Webservices

Veerle Ongenae

Overzicht

- Gegevens uitwisselen
- SOAP
- Webservices
  - Technologieën
  - Gebruik

Gegevens uitwisselen

- Data uitwisselen tussen programma's op verschillende computers
- Tot nu toe
  - EDI
  - CORBA, RMI, DCOM

EDI

- Electronic Data Interchange
- WAN voor een geografische regio, waar partners kunnen inpluggen
- Afspraak: boodschappen en protocol voor datatransfer
- Networkdetails overgelaten aan de implementatie
- Verschillende netwerken
  - Duur om in te stappen
  - Moeilijk en duur om andere netwerken te bereiken

CORBA, RMI, DCOM

- Afkortingen
  - Common Object Request Broker Architecture
  - Remote Method Invocation
  - Distributed Component Object Model
- Gedistribueerde object-infrastructuur over het Internet
- Transportprotocol bovenop TCP/IP (verschillend voor CORBA, RMI en DCOM) voor verschillende platformen
- Afspraken over communicatie tussen objecten

CORBA, RMI, DCOM

- Voordeel: de verschillende servers hoeven niet tot het zelfde netwerk te behoren
- Kunnen enkel met dezelfde infrastructuur praten, tenzij extra lagen op een reeds ingewikkelde structuur

SOAP

- SOAP zorgt voor verandering:
  - Mogelijkheid tot het uitwisselen van data overal op het web
  - Geen koppeling tussen data en transport
  - Niet binair

Overzicht

- Gegevens uitwisselen
- SOAP
- Webservices
  - Technologieën
  - Gebruik

SOAP

- Simple Object Acces Protocol
- Protocol gebaseerd op XML
- Uitwisseling van informatie in een gedecentraliseerde, gedistribueerde omgeving
- Transport via het web (HTTP, …)
- SOAP-bericht
  - Verstuurd door zender
  - Naar ontvanger

SOAP en HTTP

- De eerste lijn en de headerlijnen blijven bestaan
- Het corpus van een HTTP-aanvraag en HTTP-antwoord bevat XML ipv HTML, …
- Analoog via FTP of SMTP

SOAP

- XML-envelop
  - Met XML-inhoud
- Een aantal regels voor servers die een SOAP-boodschap ontvangen
- Maakt gebruik van bestaande transportprotocols (bv. HTTP)
- Oorspronkelijk Microsoft, ondertussen standaard van W3C

Wat is er nodig om data uit te wisselen via SOAP?

- Afgesproken DTD of XML Schema voor de uit te wisselen gegevens
- SOAP-server
  - Afhandelen aanvragen
- Client
  - Software om XML-documenten te genereren

Overzicht SOAP

- XML-tags die een SOAP-bericht beschrijven en die een framework geven voor de inhoud van het bericht
- Regels voor het uitwisselen van zelf gedefinieerd gegevenstypes (o.a. aanvaarden, verwerpen, uitzonderingen)
- Conventies voor het uitvoeren van methodes op een andere server en het voorstellen van het resultaat

SOAP Bericht: voorbeeld aanvraag

POST /ZwiftBooks HTTP/1.1
Host: www.zwiftbooks.com
Content-Type:text/xml
Content-Length: 134
SOAP Action: “Some-URI”

```
<Envelope xmlns=“http://www.w3.org/2001/09/soap-envelope”
    encodingStyle=“http://www.w3.org/2001/09/soap-encoding”>
     <Body>
          <zwiftbooks:GetGuaranteedDeliveryTime
             xmlns:zwiftbooks=“www.zwiftbooks.com”>
               <zwiftbooks:isbn>0-13-18188-222-2</zwiftbooks:isbn>
               <zwiftbooks:zipcode>75240</zwiftbooks:zipcode>
          </zwifbooks:GetGuaranteedDeliveryTime>
     </Body>
</Envelope>
```

SOAP Bericht: voorbeeld antwoord

HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 122

```
<Envelope xmlns="http://www.w3.org/2001/09/soap-envelope">
    <Body>
        <zwiftbooks:GetBestDeliveryTimeResponse
          xmlns:zwiftbooks="www.zwiftbooks.com">
            <zwiftbooks:Time>2 hours</zwiftbooks:Time>
        </zwiftbooks:GetBestDeliveryTimeResponse>
    </Body>
</Envelope>
```

Structuur SOAP Bericht

- SOAP-envelop
- SOAP-hoofding
- SOAP-corpus

Structuur SOAP Bericht: envelop

- SOAP-envelop:
  - Envelope-element
    - Attributen
      - encodingStyle: hoe geserialiseerd wordt
  - Rootelement van het XML-document dat een SOAP-bericht voorstelt
  - Verplicht

Structuur SOAP Bericht: hoofding

- SOAP-hoofding:
  - Header-element
  - Extra richtlijnen en informatie voor SOAP-server
    - Transacties
    - Beveiliging
    - Filters
    - …
  - Optioneel

Voorbeeld: SOAP-hoofding

```
<Header>
    <n:alertcontrol
      xmlns:n="http://example.org/alertcontrol">
        <n:priority>1</n:priority>
        <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
</Header>
```

Structuur SOAP Bericht: corpus

- SOAP-corpus:
  - Body-element
  - Bevat de uit te wisselen XML
    - Onafhankelijk van SOAP-structuur
    - Domein-specifieke structuur
  - Inhoud
    - Data/Resultaat
    - Aanroep van een methode
  - Verplicht

SOAP-berichtenpad

- Bestaat uit SOAP-knopen:
  - SOAP-zender
  - Nul of meerdere tussenliggende SOAP-servers
  - SOAP-ontvanger
- Tussenliggende SOAP-servers
  - Stuurt SOAP-bericht door
  - Behandelt de voor hem bedoelde SOAP-hoofdingen

SOAP-rollen

- Elke SOAP-knoop heeft één of meerdere rollen
- Mogelijke rollen
  - Gespecifieerd door SOAP:
    - none (niet bedoeld voor een specifieke knoop)
    - next (ontvanger en alle tussenliggende)
    - ultimateReceiver (ontvanger)
  - Gespecifieerd door de applicatie
    - Proxy
    - Beveiliging
    - Caching
    - …

