

... Notes kann alles*!

(auch Kaffee kochen)

Werner Motzet und Tim Pistor präsentieren:

Webservices mit Lotus Notes

Webservices mit Lotus Notes

SSL WSDL SOAP
PHP
Kaffee XML
HTTP
Isx LotusScript Provider
RPC
IPhone
Java Consumer

Tim Pistor

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

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

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

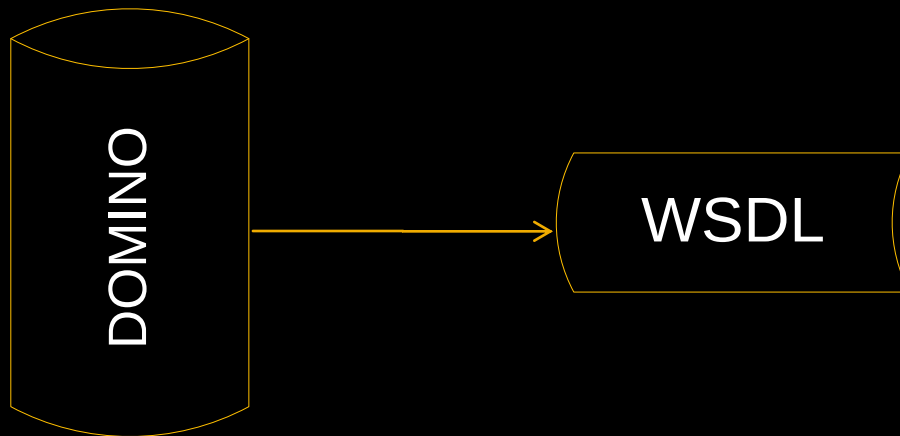
Provider



- Einfacher Funktionsaufruf
- Einfacher Funktionsaufruf mit Parameter

Demo – 1

Provider

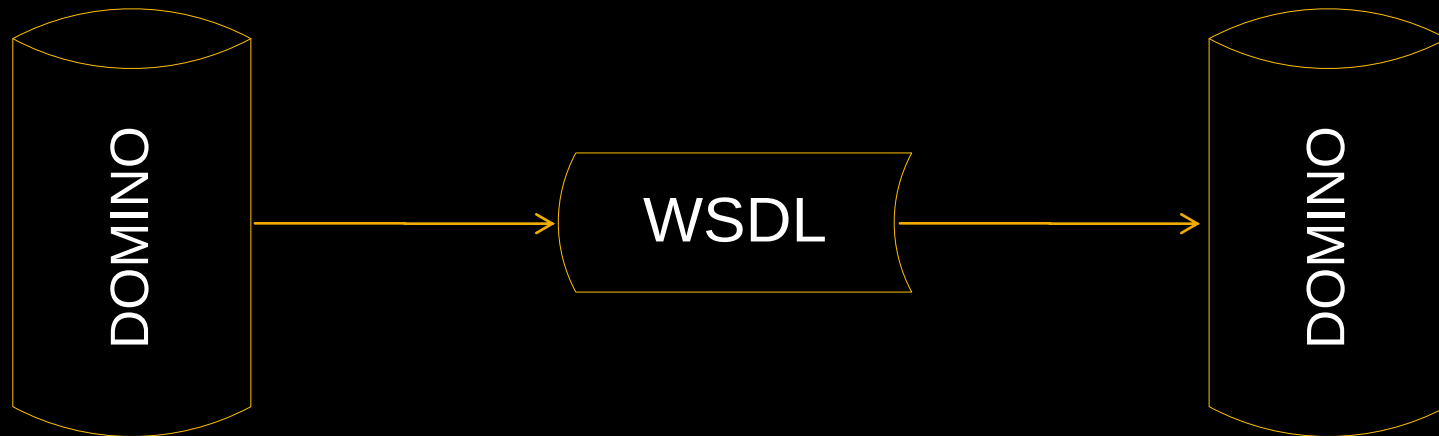


- Einfacher Funktionsaufruf
- Einfacher Funktionsaufruf mit Parameter

Demo – 1

Provider

Consumer

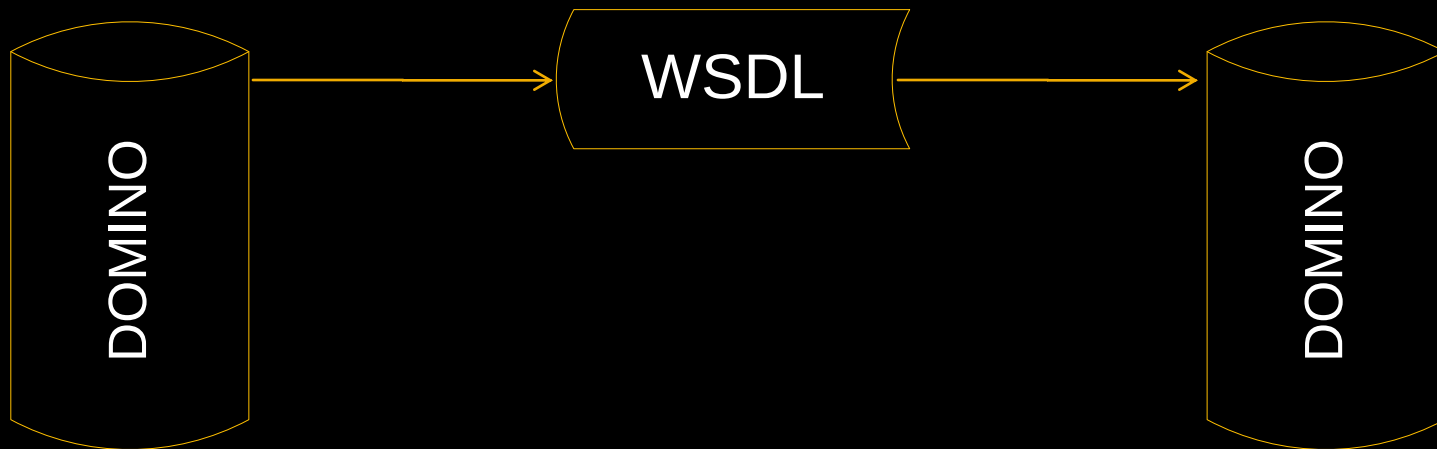


- Einfacher Funktionsaufruf
- Einfacher Funktionsaufruf mit Parameter

Demo – 1

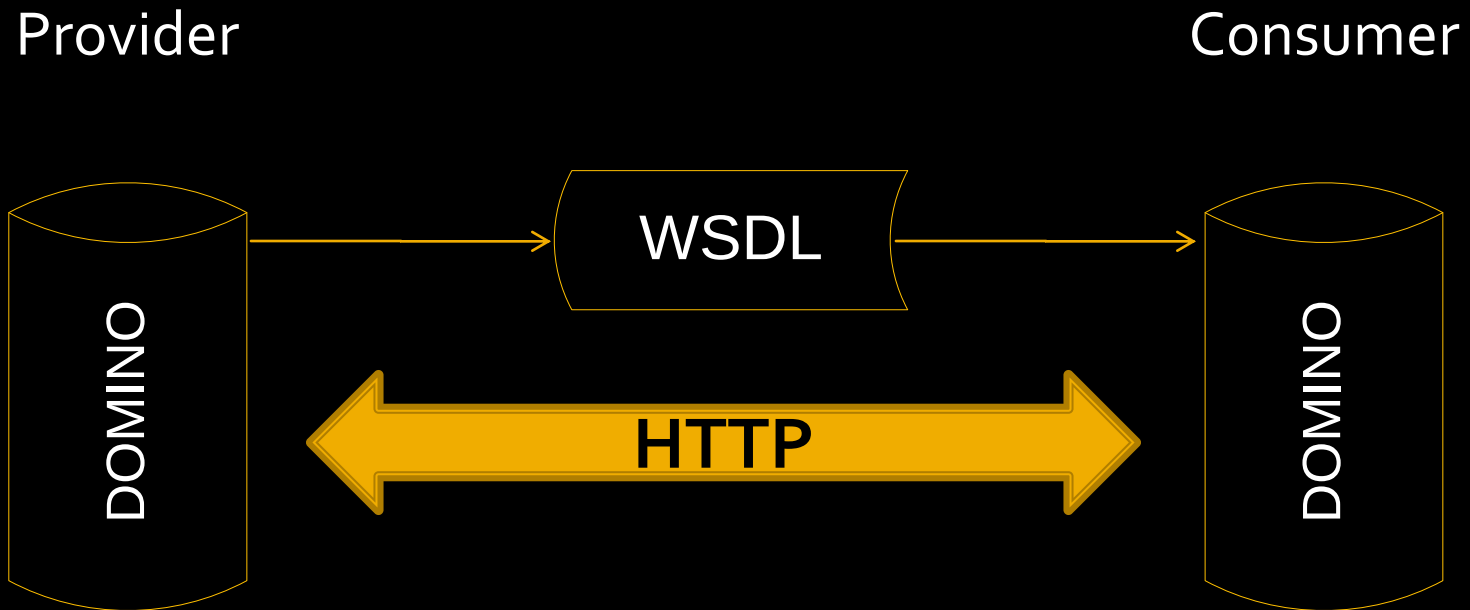
Provider

Consumer



- 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>
  </service>
</definitions>
```

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>
  </service>
</definitions>
```

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)

Liste der möglichen Nachrichten(typen) zwischen Provider und Consumer möglichen.

