Werner weiß…

Tim weiß…

Lotus weiß…

… Notes kann alles*!

*Außer Drucken. Angaben ohne Gewähr.

(auch Kaffee kochen)

Werner Motzet und Tim Pistor präsentieren:

Webservices mit Lotus Notes

Werner Motzet und Tim Pistor präsentieren:

Webservices mit Lotus Notes

SSL

WSDL

SOAP

PHP

Kaffee

HTTP

XML

lsx

LotusScript

Provider

RPC

IPhone

Consumer

Java

Ihre Referenten

Werner Motzet

- Idee zum Vortrag vom AdminCamp
- Bei atnotes „WernerMo“ („Wimpelnäher“)
- ProjektManager seit 1989/90
- Seit 2000 bei [isgroup] -> seit Jan.08 is@web
  http://www.isatweb.com
- Ursprünglich Dipl.Theol.
- 1.Sprache bayrisch <-> 2. hochdeutsch

Tim Pistor

- Seit 2008 freier Entwickler
- Dynamische (Web-) Anwendungen unter Lotus Notes / Domino
- Co-Autor von !!HELP!!

DEMO

Dynamische Webapps

DEMO

Cooler Webservice

Unsere Ziele

Unsere Ziele in dieser Session

- Wir durchleuchten Hintergründe und Möglichkeiten von Webservices
- Sie erlernen die ersten Schritte zum Erstellen von Webservices und…
  - … werden in der Lage sein einen Consumer zu erstellen
  - … werden in der Lage sein einen Provider zu erstellen

Inhalt

Inhalte

- Webservice Überblick
- Protokolle der Kommunikation
- Datenstrukturen
- Wer nutzt WS und wofür?
- Wie können wir WS aus Notes ansprechen?
- Wie können wir WS zur Verfügung stellen?
- Gesicherte Verbindungen
- Demos, Demos und noch mehr Demos

Demo zum Beispiel :

Nachbearbeitung und erste Schritte

Diese Präsentation und die Beispieldatenbanken finden Sie nach dem Camp auf

www.entwicklercamp.de

Was sind Webservices?

Was sind Webservices?

- Sie ermöglichen die Zusammenarbeit von Anwendungen auf unterschiedlichen Plattformen

Was sind Webservices?

- Sie ermöglichen die Zusammenarbeit von Anwendungen auf unterschiedlichen Plattformen

- Sie nutzen XML und Internetbasierte Protokolle (HTTP)

Was sind Webservices?

- Sie ermöglichen die Zusammenarbeit von Anwendungen auf unterschiedlichen Plattformen
- Sie nutzen XML und internetbasierte Protokolle (HTTP)

- Sie halten sich an objektorientierte Programmierstandards

Was sind Webservices?

- Sie ermöglichen die Zusammenarbeit von Anwendungen auf unterschiedlichen Plattformen
- Sie nutzen XML und internetbasierte Protokolle (HTTP)
- Sie halten sich an objektorientierte Programmierstandards

- Ihre Datenstrukturen und Methoden werden durch das WSDL Format beschrieben (XML)

Was sind Webservices?

- Sie ermöglichen die Zusammenarbeit von Anwendungen auf unterschiedlichen Plattformen
- Sie nutzen XML und internetbasierte Protokolle (HTTP)
- Sie halten sich an objektorientierte Programmierstandards
- Ihre Datenstrukturen und Methoden werden durch die WSDL beschrieben.

- Sie sind keine Webseiten

Was sind Webservices?

„Was für den Mensch die Webseite ist für den Computer der Webservice“

DEMO

Wetter

Demo – Einfacher Webservice

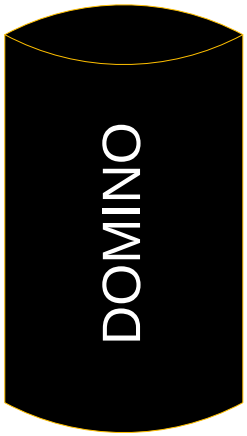
- Kontaktaufnahme zu einem Provider
- Einfacher Funktionsaufruf
- Einfacher Funktionsaufruf mit Parameter
- Kommunikation unter die Lupe genommen

DEMO

Einfacher Webservice

Demo – 1

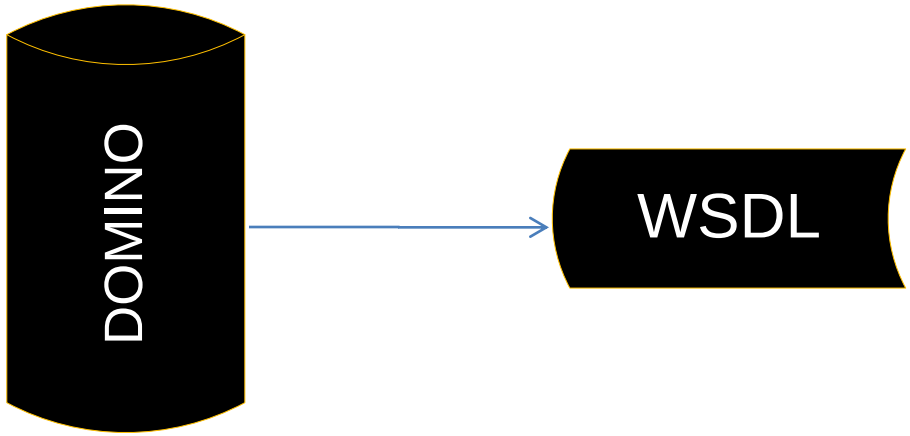
Provider



-Einfacher Funktionsaufruf
-Einfacher Funktionsaufruf mit Parameter

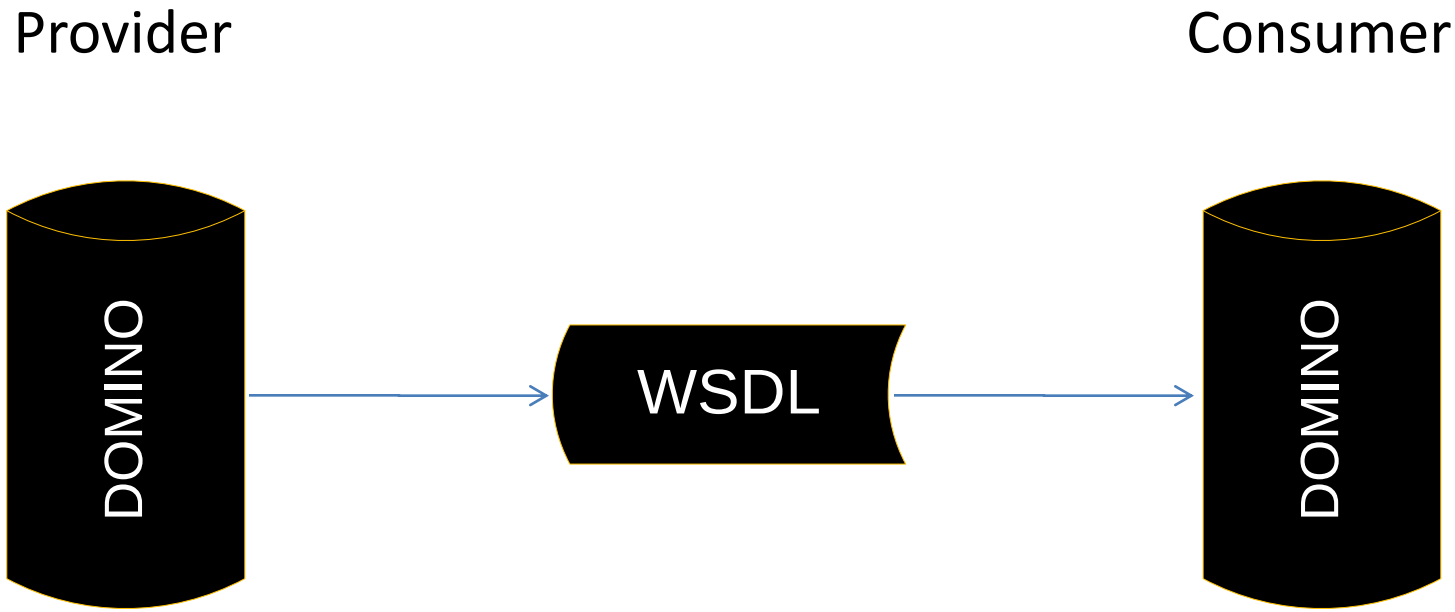
Demo – 1

Provider



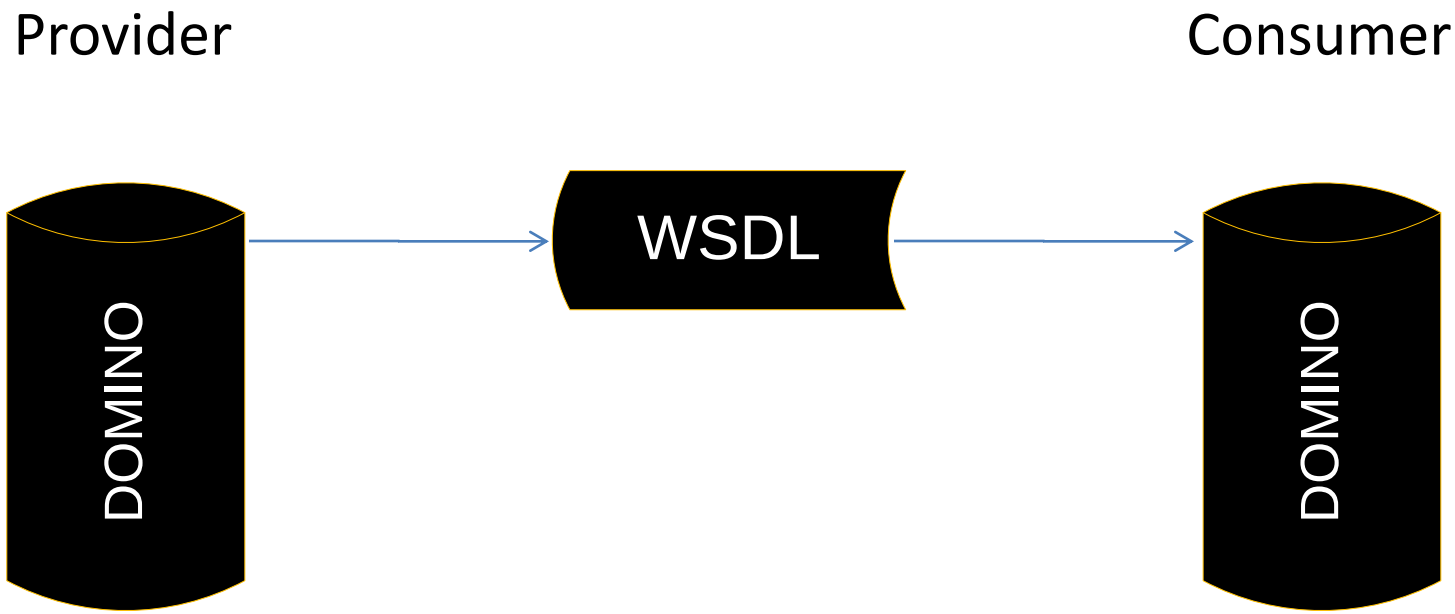
-Einfacher Funktionsaufruf
-Einfacher Funktionsaufruf mit Parameter