SOAP-rollen

- Een SOAP-hoofding kan aangeven voor welke SOAP-knoop de hoofding bedoeld is door gebruik te maken van rollen
  - Attribuut role
  - Waarde attribuut
    - Rol gespecifieerd door SOAP
    - URI SOAP-knoop

Voorbeeld: SOAP-hoofding

```
<Header
   role="http://schemas.xmlsoap.org/soap/actor/next">
    <n:alertcontrol xmlns:n="http://example.org/alertcontrol"
       mustUnderstand="true">
         <n:priority>1</n:priority>
         <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
</Header>
```

Regels voor SOAP-server

- Indentificatie van de stukken van het SOAP-bericht die voor de huidige server bedoeld zijn
- Nagaan of deze server alle delen van de headers die bedoeld zijn voor deze server en die het attribuut mustUnderstand hebben ondersteunt. Indien niet fout genereren

Regels voor SOAP-server

- Uitvoeren van de stukken van de headers bedoeld voor deze server
- Indien niet de SOAP-ontvanger: alle headers bedoeld voor deze server verwijderen en bericht doorsturen

SOAP-fouten

- Wanneer
  - Niet begrijpen bericht
  - Fout bij behandelen van bericht
- fault-element
  - Kind van body-element
  - Kinderen
    - faultcode: foutcode
    - faultstring: leesbare verklaring
    - detail: informatie over het probleem

Overzicht

- Gegevens uitwisselen
- SOAP
- Webservices
  - Technologieën
  - Gebruik

Webservices

- Kader
  - Applicaties bouwen op basis van losse componenten
- Wat?
  - Diensten aanbieden voor applicaties
  - Vergelijking
    - Presentatielaag maakt gebruik van diensten van de applicatielogica
- Technologie
  - Verschillende protocollen

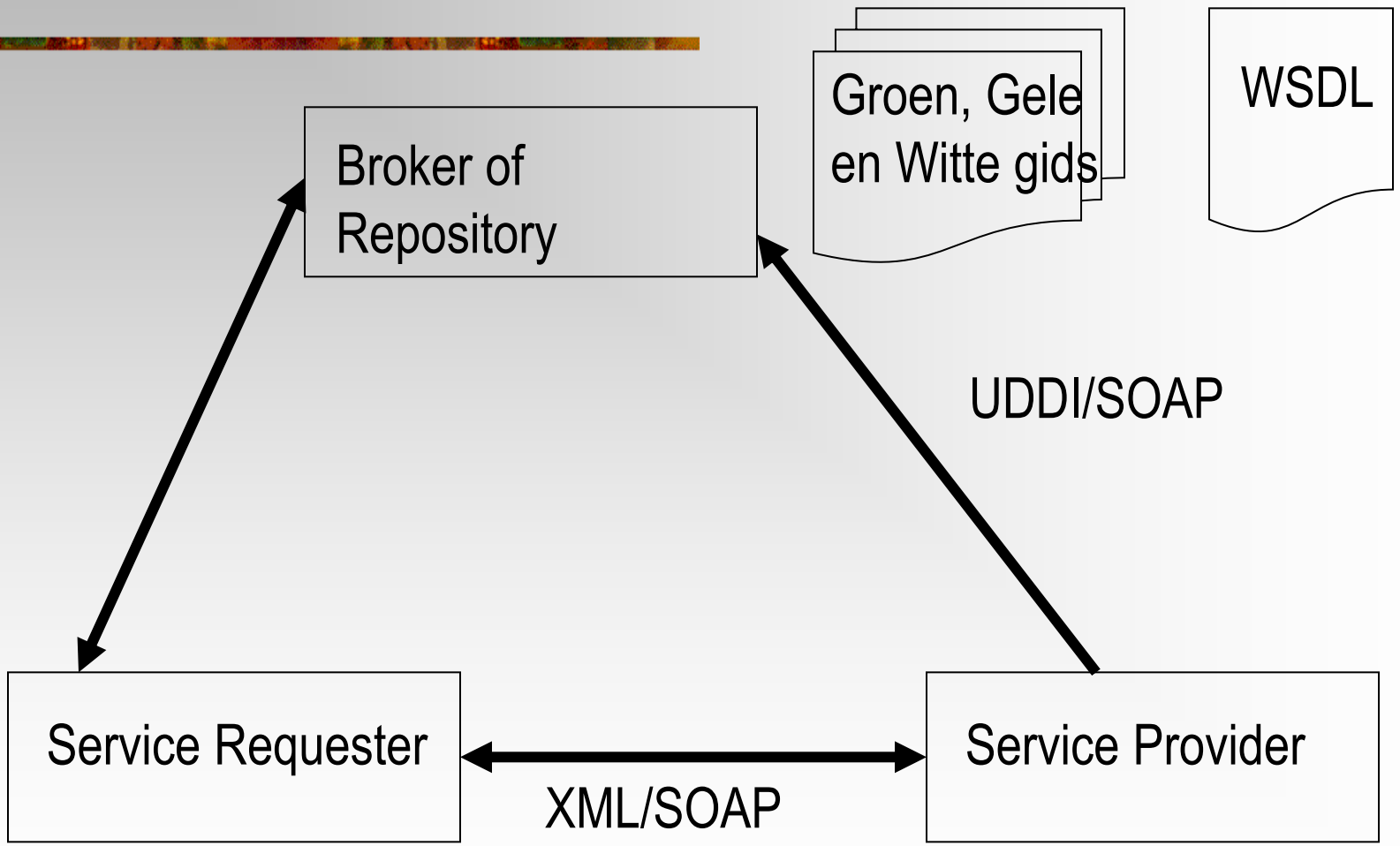
Verschillende facetten van webservices

- Beschrijving
  - Aangeboden diensten
- Publiceren
  - Bij een ‘repository’
  - Vergelijkbaar met telefoonboek: witte, gele en groene gids
- Uitvoeren
  - Een applicatie maakt gebruik van de aangeboden diensten
- Antwoorden
  - De dienst stuurt een resultaat terug naar de applicatie

