
Service

Oriented

Architecture

Von:

Sascha Becker

Dennis Griesa

Björn Grothe

Samira Joost

Christoph Koehler

Christian Schoppmeyer

Henning Tilmann

- SOA-Grundlagen
 - Definition
 - Bestandteile
 - Schritte zur Implementierung einer SOA
 - Soa vs. Corba
- Realisierung durch Web Services
 - Definition
 - Standards
- Fazit

„By 2008, more than 60 percent of enterprises will use SOA as a ‚guiding principle‘ when creating mission-critical applications and processes.“

Gartner Group (internationales IT-Forschungsunternehmen)

Definition

„A system architecture, that represents diverse and heterogeneous applications as reusable and through an open interface accessible services.“

German EAI Forum

„A service-oriented architecture is essentially a collection of services. These services communicate with each other.

The communication can involve either simple data passing or it could involve two or more services coordinating some activity.

Some means of connecting services to each other is needed.“

Gartner Group

„Eine Anwendungsarchitektur, in der alle Funktionen als unabhängige Services mit wohldefinierten, aufrufbaren Schnittstellen vorliegen, so dass eine Auswahl – in einer sinnvollen Reihenfolge aufgerufen – einen Geschäftsprozess abdecken.“

plenum Management Consulting GmbH

- ist ein Software-Design Ansatz
- ist Technologie unabhängig
- ist die Evolution von Komponenten-Architekturen
- ist kein grundlegend neuer Ansatz im Software-Design (Beispiel: Corba)

- SOA beschreibt Funktionalität als modulare und wieder verwendbare Services/Dienste
- Anwendungen werden aus Diensten “zusammengesetzt”
- lose Koppelung der Dienste untereinander
= keine starken logischen oder physischen Abhängigkeiten

- ermöglichen, Anwendungen dynamisch durch Austausch zu modifizieren, zu erweitern oder zu optimieren
- sind wiederverwendbar
- haben wohldefinierte, standardisierte Schnittstellen, unabhängig von der zugrunde liegenden Implementierung

- Online-Brokerage – der Abschluss von Wertpapiergeschäften über das Internet – kann in einer SOA auf folgenden Diensten aufsetzen:
 - einem Wertpapierdienst, der handelbare Wertpapiere (Aktien, Fonds, ...) nennt
 - einem Marktdatendienst, der zu einem Wertpapier aktuelle Börsenkurse liefert
 - einem Depotdienst, der Zugriff auf das Wertpapierdepot des Anwenders erlaubt

- Außerdem werden benötigt:
 - einem Orderdienst, der Kauf-/Verkaufsaufträge des Anwenders an eine Börse übermittelt
 - einem Archivdienst, der alle Daten der Transaktion revisionssicher archiviert
- Diese Dienste werden aus einer Webanwendung heraus aufgerufen.

