

# Wissenschaftliche Vertiefung Web Services

**Hochschule Esslingen**  
University of Applied Sciences  
**Fakultät Informationstechnik**

Studiengang Softwaretechnik  
und Medieninformatik

Esslingen, 22. Januar 2016  
Simon Schneider



# Agenda

---

1. Einführung
2. Serviceorientierte Architektur
3. SOAP Web Service
4. Standards und Protokolle von SOAP Web Services
5. Bewertung

# 1. Einführung

---

- Kommunikation zwischen verteilten Systemen ist keine neue „Erfindung“ von Web Services:
  - RMI, COM, DCOM und CORBA (Vorgänger)
  - REST, XML-RPC (Alternativen)
- Die Nachteile dieser Technologien:
  - Schlechte Interoperabilität
  - Komplizierte Implementierung
  - Keine standardisierte Schnittstellen
  - Verwendung von proprietären und binären Protokollen



---

## Vorgänger und Alternativen

# 1. Einführung

---

- Verwendung von im Web etablierten Standards
- Programmiersprachen- und Plattformunabhängigkeit
- Kopplung von Geschäftsprozessen über Unternehmensgrenzen hinweg
- Kommunikation mit verteilten, heterogenen Systemen



---

## Die Idee

# 1. Einführung

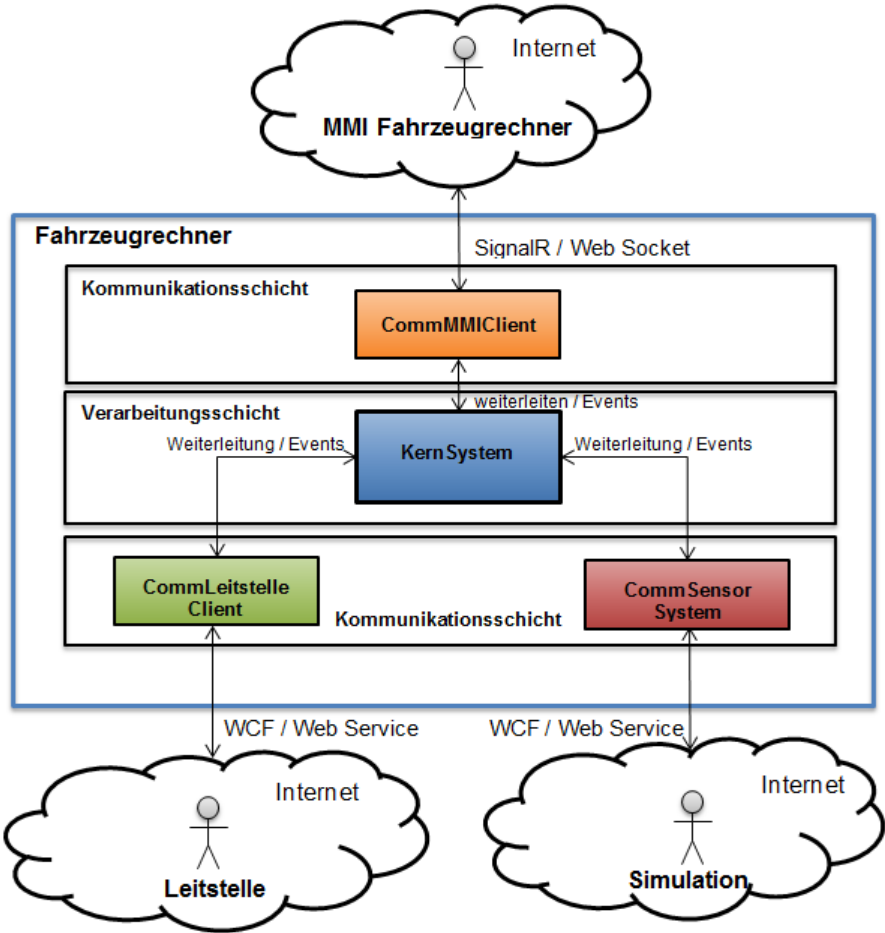
---

- Zusammenschluss von im Web standardisierten Technologien
  - Software-Komponenten
  - Verteilte, heterogene Systeme
  - Kommunikation über ein Netzwerk (Web)
  - Maschine-zu-Maschine-Interaktion
- ➔ Web Services sind eine mögliche Realisierungsform für eine Kommunikation in einer serviceorientierten Architektur.

