

# Nutzen und Herausforderungen einer Enterprise Services Architecture

Claus von Riegen

Platform Ecosystem Industry Standards  
SAP AG  
Neurottstrasse 16  
D-69190 Walldorf  
claus.von.riegen@sap.com

**Abstract:** Web Services Standards haben sich im Laufe der vergangenen Jahre stark weiterentwickelt, unter anderem aber auch viel „Hype“, d.h. überzogene Erwartungen, verursacht. Zwar gibt es für Web Services Technologie breite Unterstützung in der Softwareindustrie und der Nutzen bei der Integration heterogener Softwarelandschaften ist unbestritten. Allerdings ergeben sich insbesondere beim Einsatz in unternehmenskritischen Geschäftsanwendungen Anforderungen wie Skalierbarkeit, Verlässlichkeit, Durchsatz, Sicherheit aber auch Änderbarkeit, die noch adressiert werden müssen. Diese Arbeit orientiert sich an einem konkreten Geschäftsszenario und erläutert anhand daraus resultierender Anforderungen, wie diese durch existierende bzw. zukünftig verfügbare Web Services Technologien abgedeckt werden können. Damit stellt diese Arbeit eine Grundlage für die bedarfsorientierte Entscheidung für die Nutzung von Web Services Technologie in unternehmensweiten Integrationsprojekten dar.

## 1 Interoperabilität von Softwareanwendungen

Web Services werden oft als Allheilmittel für die Lösung aller Probleme der Integration von Softwareanwendungen angesehen. Diesen Übererwartungen steht oft eine zögerliche Haltung gegenüber, dass die Technologie noch nicht ausgereift ist und unternehmenskritischen Anforderungen nicht genügt. Eine Überwindung dieses Dilemmas ist oft nicht möglich, da Softwareanwender und Integrationsexperten oft unterschiedliche „Sprachen“ sprechen, d.h. zuwenig Erfahrung in der Abbildung der fachlichen Anforderungen auf die Möglichkeiten der Web Services Technologie existiert.

Letztlich führt an Web Services Technologie kein Weg vorbei. Sie ist Grundlage für die technische Interoperabilität von Softwareanwendungen und in praktisch jeder Softwareplattform verfügbar. Das folgende Kapitel soll anhand eines konkreten betriebswirtschaftlichen Szenarios anschaulich demonstrieren, wie eine Reihe von fachlichen Anforderungen mit Mitteln der Web Services Technologie abgedeckt werden können.

## 2 Eine Geschäftsanwendung

Das so genannte Vendor-Managed Inventory Szenario (VMI) beschreibt eine Vereinbarung zwischen dem Hersteller eines Produktes und dessen Abnehmern, dass a) der Hersteller für die automatische Nachschublieferung der Lager des Abnehmers verantwortlich ist und b) dafür der Abnehmer dem Hersteller laufend Informationen über aktuelle Bestände, Abverkäufe und offene Bestellungen zukommen lassen muss. Letztlich wird dadurch der normale Bestellprozess überflüssig – der Hersteller liefert aufgrund der Vereinbarung Ware an die Abnehmer, sobald seine Nachschubplanung aufgrund der vorliegenden Daten von den Abnehmern dies vorschlägt. Abbildung 1 gibt einen Überblick über die verschiedenen beteiligten Rollen im VMI-Szenario. Die dicken Linien repräsentieren Integrationsverbindungen, wobei die ange deuteten Unterbrechungen die Seite des jeweiligen Serviceanbieters darstellen. Zu beachten ist ferner, dass der Abnehmer sowohl unternehmensinterne (zwischen den Lagern bzw. Filialen und der Zentrale) als auch unternehmensübergreifende (zu den Herstellern) Integrationsverbindungen realisieren muss.

