

# **Service-Orientierten-Architektur und Web-Services**

Einführung und  
Grundkonzepte

Materialien erstellt  
von Claus P. Priese

# Web-Services und Service-Orientierten Architektur

Wird seit 1999 von mehreren Arbeitsgruppen des World Wide Web Consortium entwickelt.

Die wichtigsten Architektur-Dokumente sind:

- [W3C02a] World Wide Web Consortium: Web Services Architecture Requirements, W3C Working Draft, November 2002, <http://www.w3.org/TR/wss-reqs/>
- [W3C03d] World Wide Web Consortium: Web Services Glossary, W3C Working Draft, August 2003, <http://www.w3.org/TR/ws-gloss/>
- [W3C03e] World Wide Web Consortium: Web Services Architecture (WSA), W3C Working Draft, August 2003, <http://www.w3.org/TR/ws-arch>

# Architektur

Die Arbeitsgruppe mit der Aufgabe der Erstellung einer übergreifenden Referenz-Architektur für Webservices gibt in ihren *Web Services Architecture Requirements* [W3C02a] folgende Definition für den Begriff Webservice:

*“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.”*

Im von derselben Arbeitsgruppe herausgegebenen *Glossar* [W3C03d] wird folgende Definition gegeben:

*“A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.”*

# Begriffe und Konzepte

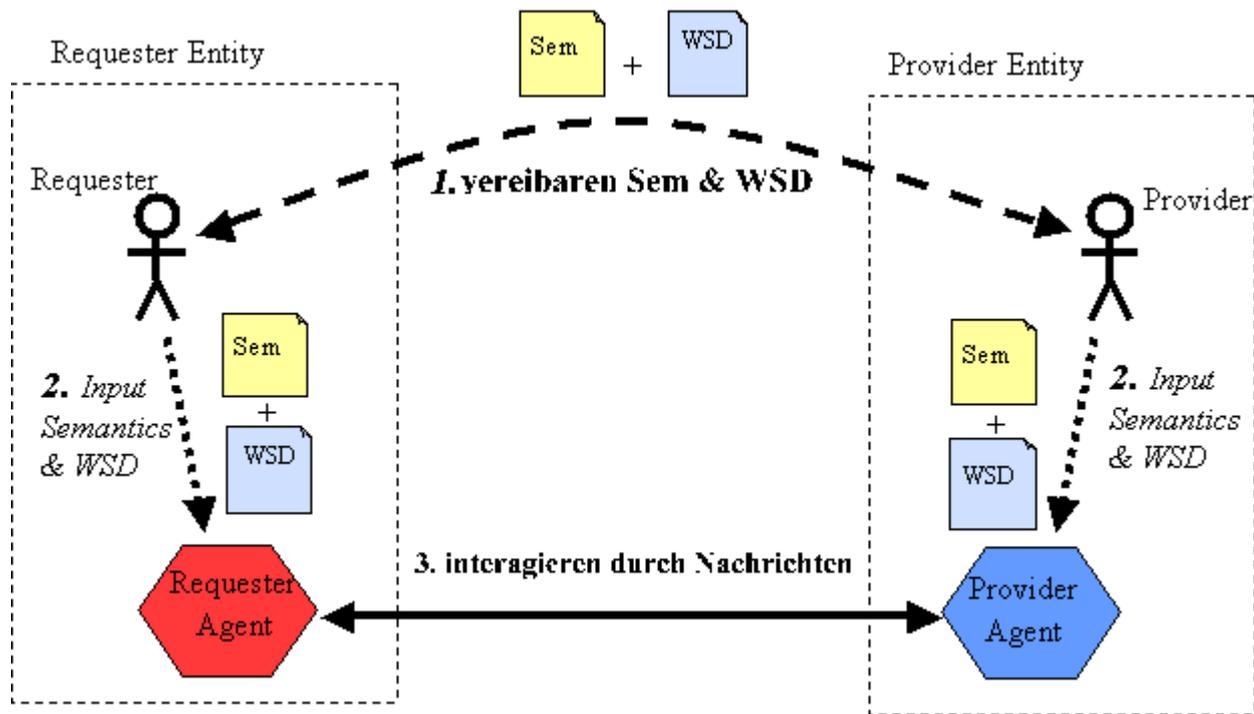
- Web Service: abstrakte Beschreibung von Funktionalität und Schnittstellen
- Agent: eine konkrete Implementierung eines Web Service, geschrieben in einer beliebigen Programmiersprache
- Provider: die Person oder Organisation welche die konkrete Implementierung eines Web Service zur Nutzung bereitstellt
- Requester: die Person oder Organisation welche die vom Provider bereitgestellte Implementierung eines Web Service nutzt; Für den Nachrichtenaustausch wird eine eigener requester-seitiger agent benutzt