---

## Was sind Web Services?

# 1. Einführung



# Ein Beispiel

## 2. Serviceorientierte Architektur (SOA)

---

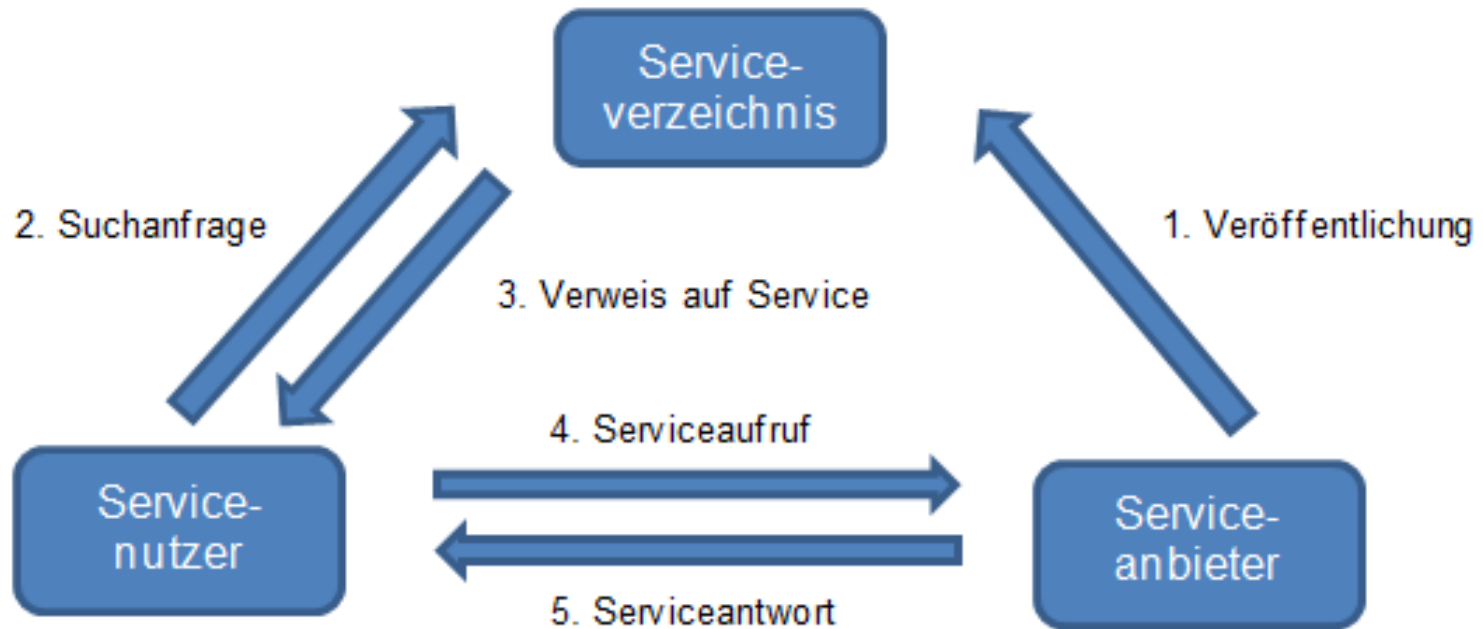
- Architekturmuster für verteilte Systeme.
  - Services als Software-Komponenten bilden Geschäftsprozesse oder Teile eines Geschäftsprozesses ab.
  - Durch den Einsatz von Services entstehen die Rollen:
    - **Serviceanbieter**
    - **Servicenutzer** und
    - ein optionales **Serviceverzeichnis**
- ➔ Funktionalitäten werden in einer **SOA** über Anwendungsgrenzen hinweg als lose gekoppelte, unabhängige und austauschbare Dienste angeboten.