Webservices: onderdelen

- Service provider
  - Biedt een interface
  - Voor software
  - Bepaalde set taken
- Service requester
  - Doet een aanvraag bij service provider
  - Krijgt antwoord van service provider
  - Maakt deel uit van een applicatie
- Broker
  - Publiceert beschikbare diensten

Overzicht webservices



Webservices: technologieën

- UDDI
  - Universal Description, Discovery and Integration
- WSDL
  - Web Services Definition Language
- SOAP

UDDI

- UDDI
  - XML-Protocol
  - Webservices beschrijven en registreren bij een broker
  - Communicatie tussen
    - Service provider en broker
    - Broker en service requester
  - Inhoud van een SOAP-envelop

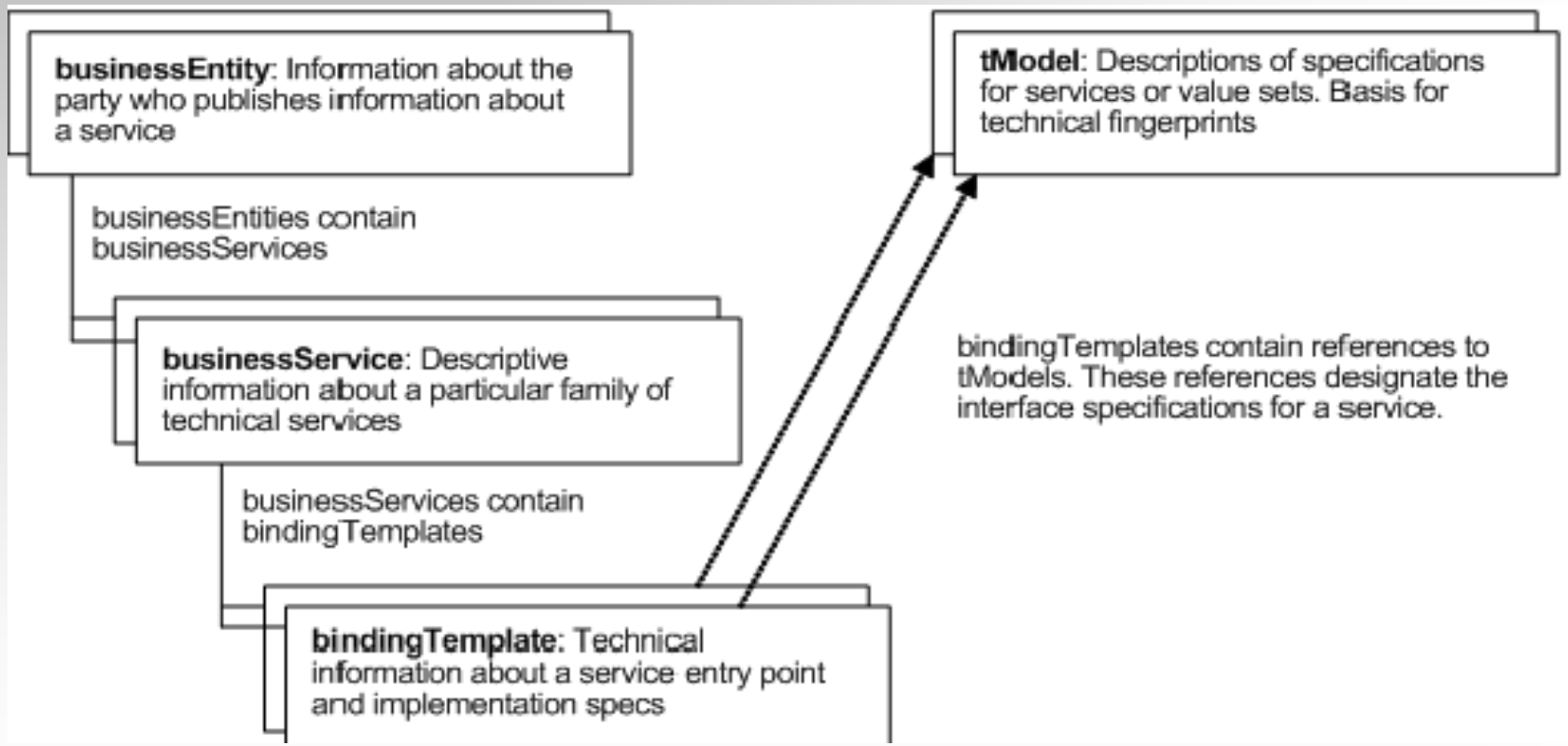
Informatie in een register

- Witte gids
  - Contactinformatie: naam, adres, webstek, …
  - Registratie van de provider
- Gele gids
  - Categorieën van diensten
  - Hoe vind een client een webservice
- Groene gids
  - Technische informatie
  - Hoe kan een client gebruik maken van een webservice

Elementen van UDDI

- businessEntity
  - Informatie over bepaalde provider
- businessService
  - Beschrijving van een bepaalde webservice bestaande uit één of meerdere diensten
- bindingTemplate
  - Technische informatie, informatie voor implementatie
- tModel
  - Technische informatie per dienst

Elementen van UDDI



WSDL

- WSDL
  - XML-standaard W3C
  - Beschrijven webservice
    - Interface
    - Implementatiedetails
  - Document kan meegestuurd worden met UDDI
  - Bepaalt hoe de clientapplicatie de aangeboden dienst moet aanspreken

WSDL

- Twee versies
  - WSDL 1.1 (http://www.w3.org/TR/2001/NOTE-wsdl-20010315)
  - WSDL 2.0 (http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/)

Voorbeeld WSDL 1.1

```
<definitions name="BoekAgent"
    targetNamespace="http://www.zwiftbooks.com/"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:tns="http://www.zwiftbooks.com/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" >

    <message name="GetDeliveryInfoRequest">
        <part name="zipcode" type="xsd:integer"/>
        <part name="isbn" type="xsd:string"/>
    </message>

    <message name="GetDeliveryInfoResponse">
        <part name="result" type="xsd:duration"/>
    </message>
```

Voorbeeld WSDL 1.1

```
<portType name="BoekAgent">
    <operation name="GetDeliveryInfo" parameterOrder="zipcode
      isbn">
        <input message="tns:GetDeliveryInfoRequest"
          name="GetDeliveryInfoRequest "/>
        <output message="tns:GetDeliveryInfoResponse "
          name="GetDeliveryInfoResponse"/>
    </operation>
    <!-- andere diensten -->
</portType>
```