# Begriffe und Konzepte

Service Description (WSD): maschinell verarbeitbare Spezifikation von Web Services Schnittstellen in WSDL mit begleitenden Informationen über Datentypen, Protokolle, Konkrete Endpunkte und Nachrichtenaustauschmuster

Semantik: hier: der Vertrag zwischen Requester und Provider über die Bedeutung und Zweck der Interaktion von Agenten, soweit noch nicht in der WSD enthalten. Die Art der Spezifikation der Semantik – also ob mündlich, umgangssprachlich oder formal, ist nicht festgelegt.

# Zusammenhänge im Überblick



# WS & SOA – Web Services Architektur

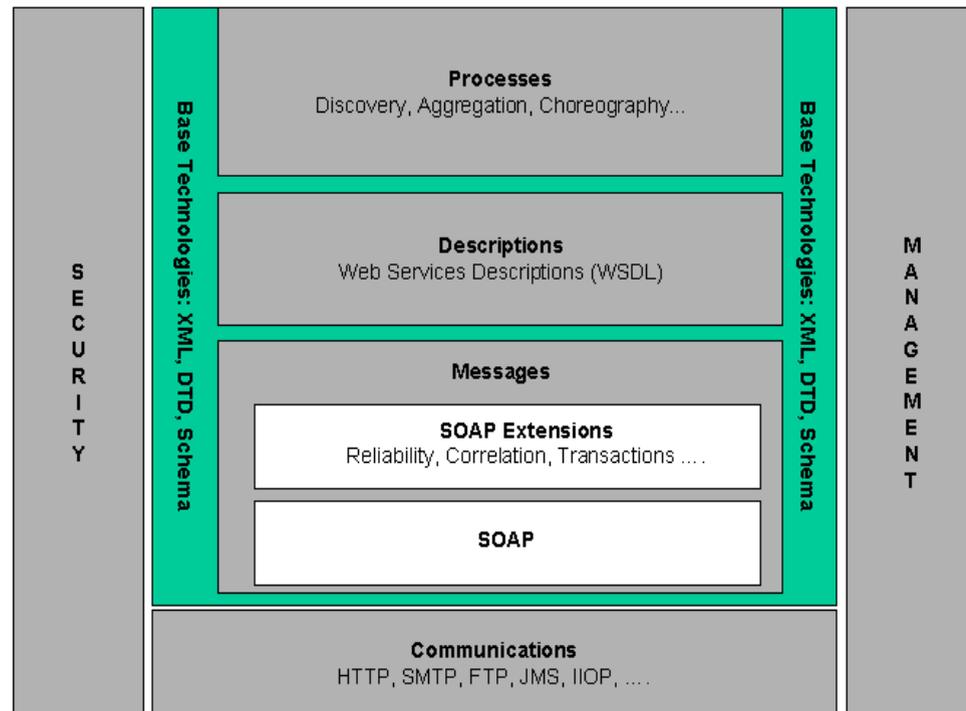
## 6 Ziele:

- Interoperabilität
- Integration mit dem World Wide Web
- Zuverlässigkeit von Web Services
- Sicherheit von Web Services
- Skalierbarkeit und Erweiterbarkeit
- Managebarkeit von Web Services

