Standards rund um Webservices

Seminararbeit Sommersemester 2002
Betreuer: Thomas Bruse
Bearbeiter: Viktor Abrams

Table of contents

1 Webservices

1.1 Allgemein

1.1.1 Einleitung

Eine weltweite Vernetzung der Unternehmen bietet uns heute eine Vielzahl von Möglichkeiten, die Unternehmenskommunikation neu zu gestalten. Sie veranlaßt immer mehr Unternehmen dazu ihre Services im Internet bereitzustellen. Die zunehmende Standardisierung dieser Kommunikation erlaubt mittlerweile auch die Serviceinanspruchnahme über Plattformgrenzen hinweg.

In diesem Zusammenhang fällt oft der Begriff "Webservice". Innerhalb dieser Arbeit soll aufgezeigt werden, welche Standards heute schon existieren und welche Aufgabenbereiche sie innerhalb einer Buisnesstransaktion abdecken.

1.1.2 Technische Definition von Webservices

Bei Webservices handelt es sich, allgemein ausgedrückt, um URL-adressierbare Software.

Ein Webservice ist eine sich selbst beschreibende, modulare Anwendung, die sich durch andere Applikationen über das Web durch Standard Webprotokolle aufrufen läßt (z.B. HTTP(S), SMTP). Webservices können zu anderen Webservices zusammengefügt werden, so daß man durch die Nutzung verschiedener Webservices zu komlexeren Anwendungen kommt. In diesem Fall spricht man von einer Composition.

1.2 Kernelemente

1.2.1 SOAP - Simple Object Access Protocoll

SOAP wird zur Kommunikation genutzt und entspricht einem Remote Procedure Call (RPC-Aufruf)(Abb.1). Es basiert auf XML und ist somit plattform- und sprachunabhängig. Die Kommunikation findet dabei über unterschiedliche Standardprotokolle wie z.B. http, https oder ftp statt und kann dabei auch über Firewalls hinweg genutzt werden.

Eine SOAP-Nachricht besteht aus dem Umschlag(Envelope), welcher die Namensräume definiert und den Kopf(Header) der Nachricht. Innerhalb des Headers können optionale Informationen über Sicherheit und Authentifizierung angegeben werden. Dem folgt der Nachrichtenkörper(Body) der den RPC-Aufruf und das Nachrichtendokument enthält.

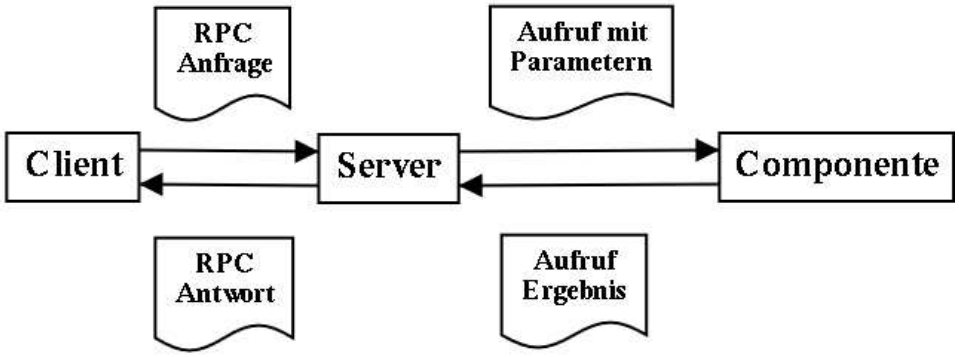

Abbildung 1

1.2.2 WSDL - Web Services Definitions Language

Die WSDL beschreibt den Service der geleistet wird.

WSDL ist eine Spezifikation zur Beschreibung XML-basierter Dienste. Dadurch gibt es den Serviceprovidern eine einfache Möglichkeit ihre Dienste, mit

Methoden und Parametern und unabhängig vom Protokoll, zu beschreiben.

1.2.3 UDDI - Univeral Description Discovery and Integration

Mit Hilfe von UDDI können Services publiziert bzw. verlegt werden (advertise/syndicate). UDDI dient dabei als Informationsdienst, der es ermöglicht durch gezielte Suchen neue Geschäftspartner zu finden, die bestimmte Services bereitstellen. Mit diesen Informationen kann dann der benötigte Service bei den so genannten Locations angefragt werden.

