

Standardisierung von Webservices

-

Teleseminar Webservices

Seminararbeit
von
Christian Seifried
aus
Mannheim

vorgelegt am
Lehrstuhl für Praktische Informatik IV
Prof. Dr.-Ing. Wolfgang Effelsberg
Fakultät für Mathematik und Informatik
Universität Mannheim

Juni 2004

Betreuer: Dipl.-Wirtsch.-Inf. Jürgen Vogel

Inhaltsverzeichnis

1	Einleitung	1
2	Abgrenzung der Begrifflichkeiten	3
3	Vergleich von Standards gegenüber proprietären Lösungen	4
4	Wo werden Standards benötigt?	6
5	Standardisierungsgremien im Bereich Web Services	9
5.1	W3C - World Wide Web Consortium	9
5.2	OASIS	10
5.3	IETF	11
5.4	UN/CEFACT	11
5.5	WS-I	11
5.6	Liberty Alliance	12
6	Wie ist die aktuelle Marktsituation?	13
7	Ausblick	15
	Literatur	16

1 Einleitung

Die Idee hinter Web Services liegt in der Übertragung von Daten und Funktionalität zwischen Kommunikationspartnern. Diese Idee ist nicht unbedingt neu. Was Web Services jedoch von der herkömmlichen Webarchitektur unterscheidet, ist die Kommunikation rein zwischen zwei Maschinen. Eine Maschine bezeichnet in diesem Fall eine Applikation, die z.B. auf einem Webserver installiert ist. Beim herkömmlichen Client/Server-Ansatz des Web fordert der Client oft nur Daten vom Server an, um diese im Browser graphisch darzustellen. Diese Funktionsweise von Web Services fasst man unter dem Begriff der "Service-orientierten Architektur - SOA" zusammen.

Da diese Dienste oder auch Services auf verschiedensten Plattformen, Betriebssystemen laufen und auch mit unterschiedlichen Entwicklungssprachen geschrieben sind, ist es nötig, Standards einzuführen, die einen Daten- und Funktionsaustausch zwischen allen Beteiligten erlaubt. Diese Standards und Spezifikationen formen letztendlich den Begriff "Web Services". Dies ist auch das Hauptthema dieser Arbeit und soll im Anschluss an diese allgemeine Einführung näher erörtert werden.

Bevor jedoch auf Standards im Bereich Web Services eingegangen wird, ist es zunächst einmal interessant zu wissen, wofür sie überhaupt benötigt und verwendet werden. Hierfür gibt es unzählige Anwendungsszenarien, die in Frage kommen. Das Bekannteste ist wohl die einfache Suche über **Google**. Hierbei können private Nutzer die **Google**-Suche per Web Service nutzen und auf ihrer eigenen Homepage anbieten. Ein weiteres solches Beispiel ist die Shop-Suche bei Amazon, die ebenfalls per Web Service genutzt werden kann. Man erkennt jedoch schnell, dass diese einfachen Beispiele nur einseitig arbeiten, und es kommt die Frage auf, ob diese Funktionalität nicht auch mit herkömmlichen Mitteln wie **RMI** oder **CORBA** bereitgestellt werden könnte. Web Services erlauben allerdings auch eine automatisierte Kommunikation in beide Richtungen. Dadurch ist es vorstellbar ganze Geschäftsanwendungen, wie z.B. Warenwirtschaftssysteme, über das Web miteinander zu verbinden. Hier kann ein Kritiker wieder einbringen, dass das auch mit existierenden Mitteln gelöst werden kann, jedoch haben Web Services einige Vorteile:

- Web Services sind im Vergleich zu **CORBA**, **RMI** und **DCOM** sehr einfach zu handhaben.
- Die Kommunikation verläuft im Regelfall über **HTTP** und ist damit über das Internet einsetzbar und gilt als "Firewall-friendly". Es können auch andere Protokolle wie **SMTP** eingesetzt werden.
- Web Services basieren, zumindest in der Grundfunktionalität, auf anerkannten Standards.
- Es stehen für fast alle wichtigen Programmiersprachen Implementierungen der Standards zur Verfügung.

- Die Web Service Technologie basiert auf XML, wofür eine Vielzahl von Tools zu dessen Bearbeitung schon existieren.

Nachteile von Web Services sind sicherlich die noch fehlenden Standards in den Bereichen Sicherheit, Prozesse und Transaktionen. Dies hat in letzter Zeit zu einer Flut an Spezifikationen von Seiten der verschiedenen Herstellern geführt, die natürlich alle versuchen möglichst schnell diese Nachteile in ihrer eigenen Lösung auszumerzen. Darauf wird im späteren Verlauf noch detaillierter eingegangen.

Das Thema wurde zusammen mit Herrn Jens Daum bearbeitet, sodass vor allem bei den Themen “Standardisierungsgremien” und “Abdeckung durch Standards” auf die Ausarbeitung von Herrn Daum verwiesen wird.

