



AFD Webservice Protocol

**Versie
20160301.1**

INHOUDSOPGAVE

1. INLEIDING.....	4
1.1 DOELSTELLING	4
1.2 DOELGROEP	4
1.3 OPZET	4
2. SPELREGELS AFD WEBSERVICE PROTOCOL.....	5
2.1 INLEIDING	5
2.2 CONCEPT STANDAARD WEBSERVICE	5
2.3 VOORBEELDEN VAN TOEPASSINGEN	7
2.4 HET AANROEPEN VAN EEN AFD WEBSERVICE.....	7
2.5 WSDL WEBSERVICE BESCHRIJVING	8
2.6 ONDERSTEUNING VAN BIJLAGEN.....	14
2.7 TEKENSET	14
2.8 REGISTRATIE AFD WEBSERVICE IN GIM REGISTER	14
3. VOORBEELD AFD WEBSERVICE	16
3.1 INLEIDING	16
3.2 BRONCODE.....	16
3.3 JAVA VS .NET	19

VERSIEBEHEER

Versie	Datum	Auteur	Status
20090701.1	01-07-2009	SIVI	Vervallen
20120401.1	01-042012	SIVI	Vervallen
20160301.1	07-03-2016	SIVI	Definitief

Wijzigingen ten opzichte van de vorige versie

Hoofdstuk/Paragraaf	Aanpassing
Titelblad	Nieuw SIVI logo

1. Inleiding

1.1 Doelstelling

Deze handleiding beschrijft het AFD Webservice Protocol.

1.2 Doelgroep

Doelgroep	Upper Management	Midden / Lijn Management	Consultant, Ontwikkelaar e.d.
Verzekeraars ¹	Nee	Nee	Ja
Intermediairs	Nee	Nee	Ja
Leveranciers	Nee	Nee	Ja

1.3 Opzet

Na deze inleiding volgen in hoofdstuk 2 de spelregels voor AFD Webservices. In hoofdstuk 3 wordt een en ander geïllustreerd met voorbeelden.

¹ Verzekeraars en volmachten. Volmachten worden in het vervolg niet apart genoemd.

2. Spelregels AFD Webservice Protocol

2.1 Inleiding

Het AFD Webservice Protocol is een protocol ten behoeve van het online/realtime aanroepen en uitvoeren van services via standaard internet protocollen in de intermediaire verzekeringsbranche, waarbij gegevens worden uitgewisseld conform het All Finance Datamodel (AFD).

Een service is een software-applicatie met een specifieke functie of een beperkte set aan functies, die aangeboden wordt door een verzekeraar. De service is aan te roepen vanuit een andere applicatie, meestal een intermediairpakket. Als voorbeeld wordt in dit document "het berekenen van de premie voor een polis" gebruikt.

Een standaard webservice is door middel van internet protocollen aan te roepen via het internet en maakt gebruik van de standaarden XML, SOAP en HTTPS. De definitie van de service wordt vastgelegd in een WSDL. Een AFD Webservice is gebaseerd op dezelfde protocollen en standaarden, waarbij als aanvullende afspraak geldt dat alleen gegevens-elementen gebruikt worden die voorkomen in het All Finance Datamodel (AFD). Het AFD Webservice protocol geeft aan hoe op basis van het AFD en AFD berichtstructuren webservices worden aangeroepen en aan welke eisen het WSDL document moet voldoen.

Het protocol is zodanig opgezet dat intermediairs vanuit hun intermediair software de AFD Webservices kunnen aanroepen, over het internet, op de server van een maatschappij.

AFD Webservices worden ingericht door verzekeraars en gebruikt door intermediairs. Het gaat hierbij om bijvoorbeeld het opvragen van tarieven, opvragen van polisgegevens enzovoort.

2.2 Concept standaard webservice

In algemene termen bestaat het concept "webservice" uit 3 delen:

- 1) De service
De software die de gewenste functie uitvoert en die gehost wordt op een over het internet te bereiken server.
- 2) De client
De applicatie die de webservice kan oproepen vanaf de computer van de gebruiker, met internetverbinding.
- 3) De publicatie
De beschikbare definitie van de service en hoe deze aangeroepen en gebruikt kan worden.

Hieronder worden in algemene zin eisen cq aanbevelingen van elk van deze 3 onderdelen beschreven.

1) De service

- De webservice is bij voorkeur 7*24 uur beschikbaar.
- De webservice moet beveiligd worden tegen misbruik.
- De webservice moet actuele en volledige gegevens leveren.
- De gebruikers moeten kunnen vertrouwen op de te leveren gegevens.
- De webservice moet op een gestandaardiseerde manier te benaderen zijn.

2) De client

- De client software dient het gebruik van de webservice eenvoudig te maken voor de gebruiker; het vullen van gegevens ten behoeve van de aanroep van de service gebeurt automatisch; hetzelfde geldt voor het verwerken van de resultaten.

- De client software stelt de gebruiker in staat om te bepalen welke maatschappijen welke functies beschikbaar stellen.

3) De publicatie

- De service dient door middel van een WSDL gedefinieerd te worden.
- De WSDL dient beschikbaar te zijn voor ontwikkelaars van client applicaties.
- Wanneer er binnen een branche afspraken gemaakt worden, kan een webservice opgenomen worden in een voor die branche afgesproken register, bijvoorbeeld het GIM register voor de intermediaire branche.
- Bilaterale koppelingen hoeven niet gepubliceerd te worden.