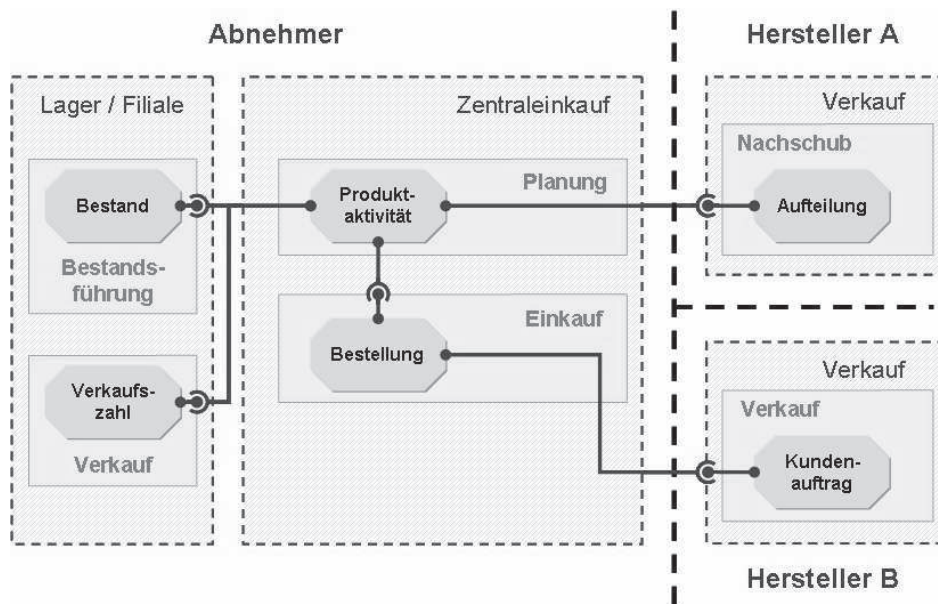


Abbildung 1: Geschäftsszenario „Vendor-Managed Inventory“

## **2.1 Transparenz des Geschäftsprozesses**

Die verschiedenen Schritte im VMI-Szenario müssen in einer bestimmten Reihenfolge durchlaufen werden, damit die vertraglichen Vereinbarungen von den Herstellern gewährleistet werden können. So müssen z.B. zunächst alle Bestandsinformationen und Abverkaufsdaten aus allen Lagern und Filialen gesammelt und mit den Informationen über offene Bestellungen, die in der Zentrale vorliegen, aggregiert werden, bevor die daraus entstandene, so genannte Produktaktivitätsbenachrichtigung an die jeweiligen Hersteller übermittelt werden kann. Das heißt, dass zunächst der Geschäftsprozess beschrieben werden muss, damit die technische Infrastruktur auch in der Lage ist, diesen Prozess zu steuern, zu überwachen und auf Ausnahmesituationen reagieren zu können.

Mit BPEL4WS [BPEL1] gibt es einen Standard, der die Notwendigkeit der Transparenz von Web-Services-basierten Geschäftsprozessen adressiert. BPEL4WS kann sowohl für unternehmensinterne als auch unternehmensübergreifende Prozesse verwendet werden und ist nahtlos mit anderen Web Services Beschreibungssprachen (WSDL [WSDL], UDDI [UDDI1]) integriert. Die Prozessdefinition kann sowohl aus vorhandenen Web Services aufgebaut werden und beschreibt damit deren Orchestrierung (Zusammenspiel), als auch der Generierung von Web Service Schnittstellen in einem gewünschten System dienen, die dann natürlich erst implementiert werden müssen.

BPEL4WS ist bereits vielfach implementiert und befindet sich in der Standardisierung bei OASIS [BPEL2].

## **2.2 Anpassungsfähigkeit des Unternehmens**

Viele interne und externe Ereignisse erfordern die zeitnahe und konsistente Anpassungsfähigkeit eines Unternehmens.

### **2.2.1 Interne Ereignisse**

Ein typisches internes Ereignis im Falle eines Einzelhandelsunternehmens ist das Eröffnen oder Schließen einzelner Filialen. Da dies immer wieder vorkommt – die Expansion in einem bestimmten Markt kann zum Beispiel Teil der Unternehmensstrategie sein – müssen die Prozesse zwischen der Zentrale und den Filialen möglichst ohne großen Aufwand auf solch geänderte Situationen angepasst werden können.

Eine Möglichkeit dazu bietet die Verwendung von UDDI als Verzeichnis der im Unternehmen verwendeten Dienste. Die einzelnen Filialen sind dafür verantwortlich, ihre Dienste über Abverkaufsdaten oder Bestandsinformationen im unternehmensinternen UDDI zu registrieren. Wird der VMI-Prozess einmalig so erweitert, dass er neu hinzugefügte oder geänderte Dienste selbsttätig im unternehmensinternen UDDI nachschlägt, kann der Prozess automatisch auf entsprechende Änderungen reagieren und immer die jeweils aktuelle Auswahl an Diensten für die Aggregation der Daten integrieren.

