



SIMPL

-

**SimTech: Information Management, Processes and
Languages**

SOA und WEB SERVICES

Daniel Brüderle



Agenda

- 1.0 SOA
- 2.0 Web Services
- 2.1 SOAP
- 2.2 WSDL



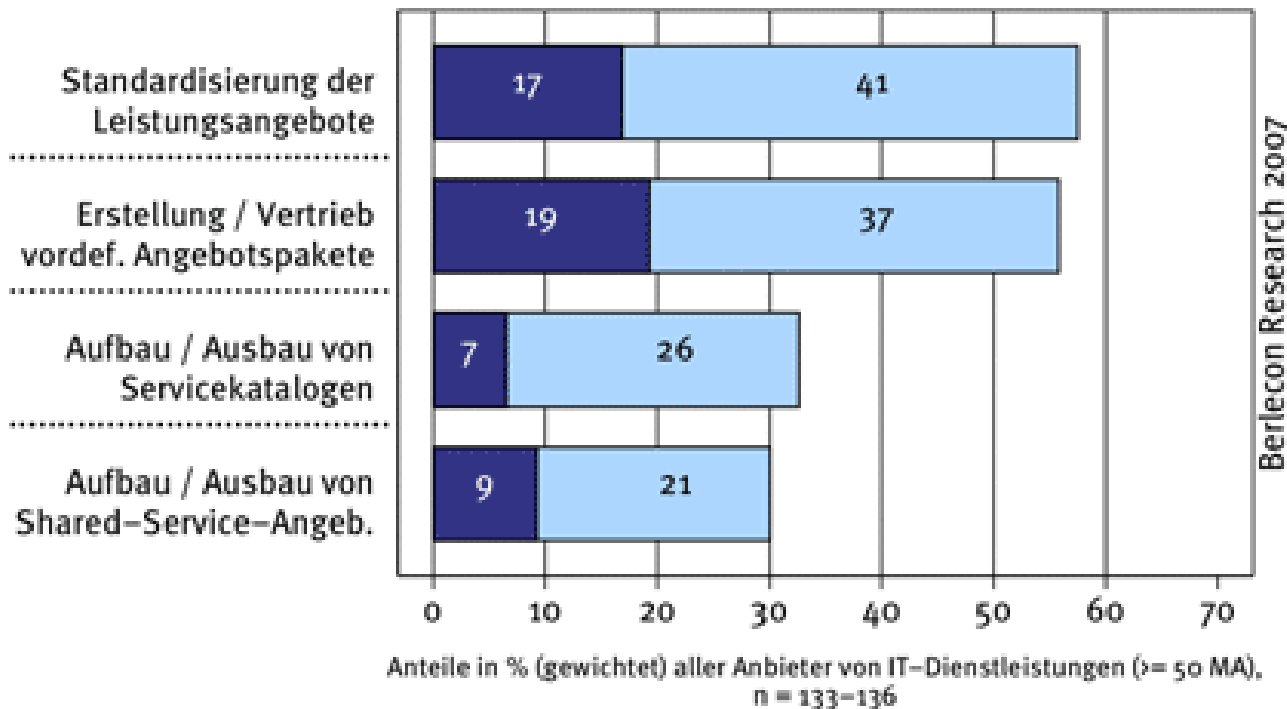
1.0 WARUM SERVICEORIENTIERTE ARCHITEKTUR (SOA) ?



Anforderungen der Industrie

Industrialisierung (Teil 1): Wie bedeutend sind die folgenden strategischen Maßnahmen für ihr Unternehmen

■ Sehr große Bedeutung ■ Große Bedeutung



Quelle : Berlecon Research GmbH, Berlin 2007



1.1 Definition SOA

- Ist eine Softwarearchitektur
- die mit einer Schnittstellen-Definition beginnt
- und die die ganze Anwendungstopologie als eine Topologie von Schnittstellen, -implementierungen und –aufrufen aufbaut.

(nach Gartner 2003)



1.3 WARUM SOA ?

Veränderungen auf **technologischer** Ebene :

Erfahrungen mit verteilten Systemen und Middleware zeigte
Notwendigkeit von Middleware-Standards und Architekturen.