Algemene regels

Het gebruik van webservices is een techniek om vanuit een client applicatie een service aan te roepen die via de server van de dienstverlener beschikbaar worden gesteld via het internet. Hierbij gelden de volgende regels:

- Webservices werken volgens het request/response model. Een client applicatie stuurt een zogenaamde soap-call naar de service aanbieder en ontvangt direct (dus online/realtime) een antwoord terug; hierbij vindt geen dialoog op het extranet plaats.
- De client kan een complete transactie uitvoeren op voorwaarde dat de gegevens volledig en correct aangeleverd worden. Indien de gegevens niet correct zijn of niet geaccepteerd worden ontvangt de client software een foutmelding die wordt getoond aan de gebruiker.
- Een correcte transactie sluit af met een resultaat dat vervolgens gebruikt wordt in de client software.
- De input/output, de functiebenaming en internetadres van een webservice worden beschreven door middel van een WSDL. Deze gegevens zijn opvraagbaar voordat de webservice wordt aangeroepen.
- Voor het aanroepen van een webservice kan een zogenaamde standaard SOAP-client gebruikt worden.
- Validatie van input gegevens vindt plaats aan de kant van de client ten behoeve van het aanleveren van een correcte en volledige set van gegevens. De service aanbieder controleert de aangeleverde gegevens bij ontvangst en. Wanneer daarbij fouten in de gegevens worden geconstateerd moeten deze gecorrigeerd worden of aangevuld worden aan clientkant voordat de service opnieuw kan worden aanroepen.

AFD Webservice regels

Een AFD Webservice is gebaseerd op de algemene webservice regels zoals hiervoor beschreven en voegt daaraan de volgende regels toe:

- AFD Webservices zijn "AFD-Compliant" webservices, dat wil zeggen webservices die het AFD gebruiken als gegevensstandaard.
- AFD Webservices kunnen vanuit allerlei applicaties worden aangeroepen worden. Dit beperkt zich niet tot intermediairsoftware.
- AFD Webservices kan worden gepubliceerd in het GIM register indien de webservice beschikbaar moet zijn voor de intermediaire branche. Over andere manieren van publiceren zijn geen afspraken gemaakt.
- Wanneer een AFD Webservice aan de intermediaire branche beschikbaar wordt gesteld gelden de door deze branche vastgestelde regels ten aanzien van beveiliging.
- AFD Webservices moeten gecertificeerd zijn door SIVI voor dat zij branchebreed in gebruik genomen worden.
- Bij gebruik van het AFD Webservice Protocol moeten alle AFD-entiteiten en -attributen in dezelfde volgorde door de client-software worden teruggeleverd aan de webservice zoals deze door de webservice (WSDL) is gedefinieerd.

2.3 Voorbeelden van toepassingen

Bij een AFD Webservice moet gedacht worden aan functies die worden ontsloten bij een bepaalde verzekeraar, zonder dat het extranet van die maatschappij door middel van het opstarten van de browser voor de gebruiker in beeld komt.

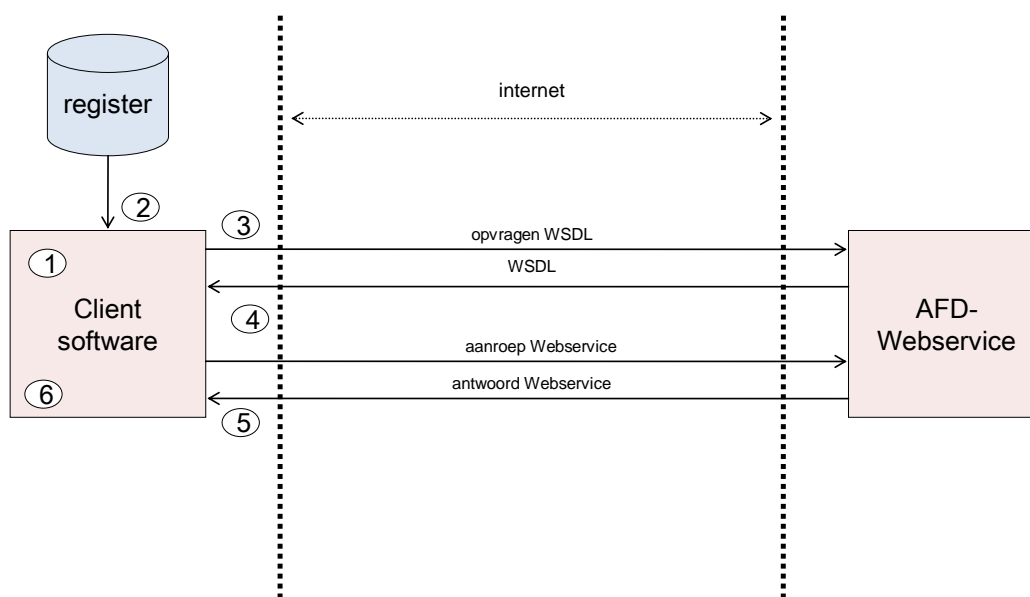
Voorbeelden zijn:

- Ophalen van tarieven in gestructureerd AFD formaat
- Ophalen van een polis in gestructureerd AFD formaat
- Ophalen van de status van een polis in gestructureerd AFD formaat

Zo kan een intermediair webservices gebruiken door vanuit zijn adviespakket de premie voor een levensverzekering op te vragen bij verschillende maatschappijen en de resultaten in 1 document af te drukken voor de klant. Het adviespakket roept daartoe sequentieel of simultaan de juiste AFD-Webservice aan bij verschillende verzekeringsmaatschappijen met dezelfde input en verzamelt de resultaten.