---

## Eigenschaften einer SOA

## 2. Serviceorientierte Architektur (SOA)

---



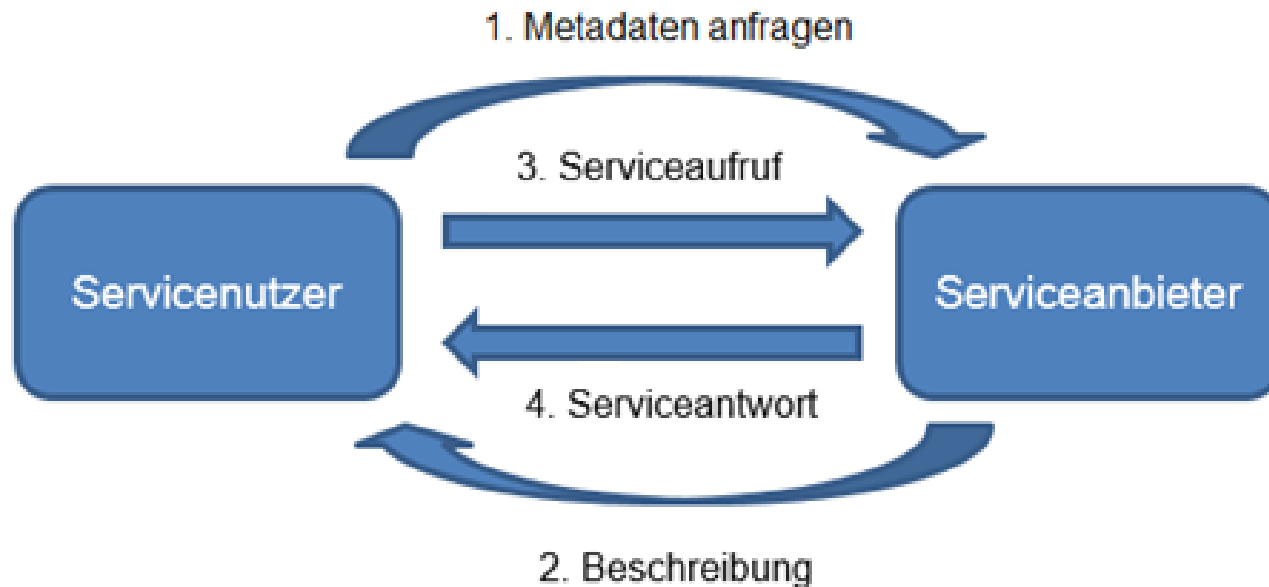
---

### Zusammenarbeit der Rollen mit Serviceverzeichnis



## 2. Serviceorientierte Architektur (SOA)

---



---

### Zusammenarbeit der Rollen ohne Serviceverzeichnis

## 2. Serviceorientierte Architektur (SOA)

---

- Verteilung
  - Geschlossenheit
  - Zustandslos
  - Lose Kopplung
  - Orchestrierbarkeit / Choreographie
  - Standardisierte Schnittstellen
  - Plattform- / Programmiersprachenunabhängigkeit
  - Verzeichnisdienst
- ➔ Diese Schlüsselmerkmale sind nach verbreiteter Meinung ein wichtiger Bestandteil von Web Services in einer SOA

