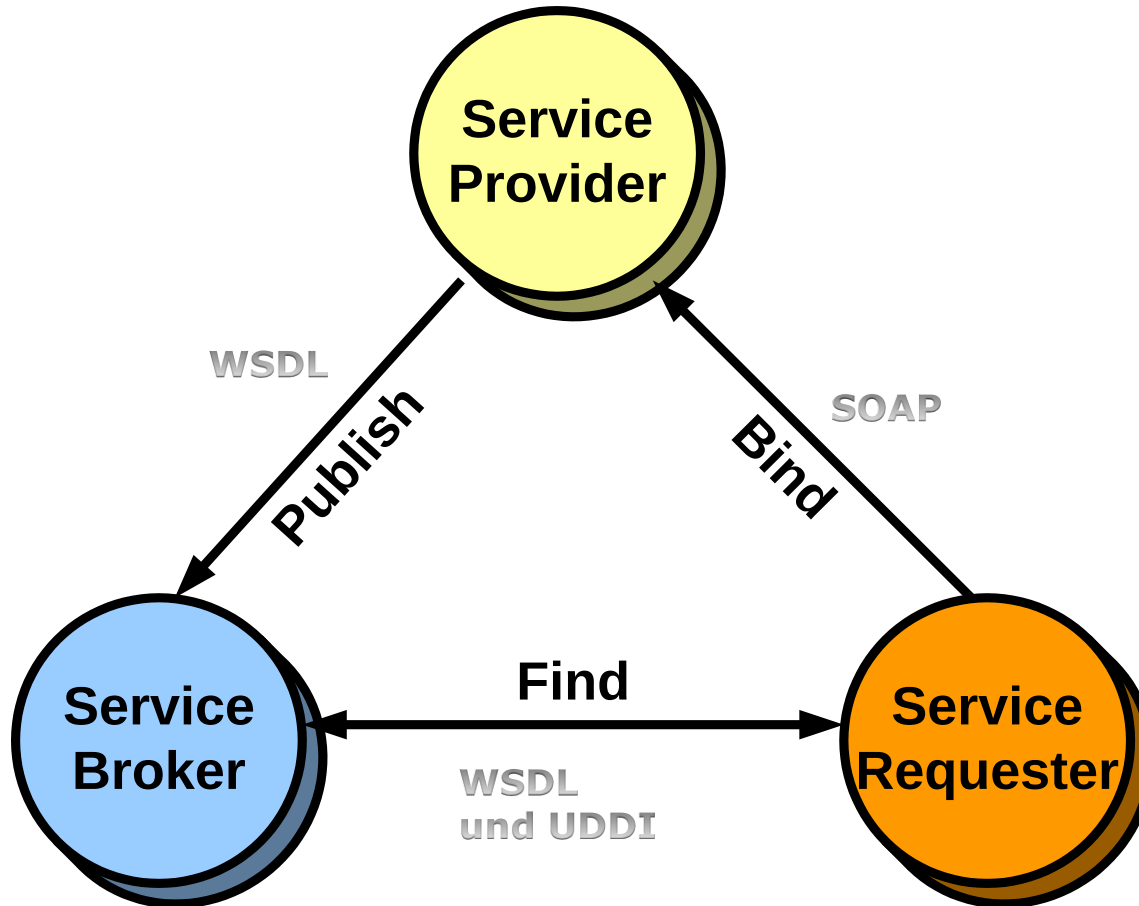




openHPI Web Services: SOAP vs. REST

Wolfram Greis
European Mainframe Academy
Stein am Rhein (CH)

- “Web services are self-contained, modular, distributed, dynamic applications that can be described, published, located, or invoked over the network to create products, processes, and supply chains.”
- Diese Anwendungen können lokal, verteilt, oder Web-basiert sein.
- Web Services bauen auf offenen Standards wie TCP/IP, HTTP, Java, HTML, XML und JSON auf.
- Inhalt wird mit XML transportiert, sofern SOAP genutzt wird. Bei REST ist häufig auch JSON mit im Spiel.

