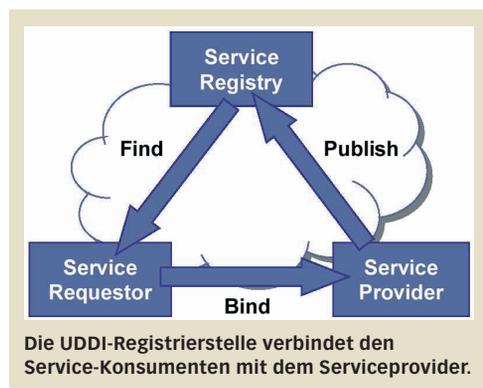


# Web Services: Eine Middl

Microsoft propagiert «Passport». Sun stemmt sich mit «SunONE» dagegen und trommelt die freie Welt zur «Liberty Alliance» zusammen. Derweil tanzt IBM elegant auf beiden Hochzeiten. Alle reden von «Web Services.<sup>TM</sup>» Doch was steht hinter der neuen Dienstleistungstechnologie? *Andreas Kapp*



Immer häufiger entwickeln sich Technologien schneller als die Vorstellungen in der Geschäftswelt, was mit den wunderbaren Errungenschaften anzufangen ist. Dies ist auch bei Web Services nicht anders. Noch immer fehlen Beispiele für plausible Geschäftsmodelle weitgehend.

## Von Maschine zu Maschine

Das ist auf den zweiten Blick weniger erstaunlich, denn Web Services sind ausschliesslich für die Maschine-zu-Maschine-Kommunikation gedacht. Web Services vernetzen Computer und Gerä-

te wie Mobiltelefone auf der Ebene der Funktionen einer Applikation. «Web Services sind als Internet-Technologie konzipiert», sagt Walter Kriha, CTO von Pythonfactory und Professor für Verteilte Systeme an der Hochschule der Medien in Stuttgart. «Im Gegensatz zu CORBA und Java J2EE sind Web Services asynchron, weniger komplex und lockerer verknüpft. Eine Zusammenarbeit zwischen unterschiedlichen Systemen und Plattformen lässt sich so leichter realisieren.»

Die Werbetrommel für ein Internet der Web-Dienste ist unterdessen auch in Chefetagen kaum mehr zu überhören. Grosse Softwarefirmen, allen voran IBM, Microsoft und SAP, basteln eifrig an zukünftigen Standards, Tools und «UDDI», den öffentlichen Registrierstellen für Web Services.

Grössere Projekte auf der Basis von Web Services sind auch in der Schweiz bereits angelaufen. So will ein Pharmakonzern sein Dokumentenverwaltungssystem im Intranet global zur Verfügung

stellen. Web Services dienen hier als Zugriffsschicht und regeln die Rechte der Benutzer und Applikationen. Realisiert wird das System kostensparend mit dem freien «SOAP::Lite for Perl».

Das Marktdatensystem «Quotes» der UBS ist vermutlich eines der ältesten Systeme mit einer Architektur, wie sie heute durch Web Services propagiert wird. Die Entscheidung für XML und HTTP und gegen CORBA fiel 1999 – aus Gründen wie Einfachheit, Betriebsstabilität und flexibleren Ausbaumöglichkeiten. Heute beantwortet die «Quotes-Engine» über 2 Millionen XML-Aufrufe pro Tag. Seit dem Start wurden das ursprünglich aus zwei Servern bestehende System auf über 20 Maschinen mit gleicher Leistung aufgestockt.

«XML und Web Services sind nicht die Lösung für alle Probleme», sagt Stefan Tramm, Senior System Engineer bei Systec. Und Andrej Vockovski, CEO von Netcetera doppelt nach: «Man kann viel mit Web Services machen. Das Problem sind nun aber der Hype und die zu grossen Erwartungen an die Technologie. Auch besteht die Gefahr, dass Web Services so komplex werden wie CORBA,

wenn man sichere Transaktionen bauen will.» Tramm und Vockovski waren für die technische Architektur des Marktdatensystems der UBS verantwortlich.