Bestandteile von SOA

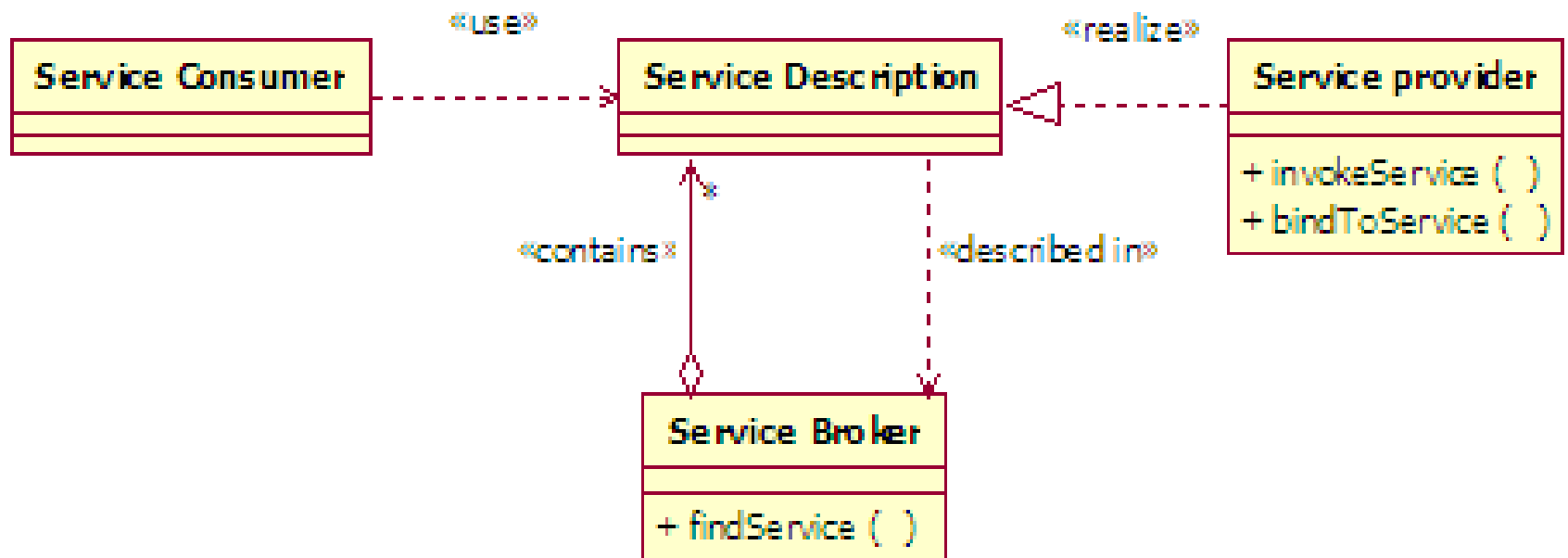
- erstellt, kauft oder mietet einen Service
- stellt den Service zur Verfügung
- stellt Spezifikation des Services zur Verfügung
 - Allgemeine Service Informationen
 - Beschreibung des Service Interface

- ist optional
- verwaltet Services eines Service-Providers über eine Datenbank, die sogenannte Service-Registry
- ermöglicht dass die katalogisierten Services, sowie die dazu verfügbare Dokumentation von „außerhalb“ zugreifbar sind

- Darstellung angebotener Services meist durch ein Web-Frontend
- Grundlegende Funktionalitäten:
 - Registrieren neuer Services
 - Ändern, löschen registrierter Services
 - Suche nach verfügbaren Services

- (sucht und) benutzt einen Service

Consumer Obtains Service description from Broker or directly from Provider (Broker is optional)



Schritte zur Implementierung von SOA

- Erfassen aller bereits (im Unternehmen) verfügbaren Dienste
- Beschreibung bzw. Spezifikation für Serviceangebote entwickeln, als Standard etablieren
- Erfassen und strukturieren der relevanten Informationen

- Implementieren der Services
- Füllen (Registrierung) des Katalogs mit entsprechenden Serviceangeboten und Schaffung von Suchmöglichkeiten
- Katalog regelmäßig aktualisieren (evtl. Schulung der mit dem Katalog arbeitenden Personen)

SOA vs. Corba

- Common Object Request Broker Architecture
- wird von der Object Management Group (OMG) entwickelt
- definiert plattformübergreifende Protokolle und Dienste
- ist nicht an eine bestimmte Programmiersprache gebunden
- vereinfacht wie SOA das erstellen verteilter Anwendungen in heterogenen Umgebungen

- SOA definiert Services, Corba Objekte
- formale Schnittstelle zu den Objekten durch die Interface Definition Language (IDL)
- Schnittstellenbeschreibung wird dann in das Objektmodell der benötigten Programmiersprachen umgesetzt
=>enge Kopplung
- SOA ist als Evolution von Corba zu verstehen

SOA Realisierung

durch Web Services

- ist eine Software-Anwendung
- wird mit einem URI eindeutig identifiziert
- Schnittstellen werden als XML-Artefakte definiert, beschrieben und gefunden
- unterstützt die direkte Interaktion mit anderen Services
- kommunizieren untereinander durch XML-basierende Nachrichten, die über Internet-Protokolle ausgetauscht werden

