

Technische Universität München

Forschungs- und Lehrereinheit Informatik III
Prof. R. Bayer Ph.D., Prof. Dr. D. Kossmann

Hauptseminar Informatik im Sommersemester 2003

Web Services

Anwendungsszenarien für Web Services

Referent in:	Katharina Brendebach
Ansprechpartnerin:	A. Reiser
Betreuer:	Dipl.-Inf. Karl Hahn
Abgabetermin:	10. April 2003
Vortragsdatum:	17. April 2003

Inhaltsverzeichnis

Seite

1	Einleitung.....	2
1.1	Unternehmenswandel hervorgerufen durch die Entwicklung des Internets.....	2
1.2	Vorteile der Web Services.....	3
1.3	Kriterien zur Verdeutlichung das Web Services eingesetzt werden sollten.....	3
2	Was ist ein Web Service.....	3
2.1	Zwiebelschalenmodell.....	3
2.2	Vieldeutigkeit von Web Services.....	5
2.3	Web Service Architecture Working Group at W3C.....	6
3	Web Service Architektur.....	7
3.1	Anforderungen einer Web Service Architektur.....	7
3.2	Basic Web Service Architektur.....	7
3.3	Erweiterte Web Service Architektur.....	9
3.3.1	Besondere Eigenschaften.....	9
3.3.2	Flows.....	10
3.3.2.1	Peer to Peer.....	10
3.3.2.2	Direkte Wechselbeziehungen.....	11
3.3.2.3	Zwischenvermittler.....	11
3.3.2.4	Nachrichtenversendung in nur einer Richtung.....	12
3.3.3	Integration zweier Nachrichtenpartner.....	12
4	Methoden zur Architektur Analyse.....	13
4.1	Critical Success Factor.....	13
4.2	Web Service Architecture Usage Scenarios.....	13
4.2.1	Nachrichten Austausch Modelle.....	13
4.2.1.1	Fire and forget to single receiver.....	13
4.2.1.2	Anfrage und Antwort.....	13
4.2.1.3	Multiple asynchronous responses.....	14
4.2.1.4	Zwischenhändler.....	14
4.2.1.5	Anfrage mit verschlüsseltem Inhalt.....	14
4.2.2	Event basierender Nachrichtenaustausch.....	14
4.2.2.1	Event Mitteilung.....	14
4.2.3	System Nachrichten.....	14
4.2.4	Service Beschreibung.....	14
4.2.4.1	Service Metadaten.....	14
4.2.4.2	Klassifizierendes System für Operationen.....	14
4.2.5	Finden von Web Services.....	15
4.2.5.1	Adress- basiertes Finden.....	15
4.2.5.2	Register- basierende Entdeckung.....	15
5	Realisierungskonzepte für Web Services.....	16
6	RosettaNet.....	16
7	Abschluss.....	21
8	Literaturverzeichnis.....	22

1 Einleitung

1.1 Unternehmenswandel hervorgerufen durch die Entwicklung des Internets

Die Programmentwicklung für das Internet hat in der letzten Zeit sehr große Ausmaße angenommen. Programme, Software- Komponenten, Dienstleistungen und auch Geschäftsbeziehungen werden immer mehr in das Internet verlagert.

Für ein Unternehmen beruht ein Geschäft immer auf Transaktionen, deren anfallenden Kosten die Geschäftsstruktur prägen. Häufig erklärt die Vermeidung der Kosten für die Abwicklung von Transaktionen über den Markt die Existenz von Firmen. Da das Internet die Kosten für die Geschäftsvorgänge erheblich senken kann, werden entsprechende Technologien benötigt, um diesen Aspekt des Internets auszuschöpfen.

Derzeit entsteht eine neue elektronische Infrastruktur zur Durchführung von Geschäftsprozessen, da internetbasierte Informationssysteme die betrieblichen Prozesse grundlegend verändern. Das bedeutet für die Unternehmen, dass sie ihre eigentlichen Geschäftsfelder ganz neu abstecken müssen. Was in der Vergangenheit als Kernkompetenz herausgestellt wurde, könnte durchaus als Dienstleistung von Dritten angemietet werden. Die Unternehmen müssen entscheiden, worauf sie sich wirklich konzentrieren, und welche Dienstleistungen sie über das Internet organisieren möchten. Den Firmen wird zur Ausgestaltung ihrer IT- Portfolios lose miteinander verknüpfte IT- Anwendungen aus dem Angebot Dritter angeboten. Somit können Organisationen und Firmen Web Services für Tätigkeiten in Anspruch nehmen, die sie nicht selbst ausführen möchten. Das bedeutet auch, dass Unternehmen nicht mehr die Hände gebunden sind wenn Technologieinvestitionen anstehen, da sie z.B. die gegebenen Ressourcen flexibler und vielseitiger nutzen können, z.B. Vereinheitlichung bzw. Integration unternehmensweiter Datenbestände.

Es entsteht die Notwendigkeit, dass verschiedene Software Applikationen die auf verschiedenen Plattformen laufen mit einander kommunizieren. Das Thema unseres Hauptseminars „Web Services“ beschäftigt sich nun mit einem Ansatz, wie konkrete Web Services aussehen müssen, die einem Unternehmen die oben genannte Hilfestellung in ihren Dienstleistungen erbringen können.

Wie wir später noch feststellen werden, ist der Begriff Web Service auf sehr unterschiedliche Weise zu definieren. Um bereits zu Beginn Missverständnissen aus dem Weg zu gehen, möchte ich eine Definition der W3C Web Services Architecture Working Group als Grundlage meines Vortrages festlegen:

“A Web service is a software system identified by a URI [RFC 2396], whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web Service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols.”

Die Begrifflichkeit, der Nutzen und die Anforderungen von Web Services soll im Folgenden näher erläutert werden.

1.2 Vorteile der Web Services

Die Einführung eines Web Services kann einem Unternehmen auf vielfältige Weise zu Gute kommen:

- erleichterte IT- Integration: Web Services sind plattformunabhängig. Sie können sehr einfach mit den bereits im Unternehmen vorhandenen Anwendungen kombiniert werden. Die Abstraktion die durch die Standardbasierte Schnittstelle bereitgestellt wird, macht keinen Unterschied, ob der Service z.B. in Java und der Browser in C++ geschrieben wurde. Genau diese Eigenschaft zeichnet die Web Services besonders aus, da es Unstimmigkeiten zwischen verschiedenen Systemen hervorragend ausgleichen kann
- geringe IT- Ausgaben: Der Gedanke der Web Service Architektur ist ihre einfache Implementation d.h. Realisierung. Langfristige Einarbeitungs- und Beratungsleistungen sind somit nicht erforderlich
- Flexibilität: Web Services sind offen für neue Funktionen. Sie sind trotz häufiger Änderungen von Geschäftsprozessen flexibel.

1.3 Kriterien zur Verdeutlichung das Web Services eingesetzt werden sollten

- in den Unternehmensabläufen herrscht viel Bewegung, so dass der Aktualität von Daten eine hohe Bedeutung für Planung und Ausführung zukommt (z.B. Materialflüsse durch eine Lieferkette)
- Anwender sind geographisch verstreut und Anwendungen können nicht sinnvoll über LAN/ WAN eingerichtet werden (z.B. Buchen von Flugtickets)
- Ein einzelner Anwender muss mit einer Reihe von Systemen zusammenarbeiten

2 Was ist ein Web Service

2.1 Zwiebschalenmodell

Es folgt zunächst eine anschauliche Erläuterung in Form eines Zwiebschalenmodells, unter Erläuterung der für uns wichtigsten Begriffe und Standards.

Schicht 1 und 2

Ein Web Service kann zunächst als Dienstleistung anerkannt werden, deren Besonderheit in der Zugreifbarkeit über das World Wide Web liegt. Dabei greift der Web Service auf Standards von Web Techniken zurück sowohl hinsichtlich der physischen Hardwareinfrastruktur als auch den Internet Protokollen wie z.B. TCP/IP.