2.4 Het aanroepen van een AFD Webservice

Het aanroepen van een AFD Webservice verloopt als volgt:



Stapsgewijs:

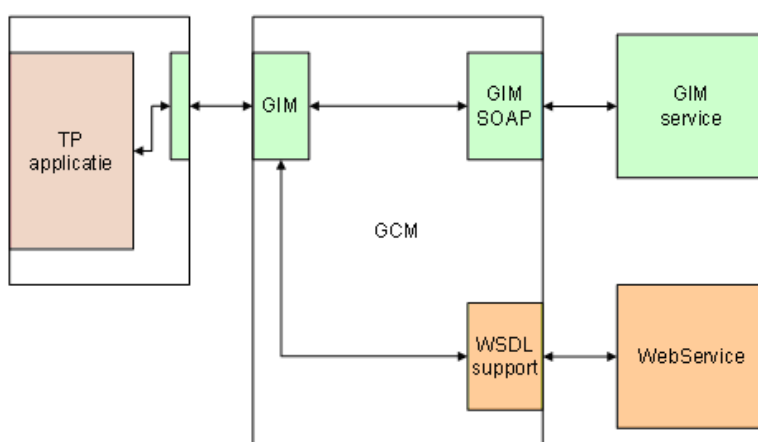
- 1) Vanuit de client applicatie wordt gekozen voor een bepaalde functie. De functie kan een webservice nodig hebben om de functie goed uit te voeren.
- 2) De applicatie zoekt de URL op die hoort bij de webservice. Voor een GIM AFD Webservice is deze URL te vinden het GIM Register.
- 3) De WSDL wordt opgevraagd en gelezen met behulp van de URL uit stap 2.
- 4) Aan de hand van de ontvangen WSDL worden de benodigde gegevens voor de aanroep van de webservice gevuld.
- 5) De webservice wordt aangeroepen en uitgevoerd op de server van de maatschappij en het resultaat wordt verstuurd.
- 6) Het resultaat van de webservice wordt ontvangen en verwerkt door de client software. Dit kan een correct resultaat zijn, maar ook een foutmelding. Ingeval van een fout, moet de fout in de client applicatie hersteld worden en kan de webservice opnieuw worden aangeroepen, waarbij de aangepaste input wordt meegegeven.

Rechtstreeks aanroepen van AFD Webservices

Client applicaties kunnen een AFD Webservice direct via internet protocollen aanroepen. Bij het koppelen hiervan kan gebruik gemaakt worden van de WSDL om delen van de SOAP-client applicatie te genereren en validaties uit te voeren. De AFD Webservice WSDL's zijn opgesteld volgens algemeen toepasbare standaarden (XML, SOAP). In Microsoft.NET, Java en AJAX omgevingen wordt dit genereren en valideren bijvoorbeeld ondersteund.

Via GCM aanroepen van AFD Webservices

Een AFD Webservice is aanroepbaar via de SIVI GIM Communicatie Module (GCM). De AFD Webservices moet dan wel worden gepubliceerd in het GIM register. Vanuit de intermediaire software wordt de webservice dan aangeroepen via de GCM. De werkwijze vanuit de intermediaire software is hierbij voor het intermediair gelijk aan het aanroepen van een GIM transactie.



Beperking bij gebruik van de GCM:

De GIM Communicatie Module ondersteunt alleen gegevenselementen van het xml type "string". Dit betekent dat in de WSDL van de webservice alle elementen van het type "string" moeten zijn.

2.5 WSDL webservice beschrijving

Voor elke AFD Webservice dient een WSDL (Web Services Description Language) beschikbaar te zijn. Input/output, functienaam en URL van een Webservice worden daarin vastgelegd. Iedere webservice heeft dan ook een eigen unieke WSDL.

De webservice dient te voldoen aan de volgende voorwaarden:

- Er moet een WSDL zijn die via het internet (of extranet) te lezen is.
- De WSDL moet een zogenaamde "SOAP" binding in zich hebben. Als standaard kiest SIVI hierbij voor de WSi Basic Profile versie 1.1 (<http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html>)
- In de Header van de SOAP envelop bevindt zich geen functionele informatie. Deze blijft dan ook (voorlopig) leeg.
- Het AFD-Bericht dat als input geldt, moet zich in de Body van de SOAP envelop bevinden. De data definitie moet geheel voldoen aan de AFD afspraken. Deze zijn opgenomen in het Handboek AFD-Berichten.
- Het AFD bericht wordt in de Body omsloten door operation tag, als voorbeeld: <Premieberekening xmlns="..."> ... </Premieberekening >. De naam van deze tag kan door de maatschappij zelf bepaald worden.
- Er moeten Namespaces gebruikt worden in AFD-berichten in de volgende gevallen:

- Namespaces moeten gebruikt worden bij pakketpolissen om ieder onderdeel uniek te maken.
- Namespaces moeten gebruikt worden bij entiteiten in een bericht als een entiteit meerdere keren met dezelfde naam (tag) en met verschillende inhoud kan voorkomen in een bericht.
- Namespaces worden door de verzekeraar bepaald. De naamconventie van W3C dient te worden gehanteerd voor de Namespaces.
- Het AFD bericht mag alleen Namespace(s) bevatten indien de verzekeraar dit nodig acht. Als een verzekeraar geen Namespaces nodig heeft voor de verwerking, moet er ook geen Namespace gebruikt worden.
- Een door het systeemhuispakket gevuld bericht bevat de door de verzekeraar aangeleverde Namespaces.

Omschrijving van de WSDL

Een WSDL staat toe dat er verschillende operations (en zelfs verschillende services) worden opgenomen in één WSDL. De operations hebben ieder hun eigen (wellicht verschillende) input- en outputberichten.

Voor AFD webservices gelden de volgende regels:

- Een WSDL bevat altijd maar maximaal één Service.
- Een WSDL mag meerdere Operations bevatten.
- Een Operation bevat altijd maar één input bericht en één output bericht (Message). Input bericht en Output bericht hoeven niet hetzelfde te zijn. De AFD regels moeten wel gehanteerd worden.
- Alle gedefinieerde berichten moeten binnen de WSDL een unieke naam hebben. Dit mag niet gebeuren op basis van Namespaces, maar moet gebeuren op basis van de naam.
- In de interface naar de tussenpersoonapplicatie hebben alle berichten echter dezelfde naam nodig. (bv Contractdocument).