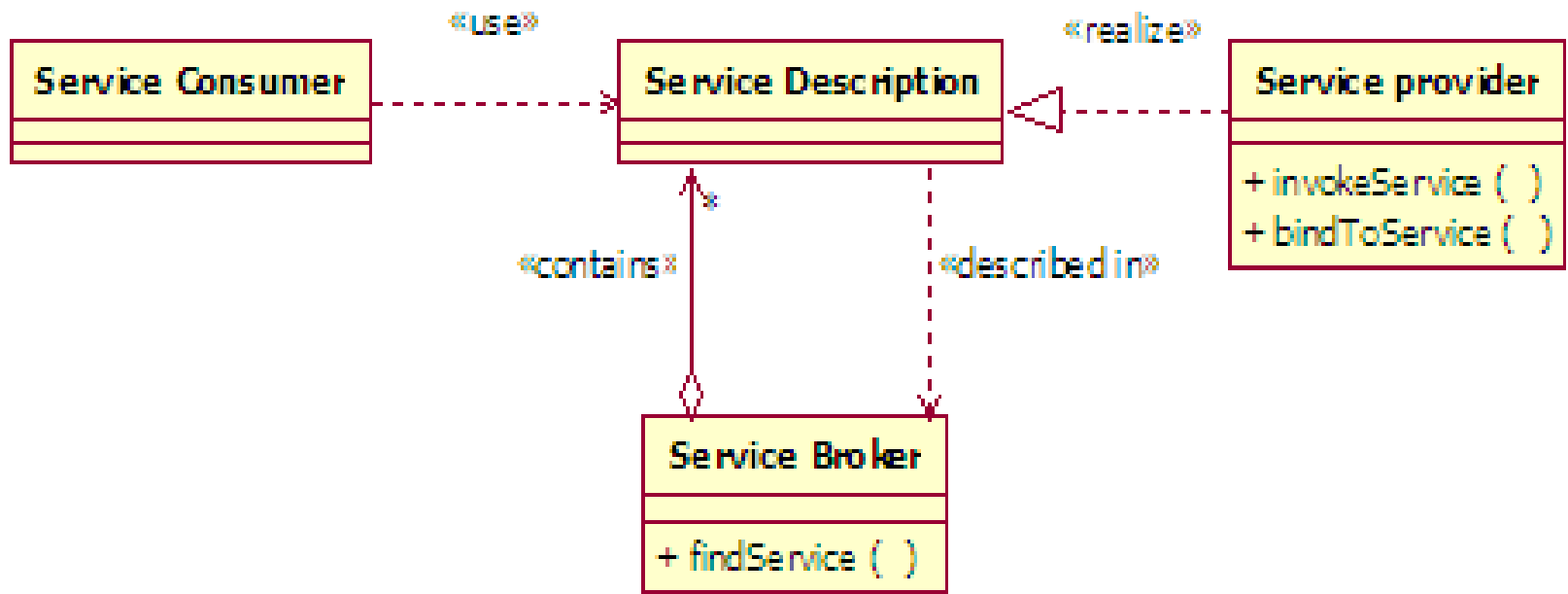
- SOA beschreibt eine Architektur, keine Technologie
- Einen Web Service zu implementieren, heisst nicht SOA-Anforderungen zu erfüllen
- Es gibt auch andere Technologien zur Umsetzung von SOA

- dennoch:
 - werden seit vielen Jahren als tragende Technologie zur Entwicklung einer SOA diskutiert
 - WS werden von vielen als der ideale Ansatz zur Umsetzung der SOA angesehen (u.a. Gartner Group)

Zum Vergleich nochmal:

Elemente einer SOA

Consumer Obtains Service description from Broker or directly from Provider (Broker is optional)



Zum Vergleich

Funktionsweise Web Service



- Anbieter veröffentlicht in einem Verzeichnis Beschreibung seiner Dienste
- Konsument durchsucht Verzeichnis, wählt gewünschten Dienst
- Dynamische Anbindung des Konsumenten an den Anbieter, Konsument greift auf Methoden zu