Um diesen Kern herum, werden nun schichtweise weitere Hilfsmittel angeordnet, um den Web Service im Internet für den Konsumenten zu ermöglichen.

Schicht 3: Inhalt

Zunächst wird ein Format gesucht, um den Inhalt der vermittelt werden soll standardisiert ausdrücken zu können. Hier bietet sich unter anderem das XML Format an, deren Bedeutung in der heutigen Datenwelt von Tag zu Tag wächst. Dieses Format bietet dem Benutzer einen großen Freiraum, die Daten können mit seiner Hilfe semi- strukturiert dargestellt werden, d.h. es wird nicht nur der Inhalt der Daten festgehalten, sondern auch die Struktur der Darstellung enthält eine semantische Bedeutung. Ein weiterer Vorteil von XML liegt in dem

Verbreitungsgrad und den damit einhergehenden Auswirkungen wie Werkzeugunterstützung, Erfahrungsberichten und anderes bereits erforsches Wissen.

Schicht 4: Transport

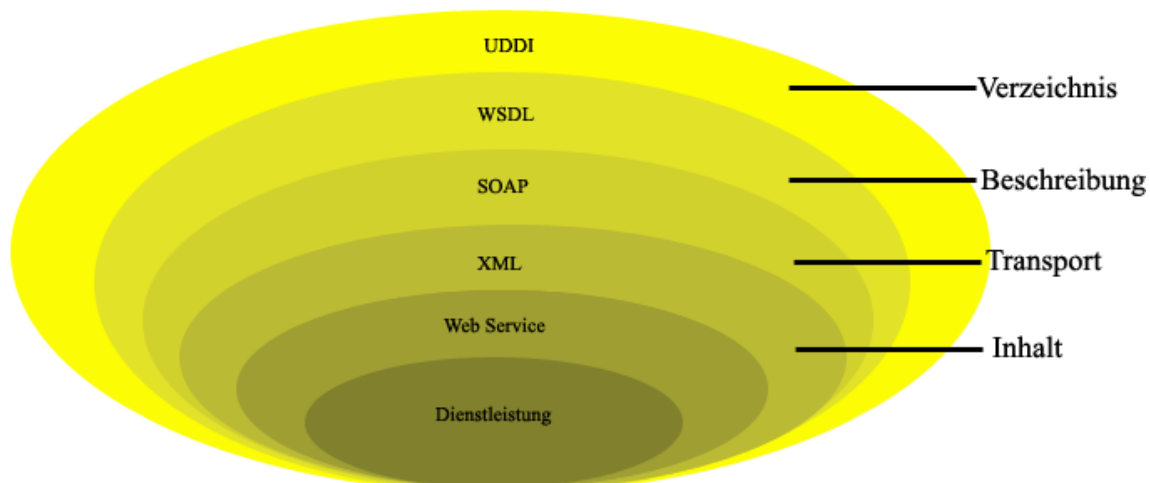
Die nächste Schicht befasst sich mit dem Kommunikationsprozess, denn die Daten sollen von verschiedenen Orten zu anderen Orten transportiert werden. Ein Standard, der sich mit dem Interaktionsablauf zwischen zwei Handlungspartnern befasst, ist der Mechanismus zur Abwicklung entfernter Methodenaufrufe sog. Remote Procedure Calls (RPC). Ein weiterer Ansatz ist das Simple Object Access Protocol (SOAP), das von XML als Inhaltsformat ausgeht und sich somit hier sehr anbietet, da es die standardisierten Kommunikationsmechanismen abgestimmten Inhalte in einem ebenfalls abgestimmten Format transportiert.

Schicht 5: Beschreibung

Diese Schicht befasst sich mit der Beschreibung des Dienstes, damit sich der Nutzer genau über das Angebot und den Abwicklungsmodalitäten (Aufrufkonventionen, zu übergebende Parameter) informieren kann. Die Web Service Description Language (WSDL) ist eine solche Beschreibungssprache, deren Konzeption sich an Darstellungsformen zur Beschreibung technischer Schnittstellen orientiert.

Schicht 6: Verzeichnis

Als letzten wichtigen zu realisierenden Aspekt ist die Ermittlung des für den potentiellen Nutzer interessantesten Diensts aus dem Angebot im Web. Hier schafft der Ansatz der Universal Service Description Discovery and Intergration (USDDI) Hilfe. Dieser Dienst ist zu vergleichen mit dem bekannten Telefonbuch Gelbe Seiten. Der Grundgedanke liegt darin, die Angebote die im UDDI Dienst eingetragen wurden zu klassifizieren und mit Hilfe einer Reihe vordefinierter Zugriffsroutinen anzubieten.



Zusammenfassend lässt sich daher ein Web-Service als Komponente auffassen, die ihre Funktionalität über eine öffentliche Schnittstelle für viele Benutzer oder Organisationen anbietet. Komplexe Dienste lassen sich durch geschickte Kombination aus Basisdiensten zusammen bauen.

2.2 Vieldeutigkeit von Web Services

Wenn man sich jedoch in der Literatur und im WWW- Angebot auf die Suche nach Web Services macht, wird man feststellen, dass man eine Vielzahl an Definitionen und Beschreibungen findet, die dem Web Service immer einen anderen Schwerpunkt zuschreiben. Hier ein kleiner Einblick in die Definitionen eines Web Service durch bekannt Firmen:

Intel

Web services encompass a vision of a fully integrated computing network that include PCs, servers, handheld devices, programs, applications and network equipment, all working together. This network can perform distributed computation with the best-matched device for the task and deliver the information on a timely basis in the form needed by the user.

Quelle:

<http://www.webservices.org/index.php/article/articleprint/113/-/1/24/?PHPSESSID=24bd4ae1c6291d93bfc3192e05f37b28>

SUN

A Web service is, simply put, application functionality made available on the World Wide Web. A Web service consists of a network-accessible service, plus a formal description of how to connect to and use the service. The language for formal description of a Web service is an application of XML. A Web service description defines a contract for how another system can access the service for data, or in order to get something done. Development tools, or even autonomous software agents, can automatically discover and bind existing Web services into new applications, based on the description of the service.

Quelle:

<http://developer.java.sun.com/developer/technicalArticles/Security/xkms/index.html>

Microsoft

A Web Service is a unit of application logic providing data and services to other applications. Applications access Web Services via ubiquitous Web protocols and data formats such as HTTP, XML, and SOAP, with no need to worry about how each Web Service is implemented. Web Services combine the best aspects of component-based development and the Web, and are a cornerstone of the Microsoft .NET programming model.

Quelle:

<http://developer.java.sun.com/developer/technicalArticles/Security/xkms/index.html>

Man erkennt, dass sich viele Firmen mit dem Thema Web Service auseinandersetzen. Das bestätigt auch die Aussage eines Reports von Dietmar Mueller im ZDFNet vom 17. Dezember 2002:

„Web Services“ waren das Thema des Jahres 2002 und werden auch in den kommenden Monaten den Ton in der Informationstechnologie angeben. Alle Beteiligten beteuern, dies sei durchaus berechtigt, da Web Services das Potential hätten, in den nächsten Jahren die informationstechnische Umwälzung von Geschäftssystemen zu dominieren...“

Doch eben diese Aussage lässt auf ein damit einhergehendes Problem schließen: Kompatibilität. So kann man in einem anderen Bericht von Ulrike Ostler im ZDFNet vom 20. Dezember 2002 lesen:

„Web Services sorgen zeitweise eher für Negativ- Schlagzeilen. Reihenweise lassen Analysten die Vision von standardisierten Komponenten sterben, die sich nahtlos, per Web

und auf Abruf zu Anwendungen zusammenfügen. Von Sicherheitslücken und fehlenden Lösungen ist die Rede. ..“

Es wird deutlich, dass sich Web Services nur bewähren und durchsetzen können, wenn man versucht Standards und Richtlinien aufzustellen, um das Erstellen von Web Services zu ermöglichen und zu erleichtern. Somit möchte ich mich im Folgenden mit einer Organisation befassen, die den Begriff und die Technologie des Web Services genau beschreibt.