UDDI ist bereits vielfach implementiert und ist von OASIS in den Versionen 2.0 [UDDI2] und 3.0 [UDDI3] standardisiert.

### 2.2.2 Externe Ereignisse

Ein typisches externes Ereignis ist die Änderung der Frequenz, in der der Abnehmer dem Hersteller Produktaktivitätsbenachrichtigungen zu übermitteln hat. Eine bestimmte Frequenz ist notwendig, damit der Hersteller rechtzeitig Nachschub liefern kann und ist abhängig von der Art des Produktes. So müssen z.B. Informationen über oft verkaufte Produkte auch öfter ausgetauscht werden. Die Frequenz kann sich aufgrund bestimmter Einflussgrößen (Saison, Promotion, etc.) durchaus ändern und muss somit auch im VMI-Szenario der Abnehmer adressiert werden.

Eine Möglichkeit zur Beschreibung und zum Austausch solcher Konfigurationsdaten ist WS-Policy [WS-P], eine Beschreibungssprache für die technischen Eigenschaften von Web Services. WS-Policy erlaubt die Entwicklung eigener Konfigurationsdatentypen, so genannter „policy assertions“ und ist nahtlos mit WSDL und UDDI integriert, um die gewünschte oder geforderte Konfiguration mit entsprechenden Web Services Artefakten (Web Service, Operation, Nachricht) zu assoziieren bzw. auffindbar zu machen.

WS-Policy wird zurzeit von einer Reihe von Unternehmen entwickelt und wird in naher Zukunft zur Standardisierung eingereicht. Erste Prototypen sind zu Testzwecken entwickelt worden.

## 2.3 Sichere Nachrichtenübertragung

Insbesondere bei der unternehmensübergreifenden Integration ergeben sich Anforderungen an die vertrauliche und unversehrte Übertragung von Nachrichten.

Vertrauliche Übertragung bedeutet, dass der Zugriff auf die Daten Dritten nicht gewährt werden darf. Eine geeignete Maßnahme dafür ist die Verschlüsselung der Daten auf der Absenderseite und die Entschlüsselung auf der Empfängerseite. Da Web Services auf der Versendung von XML Dokumenten basieren, ist die Verwendung von XML Encryption [XML-E] die richtige Technologie dafür.

Unversehrte Übertragung bedeutet, dass die Daten nachvollziehbar unverändert übertragen werden und die Authentizität des Absenders gesichert ist. Eine geeignete Maßnahme dafür ist die digitale Signierung der Daten auf der Absenderseite und die Validierung der Signatur auf der Empfängerseite. Die Verwendung von XML Digital Signature [XML-D] ist die richtige Technologie in diesem Fall.

Die Verwendung beider Technologien (XML Encryption und XML Digital Signature) in SOAP [SOAP1], [SOAP2] wird durch den OASIS Standard Web Services Security [WS-S] beschrieben. Web Services Security ist bereits von vielen Herstellern implementiert worden und diverse Interoperabilitätstests wurden durchgeführt.

## **2.4 Verlässliche Nachrichtenübertragung**

Je nach fachlichem Szenario muss die Übertragung der Nachrichten mit einer bestimmten Verlässlichkeit erfolgen. Beispielsweise muss für Benachrichtigungen wie die Produktaktivitätsbenachrichtigung gewährleistet sein, dass sie mindestens einmal übertragen werden. Eine mehrfache Übertragung wäre zwar überflüssig, würde aber zu keinem fachlichen Fehler führen. Für geschäftliche Transaktionen wie die Bestellung muss dagegen zusätzlich gewährleistet sein, dass sie höchstens einmal übertragen werden, d.h. im Endeffekt genau einmal. Eine mehrfache Übertragung würde in diesem Fall zu einem fachlichen Fehler führen, da die doppelte Menge an Waren bestellt werden würde.