Veränderungen auf **geschäftlicher** Ebene :

Trend zum Ausgliedern
Bedarf nach flexiblen Prozessen

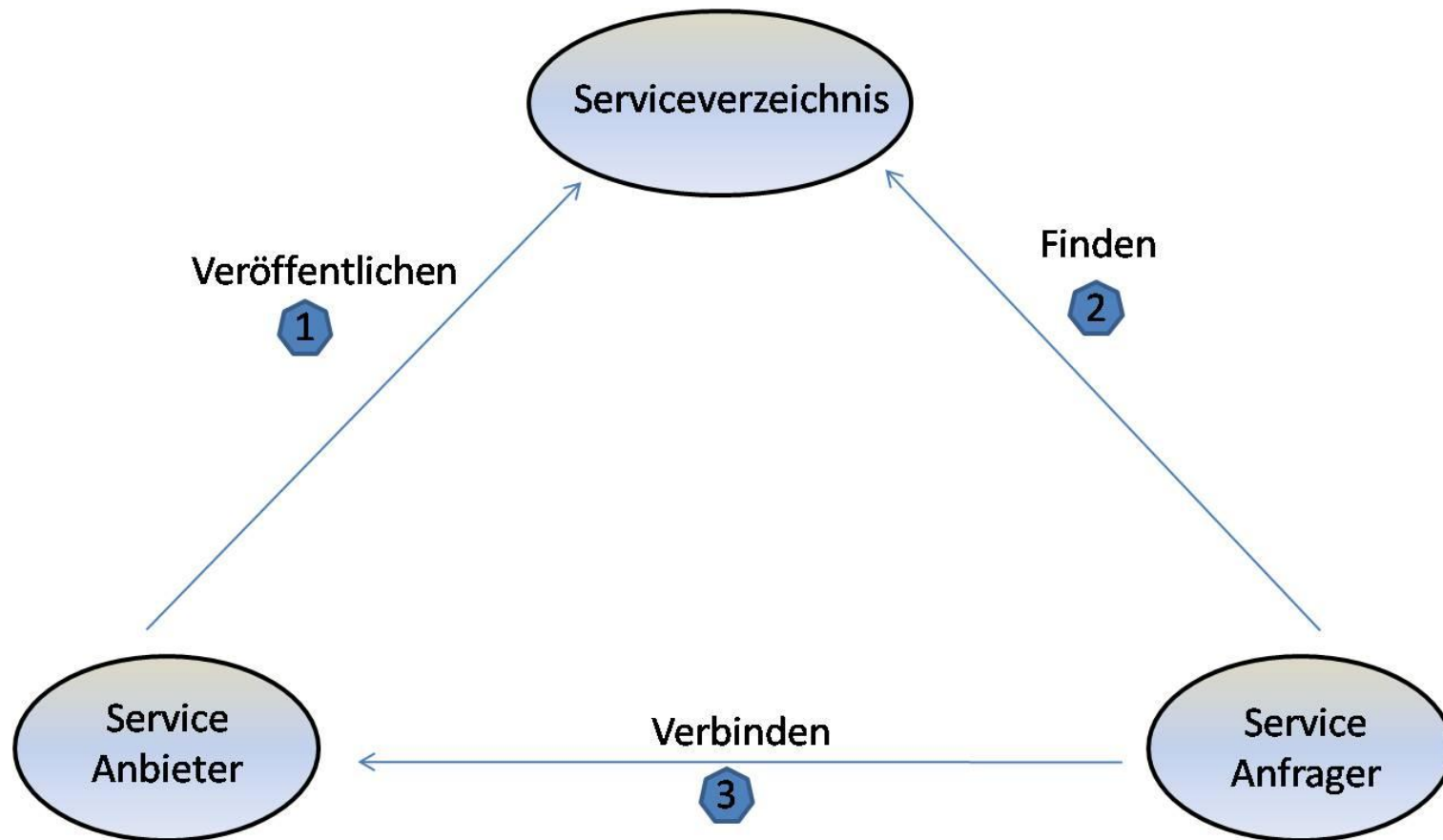


1.4 ZIELE VON SOA

- Abbildung und Unterstützung von Geschäftsprozessen
- Flexibilität beim modellieren von Prozessen
- Geringeres Entwicklungsrisiko
- Reduzierung der Technologieabhängigkeit
- Automatisierung von Prozessen



1.5 ARCHITEKTUR SOA





1.4 WAS IST EIN SERVICE ?



In der Informatik :

Eine **Softwarekomponente** die so konstruiert ist, damit sie sich auf einfache Weise mit anderen Softwarekomponenten **verbinden** lassen kann.