2.3 Web Service Architecture Working Group at W3C

Die Aktivitäten der Web Service Architecture Working Group at W3C beschäftigen sie damit, Standards für Web Services zu erarbeiten, so dass es ermöglicht wird sein ganzes Potential auszuschöpfen. Wie wir schon aus der Vielfalt der Definitionen für den Web Service erkennen können, ist es unvermeidbar einen Standard für Web Service zu erstellen, um Kompatibilität und Erweiterbarkeit der Applikationen zu ermöglichen, die miteinander kommunizieren sollen. Es liegt auf der Hand, dass der Einsatz einer neuen Idee (hier dem Web Service) erst dann sinnvoll ist, wenn sich dieser Ansatz auch durchsetzen kann. Er muss durch seinen Nutzen und somit durch seine Verbreitung überzeugen. Ziel ist eine Standard Architektur für Web Services zu entwickeln und beizubehalten bzw. zu erweitern: the standard reference architecture.

Da es eine große Anzahl von variierenden und oft scheinbar inkonsistente Motivationen für den Gebrauch von Web Service gibt, wird nun nochmals der Kern des Web Services für die Architecture Working Group dargelegt:

“A Web service is a software system identified by a URI [RFC 2396], whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web Service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols.”

Man stellt fest, dass diese Definition sehr großzügig ausgelegt ist. Es ist noch darauf hinzuweisen, dass die Mechanismen wie SOAP und WSDL nicht zwingender Maßen für einen Web Service benötigt werden, jedoch an dieser Stelle eingesetzt werden, da sie bisher die verbreiteten und best bewährten Mechanismen sind.

Mit Hilfe der Veröffentlichungen und Leitsätze der Web Service Architecture Working Group möchte ich im folgenden Web Services in Hinblick dessen Architektur und Einsatz näher beleuchten.

3 Web Service Architektur (WSA)

3.1 Anforderungen einer Web Services Architektur

Kompatibilität: Die WSA sollte die Entwicklung von vollständig kompatiblen Web Services in vielen Umgebungen ermöglichen

Funktionssicherheit/ Zuverlässigkeit: Die WSA muss über einen langen Zeitraum betriebssicher und stabil sein

Integration im World Wide Web: Die WSA muss mit den derzeitigen und den zukünftigen Entwicklungen des World Wide Web konsistent sein

Sicherheit: Die WSA muss eine sichere Umgebung für online Prozesse darstellen

Skalierbarkeit und Erweiterbarkeit: Die WSA muss Implementationen ermöglichen, die skalierbar und erweiterbar sind d.h. Millionen von Nutzern, Rechnern und Diensten

Benutzerfreundlichkeit/ „Web-friendly“: Die WSA muss mit den Bedürfnissen der Nutzer übereinstimmen, sie soll mit allen Web Trends mitgehen

Management: Die WSA muss eine kontrollierbare und verantwortliche Umgebung für Web Services Operationen gewährleisten

3.2 Basic Web Service Architektur

Die Basis Architektur muss im Stande sein folgende drei Aspekte zu realisieren, die wir bereits im Zwiebelschalenmodell kennen gelernt haben:

- Nachrichtenaustausch
- Web Service Beschreibung
- Veröffentlichen und entdecken von Web Service Beschreibungen

Die Basis Architektur definiert eine Interaktion zwischen Software Agenten (Komponenten, Module) in Form eines Nachrichtenaustausches zwischen einem Service Auftraggeber und einem Service Anbieter. **Service Auftraggeber** sind Software Agenten, die die Durchführung eines Services anfordern um benötigte Informationen zu beschaffen. Der Auftraggeber initiiert die Interaktion mit einem Service. Die Rolle des Auftraggebers kann ein Browser, ein Programm oder ein anderer Web Service einnehmen. **Service Anbieter** sind Software Agenten, die sozusagen die Besitzer des Services sind und die den Gebrauch des Services anbieten.

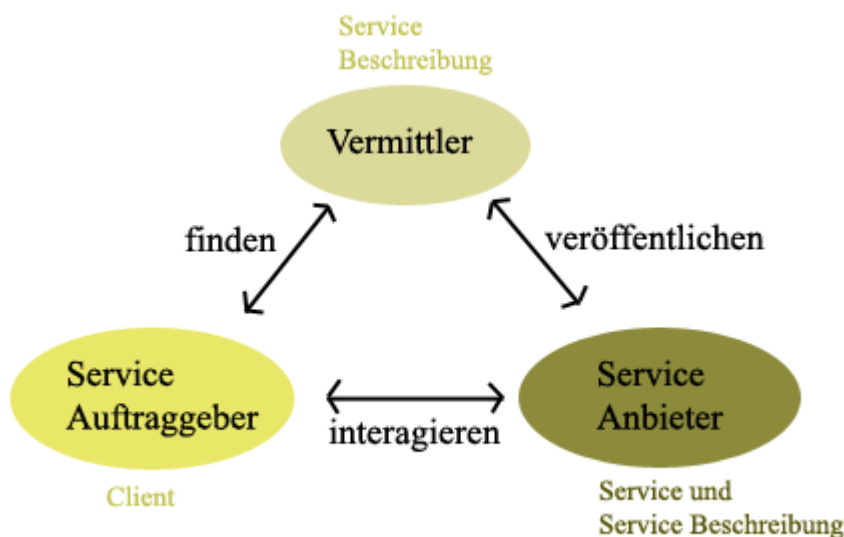
Agenten können also sowohl Auftraggeber als auch Anbieter sein, wobei der Anbieter dafür verantwortlich ist seinen Service anzubieten und der Auftraggeber in der Lage sein muss diese Beschreibungen des Services zu finden.

Die Basis Architektur modelliert die Interaktion zwischen drei Komponenten: dem Service Anbieter, dem Service Vermittler und dem Service Auftraggeber. Diese Interaktion umfasst das Veröffentlichen, Finden und Binden von Operationen. Der **Service Vermittler** ist dabei eine Sammlung von Service Beschreibungen, ein Dienst bei dem der Service Anbieter seinen

Service registrieren lassen kann. Dem Vermittler wird nur eine optionale Rolle zugesprochen, da der Service Anbieter seine Service Beschreibung auch direkt an den Service Auftraggeber senden kann.

Beispiel für ein typisches Szenario:

- Der Service Anbieter bietet eine netzwerkfähiges Software Modul (Implementierung eines Web Services) an
- Der Service Anbieter definiert über dieses Modul eine Beschreibung die den Datentyp und die Daten- Struktur sowie die Nachrichten- Austausch- Formate beschreibt. Zudem wird die Adresse des Anbieters im Netzwerk angegeben, denn der eindeutige Identifikator des Web Dienstes in seiner Umgebung ist eine URL (Uniform Resource Locator)). Die Beschreibung kann darüber hinaus auch weitere Meta Daten beinhalten, um die Nutzung durch den Service Auftraggeber zu unterstützen. Meist ist die Beschreibung mit Hilfe von XML modelliert
- Der Service Anbieter teilt diese Beschreibung dem Service Vermittler mit
- Der Service Auftraggeber benutzt eine find- Operation um diese Beschreibung lokal oder über den Service Vermittler zu bekommen und benutzt die Beschreibung um sich an das Software Modul zu binden und die Service Implementation in Anspruch zu nehmen.
- die Kommunikation zwischen Service Auftraggeber und Service Anbieter erfolgt durch ein oder mehrere Nachrichten- Austausche



Service Orientierte Architektur

Zwischen Auftraggeber und Anbieter können mehrere Zwischenhändler liegen, die jedoch den Inhalt des Nachrichtenaustausches nicht beeinflussen können. Die Rolle der Zwischenhändler liegt vielmehr in den ihnen zugeteilten Aufgaben wie z.B. Routing, Sicherheit oder Management.