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

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


WSDL – Die Schnittstellenbeschreibung

- Datentypen (types)
- Schnittstellentypen (portType)
- Ports (port)
- Nachrichten (messages)
- Bindung (binding)
- 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)
- Schnittstellentypen (portType)
- Ports (port)
- Nachrichten (messages)
- **Bindung (binding)**
- 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)
- Schnittstellentypen (portType)
- **Ports (port)**
- Nachrichten (messages)
- Bindung (binding)
- 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>
```

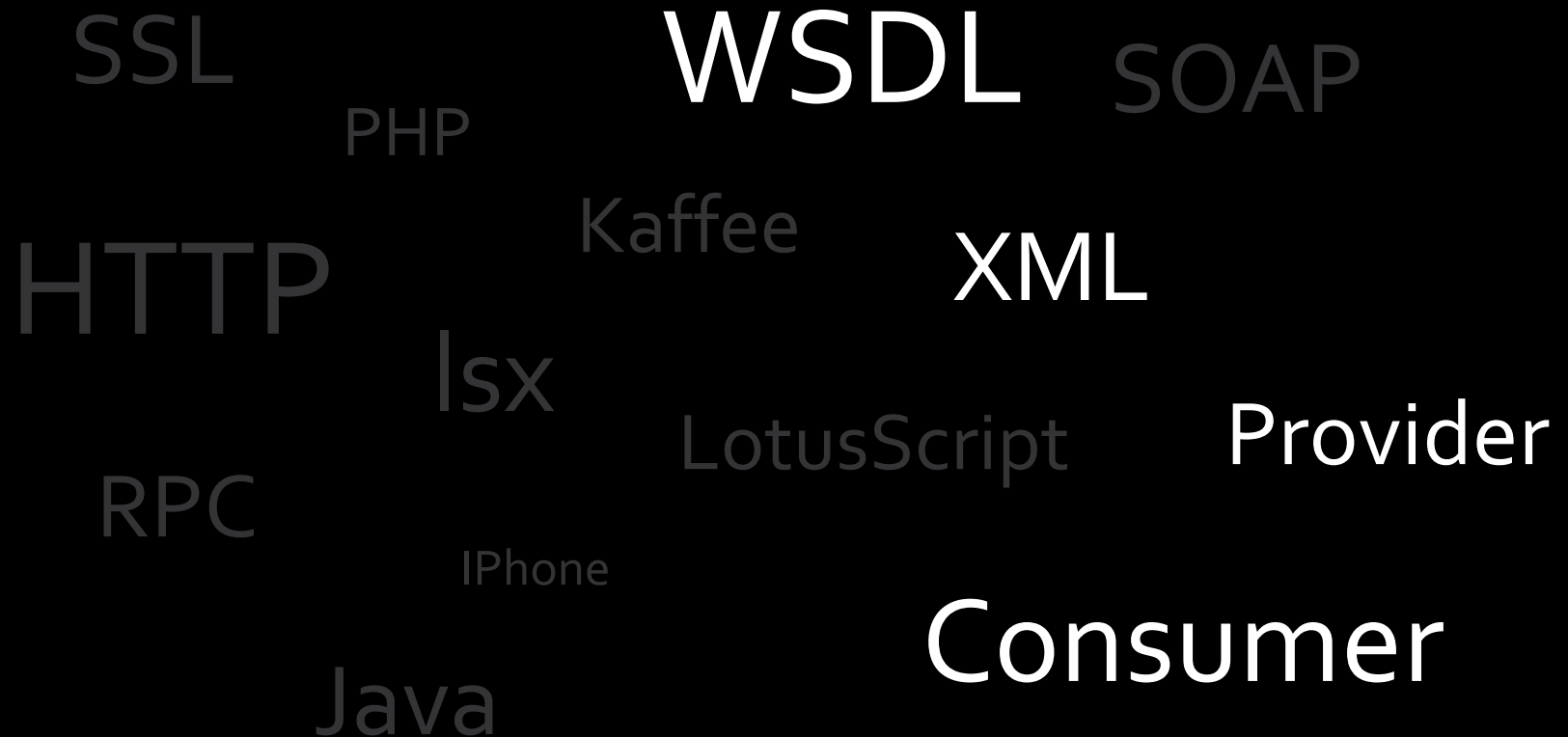
WSDL – Die Schnittstellenbeschreibung

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



A word cloud on a black background featuring various terms related to web services and Lotus Notes. The words are in different sizes and colors (white and grey). The largest word is 'WSDL' in white. Other prominent words include 'XML', 'Provider', 'Consumer', 'HTTP', 'SOAP', 'RPC', 'Kaffee', 'LotusScript', 'Isx', 'SSL', 'PHP', 'Java', 'iPhone', and 'Kaffee'.

WSDL

SOAP

XML

Provider

Consumer

HTTP

RPC

Isx

LotusScript

Kaffee

SSL

PHP

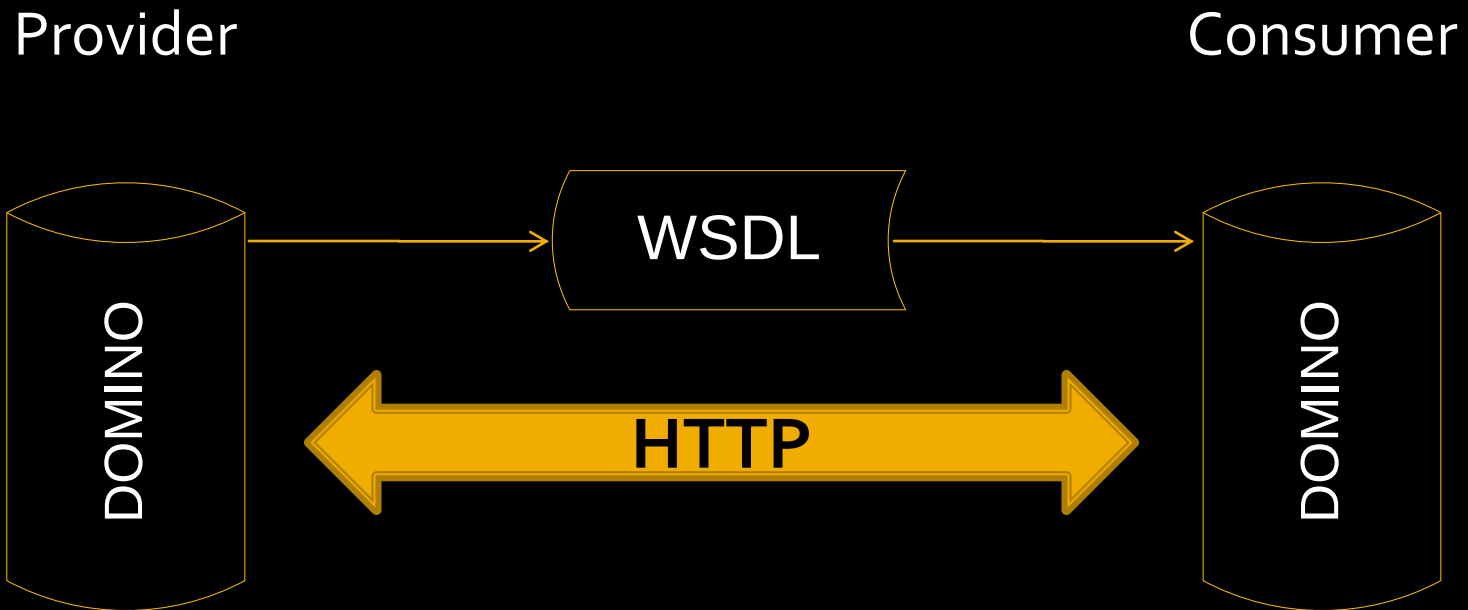
Java

iPhone

Datentypen

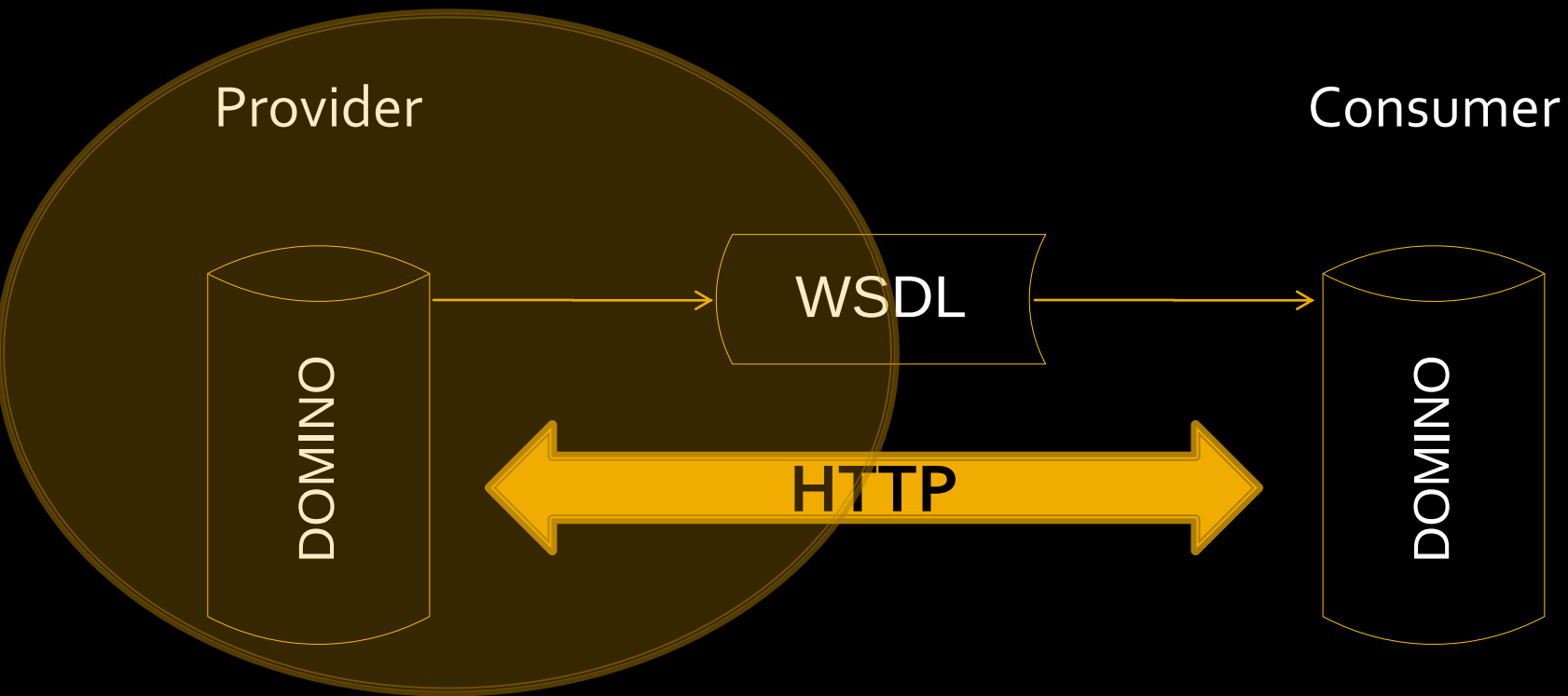
Erstellen eines Webservice Provider

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

The screenshot displays the SAP NetWeaver Designer interface. On the left, the 'Anwendungen' (Applications) tree shows the project structure for 'EC 2010 Webservice Provider'. The 'Code' folder is expanded, showing 'Agenten', 'Gemeinsame Aktionen', 'Scriptbibliotheken', 'Datenbanksript', and 'Web-Service-Provider'. The 'Web-Service-Provider' folder contains 'Beispiel01HelloWorld' and 'Beispiel02WSDLView'. The 'Web-Service-Konsumenten' folder is also visible.