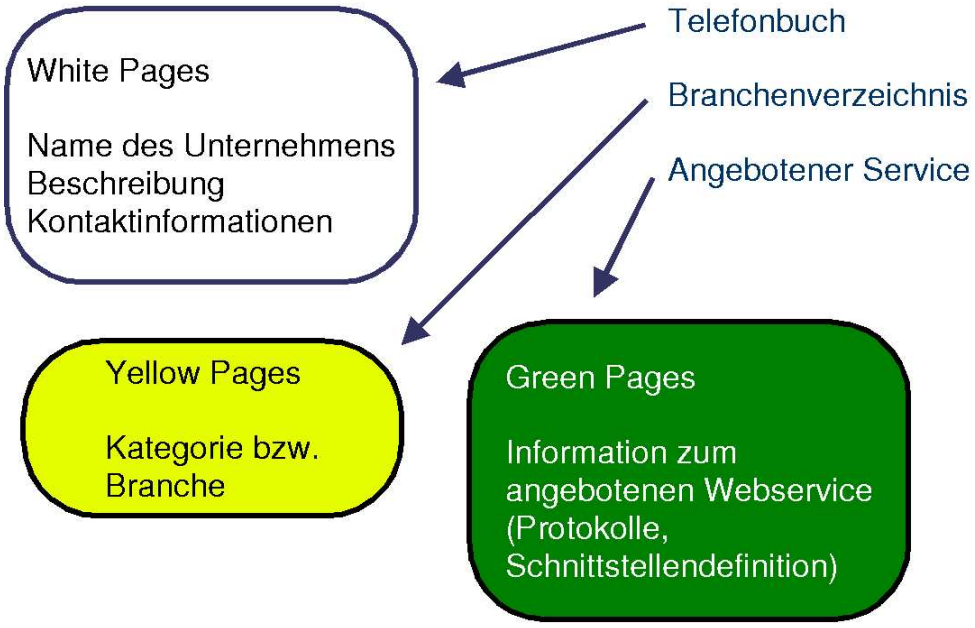

Abbildung 2

Die UDDI Registry ist dabei im wesentlichen in drei Bereiche (Abb.2) aufgeteilt: den White Pages, Yellow Pages und Green Pages. Die White Pages enthalten Angaben, wie den Namen des Unternehmens, eine Beschreibung, sowie Kontaktinformationen und spiegeln in ihrer Funktionalität ein Telefonbuch wieder. Die Yellow Pages, die dem Branchenverzeichnis ähneln - enthalten Angaben darüber, in welche Kategorie das Unternehmen einzuordnen ist (Branchen). In den Green Pages finden sich die Informationen zu den Web Services, d.h. Informationen zu den benutzten Protokollen und Schnittstellendefinitionen, die für Businesstransaktionen über die Webservicetechnolgie benötigt werden. Auf diese Weise stehen den potentiellen Geschäftspartnern oder Kunden sämtliche Informationen standardisiert zur Verfügung.

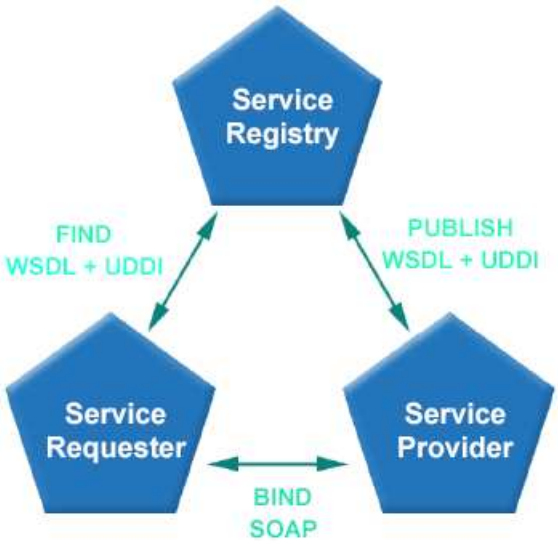

Abbildung 3

1.3 Standardisierungs Konsortien

Die Standards SOAP (http://www.w3.org/TR/SOAP/) und WSDL (http://www.w3.org/TR/wsdl.html) werden vom World Wide Web Consortioum (W3C) definiert. Bei UDDI (http://www.uddi.org/) handelt es sich um eine Cooperation verschiedener Firmen, unter anderem IBM, SUN, Microsoft, Oracle, SAP, Hewlett Packard und Intel, die sich auf einen gemeinsamen Standard geeinigt haben.

2 Technik und Praxis

2.1 Allgemeine Problemstellung

Webservices sollen es ermöglichen, auf angebotene Dienste unterschiedlicher Anbieter zurückgreifen zu können. Die verschiedenen Plattformen mit ihren unterschiedlichen Betriebssystemen verlangen bisher individuelle Anpassungen, um eine Zusammenarbeit der Systeme zu erreichen. So kann ein unmittelbarer Wettbewerb der Dienstanbieter nicht stattfinden, da ein Auftraggeber, allein durch die firmenintern eingesetzten Systeme, in der Wahl der nutzbaren Dienste eingeschränkt wird. Desweiteren verursacht ein Wechsel zu einem anderen Dienstanbieter oft Investitionskosten, um den fehlerfreien Datenaustausch zu gewährleisten. Dadurch das Webservices keine neuen Techniken einführen, sondern sich bewährte Verfahren zu nutze machen, kann auf das Wissen der bereits

aktiven Mitarbeiter zurückgegriffen werden. Dabei stützt sich die Definition eines Services auf das Dokumentenformat XML, das durch die einfache Schreibweise und Struktur mit geringem Aufwand erschließbar ist. Trotzdem stellt die strenge Definition des XML-Standards die Korrektheit eines Dokumentes sicher. Die Form der Übertragung wird dabei nicht festgelegt. Da es sich bei einem Webservice um ein einfaches Textdokument handelt, kann es über die vorhandenen Verfahren übertragen werden. So werden in der Praxis oft Übertragungsprotokolle wie HTTP(S),FTP oder auch SMTP eingesetzt.

2.2 Funktionsweise von Webservices

Am einfachsten erschließt sich die Funktionsweise von Webservices, indem man eine Analogie (Abb.4) zu der seit langem etablierten Vorgehensweise von Corba zieht. UDDI deckt dabei die Aufgabenbereiche eines Naming Services ab. So werden über einen UDDI-Zugriff die zur Verfügung stehenden Anwendungen und die Resourcen, die diesen Service bereitstellen, bekannt gemacht. Im Gegensatz dazu, definiert WSDL die Schnittstelle, die durch außenstehende Applikationen genutzt werden kann. Innerhalb einer Corba-Umgebung spezifiziert die IDL, die Art und Form des Aufrufes einer Applikation. Analog dazu wird innerhalb von WSDL eine Spezifikation der eingehenden und zurückkommenden Informationen angegeben.