2 Abgrenzung der Begrifflichkeiten

Bevor auf das eigentliche Thema der Standardisierung von Web Services eingegangen werden kann, muss zunächst geklärt werden, was man überhaupt unter einem "Standard" versteht und was diesen von Begriffen wie Spezifikation oder proprietärer Lösung unterscheidet. Wer sich mit diesem Thema in der Fachliteratur oder im Internet auseinandersetzt wird schnell bemerken, dass nicht immer ganz klar ist, was die einzelnen Autoren unter den verschiedenen Begriffen verstehen. Für diese Arbeit sind vier grundlegende Bezeichnungen von Bedeutung:

1. **Herstellerstandard** - Ein Herstellerstandard ist kein offiziell anerkannter Standard. Er wird meist von einem Unternehmen als die beste Lösung am Markt propagiert, um das eigene Produkt besser am Markt zu platzieren, indem es als standardkonform verkauft werden kann. Die Weiterentwicklung dieses sogenannten "Standards" liegt beim Unternehmen selbst, keine unabhängigen Organisationen haben darauf Einfluss oder Einsicht, vor allem deshalb, weil der Quellcode meist nicht veröffentlicht wird. Ein Beispiel wäre pdf von Adobe oder auch J2EE von SUN.
2. **Spezifikation** - Darunter versteht man eine veröffentlichte Technologiebeschreibung. Sie ist aber noch kein Standard. Die meisten veröffentlichten Dokumente im Bereich Web Services haben diesen Status.
3. **Standard** - Eine Spezifikation wird dann zu einem Standard, wenn sie sich gegenüber anderen Technologien durchgesetzt hat, und von einem unabhängigen Standardisierungsgremium verwaltet wird.
4. **Implementierung** - Ein Standard alleine besitzt noch keine Funktionalität. Erst die dazugehörigen technischen Implementierungen bieten die Möglichkeit Applikationen mit diesen Technologien standardkonform zu entwickeln. Die Implementierungen lassen sich in drei Bereiche untergliedern. Zum einen sind das die Pioniere wie SUN oder Microsoft, die um die Vorherrschaft am Markt konkurrieren. Zum anderen die Softwarehersteller von verschiedensten Anwendungen wie SAP oder Oracle, die für ihre Systeme Schnittstellen zu diesen Technologien integrieren, und zu guter letzt die vielen unabhängigen Lösungen. So existiert z.B. schon für viele Web-Skriptsprachen eine Web Service Implementierung.

Neben diesen vier Begriffen existieren natürlich noch einige mehr. Diese werden jedoch oft nur von einem bestimmten Gremium verwendet wie z.B. "RFC"¹ oder "Profile"² und werden deshalb hier nicht alle im Einzelnen erläutert.

¹Request for Comment

²Bezeichnung des WS-I

3 Vergleich von Standards gegenüber proprietären Lösungen

Unter einer proprietären Lösung versteht man eine Lösung, die keinen Standards folgt. Bevor sich ein Unternehmen für eine Anwendung im Bereich Web Services entscheidet, stellt sich zunächst einmal die Frage, ob die Lösung mit eigenen Mitteln komplett selbst entwickelt werden soll, oder ob man auf ein bestehendes Framework aufbaut, das gewisse Standards berücksichtigt. Bei den großen Softwareunternehmen wie Microsoft oder SUN ist die Entscheidung zu treffen, ob man eine neue Technologie bzw. ein neues Produkt nach den Vorgaben eines Standards konzipiert oder eigene Wege geht. Beide Möglichkeiten haben in der jeweiligen Situation ihre Vor- und Nachteile, sodass nicht von vorne herein schon gesagt werden kann, dass eine standardkonforme Lösung immer die optimale ist. Die proprietäre Lösung hat ihre Stärken vor allem darin, dass sie auf ein gegebenes Umfeld optimal angepasst und entwickelt werden kann. Ein Standard hat immer den Anspruch, für möglichst viele unterschiedliche Szenarien und Anwendungsfälle allgemein gültig zu sein. Dadurch muss er eine Vielzahl von Bedingungen abdecken, die im konkreten Fall für ein Unternehmen so niemals relevant sein werden. Das bedeutet also unter Umständen einen unnötigen Overhead, der auch zu erheblichem Mehraufwand in der Entwicklung und in der Konfiguration führen kann. Dies kann ebenso zu einer höheren Komplexität des ganzen Projektes führen. Die Performance kann ebenfalls beeinträchtigt werden, wenn man sich vorstellt, dass z.B. beim Austausch von Nachrichten jedesmal eine Vielzahl von nicht verwendeten Informationen mitgeschickt und verarbeitet werden müssen. Hier könnte man sich z.B. vorstellen, dass eine standardisierte Lösung Nachrichten nur verschlüsselt verschickt, da normalerweise über das unsichere Internet kommuniziert wird. Im konkreten Fall findet die Kommunikation aber nur intern statt, und so geht unnötig Zeit durch das Ver- und Entschlüsseln verloren.

