



WebServices

Zwischen Buzzword und Nutzen

Tobias Koenig



Übersicht

- Webservices Allgemein
- WSDL
- Anwendungsbeispiele
- Programmierung
 - Perl
 - Python
 - C++/KDE
- Zusammenfassung

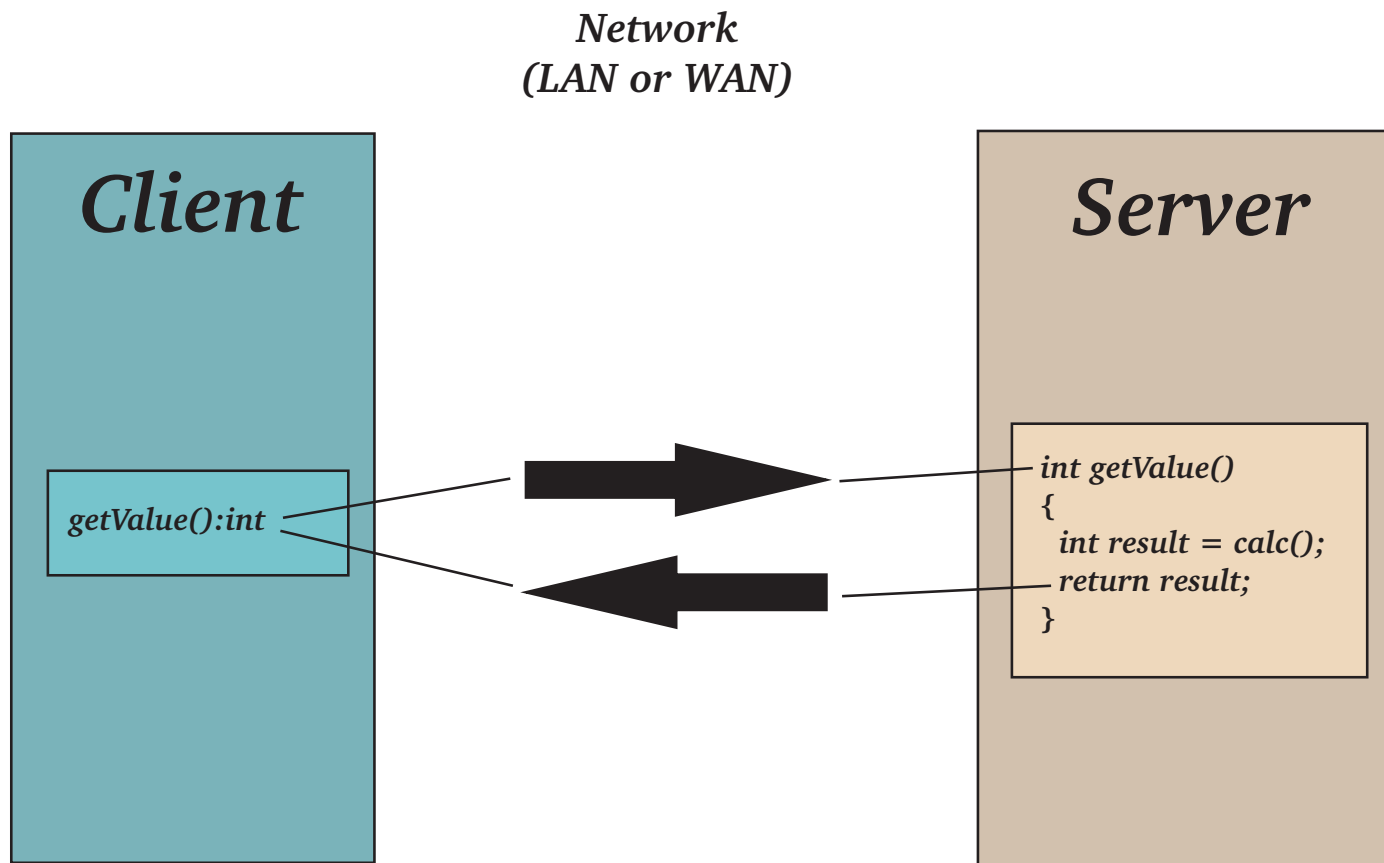
Webservices



- Trennung von Anwendung und Algorithmus
- Client-Server Architektur
- XML basierende Kommunikation