Voorbeeld WSDL 1.1

```
<binding name="BoekAgentBinding" type="tns:BoekAgent">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="GetDeliveryInfo">
        <soap:operation
          soapAction="http://zbooks.org/action/ZBooks.Get-
          DeliveryInfo" />
        <input>
            <soap:body use="encoded"
              namespace="http://zbooks.org/message/"
              encodingStyle="http://schemas.xmlsoap.org/soap/en
              coding/" />
        </input>
```

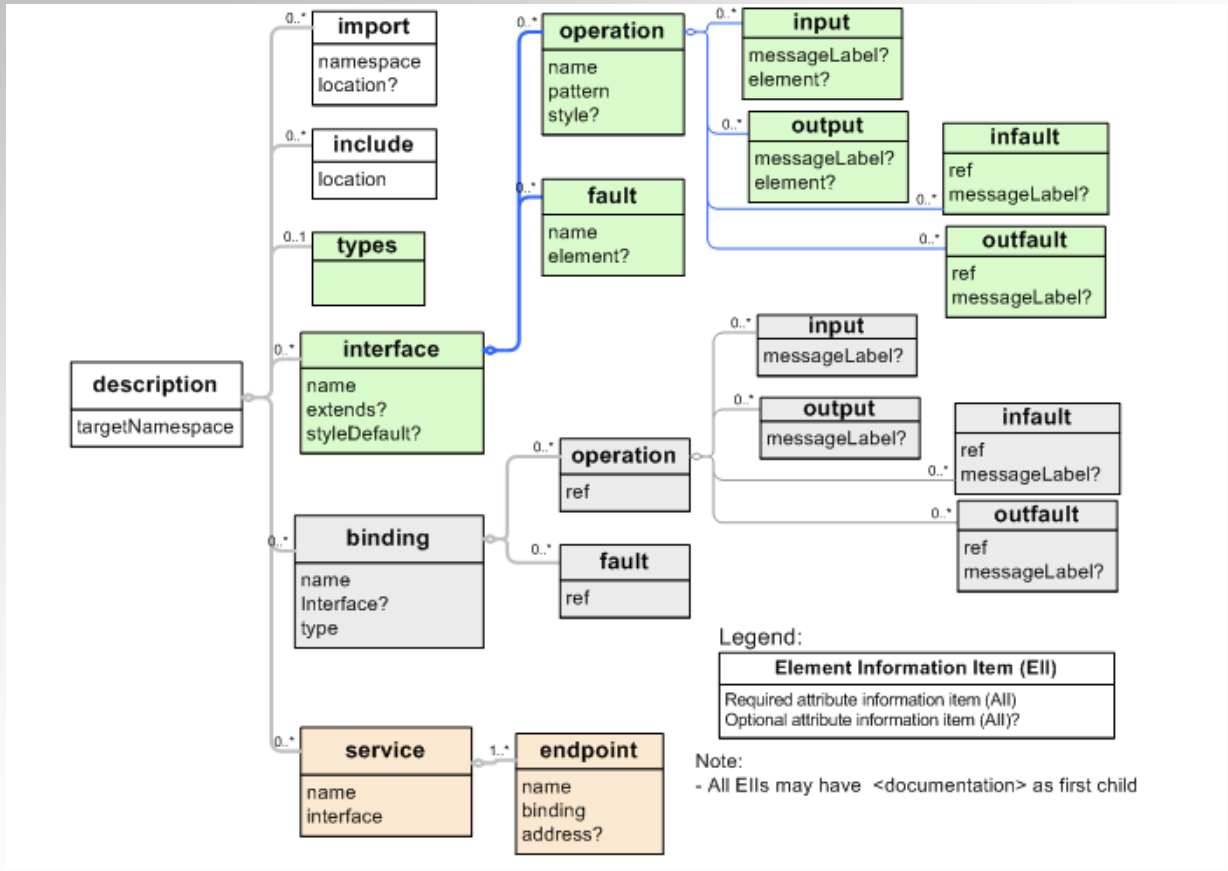
Voorbeeld WSDL 1.1

```
            <output>
                <soap:body use="encoded"
                  namespace="http://zbooks.org/message/"
                  encodingStyle="http://schemas.xmlsoap.org
                  /soap/encoding/" />
            </output>
        </operation>
    </binding>
</definitions>
```

WSDL 1.1 -structuur

- Messages
  - Berichten
- Port Types
  - Verzameling van alle diensten van een webservice
- Bindings
  - Beschrijving concreet formaat van de berichten en het gebruikte protocol
- Services
  - Beschrijving van een webservice bestaande uit verschillende “port types”

WSDL 2.0 -structuur



WSDL 2.0 -structuur

- types: beschrijving berichten in XML Schema
- interface: abstracte funcionaliteit
- binding: hoe toegang tot diensten
- service: waar toegang tot diensten

Voorbeeld WSDL 2.0

```
<?xml version="1.0" encoding="utf-8" ?>
<description xmlns="http://www.w3.org/ns/wsdl"
    targetNamespace= "http://greath.example.com/2004/wsdl/resSvc"
    xmlns:tns= "http://greath.example.com/2004/wsdl/resSvc" . . . >
    . . .
</description>
```

Voorbeeld WSDL 2.0

```
<?xml version="1.0" encoding="utf-8" ?>
<description  . . . >
    ...
<types>
    <xs:schema …>
     <xs:element name="checkAvailability"
    type="tCheckAvailability"/>
    <xs:complexType name="tCheckAvailability">
    <xs:sequence>
            <xs:element name="checkInDate" type="xs:date"/>
            <xs:element name="checkOutDate" type="xs:date"/>
            <xs:element name="roomType" type="xs:string"/>
    </xs:sequence> </xs:complexType>
        <xs:element
        name="checkAvailabilityResponse"
        type="xs:double"/>
        <xs:element name="invalidDataError"
        type="xs:string"/>
    </xs:schema>
    </types>
        . . .
    </description>
```