The main workspace shows the 'Startseite' (Home) view of the 'home\vortraege\ec10provider.nsf - G...' project. It includes a toolbar with buttons for 'Neuer Web-Service-Provider', 'Signieren', 'WSDL exportieren', and 'WSDL anzeigen'. Below the toolbar is a table listing the existing providers:

Name	Alias	Geändert	Geändert von	Kommentar
Beispiel01HelloWorld		21.02.2010 19:4...	Tim Pistor/qmesh	
Beispiel02WSDLView		27.02.2010 09:5...	Tim Pistor/qmesh	

The 'Neuer Web-Service-Provider' dialog box is open, showing the 'Neuen Web-Service-Provider erstellen.' (Create new Web-Service-Provider) step. The 'Name' field is set to 'Beispiel03Demo2'. The 'Anwendung' (Application) dropdown is set to 'EC 2010 Webservice Provider : \\comsrv01/qmesh\home\vortraege\ec10provider.nsf'. The dialog has 'OK' and 'Abbrechen' (Cancel) buttons.

Datentypen / Erstellen eines Providers

2. Klasse „Beispiel03“ anlegen.

The screenshot displays the Lotus Notes Web Service Provider interface. The left sidebar shows the project structure under 'EC 2010 Webservice Provider', with 'Web-Service-Provider' expanded to show 'Beispiel03Demo2'. The main window has tabs for 'Startseite', 'home\vortraege\ec10provider.nsf - G...', and 'Beispiel03Demo2 - Web-Service'. Below the tabs are buttons for 'WSDL importieren', 'WSDL exportieren', 'WSDL anzeigen', and 'Print'. The 'Objekte' tab is active, showing a tree view of 'Beispiel03Demo2 (Web-Serv)' with sub-items: '(Options)', '(Declarations)', 'Initialize', and 'Terminate'. The right pane, titled 'Beispiel03Demo2 (Web-Service) : (Declarations)', shows a dropdown for 'Starten' set to 'Client' and a dropdown for 'LotusScript'. The code editor displays the following LotusScript code:

```
Class Beispiel03
  Function eineFunktionMitParameter( par As Boolean) As String
    If par Then
      eineFunktionMitParameter = "ja"
    Else
      eineFunktionMitParameter = "nein"
    End If
  End Function
End Class
```

Datentypen / Erstellen eines Providers

3. Provider konfigurieren

The screenshot displays the Lotus Notes application window with the following components:

- Left Navigation Pane:** Shows the 'EC 2010 Webservice Provider' hierarchy. The 'Code' folder is expanded, showing 'Agenten', 'Gemeinsame Aktionen', 'Scriptbibliotheken', 'Datenbanksript', and 'Web-Service-Provider'. Under 'Web-Service-Provider', 'Beispiel03Demo2' is selected.
- Top Bar:** Contains tabs for 'Startseite', 'home\vortraege\ec10provider.nsf - G...', and 'Beispiel03Demo2 - Web-Service'. Below the tabs are buttons for 'WSDL importieren', 'WSDL exportieren', 'WSDL anzeigen', and 'Print'.
- Object List:** Under the 'Objekte' tab, the tree structure for 'Beispiel03Demo2 (Web-Serv)' is shown, including '(Options)', '(Declarations)', 'Initialize', and 'Terminate'.
- Declarations Editor:** The 'Beispiel03Demo2 (Web-Service) : (Declarations)' window is active. It shows a 'Starten' dropdown set to 'Client' and a 'LotusScript' dropdown. The code area contains:

```
Class Beispiel03
Function eineFunktionMitParameter( par As Boolean) As String
    If par Then
        eineFunktionMitParameter = "ja"
    Else
        eineFunktionMitParameter = "nein"
    End If
End Function
End Class
```
- Web-Service Properties:** A floating window titled 'Web-Service' is open, showing the following fields:
 - Name: Beispiel03Demo2
 - Alias: (empty)
 - Kommentar: (empty)
 - Optionen: ☐ Wamen, wenn WSDL-Schnittstelle modifiziert wird
 - Port Type: Beispiel03
 - Klasse: (empty)

Datentypen / Erstellen eines Providers

3. Provider konfigurieren

The screenshot displays the LotusScript IDE interface for configuring a Web-Service Provider. The left sidebar shows the project structure under 'Anwendungen', with 'EC 2010 Webservice Provider' expanded to show 'Web-Service-Provider' containing 'Beispiel03Demo2'. The main workspace shows the 'Beispiel03Demo2 (Web-Service) : (Declarations)' window with the following code:

```
Class Beispiel03
Function eineFunktionMitParameter( par As Boolean) As String
If par Then
eineFunktionMitParameter = "ja"
Else
eineFunk
End If
End Function
End Class
```

Below the code, the 'Web-Service' configuration panel is visible, showing the following settings:

- ☒ Als Webbenutzer ausführen
- Ausführen im Namen von: [Empty field]
- ☐ Remote-Debugging zulassen
- ☐ Profiling für diesen Web-Service aktivieren
- Laufzeit-Sicherheitsstufe festlegen: (1 = am sichersten)
1. Beschränkte Operationen nicht zulassen
- Vorgabezugriff für diesen Web-Service:
 - ☒ Alle Leser und höhere
 - OtherDomainServers: comsrv01/qmesh, Tim Pistor/qmesh
- ☐ Benutzer mit öffentl. Zugriff dürfen diesen Web-Service nutzen

Datentypen / Erstellen eines Providers

3. Provider konfigurieren

The screenshot displays the LotusScript IDE interface for configuring a Web-Service Provider. The left sidebar shows a project tree under 'EC 2010 Webservice Provider'. The main workspace is divided into three panes:

- Objekte:** A tree view showing the structure of 'Beispiel03Demo2 (Web-Service)', including 'Options', 'Declarations', 'Initialize', and 'Terminate'.
- Referenz:** A pane for references, currently empty.
- Beispiel03Demo2 (Web-Service) : (Declarations):** The main editor showing the LotusScript code for the service.

The code in the Declarations pane is as follows:

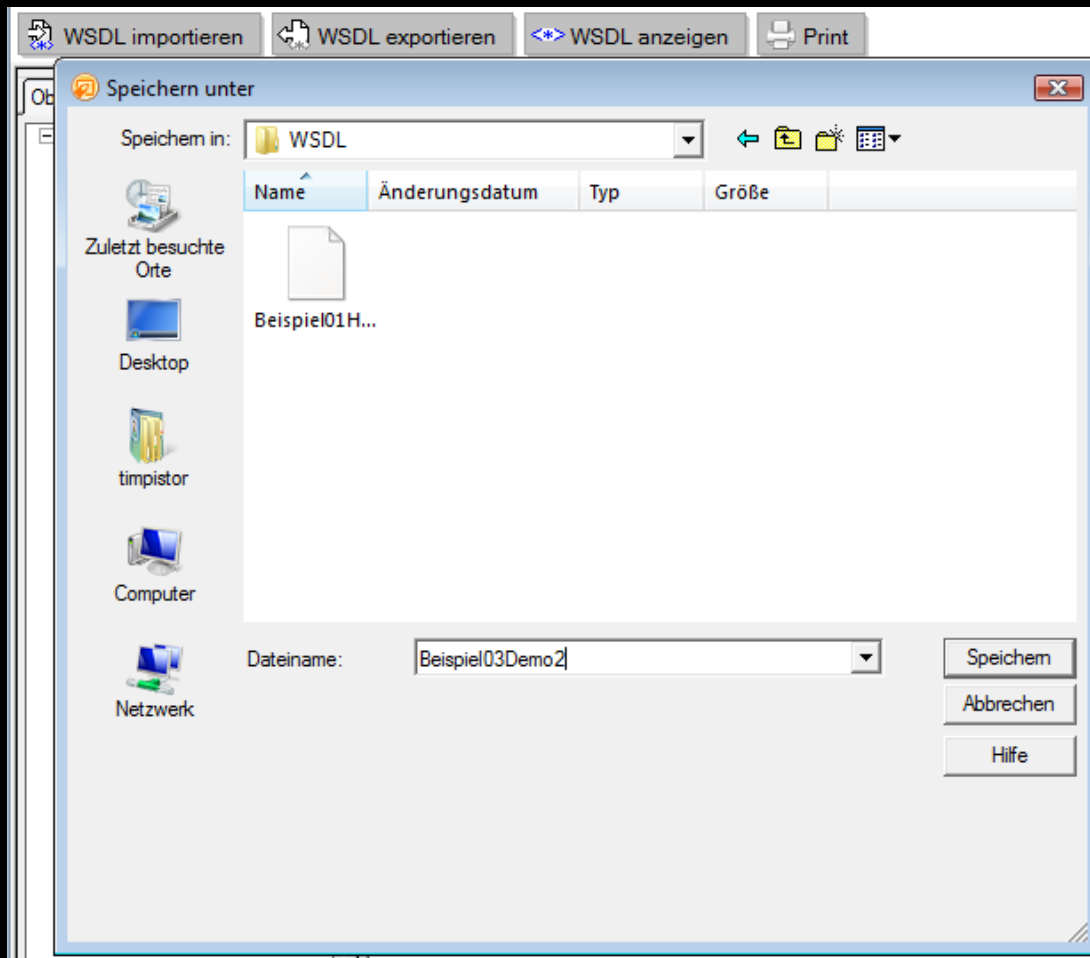
```
Class Beispiel03
  Function eineFunktionMitParameter( par As Boolean) As String
    If par Then
      eineFunktionMitParameter = "ja"
    Else
      eineFunk
    End If
  End Function
End Class
```

Below the code, a 'Web-Service' configuration dialog is open, showing the following settings:

- Starten:** Client
- LotusScript:** (selected)
- Programmiersprache:** (selected)
- Options:**
 - Programmiersprache: ☒ RPC ☐ Nachricht
 - SOAP-Nachrichtenformat: Doc/literal
 - ☒ Operationsname in SOAP-Aktion aufnehmen
- Port Type-Name:** Beispiel03
- Service-Element-Name:** Beispiel03Service
- Service-Port-Name:** Domino

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

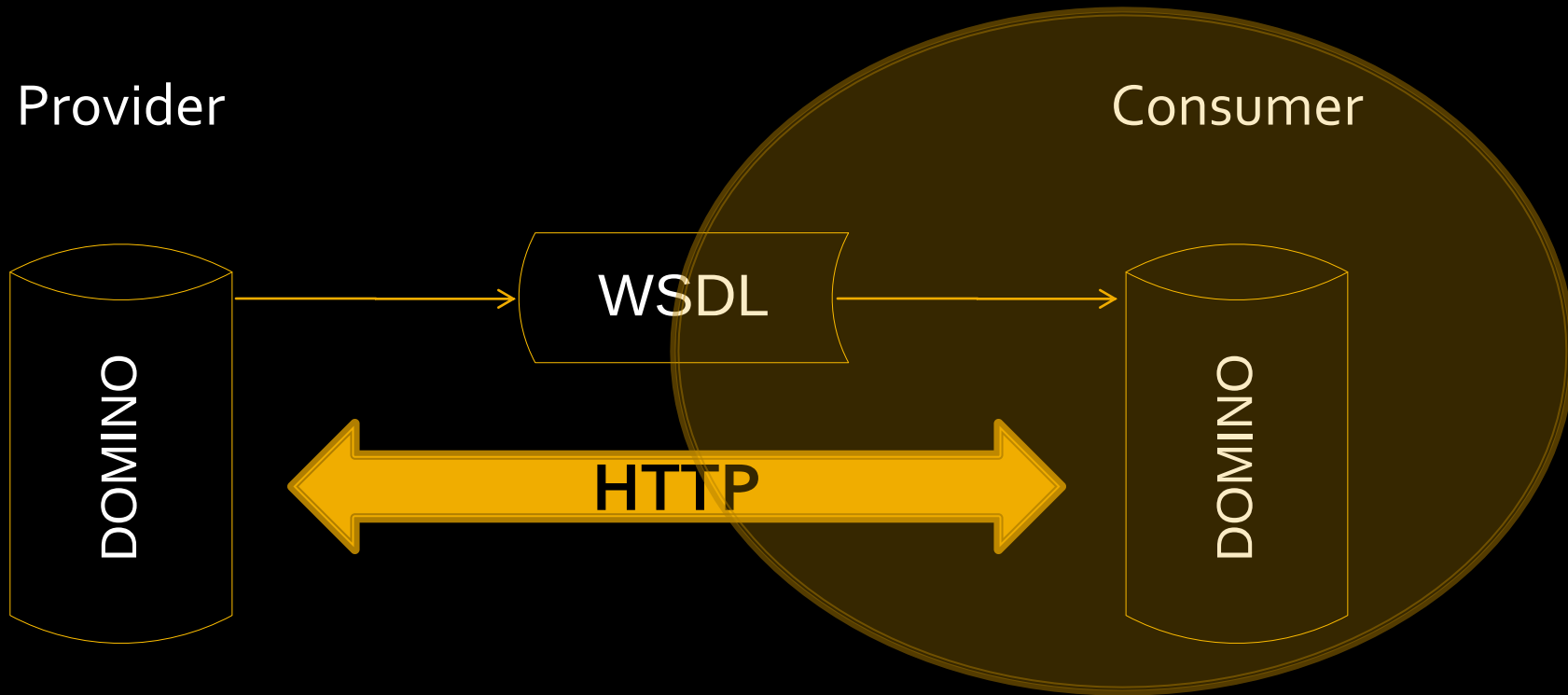
```
1. Class Beispielo4
2.     Function getDatentypen() As Datentypen
3.         Set getDatentypen = New Datentypen
4.     End Function
5. End Class

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

The screenshot displays the Lotus Notes application interface. On the left is a navigation pane with a tree view containing folders like 'Masken', 'Ansichten', 'Ordner', 'XPages', and 'Web-Service-Konsumenten'. The main window has a browser-like address bar with two tabs: 'home\vortraege\ec10provider.nsf' and 'home\vortraege\ec10consumer.nsf'. Below the address bar are buttons for 'Neuer Web-Service-Konsument', 'Signieren', 'WSDL exportieren', and 'WSDL anzeigen'. A table lists existing consumers, with one entry: 'Beispiel01HelloWorldConsumer' of type 'LotusScript'.