1.4 EIGENSCHAFTEN EINES SERVICES

Typische Eigenschaften eines Services :

- Registrierung in einem Verzeichnis
- Dynamische Anbindung
- Abgeschlossenheit und eigenständige Nutzbarkeit
- Verfügbarkeit durch verschieden Transportmöglichkeiten, Formate und QoS
- Plattformunabhängigkeit zwischen Anbieter und Nutzer
- Dauerhafte Verfügbarkeit („always on“)
- Existenz einer veröffentlichten Schnittstelle



1.5 LOSE KOPPLUNG

In Systemen die mehrere Komponenten verbinden entsteht ein höheres Risiko von Fehlern und Ausfällen.

Eine enge Kopplung erhöht das Risiko daß aus einem Fehler multiple Fehler werden die zu einem Systemfehler auswachsen können.“Insofern ist die lose Kopplung der Services die ideal Voraussetzung für Emergenz als auch Überlebensfähigkeit eines aus Services bestehenden Gesamtsystemes“.

(Masak 2007, S. 314).



Notwendigkeit einer
losen Kopplung !



1.5 LOSE KOPPLUNG



Lose Kopplung =

Geringer Grad an Abhängigkeiten
zwischen den Komponenten



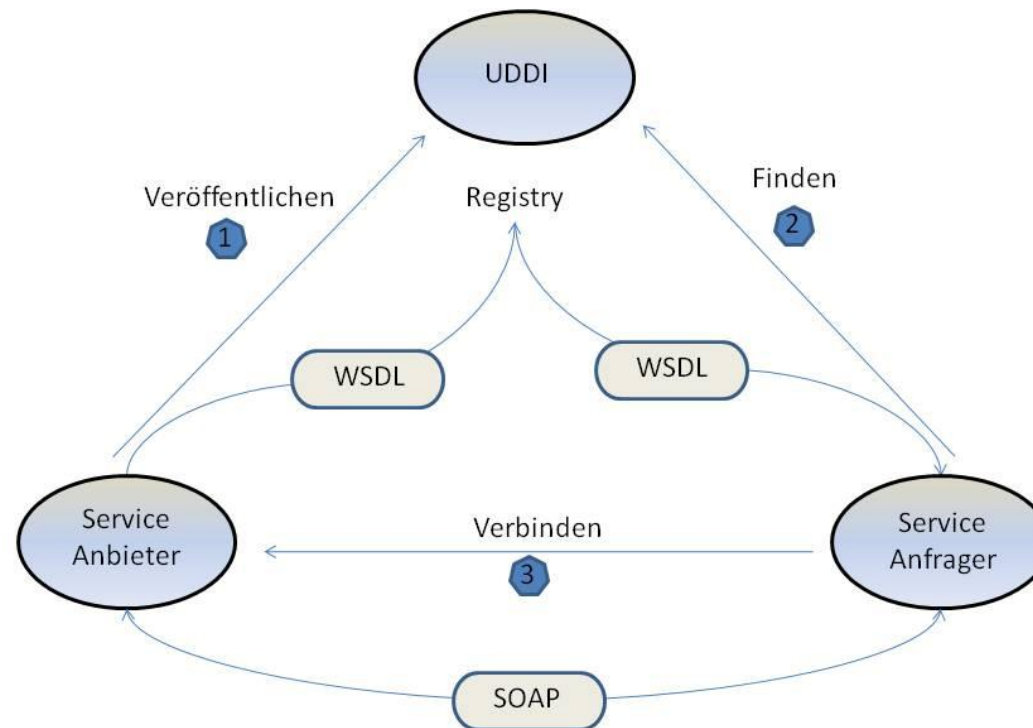
2.0 WEB SERVICES



- Softwaresystem
- Maschinen-zu-Maschinen Interaktionen
- Netzwerk
- Definierte Schnittstellen (WSDL)
- Interaktion mit anderen Systemen
- Definierter Nachrichtenaustausch (SOAP)
- Typisch HTTP Basis
- XML Serialisierung



2.0 AUFBAU UND KOMMUNIKATION





2.0 WEB SERVICES = SOA ?

Web Service = NUR eine mögl. Realisation einer SOA

SOA = Architekturkonzept



2.1 SOAP

Bis zur Version 1.1 war SOAP ein Akronym für „Simple Object Access Protocol“.