Standards

- Vier Standards die auf XML basieren
 - UDDI
 - WSDL
 - SOAP oder XML-RPC

- Universal Description, Discovery and Integration
- stellt mit Hilfe einer SOAP-Schnittstelle einen Verzeichnisdienst bereit
- zentrale Rolle im Web Service
- ermöglicht das dynamische Finden von Web Services
- stellt drei „Arten“ von Informationen bereit:
 - „White Pages“
 - „Yellow Pages“
 - „Green Pages“

- Informationen in den „White Pages“:
 - Namensregister
 - Auflistung der Anbieter mit allen Detailangaben
 - Kontaktinformationen (Telefon, E-Mail...)
- Informationen in den „Yellow Pages“
 - Branchenverzeichnis
 - Spezifische Suche nach Diensten
 - verweist auf „White Pages“
- Informationen in den „Green Pages“
 - Geschäftsmodell des Unternehmens
 - Technische Details zu den angebotenen Web Services

Navigationsmenu

Einstiegsmöglichkeiten

Allgemeine Informationen

Schnittstellenbeschreibung

Voraussetzung für den Einsatz

Qualitätseigenschaften des Service

Kategorien als mögliche Suchkriterien

Service-Name: "Timb.Kunde suchen"

Allgemeine Informationen:

Beschreibung: Dieser Service sucht nach Kundendaten.
Ansprechpartner: nicht vorhanden

Schnittstellen

Der Service hat 1 Schnittstelle

Schnittstelle 1

Name: BKI_KD_PortType

Beschreibung: [weitere Informationen...](#)

Zugriffspunkt: <http://irgendwas.com>

Service Typ: [CRM-C.Kunde suchen](#)
[CRM-B.Kunde suchen](#)

Abstimmung

Service ruft auf: grafisch
XML-Format

Prozessablauf-Beschreibung (BPEL)

