

SuwiML

Berichtstandaard 3.0

Datum

13 december 2018

Versienummer

0300 v10

Auteur

S. Hadiutomo

T. Zwaan

Opmerking

Definitief

Inhoudsopgave

1. Inleiding	4
1.1. Doel	4
1.2. Afspraken	5
1.3. Leeswijzer	5
2. Positionering	7
2.1. Structuur en relaties	7
2.2. Functionele componenten in een bericht	8
3. SuwiML Berichtstandaard	10
3.1. SuwiML Koppelvlakspecificatie	10
3.2. Basisschema	11
3.3. Koppelvlak in WSDL	12
3.4. SuwiML Namespaces	13
3.4.1 Onderdelen Basisschema	14
3.4.2 Toepassing SuwiML Basisschema binnen SuwiML Berichtschemata	18
3.4.3 Versiebeheer van het Basisschema	19
3.5. Diensten	20
3.5.1 Bodyschema's	21
3.5.2 Specifieke Typedefinitie/domeindefinitie-schema's	22
3.5.3 Include-schema	23
3.5.4 WSDL	24
3.6. FWI	26
3.7. Versiebeheer koppelvlakspecificatie	28
3.8. Identificatie van namespaces	29
4. Afspraken en principes m.b.t. gebruik	30
4.1. Validatie	30
4.2. Geen waarde voor element	30

5. Principes	34
5.1. Bepalen naam van een koppelvlak	34
5.2. Proportionaliteit en doelbinding	34
5.2.1 Bericht-op-maat	34
5.2.2 Granulariteit	36
5.2.3 Principes van granulariteit	36
5.2.4 Genericiteit van berichten	37
5.3. Integriteit van de gegevensuitwisseling	38
6. Referenties	38

1. Inleiding

Deze versie van de Berichtstandaard is in lijn gebracht met de Transactiestandaard versie 4.0.

1.1. Doel

SGR/SuwiML heeft tot doel de (elektronische) gegevensuitwisseling tussen partijen in de Suwi-keten te faciliteren. De interface (machine-machine) tussen partijen is het koppelvlak. Het definiëren van de koppelvlakspecificaties wordt gebaseerd op generieke en standaard afspraken.

De SuwiML Berichtstandaard is de uitwerking op het gebied van koppelvlakken zoals beschreven in KARWel. In de Suwi-keten kennen we naast de SuwiML Berichtstandaard ook het Suwi Gegevensregister (SGR), het SuwiML Basisschema en de SuwiML Transactiestandaard. Het SGR is een Entiteiten – Relaties gegevensmodel dat dient als een gemeenschappelijk gedragen weergave van het gehele Suwi-werkveld. Als zodanig vormt het ook de basis van de ontwikkelde en te ontwikkelen gegevensuitwisselingen en de koppelvlakken van de diensten. Alle entiteiten en attributen zijn voorzien van gezamenlijk afgesproken definities.

Voor de definitie van de inhoud van berichten, de beschrijving van de koppelvlakken en de codering van de uit te wisselen gegevens wordt gebruik gemaakt van SGR/SuwiML, in de vorm van het SGR en extensies hierop voor uitwisseling in XML-formaat. SGR/SuwiML faciliteert het snel en eenduidig definiëren van de berichtinhoud, (middels betekenis van gegevens, te gebruiken gegevenselementen, entiteiten en relaties), de berichtstructuur, de berichtnotatie en de koppelvlakbeschrijving, (middels SuwiML-tags, SuwiML basisschema, SuwiML berichtschema's en WSDL's).

Dit document beschrijft de functionele en technische richtlijnen voor het definiëren van SuwiML berichten en koppelvlakken met betrekking tot de inhoudelijke structuur van de gegevens die moeten worden uitgewisseld. Dit document beschrijft de opzet en het gebruik van het SuwiML basisschema voor het definiëren van de body van SuwiML berichten. Daarmee biedt dit document een beschrijving voor het samenstellen van SuwiML berichtschema's alsook toepassingsonafhankelijke richtlijnen voor het realiseren van gegevensuitwisseling tussen de partijen in de Suwi-keten.

De SuwiML Berichtstandaard is bedoeld voor technisch georiënteerde architecten, informatieanalisten, technisch ontwerpers, software-engineers, programmeurs en

applicatiebeheerders en is zowel gericht op het ontwikkelen van berichten als op de wijze waarop applicaties operationeel met berichten moeten omgaan.

Afspraak 1: Afspraken worden expliciet vastgelegd in genummerde afspraken.

1.2. Afspraken

Hieronder volgt een opsomming van afspraken die afgeleid zijn van specifieke onderwerpen die in dit document zijn beschreven.

Afspraak 1: Afspraken worden expliciet vastgelegd in genummerde afspraken.

Afspraak 2: Koppelvlakspecificaties worden altijd door de beheerder van SGR/SuwiML gecontroleerd voordat deze geïmplementeerd mogen worden.

Afspraak 3: In het Basisschema worden alleen, uit het totale beschikbare scala van DataTypes binnen het Suwi-domein (domeindefinities), die Simple- en ComplexTypes opgenomen, die voor het te specificeren koppelvlak van toepassing zijn.

Afspraak 4: De SuwiML berichten worden hoofdzakelijk met Typed validatie, zoals opgenomen zijn in het SuwiML Basisschema, gevalideerd.

Afspraak 5: Geen enkel van de te gebruiken basis componenten mogen aangepast worden bij het opstellen van een SuwiML Berichtschema.

Afspraak 6: Bij een nieuwe of een wijziging van een domeindefinitie, wordt de bijbehorende domeindefinitie in de Body beschreven of aangeropen. Deze nieuwe of gewijzigde domeindefinitie kan ook in een aparte map onder Diensten worden opgenomen.

Afspraak 7: De structuur en inhoud van een SuwiML bericht wordt gevalideerd aan de hand van het SuwiML Berichtschema dat eraan ten grondslag ligt.

Afspraak 8: Bericht-op-maat wordt altijd samengesteld op basis van 1 bronbericht

Afspraak 9: Een generiek bericht is gebaseerd op de uitvoering van één dezelfde wetgeving of taak.

1.3. Leeswijzer

- Hoofdstuk 2 geeft een algemene introductie op de positionering en een inleidende beschrijving van de SuwiML Berichtstandaard.
- Hoofdstuk 3 geeft een beschrijving van de opbouw en structuur van de SuwiML Koppelvlakspecificatie.
- Hoofdstuk 4 geeft een beschrijving over afspraken en principes met betrekking tot het gebruik
- Hoofdstuk 5 geeft een beschrijving over (algemene) principes.