Diese Nachteile eines Standards sollten aber in Kauf genommen werden, sobald der geplante Web Service nicht nur intern oder im B2B eingesetzt wird, sondern öffentlich angeboten und verfügbar sein soll, wie es die eigentliche Idee von Web Services vorsieht. Als Entwickler eines Web Service hat man meist das Interesse, seinen Dienst an möglichst viele Konsumenten zu verkaufen. Dies erfordert, dass möglichst viele Konsumenten den Service zunächst einmal in irgend einer Weise finden, ihn als den für sie passenden identifizieren und mit diesem dann auch kommunizieren können. Hält sich der Entwickler nicht an einen Standard, wird die Zahl der potenziellen Interessenten von vorne herein stark eingeschränkt, da eine Vielzahl von unterschiedlichen Systemen und Applikationen nicht mit der proprietären Lösung kompatibel sind. Bei den großen Softwareherstellern versuchte Microsoft seine DCOM-Technologie ohne einen zugehörigen Standard durchzusetzen und war dadurch nur auf Windows-basierten Rechnern einsetzbar. Dies führte zum Scheitern dieser Technologie, da es mittlerweile einige Alternativen gibt, die durch ihre standardkonforme Architektur auf vielen Systemen einsetzbar sind. Oft versuchen die Hersteller auch ihren proprietären Ansatz als den Standard anzuprei-

sen, um ihr Produkt besser am Markt abzusetzen. Dies widerspricht jedoch der eigentlichen Idee die hinter Standards steckt und wurde auch kurz in der Definition in Kapitel 2 erwähnt.

Es sollte hier klargemacht werden, dass Lösungen, die im Bereich Web Services langfristig eine Überlebenschance haben wollen, unbedingt zu den wichtigsten Standards kompatibel sein müssen. Sicherlich kann in speziellen Bereichen wie Sicherheit auf verschiedene Spezifikationen zurückgegriffen werden, die noch nicht den Status eines Standards haben. Die grundlegenden Bausteine wie sie im nächsten Kapitel beschrieben sind, sollten jedoch alle auf den gleichen Standards basieren, um eine Interoperabilität zwischen den unzähligen Lösungen zu gewährleisten. Wird dies nicht beachtet, kann es sein, dass die proprietäre Lösung schnell überholt und nicht mehr einsetzbar ist, oder nur mit viel Aufwand nachträglich an eine neue Technologie angepasst werden kann. Auf proprietäre Ansätze sollte man in diesem Umfeld also aus den genannten Gründen verzichten. Allgemein gilt dies für alle neuen Lösungen, die in einer verteilten bzw. vernetzten Umgebung mit vorher nicht bekannten Teilnehmern ablaufen sollen, da oftmals eine Vielzahl von unterschiedlichen Systemen im Netz beteiligt sind.

Die bisher entwickelten Lösungen sind oftmals nur im B2B Bereich eingesetzt und somit nicht unter dem hier verwendeten Begriff von Web Services einzuordnen. Ihre Aufgaben unterscheiden sich von denen eines Web Service, sodass in ihrem Fall durchaus eine proprietäre Lösung die bessere sein kann.

Im Anschluss an diese Motivation für die Berücksichtigung von Standards folgt nun ein Abschnitt, indem aufgezeigt werden soll, wo Standards im Bereich Web Services überall essentiell sind.

4 Wo werden Standards benötigt?

Web Services benötigen Standards, dies wurde im vorherigen Abschnitt deutlich. Es stellt sich nun die Frage, in welchen Bereichen diese benötigt werden und welche Aufgabe sie einheitlich übernehmen sollen. Dazu soll zuerst einmal der vereinfachte, grundlegende Aufbau der service-orientierten Architektur (SOA), der den Web Services zugrunde liegt und der allgemeine Ablauf eines Web Service beschrieben werden.

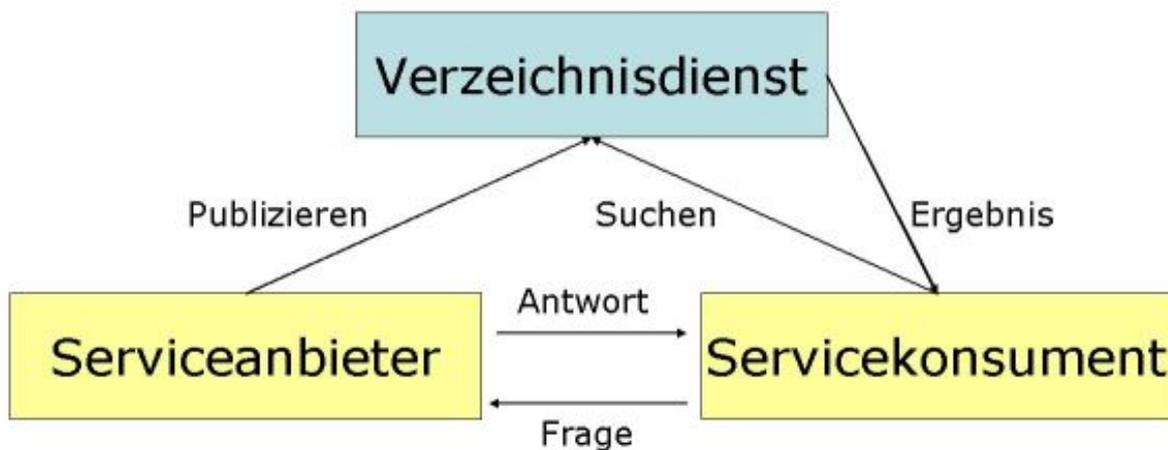


Abbildung 1: Vereinfachter Aufbau eines Web Service. Siehe auch [Hau04], Seite 16