---

## Anforderungen an Web Services

## 3. SOAP Web Service

---

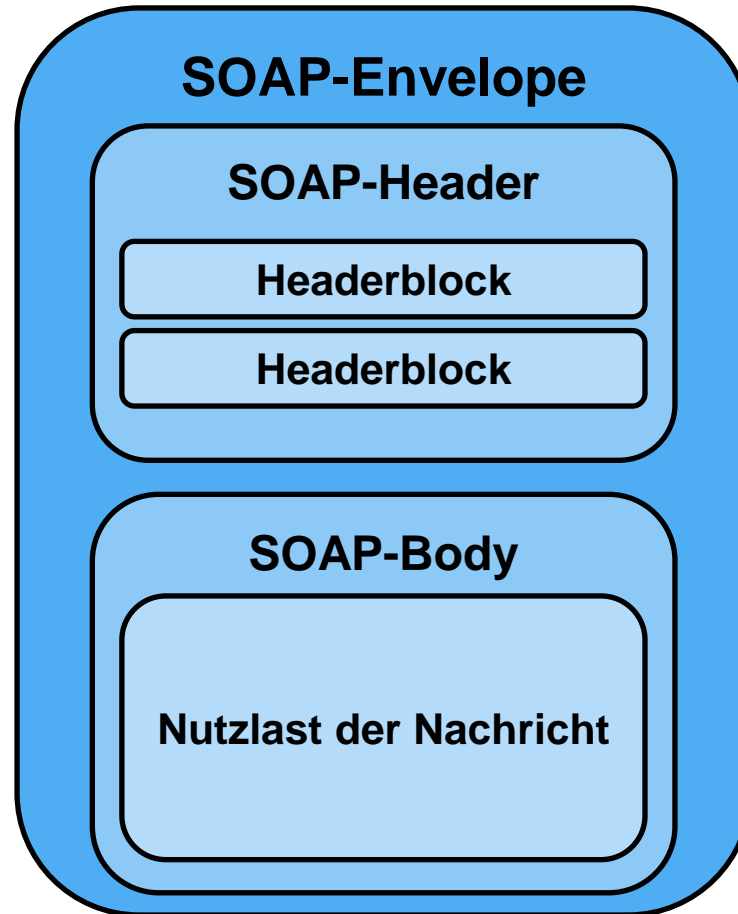
- Kommunikationsprotokoll zum Datenaustausch und RPC
  - XML-basiertes Protokoll
  - Unabhängig vom Übertragungsprotokoll
  - Schnittstellenbeschreibung durch WSDL
  - Standard erweiterbar durch WS-Spezifikationen
  - W3C-Standard
- ➔ Ein SOAP-Web Service ist durch verschiedene Technologien plattformunabhängig (SOAP), selbstbeschreibend (WSDL) und auffindbar (UDDI).

---

### Eigenschaften

# 3. SOAP Web Service

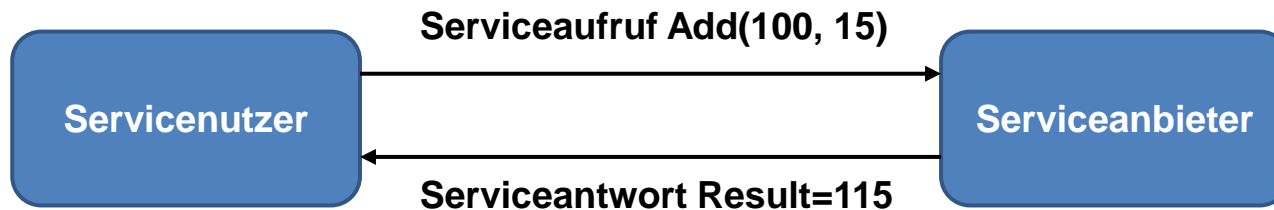
---



---

## Aufbau einer SOAP-Nachricht

# 3. SOAP Web Service



## Serviceaufruf

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <!-- optional -->
  </s:Header>
  <s:Body>
    <Add>
      <n1>100</n1>
      <n2>15</n2>
    </Add>
  </s:Body>
</s:Envelope>
```

## Serviceantwort

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <!-- optional -->
  </s:Header>
  <s:Body>
    <AddResponse>
      <AddResult>115</AddResult>
    </AddResponse>
  </s:Body>
</s:Envelope>
```