- Hoofdstuk 6 bevat een overzicht van de gehanteerde referenties.

N.B. Dit document veronderstelt dat de lezer een redelijke kennis van de XML en XML Schema standaard heeft. Voor meer informatie over deze en andere XML gerelateerde standaarden wordt verwezen naar de website van de W3C: <http://www.w3c.org>.

2. Positionering

2.1. Structuur en relaties

Alle betrokken partijen binnen de Suwi-keten wisselen gegevens uit op basis van het SGR. Het SGR is het gegevensmodel met alle voor gemeenschappelijk gegevensgebruik in de Suwi-keten relevante entiteiten, attributen en relaties. Zie het SUWI Gegevensregister deel 1 en deel 2 (<http://www.bkwi.nl>).

SuwiML berichten, zoals deze worden uitgewisseld tussen de betrokken partijen in de Suwi-keten, worden afgeleid van het SGR. Entiteiten en attributen in het SGR keren terug als XML elementen in een SuwiML bericht. Een SuwiML bericht is opgemaakt als een hiërarchische reeks van XML gecodeerde gegevens.

Berichten die behoren tot één berichttype, zijn te verdelen in twee groepen: het initiërende bericht, de Action, en het reagerende bericht, de Reaction. De structuur en inhoud van een SuwiML bericht wordt gevalideerd aan de hand van het SuwiML berichtschemata dat eraan ten grondslag ligt. De functionele inhoud (logische definitie) van het SuwiML koppelvlak wordt vastgelegd in twee SuwiML berichtschemata's (XSD's), bestaande uit een bodyschema voor het Action (BodyAction) en een bodyschema voor het Reaction (BodyReaction).

Een op deze wijze gemodelleerd berichttype is onderdeel van een generieke methode voor het uitwisselen van berichten. Deze methode omvat de technische richtlijnen voor het definiëren en uitwisselen van SuwiML berichten met betrekking tot de envelopestructuur (het transportmiddel) en de wijze van verzending (het transport). De envelopestructuur wordt verplicht gebaseerd op de SOAP standaard. Zie het document SuwiML Transactiestandaard voor een beschrijving van deze generieke methode voor het uitwisselen van berichten.

De SuwiML Berichtstandaard beschrijft de wijze waarop de SuwiML Body moet worden vormgegeven op basis van de bouwstenen vastgelegd in het SuwiML Basisschema. De SuwiML Berichtstandaard maakt deel uit van SGR/SuwiML.

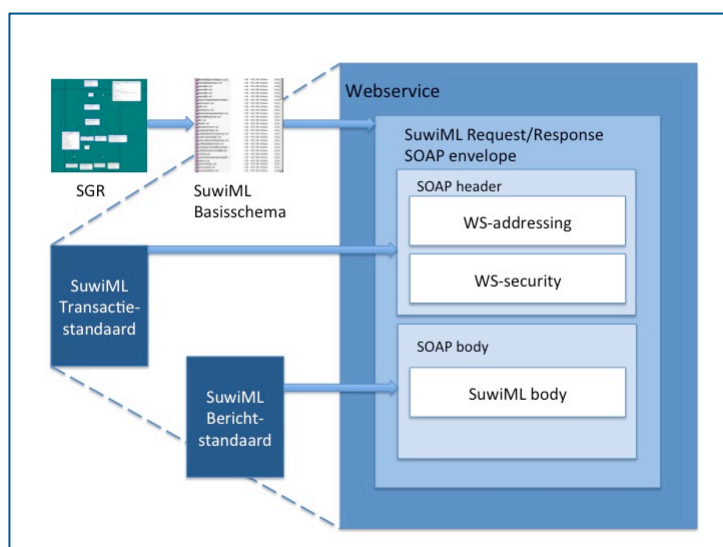
SGR/SuwiML bestaat uit de volgende onderdelen:

- Suwi Gegevensregister
(aangeduid als SGR: het gegevensmodel met alle voor gemeenschappelijk gegevensgebruik in de Suwi-keten relevante entiteiten, attributen en relaties);
- SuwiML Basisschema
(XML Schema's via welke de domeintype informatie voor de benodigde elementen uit het SGR in XML Schema formaat beschikbaar wordt gesteld, bijv. ComplexTypes, SimpleTypes, TypedXsd);
- SuwiML Transactiestandaard
(technische richtlijnen voor het definiëren en uitwisselen van SuwiML berichten met

betrekking tot de envelopestructuur (het transportmiddel) en de wijze van verzending (het transport));

- SuwiML Berichtstandaard (functionele en technische richtlijnen voor het definiëren van SuwiML berichten met betrekking tot de inhoudelijke structuur van de gegevens die moeten worden uitgewisseld).

In Afbeelding 1 worden de relaties tussen de verschillende onderdelen van SGR/SuwiML, noodzakelijk voor de opbouw van een SuwiML bericht, schematisch weergegeven. Het SuwiML Basisschema is het XML synoniem voor het SGR en is van invloed op de onderdelen van een SuwiML bericht. De SuwiML Transactiestandaard schrijft de SOAP envelopestructuur voor, alsmede de stuurgegevens binnen de Header. De SuwiML Berichtstandaard schrijft de opbouw van de SuwiML Body voor.



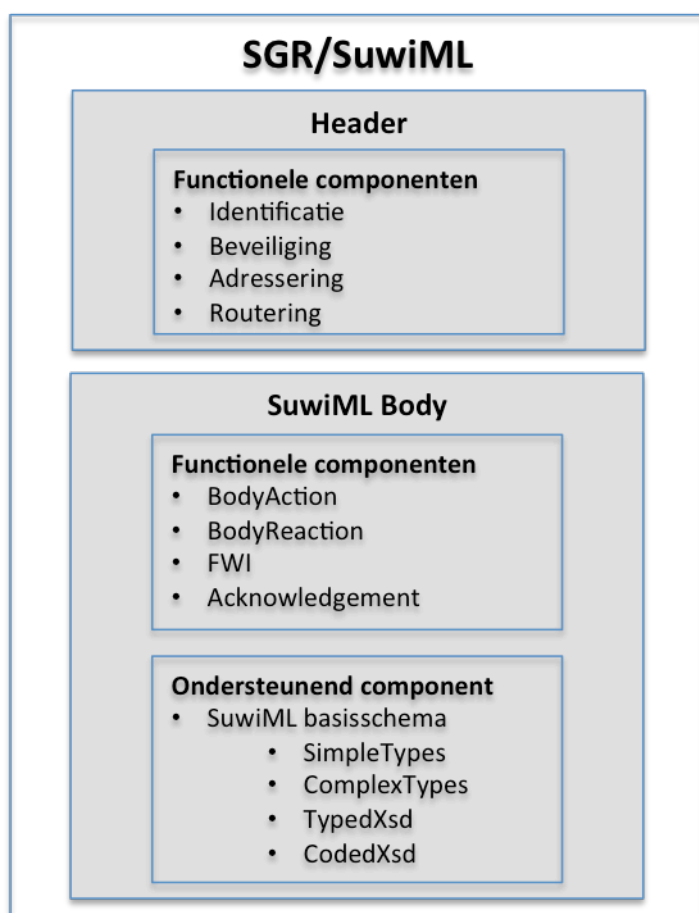
Afbeelding 1 Opbouw SuwiML berichtschema vanuit SGR/SuwiML