Basis Web Service Architektur **Komponenten** sind typischerweise in XML Applikationen definiert, die XML zur Nachrichtendefinition und Struktur und HTTP für den Transport benutzen.

Eine **Nachricht** ist als ein Konstrukt definiert, das möglicherweise Headers mit zusätzlichen Daten enthält. Ein **Header** kann Informationen über die Funktionalität eines Web Services enthalten (Sicherheit, Routing Informationen). Der Datenteil der Nachricht enthält den Informationsgehalt oder die Daten der Nachricht.

Ein **Web Service** ist in XML Notation beschrieben, in dem alle notwendigen Details des Services erläutert werden um mit dem Service zu interagieren: Nachrichtenformate, Transportprotokolle und Lage. Die Implementations- Details des Services sind jedoch nicht in der Beschreibung enthalten, damit der Service unabhängig von der Hard- und Software sowie der Programmiersprache stattfinden kann.

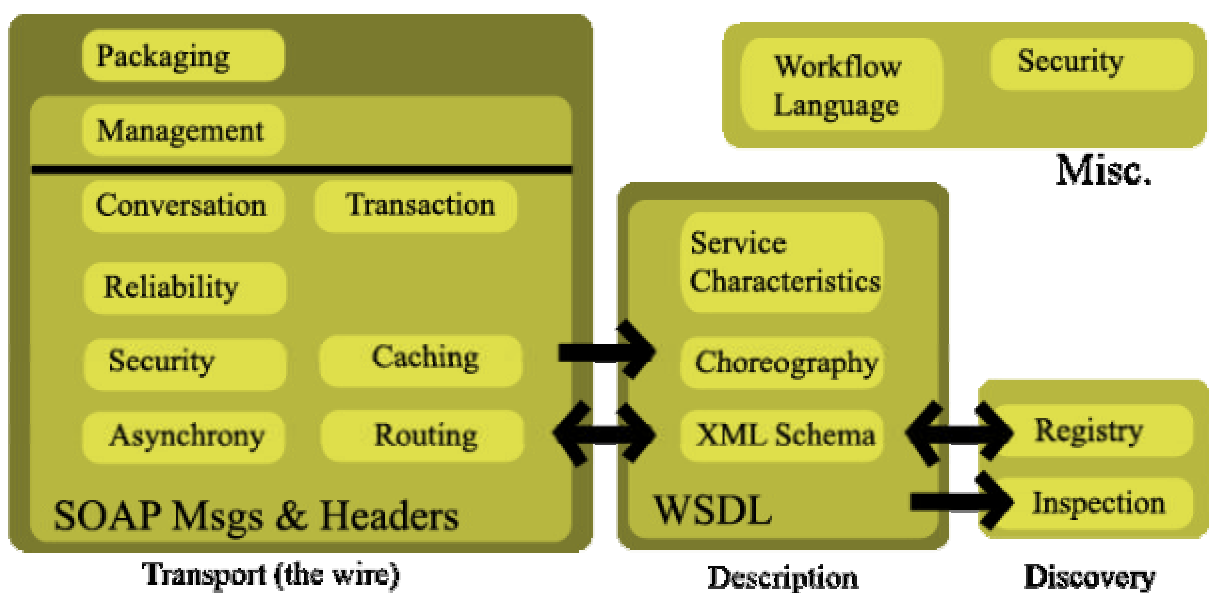
Hinweis: Die Web Service Architektur beinhaltet nicht das automatische Zusammentreffen von Anfrage und Antwort wie es z.B. manche RPC orientierte Technologien tun. Diese Zusammentreffen ist typischerweise von der Applikation abhängig.

3.3 Erweiterte Web Service Architektur

Die erweiterte Web Service Architektur berücksichtigt zusätzliche Eigenschaften und Funktionalitäten so genannte Features. Die Technologien und Komponenten der Basis Web Architektur werden um die konkrete Verwendung von Features erweitert, die an dieser Stelle nur abstrakt spezifiziert werden sollen und durch Nachrichtenaustauschformate oder Module realisiert werden sollen.

3.3.1 besondere Eigenschaften

Ein kleiner Überblick über zusätzliche Eigenschaften von Web Services und ein Hinweis auf deren konkrete Realisierung zeigt folgende Graphik:



Zu folgenden Themen werden wir noch in späteren Vorträgen etwas hören:

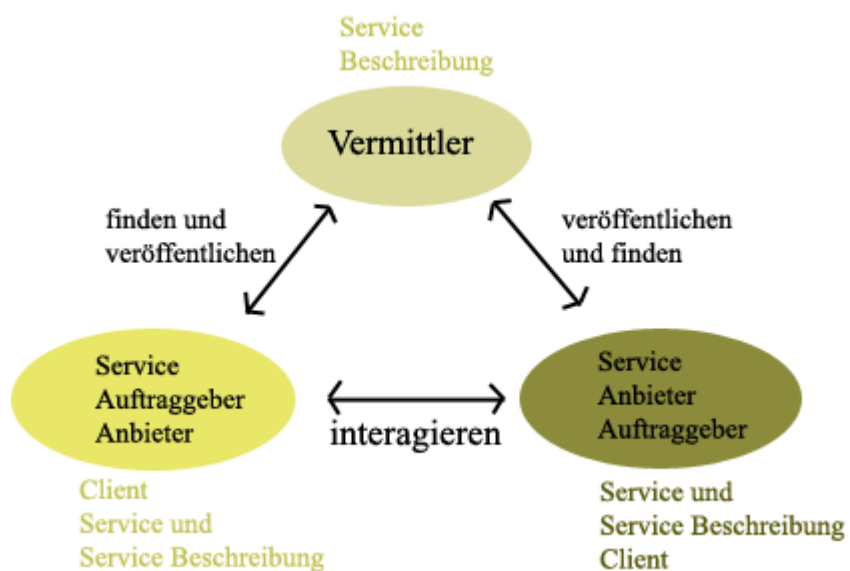
- Workflow (Arbeitsablauf, Arbeitsfluss) von Hana Stetinova
- Security von Katy Kirsche
- WSDL von Tanja Fichtner- Brunswig

3.3.2 Flows

In diesem Abschnitt werden einige abgeleitete Muster der Service Orientierten Architektur vorgestellt.

3.3.2.1 Peer to Peer

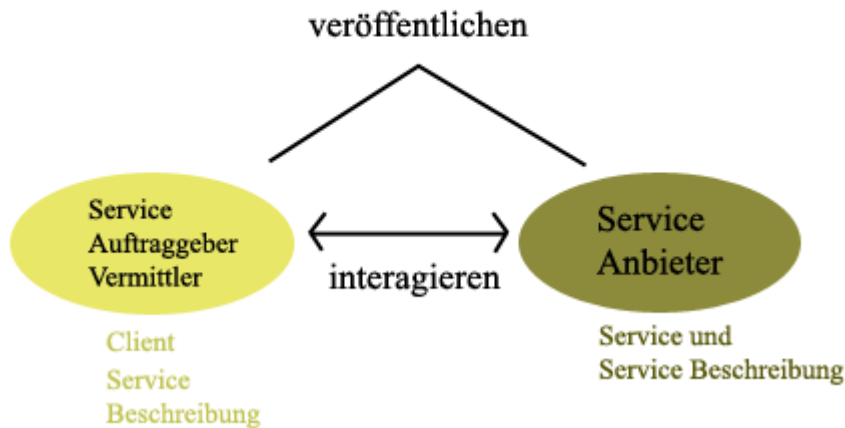
Eine Web Service Instanz kann mehrere Rollen gleichzeitig einnehmen. In einem Peer-to-Peer Szenario ist eine Web Service Instanz sowohl ein Service Auftraggeber als auch ein Service Anbieter.