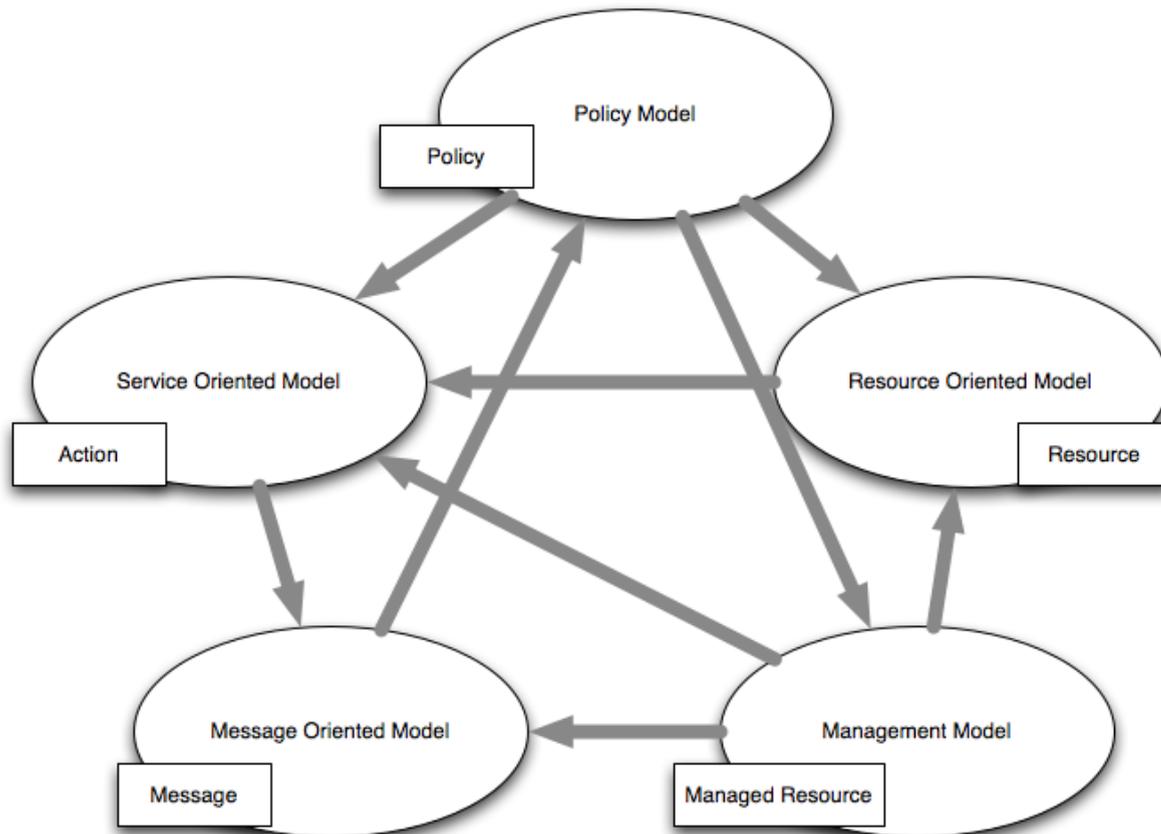
## 5 Architektur Modelle (Sichten):

- Nachrichtenorientiertes Model
- Service Oriented Model
- Resource Oriented Model
- Policy Model
- Management Model