Voorbeeld WSDL 2.0

```
<?xml version="1.0" encoding="utf-8" ?>
<description …>
    . . . <types> ... </types>
<interface name = "reservationInterface" >
<fault name = "invalidDataFault" element = "ghns:invalidDataError"/>
<operation name="opCheckAvailability" pattern="http://www.w3.org/ns/wsdl/in-out"
    style="http://www.w3.org/ns/wsdl/style/iri" wsdlx:safe = "true">
    <input messageLabel="In" element="ghns:checkAvailability" />
    <output messageLabel="Out" element="ghns:checkAvailabilityResponse" />
    <outfault ref="tns:invalidDataFault" messageLabel="Out"/>
</operation> </interface> . . . </description>
```

Voorbeeld WSDL 2.0

```
<?xml version="1.0" encoding="utf-8" ?>
<description … > . . . <types> . . . </types>
<interface name = "reservationInterface" > ... </interface>
<binding name="reservationSOAPBinding" interface="tns:reservationInterface"
    type="http://www.w3.org/ns/wsdl/soap"
    wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/">
    <operation ref="tns:opCheckAvailability"
    wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-response"/>
    <fault ref="tns:invalidDataFault" wsoap:code="soap:Sender"/>
</binding>
. . .
</description>
```

Voorbeeld WSDL 2.0

```
<?xml version="1.0" encoding="utf-8" ?>
<description …>
. . . <types> . . . </types>
<interface name = "reservationInterface" > . . . </interface>
<binding name="reservationSOAPBinding" interface="tns:reservationInterface" . . . >
. . . </binding>
<service name="reservationService" interface="tns:reservationInterface">
    <endpoint name="reservationEndpoint" binding="tns:reservationSOAPBinding"
    address ="http://greath.example.com/2004/reservation"/>
</service>
</description>
```

Webservices: technologieën

- SOAP
  - Communicatie tussen broker, service provider en service requester
  - SOAP-corpus bevat UDDI

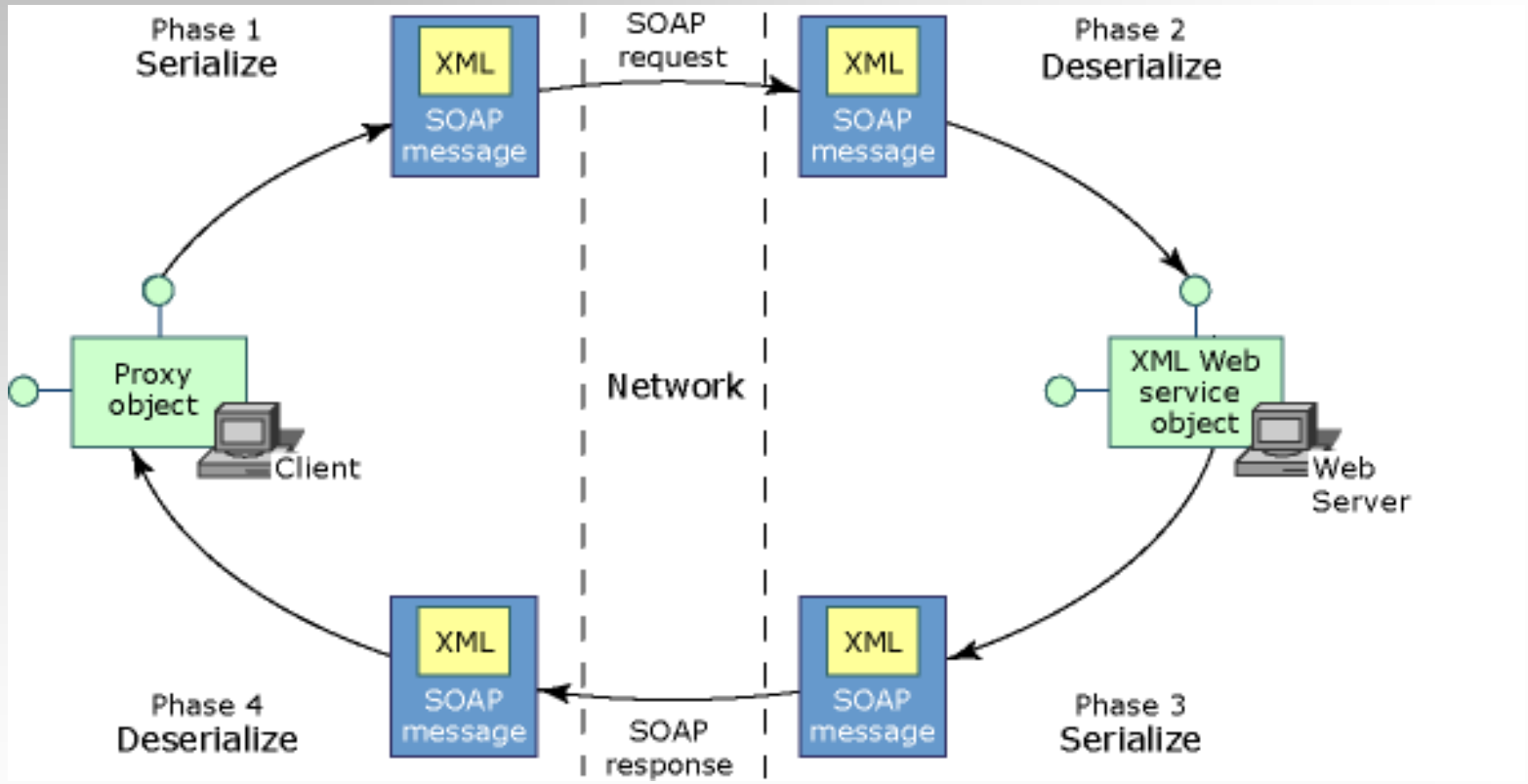
Overzicht

- Gegevens uitwisselen
- SOAP
- Webservices
  - Technologieën
  - Gebruik

Wat is een webservice?