Erweiterung der Service Orientierten Architektur: Peer to Peer

3.3.2.2 Direkte Wechselbeziehungen

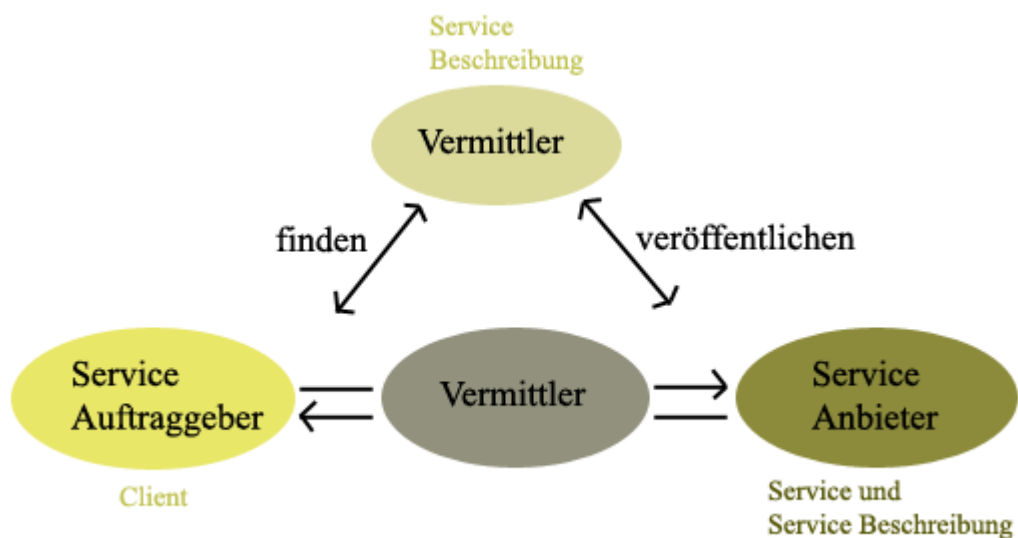
Wie oben bereits erwähnt, spielt der Service Vermittler nur eine optionale Rolle, denn die Rolle des Service Auftraggebers und des Vermittlers können gemeinsam durch den Client dargestellt werden.



Erweiterung der Service Orientierten Architektur: Direkte Interaktion

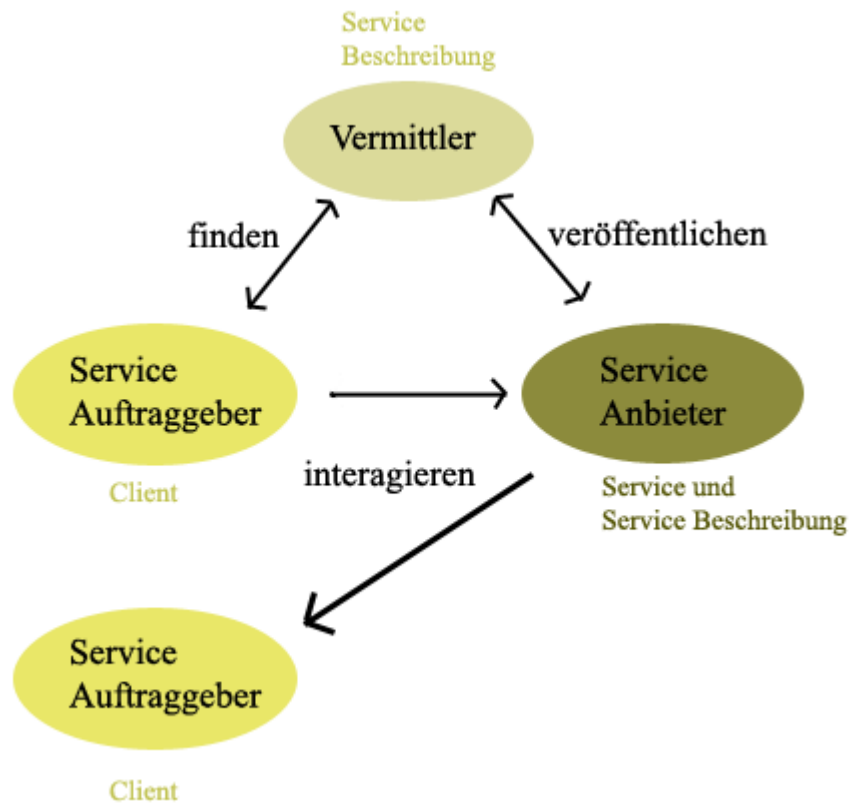
3.3.2.3 Zwischenvermittler

Zwischen Service Anbieter und Service Auftraggeber können sich Zwischenvermittler befinden.



Erweiterung der Service Orientierten Architektur: Vermittler

3.3.2.4 Nachrichtenversendung in nur einer Richtung



Erweiterung der Service Orientierten Architektur: Nachrichtenversendung in nur eine Richtung

3.3.3 Interaktion zweier Nachrichtenpartner

Bei der Durchführung der drei Operationen Veröffentlichen, Finden und Binden muss Kompatibilität gewährleistet werden, d.h. es müssen für jede Art der Interaktion zwischen den verschiedenen Rollen technische Standards entwickelt werden. Dabei gilt allgemein für jede Interaktion, dass Sicherheit und Zuverlässigkeit der Übertragung gewährleistet sein muss. Die genaue Erläuterung der Realisierung der verschiedenen Interaktionen wie z.B.:

- wire stack zur Interaktion von Service Auftraggeber und Service Anbieter (SOAP)
- Beschreibung des Services des Anbieters und dessen Veröffentlichung (UDDI, WSDL)

4 Methoden zur Architektur Analyse

Heutzutage kommt man bei der Entwicklung eines neuen Systems um eine Analyse des selbigen nicht herum. Es gibt bereits eine große Anzahl an Methoden, um ein Softwaresystem auf seine Funktionalität zu überprüfen. An dieser Stelle möchte ich nur kurz auf zwei Verfahren eingehen, die die Web Services Architecture Working Group zur Erstellung und Analyse der Architektur zur Hilfe genommen haben:

Critical Success Factor (CSF)

Usage Scenarios

Der Critical Success Factor ist hauptsächlich für die Erfüllung der oben genannten Ziele von Web Services verantwortlich und die Usage Scenarios sollen die Durchführbarkeit der von den Benutzern gewünschten Aktivitäten und Eigenschaften (siehe Graphik in 3.3.1) des Web Services überprüfen.

4.1 Critical Success Factor

Die CSF Methode zum Finden von Anforderungen geht von den Anforderungen der Organisation aus. Diese werden jeweils in Ziele mit hohem Level eingeteilt. Anschließend versucht man die Ziele anhand des Levels herunter zu brechen, bis man zu den so genannten kritischen Faktoren kommt, die die Ursache der Anforderung ausdrückt. Bricht man wiederum diese Faktoren herunter, so bekommt jeder Faktor eine konkrete Bedeutung für das System, so dass man erkennt, wo eventuell noch etwas zu verbessern ist. Einige SCF die den Gedanken dieser Analyse widerspiegeln sind:

- keine Anbieter bevorzugen
- Alle Schnittstellen und Protokolle standardisieren
- Einfachheit, Minimalität, Deklarativität
- Mechanismen zur Konsistenzprüfung
- Vorhersehbarkeit (Verhalten, Performance)
- Neue Versionen müssen abwärts kompatibel sein

4.2 Web Service Architecture Usage Scenarios

Da die Liste der Usage Scenarios sehr lang ist und die meisten Fälle recht verständlich sind, möchte ich an dieser Stelle nur einige wenige Szenarien vorstellen. Einige von ihnen lassen einige Vortragsthemen unseres Seminars einordnen. Die komplette Auflistung der Szenarien findet man in der Publikation des W3C: Web Service Usage Scenarios (siehe Literaturverweis).