## Technologien:

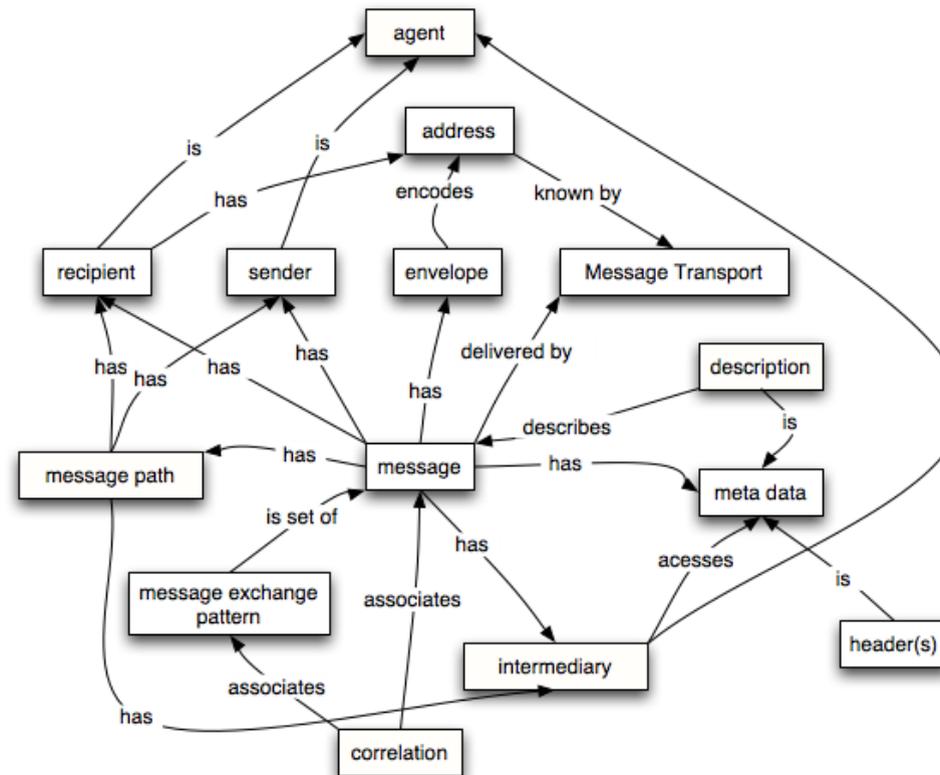


# Zusammenhang der Modelle



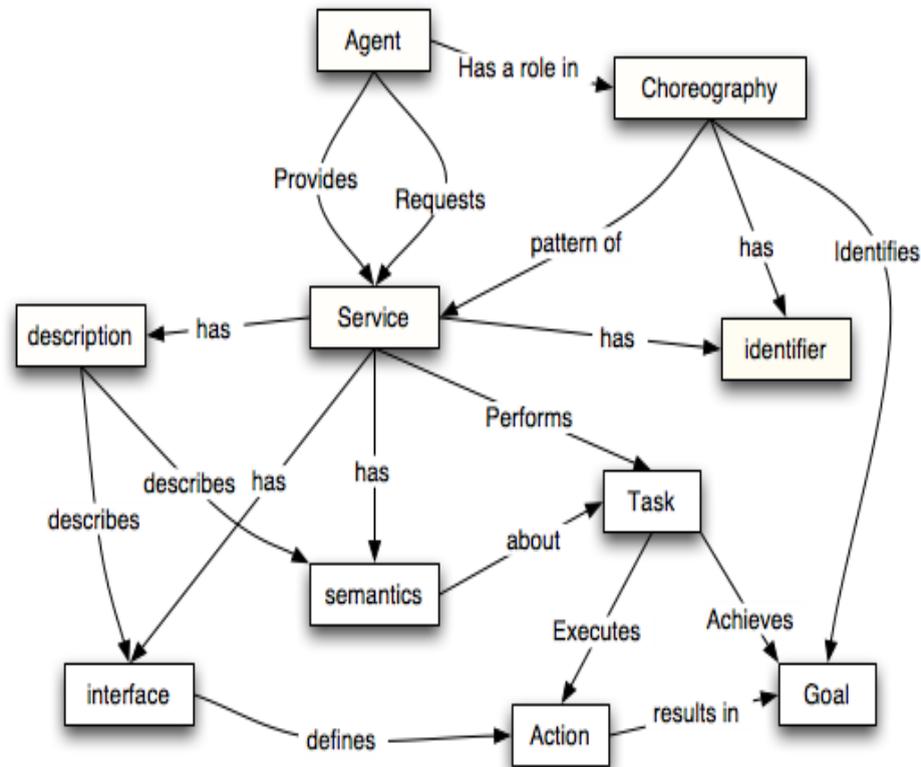
# WS & SOA – Nachrichtenorientiertes Modell

- Nachricht := Daten gesendet von Agent zu Agent, die Datenstruktur wird in Service-Beschreibungssprache bestimmt
- Gegenstand: Beziehung zwischen Sender und Empfänger (receiver)
- Empfänger hat immer transportmechanismus-spezifischen Identifikator (can be an URI)
- Nachrichten-Austausch-Muster beschreiben Gruppen von Nachrichten zwischen Agenten