In der Zwischenzeit steht SOAP für sich selbst und ist kein Akronym mehr – es steht für eine Nachrichten Architektur ! (vgl Leymann, Skript, S. 31)

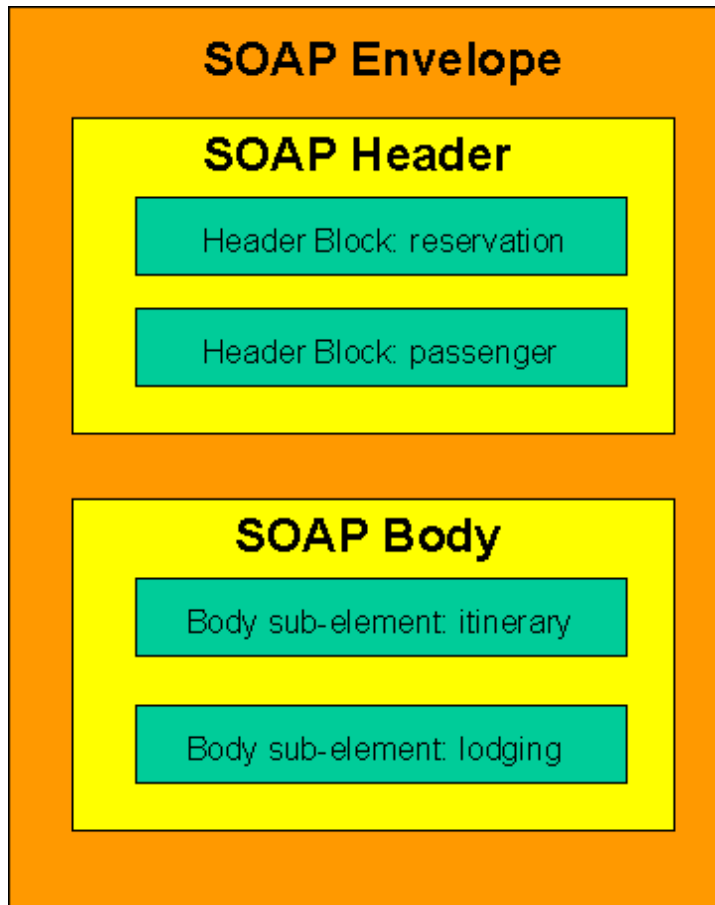


2.1 SOAP

- XML basierende Definitionen
- Einsatz in dezentraler Umgebung
- Verarbeitungsvorschriften
- „One-way“Kommunikation
- Komplexe Strukturen möglich