4.2.1 Nachrichten Austausch Modelle: asynchrone, sichere und zuverlässige Gespräche

4.2.1.1 Fire and forget to single receiver

Ein Sender möchte eine unbeantwortete Meldung zu einem einzigen Empfänger schicken z.B. ein Aktienkurs update nach jeweils 15 Minuten.

4.2.1.2 Anfrage und Antwort

Zwei Handelspartner möchten auf elektronischem Weg ihre Dokumente austauschen. Der Sender verschickt Dokumente (Bestellungsaufträge, Fertigungsinformationen) in einer Anfrage Nachricht zum Empfänger. Daraufhin verschickt der Empfänger an den Sender eine entsprechende Antwort (Auftragsbestätigung, Änderungssteuerungsinformationen, Vertragliche Richtlinien).

4.2.1.3 Multiple asynchronous responses

Eine Applikation erfragt Information von einem Server. Diese Informationen werden später in mehreren Antworten geliefert. (möglicherweise weil nicht die komplette Antwort zum Zeitpunkt des Verschickens der Antwort vorhanden war)

4.2.1.4 Zwischenhändler

Ein Auktionsmarkt dient als Vermittler zwischen Käufern und Verkäufern. Die Käufer äußern ihre Wünsche und Anforderungen an die Marktplatz- Drehscheibe, die diese Informationen an verschiedene Verkäufer weiterleitet. Die Verkäufer senden ihr Angebot an die Marktplatz- Drehscheibe und diese vermittelt die Information weiter an den Käufer.

4.2.1.5 Anfrage mit verschlüsseltem Inhalt

Der Sender möchte beim Datenaustausch mit dem Empfänger die ganze oder nur einen Teil der Nachricht verschlüsseln, dabei sind die Sende- und Empfangs- Applikationen von der Verschlüsselung abhängig.

- Daten werden verschlüsselt und via SOAP an den Empfänger verschickt
- die Daten erreichen die Empfänger Applikation unberührt
- die Daten werden bei dem Empfänger entschlüsselt.

Zum Thema Sicherheit und Verschlüsselung werden wir von Katy Kirsche mehr erfahren.

4.2.2 Event basierender Nachrichtenaustausch

4.2.2.1 Event Mitteilung

Eine Applikation abonniert Meldungen von verschiedenen Events (Ereignissen) einer Eventquelle. Wenn einer dieser Events auftritt, werden entsprechende Meldungen an die Applikation verschickt. So kann z.B. eine Applikation über den Status eines Druckers informieren: Papier ist leer, Papierzufuhr verstopft, nicht genügend Tinte.

4.2.3 System Nachrichten

Ein Sender sendet Nachrichten zu einem Empfänger um den Status des Services/ der Nachricht zu erfragen. Somit kann die Durchführung einer Nachricht kontrolliert werden.

- der Sender ermittelt mit Hilfe einer synchronen Nachricht den Status des Services
- der Sender verschickt asynchron die eigentliche Nachricht
- nach einiger Zeit erfragt der Sender durch verschicken einer synchronen Nachricht den Status des Services und erhält somit Auskunft über die Durchführung der eigentlichen Nachricht

4.2.4 Service Beschreibung

4.2.4.1 Service Metadaten

Service Provider können kundenspezifische Daten in Ihrer Beschreibung anbieten. Diese Attribute können Applikations- spezifisch sein und werden von dem Web Service Betreiber in einem zusätzlichen Dokument (ein spezielles Schema) beschrieben. Diese Informationen können z.B. erforderliche Sicherheitsaspekte für bestimmte Nachrichten sein oder sie können eine benötigte Konfiguration für den Nachrichtenaustausch festlegen (z.B. maximale Antwortzeit).

4.2.4.2 Klassifizierendes System für Operationen

Grundlage ist ein Framework von Komponenten, bei dem die Komponenten und deren Operationen mit Hilfe von WSDL beschrieben sind. Zudem verwenden die Komponenten untereinander ihre Operationen. Die Besonderheit liegt nun darin, dass in dem Programmcode

keine „hard- wired“ Funktionsaufrufe beschrieben sind, sondern vielmehr eine „semantische Beschreibung bzw. Referenz“ der erst zur Laufzeit auszuwählenden Operation. Mit dieser semantischen Beschreibung muss nun eine Suche nach möglichen Operationen, die in einer zentralen Kartei (UDDI) gespeichert sind, gestartet werden. Resultat der Suche sind alle Operationen die der semantischen Beschreibung entsprechen, zusammen mit den Parametern (Protokoll, URL, Signatur), die für die Einbindung notwendig sind.

Ottmar Hilliges wird sich ausführlich mit dem Thema Web & Ontologien & Semantik im Web beschäftigen.

4.2.5 Finden von Web Services

4.2.5.1 Adress- basiertes Finden

Eine bestimmte Adresse (URL) eines Services ist gegeben und ein Sender möchte die Beschreibung eines Service ermitteln.

4.2.5.2 Register- basierende Entdeckung

Personen oder eine Software benutzt eine Kartei, um den web Service und die Schnittstellen Spezifikation zu finden

5 Realisierungskonzepte für Web Services

Derzeit existieren zwei große Modelle für die Umsetzung von Web Services: Java und .Net (Microsoft) basierende Anwendungen.

Die Anbieter der Java Technologie bemühen sich sehr um eine exakte Architektur zum Implementieren von Web Services mit Hilfe von Frameworks der Java 2 Enterprise Edition Specification.

Über diese beiden Programmier Modelle werden wir in den beiden letzten Vorträgen des Seminars durch Alexander Ernst und Roger Riff genauere Informationen erhalten.

Allgemein gilt für die Entwicklung eines Web Services, dass Ideen aus dem Geschäftsbereich mit den Realisierungsmöglichkeiten aus dem Programmierbereich unter einen Hut gebracht werden müssen.

So stellt sich ein Programmierer z.B. folgende Fragen: Wie kann ich die gleichzeitige Nutzung durch verschiedene Benutzer gewährleisten? Wie kann ich Konsistenz der Objekte sicherstellen? Wie bekomme ich den Service schneller?

Im Gegensatz hierzu stellt sich ein Verantwortlicher aus dem Geschäftsfeld eventuell folgende Fragen: Wie kann ich den Zugang zur Applikation vor Unbefugten schützen? Wie kann ich Ausfallsicherheit von den Web Service Transaktionen erhalten?

Es wird deutlich, dass hier Konflikte zwischen einerseits Robustheit, Leistungsverhalten und Schlichtheit und andererseits Sicherheit, Vertrauen und Funktionssicherheit entstehen, die in dem Web Service geeignet vereinigt werden müssen.

6 RosettaNet

Um meine Einführung in das Thema Web Services abzuschließen, möchte ich eine weitere Organisation vorstellen: RosettaNet. Ihr Ziel ist, die allgemeinen Geschäftsaktivitäten im High- Tech- Production- Bereich weltweit zu unterstützen. Es sollen offene, für die gesamte Industrie einheitliche eBusiness- Standards geschaffen werden. Diese Prozess- Standards sollen die Basis für die Angleichung der Geschäftsprozesse entlang der gesamten Beschaffungskette bilden.

Dazu werden für geschäftliche Transaktionen Nachrichtenformate festgelegt und es werden Abläufe und Regeln aufgestellt.

Das RosettaNet- Konsortium ist eine Selbstfinanzierte und gemeinnützige Organisation großer Produzenten in den Bereichen Informationstechnologie, Produktion elektronischer Komponenten und Halbleitererzeugung. Sie umfasst derzeit über 400 führenden IT-, Electronic Components- und Semiconductor Manufacturing- und Solution Provider- Unternehmen wie z.B. Microsoft, IBM, Intel, American Express.