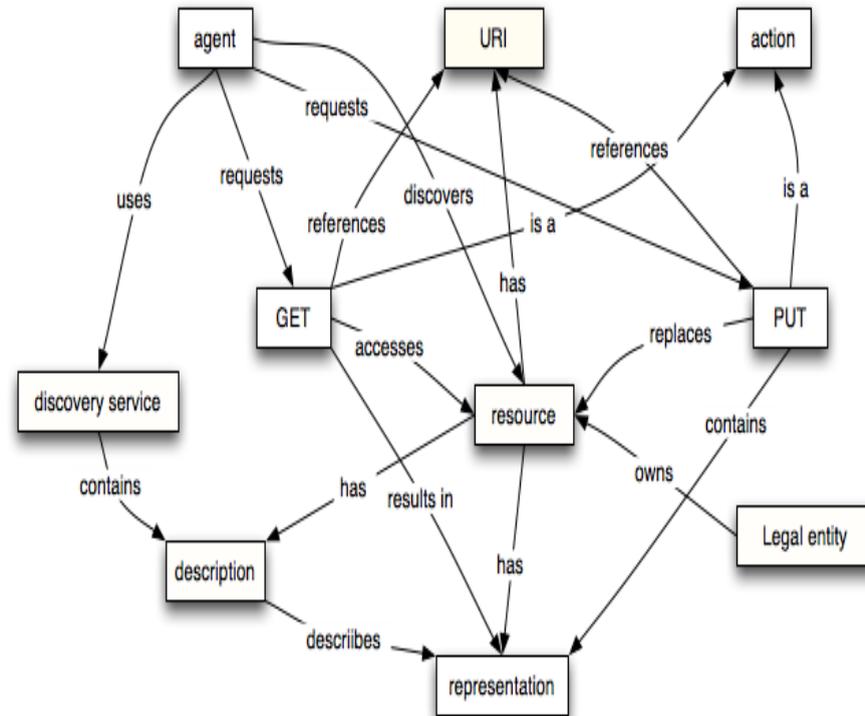
# WS & SOA – Serviceorientiertes Modell

- Service := abstrakte Resource, führt Aufgabe für Person oder Organisation aus
- Service hat Schnittstelle, Beschreibung und Semantik(Aufgabe)
- Choreographie beschreibt Abfolge und Bedingung für zusammenwirken von mehreren Agenten in Choreographie Beschreibungssprache (WS CDL)



# WS & SOA – Ressourcenorientiertes Modell

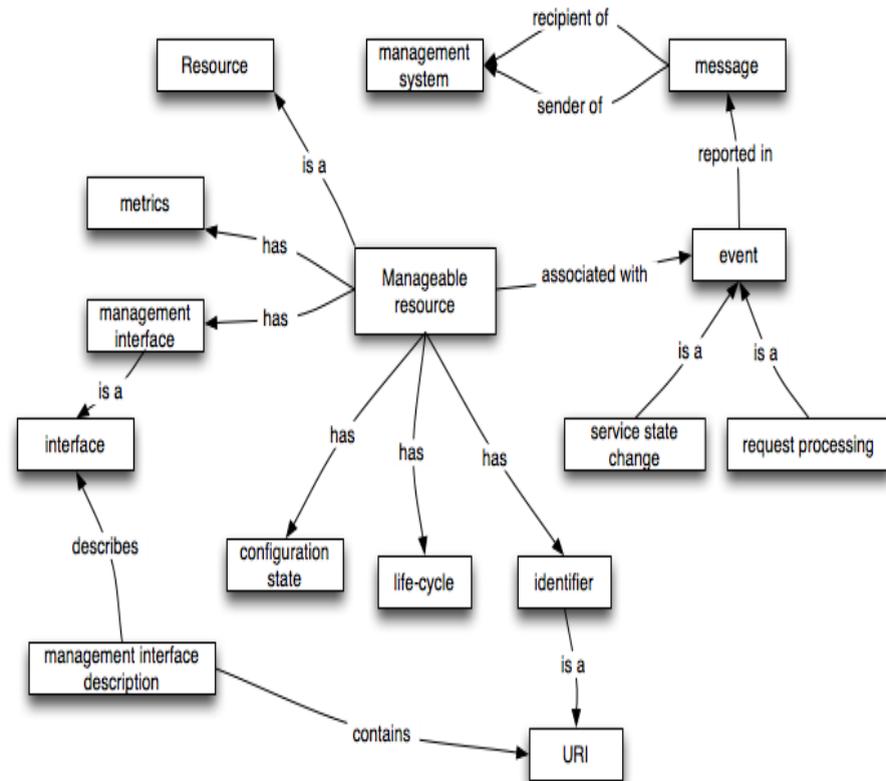
- Ausgelegt als Schicht über ServiceModell
- Resource hat immer Identifikator (URI) and Beschreibung (description)
- Ressourcen können von Agenten entdeckt werden mittels Entdeckungs-Service





# WS & SOA – Management Modell

- Beschreibt Beziehung zwischen eingesetzten Elementen (Ressourcen) und anderen Architekturbestandteilen



# WSDL

WSDL steht für **Web Services Description Language** und wird von Arbeitsgruppen des W3C zusammen mit der Web Services Architektur entwickelt.