- Uitvoeren van een methode op een andere server
- Een dienst
  - Eén of meerdere methodes
  - Een stuk programma met een bepaalde functionaliteit
  - Toegankelijk via XML (SOAP) en HTTP

Overzicht


Phase 1
Serialize
XML
SOAP
message
SOAP
request
XML
SOAP
message
Phase 2
Deserialize
Proxy
object
Client
Network
XML Web
service
object
Web
Server
XML
SOAP
message
XML
SOAP
message
Phase 4
Deserialize
SOAP
response
Phase 3
Serialize

Versturen SOAP-aanvraag

- Proxy-object: een lokaal object voor de clientapplicatie die webservice “voorstelt”
- Client-applicatie roept methodes van het proxy-object op
- Infrastructuur serialiseert → SOAP-bericht
- Bericht wordt verstuurd via het netwerk
  - Communicatie via HTTP en het SOAP-protocol

Verwerken SOAP-bericht

- Serverinfrastructuur
  - Deserialisatie van het SOAP-bericht
  - Al dan niet aanmaken object webservice
  - Methode webservice uitvoeren
  - Antwoord serialiseren → SOAP-bericht
  - SOAP-antwoord wordt verstuurd via het netwerk

SOAP-antwoord vertalen

- Clientinfrastructuur
  - Ontvangt SOAP-antwoord
  - Deserialiseert
  - Geeft resultaat door aan proxy-object
- Proxy-object genereert resultaat voor clientapplicatie

Voorbeeld

- Boek-object opvragen aan de hand van ISBN/ID
- In .NET
  - http://localhost/WebCatalogus/BoekenLijst.asmx
- In J2EE
  - http://localhost:8084/WebCatalogus/CatalogusService

SOAP

- Structureert HTTP-corpus
- Envelop-tag
  - Header-tag(s)
  - Body-tag
    - Bevat XML
    - Aan te roepen methode + argumenten
    - Resultaat van een methode

SOAP-aanvraag

```
POST /iit/webservices/Catalogus.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: ...
SOAPAction: "http://be.hogent.iii/webservices/GetBoek"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="..." xmlns:xsd="..."
    xmlns:soap="...">
        <soap:Body>
            <GetBoek xmlns="...">
                    <ISBN>...</ISBN>
            </GetBoek>
        </soap:Body>
</soap:Envelope>
```

SOAP-antwoord

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: ...

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="..." xmlns:xsd="..."
    xmlns:soap="..."> <soap:Body>
          <GetBoekResponse xmlns="...">
            <GetBoekResult>
                  <Isbn>...</Isbn>
                  <Titel>...</Titel>
                  <Prijs>...</Prijs>
             </GetBoekResult>
          </GetBoekResponse>
</soap:Body> </soap:Envelope>
```

Informatie voor proxy-object

- De structuur van de SOAP-berichten
  - Methodes
  - Argumenten
  - Teruggeefwaarden
- Locatie webservices
- Transportprotocol
- WSDL: Webservice Description Language

Structuur WSDL

- Types: Type-declaraties van argumenten, methodes, teruggeefwaardes, …
- Message: Beschrijving bericht (aanvraag/antwoord)
- PortType: Beschrijving mogelijke opdracht (naam, input, output)
- Binding: Verbinden portType met een locatie en een protocol (SOAP, HTTP GET, …)
- Service: Uit welke portTypes bestaat de webservice

Voorbeeld WSDL

- .NET
  - http://localhost/WebCatalogus/BoekenLijst.asmx?WSDL
- J2EE
  - http://localhost:8080/WebCatalogus/catalogusservice?WSDL

Werkwijze

- Maken webservice
  - Klassen, interface, … met eigenlijke functionaliteit maken
  - Webservice maken
  - WSDL genereren
- Clientapplicatie
  - Op basis van WSDL proxy-klassen genereren
  - Clientapplicatie maakt gebruik van proxy-klassen/objecten

Voorbeeld in .NET

Werkwijze in .NET

- Maken webservice
  - Klassen, interface, … met eigenlijke functionaliteit maken (Catalogus.dll)
  - Webservice maken (.asmx-bestand + codebehind)
  - WSDL genereren (automatisch)
- Clientapplicatie
  - Op basis van WSDL proxy-klassen genereren (Wsdl.exe)
  - Clientapplicatie maakt gebruik van proxy-klassen/objecten

Eigenlijke functionaliteit

- Project Catalogus
  - Klasse Boek
    - Resultaat methode
    - Moet geserialiseerd worden
      - Default-constructor
      - set- en get-declaraties voor eigenschappen → anders NIET in XML
  - Klasse BoekenMagazijn
    - Indexer
      - Boek op basis van isbnnummer

Werkwijze in .NET

- Maken webservice
  - Klassen, interface, … met eigenlijke functionaliteit maken (Catalogus.dll)
  - Webservice maken (.asmx-bestand + codebehind)
  - WSDL genereren (automatisch)
- Clientapplicatie
  - Op basis van WSDL proxy-klassen genereren (Wsdl.exe)
  - Clientapplicatie maakt gebruik van proxy-klassen/objecten

Webservice

- Klasse afgeleid van System.Web.Services.WebService
- Klasse en methode
  - Extra attribuut
    - WebService
    - WebMethod
    - Tussen [en] voor klassedefinitie of methodedeclaratie
- WebCatalogus.BoekenLijst.cs

Webservice

- asmx-bestand
- Bestaat uit één richtlijn nl.