Demo – 1



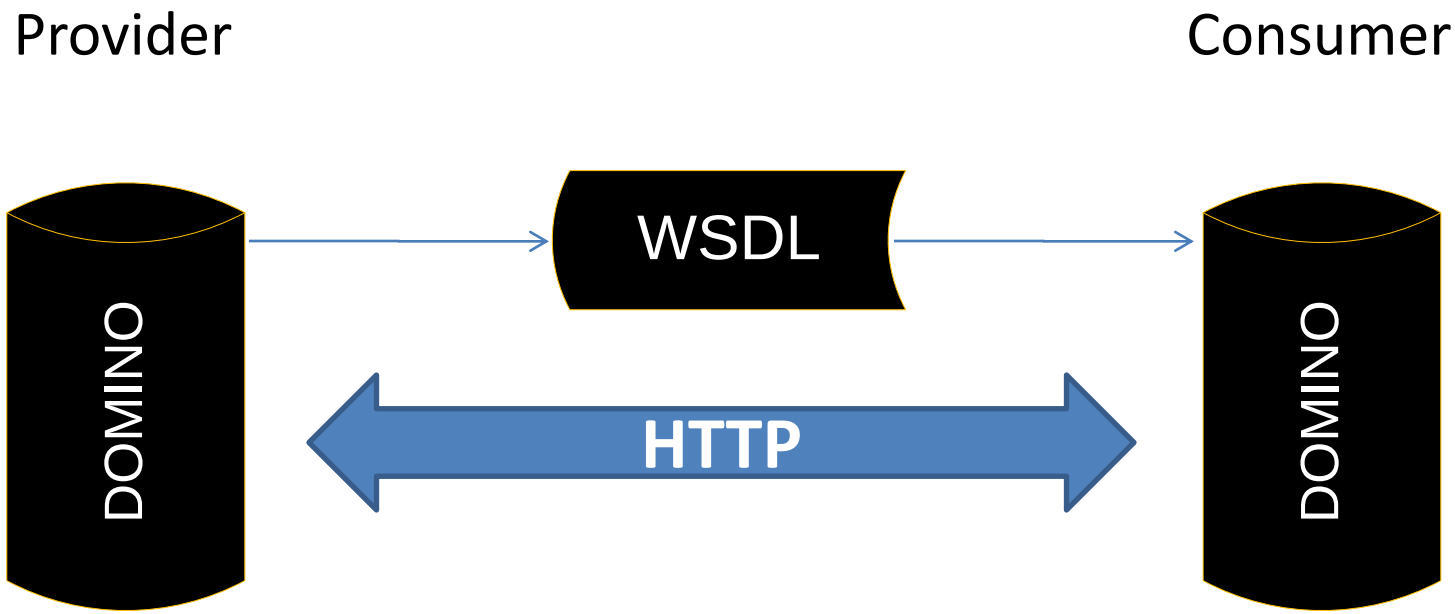
-Einfacher Funktionsaufruf
-Einfacher Funktionsaufruf mit Parameter

Demo – 1



-Einfacher Funktionsaufruf
-Einfacher Funktionsaufruf mit Parameter

Demo – 1



-Einfacher Funktionsaufruf
-Einfacher Funktionsaufruf mit Parameter

WSDL – Die Schnittstellenbeschreibung

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace="urn:DefaultNamespace" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="urn:DefaultNamespace"
        xmlns:intf="urn:DefaultNamespace" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
        xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <wsdl:types>
  <schema targetNamespace="urn:DefaultNamespace" xmlns="http://www.w3.org/2001/XMLSchema">
   <element name="EINEFUNCTIONReturn" type="xsd:boolean"/>
  </schema>
 </wsdl:types>
 <message name="EINEFUNCTIONResponse">
  <part element="impl:EINEFUNCTIONReturn" name="EINEFUNCTIONReturn"/>
 </message>
 <message name="EINEFUNCTIONRequest">
 </message>
 <portType name="wsdlview">
  <operation name="EINEFUNCTION">
   <input message="impl:EINEFUNCTIONRequest" name="EINEFUNCTIONRequest"/>
   <output message="impl:EINEFUNCTIONResponse" name="EINEFUNCTIONResponse"/>
  </operation>
 </portType>
 <binding name="DominoSoapBinding" type="impl:wsdlview">
  <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="EINEFUNCTION">
   <wsdlsoap:operation soapAction="EINEFUNCTION"/>
   <input name="EINEFUNCTIONRequest">
    <wsdlsoap:body use="literal"/>
   </input>
   <output name="EINEFUNCTIONResponse">
    <wsdlsoap:body use="literal"/>
   </output>
  </operation>
 </binding>
 <service name="wsdlviewService">
  <port binding="impl:DominoSoapBinding" name="Domino">
   <wsdlsoap:address location="http://localhost"/>
  </port>
```

WSDL – Die Schnittstellenbeschreibung

- Web Service Description Language (XML)

- Enthält alle nötigen Informationen über

  - Austauschprotokolle
  - Funktionen
  - Daten
  - Datentypen

WSDL – Die Schnittstellenbeschreibung

- XML-Hauptelemente
  - Datentypen (types)
  - Nachrichten (messages)
  - Schnittstellentypen (portType)
  - Bindung (binding)
  - Ports (port)
  - Services (service)

WSDL – Die Schnittstellenbeschreibung

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions targetNamespace="urn:DefaultNamespace" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="urn:DefaultNamespace"
        xmlns:intf="urn:DefaultNamespace" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
        xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <wsdl:types>
  <schema targetNamespace="urn:DefaultNamespace" xmlns="http://www.w3.org/2001/XMLSchema">
   <element name="EINEFUNCTIONReturn" type="xsd:boolean"/>
  </schema>
 </wsdl:types>
 <message name="EINEFUNCTIONResponse">
  <part element="impl:EINEFUNCTIONReturn" name="EINEFUNCTIONReturn"/>
 </message>
 <message name="EINEFUNCTIONRequest">
 </message>
 <portType name="wsdlview">
  <operation name="EINEFUNCTION">
   <input message="impl:EINEFUNCTIONRequest" name="EINEFUNCTIONRequest"/>
   <output message="impl:EINEFUNCTIONResponse" name="EINEFUNCTIONResponse"/>
  </operation>
 </portType>
 <binding name="DominoSoapBinding" type="impl:wsdlview">
  <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="EINEFUNCTION">
   <wsdlsoap:operation soapAction="EINEFUNCTION"/>
   <input name="EINEFUNCTIONRequest">
    <wsdlsoap:body use="literal"/>
   </input>
   <output name="EINEFUNCTIONResponse">
    <wsdlsoap:body use="literal"/>
   </output>
  </operation>
 </binding>
 <service name="wsdlviewService">
  <port binding="impl:DominoSoapBinding" name="Domino">
   <wsdlsoap:address location="http://localhost"/>
  </port>
```

WSDL – Die Schnittstellenbeschreibung

- **Datentypen (types)**
- Nachrichten (messages)
- Schnittstellentypen (portType)
- Bindung (binding)
- Ports (port)
- Services (Service)

Definition der Datentypen, die zum Austausch der messages genutzt werden

```
<wsdl:types>
 <schema targetNamespace="urn:DefaultNamespace"
         xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="HELLOWORLDReturn" type="xsd:string"/>
  <element name="HELLOWORLDNOWReturn" type="xsd:string"/>
  <element name="PAR" type="xsd:string"/>
  <element name="HELLOWORLDPINGPONGReturn" type="xsd:string"/>
 </schema>
</wsdl:types>
```

WSDL – Die Schnittstellenbeschreibung

- Datentypen (types)
- **Nachrichten (messages)**
- Schnittstellentypen (portType)
- Bindung (binding)
- Ports (port)
- Services (Service)

Abstrakte Definitionen der übertragenen Daten, bestehend aus mehreren logischen Teilen, von denen jedes mit einer Definition innerhalb eines Datentyps verknüpft ist.

```
<message name="HELLOWORLDRequest">
 </message>

<message name="HELLOWORLDResponse">
  <part element="impl:HELLOWORLDReturn" name="HELLOWORLDReturn"/>
 </message>
```

WSDL – Die Schnittstellenbeschreibung

- Datentypen (types)
- Nachrichten (messages)

- **Schnittstellentypen (portType)**
- Bindung (binding)

- Ports (port)
- Services (Service)

Eine Menge von abstrakten Arbeitsschritten :

```
<portType name="exampleOne">
   <operation name="HELLOWORLD">

    <input message="impl:HELLOWORLDRequest„
           name="HELLOWORLDRequest"/>

    <output message="impl:HELLOWORLDResponse"
           name="HELLOWORLDResponse"/>

   </operation>
</portType>
```

WSDL – Die Schnittstellenbeschreibung

- Datentypen (types)
- Nachrichten (messages)
- Schnittstellentypen (portType)
- **Bindung (binding)**
- Ports (port)
- Services (Service)

Bestimmt das konkrete Protokoll und Datenformat für die Arbeitsschritte

```
<binding name="DominoSoapBinding" type="impl:exampleOne">
 <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
 <operation name="HELLOWORLD">
  <wsdlsoap:operation soapAction="HELLOWORLD"/>
  <input name="HELLOWORLDRequest">
   <wsdlsoap:body use="literal"/>
  </input>
  <output name="HELLOWORLDResponse">
   <wsdlsoap:body use="literal"/>
  </output>
 </operation>
</binding>
```

(gekürzt)

WSDL – Die Schnittstellenbeschreibung

- Datentypen (types)
- Nachrichten (messages)

- Schnittstellentypen (portType)
- Bindung (binding)

- **Ports (port)**
- Services (Service)

Spezifiziert eine Adresse für eine Bindung, also eine Kommunikationsschnittstelle, üblicherweise ein URI

```
<port binding="impl:DominoSoapBinding" name="Domino">
  <wsdlsoap:address location="http://localhost"/>
 </port>
```

WSDL – Die Schnittstellenbeschreibung

- Datentypen (types)
- Nachrichten (messages)

- Schnittstellentypen (portType)
- Bindung (binding)

- Ports (port)
- **Services (Service)**

Fassen eine Menge von verwandten Ports zusammen.