The 'Neuer Web-Service-Konsument' dialog box is open, showing the following fields and options:

- Name:** Beispiel05Datentypen
- Sprache:** LotusScript
- Web-Service-Beschreibung abrufen von:**
 - ☒ Lokale WSDL-Datei: Z:\Vortrag EC 2010\WSDL\Beispiel05Datentypen.wsdl (with a 'Durchsuchen...' button)
 - ☐ URL, die auf eine WSDL-Datei verweist
- Anwendung:** EC 2010 Webservice Consumer : \\comsrv01\qmesh\home\vortraege\ec10consumer.nsf

At the bottom of the dialog are 'OK' and 'Abbrechen' buttons.

Konsumieren - Erstellen eines Consumers

The screenshot displays a software development interface with two main panels. The left panel, titled 'Objekte' and 'Referenz', shows a project tree for 'Beispiel05Datentypen (Web-)'. The right panel, titled 'Beispiel05Datentypen (Web-Service-Konsument) : (Declarations)', shows the code declarations for the web service consumer.

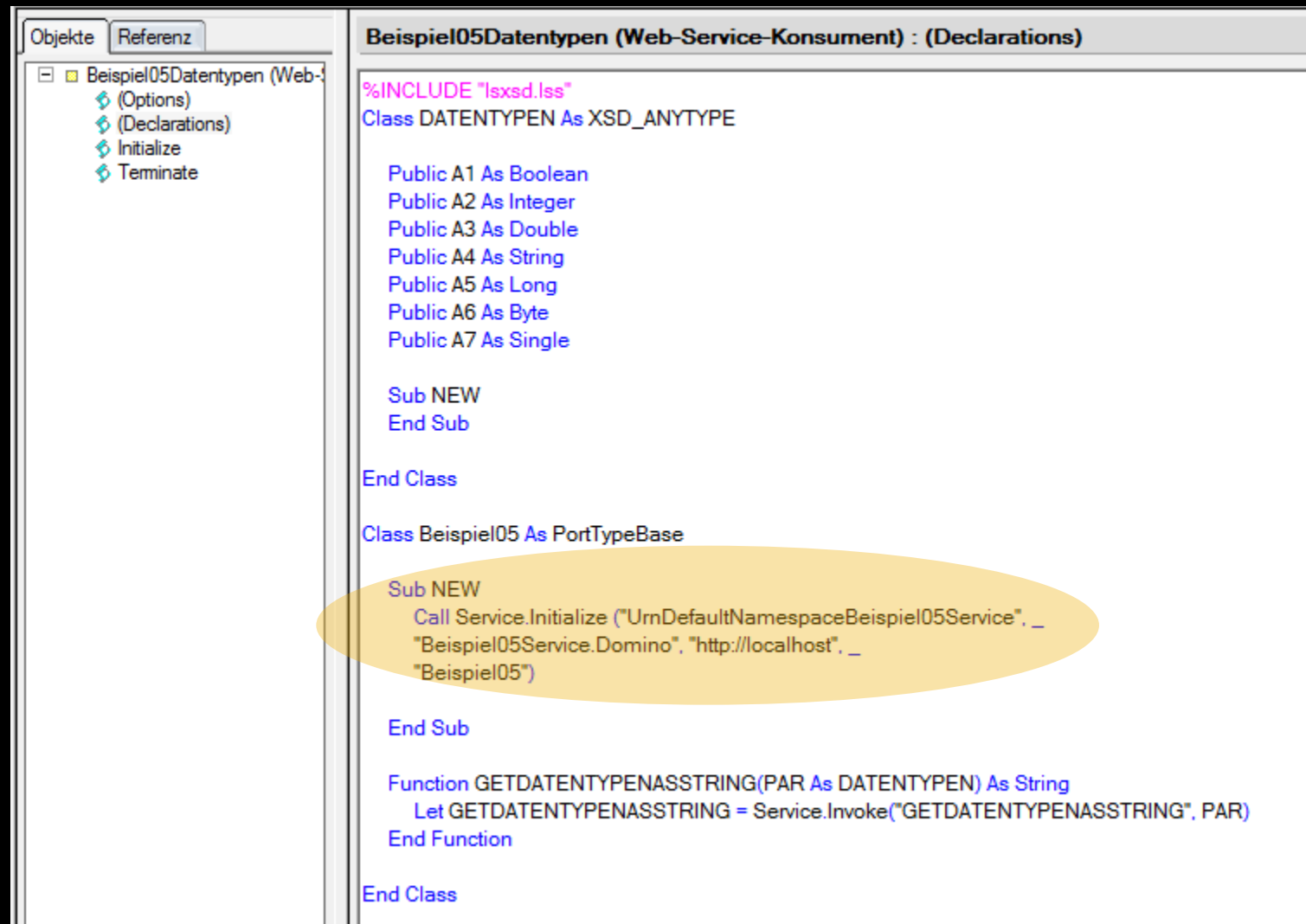
Objekte | **Referenz**

- Beispiel05Datentypen (Web-)
 - (Options)
 - (Declarations)
 - Initialize
 - Terminate

Beispiel05Datentypen (Web-Service-Konsument) : (Declarations)

```
%INCLUDE "Isxsd.lss"  
Class DATENTYPEN As XSD_ANYTYPE  
  
    Public A1 As Boolean  
    Public A2 As Integer  
    Public A3 As Double  
    Public A4 As String  
    Public A5 As Long  
    Public A6 As Byte  
    Public A7 As Single  
  
    Sub NEW  
    End Sub  
  
End Class  
  
Class Beispiel05 As PortTypeBase  
  
    Sub NEW  
        Call Service.Initialize ("UrnDefaultNamespaceBeispiel05Service", _  
            "Beispiel05Service.Domino", "http://localhost", _  
            "Beispiel05")  
    End Sub  
  
    Function GETDATENTYPENASSTRING(PAR As DATENTYPEN) As String  
        Let GETDATENTYPENASSTRING = Service.Invoke("GETDATENTYPENASSTRING", PAR)  
    End Function  
  
End Class
```


Konsumieren - Erstellen eines Consumers



Objekte Referenz

Beispiel05Datentypen (Web-Service-Konsument)