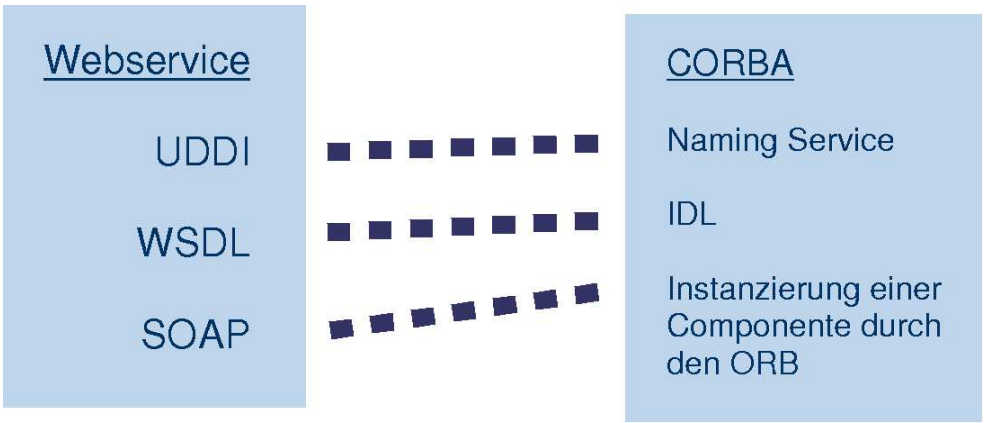

Abbildung 4

Über die SOAP-Schnittstelle wird der eigentliche Aufruf eines Services initiiert. So werden die innerhalb von WSDL aufgeführten Daten in das SOAP-Dokument eingebettet und verschickt. Dieses Verschicken, eines so aufbereiteten Dokumentes an einen Dienstgeber, führt dann zum eigentlichen Aufruf des Services. Innerhalb von Corba wird dieser Aufruf durch die Instanzierung einer Componente durch den ORB realisiert.

2.3 Die Techniken hinter den Services

Services im Internet anzubieten gehört schon lange zu einer sinnvollen

Unternehmensstrategie. Anfänglich beschränkte man sich dabei auf statische Inhalte die mit Hilfe eines Webservers im Internet angeboten wurden. Dabei ging es in erster Linie nicht darum automatisierte Dienste bereitzustellen, sondern vor allem eine Werbe- und Informationsfläche zur Verfügung zu stellen, die rund um die Uhr für jeden potentiellen Kunden zugreiffbar war. Dieses Informationsangebot deckte allerdings nicht die neu geschaffenen Anwendungsmöglichkeiten ab, die sich durch das Internet ergaben.

Duane K. Fields und Mark A. Kolb schreiben dazu [JSP]:

"Die ersten HTTP-Server besaßen keinen eingebauten Mechanismus, mit dem sie Inhalte dynamisch erzeugen konnten. Stattdessen erhielten sie Schnittstellen, mit denen sie andere Programme aufrufen konnten, welche zur Laufzeit Inhalte erzeugten. Der erste Standard für dynamische Webinhalte basierte auf dem Common Gateway Interface (CGI). Dabei handelt es sich um einen Mechanismus, mit dessen Hilfe Webserver empfangene Browseranfragen an fremde Programme weiterleiten konnten, die ausgeführt wurden und zur Laufzeit Antworten erzeugt und an den Webserver zurückgeliefert haben." (Abb.5)

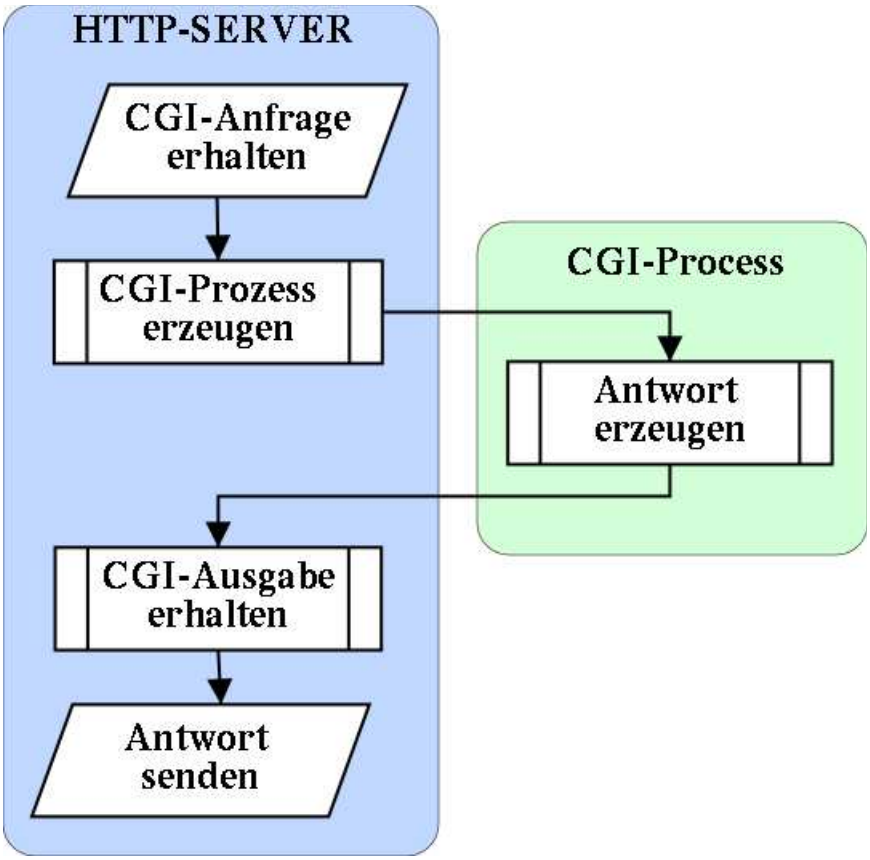

Abbildung 5

"Jede Sprache, in der sich eigenständig auf dem HTTP-Server lauffähige Programme schreiben lassen, eignet sich für CGI. Sie können CGI-Programme beispielsweise in jeder Scriptsprache schreiben, die das lokale Betriebssystem unterstützt."