Die wichtigsten WSDL-Dokumente sind:

- [W3C03a] World Wide Web Consortium: Web Services Description Language (WSDL) Version 1.2 Part 1: Core Language, W3C Working Draft, Juni 2003, <http://www.w3.org/TR/wsdl12>
- [W3C03b] World Wide Web Consortium: Web Services Description Language (WSDL) Version 1.2 Part 2: Message Patterns, W3C Working Draft, Juni 2003, <http://www.w3.org/TR/wsdl12-patterns>
- [W3C03c] World Wide Web Consortium: Web Services Description Language (WSDL) Version 1.2 Part 3: Bindings, W3C Working Draft, Juni 2003, <http://www.w3.org/TR/wsdl12-bindings>

# WSDL - Eigenschaften

- Eigenes WSDL-Komponenten-Model
- Eigener WSDL-Namensraum *wSDL*
- Unabhängigkeit von spezifischem Format für externe Repräsentation (Serialisierung), default ist aber XML/XML-Schema

# WSDL - Definitions

```
<definitions  
  targetNamespace="xs:anyURI" >  
  <documentation />?  
  [ <import /> | <include /> ]*  
  <types />?  
  [ <message /> | <interface /> | <binding /> | <service /> ]*  
</definitions>
```

- definitions-Element ist Behälter für andere WSDL-Elemente (message, interfaces, bindings, services)
- Ort für targetNamespace-Spezifikation enthaltener Elemente
- Ort für type-Spezifikationen im element types
- Ort für Modularisierung von WSDL-Dokumenten mittels import/include

# WSDL – Nachrichten (message)

```
<definitions> <message  
  name="xs:NCName" >  
  <documentation />?  
  <part />*  
</message> </definitions>
```

- Beschreibung des abstrakten Nachrichtenformats, typischerweise in Ausdrücken von XML-Elementen und XML-Attributen

# WSDL – Teile (part)

```
<definitions> <message> <part  
  name="xs:NCName"  
  element="xs:QName"?  
  type="xs:QName"? >  
  <documentation />?  
</part> </message> </definitions>
```

- Beschreibt die Bestandteile einer Nachricht
- Entweder durch Referenz auf eine Element- oder Typ-Deklartion
- Defaultmäßig XML-Schema, aber auch andere Sprachen erlaubt

# WSDL – Schnittstellen (interfaces)

```
<definitions>
  <interface
    name="xs:NCName"
    extends="list of xs:QName"? >
    <documentation />?
    [ <operation /> | <feature /> | <property /> ]*
  </interface>
</definitions>
```

- interface-Element Gruppiert für einen Service eine Menge von Nachrichtenaustauschen beschrieben in Operationen
- Kann erweitert werden bzw. in einer Ableitungshierarchie enthalten sein

# WSDL – Merkmale (Feature)

```
<feature
  uri="xs:QName"
  required="xs:boolean"? >
  <documentation />?
</feature>
```

- Merkmale ermöglichen externe Bedingungen und Regeln (spezifiziert über URI) für Nachrichtenaustausche hinzuzufügen
- Mehr als ein Merkmal kann gleichzeitig vorhanden sein, alle gelten zugleich
- Required besagt ob ein Requester die Regeln beachten MUSS, oder nur KANN

# WSDL - Eigenschaften (Property)

```
<property
  uri="xs:QName"
  required="xs:boolean"? >
  <documentation />?
  [ <value /> | <constraint /> ]
</property>
```

- Eigenschaften (Properties) beschreiben Bedingungen für Merkmale (Features)
- Bedingungen des Eigenschafts-Werts mittels `<constraint />`-Element angeben
- Konstanten mit `<value />` angegeben

# WSDL - Schnittstellen Operationen (Interface Operation)

```
<definitions> <interface> <operation  
  name="xs:NCName"  
  pattern="xs:anyURI" >  
  <documentation />?  
  [ <feature /> | <property /> |  
    [ <input /> | <output /> | <infault /> | <outfault /> ]+ ]*  
</operation> </interface> </definitions>
```

- Operationen modellieren den Nachrichtenaustausch(inputs, outputs) für ein Interface eines Service
- Stellt Verbindung zu Nachrichtenaustauschmuster her, welches exakte Abfolge und Anzahl (Kardinalität) bestimmt
- Enthält Regeln für die enthaltenen Nachrichten-Verweise (<input />, <output /> ..)