Een SuwiML bericht bestaat altijd uit een Envelope die een Header (de stuurgegevens) en een Body (de berichtgegevens) bevat. De opbouw van een bericht door Envelope, Header en Body staat beschreven in de SuwiML Transactiestandaard en is gebaseerd op de SOAP standaard. Dit document, de SuwiML Berichtstandaard, beschrijft de vormgeving van de body (de feitelijke berichtinhoud) van een SuwiML bericht.

2.2. Functionele componenten in een bericht

In afbeelding 2 wordt de samenhang van de binnen SGR/SuwiML gebruikte standaarden en schema's ten opzichte van elkaar weergegeven. De standaarden gezamenlijk beschrijven een generiek SuwiML bericht zoals in afbeelding 1 weergegeven is. De SuwiML

Transactiestandaard maakt gebruik van de standaard SOAP envelopestructuur. De SuwiML Berichtstandaard specificeert de wijze waarop het bodyschema, de SuwiML Body, moet worden vormgegeven met behulp van het SuwiML Basisschema.



Afbeelding 2 Samenhang van standaarden en schema's binnen SGR/SuwiML

De SuwiML koppelvlakspecificatie wordt gedefinieerd door de beheerders van SGR/SuwiML en is geënt op de formele richtlijnen zoals vastgesteld door SGR/SuwiML. De SuwiML koppelvlakspecificatie is een onderdeel van de SGR/SuwiML standaard en vormt de kaders voor de implementatie van de koppelvlakspecificatie. In geval dat de SuwiML koppelvlakspecificatie niet door de beheerders van SGR/SuwiML wordt gedefinieerd moet deze, voordat deze in gebruik genomen kan worden, worden getoetst aan de geldende SGR/SuwiML standaard. Deze toetsing wordt uitgevoerd door de beheerder van SGR/SuwiML, i.e. het BKWI.

Afspraak 2: Koppelvlakspecificaties worden altijd door de beheerder van SGR/SuwiML gecontroleerd voordat deze geïmplementeerd mogen worden.

3. SuwiML Berichtstandaard

3.1. SuwiML Koppelvlakspecificatie

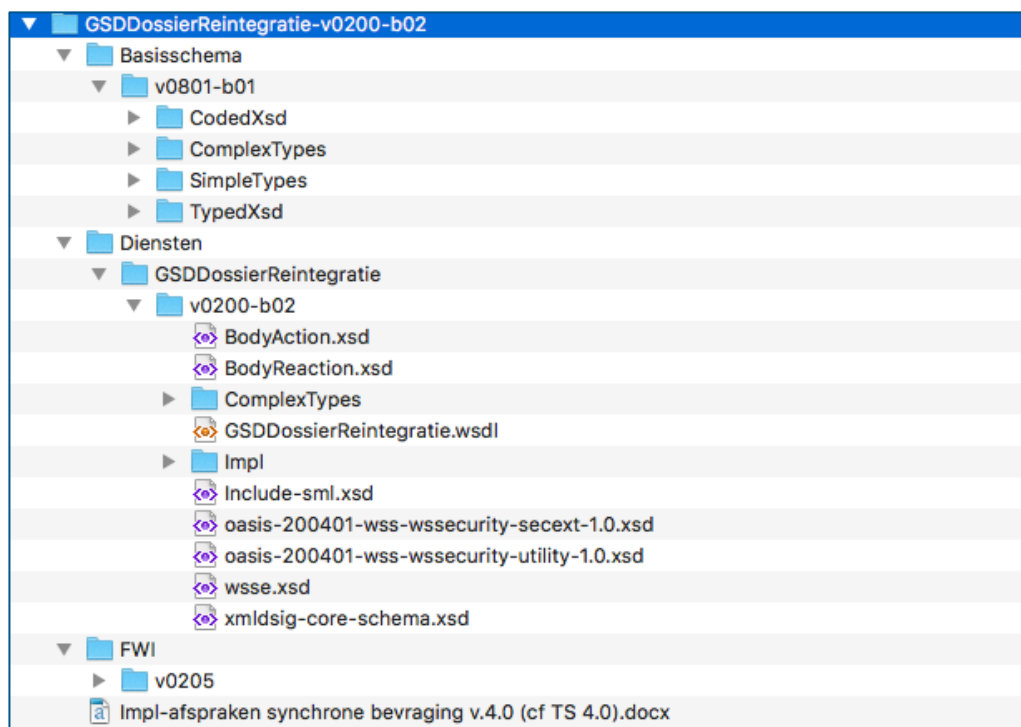
Een SuwiML koppelvlakspecificatie, in de vorm van een zip-bestand, bestaat uit een aantal XML-schema's (XSD's) en de webservice beschrijving (WSDL) die met elkaar verbonden zijn. Een XML-schema beschrijft de afzonderlijke bouwstenen die toegepast worden binnen de webservice. Het gehele pakket aan schema's bevat, naast de generieke standaard te gebruiken schema's en uiteraard, de specifiek gedefinieerde schema's.

Hieronder de generieke onderdelen in een koppelvlak specificatie.

1. Basisschema
2. Diensten: Het specifieke deel (inhoudelijk deel van de service) van de/het koppelvlak/service
3. FWI: Het FWI-schema welke gebruikt wordt om een Fout, Waarschuwing en/of Informatie te kunnen communiceren.

Deze structuur is van toepassing voor alle specificaties van SuwiML koppelvlakken/services/berichten. In de hoofdmap is tevens een implementatiedocument opgenomen m.b.t. tot de generieke werking van het koppelvlak.