Wie in der Abbildung zu sehen ist, existieren drei Basiskomponenten:

- **Serviceanbieter** - Der Serviceanbieter entwickelt eine Web Service Applikation. Er möchte diesen Service beim Verzeichnisdienst öffentlich zur Verfügung stellen, damit er von Suchdiensten gefunden werden kann. Der Dienst selbst bleibt auf dem Anbieterserver, der Verzeichnisdienst stellt nur eine Art Telefonbuch dar.
- **Servicekonsument** - Der Konsument möchte einen Dienst in Anspruch nehmen, der die vom Anbieter gebotene Funktionalität hat. Er sucht diesen Service zuerst beim Verzeichnisdienst und findet ihn auch dort. Anschließend erhält er eine URL des Anbieters und eine direkte Kommunikation zwischen Anbieter und Konsument findet statt.
- **Verzeichnisdienst** - Der Verzeichnisdienst stellt einen zentralen Zugriffspunkt zum Bereitstellen bzw. Auffinden von Services dar. Er ermöglicht dem Konsumenten nach bestimmten Kriterien einen Service zu suchen. Dies geschieht über

Matchmaking, einer Technologie die nach bestimmten Übereinstimmungen zwischen Angebot und Anfrage sucht und dadurch einen möglichst passenden Dienst findet. Für den Anbieter ist es also entscheidend, seinen Dienst bei möglichst vielen Verzeichnisdiensten mit den richtigen Schlagwörtern anzubieten, damit möglichst viele Service-Konsumenten den Dienst finden können und diesen auch als zutreffend deklarieren.

Man sieht also, dass es verschiedene Bereiche gibt, die einer einheitlicher Regelung unterliegen müssen. Im Folgenden werden die einzelnen Bereiche genannt und aufgezeigt, warum es nötig ist, dass hier Standards eingesetzt werden.

Der wohl wichtigste Bereich ist die Übermittlung. Für die Kommunikation zwischen Service-Anbieter, Service-Konsument und Service-Verzeichnis ist die Übermittlung von Nachrichten nötig. Sind diese Nachrichten nicht standardisiert, so können die einzelnen Komponenten sich nicht verständigen und es könnte kein Austausch von Daten und Funktionalität stattfinden. Vor allem bei unterschiedlichen Plattformen und verschiedenen Sprachen, mit denen die einzelnen Services implementiert wurden, ist schnell klar, dass ein von allen verständliches Protokoll existieren muss. Hier hat sich das auf XML basierende SOAP gegenüber anderen Ansätzen wie z.B. XML-RPC durchgesetzt.

Ebenso muss die Beschreibung der Dienste einer standardisierten Norm folgen. Der Service-Anbieter muss eine Möglichkeit haben dem Service-Konsumenten seinen Service und seine bereitgestellten Methoden zu beschreiben. Diese Beschreibung muss natürlich vom Konsumenten verstanden werden. Idealerweise sollte dieser Dienst ohne Einwirken einer Person für eine vorliegende Aufgabe beim Konsumenten anhand einer solchen Beschreibung gefunden werden können. Dahinter steckt der Wunsch nach einer reinen "Maschine-zu-Maschine" Kommunikation. Für die Beschreibung ist der WSDL-Standard vorgesehen, auf den später nochmals kurz eingegangen wird.

Der Verzeichnisdienst muss natürlich auch eine einheitliche Funktionalität bieten, um einerseits einen Service bei ihm anzumelden, und auf der anderen Seite nach diesem Service auch suchen zu können. Desweiteren muss auch bekannt sein, in welcher Form das Ergebnis einer Suchanfrage vom Verzeichnisdienst zurückgeliefert wird. Unter den Standards für den Verzeichnisdienst hat sich UDDI durchgesetzt, daneben existieren aber noch eine Reihe von herstellerepezifischen Lösungen wie DISCO von Microsoft.

Die Standards dieser drei Bereiche bilden die eigentliche Web Service Technologie. Darüber hinaus existieren aber noch eine Reihe weiterer Bereiche, bei denen der Bedarf an Standards vorliegt. Dies ist insbesondere dann der Fall, wenn Web Services komplexere Geschäftsprozesse abbilden sollen. Hierzu zählt u.a. der Bereich der Sicherheit. Sobald Web Services über das öffentliche Netz miteinander kommunizieren - wofür sie ja auch konzipiert wurden - und vertrauliche Daten übertragen werden, ist es nötig sich auf bestimmte Kryptographieverfahren im Voraus zu einigen. Mögliche Verfahren betreffen z.B. die Authentifizierung, die Autorisierung oder auch die Verschlüsselung von Daten mit einem `publicKey`-Verfahren.