Anforderungen dieser Art, d.h. an verlässliche Nachrichtenübertragung, werden von der WS-ReliableMessaging Spezifikation [WS-R] adressiert. WS-ReliableMessaging ermöglicht die garantierte Übertragung, Eliminierung von Duplikaten und Einhaltung einer bestimmten Nachrichtenreihenfolge - auch und vor allen Dingen auf Basis von sonst nicht verlässlichen Transportprotokollen, wie z.B. HTTP.

WS-ReliableMessaging wurde von einer Reihe von Unternehmen entwickelt und im April 2005 bei OASIS zur Standardisierung eingereicht. Erste Prototypen sind zu Testzwecken entwickelt worden.

## **3 Fazit**

Für den nutzbringenden Einsatz von Web Services Technologie ist es erforderlich, sowohl frühzeitig Erfahrungen mit der Technologie zu machen, als auch den Nutzen und die Verwendung einzelner Web Services Standards zu kennen und für die Integration von Softwareanwendungen ausprägen. Die oben beschriebenen Anforderungen an Transparenz von Geschäftsprozessen, Anpassungsfähigkeit des Unternehmens sowie sichere und verlässliche Nachrichtenübertragung werden heute schon von Web Services Standards abgedeckt, auch wenn die vollständige und interoperable Umsetzung von verschiedenen Herstellern zum Teil noch sichergestellt werden muss.

Web Services Technologie an sich bringt noch keinen Gewinn. Erst die dadurch erreichte Interoperabilität kann - dafür allerdings entscheidend - zur Senkung der Integrationskosten beitragen. Ein Einsatz von Web Services Technologie in konkreten Projekten sollte insofern an Kriterien wie a) der Kosteneinsparung, beispielsweise durch Fehlervermeidung (z.B. durch Erfassung von manuell übertragenen Daten) und Beschleunigung von Geschäftsprozessen (z.B. durch immer aktuelle Daten) und b) der Nutzungshäufigkeit, beispielsweise anhand der Anzahl der Verwender und der Häufigkeit des Aufrufs, gemessen werden. Erste Erfahrungen sollten insofern anhand von Projekten gesammelt werden, die einen schnellen Return-On-Investment (ROI) erwarten lassen.

Mit der Enterprise Services Architecture hat die SAP ihre Vorstellungen präzisiert, wie Geschäftsanwendungen auf einer service-orientierten Architektur basieren können. Eine Grundlage dafür ist die breite Unterstützung von Web Services Standards in SAP NetWeaver, der von SAP angebotenen Softwareplattform für die Entwicklung und Integration von Geschäftsanwendungen.

## Literaturverzeichnis

- [BPEL1] Andrews, T. et al: Business Process Execution Language for Web Services, <http://ifr.sap.com/bpel4ws/index.html>.
- [BPEL2] OASIS Web Services Business Process Execution Language (WSBPEL) TC, [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsbpel](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel).
- [SOAP1] Box, D. et al: Simple Object Access Protocol (SOAP) 1.1, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [SOAP2] Gudgin, M. et al: SOAP Version 1.2 Part 1: Messaging Framework, <http://www.w3.org/TR/soap12-part1/>
- [UDDI1] von Riegen, C.; Trickovic, I.: Using BPEL4WS in a UDDI registry, <http://www.oasis-open.org/committees/uddi-spec/doc/tns.htm#bpel>.
- [UDDI2] Bellwood, T.: UDDI Version 2.04 API Specification, <http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm>.
- [UDDI3] Clement, L.; Hatley, A.; von Riegen, C.; Rogers, T.; UDDI Version 3.0.2, <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>.
- [WSDL] Christensen, E; Curbera, F.; Meredith, G.; Weerawarana, S.: Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/wsdl>
- [WS-P] Bajaj, S. et al: Web Services Policy Framework (WS-Policy), <http://ifr.sap.com/ws-policy/index.html>.
- [WS-R] Bilorusets, R. et al: Web Services Reliable Messaging Protocol (WS-ReliableMessaging), <http://specs.xmlsoap.org/ws/2005/02/rm/ws-reliablemessaging.pdf>.
- [WS-S] Nadalin, T. et al: Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>.
- [XMLD] Eastlake, D.; Reagle, J.; Solo, D.: XML-Signature Syntax and Processing, <http://www.w3.org/TR/xmlsig-core/>.
- [XMLENC] Eastlake, D.; Reagle, J.: XML Encryption Syntax and Processing, <http://www.w3.org/TR/xmlenc-core/>.