De opbouw van een SOAP bericht om een webservice aan te roepen ziet er derhalve als volgt uit (voorbeeld):

```
POST /getOne/Service.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.sivi.org/Premieberekening"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Premieberekening xmlns="http://www.sivi.org/">
      <Contractdocument>
        <AL>
          <AL_VRWRKCD>0</AL_VRWRKCD>
          <AL_FUNCTIE>08</AL_FUNCTIE>
          <AL_BRON>AAC</AL_BRON>
        </AL>
        <PP>
          <PP_VRWRKCD>0</PP_VRWRKCD>
          <PP_NUMMER>1234567</PP_NUMMER>
          <PP_INGDAT>20070401</PP_INGDAT>
          <PP_BTP></PP_BTP>
          <VP>
            <VP_VRWRKCD>0</VP_VRWRKCD>
            <VP_ANAAM>Akkerman</VP_ANAAM>
            <VP_GEBDAT>19640213</VP_GEBDAT>
          </VP>
        </PP>
      </Contractdocument>
    </Premieberekening>
  </soap:Body>
</soap:Envelope>
```

De te gebruiken SOAP action wordt in de WSDL van de webservice kenbaar gemaakt en moet dan ook in het aanvraagbericht als zodanig gebruikt worden.

Ondersteunde XML-datatypes

Het AFD Webservice protocol ondersteunt alle XML-datatypes zoals deze door de W3C zijn beschreven, op voorwaarde dat een één-op-één vertaling naar een AFD-datatype mogelijk is.

Dat betekent dat alleen de volgende XML-datatypes toegestaan worden:

- decimal
- gDay
- gYear
- gMonth
- string

Dat betekent ook dat de volgende XML-datatypes NIET toegestaan worden:

- Date, time, gYearMonth, gMonthDay, float, double, Boolean, Notation, Duration, dateTime, hexBinary, qName, base64Binary, anyURI

Onderstaande tabel geeft een nadere specificatie van AFD datatypes in XML representatie:

Waarde	XML-Datatype	Omschrijving
Pn	<xsd:restriction base="xsd:decimal"> <xsd:totalDigits value="x"/> <xsd:fractionDigits value="n"/> </xsd:restriction>	Percentage met maximaal n decimalen b.v. P5 = Percentage met maximaal 5 decimalen
Bn	<xsd:restriction base="xsd:decimal"> <xsd:totalDigits value="x"/> <xsd:fractionDigits value="n"/> </xsd:restriction>	Bedrag met maximaal n decimalen b.v. B0 = Bedrag zonder decimalen
An	<xsd:restriction base="xsd:decimal"> <xsd:totalDigits value="x"/> <xsd:fractionDigits value="n"/> </xsd:restriction>	Aantal met maximaal n decimalen b.v. A1 = Aantal met maximaal 1 decimaal
D1	<xsd:restriction base="xsd:string"> <xsd:length value="8"/> <xsd:pattern value="(((\d{4})(0[13578] 10 12)(0[1-9] 12)[0-9] 3[01])) ((\d{4})(0[469] 11)([0][1-9] 12)[0-9] 30)) ((\d{4})(02)(0[1-9] 1[0-9] 2[0-8])) (((02468)[048]00)(02)(29)) (((13579)[26]00)(02)(29)) (((0-9)[0-9][0][48])(02)(29)) (((0-9)[0-9][2468][048])(02)(29)) (((0-9)[0-9][13579][26])(02)(29)))?"/> </xsd:restriction>	AFD-Datum formaat EEJMMDD Geen xsd:Date hanteren omdat deze scheidingstekens bevat.
D3	<xsd:restriction base="xsd:gYear"> </xsd:restriction>	AFD-Datum formaat EEJJ
D5	<xsd:restriction base="xsd:string"> <xsd:length value="4"/> <xsd:pattern value="(0[1-9] 1[012])(0[1-9] 12)[0-9]3[01]"/> </xsd:restriction>	AFD-Datum formaat MMDD
D6	<xsd:restriction base="xsd:string"> <xsd:length value="6"/> <xsd:pattern value="[0-9]{4}((0[1-9]) (1[0-2]))"/> </xsd:restriction>	AFD-Datum formaat EEJJMM
T1	<xsd:restriction base="xsd:string">	AFD-Tijd formaat UUMM

Waarde	XML-Datatype	Omschrijving
	<pre><xsd:length value="4"/> <xsd:pattern value="([01]\d 2[0123])([0-5]\d)"/> </xsd:restriction></pre>	Geen xsd:Time hanteren omdat deze een scheidingsteken bevat.
JN	<pre><xsd:restriction base="xsd:string"> <xsd:length value="1" fixed="true"/> <xsd:enumeration value="J"/> <xsd:enumeration value="N"/> </xsd:restriction></pre>	Logische waarde J of N Xml-datatype Boolean is iets heel anders en dus verboden.
RK	<pre><xsd:restriction base="xsd:decimal"> <xsd:totalDigits value="10"/> </xsd:restriction></pre>	11-proef voor bankrekeningnummer
--	<pre><xsd:restriction base="xsd:string"> <xsd:maxLength value="x"/> </xsd:restriction></pre>	Tekstveld zonder codelijst of waarde lijst
--	<pre><xsd:restriction base="xsd:string"> <xsd:length value="x" fixed="true"/> <xsd:enumeration value="A"/> <xsd:enumeration value="B"/> <xsd:enumeration value="C"/> </xsd:restriction></pre>	Element met (subset) codelijst, of waardenlijst.

n = maximaal aantal cijfers achter de decimale punt.
x = (maximale) lengte van het hele veld.