Aus diesem Sachverhalt heraus entwickelten sich viele unterschiche Sprachen, die zum Einsatz kamen, wenn es darum ging dynamische Inhalte zu erzeugen.

Vergleicht man die Vorgehensweise, die sich hinter dem CGI verbirgt, so zeigen sich Parallelen zum Standard SOAP. Wie auch CGI, definiert SOAP ein Dokument in dem sich Informationen zu den angefragten Service befinden. Aus diesen Informationen heraus führt der Webserver eine bestimmte Anwendung aus, deren Ergebnis er dann an den Endanwender zurückschickt. Dazu nutzt SOAP auch das Kommunikationsprotokoll HTTP, kann allerdings auch mit jedem anderen Protokoll gekoppelt werden, das in der Lage ist BASE64-Encoded Dokumente zu versenden.

So kann man SOAP als Weiterentwicklung des CGIs betrachten, die sich nicht darauf beschränkt einfache Webinhalte zu publizieren, sondern jede Form von Daten über eine Standardisierte Schnittstelle zu empfangen oder zu verschicken. Demnach ist es, wie auch bei CGI, möglich jede beliebige Anwendung hinter der SOAP-Schnittstelle laufen zu lassen, sofern das Programm in der Lage ist XML-Dokumente im Allgemeinen und die SOAP-Spezifikation im Besonderen auszuwerten und zu beantworten.

2.4 Die Produkte

Das .NET Framework von Mircrosoft (www.microsoft.com) ist Bestandteil

der .NET Strategie und stellt eine Umgebung für das Entwickeln, Verteilen und Ausführen von webbasierten Diensten und anderen Applikationen zur Verfügung. Es besteht aus Klassen, Common Language Runtime (CLR) sowie den Diensten (ASP.NET) und soll ein vereinheitlichtes Programmiermodell, mit der Möglichkeit zur Ansteuerung aller bisher vorhandenen Modelle, bieten. Mit ASP.NET soll es möglich sein, z.B. aus dem Browser heraus, auf unterschiedliche Dienste zurückgreifen zu können. Es dient unter anderem dazu, von den bisher statischen Webseiten der Anbieter wegzukommen und, durch die Ausführung von Programmen innerhalb eines Browsers, zu mehr Interaktion mit dem Benutzer zu gelangen. So hat Microsoft z.B. den sogenannten .NET Framework Data Provider für Datenbanken veröffentlicht, der einen direkten und schnelleren Zugriff auf Datenbanken ermöglicht, ohne den Umweg über so genannte Managed OLE DB Provider gehen zu müssen. Implementiert werden sollen die eigenen Programme, in der von Microsoft neu geschaffenen Programmiersprache C#. Nachdem Microsoft Ende Januar 2001 im Rahmen eines Rechtsstreits gegenüber Sun einwilligte, die Java-Technologie nicht in abgeänderter Form an die Endkunden weiterzugeben bzw. die eigene Entwicklung nicht mit dem Namen Java zu belegen, entwickelte Microsoft die Programmiersprache C#, die sich im wesentlichen sowohl von der Syntax als auch von der Semantik an SUNs Java anlehnt. Desweiteren wurde eine Reihe von Basisklassen mit den zugehörigen Methoden entworfen, die grundlegende Systemfunktionalitäten und Programmierwerkzeuge bereitstellen.

Dieser Ansatz war vom seiner Funktionsweise, zu diesem Zeitpunkt schon bekannt. SUN (www.sun.com) hat 1995 mit der Veröffentlichung der Programmiersprache Java ein vergleichbares Konzept vorgelegt. So war die Idee, nicht mehr plattformspezifische ausführbare Programme zu generieren, sondern eine Art virtuelle Maschine zu schaffen, für die man ausführbaren Programmcode erzeugt. Dieser wiederum sollte dann durch einen Interpreter bzw. durch eine virtuelle Maschine, auf dem jeweiligen Zielrechner zur Laufzeit ausgewertet werden.

Eine Weiterführung dieses Ansatzes stellt die SUN-ONE-Strategie dar, die unter anderem mit der iPlanet Plattform, eine komplette Software-Umgebung zum Erstellen, Assemblieren und Bereitstellen offener, intelligenter Webservices mitbringt. Anders als Microsoft, nutzt SUN dafür natürlich die eigens entwickelte Java 2 Enterprise Edition (J2EE).

So kommt es den auch dazu, das diese beiden konkurierenden Modelle am offenen Markt nach Unterstützung suchen. Mit dem Web Application Server, der Bestandteil IBMs Websphere ist, bietet IBM (www.ibm.com) die Möglichkeit SOAP-Servlets bereitzustellt und den Zugriff auf die UDDI-Registrierung zu realisiert. Dabei kann sowohl auf das ASP.NET Framework als auch auf die J2EE zurückgegriffen werden, um diese zu entwickeln. Einen ebenso flexiblen Ansatz bietet Plumtree (www.plumtree.com) mit dem Gadget Web Services, mit dem auch eine Nutzung beider Technologien ermöglich wird. SAP (www.sap.com) mit dem Web Application Server bietet dagegen nur eine Unterstützung für die J2EE. Dem schließt sich Oracle (www.oracle.com) mit dem Application Server und HP

(www.hp.com) mit ihrer Web Services Platform an.