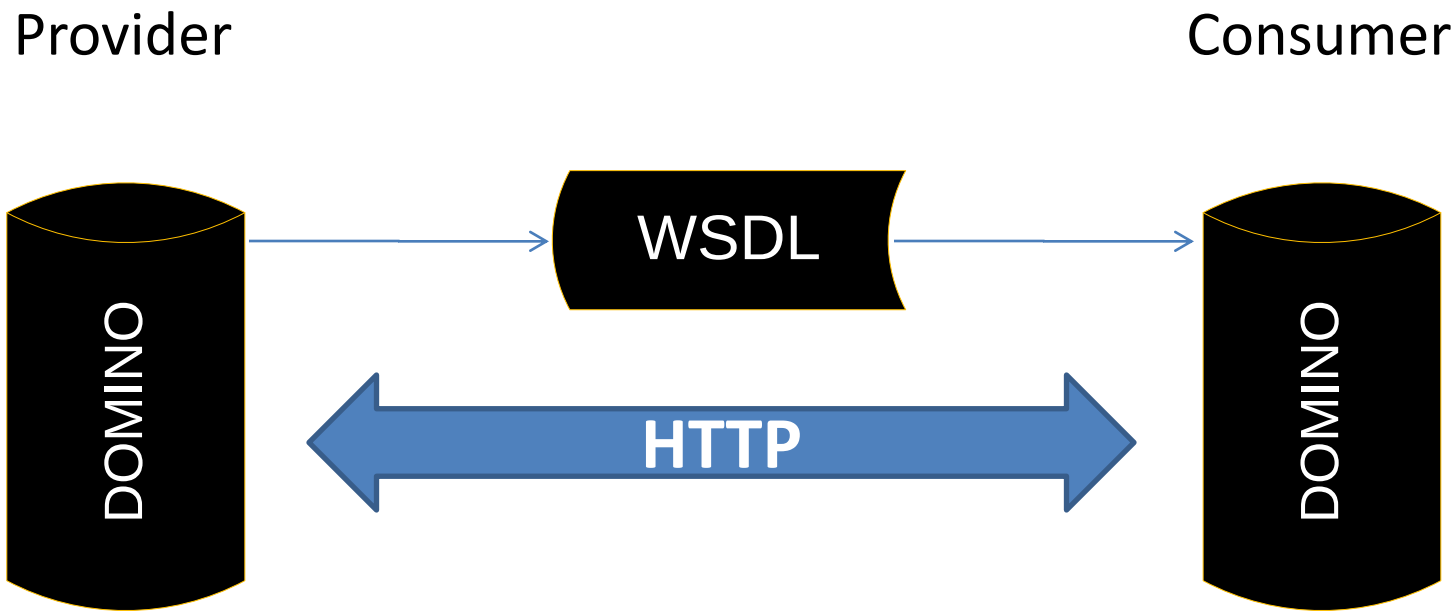
```
<service name="exampleOneService">
 <port binding="impl:DominoSoapBinding" name="Domino">
  <wsdlsoap:address location="http://localhost"/>
 </port>
</service>
```

Webservices mit Lotus Notes

Datentypen

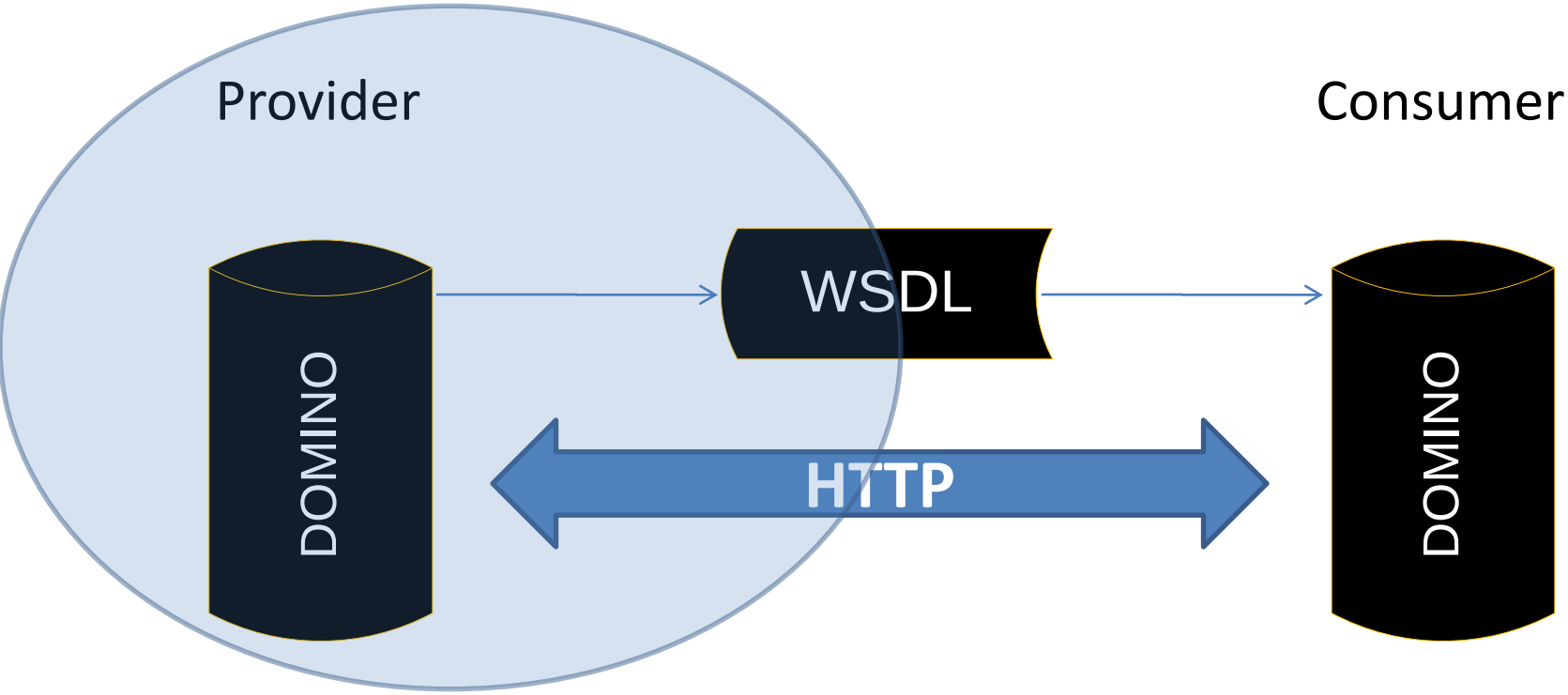
Erstellen eines Webservice Provider

Datentypen / Erstellen eines Providers



-Einfacher Funktionsaufruf
-Einfacher Funktionsaufruf mit Parameter

Datentypen / Erstellen eines Providers



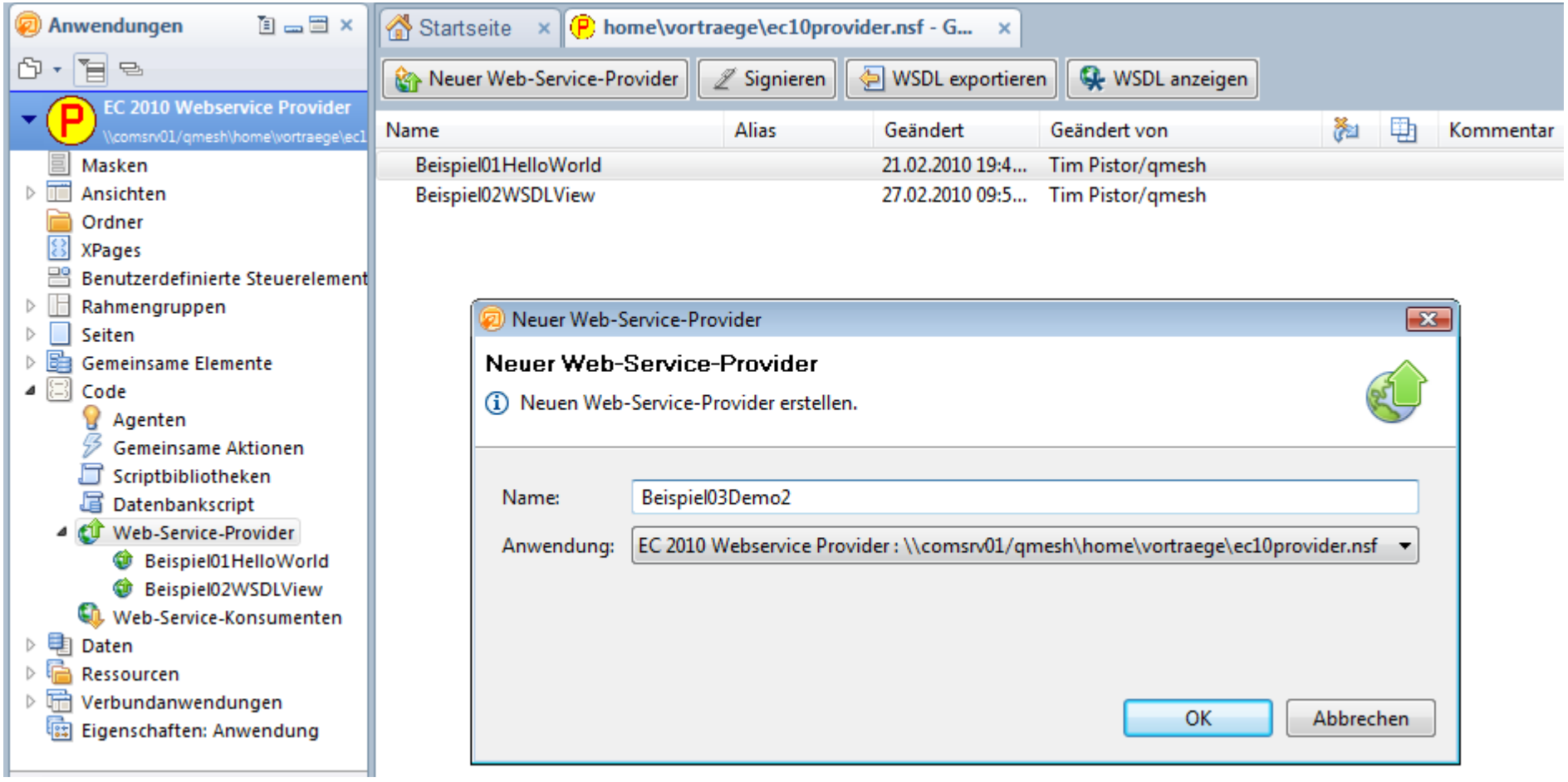
-Einfacher Funktionsaufruf
-Einfacher Funktionsaufruf mit Parameter