Om gebruik te maken van de AFD-datatypes kan binnen de WSDL de onderstaande type definitie gehanteerd worden. Een WSDL met daarin de genoemde definitie kan dan weer gebruikt worden bij het maken van een webservice.

afdDate

```
<xsd:simpleType name="afdDate">
  <xsd:restriction base="xsd:string">
    <xsd:length value="8"/>
    <xsd:pattern value="((\d{4}) (0[13578]|10|12) (0[1-9]|12)
[0-9]|3[01])) | ((\d{4}) (0[469]|11) ([0][1-9]|12)[0-9]|30)) | ((\d{4}) (02)
(0[1-9]|1[0-9]|2[0-8])) | (([02468][048]00) (02) (29)) | (([13579][26]00)
(02) (29)) | (([0-9][0-9][0][48]) (02) (29)) | (([0-9][0-9][2468][048]) (02)
(29)) | (([0-9][0-9][13579][26]) (02) (29))) ?"/>
  </xsd:restriction>
</xsd:simpleType>
```

afdYearMonth

```
<xsd:simpleType name="afdYearMonth">
  <xsd:restriction base="xsd:string">
    <xsd:length value="6"/>
    <xsd:pattern value="[0-9]{4}((0[1-9])|(1[0-2]))"/>
  </xsd:restriction>
</xsd:simpleType>
```

afdMonthDay

```
<xsd:simpleType name="afdMonthDay">
  <xsd:restriction base="xsd:string">
    <xsd:length value="4"/>
    <xsd:pattern value="(0[1-9]|1[012]) (0[1-9]|12)[0-9]|3[01])"/>
  </xsd:restriction>
</xsd:simpleType>
```

afdTime

```
<xsd:simpleType name="afdTime">
  <xsd:restriction base="xsd:string">
    <xsd:length value="4"/>
    <xsd:pattern value="([01]\d|2[0123])([0-5]\d)"/>
  </xsd:restriction>
</xsd:simpleType>
```

In een XML-Schema wordt aan deze type definities gerefereerd als volgt (voorbeeld):

```
<xsd:element name="PP_INGDTW" type="afdDate" minOccurs="0"/>
```

Toestaan van 'leeg-waarden'

Teneinde het mogelijk te maken om lege numerieke waarden te kunnen communiceren, wordt het attribuut 'nillable' toegestaan binnen de definitie van de WSDL.

Voorbeeld definitie in WSDL:

```
<xsd:element name="PP_CDUUMND" minOccurs="0" nillable="true">
  <xsd:simpleType>
    <xsd:restriction base="xsd:decimal">
      <xsd:totalDigits value="3"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

In een bericht dient het weglaten van de waarde als volgt te worden doorgegeven:

```
<PP_CDUUMND xsi:nil="true"/>
```

Output

De output definitie ligt vast in de Web Services Description Language (WSDL).

Hiervoor gelden de volgende voorwaarden:

- In de Header van de SOAP envelop bevindt zich geen functionele informatie, dus blijft (voorlopig) leeg;
- De output van de Webservice kan een correct resultaat zijn maar kan ook een foutmelding zijn.

Resultaat

- Het AFD Bericht dat als output geldt, moet zich in de Body van de SOAP envelop bevinden. De data definitie moet geheel voldoen aan de AFD afspraken. Deze zijn opgenomen in de Handleiding AFD Berichten.
- Het AFD bericht wordt in de Body omsloten door bijvoorbeeld de XML tag <PremieberekeningResponse> ... </PremieberekeningResponse>.
- Het bericht daarbinnen moet van het zelfde type zijn als het AFD bericht dat in de input wordt verstrekt. In het onderstaande voorbeeld Contractdocument.
- In het AFD Bericht zelf wordt aangegeven (XG_STATUS) wat de status van het bericht is. Bijvoorbeeld 8 (geaccepteerd), 0 (afgewezen), 9 (geaccepteerd en afgehandeld) etc. Wanneer een bericht is afgewezen bevindt de reden van afwijzing zich in de status tekst (XG_STATUST).

De opbouw van een resultaat SOAP bericht van een webservice ziet er daardoor als volgt uit (voorbeeld):

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

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <PremieberekeningResponse xmlns="http://www.sivi.org/">
      <Contractdocument>
        <AL>
          <AL_FUNCTIE>08</AL_FUNCTIE>
          <AL_BRON>AAC</AL_BRON>
        </AL>
        <PP>
          <PP_NUMMER>1234567</PP_NUMMER>
          <PP_INGDAT>20070401</PP_INGDAT>
        </PP>
      </PremieberekeningResponse>
    </soap:Body>
  </soap:Envelope>
```

```

        <PP_BTP>100</PP_BTP>
        <VP>
            <VP_ANAAM>Vrolijk</VP_ANAAM>
            <VP_GEBDAT>19640213</VP_GEBDAT>
        </VP>
    </PP>
</Contractdocument>
</PremieberekeningResponse>
</soap:Body>
</soap:Envelope>