## XML ist geschwätzig

Die grösste Herausforderung für den Internet-Einsatz von Web Services sieht Stefan Tramm im Consumerbereich, wie ihn Microsoft mit .Net und die Liberty Alliance beabsichtigen. Auch sollte man die «Geschwätzigkeit» von XML nicht unterschätzen: «Ab 150 Kilobytes lohnt sich die Kompression im lokalen Netzwerk. Hier braucht es eine Kompression wie beim Videostreaming.» Der Experte für Server-Cluster prophezeit denn auch: «XML im Massengeschäft wird ein teures Geschäft. Der Einsatz von Web Services ist heute eher im Intranet zu empfehlen.»

Unterdessen sind «My Services/Hailstorm», die Web Services für das Surf-Volk, Geschichte. Microsoft hat das Projekt wegen mangelndem Interesse eingestellt. Beobachter vermuten, die Unternehmen waren nicht bereit, ihre Kundendaten von Redmond verwalten zu lassen.

## «In 18 Monaten spricht niemand mehr davon.»

Sind Web Services nur ein Hype? Ja, meint Roy T. Fielding, Chief Scientist bei Day und Vorsitzender der Apache Software Foundation. Die Netzwoche wollte wissen warum. *Interview: Andreas Kapp*

**Jeder redet über Web Services auf der Basis von XML. Was ist der Grund für diesen Hype?**

IBM und Microsoft verkaufen Web Services gemeinsam.

**Handelt es sich dabei um eine neue Technologie?**

Im Grunde sind Web Services ein verteilter Remote-Procedure-Call-Mechanismus (RPC). Im Gegensatz zu RPC kann bei Web Services der Aufrufende die Syntax des Aufrufs dynamisch erweitern. In der Praxis stellt sich aber die Frage, wie der Empfänger auf diese dynamischen Attribute reagieren soll.

**Wozu werden Web Services gebraucht?**

Web Services bringen eine gemeinsame Syntax für applikationsspezifische Programmierschnittstellen. Im Grunde bieten sie nicht mehr als CORBA und Microsoft DCOM. Der einzige Unterschied besteht darin, dass die Spezifikation in der Windows- und der UNIX-Welt gleichermaßen anwendbar ist.

**Was für Applikationen lassen sich mit Hilfe von Web Services entwickeln?**

**Welche Architektur wird dabei unterstützt?**

Die Beispiele für Web Services wie der Börsenticker beschreiben meistens Szenarien, die man besser als gewöhnliche Webseiten realisiert. Web Services können jedoch hilfreich sein für die Integra-

tion unterschiedlicher Legacy-Anwendungen im Intranet der Unternehmen. Doch Web Services sind eine antiquierte Technologie, sie wiederholen CORBA mit Hilfe von XML und sind dabei genauso schlecht im Internet oder WAN (Wide Area Network) zu gebrauchen. Denn hier sind die Komplexität der Abhängigkeiten und die Wartezeit im Netz (Latency) viel grösser.

**Welchen Vorteil bringen Web Services den Anwendern?**

Die leichtere Integration von unterschiedlichen Systemen nützt den Anwendern wenig. Web Services sind nur ein Mittel, um Schnittstellen zu beschreiben. Inhaltlich definieren sie diese



Roy T. Fielding, Architekt des «Hypertext Transfer Protocol» und Gründungsmitglied des Apache-HTTP-Server-Projekts.

Schnittstellen jedoch nicht. Anwender erwarten aber eine solche semantische Integration, eine Abstraktion und konsistente Sicht auf ihre Anwendungen.

**Was verstehen Sie unter einer modernen Web-Architektur?**

Unser Ansatz für eine Web-Architektur

# eware für das Internet?

## Web Services in Stichworten

Die Web-Services-Technologien basieren auf XML. Entwürfe zu Standards sind meist bei der W3.org oder der IETF.org deponiert.