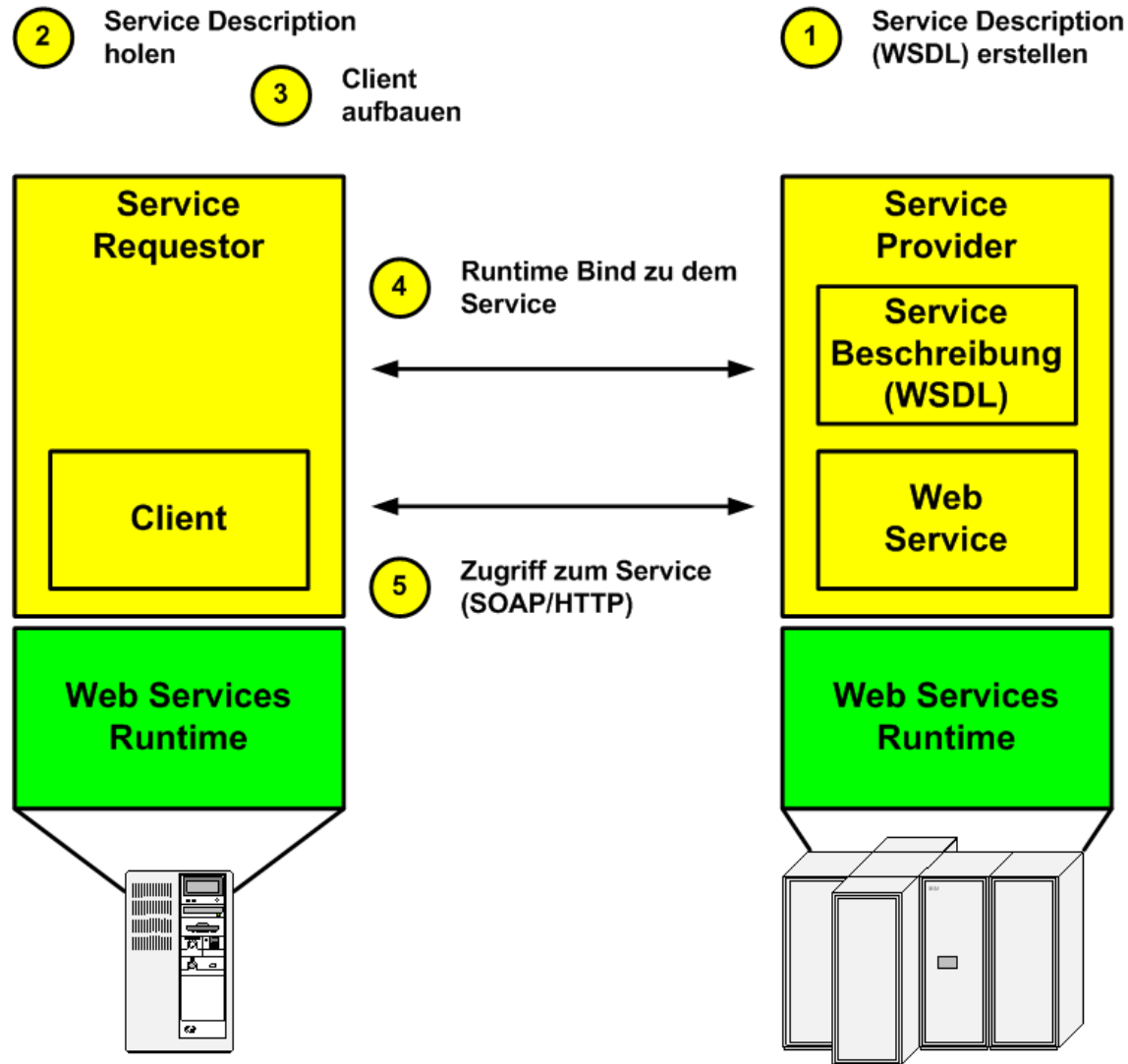


- XML-basierte Beschreibungssprache
- ... so etwas wie eine IDL, wie wir sie von RPCs oder CORBA kennen
- Die Beschreibungen können von Anwendungen analysiert werden
- WSDL Definitionen:
 - Web Service Interfaces
 - Web Service Access Protocol (i.e. SOAP over HTTP)
 - Web Service Endpoints (i.e. Web Service URL)
- www.w3.org/TR/wsdl