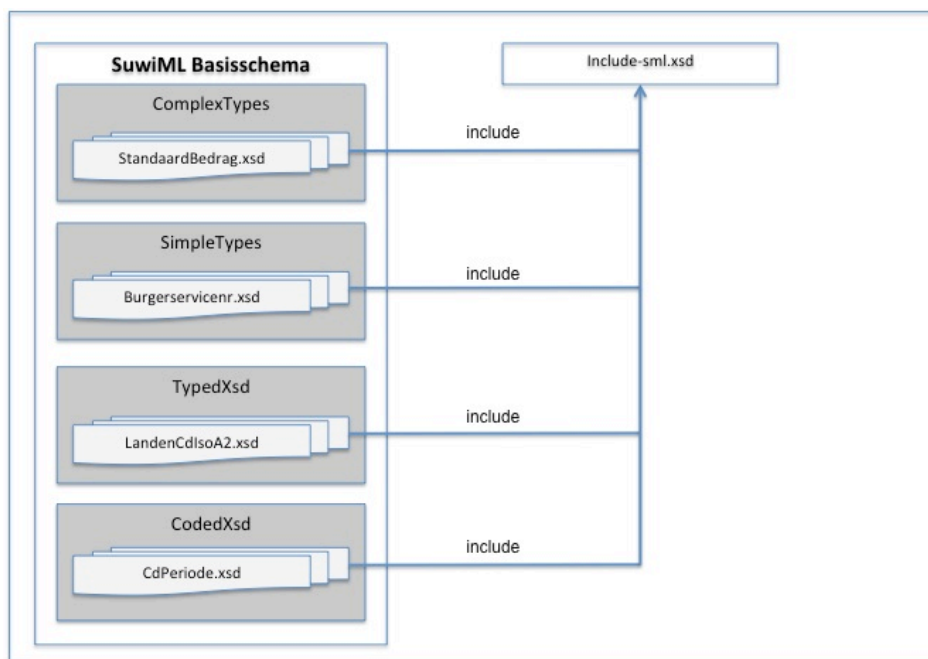
Zie het voorbeeld in Afbeelding 5 Voorbeeld gedeclareerde XSD's in WSDL.



Afbeelding 3 Voorbeeld koppelvlakspecificatie structuur in zip-bestand

3.2. Basisschema

Het Basisschema wordt direct afgeleid uit het SGR en bestaat uit een verzameling 'standaard componenten' op basis waarvan berichtschema's worden gedefinieerd. Het Basisschema bestaat uit de volgende modules: ComplexTypes (SGR-entiteiten), SimpleTypes, CodedXsd, TypedXsd (zie Afbeelding 6). De relaties tussen entiteiten worden niet in het basisschema gedefinieerd, deze worden (met het SGR als uitgangspunt) voor ieder afzonderlijk berichttype vastgelegd in de BodyAction en BodyReaction berichtschema's.



Afbeelding 4 Opbouw en onderlinge verhouding SuwiML Basisschema modules

3.3. Koppelvlak in WSDL

De koppelvlak specificaties van een SuwiML Webservice worden vastgelegd in en bepaald door een WSDL beschrijving (zie afspraak 6 van de SuwiML Transactiestandaard). In die WSDL worden de bijbehorende BodyAction, BodyReaction en FWI in worden gedeclareerd. Afbeelding 1 geeft hier een voorbeeld van.

```

<wsdl:types>
  <xs:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://bkwi.nl/SuwiML/Diensten/Bijstandsregelingen/v0500">
    <xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="xmldsig-core-schema.xsd"/>
    <xs:include schemaLocation="BodyAction.xsd"/>
    <xs:include schemaLocation="BodyReaction.xsd"/>
  </xs:schema>
  <xs:schema targetNamespace="http://bkwi.nl/SuwiML/FWI/v0205">
    <xs:include schemaLocation="../../FWI/v0205/FWI.xsd"/>
  </xs:schema>

```

Afbeelding 5 Voorbeeld gedeclareerde XSD's in WSDL

Het SuwiML Basisschema, de BodyAction, de BodyReaction, de FWI en de WSDL vormen samen de SuwiML koppelvlakspecificatie voor het betreffende berichttype.

3.4. SuwiML Namespaces

SuwiML berichten worden beschreven in de volgende volgorde: een Envelope, een Header en een Body. De berichtschema's die de BodyAction en BodyReaction beschrijven, vormen samen een Dienst of Service. Om de verschillende schema's te onderscheiden wordt gebruik gemaakt van namespaces. Een namespace is een unieke naam die een specifiek definitiegebied aanduidt. Door een schema te definiëren binnen een eigen namespace worden conflicten voorkomen in naamgeving. En kan de eenduidigheid worden gewaarborgd van XML element- en type definities. Bij het vaststellen van een namespace moet rekening worden gehouden met de Bericht-/KoppelvlakNaam, VersieMajor en VersieMinor van het koppelvlak. De namespaces binnen SuwiML worden als volgt opgebouwd:

Tabel 1 Opbouw SuwiML namespaces

Schema	Namespace	Prefix
Basisschema	http://bkwi.nl/SuwiML/Basisschema/v<VersieMajor><VersieMinor> bijv.: http://bkwi.nl/SuwiML/Basisschema/v0801	sml
BodyAction / BodyReaction	http://bkwi.nl/SuwiML/Diensten/<KoppelvlakNaam>/v<VersieMajor> <VersieMinor> bijv.: http://bkwi.nl/SuwiML/Diensten/UWVDossierLoonheffingen/v0100	smls
FWI	http://bkwi.nl/SuwiML/FWI/v<VersieMajor><VersieMinor> bijv.: http://bkwi.nl/SuwiML/FWI/v0205	fwi
Envelope	http://schemas.xmlsoap.org/soap/envelope/	SOAP-ENV

Het gebruik van codelijsten vanuit een schema is vervangen door gebruik te maken van de Codelijsten-service. De namespaces worden nu nog wel uitgeleverd in bestaande koppelvlakspecificaties maar worden niet meer aangeroepen vanuit het berichtschema.