- (Options)
- (Declarations)
- Initialize
- Terminate

Beispiel05Datentypen (Web-Service-Konsument) : (Declarations)

```
%INCLUDE "Isxsd.lss"
Class DATENTYPEN As XSD_ANYTYPE

    Public A1 As Boolean
    Public A2 As Integer
    Public A3 As Double
    Public A4 As String
    Public A5 As Long
    Public A6 As Byte
    Public A7 As Single

    Sub NEW
    End Sub

End Class

Class Beispiel05 As PortTypeBase

    Sub NEW
        Call Service.Initialize ("UrnDefaultNamespaceBeispiel05Service", _
            "Beispiel05Service.Domino", "http://localhost", _
            "Beispiel05")
    End Sub

    Function GETDATENTYPENASSTRING(PAR As DATENTYPEN) As String
        Let GETDATENTYPENASSTRING = Service.Invoke("GETDATENTYPENASSTRING", PAR)
    End Function

End Class
```

Konsumieren - Erstellen eines Consumers

The screenshot shows a software development environment with two main panes. The left pane, titled 'Objekte', contains a tree view for 'Beispiel05Datentypen (Web-)'. The 'Declarations' node is selected and highlighted with a blue dashed border. Below it are 'Initialize' and 'Terminate' nodes. The right pane, titled 'Beispiel05Datentypen (Web-Service-Konsument) : (Declarations)', displays the following code:

```
%INCLUDE "Isxsd.lss"
Class DATENTYPEN As XSD_ANYTYPE

    Public A1 As Boolean
    Public A2 As Integer
    Public A3 As Double
    Public A4 As String
    Public A5 As Long
    Public A6 As Byte
    Public A7 As Single

    Sub NEW
    End Sub

End Class

Class Beispiel05 As PortTypeBase

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

    Function GETDATENTYPENASSTRING(PAR As DATENTYPEN) As String
        Let GETDATENTYPENASSTRING = Service.Invoke("GETDATENTYPENASSTRING", PAR)
    End Function

End Class
```

A yellow oval highlights the 'Sub NEW' block of the 'Beispiel05' class, which contains the initialization logic for the service.

Konsumieren - Erstellen eines Consumers

```
Sub NEW
```

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

```
End Sub
```

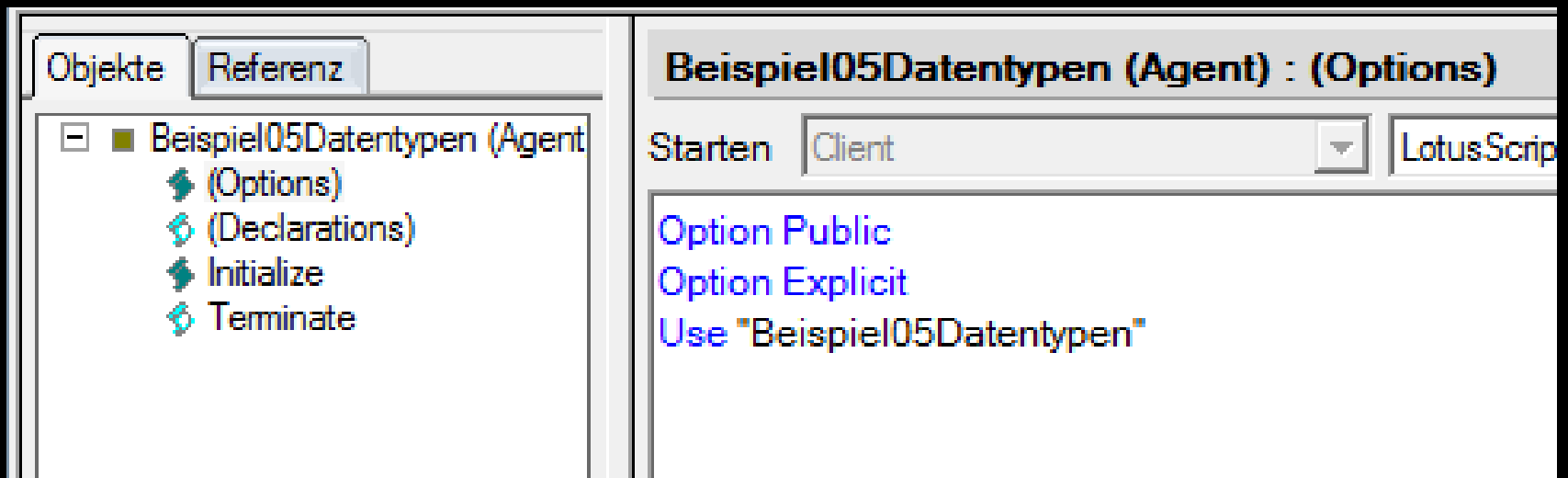
- URL zur Datenbank + „/“ Webservicename

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

The screenshot displays the LotusScript development environment. On the left, a project tree under 'Objekte' shows the agent 'Beispiel05Datentypen (Agent)' with its methods: '(Options)', '(Declarations)', 'Initialize', and 'Terminate'. The 'Referenz' tab is active. The main editor window, titled 'Beispiel05Datentypen (Agent) : Initialize', shows the 'Starten' button set to 'Client' and the script language set to 'LotusScript'. The code in the editor defines a 'Sub Initialize' that declares two arrays, 'data' and 'bsp5', and initializes them with specific values. It then calls a method 'getDatentypenAsString' on 'bsp5' and displays the result in a message box.

```
Sub Initialize
    Dim data As New Datentypen
    Dim bsp5 As New Beispiel05

    data.a1 = False
    data.a2 = 2
    data.a3 = 3.3333
    data.a4 = "vier"
    data.a5 = 5.5555
    data.a6 = 6
    data.a7 = 7

    MsgBox bsp5.getDatentypenAsString (data)
End Sub
```