2.5 Alternativen

Wo anfänglich nur die J2EE von SUN und die .NET Strategie von Microsoft im Blickpunkt der Betrachtung standen, bringen heute viele Scriptsprachen eine Unterstützung für SOAP mit. So werden durch den Einsatz von Webservices nicht mehr unzählige neue administrative Arbeiten geschaffen, die der Betrieb eines Portalservers mit sich bringt, sondern es kann auf bereits bestehende Server wie z.B. einem Webserver zurückgegriffen werden, der dann einfache Services bereitstellt. Desweiteren ermöglichen Scriptsprachen eine einfache Implementierung eines Services. Programmiersprachen wie Java oder auch C# verlangen dem Programmierern eine exakte Typendeklaration ab und setzen so ein detailiertes Wissen über die Verwendbarkeit und Definitionsräume der verwendeten Datentypen voraus. Dadurch kann für kleine und mittelständische Unternehmen eine Hürde bei der Nutzung und Bereitstellung von Webservices entstehen, da Programmierer aufgrund höherer Qualifizierung einen weiteren Kostenfaktor darstellen. In Scriptsprachen wie Perl, Python oder auch PHP, die eine SOAP-Schnittstelle mitbringen, existiert zwar auch eine Unterscheidung von Datentypen, diese jedoch beschränkt sich auf die Definition eines Zahlen- bzw. Texttypes. Desweiteren kann eine Variable ohne vorherige Zuweisung zu einem bestimmten Datentyp benutzt werden, wobei der Datentyp jederzeit durch eine erneute Zuweisung geändert werden kann. Listen und Wörterbücher, die in den genannten Scriptsprachen implementiert sind, können über eine einfache Syntax erstellt werden. Schlüssel und Inhalte können dann einfach zugewiesen werden. Zwar implementieren Programmiersprachen ähnliche Mechanismen, diese setzen aber die Kenntnis der eigens dafür entworfenen Klassen voraus.

Der Webserver Apache von der Apache Software Foundation (www.apache.org) ist über ein Modul in der Lage SOAP basierte Web Services durchzuführen. Das von der Firma Convanet (http://www.covalent.net) für den Webserver Apache entwickelte Modul mod_asp.net ermöglicht desweitern den Zugriff auf die interne .NET-Infrastruktur, ist dadurch aber auf Windows angewiesen und so auch nur für die Windows-Version des Apache erhältlich.

3 Sicherheit

3.1 Allgemein

Neben der Standardisierung der Kommunikation mit den Dienstanbietern, stehen vorallem Sicherheitsaspekte während der gesamten Geschäftstransaktion im Vordergrund. Produkte die sich derzeit auf dem Markt befinden nutzen zumeist

eigene Sicherheitsmechnismen, die zu anderen Produkten inkompatibel sind. So wird in vielen Fällen auf einen Kerberosserver zur Autentifikation und Authorisation Bezug genommen, um einen unauthorisierten Zugriff zu vermeiden. Das setzt in der Praxis aber eine Registrierung bei den jeweiligen Dienstanbieter voraus und erschwert die Interoperabilität der Serviceanbieter. Der kleinste gemeinsame Nenner ist in vielen Fällen der Verzicht auf eine, neben der eigendlichen Verschlüsselung der Kommunikation über SSL (Secure Socket Layer), zusätzliche Absicherung der Geschäftsprozesse. Im Folgenden soll die Notwendigkeit zusätzlicher Sicherheitsmechanismen aufgezeigt werden.

Um Webservices innerhalb von Geschäftsprozessen nutzen zu können, müssen sie nach [MColan] auf jeden Fall folgende Sicherheitsfunktionen anbieten.

- Identifikation:
  Jeder Anbieter und jeder Nutzer muss die Identität des jeweils anderen kennen.
- Authentifizierung:
  Identitäten müssen zweifelsfrei nachweisbar sein.
- Autorisierung:
  Bestimmte Webservices sollen nur von bestimmten Anwendern genutzt werden können.
- Datenintigrität:
  Die gesendeten bzw. empfangenen Daten dürfen auf dem Weg vom Sender zum Empfänger nicht unberechtigt verändert werden können.
- Vertraulichkeit:
  Es muss sichergestellt werden können, dass niemand ausgetauschte Informationen mitlesen kann, z.B. durch den Einsatz von Verschlüsselungstechniken.
- Protokollierung:
  Ausgetauschte Informationen müssen bei Bedarf mitprotokolliert werden können, damit auch im Nachhinein die Möglichkeit besteht, erfolgte Kommunikation nachvollziehen zu können.
- Nachweisbarkeit (Non-Repudiation):
  Sowohl der Sender als auch der Empfänger einer Nachricht müssen zweifelsfrei nachweisen können, das erstens der Sender die Nachricht auch wirklich verschickt hat, und das zweitens der Empfänger die gleiche Nachricht auch wirklich erhalten hat.

Mit Hilfe von HTTPS ist es möglich den Nachrichtentransport des SOAP-Dokumentes zwischen Sender und Empfänger abzusichern, jedoch wird dadurch die Nachweisbarkeit dieser Kommunikation nicht gewährleistet, so daß allein über HTTPS die oben genannten Sicherheitsfunktionen nicht abgedeckt werden können.

Nach [SaHi] sprechen daher die folgenden Punkte dafür, die Sicherheitsvorkehrungen nicht auf Transportebene, sondern auf Nachrichtenebene zu treffen:

- Ende-zu-Ende Sicherheit:
  SOAP-Nachrichten können mit Hilfe unterschiedlicher Protokolle übertragen werden. Da auf dem Weg vom Sender zum Empfänger mehrere Zwischenstationen liegen können, ist es nicht mehr möglich die Ende-zu-Ende Sicherheit zu gewähleisten, sobald auch nur eine der Zwischenstationen nicht vertrauenswürdig ist.
- Unabängigkeit von der Anwendung:
  Ein standardisierter, anwendungsunabhängiger Security-Layer kann Sicherheit zur Verfügung stellen, ohne dass der Anwendungsentwickler selbst Verschlüsselungsalgorithmen implementieren muss.
- Unabhängigkeit vom Transportprotokoll:
  Bei der Versendung einer SOAP-Nachricht über mehrere Zwischenstationen müsste die jeweilige Station die Sicherheitsinformationen gegebenfalls in andere Protokolle umsetzen.
- Sicherheit auch für persistierte Nachrichten:
  Sicherheit auf Transportebene schützt ein Dokument zwar während des Versendens, kann aber keinen Schutz bieten, wenn das Dokument kurzzeitig auf einer der Zwischenstationen gespeichert werden muß.