### Suchen, Registrieren:

**Universal Description Discovery and Integration (UDDI)** ist ein globaler Verzeichnisdienst für das Publizieren und Suchen von Web Services. Öffentliche UDDI-Server im Testbetrieb haben HP, IBM, Microsoft und SAP bereits aufgestellt. Die Einträge der Verzeichnisse werden untereinander inhaltlich abgeglichen. Firmen können ihre Web Services hier eintragen.

[www.uddi.org](http://www.uddi.org)

**Web Services Inspection Language (WS-Inspection)** ergänzt UDDI und hilft, die vorhandenen Web Services einer Website zu entdecken.

[www.ibm.com/developerworks/webservices/library/ws-wsilspec.html](http://www.ibm.com/developerworks/webservices/library/ws-wsilspec.html)

### Beschreiben

**Web Services Description Language (WSDL)** (Vergleichbar mit der «Interface Definition Language» bei CORBA) spezifiziert einen Web Service. Aus der WSDL-Spezifikation kann dann der Kommunikationsteil in Java oder C# generiert werden. WSDL Version 1.1 basiert auf SOAP und HTTP, kann aber auch andere Formate und Protokolle unterstützen.

[www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl)

**Web Services Flow Language (WSFL)** von IBM beschreibt, wie verschiedene Web Services zu einem Geschäftsvorfall zusammengebunden sind. Ähnliches will Microsoft mit XLANG erreichen. Möglicherweise wird XLANG mit WSFL zusammengeführt.

<http://xml.coverpages.org/wsfl.html>

**Web Services for Remote Portals (WSRP)** bauen auf WSDL auf und erweitern Web Services um eine grafische Repräsentation (GUI) für Menschen. Ein Dienst liefert zusammen mit der Antwort eine Beschreibung mit, wie die Darstellung des Resultats im Browser, Portal oder Java-Handy aussehen soll. Ähnliches will die OASIS-Initiative «Web Services for Interactive Applications» (WSIA) erreichen.

[www.oasis-open.org](http://www.oasis-open.org)

### Verpacken, Erweiterungen

**Simple Object Access Protocol (SOAP)** ist ein XML-Format zur Beschreibung von Nachrichten-Verpackungen, die zwischen zwei Maschinen ausgetauscht werden. SOAP ist vergleichbar mit einem adressierten, aber leeren Briefumschlag. Initianten von SOAP sind IBM und Microsoft.

[www.w3.org/TR/SOAP](http://www.w3.org/TR/SOAP)

**SOAP Security Extensions (SOAP-DSIG)** definierten die digitale Unterschrift für SOAP-Nachrichten. Die Übertragung kann mit SSL gesichert werden.

[www.w3.org/TR/SOAP-dsig](http://www.w3.org/TR/SOAP-dsig)

**SOAP Messages with Attachments (SWA)** verfolgten die Verbindung von SOAP mit MIME (Multipurpose Internet Mail Extensions).

[www.w3.org/TR/SOAP-attachments](http://www.w3.org/TR/SOAP-attachments)

**Direct Internet Message Encapsulation (DIME)** spezifiziert den Transport von binären Multimedia-Inhalten (Video, Grafik, Sound) durch die «textfreundliche» Infrastruktur des Internets. Microsoft hat DIME bei der Internet Engineering Task Force zur Standardisierung vorgelegt.

[www.ietf.org](http://www.ietf.org)

### Transport

#### Hypertext Transfer Protocol (HTTP/1.1)

Ist der Internetstandard, die Grundlage des World Wide Web.

[www.w3.org/Protocols](http://www.w3.org/Protocols)

#### Reliable HTTP (HTTPR)

Will HTTP/1.1 um den verlässlichen Transport der Daten erweitern.