```
<%@ WebService Language="c#"
   Codebehind="BoekenLijst.asmx.cs"
   Class="WebCatalogus.BoekenLijst" %>
```

Werkwijze in .NET

- Maken webservice
  - Klassen, interface, … met eigenlijke functionaliteit maken (Catalogus.dll)
  - Webservice maken (.asmx-bestand + codebehind)
  - WSDL genereren (automatisch)
- Clientapplicatie
  - Op basis van WSDL proxy-klassen genereren (Wsdl.exe)
  - Clientapplicatie maakt gebruik van proxy-klassen/objecten

Proxy-klassen genereren

Wsdl /namespace:Be.Hogent.III.ClientWebservice
http://localhost/WebCatalogus/BoekenLijst.asmx

- Wsdl.exe → BoekenLijst.cs
- BoekenLijst.cs
  - BoekenLijst -klasse
    - GetBoek-methode
  - Boek-klasse

Proxy-klassen genereren

- Alternatief: menu “Add Service References”
  - url webservice opgeven
- ClientWebCatalogus\Service References\ServiceWebCatalogus\Reference.cs
- Klassen bekijken in objectbrowser

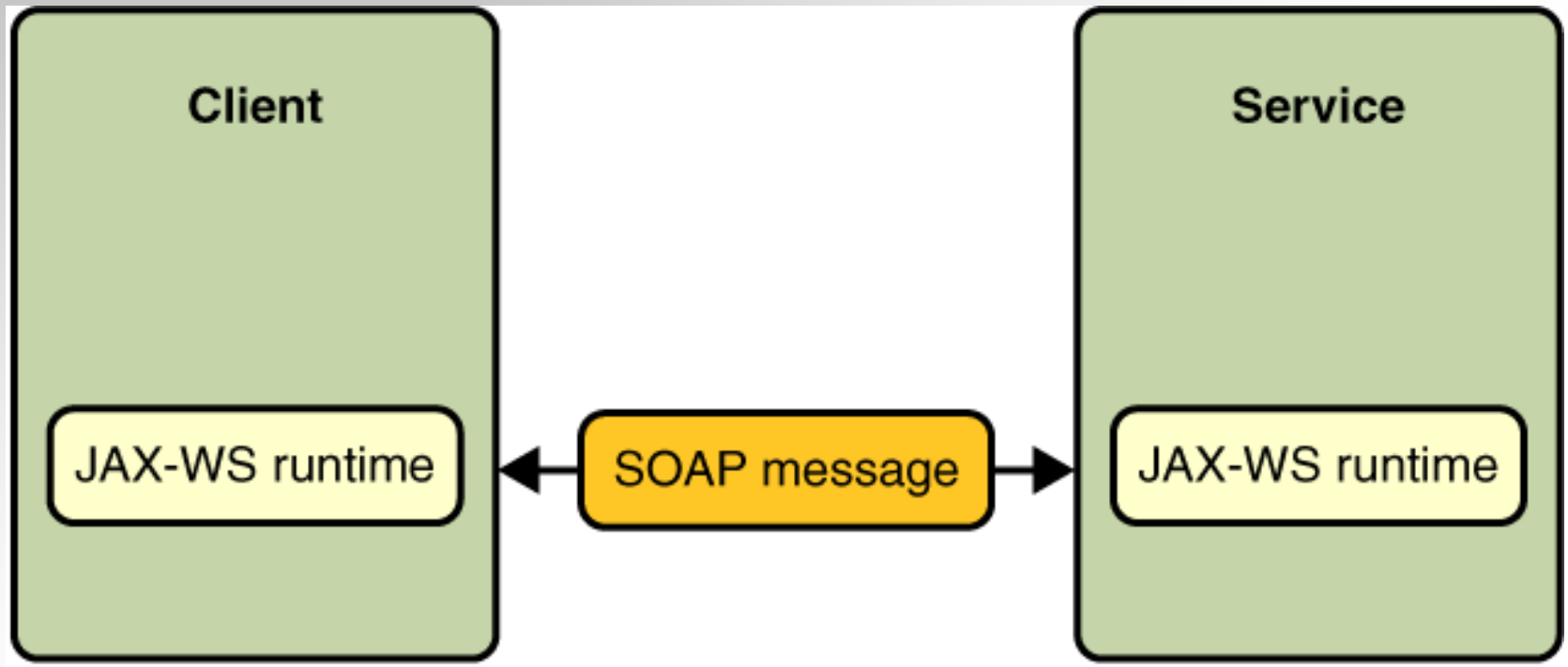
Werkwijze in .NET

- Maken webservice
  - Klassen, interface, … met eigenlijke functionaliteit maken (Catalogus.dll)
  - Webservice maken (.asmx-bestand + codebehind)
  - WSDL genereren (automatisch)
- Clientapplicatie
  - Op basis van WSDL proxy-klassen genereren (Wsdl.exe)
  - Clientapplicatie maakt gebruik van proxy-klassen/objecten

Clientapplicatie

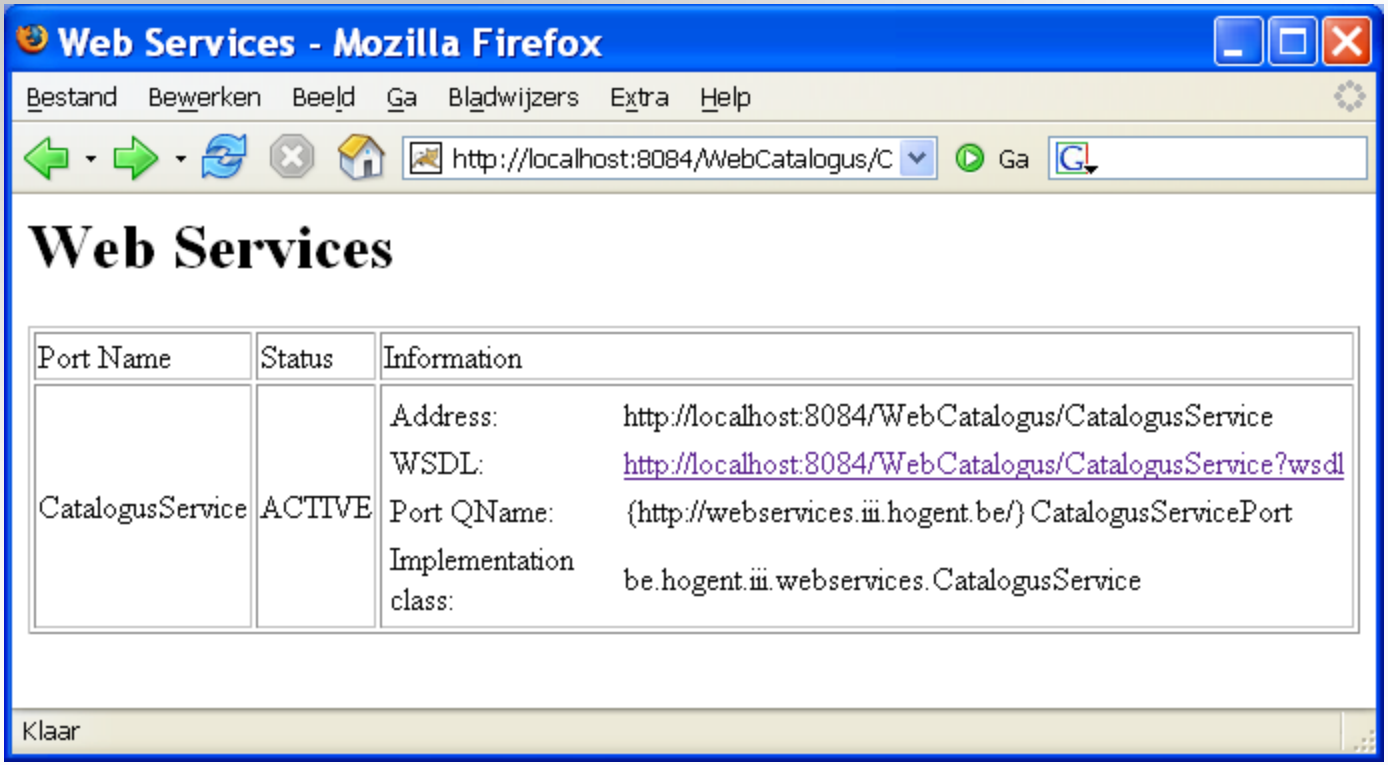
- Boeken met ID/ISBN van 1 tot 5 uitschrijven
- TestGetBoek.cs
- In objectbrowser beschikbare klassen bekijken

Webservices in J2EE



Webservices in J2EE

- Gelijkaardige webservice en clientapplicatie



Werkwijze in J2EE

- Maken webservice
  - Klassen, interface, … met eigenlijke functionaliteit maken (bo en jdbc)
  - Webservice maken
  - WSDL genereren (wsgen.exe)
- Clientapplicatie
  - Op basis van WSDL proxy-klassen genereren (wsimport.exe)
  - Clientapplicatie maakt gebruik van proxy-klassen/objecten

Eigenlijke functionaliteit

- be.hogent.iii.catalogus.Boek
  - Default-constructor
  - get/set-methodes voor id, titel en prijs
- be.hogent.iii.catalogus.BoekenLijst
  - Interface
    - Methode geefBoek
- be.hogent.iii.catalogus.impl.BoekenLijstImpl
  - Implementatie interface BoekenLijst
- In jar catalogus.jar

Werkwijze in J2EE

- Maken webservice
  - Klassen, interface, … met eigenlijke functionaliteit maken (bo en jdbc)
  - Webservice maken
  - WSDL genereren (wsgen.exe)
- Clientapplicatie
  - Op basis van WSDL proxy-klassen genereren (wsimport.exe)
  - Clientapplicatie maakt gebruik van proxy-klassen/objecten

Webservice

- Klasse met annotatie @Webservice()
- Methode met annotatie @WebMethod
  - Parameters met annotatie @WebParam(…)