Aufgrund dieser Aspekte ist es notwendig die Kommunikation mit zusätzlichen Mechanismen abzusichern.

3.2 Sicherheitsmechanismen

Im Folgenden soll anhand der bereits ausgearbeiteten Sicherheitsstandards aufgezeigt werden, mit welchen Methoden es möglich ist, die oben erwähnten Sicherheitsaspekte, beim Einsatz von Webservices abzudecken (Abb.6) und so die Geschäftsprozesse zu sichern.

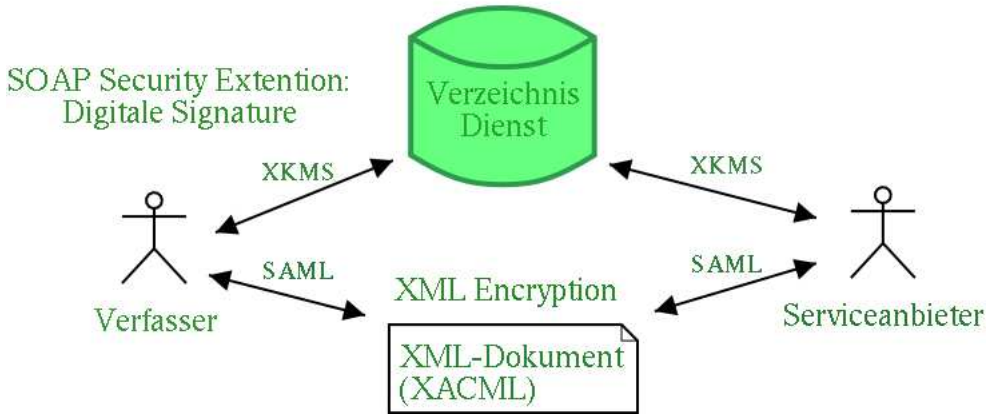

Abbildung 6

3.2.1 SOAP Security Extensions: Digitale Signature

IBM und Microsoft haben schon im Januar 2001 gemeinsam beim W3C die "SOAP Security Extensions: Digitale Signatur" als Erweiterung zum SOAP-Standards eingereicht, die die Möglichkeit bietet, SOAP-Nachrichten digital zu signieren.

Mit Hilfe dieser Signaturen kann sowohl der Verfasser als auch die Integrität einer Nachricht gewährleistet werden. Es ist allerdings nicht möglich den Sender zu verifizieren. So kann ein und dieselbe Nachricht durch eine dritte Stelle erneut gesendet werden und für Verunsicherung innerhalb eines Geschäftsprozesses sorgen. Es ist unumgänglich, neben diesem Mechanismus noch andere Sicherheitsmassnahmen zu treffen.

Zur Signatur einer SOAP-Nachricht soll demnach die W3C/IETF XML Signatur Specification (http://www.w3c.org/Signature/) genutzt werden. Dabei wird über ein RSA-Zwei-Schlüssel-Verfahren die Nachricht mit dem privaten Key des Senders verschlüsselt. Um sich vor dem wiederholten senden einer Nachricht zu schützen wird zusätzlich entweder eine Sequenznummer oder ein Zeitstempel bzw. ein Verfallsdatum innerhalb der verschlüsselten Nachricht vergeben.

Voraussetzung dabei ist natürlich, dass dem Empfänger der öffentliche Schlüssel des anderen zur Verfügung steht.

3.2.2 XML Encryption

Neben der Integrität einer Nachricht muss natürlich auch die Vertraulichkeit der gesendeten Daten gesichert werden. Die W3C XML Encryption Working Group (http://www.w3c.org/Encryption/2001/) befasst sich zu diesem Zweck mit einem Standard, der die Verschlüsselung XML-basierter Dokumente ermöglicht. Der Standard schreibt allerdings nur die Implementierung des Advanced Encyption Standard (AES) und des Triple Data Encryption Standards (DES) für XML Encyption konforme Systeme vor und lässt dabei offen, welche Art von Verfahren und welcher Algorithmus zum Einsatz kommt. Das setzt demzufolge eine genaue Absprache zwischen Sender und Empfänger einer solchen Nachricht, über das eingesetzte Verschlüsselungsverfahren, vorraus.

3.2.3 XML Key Managment Spezification (XKMS)

Ein grosser Teil der eingesetzten Verfahren zur Verschlüsselung und zur Signatur von Nachrichten basieren auf einem asymetrischen Verschlüsselungsverfahren. Dabei werden zwei Schlüssel verwendet, wobei der erste zur Verschlüsselung und der zweite zur Entschlüsselung benutzt wird. In all diesen Verfahren wird vorrausgesetzt, dass der jeweils öffentliche Schlüssel des Geschäftspartners zur Verfügung steht. Um dieses zu gewährleisten, kann dieser Schlüssel entweder direkt an den Partner versand werden oder man nutzt das Vorhandensein von öffentlichen, vertrauenswürdigen Verzeichnissen, die diesen Schlüssel über ein sicheres Verfahren publizieren.

Die XML Key Managment Spezification (XKMS) setzt sich mit der Standardisierung des Zugriffs auf diese Art von öffentlichen Verzeichnisdiensten auseinander. So kann über diesen Standard, sowohl die Registrierung, als auch die nachfolgende Verteilung der Schlüssel erfolgen. XMKS spezifiziert dabei Operationen zum Zugriff auf ein Key-Managment-System, das selbst als Webservice implementiert ist [XKMS].