DEMO

Erstellung eines Providers

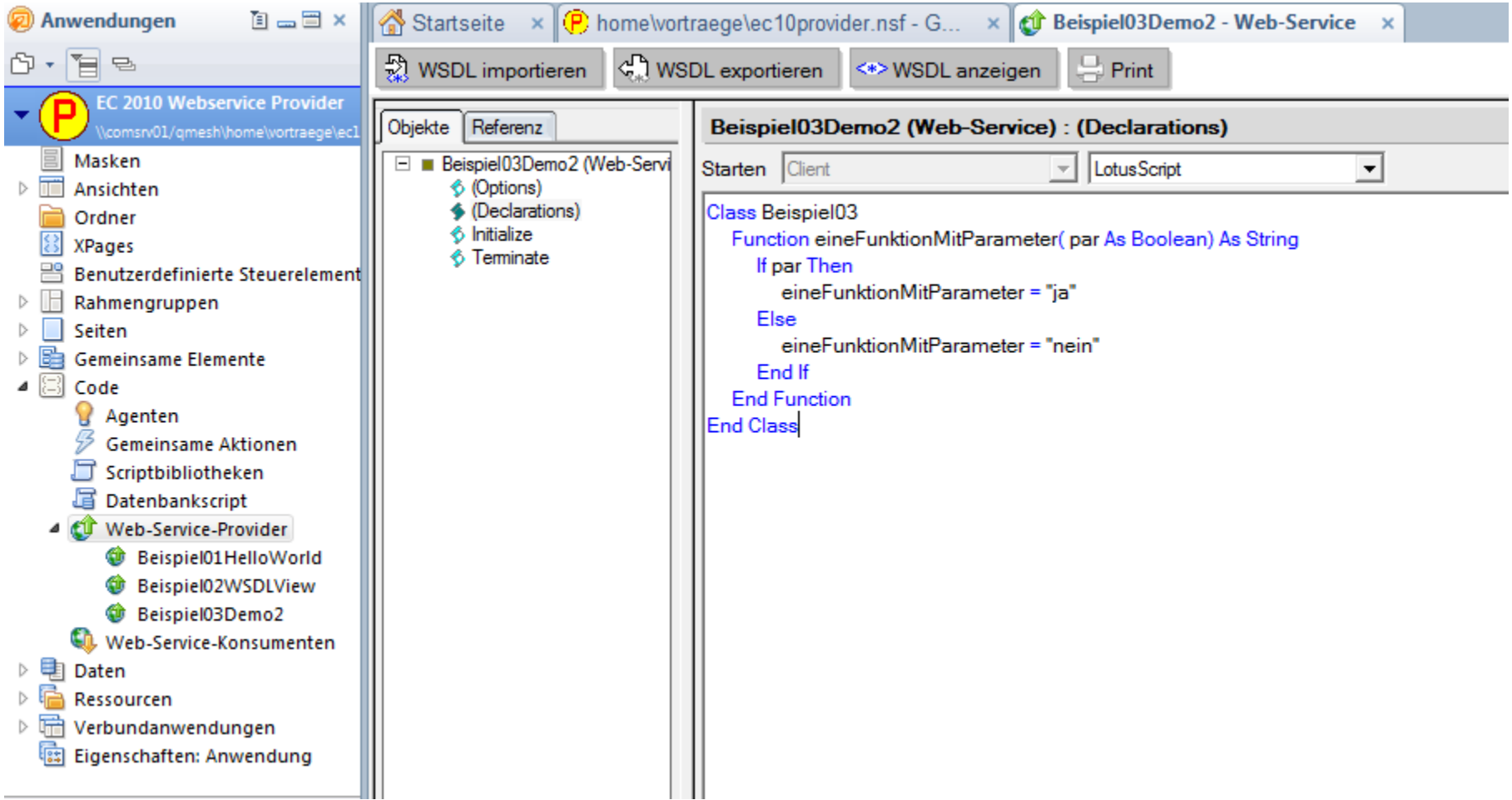
Datentypen / Erstellen eines Providers

1. Neuer Webservice Provider im Designer anlegen



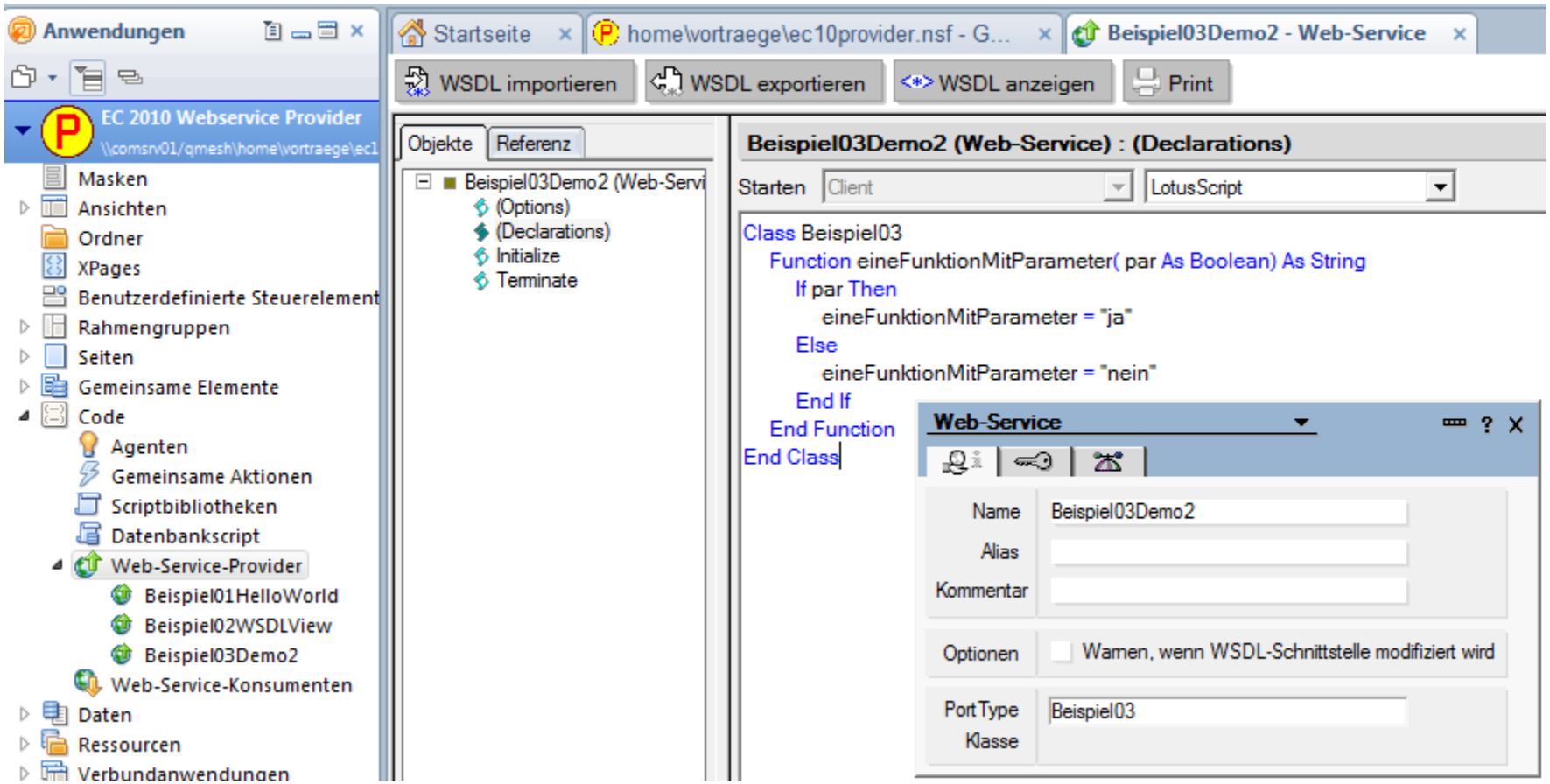
Datentypen / Erstellen eines Providers

2. Klasse „Beispiel03“ anlegen.



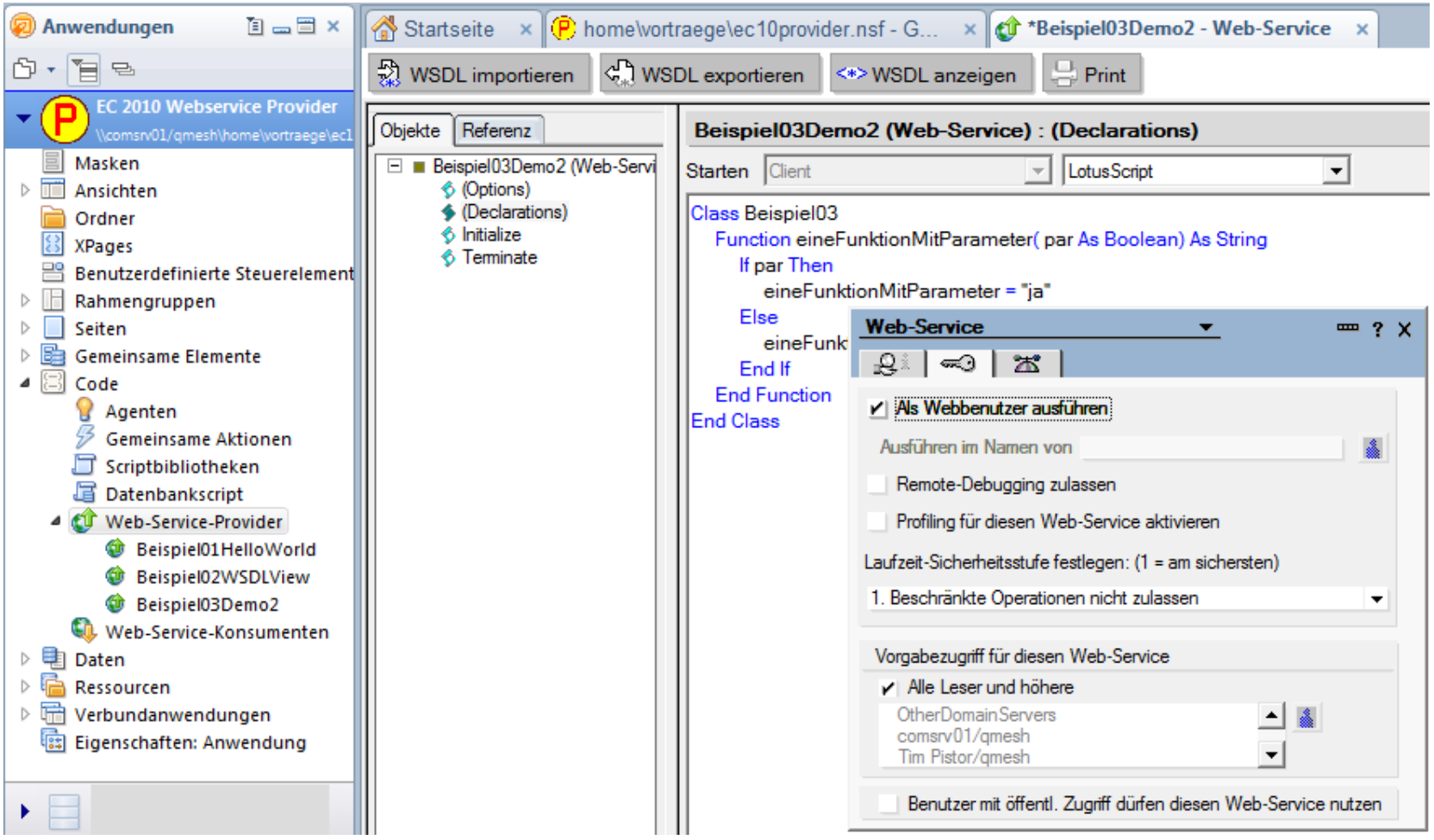
Datentypen / Erstellen eines Providers

3. Provider konfigurieren



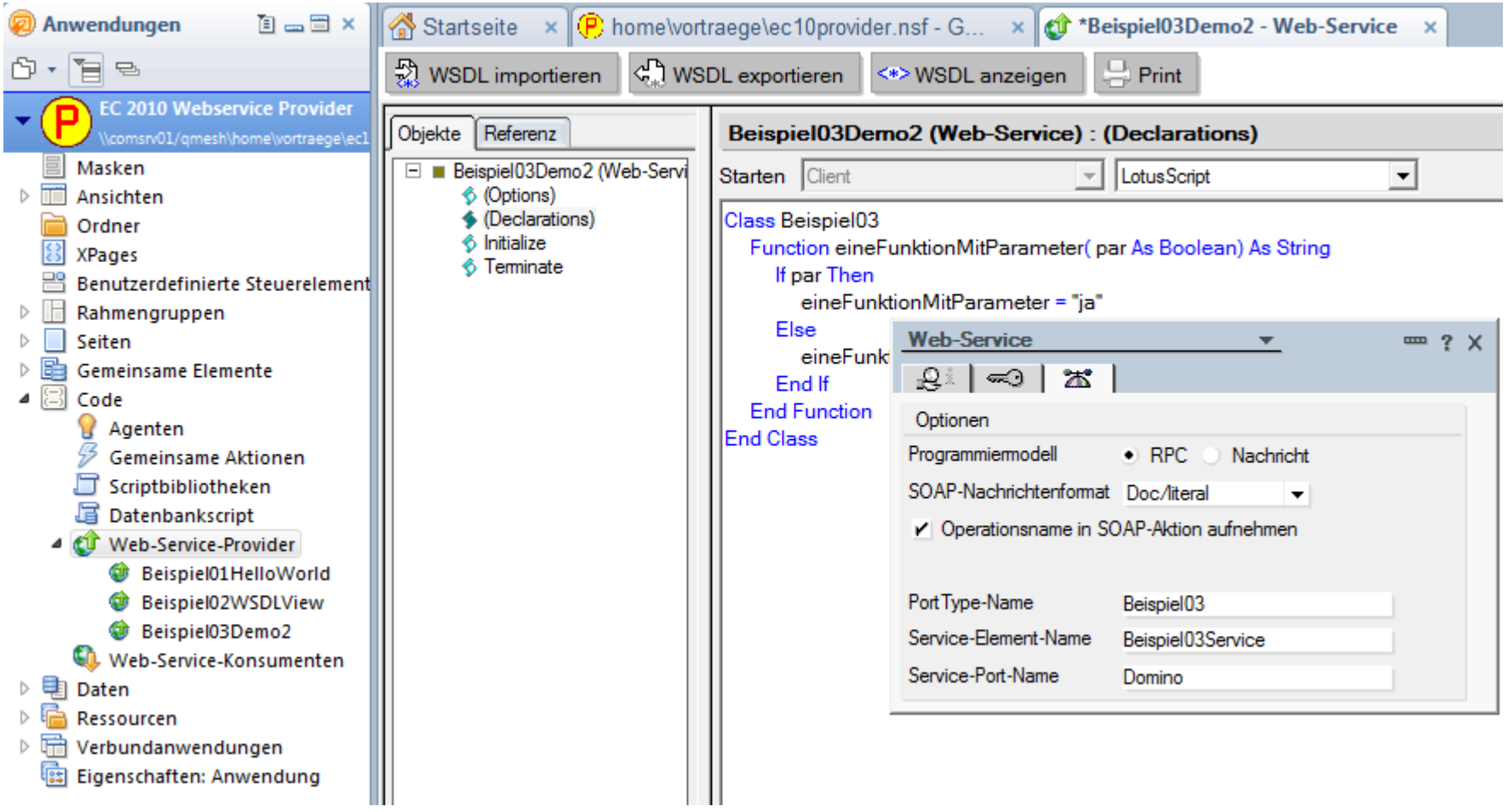
Datentypen / Erstellen eines Providers

3. Provider konfigurieren



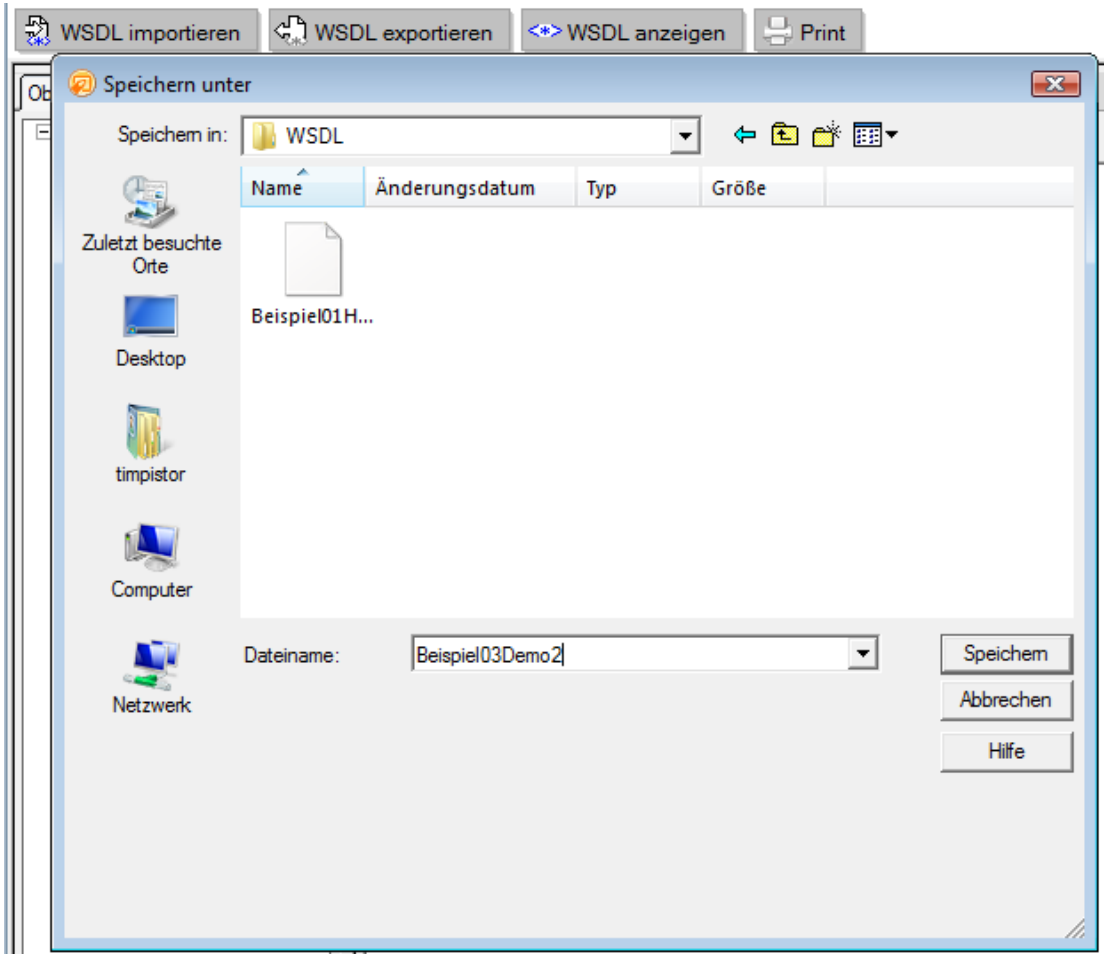
Datentypen / Erstellen eines Providers

3. Provider konfigurieren



Datentypen / Erstellen eines Providers

4. WSDL Exportieren



Datentypen / Erstellen eines Providers

- Es sind alle „Standard“ Datentypen erlaubt.
  - String
  - Integer
  - Double
  - Float
  - Boolean
  - Date

Datentypen / Erstellen eines Providers

- Es sind alle „Standard“ Datentypen erlaubt.
  - Public a1 As Boolean
    - <element name="A1" type="xsd:boolean"/>