Da Web Services vom Ansatz her zustandslos sind, ist im Bereich der Transaktionskontrolle ebenfalls eine Vereinbarung auf bestimmte Verfahren nötig, sobald komplexere Prozesse abgebildet werden sollen. Hierunter könnte man sich eine Warenkorbfunktionalität vorstellen, die aus mehreren einzelnen Anfragen eine geschlossene Transaktion bildet. Wenn man bei diesem Beispiel bleibt und es noch mit einer zusätzlichen Funktion "Lagerkontrolle" erweitern möchte, die in einem eigenen Web Service schon zur Verfügung steht, dann sieht man, dass auch die Koordination dieser Services irgendwie geregelt sein muss. Es muss geregelt werden, wie die Daten von einem Service an den nächsten gegeben werden und wann die einzelnen Services ihre Aufgaben erledigen müssen. Diesen Technologiebereich fasst man unter dem Begriff des "Management von Web Services" zusammen, oft werden aber auch die Begriffe "Choreographie bzw. Orchestrierung von Web Services" verwendet.

Da viele Anbieter von Web Services natürlich auch kommerzielle Interessen haben, ist es verständlich, dass für sie die Abrechnung von Diensten eine wichtige Rolle spielt. Man kann sich leicht vorstellen, dass es nicht unbedingt einfach ist, ein geeignetes Maß für die Benutzung von Services zu finden, sodass der Service weiterhin für den Konsumenten lukrativ bleibt. Dieses Gebiet ist im Bereich Web Services noch relativ neu, sodass unter dem Begriff des "Billing" erst wenige Vorschläge existieren, erst recht liegen noch keine verabschiedeten Standards vor.

Die Liste an eingereichten Spezifikationen in den letztgenannten Bereichen ist mittlerweile zu einer unüberschaubaren Liste angewachsen, sodass im nächsten Kapitel bei der Vorstellung der Gremien im Bereich Web Services und deren Standards nur auf die eigentlichen, bis heute verabschiedeten Standards eingegangen werden kann.

5 Standardisierungsgremien im Bereich Web Services

Damit ein Standard sich auch wirklich “Standard” nennen darf, ist wie im Kapitel 2 bereits erwähnt ein unabhängiges Gremium notwendig. Meist werden neue Technologien von einzelnen Unternehmen oder Kooperationen von Unternehmen entwickelt und dann bei einem Gremium zur Standardisierung eingereicht. Die Unternehmen streben natürlich nach einer möglichst schnellen Standardisierung ihrer Technologie, da dies dem Kunden, der diese später kauft und verwendet eine gewisse Sicherheit gibt, dass sich diese Technologie nicht als Flop herausstellt und schnell durch eine neuere Technologie ersetzt wird.

Der Ablauf der Standardisierung unterscheidet sich natürlich bei den einzelnen Gremien, beispielhaft soll hier nur auf das W3C eingegangen werden. Der ausführliche Aufbau des Gremiums lässt sich auf der Homepage vom W3C finden³.

Zunächst erhalten alle beim W3C eingereichten Vorschläge durch eigene Arbeitsgruppen oder von extern den Status “Note”. Dieser liegt außerhalb der vier eigentlichen Stufen zur Standardisierung und sagt nichts über die weitere Bearbeitung des Vorschlags aus. Die nächste Stufe ist das “Working Draft”. Es sagt aus, dass sich eine Spezifikation aktuell in Arbeit befindet und zur Diskussion freigegeben wurde.

Auf die Stufe der “Candidate Recommendation” kommt eine Spezifikation nach mehreren Working Drafts. Es werden nun erste Implementierungen entwickelt, die eine Praxistauglichkeit der Spezifikation testen sollen.

Die dritte Stufe im Lebenszyklus einer Spezifikation ist die “Proposed Recommendation”. Sie wird noch einmal von allen Mitgliedern des W3C geprüft und dann abschließend beurteilt.

Die höchste Stufe, also die Stufe des Standards, ist die “Recommendation”. Eine Spezifikation ist damit als **Standard** verabschiedet und wird nicht mehr weiter entwickelt oder geändert. Dies wird erst in eine neue Version des Standards einfließen.

Wie man sieht ist also der Standardisierungsprozess eine recht lange Prozedur, die vielen Unternehmen zu lange dauert. So werden oft Standards zuerst beim einen Gremium eingereicht, dort aber nicht gleich bearbeitet und deshalb zu einem anderen Gremium übertragen. Ein Beispiel hierfür ist BPEL. Es wurde von Microsoft erst beim W3C eingereicht, nach Kritik am Vorgehen vom W3C aber an OASIS übergeben. Im nachfolgenden Abschnitt werden die Gremien im Detail vorgestellt.