Qualität

Kundeneinschätzung

Nutzungshäufigkeit

Performance

Ausfallverhalten

Suche Services Administration

Namensuche Katalogsuche Serviceliste about Hilfe

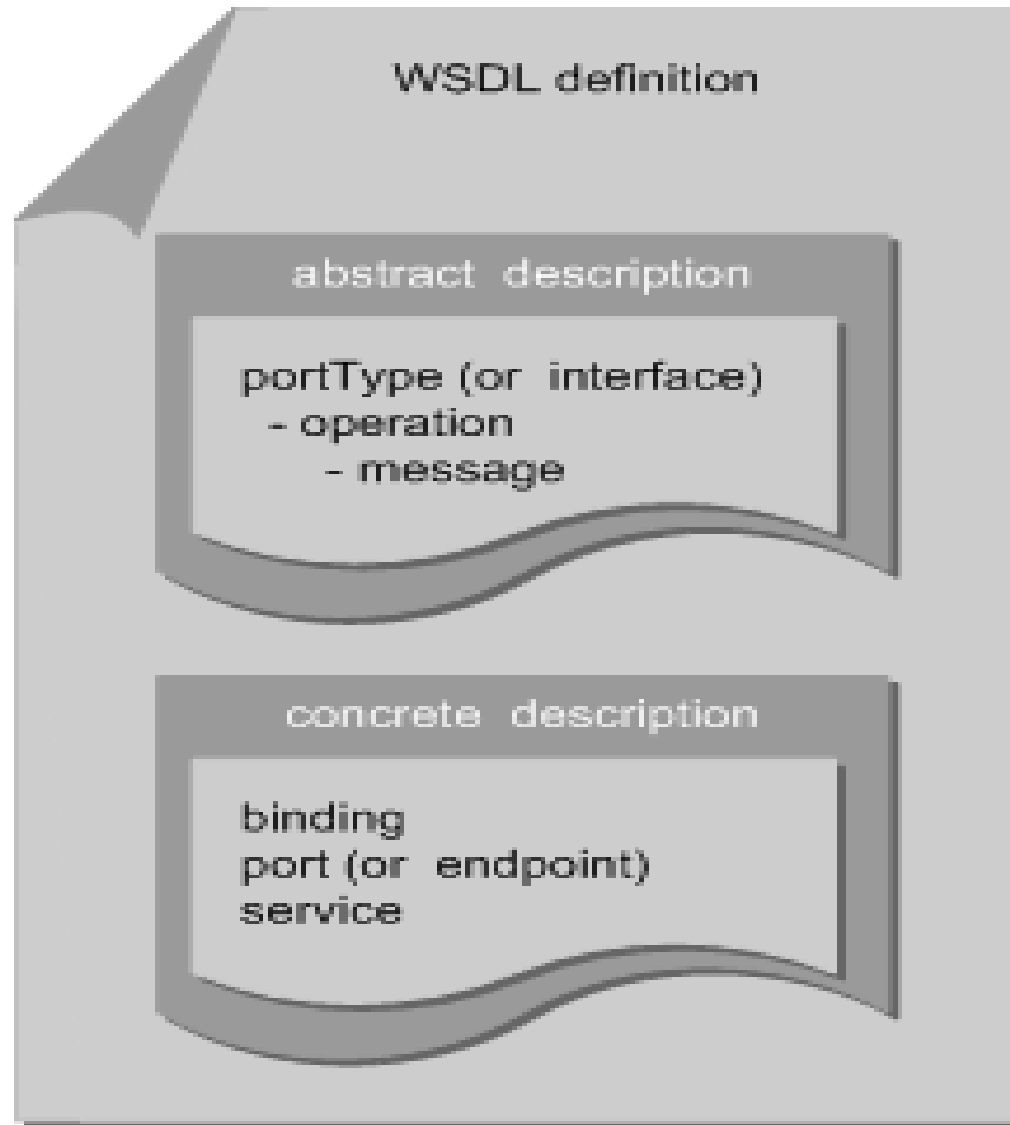
Service-Suche - Suche über den Service-Katalog nach Kategorien

- Kategorien
 - Anwendung
 - AL
 - Call Center Infrastruktur
 - CRM
 - CRM-B
 - CRM-C
 - CRM-TIMB
 - FE2000
 - EE2001
 - INDIV
 - Yus-GK
 - DWH
 - KIQ
 - MVW
 - MWD
 - Weblösungen
 - Daten
 - Funktion
 - Organisation
 - Produkt
 - Prozess