```

Foutsituatie

Een webservice geeft fouten door in een zogenaamd “Fault” element.

```

<?xml version="1.0"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultactor/>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Service not available</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Het gaat hier om technische fouten. De webservice heeft haar functie niet kunnen uitvoeren en kan dan ook geen resultaat aanmaken.

Dit is iets anders dan de “status” van de aanvraag. Bijvoorbeeld wanneer de status is “afgewezen” dan wordt in de statutekst de reden van afwijzing vermeld in het bericht en niet in de faultstring. Indien fouten in de input geconstateerd worden, worden deze in het bericht gecommuniceerd.

Strictly typed

De AFD Webservices zijn van het soort “strictly typed”. Hiermee wordt bedoeld dat de volledige datadefinitie te vinden is in de WSDL. De webservice is dus geen “doorgeefluik” van een XML document dat door een achterliggend proces verwerkt wordt.

Ondanks “strictly typed” moeten wel alle gegevens gevalideerd worden door de webservice. Aanpassing in de datadefinitie betekent een nieuwe versie van de webservice die geïnstalleerd moet worden en kenbaar gemaakt moet worden aan de gebruikers.

2.6 Ondersteuning van bijlagen

Om bijlagen in het AFD Webservice protocol te kunnen gebruiken, wordt binnen het AFD-bericht de entiteit Bijlage (<BY>) opgenomen. De entiteit Bijlage (<BY>) bevat minimaal de volgende attributen:

AFD label	Omschrijving	Formaat	Max.lengte
BY_VRWRKCD	Entiteitsverwerkingscode (waarde altijd 0)	string	1
BY_VOLGNUM	Volgnummer	decimal	3
BY_BYLSRT	Soort bijlage, code (codelijst soort bijlage)	string	2
BY_BYLOMS	Bijlage omschrijving	string	70
BY_EXT	Extensienaam van het bestand waaronder de bijlage kan worden opgeslagen. Bijvoorbeeld "PDF", "DOC", etc.	string	70
BY_DATA	De datastring waarin de bijlage is gecodeerd. De codering is CodeBase64	string	onbeperkt

Voorbeeld van een deel van een AFD bericht met daarin een bijlage:

```
<AL>
  <AL_VRWRKCD/>
  .....
  <BY>
    <BY_VRWRKCD>0</BY_VRWRKCD>
    <BY_VOLGNUM>1</BY_VOLGNUM>
    <BY_BYLSRT>11</BY_BYLSRT>
    <BY_XPATH>Contractdocument/PP/VP[1]</BY_XPATH>
    <BY_BYLID>OFF1234567890</BY_BYLID>
    <BY_BYLOMS>Offerte dhr Jansen</BY_BYLOMS>
    <BY_FILENM>OFFJANS</BY_FILENM>
    <BY_EXT>PDF</BY_EXT>
    <BY_DATA>JKDhwqiq8siqw1JS1kakwuwdaswd..... </BY_DATA>
  </BY>
</AL>
<PP>
  <PP_VRWRKCD>
  <PP_NUMMER>
  .....
```

2.7 Tekenset

Voor de SOAP envelop geldt dat de UTF-8 tekenset moet worden gebruikt. Voor het AFD gedeelte in het bijzonder, geldt een beperking op de UTF-8 tekenset. Zie hiervoor: nader in te vullen.

2.8 Registratie AFD Webservice in GIM Register

Als de webservice ter beschikking moet komen aan GIM gebruikers, dan moet de webservice geregistreerd worden in het GIM Register. De eigenaar (OwnerId), de identificatie (ObjectId), de functie (FunctionId), context (ContextId) en productcode (ProductId) moeten hierin opgenomen worden. Onder deze sleutel bevinden zich de URL van de WSDL en de Operation waaronder de webservice te vinden is in de WSDL.

In het GIM Register moet een "GoWsdI" parameter opgenomen worden die het volgende bevat:

- 1) De URL waaronder de webservice WSDL te vinden is. In de WSDL bevindt zich het adres waar de webservice zelf te vinden is.
- 2) De operation die moet worden aangeroepen binnen de WSDL. Op dat moment is ook de input en output beschrijving bekend voor GCM.

OwnerId en ObjectId worden altijd door SIVI uitgegeven.

Verzekeraars moeten hun actuele GIM Registratiebericht naar SIVI sturen zodat SIVI dit op kan nemen in het totale GIM Register en kan publiceren op de website van SIVI. Systeemhuizen worden door SIVI op de hoogte gebracht van de aanwezigheid hiervan en eventuele wijziging hiervan.

Wanneer het gaat om het aansluiten van een service van een "branchevreemde" partij is SIVI het instituut om het registratiebericht daarvoor op de SIVI site te publiceren.

Voor een webservice moeten zowel geefSjabloon als doeFunctie gespecificeerd worden in het GIM Registratiebericht.

In het GIM Register wordt een webservice als volgt geregistreerd (voorbeeld):

```
.....
<Co_Product>
  <Product Id="00001">
    <Branche>021</Branche>
    <Name>Personenautoverzekering</Name>
    <ProductType>Contract</ProductType>
    <Co_Function>
      <Function Id="100.2013">
        <GoWsdL Id="geefSjabloon">
          <URL>https://www.maatschappij.nl/service.asmx?wsdl</URL>
          <Operation>Premieberekening</Operation>
          <Service>Motorrijtuigen</Service>
        </GoWsdL>
        <GoWsdL Id="doeFunctie">
          <URL>https://www.maatschappij.nl/service.asmx?wsdl</URL>
          <Operation>Premieberekening</Operation>
          <Service>Motorrijtuigen</Service>
        </GoWsdL>
      </Function>
      <Function Id="100.2000">
        <GoWsdL Id="geefSjabloon">
          <URL>https://www.maatschappij.nl/service.asmx?wsdl</URL>
          <Operation>Offerte</Operation>
          <Service>Motorrijtuigen</Service>
        </GoWsdL>
        <GoWsdL Id="doeFunctie">
.....
```

3. Voorbeeld AFD Webservice

3.1 Inleiding

Hierna volgt de beschrijving van een eenvoudige webservice die voldoet aan de beschreven voorwaarden. Het gaat hier om een in C# geschreven webservice om een premie te berekenen. De berekening is simpel, de bruto termijn premie (PP_BTP) wordt altijd op 100 gezet. De input wordt direct weer als output teruggegeven en aangevuld met de "berekende" termijnpremie.

3.2 Broncode

```
using System;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;

public struct AL TYPE
{
    public string AL_FUNCTIE;
    public string AL_BRON;
}

public struct VP TYPE
{
    public string VP_ANAAM;
    public string VP_GEBDAT;
}

public struct PP TYPE
{
    public string PP_NUMMER;
    public string PP_INGDAT;
    public string PP_BTP;
    public VP TYPE VP;
}

public struct Contract
{
    public AL TYPE AL;
    public PP TYPE PP;
}

[WebService(Namespace = "http://www.sivi.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]

public class Service : System.Web.Services.WebService
{
    public Service()
    {
        //Uncomment the following line if using designed components
        //InitializeComponent();
    }