Ieder SuwiML bericht wordt in de vorm van een SOAP envelope verstuurd. Dit betekent dat de Header en de SuwiML Body in de SOAP envelope worden geïntegreerd. De koppeling tussen namespace en URL (locatie) wordt gemaakt met behulp van een XML Schema-instantie constructie, getoond in het XML voorbeeld in Afbeelding 6.

```

2 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
3   <soap:Header>
4     <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
5       xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-w
6     <wsse:BinarySecurityToken [4 lines]
11    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SIG-28f0021b
76    <wsu:Timestamp wsu:Id="TS-63108a2d-be6b-4ce9-8492-69032d3f38bb"> [3 lines]
80  </wsse:Security>
81  <Action xmlns="http://www.w3.org/2005/08/addressing"
82    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurit
83    wsu:Id="id-172c7408-ee48-4808-9fb4-2b989decc0be">
84    http://bkwi.nl/SuwiML/Diensten/VoorbeeldService/Aanvraag
85  </Action>
86  <MessageID xmlns="http://www.w3.org/2005/08/addressing" xmlns:wsu="http://docs.o
87  <From xmlns="http://www.w3.org/2005/08/addressing" xmlns:wsu="http://docs.oasis-
88  <To xmlns="http://www.w3.org/2005/08/addressing" xmlns:wsu="http://docs.oasis-op
89  </soap:Header>
90  <soap:Body>
91  <smls:AanvraagInfo xmlns:smls="http://bkwi.nl/SuwiML/Diensten/VoorbeeldService">
92    <Burgerservicenr>123456789</Burgerservicenr>
93  </smls:AanvraagInfo>
94  </soap:Body>
95 </soap:Envelope>

```

Afbeelding 6 Voorbeeld SuwiML bericht met namespace verwijzingen

3.4.1 Onderdelen Basisschema

Ieder SGR-attribuut heeft een SGR-domeindefinitie. Een SGR-domeindefinitie wordt als een SimpleType gedefinieerd in SuwiML-Basischema. ComplexTypes worden weer samengesteld uit een domeindefinitie welke uit meer dan een SimpleTypes en eventueel ComplexTypes kan bestaan. De Complextypes kunnen in het SGR gedefinieerd worden als standaardstructuren.

De domeindefinities kunnen gedefinieerd worden als een *SimpleType* (een tekenreeks zoals Burgerservicenr) of als een *ComplexType* (een structuur zoals StandaardBedr), al naar gelang de aard van de SGR domeindefinitie die eraan ten grondslag ligt.

Het SGR bevat ook domeindefinities die een waardenbereik beschrijven. De domeindefinities die slechts het formaat bevatten (geen waardenlijst), worden gedefinieerd als *TypedXsd* en de domeindefinities die ook het waardebereik (*enumeration*) bevatten, worden gedefinieerd als *CodedXsd*. Alleen als de waarden in het bericht gevalideerd moeten kunnen worden door het basisschema, wordt een CodedXsd opgenomen.

Het beleid op dit moment is dat de SuwiML berichten hoofdzakelijk op het formaat van de elementen worden gevalideerd. Dit wordt *Typed* validatie genoemd; i.e. de validatie vindt plaats op basis van de TypedXsd.

Daarentegen laat de *Coded* validatie exact en alleen het vooraf gedefinieerde gecodeerde waardebereik toe voor de betreffende domeindefinitie. Vanwege het huidige beleid, bevat de

CodedXsd slechts een aantal domeindefinities (hooguit 5 domeindefinities), die een statisch waardebereik hebben, i.e. de codes wijzigen niet of nauwelijks.

Iedere domeindefinitie waarvoor een codelijst beschikbaar is, wordt vastgelegd in het XML Typed Schema. Met name de codelijsten met een dynamisch karakter (codes die vaak wijzigen), worden opgenomen in de Codelijsten service. Het onderhoud van deze dynamische codelijsten vindt eenvoudiger plaats in de Codelijsten-service. Het eenvoudigweg uploaden van een nieuwe codelijst moet volstaan zonder de bijbehorende programmatuur te moeten herprogrammeren of herconfigureren om de programmatuur weer up-to-date te brengen.

Afspraak 3: *In het Basisschema worden alleen, uit het totale beschikbare scala van DataTypes binnen het Suwi-domein (domeindefinities), die Simple- en ComplexTypes opgenomen, die voor het te specificeren koppelvlak van toepassing zijn.*

Afspraak 4: *De SuwiML berichten worden hoofdzakelijk met Typed validatie, zoals opgenomen zijn in het SuwiML Basisschema, gevalideerd.*

Alle SGR domeindefinities zijn direct of indirect gebaseerd op XML Schema datatypes (built-in types), zoals *string*, *integer*, *decimal*, *date*, etc. al dan niet voorzien van een beperkend *pattern*, *maxLength*, etc. De XML Schema datatypes (built-in types) zijn gedefinieerd in de XML Schema standaard. Deze worden niet apart gedefinieerd binnen de SuwiML schema's.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://
bkwi.nl/SuwiML/Basisschema/v0704">
  <xs:simpleType name="Burgerservicnr">
    <xs:restriction base="xs:string">
      <xs:length value="9"/>
      <xs:pattern value="\d*" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Afbeelding 7 Voorbeeld van een SimpleType

Afbeelding 7 toont een voorbeeld van een SimpleType, waarin de domeindefinitie wordt gedefinieerd welke geen gecodeerd waardebereik hebben. De SGR domeindefinitie *Burgerservicnr* wordt vastgelegd met behulp van een XML *SimpleType* definitie, welke wordt gedefinieerd als built-in datatype *string* met *pattern* '\d*' en *Length* '9'.


```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sml="http://bkwi.nl/SuwiML/Basisschema/v0704"
  targetNamespace="http://bkwi.nl/SuwiML/Basisschema/v0704"
  elementFormDefault="unqualified">
  <include schemaLocation="../CodedXsd/CdPeriode.xsd"/>
  <include schemaLocation="../CodedXsd/CdPositiefNegatief.xsd"/>
  <include schemaLocation="../SimpleTypes/WaardeBedr.xsd"/>
  <include schemaLocation="../TypedXsd/CdMunteenheid.xsd"/>
  <complexType name="StandaardBedr">
    <sequence>
      <element name="CdMunteenheid" type="sml:CdMunteenheid" minOccurs="0"/>
      <element name="CdPositiefNegatief" type="sml:CdPositiefNegatief" minOccurs="0"/>
      <element name="WaardeBedr" type="sml:WaardeBedr" minOccurs="0"/>
      <element name="CdPeriodeEenheidGeldigheidBedr" type="sml:CdPeriode" minOccurs="0"/>
    </sequence>
  </complexType>
</schema>

```

Afbeelding 8 Voorbeeld van een ComplexType

Afbeelding 8 toont een voorbeeld van een ComplexType, waarin een Standaardstructuur-entiteit zoals *StandaardBedr* middels XML ComplexTypes zijn gedefinieerd. Verder is te zien dat bijvoorbeeld het attribuut *CdPeriodeEenheidGeldigheidBedr* gebruik maakt van de domeindefinitie *CdPeriode*.