3.2.4 Security Assertion Markup Language (SAML)

Mit Security Assertion Markup Language [SAML] wird von OASIS daran gearbeitet, die Zugriffsteuerung über eine standardisierte Schnittstelle zu ermöglichen. Bisher wird dieses Feld von den jeweiligen Schnittstellen der einzelnen Anbieter besetzt, wodurch es nicht selten zu Inkompatibilitäten kommt.

SAML stellt dabei keine eigenen Funktionalitäten zur Autorisierung bzw. Authentifizierung zur Verfügung, sondern sichert lediglich zu, das diese bereits an einer anderen Stelle erfolgt ist.

3.2.5 Extensible Access Control Markup Language (XACML)

Die Extensible Access Control Markup Language ist eine XML-basierte Sprache, mit der Zugriffsrechte auf XML-basierte Nachrichten festgelegt werden können. Sie wird ebenfalls von der OASIS-Arbeitsgruppe ausgearbeitet. Mit Hilfe von XACML lassen sich elementweise Zugriffstbeschränkungen für XML-Dokumente ausdrücken. So kann z.B. festgelegt werden, welche Teilelemente eines XML-Dokumentes für wen zugreifbar sind.

3.2.6 Fazit Sicherheit

Mit Hilfe der, von der Oasis-Arbeitsgruppe ausgearbeiteten, Standards wäre man in der Lage Geschäftsprozesse nach [MColan] ausreichend abzusichern. Allerdings fehlt es noch an Unterstützung durch die Produkte der großen Dienstanbieter, da eine Einigung auf gemeinsame Standards noch aussteht.

4 Geschäftsprozesse und Workflows

4.1 Koppelung des firmeninternen Workflows an externe angebotene Services

4.1.1 Übersicht

Die Bereitstellung von Web Services, an externe Geschäftspartner, beläuft sich nicht immer nur auf die Ausführung von statischen Applikationen. In vielen Fällen wird ein komplexer Workflow innerhalb des Unternehmens ausgelöst. Eine Transformation des Web Service, zu einem intern auszuführenden Workflow, kann sich als schwierig und zeitaufwändig herausstellen.

Um eine unmittelbare Weiterbearbeitung, innerhalb des Unternehmens, zu gewährleisten, entwickelte IBM den im folgenden näher erläuterten Standard.

4.1.2 Web Services Flow Language (WSFL)

Bei WSFL handelt es sich um eine datenflußorientierte Composition mehrerer Web Services. So bietet dieser Standard, über den Entwurf eines Workflows, eine Hintereinanderschaltung mehrerer Services. WSFL ist eine XML-basierte Sprache, die das Darstellen von Geschäftsprozessen als Composition von Webservices ermöglicht. So kann durch die Koppelung des firmeninternen Workflows, mit den extern angebotenen Services, eine unmittelbare Bearbeitung der eingehenden Aufträge erfolgen (Abb.7).

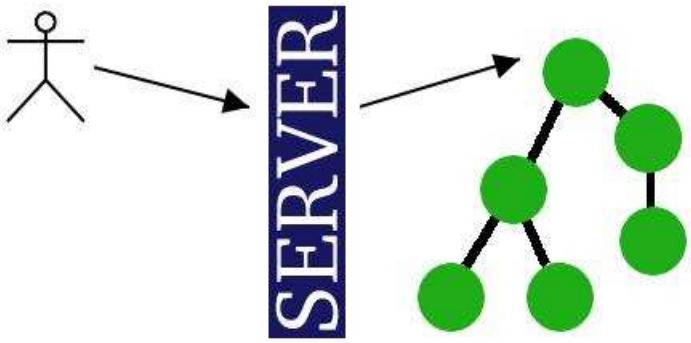

Abbildung 7

In [WSFL] schreibt Prof. Dr. Frank Leymann dazu, dass WSFL zwei Typen von Beschreibungen unterstützt:

- Der Erste bildet die zu einer Composition von Web Services zugehörigen und zu benutzenden Services ab. Als Ergebnis erhält man eine Abbildung des jeweiligen Geschäftsprozesses (Flow Model). Das resultierende sogenannte Flow Model beschreibt den aufeinanderfolgenden Aufruf von Web Services und die weiterzureichenden Resultate an darauffolgende Web Services. Dieses Flow Model stellt wiederum, im Sinne von WSDL, einen Webservice dar und kann als solcher aufgerufen werden.
- Der Zweite ermöglich die Abbildung des Interaktionsprozesses, innerhalb einer Composition von Web Services. In diesem Fall erhält man als Ergebnis eine allumfassende Beschreibung der zwischenpartnerlichen Interaktionen (Global Model). Hier wird nicht die Aufrufreihenfolge festgelegt sondern die Zusammenhänge und die Zusammenarbeit beschrieben. So gibt uns das Global Model Auskunft darüber, wo die innerhalb des Flow Models in Anspruch genommenen Services bereitgestellt werden und wie diese aufgerufen werden.

IBM bietet mit dem Produkt Websphere MQ Workflow (MQSeries Workflow) [MQWF] eine Schnittstelle an, mit der das Erstellen und Verändern firmeninterner Workflows einfach erledigt werden kann. Desweiteren können diese dann als Web Services angeboten werden und stoßen beim Aufruf den spezifizierten Workflow an.

4.2 Kommunikation mit externen Partnern

4.2.1 Überblick

Neben WSFL existieren noch weiter Standards, die allerdings in erster Linie die Kommunikation mit externen Geschäftspartnern sichern sollen. Während WSFL die Koppelung des internen Workflows mit den extern angebotenen Services berücksichtigt und die gebildeten Kompositionen wiederum als Web Service bereitstellt, beschreiben die folgenden Standards nur den Datenaustausch von Geschäftsdokumenten mit externen Geschäftspartnern.