```
@WebService()
public class CatalogusService {
   @WebMethod
   public Boek geefBoek(@WebParam(name = "isbn") String isbn) {
      try {
         BoekenLijst boeken = new BoekenLijstImpl();
         return boeken.geefBoek(isbn);
      } catch (Exception e) {
         return null;
      }  }  }
```

Werkwijze in J2EE

- Maken webservice
  - Klassen, interface, … met eigenlijke functionaliteit maken (bo en jdbc)
  - Webservice maken
  - WSDL genereren (wsgen.exe)
- Clientapplicatie
  - Op basis van WSDL proxy-klassen genereren (wsimport.exe)
  - Clientapplicatie maakt gebruik van proxy-klassen/objecten

WSDL genereren

- Automatisch bij build project in Netbeans
  - CatalogusServiceService.wsdl
  - CatalogusServiceService_schema1.xsd
    - beschrijft types
  - Wsdl af te halen via http://localhost:8084/WebCatalogus/CatalogusService
  - wsgen: tool om dit handmatig te doen

Webservice deployen

- War-bestand
  - Aantal extra XML-bestanden o.a.
    - sun-jaxws.xml
    - wsdl
    - web.xml bevat gegenereerd info
  - Doel
    - wrappen in een servlet
- Publiceren webservice

Werkwijze in J2EE

- Maken webservice
  - Klassen, interface, … met eigenlijke functionaliteit maken (bo en jdbc)
  - Webservice maken
  - WSDL genereren (wsgen.exe)
- Clientapplicatie
  - Op basis van WSDL proxy-klassen genereren (wsimport.exe)
  - Clientapplicatie maakt gebruik van proxy-klassen/objecten

Genereren proxy-klassen

- Automatisch in Netbeans in een webproject
  - New → Web Service Client
    - WSDL doorgeven
- Klassen terug te vinden in
  - build\generated\
- In Projectsvenster
  - Map Web Service References
    - CatalogusServiceService

Werkwijze in J2EE

- Maken webservice
  - Klassen, interface, … met eigenlijke functionaliteit maken (bo en jdbc)
  - Webservice maken
  - WSDL genereren (wsgen.exe)
- Clientapplicatie
  - Op basis van WSDL proxy-klassen genereren (wsimport.exe)
  - Clientapplicatie maakt gebruik van proxy-klassen/objecten

Clientapplicatie

- Project GebruikWebservice
  - gebruikwebservice.Main.java
- Oproepen methode webservice
  - Rechts klikken in editeervenster
    - Web Service Client Resource
      - Call Web Service Operation
    - Genereert de nodige code om de proxyobjecten te gebruiken

Informatie over webservices

- W3C SOAP Tutorial (http://www.w3.org/TR/2007/REC-soap12-part0-20070427/)
- W3C SOAP 1.2 (http://www.w3.org/TR/soap12-part1/)
- UDDI (http://uddi.org/pubs/uddi_v3.htm)

Informatie over webservices

- XML Web Services Created Using ASP.NET and XML Web Service Clients

(http://msdn2.microsoft.com/en-us/library/7bkzywba.aspx)

- The JavaEE 5 Tutorial (http://java.sun.com/javaee/5/docs/tutorial/doc/)
- Java™ 2 Platform Enterprise Edition, v 5.0 API Specification (http://java.sun.com/javaee/5/docs/api/)
- Webservices in Netbeans (http://www.netbeans.org/kb/trails/web.html)