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://bkwi.nl/SuwiML/Basisschema/v0704">
  <simpleType name="Geslacht">
    <restriction base="string">
      <enumeration value="0">
        <annotation>
          <documentation>onbekend</documentation>
        </annotation>
      </enumeration>
      <enumeration value="1">
        <annotation>
          <documentation>mannelijk</documentation>
        </annotation>
      </enumeration>
      <enumeration value="2">
        <annotation>
          <documentation>vrouwelijk</documentation>
        </annotation>
      </enumeration>
      <enumeration value="9">
        <annotation>
          <documentation>niet gespecificeerd (het gegevenselement wordt niet gevoerd in de administratie)</documentation>
        </annotation>
      </enumeration>
    </restriction>
  </simpleType>
</schema>

```

Afbeelding 9 Voorbeeld van een CodedXsd

Afbeelding 9 toont een voorbeeld van een CodedXsd, waarin de domeindefinitie voor *Geslacht* wordt gedefinieerd met een gecodeerd waardebereik. *Geslacht* wordt vastgelegd met behulp van een XML *SimpleType* definitie, welke wordt gedefinieerd als built-in datatype *string* met daarop een *restriction* van *enumerations* '0', '1', '2' en '9', inclusief de bijbehorende *annotations*.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://bkwi.nl/SuwiML/Basisschema/v0704">
  <simpleType name="Geslacht">
    <restriction base="string">
      <length value="1"/>
      <pattern value="\d*"/>
    </restriction>
  </simpleType>
</schema>
```

Afbeelding 10 Voorbeeld van een TypedXsd

Afbeelding 10 toont een voorbeeld van een TypedXsd, waarin de domeindefinitie voor *Geslacht*, waarbij het gecodeerde waardebereik bewust achterwege is gelaten. Bijna 98% van het gebruik van het SuwiML Basisschema wordt de *Typed* variant gebruikt voor validatie. De *Typed* validatie laat alle mogelijke waarden toe die voldoen aan de XML *SimpleType* definitie voor de betreffende domeindefinitie; hierbij zijn dus ook waarden toegestaan die niet noodzakelijk in het gecodeerde waardebereik vallen. In het voorbeeld hierboven voor *Geslacht*, zou bij gebruik van de *Typed* definitie ook de waarde '6' zijn toegestaan, hoewel deze niet voorkomt in het gecodeerde waardebereik.

Met deze Typed validatie kan er dus niet worden gegarandeerd dat ieder verzonden/ontvangen bericht alléén geldige codewaarden conform het SGR bevat. Hiermee moet de verzendende/ontvangende applicaties zodanig worden geconfigureerd, dat ze altijd aan het meest actuele waardebereik voldoen en dat er in de applicaties zelf wordt gecontroleerd of de uitgewisselde codewaarden aan het vereiste waardebereik is voldaan.

Het voordeel van deze strategie is, dat bij aanpassingen van de aan het bericht gerelateerde codelijsten of waarde bereiken, het berichtschema niet noodzakelijk hoeft te worden aangepast. Hiermee wordt de stabiliteit van de (operationele) elektronische ketenberichten bevorderd, hetgeen voordelen oplevert voor de beheerlast en implementatielast van bericht-/koppelvlakspecificaties. Het actueel houden van de codelijsten binnen de applicaties vindt zoveel als mogelijk automatisch plaats door het inlezen van de nieuwste versie van het SuwiML Basisschema via de Codelijsten-service.

3.4.2 Toepassing SuwiML Basisschema binnen SuwiML Berichtschemata

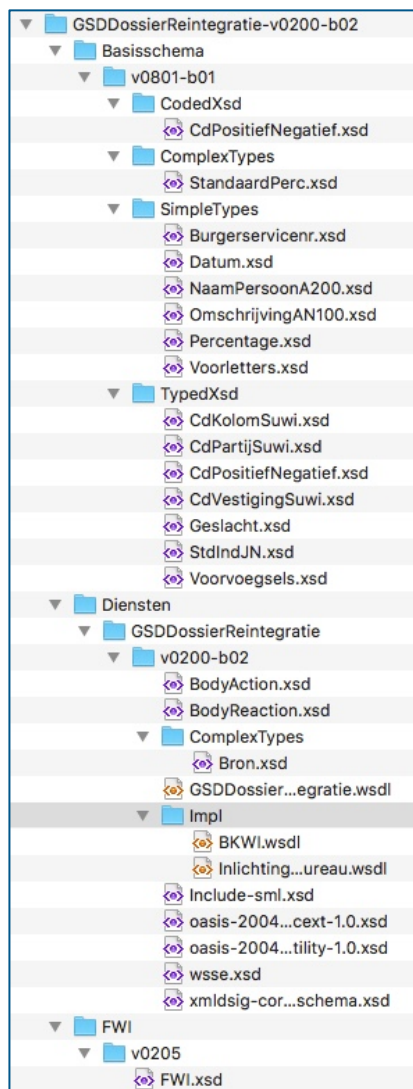
Alle onderdelen van het SuwiML Basisschema worden in één en dezelfde namespace gedefinieerd: [http:// bkwi.nl/SuwiML/Basisschema/v<VersieMajor><VersieMinor>](http://bkwi.nl/SuwiML/Basisschema/v<VersieMajor><VersieMinor>).

Binnen een SuwiML Berichtschemata wordt het SuwiML Basisschema geïntegreerd met behulp van een Include-bestand: Include-sml.xsd. Het Include-bestand incorporeert de voor het betreffende SuwiML Berichtschemata benodigde Basisschema's. Bij het opstellen van een SuwiML Berichtschemata mag het SuwiML Basisschema niet worden aangepast.

Afspraak 5: Geen enkel van de te gebruiken basis componenten mogen aangepast worden bij het opstellen van een SuwiML Berichtschemata.

SuwiML Berichtschemata's kunnen op een vergelijkbare manier als de applicaties gebruik maken van de structuur van het SuwiML Basisschema. Het idee hierbij is dat een bepaalde versie van een SuwiML Berichtschemata gebruik maakt van een bepaalde versie (freeze) van het SuwiML Basisschema. Eventuele aanpassingen in het SuwiML Basisschema worden dus pas in een nieuwe versie van het SuwiML Berichtschemata, wegens structuur-technische aanpassingen, toegepast/gebruikt. Bij functionele aanpassingen in het SuwiML Berichtschemata, kan dezelfde versie van het SuwiML Basisschema nog steeds in gebruik worden gehouden. De nieuwe benodigde typedefinities/domeindefinities worden dan in het Body-schemata zelf beschreven of aangeroepen. De nieuwe of gewijzigde typedefinities/domeindefinities kunnen ook in een aparte map onder Diensten worden opgenomen.