5.1 W3C - World Wide Web Consortium

Das bereits mehrmals erwähnte W3C⁴ ist das bedeutendste Gremium im Bereich der Web Services. Es wurde von Tim Berners-Lee gegründet, der als Internet-Vordenker gilt. Das

³siehe [Wor]

⁴<http://www.w3.org>

W3C verwaltet einige sehr wichtige Standards im Bereich des Internets, wie z.B. XML oder auch HTML. So ist es auch nicht verwunderlich, dass beim W3C auch die Standards, die für die Übermittlung von XML-Nachrichten verantwortlich sind, verwaltet werden. Im speziellen sind das SOAP und WSDL. SOAP hat sich gegenüber anderen Technologien wie XML-RPC im Bereich der Nachrichtenübermittlung durchgesetzt. SOAP steht ursprünglich für Simple Object Access Protocol, was aber die eigentlichen Aufgaben von SOAP nicht richtig wiedergibt, da es ja nicht dem direkten Objektzugriff dient wie beispielsweise RMI oder CORBA. SOAP wurde im April 2000 von Microsoft beim W3C eingereicht. Der zweite wichtige Web Service Standard beim W3C ist WSDL, die führende Technologie um Web Services zu beschreiben, wie auch schon in Kapitel 4 erwähnt wurde. WSDL wurde im März 2001 beim W3C eingereicht. Mitwirkende waren u.a. BEA, Commerce One, IBM, SAP, Microsoft und Intel. Das W3C ist in mehrere Arbeitsgruppen untergliedert, die für verschiedene Bereiche verantwortlich sind, wie z.B. die "WS Architecture Working Group" oder die "WS Description Working Group".

5.2 OASIS

OASIS⁵ ist hinter dem W3C das wohl bedeutendste Gremium im Bereich Web Services. Es wurde 1993 unter dem Namen "SGML Open" gegründet. SGML gilt als der Vorgänger von XML. Als dies aber vom W3C übernommen wurde, änderte OASIS daraufhin 1998 seinen Namen. OASIS betreut heute hauptsächlich auf geschäftliche Anwendungen bezogene Standards, wie ebXML. ebXML⁶ ist ein Satz an Spezifikationen zur Abwicklung zwischenbetrieblicher Geschäftsprozesse mittels XML-Nachrichten. Es ist der Nachfolger von EDI. Zudem werden von OASIS aber auch andere Spezifikationen rund um Web Services betreut, wie z.B. der Verzeichnisdienst UDDI oder WS-Security als Sicherheitslösung. OASIS selbst bezeichnet sich als ein "globales, non-profit Konsortium für e-Business Standards".

OASIS wird heute von vielen Unternehmen bei der Standardisierung vorgezogen, da es als flexibler und schneller gilt als das W3C, bei dem eher die kontextunabhängigen Standards verwaltet werden. Die schnellere Bearbeitung folgt vor allem aus der reduzierten Anzahl von Stufen, die eine Spezifikation bis zum Standard durchlaufen muss, im Vergleich zum W3C. Es existieren hier nur die drei Stufen OASIS Committee Draft, Committee Specification und Standard.

⁵Organization for the Advancement of Structured Information Standards,
<http://www.oasis-open.org>

⁶Electronic Business eXtensible Mark-up Language

5.3 IETF

Die IETF⁷ ist zwar weniger bekannt als die beiden oben genannten im Bereich Web Services, jedoch verwaltet sie einige der wichtigsten Internet-Standards, unter anderem auch HTTP. Beim IETF existiert auch eine Alternative zu Web Services: AS2. AS2 bietet grundlegende, aber nicht erweiterbare Sicherheits- und Zuverlässigkeitsfeatures für den Transfer von XML- oder Binärdateien.

Die Standards des IETF sind wohl jedem schon mal aufgefallen. Es sind die RFC (Request for Comment), die mit einer fortlaufenden vierstelligen versehen abgelegt werden. Aktuellere oder neue Standards haben also eine höhere Nummer als die alten. Das IETF hat vom "Internet-Draft" bis zum "Full Standard" wie das W3C fünf Standardisierungsstufen.

Die RFC können im Web unter <http://ietf.org/rfc/rfcXXXX.txt> angeschaut werden, wobei "XXXX" die oben erwähnte vierstellige Nummer ist.

5.4 UN/CEFACT

Die CEFACT⁸ ist eine Organisation der Vereinten Nationen (UN). Sie kümmert sich unter anderem um den Bereich E-Business. Sie ist die zweite Kraft hinter OASIS, die ebXML vorantreibt. Man kann sich jedoch vorstellen, dass bei der gemeinsamen Standardisierung oft Konflikte im Bezug auf Ziele und Ausrichtungen auftreten, da unterschiedliche Interessen bei einer politisch, öffentlichen Organisation und einer Organisation wie OASIS, deren Mitglieder aus der freien Wirtschaft aus großen Unternehmen stammen, existieren.

5.5 WS-I

*"The Web Services Interoperability Organization is an open industry effort chartered to promote Web Services interoperability across platforms, applications, and programming languages. The organization brings together a diverse community of Web services leaders to respond to customer needs by providing guidance, recommended practices, and supporting resources for developing interoperable Web services."*⁹

Das WS-I¹⁰ ist das jüngste der bisher vorgestellten Gremien. Es wurde unter der Vorwand gegründet, Interoperabilität zwischen den einzelnen Web Service Implementationen zu schaffen. Der eigentliche Grund war jedoch ein anderer: IBM und Microsoft [Mic], die Gründer des WS-I wollten dem W3C etwas Einfluss entziehen im Bereich

⁷Internet Engineering Task Force, <http://www.ietf.org>

⁸United Nations Centre for Trade Facilitation and Electronic Business,
<http://www.unece.org/cefact>

⁹<http://www.ws-i.org/AboutUS.aspx>

¹⁰Web Service Interoperability Organization, <http://www.ws-i.org>

Web Services. Ein weiterer Grund war wohl auch die Schwächung eines ihrer stärksten Konkurrenten (SUN) in diesem Bereich. Jedoch ist SUN mittlerweile auch diesem Gremium beigetreten. Das WS-I ist ähnlich wie das W3C auch in Arbeitsgruppen, die für verschiedene Aufgaben zuständig sind, aufgeteilt.