  - Public a2 As Integer
    - <element name="A2" type="xsd:short"/>
  - Public a3 As Double
    - <element name="A3" type="xsd:double"/>
  - Public a4 As String
    - <element name="A4" type="xsd:string"/>

Datentypen / Erstellen eines Providers

- Es sind alle „Standard“ Datentypen erlaubt.
  - Public a5 As Long
    - <element name="A5" type="xsd:int"/>
  - Public a6 As Byte
    - <element name="A6" type="xsd:unsignedByte"/>
  - Public a7 As Single
    - <element name="A7" type="xsd:float"/>

Datentypen / Erstellen eines Providers

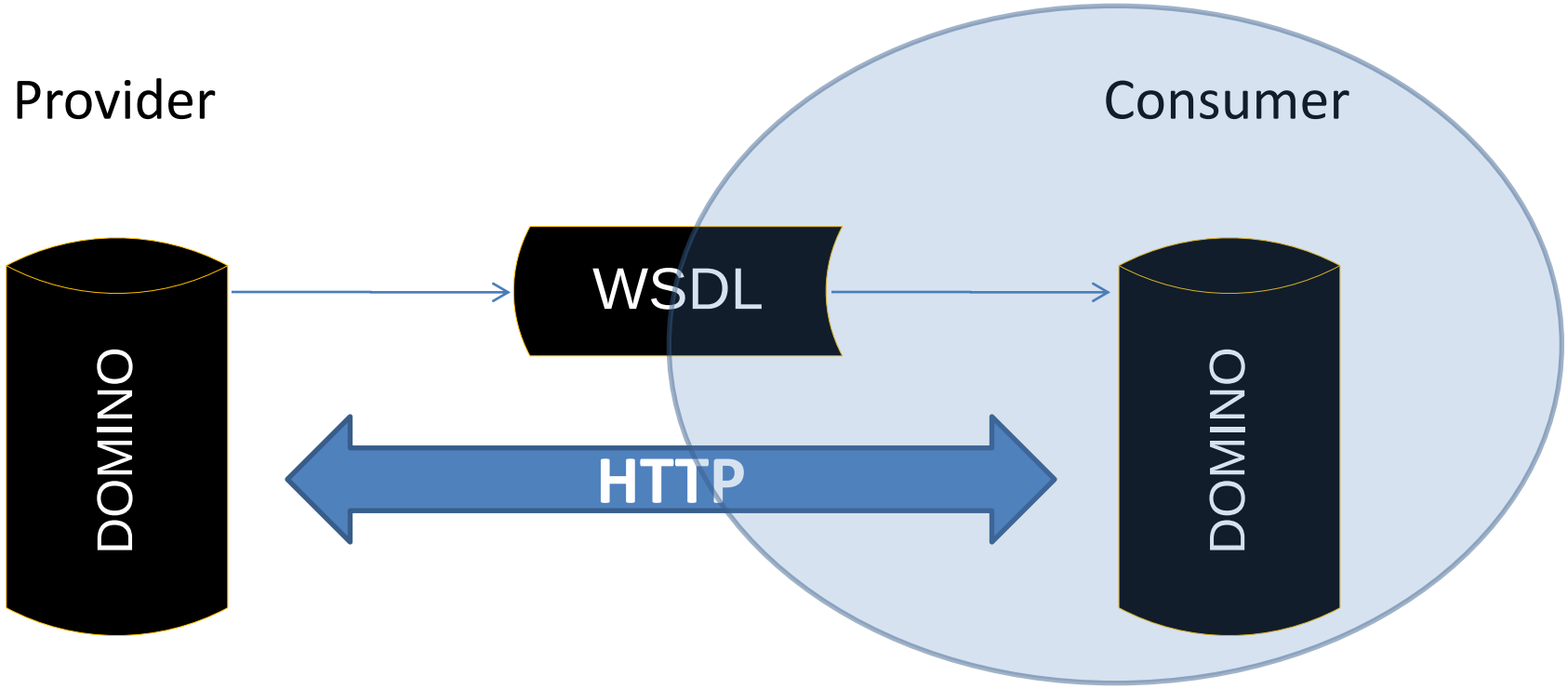
```
1.  Class Beispiel04
2.      Function getDatentypen() As Datentypen
3.              Set getDatentypen = New Datentypen
4.      End Function
5.  End Class

6.  Class Datentypen
7.      Public a1 As Boolean
8.      Public a2 As Integer
9.      Public a3 As Double
10.     Public a4 As String
11.     Public a5 As Long
12.     Public a6 As Byte
13.     Public a7 As Single
14. End Class
```