    [WebMethod]

    public Contract Premieberekening(Contract Contractdocument)
    {
        // Berekenen van de premie

        Contractdocument.PP.PP_NUMMER = "1234567";
        Contractdocument.PP.PP_INGDAT = "20070401";
        Contractdocument.PP.PP_BTP = "100";

        Contractdocument.PP.VP.VP_ANAAM = "Vrolijk";
        Contractdocument.PP.VP.VP_GEBDAT = "19640213";
    }
}
```



```

    }
    return Contractdocument;
}
}

```

WSDL

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.sivi.org/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://www.sivi.org/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.sivi.org/">
      <s:element name="Premieberekening">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="Contractdocument"
              type="tns:Contract" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="Contract">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="AL" type="tns:AL TYPE" />
          <s:element minOccurs="1" maxOccurs="1" name="PP" type="tns:PP TYPE" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="AL_TYPE">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="AL FUNCTIE" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="AL_BRON" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="PP_TYPE">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="PP NUMMER" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="PP_INGDAT" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="PP_BTP" type="s:string" />
          <s:element minOccurs="1" maxOccurs="1" name="VP" type="tns:VP_TYPE" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="VP_TYPE">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="VP_ANAAM" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="VP_GEBDAT" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="PremieberekeningResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="PremieberekeningResult"
              type="tns:Contract" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="PremieberekeningSoapIn">
    <wsdl:part name="parameters" element="tns:Premieberekening" />
  </wsdl:message>
  <wsdl:message name="PremieberekeningSoapOut">
    <wsdl:part name="parameters" element="tns:PremieberekeningResponse" />
  </wsdl:message>
  <wsdl:portType name="ServiceSoap">
    <wsdl:operation name="Premieberekening">
      <wsdl:input message="tns:PremieberekeningSoapIn" />
      <wsdl:output message="tns:PremieberekeningSoapOut" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ServiceSoap" type="tns:ServiceSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  </wsdl:binding>

```

```

    <wsdl:operation name="Premieberekening">
      <soap:operation soapAction="http://www.sivi.org/Premieberekening"
        style="document" />
      <wsdl:input>
        <soap:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:binding name="ServiceSoap12" type="tns:ServiceSoap">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="Premieberekening">
      <soap12:operation soapAction="http://www.sivi.org/Premieberekening"
        style="document" />
      <wsdl:input>
        <soap12:body use="literal" />
      </wsdl:input>
      <wsdl:output>
        <soap12:body use="literal" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="Service">
    <wsdl:port name="ServiceSoap" binding="tns:ServiceSoap">
      <soap:address location="http://localhost:1199/SimpelServer/Service.asmx" />
    </wsdl:port>
    <wsdl:port name="ServiceSoap12" binding="tns:ServiceSoap12">
      <soap12:address location="http://localhost:1199/SimpelServer/Service.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

AFD bericht in SOAP body

Voorbeeld:

```

<Contractdocument>
  <AL>
    <AL_FUNCTIE></AL_FUNCTIE>
    <AL_BRON></AL_BRON>
  </AL>
  <PP>
    <PP_NUMMER></PP_NUMMER>
    <PP_INGDAT></PP_INGDAT>
    <PP_BTP></PP_BTP>
    <VP>
      <VP_ANAAM></VP_ANAAM>
      <VP_GEBDAT></VP_GEBDAT>
    </VP>
  </PP>
</Contractdocument>

```

Input

Voorbeeld:

```

POST /SimpelServer/Service.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.sivi.org/Premieberekening"

```

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Premieberekening xmlns="http://www.sivi.org/">
      <Contractdocument>
        <AL>
          <AL_FUNCTIE>08</AL_FUNCTIE>
          <AL_BRON>AAC</AL_BRON>
        </AL>
        <PP>
          <PP_NUMMER>1234567</PP_NUMMER>

```

```

        <PP_INGDAT>20070401</PP_INGDAT>
        <PP_BTP></PP_BTP>
        <VP>
            <VP_ANAAM>Vrolijk</VP_ANAAM>
            <VP_GEBDAT>19640213</VP_GEBDAT>
        </VP>
    </PP>
</Contractdocument>
</Premieberekening>
</soap:Body>
</soap:Envelope>

```

Output

Voorbeeld:

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

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <PremieberekeningResponse xmlns="http://www.sivi.org/">
      <Contractdocument>
        <AL>
          <AL_FUNCTIE>08</AL_FUNCTIE>
          <AL_BRON>AAC</AL_BRON>
        </AL>
        <PP>
          <PP_NUMMER>1234567</PP_NUMMER>
          <PP_INGDAT>20070401</PP_INGDAT>
          <PP_BTP>100</PP_BTP>
          <VP>
            <VP_ANAAM>Vrolijk</VP_ANAAM>
            <VP_GEBDAT>19640213</VP_GEBDAT>
          </VP>
        </PP>
      </Contractdocument>
    </PremieberekeningResponse>
  </soap:Body>
</soap:Envelope>

```

Foutmeldingen

Een webservice geeft fouten door in een zogenaamd "Fault" element.

Voorbeeld:

```

<?xml version="1.0"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultactor/>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Service not available</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

3.3 Java vs .Net

In dit hoofdstuk zijn een aantal voorbeelden gegeven op basis van .Net-implementaties. Webservices kunnen echter ook prima gebaseerd worden op Java technologie. Helaas blijkt dat de gehanteerde taal ook consequenties heeft voor de implementatie. Bijgaand twee WSDL's die gebaseerd zijn op dezelfde programmatuur. Een is geschreven in Java en de ander in .Net. Beide WSDL's definiëren een AFD Webservice die uitstekend benaderd kan

worden met behulp van de GCM. En dat terwijl ze beide een andere SOAP-envelop opleveren.