# WSDL - Nachrichten Verweise (Message Reference)

```
<definitions> <interface> <operation>  
  <input  
    name="xs:NCName"?  
    message="xs:QName" >  
  <documentation />?  
</input>  
  <output  
    name="xs:NCName"?  
    message="xs:QName" >  
  <documentation />?  
</output>  
</operation> </interface> </definitions>
```

Nachrichten Verweis: verbindet Operation mit konkreten benannten Nachrichten (messages)

# WSDL - Fehler Verweise (Fault Reference)

```
<definitions> <interface> <operation>
  <infault
    name="xs:NCName"
    messages="list of xs:QName" >
  <documentation />?
</infault>
  <outfault
    name="xs:NCName"
    messages="list of xs:QName" >
  <documentation />?
</outfault>
</operation> </interface> </definitions>
```

Fehler Verweis: verbindet Operation mit konkreten benannten Fehler-Nachrichten (fault messages)

# WSDL - Bindungen (Binding)

```
<definitions> <binding
  name="xs:NCName"
  interface="xs:QName" >
  <documentation />?
  [ <feature /> | <property /> | <operation /> ]*
</binding> </definitions>
```

- Spezifiziert konkrete Details über Dienst-Schnittstellen und deren Operationen, über benutzen Protokolle und konkrete Nachrichtenformate

# WSDL - Bindungs Operationen (Binding Operation)

```
<definitions> <binding> <operation  
  name="xs:NCName" >  
  <documentation />?  
  [ <feature /> | <property /> |  
    [ <input /> | <output /> | <infault /> | <outfault /> ]+  
  ]*  
</operation> </binding> </definitions>
```

- Beschreibt konkret Bindung einer Schnittstellen-Operation an ein spezifisches Nachrichtenformat
- Bindungs-Operationen sind lokal innerhalb einer Bindung

# WSDL - Bindungs Nachrichten Verweise (Binding Message Reference)

```
<definitions> <binding> <operation>  
  <input name="xs:NCName"? > <documentation />?  
  </input>  
  <output name="xs:NCName"? > <documentation />?  
  </output>  
  <infault name="xs:NCName" > <documentation />?  
  </infault>  
  <outfault name="xs:NCName" > <documentation />?  
  </outfault>  
</operation> </binding> </definitions>
```

- Beschreibt konkrete Bindung von Nachricht einer Operation an konkretes Nachrichtenformat
- Bindungs Nachrichten Verweise sind lokal innerhalb einer Operation Bindung

# WSDL - Dienste (Service)

```
<definitions> <service
  name="xs:NCName"
  interface="xs:QName"
  targetResource="xs:anyURI"? >
  <documentation />?
  <endpoint />*
</service> </definitions>
```

- Service beschreibt eine Schnittstelle(interface) und entsprechende Endpunkte

# WSDL - Endpunkte (Endpoint)

```
<definitions> <service> <endpoint  
  name="xs:NCName"  
  binding="xs:QName" >  
  <documentation />?  
</endpoint> </service> </definitions>
```

- Endpunkt gibt die Details zu verfügbaren Implementierungen eines Service an
- Endpunkte sind lokal innerhalb einen Service angegeben
- Ein Endpunkt muss eindeutig innerhalb eines Service sein

# WSDL - Typen (Types)

```
<definitions> <types>  
  <documentation />?  
  [extension elements]*  
</types> </definitions>
```

- Enthält alle Nachrichten- und Fehler-Datentypen
- Wird (wie im Defaultfall) XML als externe Repräsentationsprache verwendet, so werden die Datentypen als XML-Schema-Elemente dargestellt

# Zusammenfassung und Ausblick

- SOAs stellen das heutige Mittel der Wahl zur Verknüpfung eines verteilten Systems dar
- Web Services und WSDL sind eine angemessene Architektur und Infrastruktur-Standards
- Was macht man nun mit Web Services?

Antwort: Konstruktion von Systemen für Arbeits- und Geschäftsprozess-orientierte Kombination von verteilt angebotenen Diensten; dazu:

- Modellierungs-Sprachen und Prozess-Beschreibungs Standards
- Petri Netze zur sicheren formal fundierten Prozess-Modellierung