[www.ibm.com/developerworks/webservices/library/ws-phtt](http://www.ibm.com/developerworks/webservices/library/ws-phtt)

gründet auf einer «Representational State Transfer» (REST) genannten Architektur. Das Ziel von REST ist, eine für das Internet geeignete Abstraktion auf die Applikation zu bieten. Dabei sehe ich eine Applikation als Menge von spezifischen Daten bzw. Dokumenten. Web Services beschreiben nur die Schnittstelle zur Kontrolle und Manipulation der Daten. Das ist vergleichbar mit dem Fenster-Menü bei der Textverarbeitung. Was aber die User wirklich interessiert, ist der Abschnitt, der Titel, der Inhalt. Mit REST fokussieren wir das Ansprechen der Daten als Dokumente. Web Services dagegen beschreiben die Prozeduren und Methoden, aber nicht das Ziel ihrer Operationen.

**Das Internet ist bekannt als langsam, unsicher und unzuverlässig. Ist dies eine brauchbare Grundlage für Web Services?**

Ich würde das Internet nicht als langsam bezeichnen. Das Web wurde als Hyper-

medium mit Links zu Dokumenten und hohen Latenzzeiten entworfen. Das Web wurde so gebaut, dass es skaliert und dass Roboter wie Hacker vom Web Server entsprechend behandelt werden, ohne dass das System gleich zusammenbricht.

**Wie verhält es sich mit den Verzeichnisdiensten? Wird UDDI erfolgreich sein?**

Ich glaube nicht. Es kommt selten vor, dass jemand eine Online-Dienstleistung in sein System einbinden will, von der er nicht bereits weiss, dass sie existiert, wer sie anbietet und wie sie funktioniert. UDDI wird mit den Telefonnummern in den gelben Seiten verglichen. Doch eine Telefonnummer ist ein klar definierter Dienst. Dagegen ist eine UDDI-Beschreibung bloss die Referenz auf einen Dienst. Eine URL im Web ist wie eine Telefonnummer. Hingegen liefert mir UDDI keine Garantie, dass der Dienst auch in der Qualität verfügbar ist, die ich brau-

che. Dazu macht man üblicherweise zuerst einen Vertrag mit dem Anbieter des Dienstes. Das Web-Service-Verzeichnis von UDDI macht also nichts wirklich Nützliches.

**Werden sich Web Services durchsetzen können?**

In 18 Monaten wird niemand mehr über Web Services reden.

**Einige der Werkzeuge und Bibliotheken für Web Services kommen von der Apache Software Foundation.**

Unsere wichtigste Anliegen ist, dass Internetstandards offen sind und niemandem alleine gehören. Daher probieren wir neue Ideen frühzeitig aus. Unser Ziel ist es, interessante Technologien mit freier Software früh zu unterstützen, um so Erfahrungen zu sammeln.

**Gibt es einen Zusammenhang zwischen Web Services und dem «Semantic Web», wie es der Erfinder des Webs Tim Berners-Lee fordert?**

Das sind sehr unterschiedliche Konzepte. Web Services verwenden einen Tunnel durch die Firewall mittels HTTP-Protokoll. Dagegen wollen wir beim Semantic Web das HTTP-Protokoll und die URL erweitern. Schlussendlich konkurrieren sich die beiden Ansätze. Web Services haben nichts mit dem Web gemeinsam, ausser, dass sie durch einen HTTP-Tunnel kommunizieren.

#### links

**Roy T. Fielding, Richard N. Taylor: Principled Design of a Modern Web Architecture**

[www.cs.virginia.edu/~cs650/assignments/papers/p407-fielding.pdf](http://www.cs.virginia.edu/~cs650/assignments/papers/p407-fielding.pdf)

**Semantic Web**

[www.w3.org/2001/sw/](http://www.w3.org/2001/sw/)

**Apache Foundation**

[www.apache.org](http://www.apache.org)