## Beispiel einer Kommunikation mit SOAP

## 4. Verwendete Standards und Protokolle

---

- **Web Service Description Language**
- XML-basierte Sprache
- Schnittstellenbeschreibungssprache
- Maschinenlesbar und automatisch verarbeitbar
- Programmiersprachen- und plattformunabhängig
- Aufteilung in abstrakte und konkrete Definition
- ➔ WSDL bietet eine möglichst vollständige Beschreibung der Schnittstellen von Web Services unabhängig von der Implementierung des Serviceanbieters.

---

### Eigenschaften von WSDL

# 4. Verwendete Standards und Protokolle

```
<wsdl:definitions name="CalculatorService">
  <wsdl:types>
    <!-- Definition der verwendeten Datentypen -->
  </wsdl:types>
  <!-- abstrakte Definition der zu übertragenden Nachrichten -->
  <wsdl:message name="ICalculator_Add_InputMessage">
    <wsdl:part name="parameters" element="Add"/>
  </wsdl:message>
  <wsdl:message name="ICalculator_Add_OutputMessage">
    <wsdl:part name="parameters" element="AddResponse"/>
  </wsdl:message>
  <!-- abstrakte Definition der Schnittstelle
  sowie die ausführbaren Operationen -->
  <wsdl:portType name="ICalculator">
    <wsdl:operation name="Add">
      <wsdl:input message="ICalculator_Add_InputMessage"/>
      <wsdl:output message="ICalculator_Add_OutputMessage"/>
    </wsdl:operation>
  </wsdl:portType>
```