Konsumieren

Erstellen eines Webservice Consumers

Konsumieren - Erstellen eines Consumers



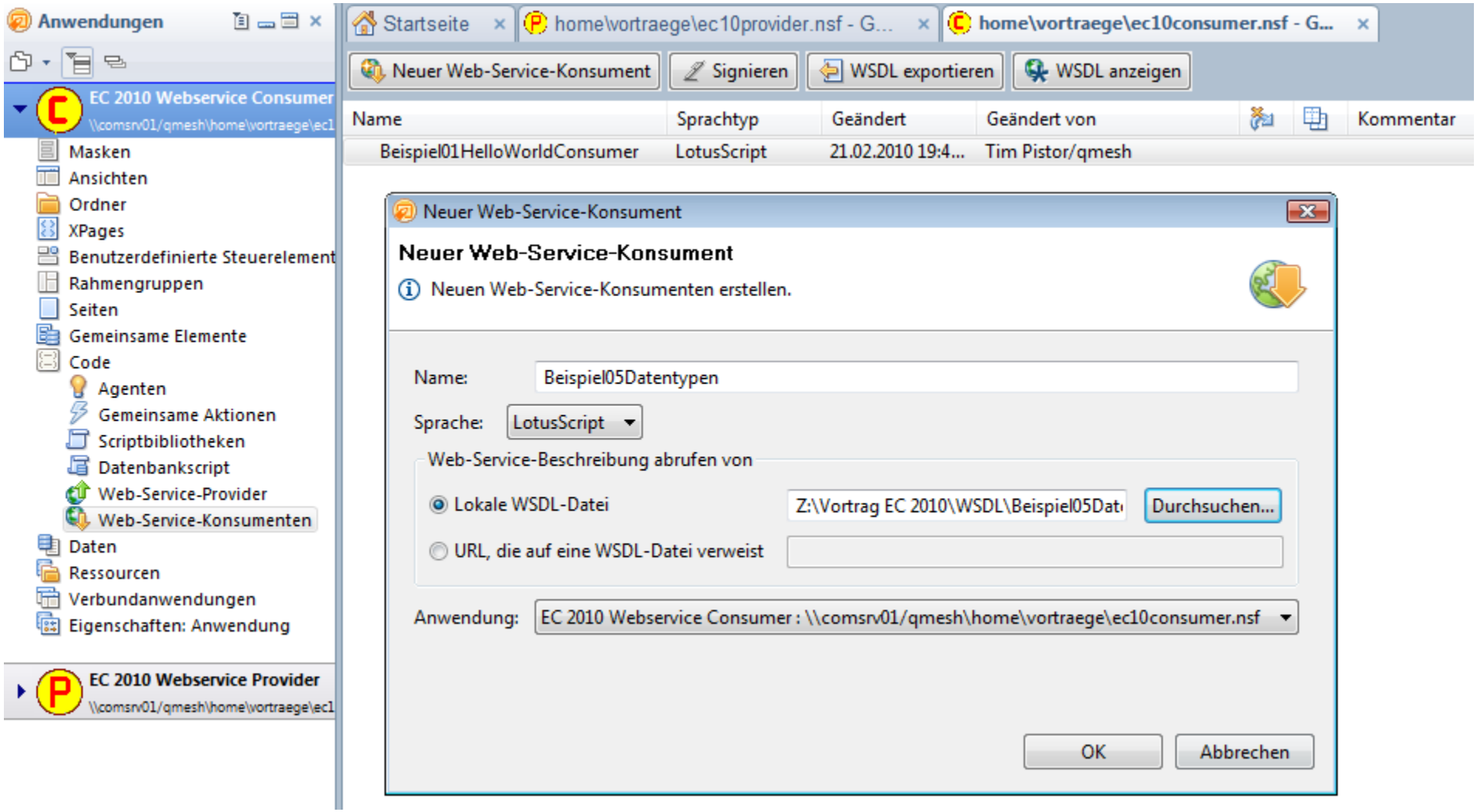
-Einfacher Funktionsaufruf mit Parameter
aus Beispiel05Datentypen

Konsumieren - Erstellen eines Consumers

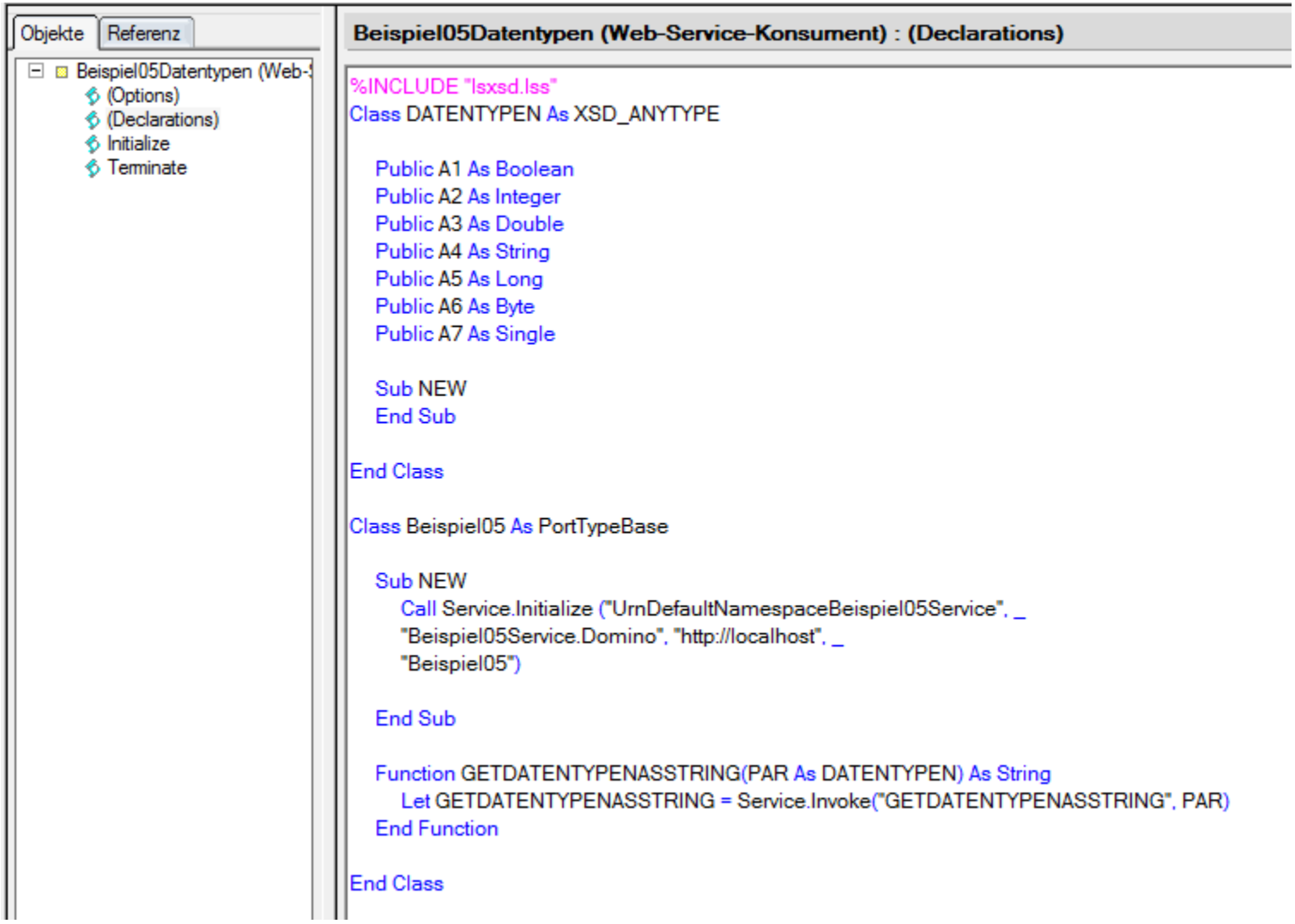
```
1.  Class Beispiel05
2.          Function getDatentypenAsString(par As Datentypen) As String
3.                  getDatentypenAsString = par.toString
4.          End Function
5.  End Class

6.  Class Datentypen
7.          Public a1 As Boolean        Public a2 As Integer
8.          Public a3 As Double         Public a4 As String
9.          Public a5 As Long           Public a6 As Byte
10.         Public a7 As Single
11.
12.         Public Function toString As String
13.             toString = a1 & " - " & a2 & " - " & a3& " - " &  a4 & " - " & a5& " - " &  a6& " - " & a7
14.         End Function
15. End Class
```