Hierna volgen twee WSDL's die gegenereerd zijn op basis van verschillende ontwikkelomgevingen (Java en .Net) voor een zelfde applicatie.

WSDL op basis van JAVA

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:tns="http://www.sivi.org/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soap12="http://www.w3.org/2003/05/soap-envelope"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc11="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soapenc12="http://www.w3.org/2003/05/soap-encoding"
  xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://www.sivi.org/">
  <wsdl:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      attributeFormDefault="qualified" elementFormDefault="qualified"
      targetNamespace="http://www.sivi.org/">
      <xsd:element name="contractdocument" type="tns:Contractdocument"/>
      <xsd:element name="AFDResult" type="tns:Contractdocument"/>
      <xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema"
        name="Contractdocument">
        <xs:sequence>
          <xs:element xmlns:ns1="http://www.sivi.org/" name="AL"
            type="ns1:AL_TYPE"/>
          <xs:element xmlns:ns2="http://www.sivi.org/" name="PP"
            type="ns2:PP_TYPE"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema"
        name="AL_TYPE">
        <xs:sequence>
          <xs:element minOccurs="0" name="AL_FUNCTIE" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType xmlns:xs="http://www.w3.org/2001/XMLSchema"
        name="PP_TYPE">
        <xs:sequence>
          <xs:element minOccurs="0" name="PP_NUMMER" type="xs:string"/>
          <xs:element minOccurs="0" name="PP_BTP" type="xs:string"/>
          <xs:element minOccurs="0" name="PP_INGDTW" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="AFDRequest">
    <wsdl:part name="contractdocument" element="tns:contractdocument"/>
  </wsdl:message>
  <wsdl:message name="AFDResponse">
    <wsdl:part name="AFDResult" element="tns:AFDResult"/>
  </wsdl:message>
  <wsdl:portType name="ServiceSoap">
    <wsdl:operation name="AFD">
      <wsdl:input name="AFDRequest" message="tns:AFDRequest">
      </wsdl:input>
      <wsdl:output name="AFDResponse" message="tns:AFDResponse">
      </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ServiceHttpBinding" type="tns:ServiceSoap">
    <wsdlsoap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="AFD">
      <wsdlsoap:operation soapAction="http://www.sivi.org/AFD"/>
      <wsdl:input>
        <wsdlsoap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <wsdlsoap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

```

    <wsdl:service name="Service">
      <wsdl:port name="ServiceHttpPort" binding="tns:ServiceHttpBinding">
        <wsdlsoap:address
          location="http://10.0.0.162:8080/Londen/services/Service"/>
        </wsdl:port>
      </wsdl:service>
    </wsdl:definitions>

```

WSDL op basis van .Net

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://www.sivi.org/" xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://www.sivi.org/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
      targetNamespace="http://www.sivi.org/">
      <s:element name="AFD">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1"
              name="Contractdocument"
              type="tns:Contractdocument"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="Contractdocument">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="AL"
            type="tns:AL_TYPE"/>
          <s:element minOccurs="1" maxOccurs="1" name="PP"
            type="tns:PP_TYPE"/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="AL_TYPE">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="AL_FUNCIE"
            type="s:string"/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="PP_TYPE">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="PP_NUMMER"
            type="s:string"/>
          <s:element minOccurs="0" maxOccurs="1" name="PP_BTP"
            type="s:string"/>
          <s:element minOccurs="0" maxOccurs="1" name="PP_INGDTW"
            type="s:string"/>
        </s:sequence>
      </s:complexType>
      <s:element name="AFDResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="AFDResult"
              type="tns:Contractdocument"/>
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="AFDSoapIn">
    <wsdl:part name="parameters" element="tns:AFD"/>
  </wsdl:message>
  <wsdl:message name="AFDSoapOut">
    <wsdl:part name="parameters" element="tns:AFDResponse"/>
  </wsdl:message>
  <wsdl:portType name="ServiceSoap">
    <wsdl:operation name="AFD">
      <wsdl:input message="tns:AFDSoapIn"/>
      <wsdl:output message="tns:AFDSoapOut"/>
    </wsdl:operation>

```

```

</wsdl:portType>
<wsdl:binding name="ServiceSoap" type="tns:ServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="AFD">
    <soap:operation soapAction="http://www.sivi.org/AFD" style="document"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ServiceSoap12" type="tns:ServiceSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="AFD">
    <wsdl:input>
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="Service">
  <wsdl:port name="ServiceSoap" binding="tns:ServiceSoap">
    <soap:address
      location="http://localhost:8001/SimpelServer/Service.asmx"/>
  </wsdl:port>
  <wsdl:port name="ServiceSoap12" binding="tns:ServiceSoap12">
    <soap12:address
      location="http://localhost:8001/SimpelServer/Service.asmx"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Beide WSDL's leveren een andere SOAP-envelop op. Die overigens beide uitstekend door de GIM Communicatie Module worden verwerkt.

SOAP vanuit .Net WSDL

```

<soap:Body>
  <PremieberekeningResponse xmlns="http://www.sivi.org/">
    <Contractdocument>
      <AL>
        <AL_FUNCTIE>08</AL_FUNCTIE>
      </AL>
    </Contractdocument>
  </PremieberekeningResponse>
</soap:Body>
...

```

SOAP vanuit Java WSDL

```

<soap:Body>
  <Contractdocument>
    <AL>
      <AL_FUNCTIE>08</AL_FUNCTIE>
    </AL>
  </Contractdocument>
</soap:Body>
...

```

Beide SOAP-requests wijken af, maar kunnen met de GIM Communicatie Module verwerkt worden.