Afspraak 6: Bij een nieuwe of een wijziging van een domeindefinitie, wordt de bijbehorende domeindefinitie in de Body beschreven of aangeroepen. Deze nieuwe of gewijzigde domeindefinitie kan ook in een aparte map onder Diensten worden opgenomen.



Afbeelding 11 Voorbeeld structuur SuwiML Basisschema en een nieuwe domeindefinitie onder Diensten

Afbeelding 11 toont een voorbeeld van de structuur SuwiML Basisschema-v0801-b01 waarin de benodigde type-/domeindefinities voor een bepaald koppelvlak/bericht een onderdeel is van de specificatie. Een eventuele nieuwe type-/domeindefinitie als een *SimpleType* definitie en/of een samengestelde *ComplexType* (Bron) worden geplaatst onder de dienst zelf.

3.4.3 Versiebeheer van het Basisschema

Indien voor een SGR-domeindefinitie een wijziging wordt doorgevoerd, dan heeft dit gevolgen voor de versienummering van het SuwiML Basisschema.

De mogelijke aanleidingen wijzigingen zijn:

- Structurele wijziging in de SGR-entiteiten en –attributen.

- Het pattern van een SGR-domeindefinitie wordt aangepast.
- De lengte, de maximale lengte of de minimale lengte van een SGR-domeindefinitie wordt aangepast.
- Het wijzigen van het voorkomen van bepaalde elementen: de minOccurs en de maxOccurs.

De regels zijn in het kort als volgt.

- Als een SGR-domeindefinitie, -attribuut of -entiteit wordt gewijzigd, dan verandert ook het versienummer (versie major, versie minor en/of buildnummer) van het Basisschema.
- Als er een nieuwe domeindefinitie in het Basisschema wordt toegevoegd, dan wordt de versie major van het Basisschema met één opgehoogd. Alle versie minor en buildnummers worden dan op '01' teruggezet.
- Als de bestaande domeindefinitie in het Basisschema op formaat wordt aangepast, dan wordt de versie minor van het Basisschema met één opgehoogd. Alle buildnummers worden dan op '01' teruggezet;
- Als er een wijziging plaatsvindt op de naamgeving van de bestaande domeindefinitie dan wordt het buildnummer van het Basisschema met één opgehoogd;
- Grote wijzigingen, zoals de opbouw of het herdefiniëren van entiteiten en attributen, leiden tot een wijziging van de versie major. Alle versie minor en buildnummers worden dan op '01' teruggezet.

Codelijsten-service is een aparte service, waar het onderhoud van de codelijsten plaatsvindt.

De mogelijke type wijzigingen op codelijsten zijn:

- Er wordt een nieuwe code toegevoegd aan een SGR-domeindefinitie.
- Het beëindigen van de geldigheid van een code (waarbij de code dus niet uit de codelijst wordt verwijderd).
- Het aanpassen van de omschrijving behorend bij een bestaande code.
- Het aanpassen van de omschrijving, waarvan qua betekenis gelijk blijft aan de oorspronkelijke omschrijving.

3.5. Diensten

Diensten beschrijft wat voor koppelvlak het is. Het koppelvlak wordt gekarakteriseerd door middel van de naam van het koppelvlak, het versienummer (major en minor) en het buildnummer van het koppelvlak. Diensten bestaande uit de Bodyschema's, de specifiek voor de Body gedefinieerde Typedefinitie/domeindefinitie-schema's (ComplexTypes en SimpleTypes), het Include-schema, bestaande uit de schema's uit Basisschema en de WSDL (incl. Implementatie-WSDL).

3.5.1 Bodyschema's

Iedere Body van een SuwiML bericht is gerelateerd aan een SuwiML schemaset. Een SuwiML schemaset bestaat uit een SuwiML-schema voor de Action (BodyAction) en een SuwiML schema voor de Reaction (BodyReaction). Het SuwiML bodyschema beschrijft exact hoe de structuur van een specifiek bericht wordt opgebouwd.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:sml="http://bkwi.nl/SuwiML/Basisschema/v0704"
xmlns:smls="http://bkwi.nl/SuwiML/Diensten/GSDDossierReintegratie/v0100" targetNamespace="http://bkwi.nl/
SuwiML/Diensten/GSDDossierReintegratie/v0100"
elementFormDefault="unqualified"
attributeFormDefault="unqualified">
  <!--Importeer het SuwiML basisschema.-->
  <import namespace="http://bkwi.nl/SuwiML/Basisschema/v0704"
schemaLocation="Include-sml.xsd"/>
  <!--Definities voor de body structuur.-->
  <element name="GSDReintegratielInfo">
    <complexType>
      <sequence>
        <element name="Burgerservicentr" type="sml:Burgerservicentr"/>
      </sequence>
    </complexType>
  </element>
</schema>
```

Afbeelding 12 Voorbeeld Bodyschema

3.5.2 Specifieke Typedefinitie/domeindefinitie-schema's

De specifiek voor Bodyschema's gedefinieerde Typedefinitie/domeindefinitie-schema's worden beschreven in de vorm van ComplexType-schema's en/of SimpleType-schema's. Deze schema's zijn geen onderdeel van het SuwiML Basisschema. Deze specifieke Typedefinitie/domeindefinitie-schema's worden binnen het Bodyschema zelf aangeroepen. Later kunnen deze specifieke schema's genomineerd worden om te worden opgenomen in de nieuwere versie van het Basisschema.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:sml="http://bkwi.nl/SuwiML/Basisschema/v0704" xmlns:smls="http://bkwi.nl/SuwiML/Diensten/GSDDossierReintegratie/v0100"
targetNamespace="http://bkwi.nl/SuwiML/Diensten/GSDDossierReintegratie/v0100"
elementFormDefault="unqualified">
  <!--Importeer het SuwiML basisschema.-->
  <import namespace="http://bkwi.nl/SuwiML/Basisschema/v0704" schemaLocation="../Include-sml.xsd"/>
  <include schemaLocation="../SimpleTypes/NaamAdresNederlandAN80.xsd"/>
  <complexType name="Straatadres">
    <sequence>
      <element name="Locatieoms" type="sml:NaamAdresNederlandAN70" minOccurs="0"/>
      <element name="Postcd" type="sml:Postcd" minOccurs="0"/>
      <element name="Woonplaatsnaam" type="smls:NaamAdresNederlandAN80" minOccurs="0"/>
      <element name="Gemeentenaam" type="sml:NaamAdresNederlandAN40" minOccurs="0"/>
      <element name="Straatnaam" type="sml:NaamAdresNederlandAN24" minOccurs="0"/>
      <element name="Huisnr" type="sml:NummerAdresNederlandN5" minOccurs="0"/>
      <element name="Huisnrtoevoeging" type="sml:NummerAdresNederlandAN6" minOccurs="0"/>
    </sequence>
  </complexType>
</schema>

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://bkwi.nl/SuwiML/Diensten/GSDDossierReintegratie/v0100">
  <simpleType name="NaamAdresNederlandAN80">
    <restriction base="string">
      <maxLength value="80"/>
    </restriction>
  </simpleType>
</schema>
```