2.1.1 SOAP NACHRICHT

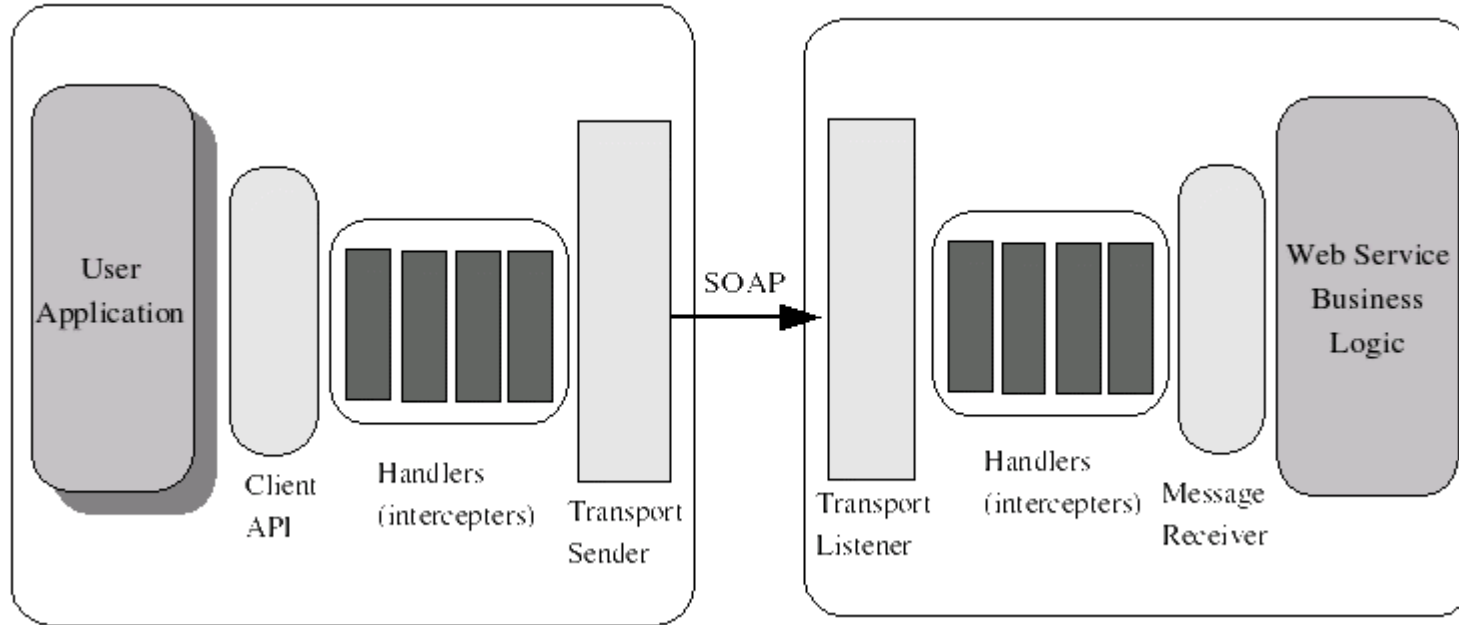


Envelope, Header & Body

(W3C, <http://www.w3.org/2000/xp/Group/2/06/LC/primer-figure-1.png>)



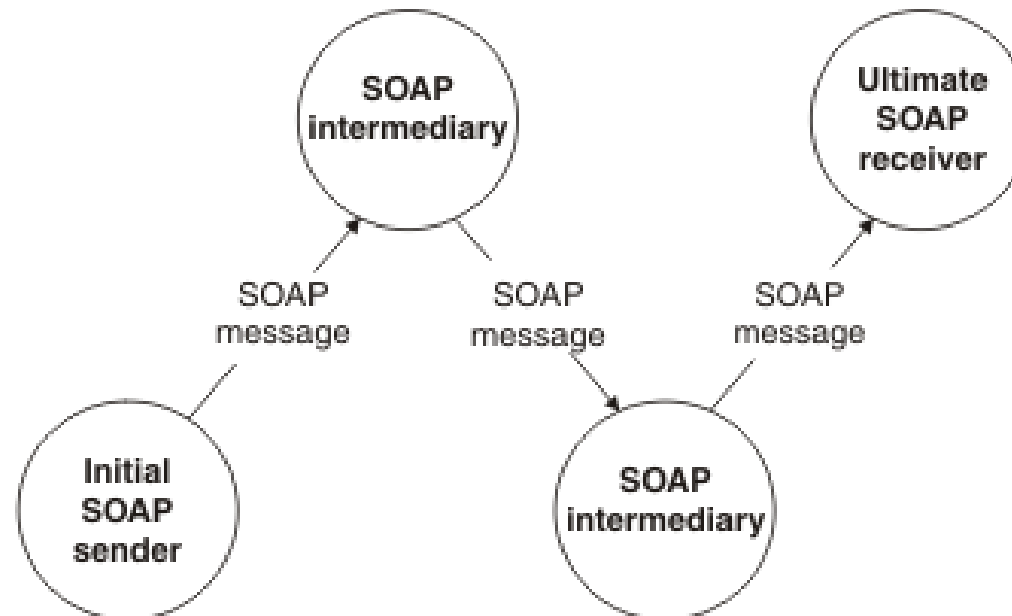
3.2 SOAP VERFAHRENSMODEL



Nachricht wird durch für den SOAP Versand serialisiert.



2.1.2 SOAP VERFAHRENSMODEL



Definierung des Pfades durch Header.



2.1.2 SOAP VERFAHRENSMODEL

- SOAP definiert die **Verarbeitung**
- SOAP definiert **NICHT den Inhalt**



Hohe Flexibilität



2.2 WEB SERVICE DESCRIPTION LANGUAGE (WSDL)

“SOAP ist ein Nachrichtenformat das jeder in der Web Service Welt versteht. **WSDL** ist das was alle benutzen um anderen **mitzuteilen was sie können.**”

(vgl. Leymann et al. 2006, S.105)



2.2 WAS IST WSDL ?

Eine Beschreibungssprache zur :

- Lokalisierung von Diensten
- Beschreibung von Funktionen
- Beschreibung des Zugriffs



2.2 WSDL ARCHITEKTURKONZEPT

- **Erweiterbarkeit**
- **Unterstützung für andere Systematiken**
- **Unifying Messaging und RPC**
- **Trennung von „Was“ von „Wie“ und „Wo“**



2.3 WICHTIGE WSDL ELEMENTE

„**message**“ : Auszutauschende Daten.

„**types**“ : Datentypen.

„**portType**“: Verfügbare Operationen

„**binding**“ : Spezifizierung Protokoll u. Datenformat

„**definitions**“ : Definiert Menge von Definitionen (namespace)

(vgl. W3C, WSDL 1.1)



Vielen Dank für Ihre Aufmerksamkeit.