RosettaNet kündigt eine Reihe von Meilensteinen bei seiner Bemühung um die Schaffung einer universellen Handlungsumgebung auf XML- Basis an. So äußert sich z.B. der Direktor der Abteilung B2B- Integration und RosettaNet Programmen bei Intel: „RosettaNet ist real. Unternehmen in der Elektronikbranche können reale B2B- Integrationen vornehmen und RosettaNet für die Durchführung realer Unternehmenstransaktionen verwenden.“

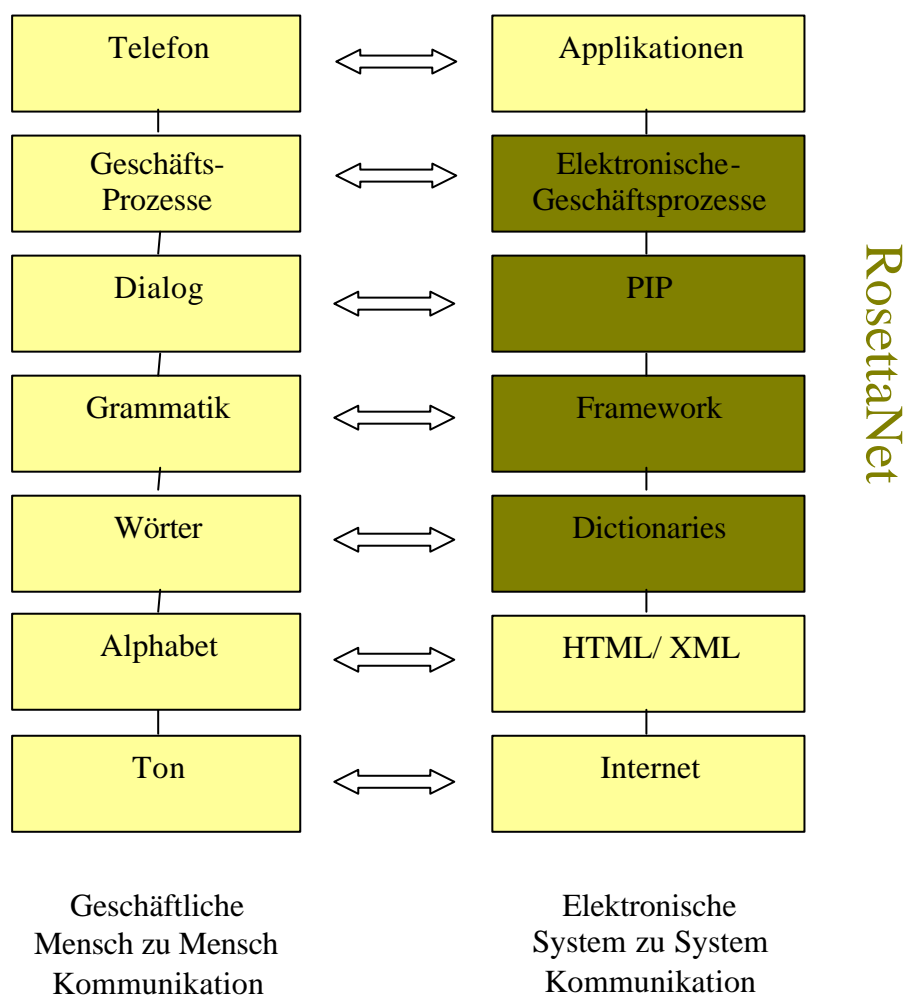
Die Verbreitung von RosettaNet- Verbindungen liegt in der erheblichen Reduzierung der Kosten und der Effizienzsteigerung für Unternehmen gezielter Branchen begründet. Dabei wird die Effizienz durch grundlegende Fragestellungen des jeweiligen Unternehmens erzielt. Besonders Augenmerk wird dabei z.B. der Beschreibung von Daten, der Charakterisierung von Geschäftsabläufen und die Umsetzung von Interaktionen zugeteilt. Es ist wichtig, dass

bei den Transaktionen weniger aber dafür aussagekräftige und funktionelle Berührungspunkte bestehen.

Der durch den RosettaNet entwickelte Standard ermöglicht den Unternehmen einen raschen Übergang zu automatisierten Transaktionen, Logistikketten- Management und die gemeinsame Datennutzung mit anderen Mitgliedern des Konsortiums. RosettaNet bietet einen bedeutenden Einblick in die Möglichkeit der effektiven Zusammenarbeit zwischen großen Branchen, um so einen nützlicheren und wettbewerbsfähigen e- Commerce- Markt zu schaffen.

Der Ansatz von RosettaNet liegt zunächst in der Abwicklung von Geschäften zweier Verhandlungspartner und gliedert diese in einem Schichtenmodell. Dieses Modell wird in korrespondierende technische und formale Konzepte projiziert.

Schichtmodell RosettaNet:



Der RosettaNet Standard umfasst:

- **Dictionaries** zur konsistenten Beschreibung geschäftlicher und technischer Sachverhalte der abgedeckten Prozesse (Metadaten), zur Orientierung für PIPs und sorgt für die Datenstruktur durch DTDs
 - o RosettaNet **Business Dictionary**: produktabhängige Spezifikationen der Begriffe und Terminologien der abgedeckten Prozesse
 - o RosettaNet **Technical Dictionary**: zur Produktdatenstruktur und auch zur Grundlage für Anwendungsfällen (Generierung technischer Spezifikationen, Pflege technischer Informationsdatenbanken, Suche in elektronischen Katalogen)
- RosettaNet **ImplementationFramework (RNIF)**: spezifiziert Informationsaustausch über XML
- RosettaNet **Partner Interface Processes (PIPs)**: unterteilt in Business- Message(s) und Message- Dialog
 - o **Business- Message**: Geschäftsinformationen (Datenmodell)
 - o **Message - Dialog**: Reihenfolge des Nachrichtenaustausch (Geschäftsprotokoll)

Es folgt nun die Einordnung dieser Begriffe.

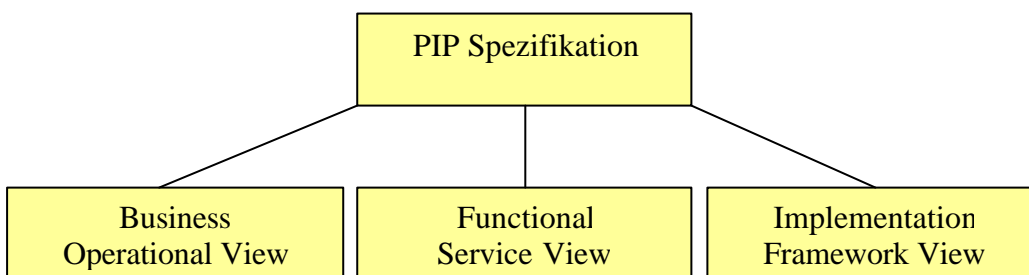
Zunächst beschreibt die obere Ebene der Spezifikation sieben Cluster:

Cluster	Bezeichnung	Beschreibung	Beispiel
0	Support	administrative Funktionen rund um RosettaNet selbst	
1	Partner, Produkt und Dienstleistungen	Gibt Geschäftspartnern die Möglichkeit, ihre Stammdaten, Verträge und zu verkaufende bzw. zu kaufende Produkte zu warten	
2	Produktinformationen	Produktbeschreibung, Änderungsnachweise, automatische Synchronisation der Produktkataloge	
3	Auftragsmanagement	Zentrale Geschäftsprozess zur Auftragabwicklung	Bestellung, Konfiguration, Versenden, eventuell Rücklieferungen, finanzielle Abwicklung der Transaktion
4	Lagerverwaltung	Wiederauffüllen des Lagers, Auswertung des Lagerbestandes	
5	Marketing	wichtiger Marketingaktionen	Restpostenverkäufe
6	Service und Support	Beschreibung von Prozessen, die nach dem Verkauf stattfinden	Garantieleistungen, technischer Support
7	Herstellung	: noch nicht vollständig definiert, umfasst Prozesse, die letztlich ein virtuelles Unternehmen abbilden können sollen	Produktdesign, verteiltes Arbeiten, Definition und Austausch globaler Prozesse

Diese Cluster werden weiter zerlegt in Segmente, die die Prozesse genauer beschreiben. Jedes Segment wird durch die PIPs zusammengesetzt, die Aktionen zwischen Geschäftspartnern auf der unteren Ebene beschreiben.