Was das WS-I jedoch von den anderen Gremien ein wenig unterscheidet, ist die Tatsache, dass das WS-I nicht versucht, noch mehr Standards zu etablieren, sondern bestehende Standards verschiedener Gremien zu einem sogenannten "Basic Profile" zusammenzufassen. Es pickt sich also einige Spezifikationen aus einem Bereich heraus, die eine Kommunikation und Austausch von Daten und Funktionalität der verschiedenen Implementierungen auf Basis dieser Profiles dann gewährleisten sollen. Die zu unterstützenden Spezifikationen, mit den jeweiligen Verweisen auf die Gremien, sind immer im "Anhang A" eines solchen Profiles untereinander aufgelistet.

5.6 Liberty Alliance

*"The Liberty Alliance's vision is one of a networked world in which individuals and businesses can more easily interact with one another while respecting the privacy and security of shared identity information."*¹¹

Das letzte der hier vorgestellten Organisationen ist die Liberty Alliance. Sie ist nicht unbedingt mit den vorangehenden Organisationen gleichzustellen, da ihr eigentliches Gründungsziel nicht die Schaffung von Standards war. Vielmehr wurde sie von SUN Microsystems gegründet, die Kritik an dem von Microsoft entwickelten ".NET Passport" hegten¹².

Microsoft wollte mit dem virtuellen Internet-Ausweis eine einheitliche Möglichkeit schaffen, eine Identität im Internet zu prüfen. .NET Passport ist ein Onlinedienst, über den man sich mit einer E-Mail-Adresse und einem einzigen Kennwort bei einer verschiedenen Partnersites bzw. bei jedem Partnerservice von .NET Passport anmelden kann. Dafür sollten aber alle nötigen und vor allem auch vertraulichen Daten zentral auf Microsoft Server gespeichert werden und es sind wie oben schon erkennbar wird, nur die von Microsoft unterstützen Seiten mit diesem System zugänglich.

Man kann sich leicht ausmalen, dass dies große Bedenken bei Datenschützern und großen Unternehmen wie Banken etc. hervorbrachte. So umfasst die Organisation heute mittlerweile 40 kommerzielle und nicht-kommerzielle Organisationen, u.a. auch Siemens, T-Online oder SAP.¹³ Heute verwaltet das Liberty Alliance Project neben Spezifikationen zur Authorisierung und Identitätsprüfung einige weitere Spezifikationen im Bereich Sicherheit.¹⁴

¹¹<http://www.projectliberty.org/about/index.html>

¹²siehe auch [Mica]

¹³Eine Liste aller Mitglieder ist unter <http://www.projectliberty.org/membership/members.asp> einzusehen

¹⁴http://www.projectliberty.org/specs/index.html#ID-FF_Specs

6 Wie ist die aktuelle Marktsituation?

Nachem in den bisherigen Kapiteln aufgezeigt wurde, welche Organisationen sich in welchen Bereichen um Standards bemühen, ist es natürlich auch interessant zu sehen, wie diese Standards von den Unternehmen und Endkunden angenommen bzw. implementiert werden.

Bei den Implementierungen haben Microsoft mit seiner .NET Strategie auf der einen Seite, und SUN Microsystems mit seiner J2EE Technologie auf der anderen, die führende Marktstellung. Weitere Anbieter wie Oracle, IBM oder auch BEA setzen bisher alle auf die J2EE-Plattform auf, wobei die meisten Implementierungen Web Services mit Hilfe von EJBs¹⁵ realisieren. Dies gilt ebenso für das von Apache stammende Axis¹⁶, das eine Implementierung des SOAP Standards 1.1 ist. Man sieht, dass die meisten größeren Serveranbieter auf Java J2EE setzen. Meist wird als Grund die bessere Skalierbarkeit genannt. Auf einigen Webseiten von Anbietern wird mittlerweile eine Unterstützung von .NET angekündigt, wie z.B. bei Oracle.

Einfache Web Service Implementierungen können in Java mithilfe des Java WSDP¹⁷ realisiert werden. Hierin sind verschiedene APIs enthalten, mit denen SOAP Nachrichten, SOAP Nachrichten mit Anhängen verarbeitet oder auch Verzeichnisdienste realisiert werden können. Das WSDP 1.4 unterstützt die Standards SOAP, WSDL, UDDI und ebXML, desweiteren ist es konform zum Basic Profile 1.1 with Attachments Profile 1.0 vom WS-I.

