WS-Trust[18]: ein Modell, das die Basis für ein sicheres und vertrauensvolles Zusammenspiel von XML-Web-Services schafft;
- WS-Privacy[18]: ein Modell für Datenschutz-Angaben;
- WS-Authorization[18]: beschreibt Autorisierungsrichtlinien und wie entsprechende Daten verwaltet werden;
- WS-Federation[18]: beschreibt Verwaltungs- und Brokerfunktionalitäten für Trust-Beziehungen in einer heterogenen, aber föderativen Umgebung im Sinne von Geschäfts-Ökosystemen;
- WS-SecureConversation[18]: beschreibt die Verwaltung und Authentifizierung des Nachrichtenaustausches mit Berücksichtigung des Sicherheitskontextes;
- WS-Policy[18]: für die Beschreibung der Sicherheitsanforderungen wie benötigte Sicherheitstoken und unterstützte Verschlüsselungsalgorithmen.

Künftige GXA-Protokolle werden über die Sicherheitsaspekte hinaus die bestehenden XML-Web-Services-Funktionen gezielt erweitern. Mit der Realisierung und Standardisierung dieser Protokollfamilien im Rahmen von GXA haben XML-Web-Services das Potenzial, sich zur wichtigsten Applikationstechnologie der kommenden Jahre zu entwickeln.

Adresse des Autors

Urs Müller, Informatik-Ing. HTL, Microsoft Schweiz AG, Zürich, ursm@microsoft.com

[1] Bei Legacy-Anwendungen handelt es sich um bestehende, produktive Anwendungen, die meist auf Host-Systemen laufen, im praktischen Einsatz getestet und in oft jahrelanger Arbeit an die unternehmenseigenen Bedürfnisse angepasst wurden. Dem Vorteil der massgeschneiderten Verlässlichkeit steht der Nachteil gegenüber, dass sie – im Gegensatz zu modernen, mehrschichtigen Architekturen – als geschlossene Anwendung entwickelt wurden und insbesondere nicht auf die Kommunikation mit anderen Systemen ausgerichtet sind. Doch gerade der Informationsaustausch zwischen verschiedenen Applikationen gilt heute als erfolgskritische Voraussetzung für die Informatik eines Unternehmens. Daher müssen Legacy-Systeme im Rahmen einer gezielten Modernisierungsstrategie in neue, offene Lösungen integriert werden.
[2] XML: Extensible Markup Language
[3] W3C: World Wide Web Consortium. Dieses Forum wurde gegründet, um alle Möglichkeiten des Webs zu erschliessen. Dazu werden einheitliche Technologien (Spezifikationen, Richtlinien, Software und Tools) entwickelt, die den Fortschritt des Webs fördern und seine Interoperabilität sicherstellen. www.w3.org
[4] OASIS: Eine internationale Non-Profit-Organisation, die Industriestandard-Spezifikationen für auf XML basierende Interoperabilität entwickelt. www.oasis-open.org
[5] Beispielsweise für Lagerhaltung und Bestellwesen
[6] SOAP: Simple Object Access Protocol. www.w3.org/TR/SOAP
[7] HTTP: Hypertext Transport Protocol. www.w3.org/Protocols/
[8] www.isis-eai.de
[9] Loosely Coupled Systems
[10] SMTP: Simple Mail Transfer Protocol. Standard-Protokoll für den Versand von E-Mails im Internet.
[11] TCP; Transmission Control Protocol.
[12] www.jabber.org
[13] www.uddi.org
[14] http://msdn.microsoft.com/webservices/understanding/gxa/
[15] X.509: Public-Key-Infrastructure. www.ietf.org
[16] Kerberos: The Network Authentication Protocol. http://web.mit.edu/kerberos/www/
[17] www.w3.org/Signature/
[18] Microsoft und IBM: Security in a Web-Services World: A Proposed Architecture and Roadmap. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwssecur/html/securitywhitepaper.asp, April 2002.
[19] META Group ist eine auf Forschung und Consulting im Bereich von «Information Technology» und «Business Transformation Strategies» spezialisierte Firma. www.metagroup.com

Environnements d'application partagés avec les XML-Web-Services

Les XML-Web-Services sont une nouvelle technologie d'application offrant un potentiel important. Sur la base de spécifications standardisées comme XML, SOAP, UDDI et WSDL ils permettent de mettre à disposition, au-delà des limites d'entreprises, des processus commerciaux sous forme de services pour les rendre utilisables à volonté – Internet devient un lieu d'applications intégrées indépendantes des systèmes et plates-formes. L'objectif du développement actuel est la réalisation d'une architecture globale de services Web XML à protocoles d'infrastructure modulaires permettant l'intégration flexible et sûre d'applications sur Web en environnements entièrement nouveaux au sens d'«écosystèmes» économiques.