## Abstrakter Teil:

Definition der ausgetauschten Nachrichten und der notwendigen Datentypen unabhängig von Kommunikations- und Transportmechanismen.

```
<!-- bindet eine abstrakte Schnittstelle an ein konkretes Übertragungsprotokoll -->
  <wsdl:binding name="BasicHttpBinding_ICalculator" type="ICalculator">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Add">
      <soap:operation soapAction="http://tempuri.org/ICalculator/Add" />
    </wsdl:operation>
  </wsdl:binding>
  <!-- gibt den Namen des Service an und enthält alle ports -->
  <wsdl:service name="CalculatorService">
    <!-- weist einem binding eine konkrete Adresse zu -->
    <wsdl:port name="BasicHttpBinding_ICalculator"
      binding="tns:BasicHttpBinding_ICalculator">
      <soap:address location="http://localhost:Port/CalculatorService"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

## Konkreter Teil:

Beschreibt, wie der Service genutzt werden kann, und legt eine konkrete Bindung an Kommunikations- und Transportmechanismen fest.

## Beispiel eines WSDL-Dokuments

## 4. Verwendete Standards und Protokolle

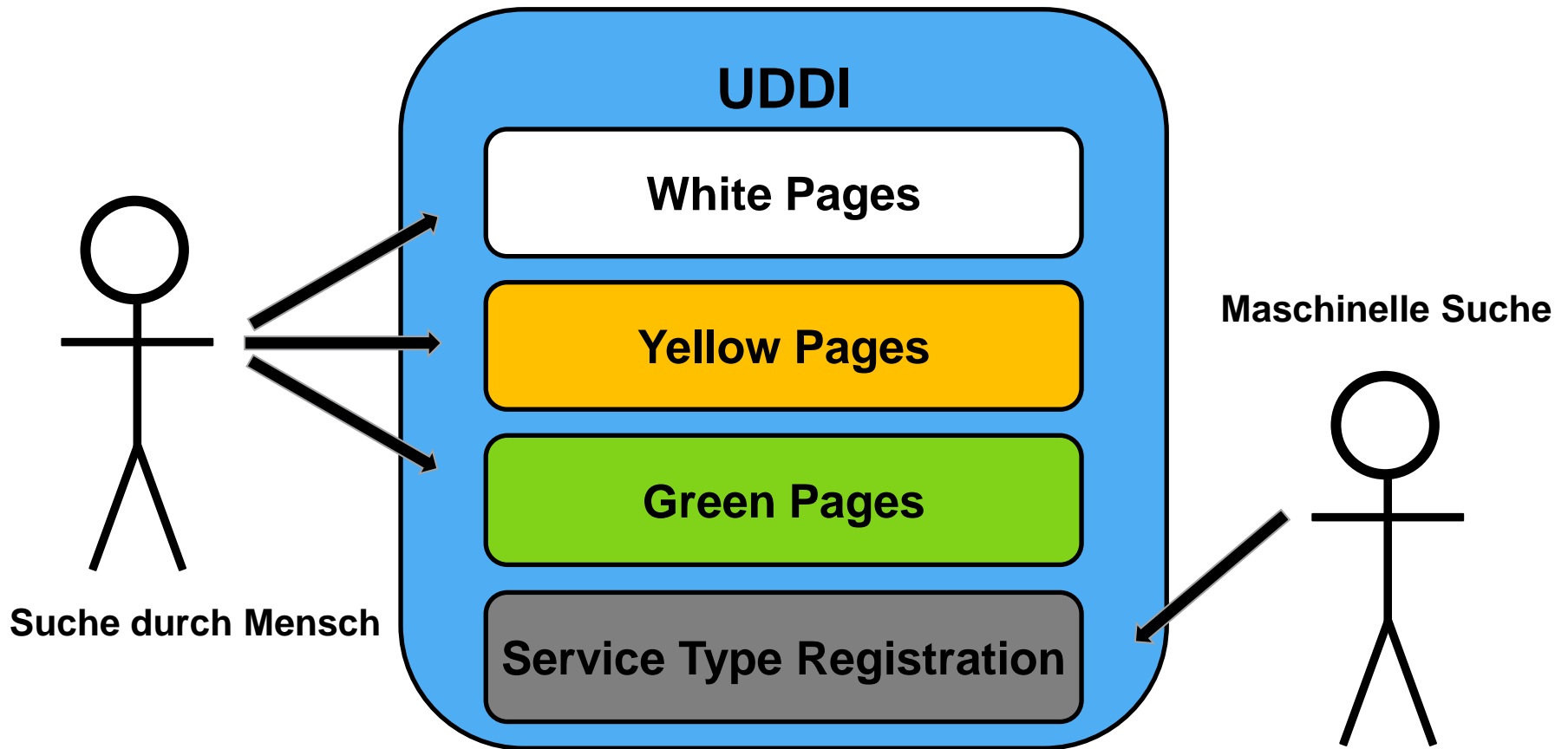
---

- **Universal Description, Discovery and Integration**
  - Serviceverzeichnis zum Veröffentlichen und Auffinden von Web Services.
  - Globale UDDI-Verzeichnisse werden heute praktisch nicht mehr eingesetzt.
  - Lokale UDDI-Verzeichnisse in Firmen zur Unterstützung von Entwicklerteams.
  - Alternative: Web Services verteilen ihre Metabeschreibung selbst (WS-MetadataExchange).
  - ➔ UDDI enthält mehr Informationen als nur Metainformationen. Dadurch können Applikationen sowie menschliche Benutzer darauf zugreifen.
- 

### Eigenschaften von UDDI



# 4. Verwendete Standards und Protokolle



## Inhalte von UDDI

## 5. Bewertung

---



### Vorteile

- Plattformunabhängig
- Programmiersprachenunabhängig
- Unabhängig vom Übertragungsprotokoll
- Standard erweiterbar
- Standardisierte Schnittstellenbeschreibung



### Nachteile

- Performance
- Protokoll-Overhead
- Bandbreite
- Leichtgewichtigkeit

---

## Vor- und Nachteile von SOAP Web Services



Vielen Dank für Ihre Aufmerksamkeit!