Das .NET Framework wird hingegen auf der Clientseite von den meisten Endkunden eingesetzt. Die Gründe hierfür sind die einfachere und schnellere Entwicklungszeit für die Erstellung eines einfachen Clients. Zudem existieren für immer mehr Programmiersprachen Compiler, die in den sprachenunabhängigen MSIL Bytecode übersetzen.

Da die Web Service Technologie noch recht jung ist, existieren wenige Statistiken, die konkrete Zahlen für diesen Bereich liefern. Eine der wenigen Umfragen ist von der Gartner Group¹⁸ vom Februar 2003. Hier wurden 44 Beratungsunternehmen und Systemhäuser in Nordamerika nach dem Einsatz von Web Services gefragt. Dabei stellte sich heraus, dass .NET von 58% als eines der Top Three Produkte angesehen wird. 40% geben IBM WebSphere und 31% Oracle als eines unter drei favorisierten Produkten an. Eine im August 2002 erstellte Umfrage ergab, dass 33% der Endkunden einen Einsatz mit .NET planen, wohingegen 39% auf die J2EE Technologie setzen. Große Firmen scheinen aber eher zu Java zu tendieren, während kleine Unternehmen eine schnellere Entwicklung mit .NET vorziehen.

Bei einer Befragung¹⁹ von 1000 internationalen Unternehmen im Februar 2004 zum all-

¹⁵Enterprise Java Beans

¹⁶siehe [Apa]

¹⁷Web Service Development Pack

¹⁸<http://www.gartner.com>

¹⁹siehe [Wes04]

gemeinen Einsatz von Web Services gaben 44% an, auf der Suche nach einem WS-Tool zu sein, 37% setzen bereits ein solches ein, 26% planen den Einsatz in den nächsten sechs Monaten. Hier wurde angegeben, dass 43% mit .NET und nur 35% mit Java entwickelt wurde. Beachtlich ist, dass bisher 70% der Nutzer von Web Services dies in ihrem internen Netz tun. Es gaben jedoch 48% an, eine externe Nutzung in nächster Zeit zu planen.

7 Ausblick

Ruft man sich die Argumentation bezüglich der Notwendigkeit von Standards in Erinnerung, so ist es verwunderlich, dass sowohl der Marktüberblick als auch die Übersicht über konkurrierende Spezifikationen speziell in den Bereichen Web Service Sicherheit und Web Service Prozessmanagement auf den ersten Blick ein sehr heterogenes Bild bieten. Auf den zweiten Blick lässt sich konstatieren, dass berechnete Standards - losgelöst von allen wirtschaftlichen Interessen - sich erst am Ende einer technologischen Lernkurve herauskristallisieren können. Die zahlreichen, konkurrierenden Spezifikationen und Implementierungen dokumentieren somit die fortschreitende und anhaltende Entwicklung im Bereich Web Services, welche auf dem Kern ²⁰, bestehend aus SOAP, WDSL und UDDI aufbaut. Überspitzt formuliert, verbietet sich eine frühzeitige und weitgehende Standardisierung in den Randbereichen, da technologische Spitzenleistung bzw. technologischer Fortschritt sich nicht standardisieren lassen.

Andererseits hängt die Weiterentwicklung von Web Services bzw. deren Verbreitung stark von den wirtschaftlichen und wirtschaftspolitischen Interessen der beteiligten Firmen und Organisationen auf allen Stufen ²¹ ab. Sollte sich die in der Einführung genannte Vision der Web Service Befürworter nicht einstellen oder zu lange verzögern, werden die Pioniere am Softwaremarkt allein aus Eigeninteresse ein neues weiterentwickeltes Produkt zu etablieren suchen ²². Ebenso wie Endanwender nach neuen Möglichkeiten Ausschau halten werden, welche Ihre Anforderungen besser oder weitgehender erfüllen können.

²⁰ vgl. Web Service Architecture Stack, Kapitel 4 aus der Arbeit von Herrn Jens Daum

²¹ Entwickler, Schnittstellenanbieter, Endanwender und Standardisierungsgremien

²² Zur Untermauerung dieser Aussage betrachte man nur die Entwicklung von DCOM, welches von Microsoft zu Gunsten von .NET und Web Services aufgegeben wurde.

Literatur

- [Apa] APACHE SOFTWARE FOUNDATION: *Axis User's Guide*. Internet Site. <http://ws.apache.org/axis/java/user-guide.html>.
- [Hau04] HAUSER, TOBIAS UND LÖWER, ULRICH M.: *Web Services - Die Standards*. Galileo Press GmbH, Bonn, 1. Auflage Auflage, 2004.
- [Mica] MICROSOFT: *.NET Passport*. Internet Site. <http://www.passport.net>.
- [Micb] MICROSOFT CORPORATION: *Web Services Specifications Index Page*. Internet Site. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/wsspecsover.asp>.
- [Wes04] WESTBRIDGE TECHNOLOGY: *Web Service Usage Survey*. Internet Publication, 2004. <http://www.westbridgetech.com/registerP.jsp?c=210>.
- [Wor] WORLD WIDE WEB CONSORTIUM: *Web Services Activity*. Internet Site. <http://www.w3.org/2002/ws/>.