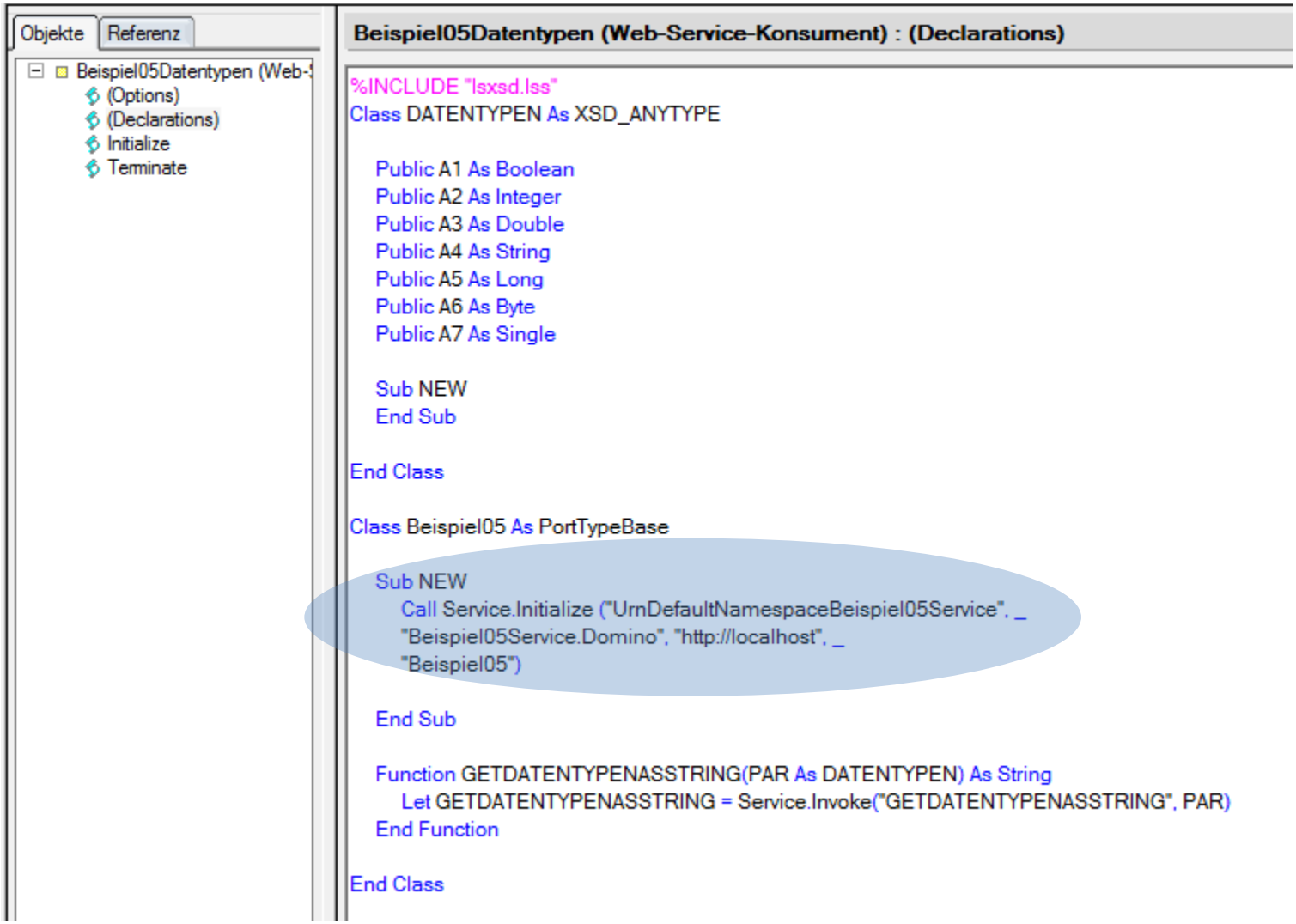
Konsumieren - Erstellen eines Consumers



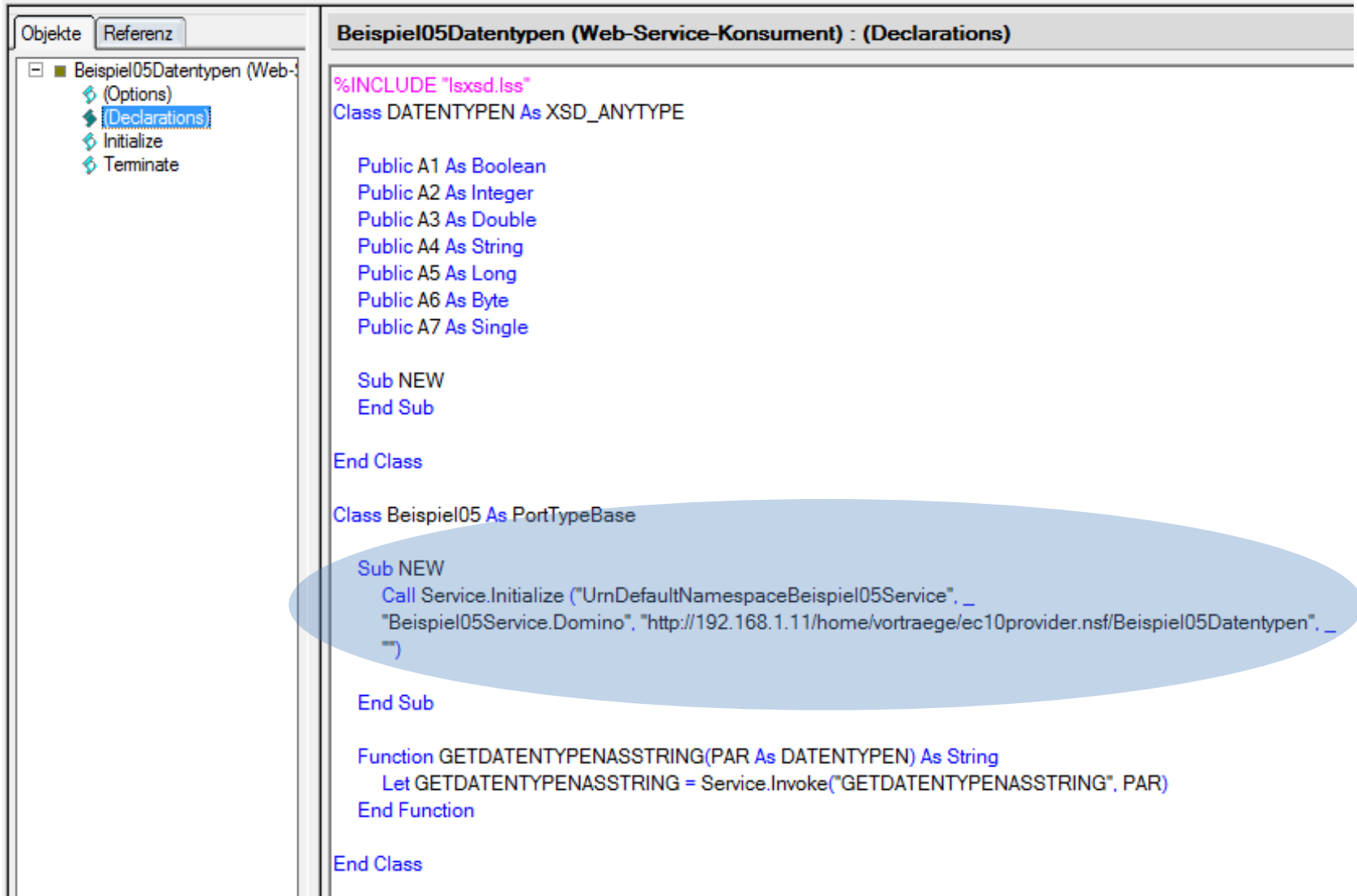
Konsumieren - Erstellen eines Consumers



Konsumieren - Erstellen eines Consumers



Konsumieren - Erstellen eines Consumers



Konsumieren - Erstellen eines Consumers

```
Sub NEW
    Call Service.Initialize ("UrnDefaultNamespaceBeispiel05Service", _
    "Beispiel05Service.Domino", "http://192.168.1.11/home/vortraege/ec10provider.nsf/Beispiel05Datentypen", _
    "")

End Sub
```

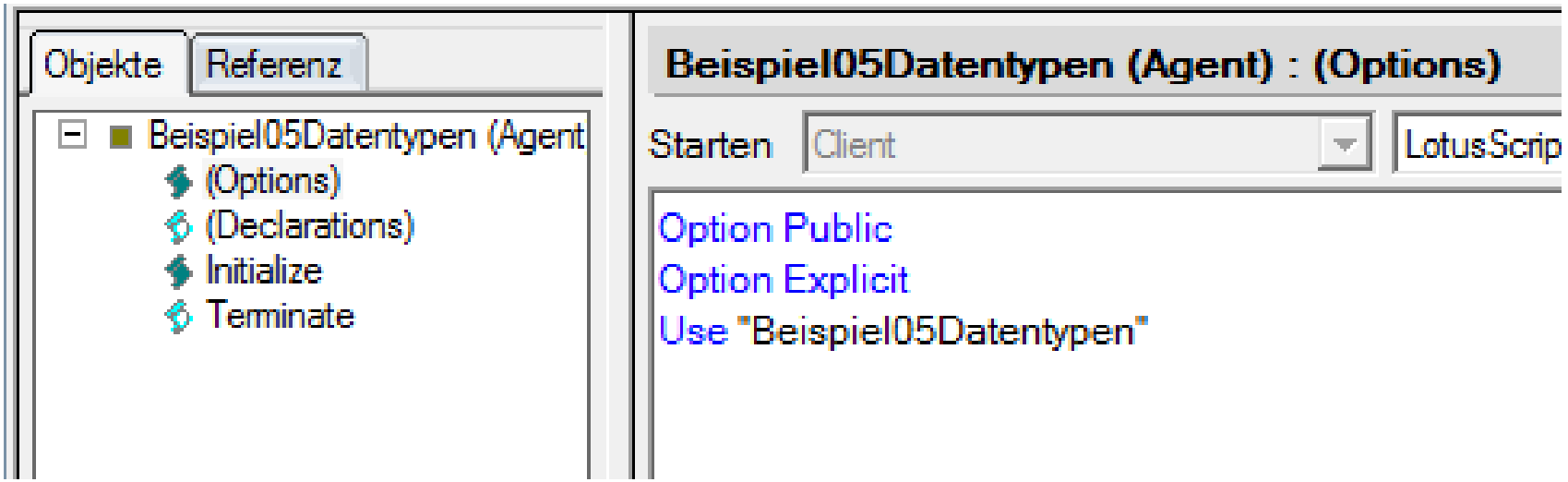
- URL zur Datenbank + „/“ Webservicename

Konsumieren - Erstellen eines Consumers

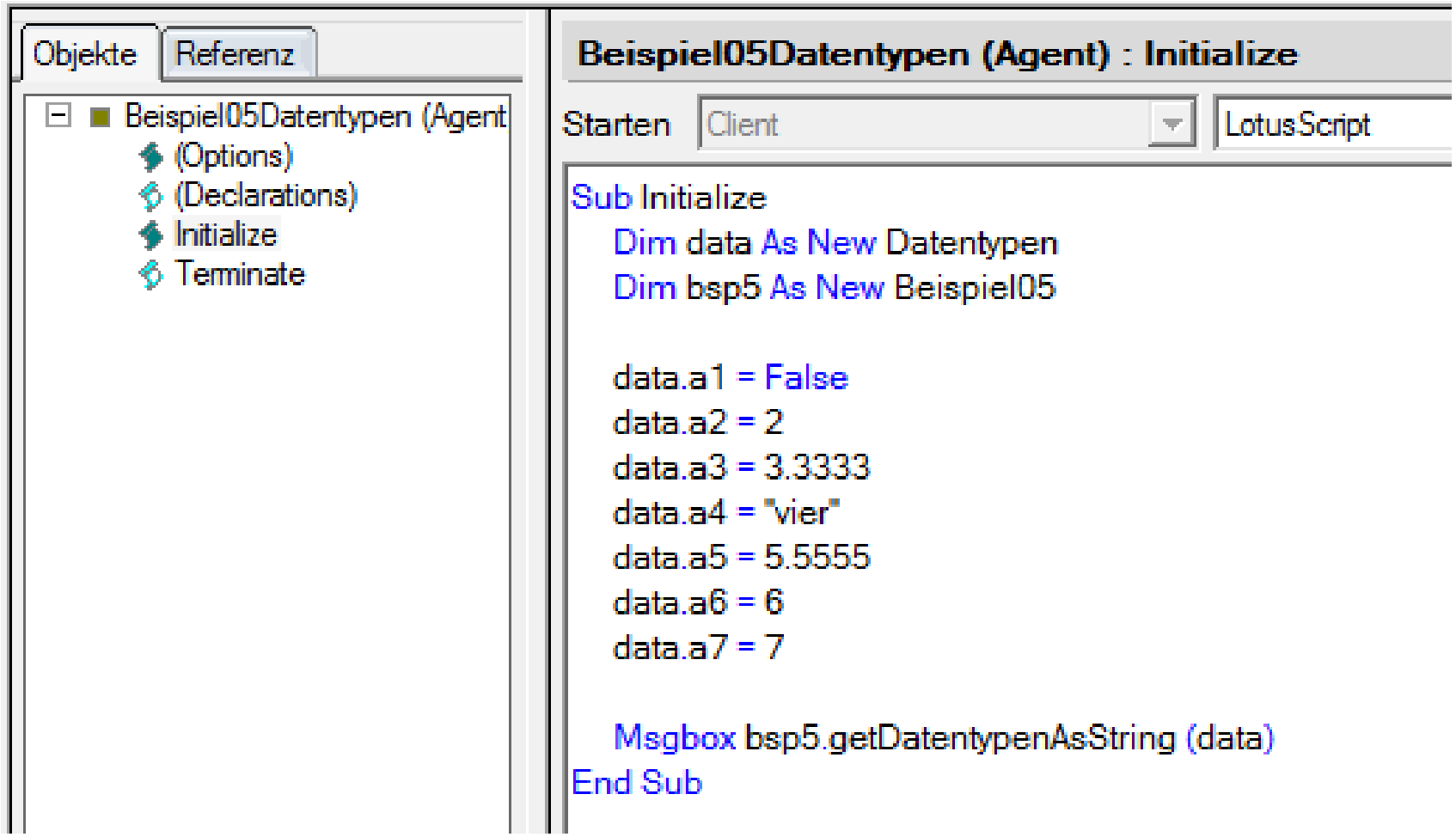
Der fertige Consumer kann nun in Agenten, Buttons usw. eingebunden werden.