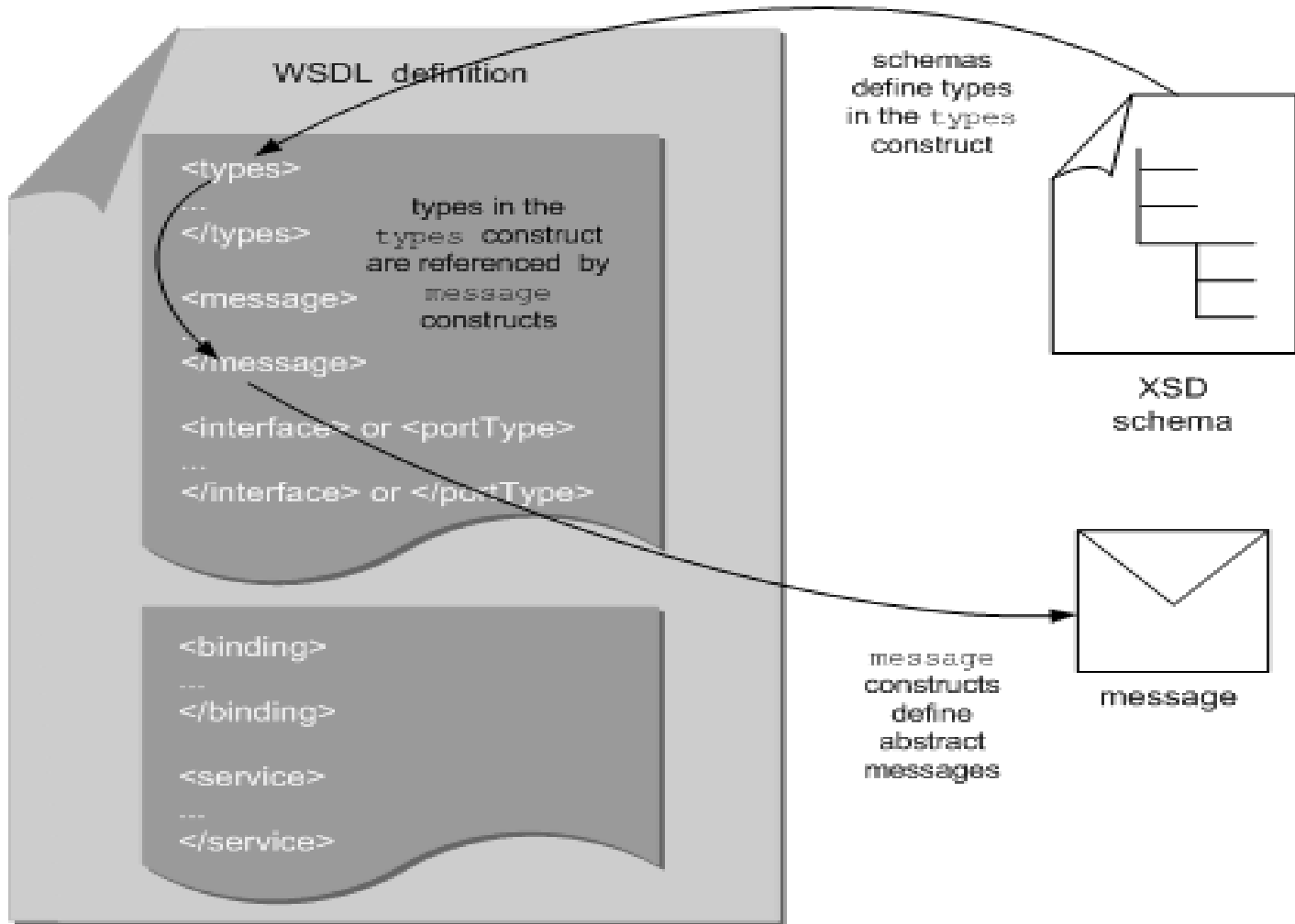
Service Search Engine

- Web Service Definition Language
- ist eine Metasprache
- ist ein plattform-, programmiersprachen- und protokollunabhängiger XML-Standard
- definiert formal die Funktionalität eines Web Service die „von außen“ zugreifbar ist:
 - Funktionen, Parameter und Rückgabewerte
 - Details zum Deployment
- liefert somit die Information für die „Green Pages“

Wie ist WSDL aufgebaut?



Wie ist WSDL aufgebaut?



```
<?xml version="1.0" encoding="UTF-8"?>  
<definitions name="EchoService"  
  targetNamespace="http://www.echoserviceservice.com/EchoService-interface"  
  xmlns="http://schemas.xmlsoap.org/wsdl/">  
  
</definitions>
```

root-Element der
Service-Definition

abstract description -> message

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="EchoService"
  targetNamespace="http://www.echoserviceservice.com/EchoService-interface"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <message name="InechoRequest">
    <part name="meth1_inType1" type="xsd:string"/>
  </message>
  <message name="OutechoResponse">
    <part name="meth1_outType" type="xsd:string"/>
  </message>

</definitions>
```

Nachrichten definieren



abstract description → interface, operation

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="EchoService"
  targetNamespace="http://www.echoserviceservice.com/EchoService-interface"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <message name="InechoRequest">
    <part name="meth1_inType1" type="xsd:string"/>
  </message>
  <message name="OutechoResponse">
    <part name="meth1_outType" type="xsd:string"/>
  </message>

  <interface name="EchoService">

    <operation name="echo">
      <input message="InechoRequest"/>
      <output message="OutechoResponse"/>
    </operation>

  </interface>
```

Schnittstelle



Methode



concrete description -> binding

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="EchoService"
  targetNamespace="http://www.echoserviceservice.com/EchoService-interface"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  ...