Konsumieren - Erstellen eines Consumers

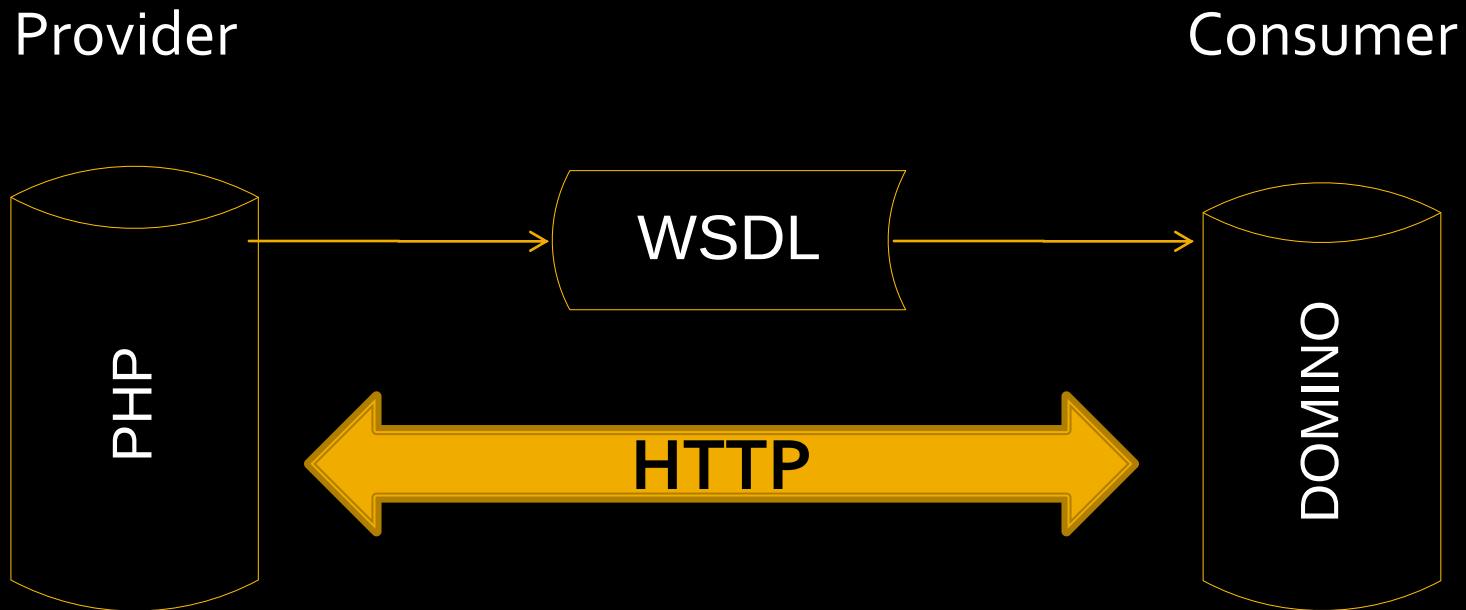
Ein Webservice Consumer in PHP erstellt :

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

Kommunikation

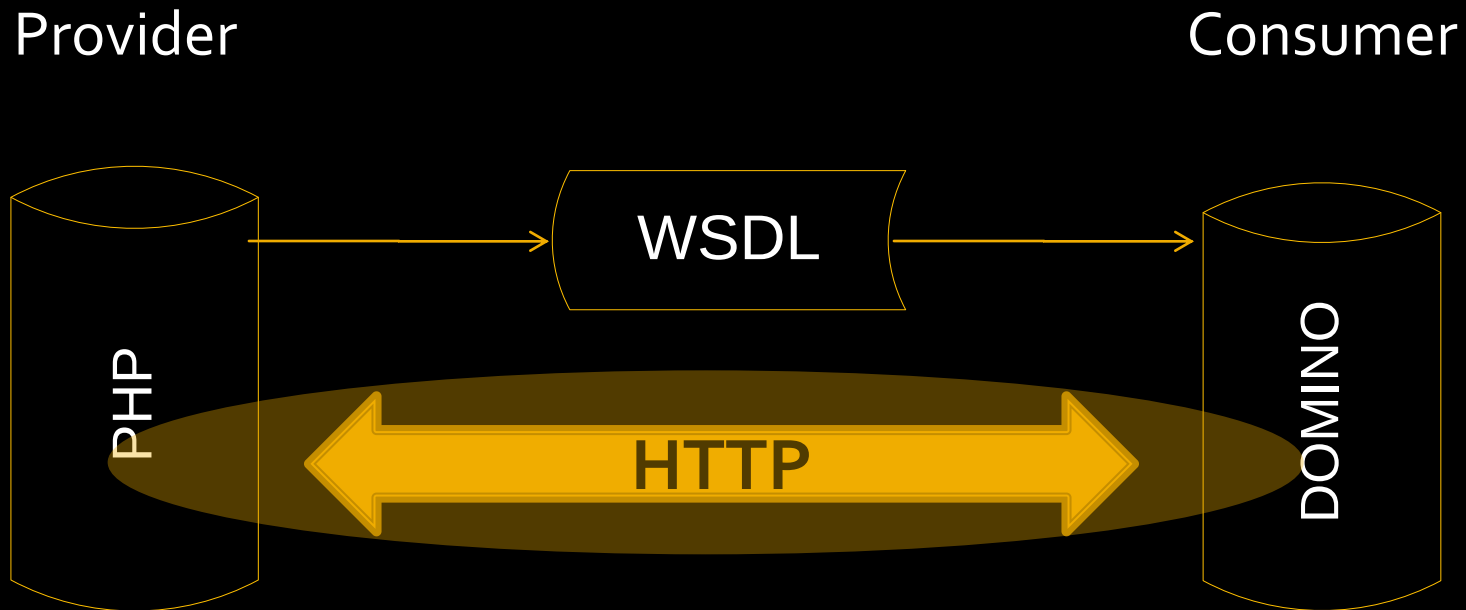
Datenaustausch per XML

Kommunikation – Datenaustausch per XML



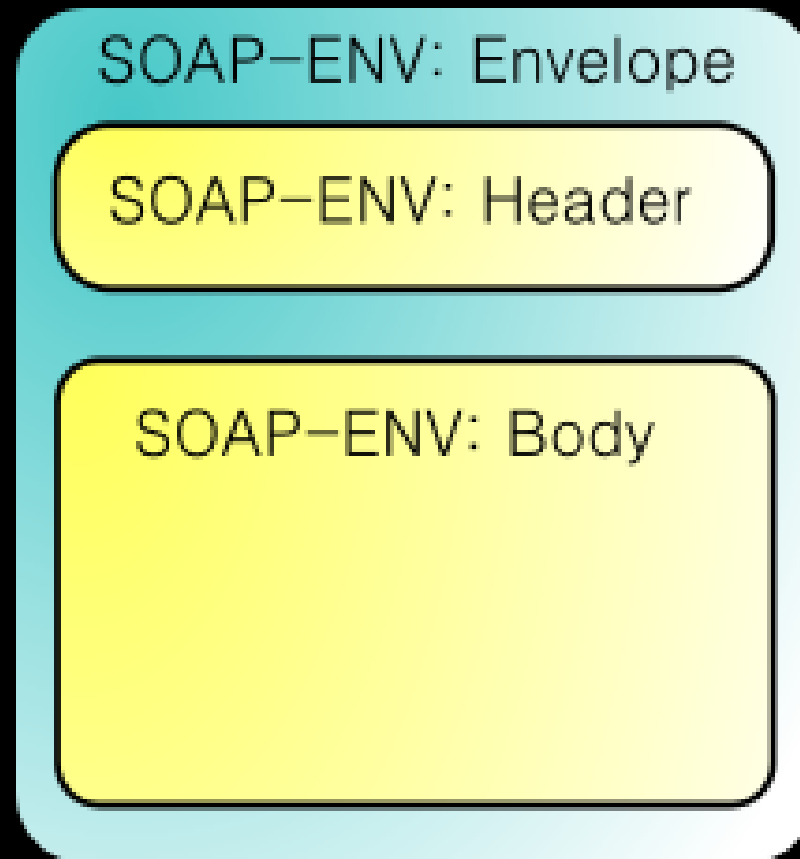
-Einfacher Funktionsaufruf mit Parameter aus PHP heraus

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

Authentication
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?)
- Lxsd.Iss (im Notesverzeichnis)