Konsumieren - Erstellen eines Consumers

Der fertige Consumer kann nun in Agenten, Buttons usw. eingebunden werden.



Konsumieren - Erstellen eines Consumers



Konsumieren - Erstellen eines Consumers

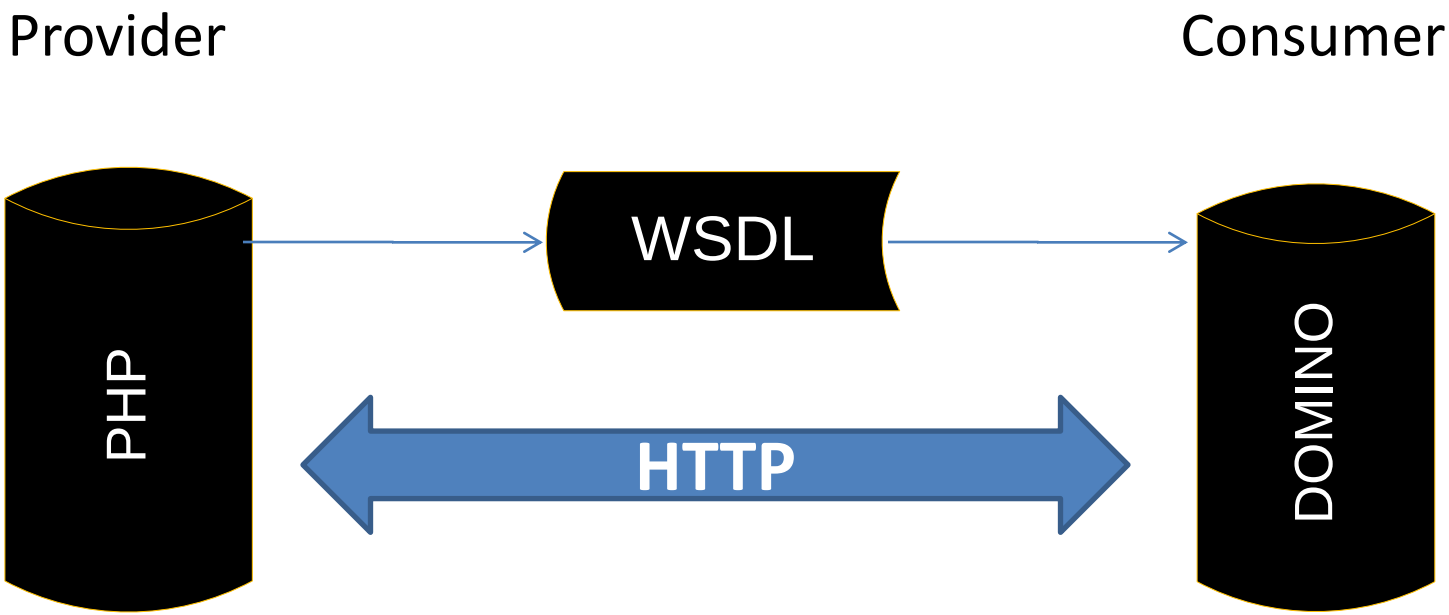
Ein Webservice Consumer in PHP erstellt :

1. $bsp1url = "http://server/dbnsf/provider?WSDL";
2. $bsp1client = new SoapClient($bsp1url);
3. $res1 = $bsp1client->helloworld();

Kommunikation

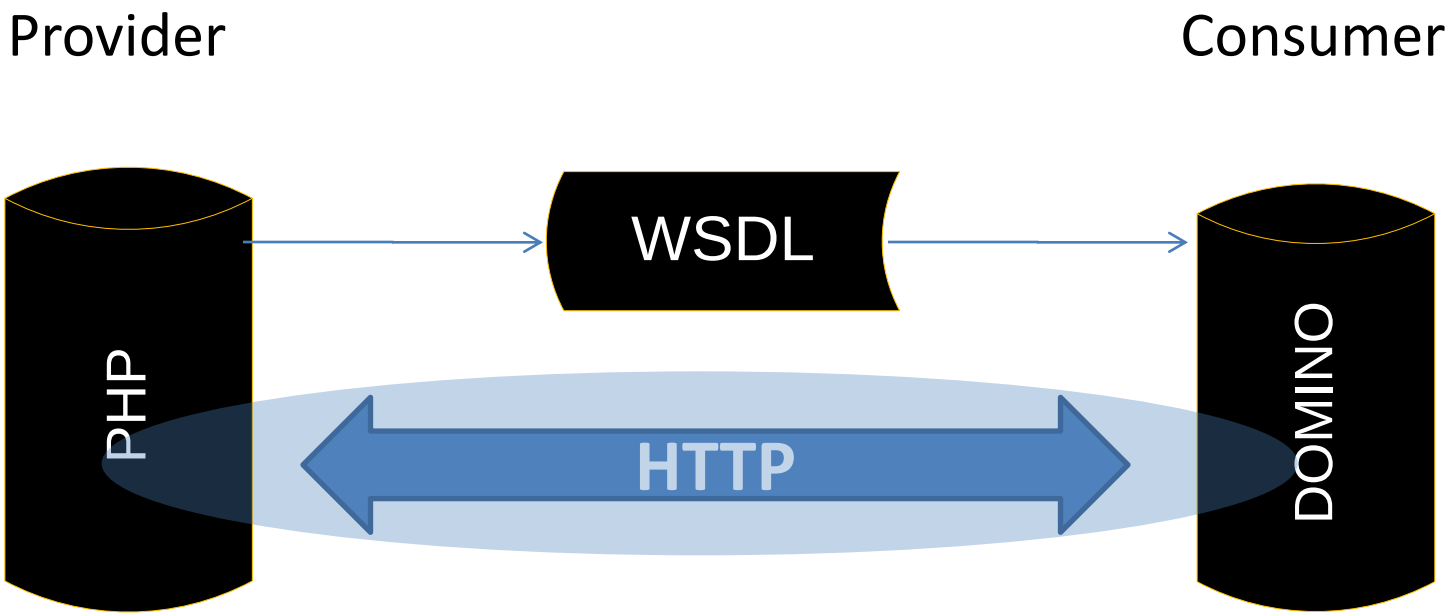
Datenaustausch per XML

Kommunikation – Datenaustausch per XML



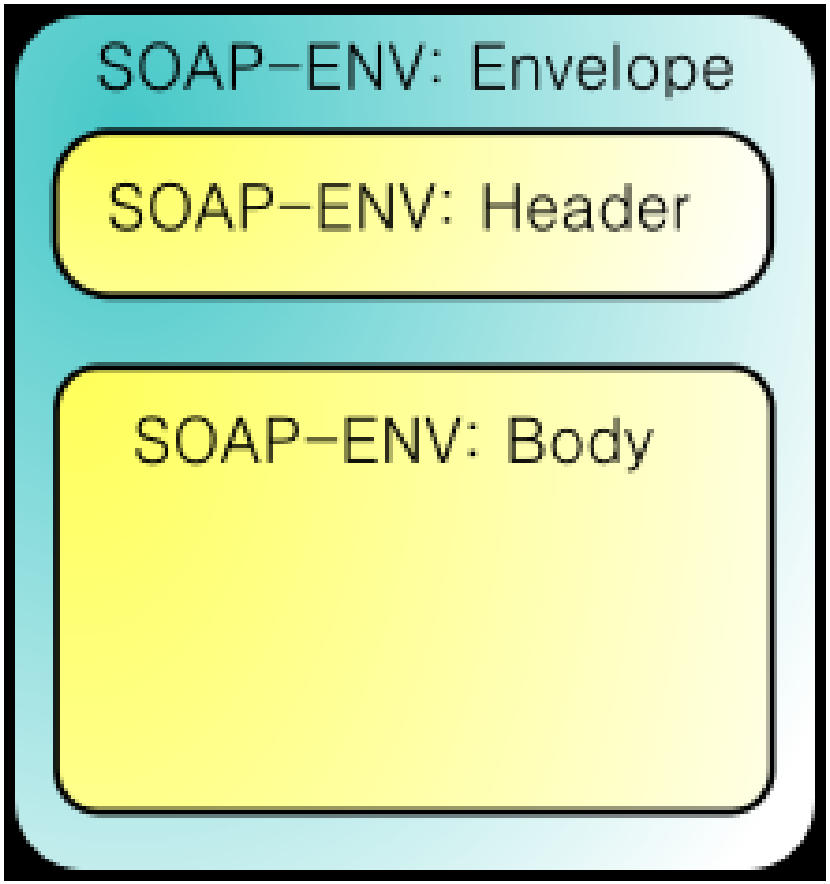
-Einfacher Funktionsaufruf mit Parameter aus PHP heraus

Kommunikation – Datenaustausch per XML



-Einfacher Funktionsaufruf mit Parameter aus PHP heraus

Kommunikation – Datenaustausch per XML



Kommunikation – Datenaustausch per XML

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
   <s:Body>
      <m:TitleInDatabase xmlns:m="http://www.lecture-db.de/soap">
         DOM, SAX und SOAP
      </m:TitleInDatabase>
   </s:Body>
</s:Envelope>
```

Kommunikation – Datenaustausch per XML

```
<?xml version="1.0"?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">
   <s:Header>
      <m:RequestID xmlns:m="http://www.lecture-
db.de/soap">a3f5c109b</m:RequestID>
   </s:Header>
   <s:Body>
      <m:DbResponse xmlns:m="http://www.lecture-db.de/soap">
         <m:title value="DOM, SAX und SOAP">
            <m:Choice value="1">Arbeitsbericht Informatik</m:Choice>
            <m:Choice value="2">Seminar XML und Datenbanken</m:Choice>
         </m:title>
      </m:DbResponse>
   </s:Body>
</s:Envelope>
```

DEMO

Online-Shop
Kaffeekochen

DEMO

Authentification Amazon

Give-Aways

- ws.Setcredentials(*"User", "Password")*
  - Run as Web User

- Ws.Setendpoint(„url“)
  - Lenkt den Webservice auf diese URL
  - Zur Laufzeit (nicht im Consumer Hardcodieren)
  - Danke Bob Balaban!

Fragen

???

Vielen Dank

Tim Pistor

Tim Pistor

Hornschuchstrasse 4

95336 Mainleus

Tel. : 09229 973260

timpistor@qmesh.de

Bleedyellow
Greenhouse
Skype (timpistor)
ICQ : 307-369-605
XING
Atnotes.de

Weiterführende Infos:

- Designerhilfe (ab 8?)
- Lsxsd.lss (im Notesverzeichnis)