  <binding name="EchoServiceBinding" type="EchoService">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  </binding>
  ...
</definitions>
```

Verknüpfung/Bindung
des „EchoService an
das SOAP-Protokoll

concrete description -> binding

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="EchoService"
  targetNamespace="http://www.echoserviceservice.com/EchoService-interface"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  ...

  <binding name="EchoServiceBinding" type="EchoService">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

    <soap:operation soapAction="urn:echoservice-service"/>
    <input>
      <soap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="urn:echoservice-service"
        use="encoded"/>
    </input>
    <output>
      <soap:body
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="urn:echoservice-service" use="encoded"/>
    </output>
  </operation>

</binding>

  ...

</definitions>
```

Verknüpfung/Bindung
des „EchoService“ an
das SOAP-Protokoll

concrete description -> service, port

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="EchoService"
  targetNamespace="http://www.echoserviceservice.com/EchoService-interface"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  ...

  <service name="EchoService">
    <port binding="EchoServiceBinding" name="EchoServicePort">
      <soap:address location="http://localhost:8080/soap/servlet/rpcrouter"/>
    </port>
  </service>
  ...
</definitions>
```

service-Element gibt mit Hilfe des port-Elements die physikalische Adresse des Service an

- ursprünglich: **S**imple **O**bject **A**ccess **P**rotocol, seit Version 1.2 Eigenname
- ist ein Internetprotokoll
- gedacht für den Austausch von Daten und RPCs
- nutzt XML für Datenaustausch (messages)
- nutzt Internetprotokolle (meist Kombination von HTTP und TCP)

- ist nach dem Head-Body Pattern modelliert (vgl. HTML)
- eine nachricht wird in einem „Container“ verschickt, dem „SOAP-Envelope“

Aufbau des Envelope-Containers

- **Header** enthält Metainformationen (optional)
 - Routing der Nachricht
 - eventuelle Verschlüsselung
 - Zugehörigkeit zu einer Transaktion
- **Body** enthält Nutzdaten
 - müssen vom Empfänger interpretiert werden
 - z.B. als RPC, als Rückgabewert eines RPCs

Anfrage an einen Server

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <validate soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns="urn:CardValidator">
      <number xsi:type="xsd:string">1234 5678 9876 5432</number>
      <valid xsi:type="xsd:string">12/08</valid>
    </validate>
  </soapenv:Body>
</soapenv:Envelope>
```

Methode

Service

Argumente

mögliche Antwort des Servers

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <soapenv:Body>

    <validateResponse soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns="urn:CardValidator">
      <addReturn xsi:type="xsd:string">true</addReturn>
    </validateResponse>

  </soapenv:Body>
</soapenv:Envelope>
```

Rückgabewert



Fazit

- erstellen von verteilten Anwendungen in heterogenen Umgebungen
- keine Bindung an eine bestimmte Programmiersprache
- Verhindern von redundanten Entwicklungen
- Reduktion von Investitionskosten durch Wiederverwendung

- macht keine Aussagen über Sicherheit
- Identifizierung der richtigen Granularität von Services
 - zu klein: hoher Entwicklungsaufwand
 - zu groß: erfüllt nicht das Ziel Service
- weniger Performanz durch weit verbreiteten Ansatz XML zu verwenden
=>Overhead, Parsen zeitaufwändig...

- CIO Magazine, Jan. 15, 2004
- Informatik Spektrum, Oct. 7, 2005
- Jeff Hanson, *Coarse-grained Interfaces Enable Service Composition in SOA*, JavaOne, August 29, 2003
- IBM White Paper, *New to SOA and Web Services*, www.DeveloperWorks.com

- Meta Group White Paper, *Practical Approaches to Service-Oriented Architecture*, November 2003
- <http://de.wikipedia.org>
- <http://www.w3.org>