- Netzwerk-neutrale Schnittstelle für den Austausch von Informationen zwischen entfernten Anwendungen
- Seit SOAP 1.2 hat das Kürzel keine Bedeutung mehr
- Das XML-basierte Protokoll hat drei Teile:
 - Ein Umschlag (Envelope), um den Inhalt zu beschreiben und wie dieser verarbeitet wird
 - Satz von Regeln für die Definition der Datentypen
 - Konventionen um RPCs und Antworten (Responses) zu repräsentieren
- <http://www.w3.org/TR/soap/>





```
POST /soapsamples/servlet/rpcrouter HTTP/1.0
Host:localhost
Content-Type:text/xml;charset=utf-8
Content-Length:460
SOAPAction:""

<?xml version='1.0'encoding='UTF-8'?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getQuote xmlns:ns1="urn:xmltoday-delayed-quotes"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <symbol xsi:type="xsd:string">IBM</symbol>
    </ns1:getQuote>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
HTTP/1.1 200 OK
Server:IBM_HTTP_SERVER/1.3.19 Apache/1.3.20 (Win32)
Content-Length:479
Connection:close
Content-Type:text/xml;charset=utf-8
Content-Language:en

<?xml version='1.0'encoding='UTF-8'?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getQuoteResponse xmlns:ns1="urn:xmltoday-delayed-quotes"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="xsd:float">108.53</return>
    </ns1:getQuoteResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

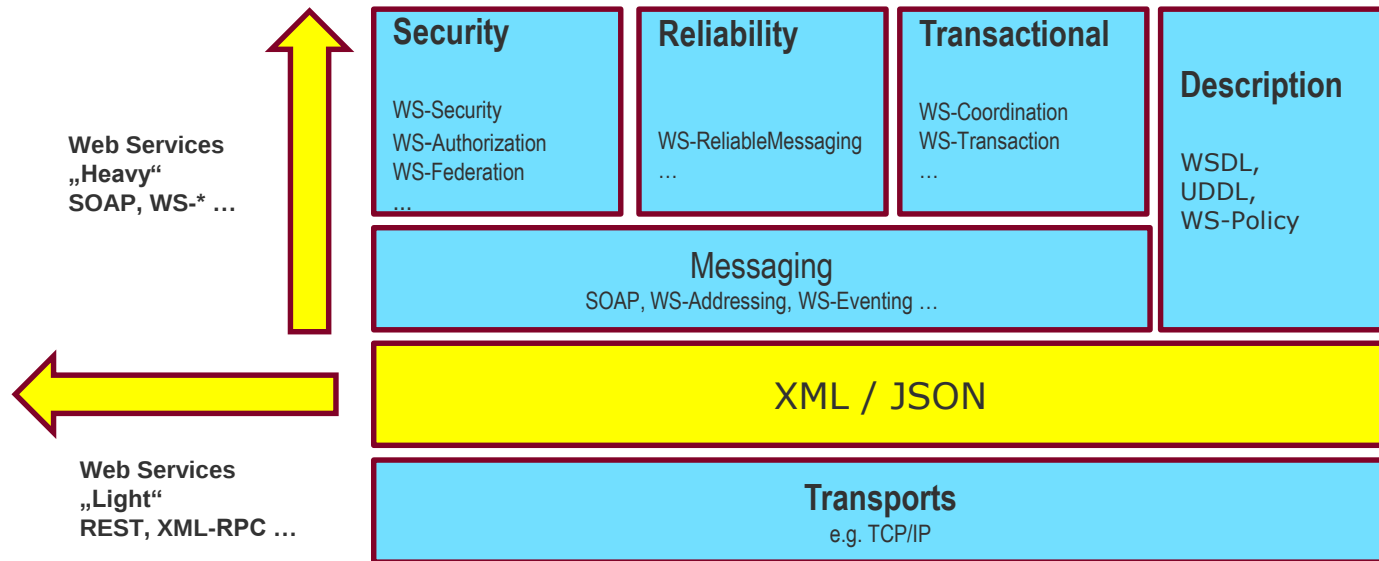
- „Das muss doch einfacher gehen!“
- Dissertation von Roy Fielding
 - REpresentational State Transfer (REST)
 - Alles dreht sich um „Ressourcen“
 - Eine Ressource wird durch einen URI repräsentiert
 - Z.B. könnte ein EMA Teilnehmer repräsentiert werden durch:
<http://www.mainframe-academy.de/tn/EMA0801>
 - Standard Methoden: HTTP GET/PUT/DELETE/POST etc.
 - Diverse Formate, meist XML und/oder JSON

```
GET /modules HTTP/1.0  
ACCEPT: application/json
```

```
HTTP/1.0 200 OK
Content-Type: application/json
Content-Length: 298
[ { uri:"ws.mainframe.academy.de",
  name: „Security“,
  start: 29.1.2015 }
{ uri:"ws.mainframe.academy.de",
  name: „Performance Management“,
  start: 8.2.2015 } ]
```

Web Services heavy vs. light

- Web Services „Light“: REST, XML-RPS, ...
- Web Services „Heavy“: WS-* standards



Sofern es keinen zwingenden Grund gibt, SOAP zu nutzen, sollte REST verwendet werden!

■ Wann REST verwenden?

- REST nutzt Standard-HTTP und ist sehr einfach.
- REST unterstützt unterschiedliche Formate, nicht nur XML.
- JSON beispielsweise parsed schneller und eignet sich besser für Browser-Clients, da diese JSON von Haus aus unterstützen.
- REST skaliert besser.

■ Wann SOAP verwenden?

- SOAP unterstützt WS-Security.
- SOAP unterstützt WS-AtomicTransaction (wenn ACID Merkmale gefordert werden).
- SOAP unterstützt WS-ReliableMessaging

- *"Extend modern application lifecycle management to IBM System z through the IBM Rational® Software Delivery Platform (SDP)"*
- Beispiele für angebotene Tutorials
 - Create COBOL programs and Web Services without writing Code
 - Create a Web Page to invoke a COBOL CICS Web Service
 - Create a Web Service form a CICS Application
 - Create a Web Front End for your Web Service
 - Build a Web Page from a Web Service



Fragen / Diskussionen
gerne im Forum zu diesem MOOC