Webservices Aufbau



WSDL



- **Web Service Description Language**
- Webservice Beschreibung in XML
- Erhältlich beim Anbieter des Webservices



WSDL Grundstruktur



- Types
 - Typenbeschreibung der Argumente/Rückgabewerte in XML Schema
- Messages
 - Definition der angebotenen Eingabe- und Ausgabemethoden
- PortTypes
 - Bindung der Eingabe- an die Ausgabemethoden



WSDL Grundstruktur



- Bindings
 - Definition des verwendeten Übertragungsprotokoll (SOAP, HTTP, Mime)
- Service
 - Definition des Einstiegspunktes (URL) des Webservices



WSDL Beispiel



```
<?xml version="1.0"?>
<definitions name="TestService"
  xmlns:typens="urn:TestService"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  ...
</definitions>
```



WSDL Beispiel: Types



```
<types>
  <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:TestService">

    <xsd:complexType name="Temperature">
      <xsd:all>
        <xsd:element name="degree" type="xsd:int"/>
        <xsd:element name="fahrenheit" type="xsd:int"/>
      </xsd:all>
    </xsd:complexType>

  </xsd:schema>
</types>
```



WSDL Beispiel: Messages



```
<message name="getTemperature">  
  <part name="station" type="xsd:string"/>  
</message>
```

```
<message name="getTemperatureResponse">  
  <part name="return" type="typens:Temperature"/>  
</message>
```



WSDL Beispiel: PortTypes



```
<portType name="TestServicePort">  
  <operation name="getTemperature">  
    <input message="typens:getTemperature"/>  
    <output message="typens:getTemperatureResponse"/>  
  </operation>  
</portType>
```



WSDL Beispiel: Bindings



```
<binding name="TestServiceBinding" type="typens:TestServicePort">
  <soap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getTemperature">
    <soap:operation soapAction="urn:TestServiceAction"/>
    <input>
      <soap:body use="encoded" namespace="urn:TestService"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:TestService"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>
```



WSDL Beispiel: Services



```
<service name="TestServiceService">  
  <port name="TestServicePort" binding="typens:TestServiceBinding">  
    <soap:address location="http://ws.myserver.com/apps/testservice.php">  
  </port>  
</service>
```



WSDL Beispiel: Anfrage



```
<?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/1999/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getTemperature xmlns:ns1="urn:TestService"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
      <location xsi:type="xsd:string">Dresden</location>
    </ns1:getTemperature>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



WSDL Beispiel: Antwort



```
<?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/1999/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:getTemperatureResponse xmlns:ns1="urn:TestService"
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
      <return xsi:type="ns1:Temperature">
        <degree xsi:type="xsd:int">0</degree>
        <fahrenheit xsi:type="xsd:int">32</fahrenheit>
      </return>
    </ns1:getTemperatureResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



Anwendungsbeispiele

- Verbindung von Komponenten in großen Systemen (SAP)
- E-Business im Internet (Pizzaservice)
- Integration von Drittanbietersoftware (RuDI)

Nutzung via Perl



```
use SOAP::Lite;
```

```
$ws = SOAP::Lite->service( "http://url/to/webservice.wsdl" );
```

```
$temperature = $ws->getTemperature( "Dresden" );
```



Nutzung via Python



```
import PySOAP, Webservice
```

```
ws = Webservice.ServiceProxy( "http://url/to/webservice.wsdl" )
```

```
temperature = ws.getTemperature( "Dresden" )
```



Nutzung via C++/KDE

- C++ keine dynamische Sprache
-> Codegenerierung notwendig
- gSOAP für reinen C Code
- kwsdl für Qt/KDE C++ Code

Nutzung via C++/KDE

- C++ keine dynamische Sprache
-> Codegenerierung notwendig
- gSOAP für reinen C Code
- kwsdl für Qt/KDE C++ Code

Webservice anbieten

- Server muss Kommunikation unterstützen
 - HTTP
 - XML Parsing/Synthese
- Apache + PHP4
- Python-only Lösung

Zusammenfassung



- WebServices ermöglichen dezentrale Programme
- WSDL beschreibt WebService vollständig
- Skriptsprachen bieten exzellente Unterstützung
- Stärkere Verbreitung in Zukunft





Noch Fragen?





Vielen Dank