4.2.2 ebXML BPSS (electronic business XML Business Process Specification Schema)

Mit Hilfe von ebXML BPSS kann die Kommunikation mit externen Geschäftspartnern umschrieben werden. Dabei steht die Übertragung von Geschäftsdokumenten an externe Geschäftspartner im Vordergrund. Unabhängig vom jeweils genutzten Kommunikationsprotokoll können, in einer ebXML Message, für den geleisteten Service, benötigte Daten ausgetauscht werden.

UN/CEFACT der United Nations und OASIS beschlossen im Mai 2001 gemeinsam die ebXML 1.0 Spezifikation zum Austausch von Geschäftsdaten.

In erster Linie zielt es darauf ab, XML-basierte Geschäftsdokumente zu übertragen, kann aber auch für andere Formate genutzt werden. ebXML spezifiziert einen Envelope-Typ für die sichere und zuverläßige Übertragung von Geschäftsinformation. Es basiert auf einer Erweiterung von SOAP bzw. von SOAP Messages with Attachments.

Abbildung 8

Eine ebXML Message (Abb. 8) ist dabei in zwei logische Abschnitte aufgeteilt. Im ersten Abschnitt befindet sich das SOAP 1.1 konformes Dokument, das den zu leistenden Service umschreibt und gegebenenfalls Bezug auf die mitgesendeten Dokumente nimmt. Dieser Bezug wird über ebXML spezifische Angaben, innerhalb der SOAP Nachricht, realisiert. Im zweiten Abschnitt können die zusätzliche Dokumente angegeben werden, die für den zu leistenden Dienst benötigt werden. Diesen zweiten Bereich bezeichnet man als Payload Container.

4.2.3 XLANG

Bei XLANG handelt es sich um einen Vorschlag von Microsoft [XLANG], öffentliche Prozesse anzubieten. Um die Services eines Dienstanbieters zu umschreiben wird auf WSDL zurückgegriffen. Mit einem Exception Handling kann innerhalb eines XLANG Dokumentes genau festgelegt werden, was im Fehlerfall zu tun ist bzw. welche Aktion bei einer erfolgreichen Ausführung eines Service angestoßen werden soll. Desweitern ist es möglich Transaktionen festzulegen und diese auch als atomar anzugeben. So sollte dann sichergestellt werden, dass ein spezifizierter Service entweder vollständig und fehlerfrei oder im Fehlerfall überhaupt nicht ausgeführt wird bzw. gemachte Veränderungen wieder in den vorherigen Zustand bringt.

5 Zusammenfassung und Ausblick

Die Analyse der existierenden Standards hat gezeigt, dass es noch einer Einigung, bei der Standardisierung der Sicherheitsmechanismen, bedarf. So bieten die auf dem Markt existierenden Produkte zwar jeweils ihre Mechanismen an, diese können aber bei einer automatisierten Abwicklung der Geschäftsprozesse über Produktgrenzen hinweg, hinderlich sein. Zusätzlich bedarf es einer gesonderten Registrierung bei den jeweiligen Dienstanbietern.

So führen diese Barrieren entweder dazu, das ein Unternehmen auf die Nutzung webbasierter Services vorerst verzichtet oder was weitaus schwerwiegender ist, allein auf eine Verschlüsselung der Kommunikation, bei der Nutzung von Web Services über Produktgrenzen hinweg, setzt.

Eine Anbindung der zu leistenden Services an den firmeninternen Workflow wird zukünftig immer mehr an Gewicht gewinnen. Mit WSFL hat IBM einen Standard ausgearbeitet, mit dem diese Anbindung vollzogen werden kann. Für kleine und mittlere Unternehmen bedarf es einer individuellen Überprüfung, ob eine Einführung eines Workflowmanagmentsystem eine ausreichende Kostenersparnis bringt, um die zusätzlichen Kosten, die durch die Umstellung der internen Abläufe auftreten, zu decken. Allerdings kann eine zunehmende Automatisierung den Schritt der Umstellung unumgänglich machen. Vorerst bleibt es größeren Unternehmen vorbehalten ihre angebotenen Services in das intern genutzte Workflow Managment System zu integrieren. Eine schnelle Einigung der großen Anbieter IBM, Microsoft und SUN, womöglich auf den von IBM entworfenen Standard, würde diesen Unternehmen mehr Investitionssicherheit bieten.

6 Literatur

[SOAP] Simple Object Access Protokol (SOAP)
http://www.w3c.org/TR/SOAP

[WSDL] Web Service Description Language (WSDL)
http://www.w3c.org/TR/wsdl.html

[UDDI] Universal Description Discovery and Integration (UDDI)
http://www.uddi.org/

[JSP] Fields, Duane K.: Java Server Pages : dynamische Webservices entwickeln / Duane K. Fields ; Mark A. Kolb ISBN 3-8273-1804-1

[MColan] Colan, M.: SOAP - Security and Reliability, Issues and Solution
ftp://www6.software.ibm.com/software/developer/library/mcolan/SOAP-SecurityIssues-Solutions.pdf

[SaHi] Satoshi, H. und Hiroshi, M.: SOAP Security Extensions
http://www.trl.ibm.com/projects/xml/soap/wp/wp.html

[XKMS] XML Key Managment Spezifikation (XKMS)
http://www.w3c.org/TR/xmks

[SAML] Security Assertion Markup Language (SAML)
http://www.oasis-open.org/committees/security/

[XACML] Extensible Access Control Markup Language (XACML)
http://www.oasis-open.org/committees/xacml/

[WSFL] Web Services Flow Language
http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf

[MQWF] MQ Workflow
http://www-3.ibm.com/software/ts/mqseries/workflow/

[XLANG] XLANG
http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.html