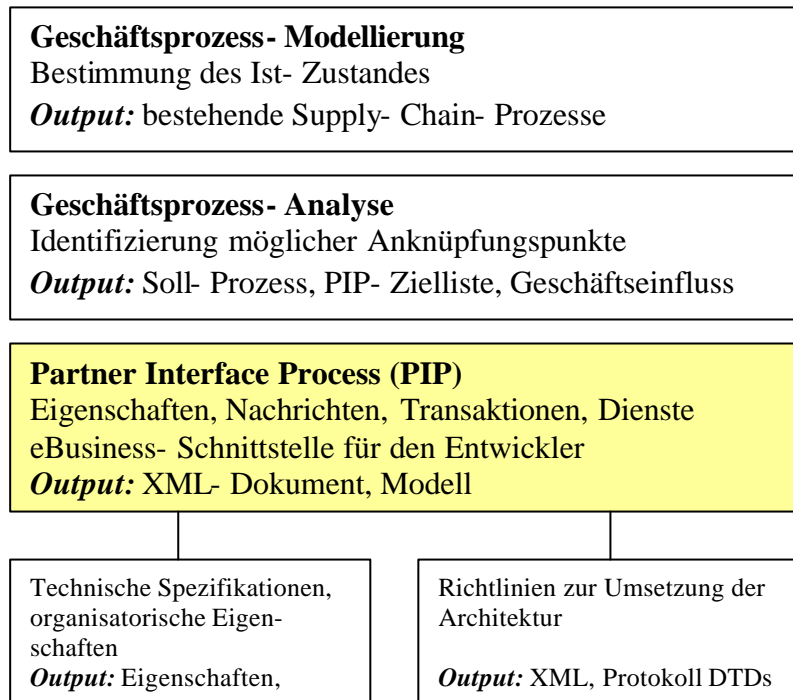
Die PIPs (Partner Interface Processes) werden von den Mitgliedern der RosettaNet- Initiative erarbeitet und potentiellen Geschäftspartnern zu Verfügung gestellt. Sie stellen eine standardisierte Kommunikation zwischen den Geschäftspartnern und vereinheitlichen Daten (Produktdaten und Geschäftsdaten) dar. PIPs sind XML basierte Dialoge. Jede Spezifikation enthält ein oder mehrere Geschäftsdokumente mit einem Vokabular und Angaben für den Nachrichtendialog. Diese werden mittels UML Aktivitäts- und Sequenzdiagrammen spezifiziert. PIP Spezifikationen enthalten ebenfalls Elemente für die Gewährleistung der Sicherheit.

Die PIP Spezifikation beinhaltet folgende Schichten:



- Die Business Operational View beinhaltet alle Geschäftlichen Aspekte wie gegenseitige Anforderungen und Verpflichtungen
- Functional Service View ist die technische Schnittstelle und enthält Funktionen, vor allem Protokolle
- Das RosettaNet Implementation Framework (RNIF) wird benötigt, um ein PIP zu implementieren. Ein solches RNIF definiert Implementierungsrichtlinien für Software, die PIPs ausführt. Es legt Protokolle und Nachrichtenformate für den Nachrichtentransport fest. Durch diese Spezifikation wird gewährleistet, dass PIPs zwischen verschiedenen Geschäftspartnern funktionieren. RNIF bietet Routing über Hubs und direkte Verbindungen, Packaging, Integration von Sicherheitskonzepten, Einsatz von Trading- Partner- Agreements

Unter Trading Partner Agreement (TPA) versteht man die Vereinbarung der Handelspartner über die zu verwendenden PIPs für den Nachrichtenaustausch. Anschließend kann eine PIP Instanz als eine Sequenz von Geschäftsnachrichten ausgeführt werden.



Entwicklung von PIP Konfigurationen

Abschließende Bemerkung zu RosettaNet:

Abgesehen von dem was RosettaNet leisten kann, müssen sich die IT- Organisationen gründlich auf den Übergang zu einer neuen Lösung des E- Commerce vorbereiten. Sie müssen enorme Projektplanungs- und Stellenbesetzungsaktivitäten vornehmen, die bestehende Infrastruktur anpassen, damit sie den Einsatz unterstützt, und Datenvorbereitungsaufgaben wahrnehmen. Ein Unternehmen muss seine Geschäftsabläufe verstehen und muss in der Lage sein diese so umzudefinieren, dass sie einen Wettbewerbsvorteil erzielen können. Nur mit diesen Voraussetzungen kann RosettaNet das leisten, wozu es im Stande ist. Mehr zum Thema Workflow werden wir im Vortrag von Hana Stetinova hören.

7 Abschluss

Web- Services und die damit in Verbindung stehenden Standards sind in aller Munde. Jedoch klaffen Anspruch und Realität derzeit noch auseinander, d.h. es gilt zu unterscheiden, was Web- Services heute sind und was sie in Zukunft sein können.

In meiner Dokumentation habe ich mich darauf konzentriert, welche Rolle Web Services heute spielen. Ich habe festgestellt, dass die Geschäftsabläufe in den letzten Jahren immer arbeitsteiliger geworden sind. Ein Geschäftsprozess wird nicht mehr von einer einzelnen Person, sondern verteilt über Abteilungen oder sogar von einer Menge von Firmen abgearbeitet.

IT- Systeme, die derartige Geschäftsprozesse unterstützen, müssen in die bestehenden Systeme der Unternehmen integriert werden. Hieraus erkennt man die Notwendigkeit, Standards aufzustellen um Kompatibilität zu gewährleisten. Auf diesen Aspekt und weitere Anforderungen die an Web Services gestellt werden bin ich im Laufe des Dokumentes eingegangen, in dem ich mit Hilfe der W3C Web Services Architecture Working Group veröffentlichten Richtlinien die Ansprüche an Web Service erläutert habe. Daraus ließ sich eine grundlegende Architektur für Web Services aufstellen, die den Anforderungen der Unternehmen gerecht werden soll.

Darüber hinaus habe ich zwei Realisierungskonzepte für Web Services einleitend angesprochen, die am Ende unseres Seminars noch ausführlich erläutert werden.

Abschließend ist darauf hinzuweisen, dass Web Services allein nur Schnittstellen- Syntax definieren. Idealerweise müssen sie mit Standards für Geschäftsdokumente angereichert werden, um auch die Semantik von Geschäftstransaktionen zu bestimmen. Als entsprechenden Ansatz habe ich an dieser Stelle die Organisation RosettaNet vorgestellt, die sich zum Ziel gemacht hat, für die gesamte Industrie einheitliche eBusiness- Standards zu schaffen.

8 Literaturverzeichnis

Web Services:

- [1] James Snell, Doug Tidwell, Pavel Kulchenko: Programming Web Services with SOAP, O'Reilly
- [2] Jörg M. Freiburger, Andreas Maslo: Web Services in Visual Basic.NET
- [3] Kris Gopalakrishnan: Aus der Praxis
http://www.infosys.com/german/infosys_perspective/collaberatingweb.htm
- [4] Web Service Architecture Group at W3C
<http://www.w3.org/2002/ws/arch/>
- [5] Donald Kossmann: Web Services, Web Informationssysteme, WS 2002/03

RosettaNet:

- [6] Intel Corporation: RosettaNet kommt in den Gang
<http://www.intel.com/deutsch/ebusiness/products/RosettaNet/ar012501.htm>
- [7] Armin Pulic: RosettaNet
<http://www.transaktionsstandard.de/RosettaNet/>
- [8] Alma Schütter : B2B
http://www.ifi.unizh.ch/richter/Classes/sem_cutting_edge/Summary_B2B.pdf