Afbeelding 13 Type-definitie/Domein-definitie

3.5.3 Include-schema

In dit Include-schema worden alle benodigde Typedefinitie-schema's uit het gebruikte Basisschema opgenomen die nodig zijn binnen de dienst.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://bkwi.nl/SuwiML/
Basisschema/v0704">
  <!--Importeer de benodigde onderdelen uit het SuwiML basisschema.-->
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/Burgerservicenr.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/Postcd.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/Datum.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/NaamPersoonA200.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/NaamAdresNederlandAN24.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/NaamAdresNederlandAN40.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/NaamAdresNederlandAN70.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/NummerAdresNederlandN5.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/NummerAdresNederlandAN6.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/OmschrijvingAN40.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/OmschrijvingAN100.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/ComplexTypes/StandaardPerc.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/SimpleTypes/Voorletters.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/TypedXsd/CdKolomSuwi.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/TypedXsd/CdPartijSuwi.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/TypedXsd/CdVestigingSuwi.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/CodedXsd/CdPositiefNegatief.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/TypedXsd/StdIndJN.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/TypedXsd/Geslacht.xsd"/>
  <include schemaLocation="../../Basisschema/v0704-b03/TypedXsd/Voorvoegsels.xsd"/>
</schema>
```

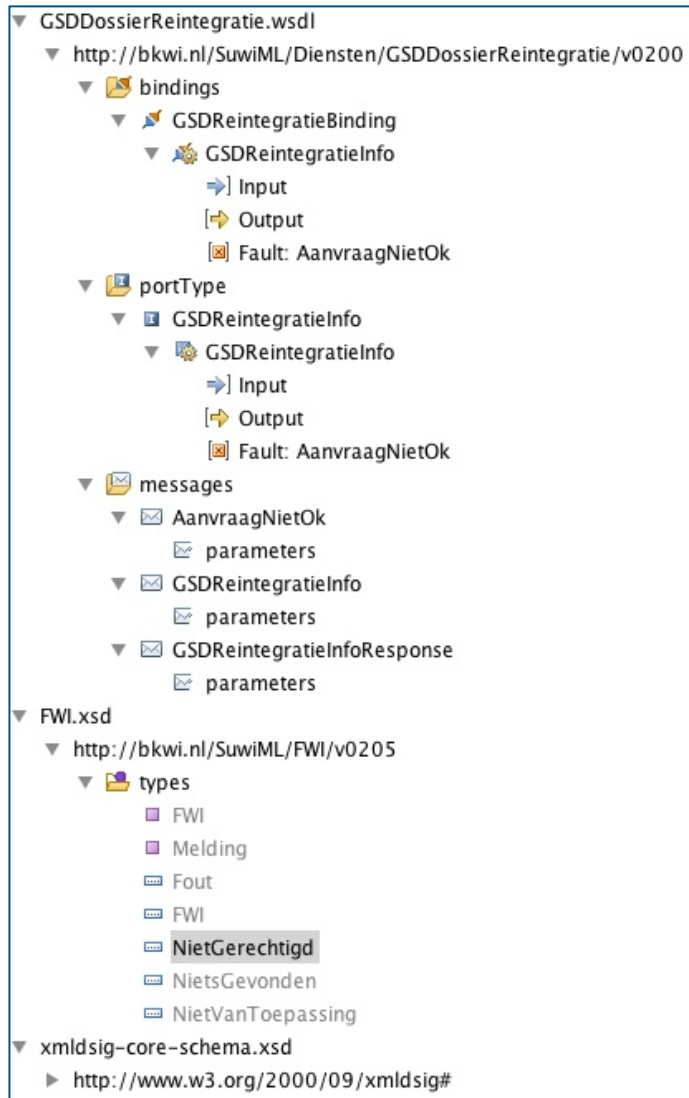
Afbeelding 14 Include-schema

3.5.4 WSDL

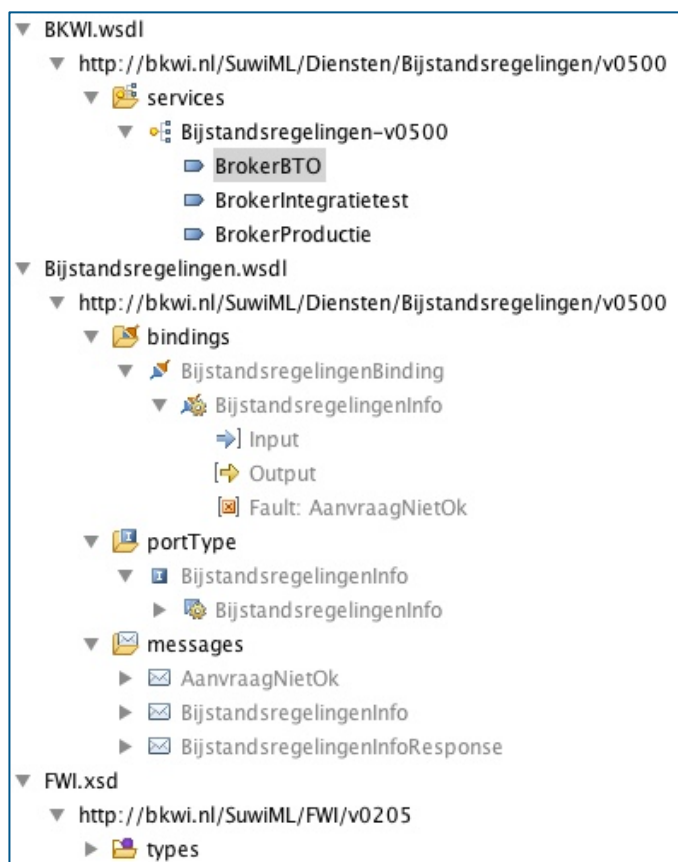
De koppelvlakken van SuwiML-webservices worden beschreven met behulp van Webservices Description Language (WSDL). Dit is mede conform het WS-I Basic Profile 1.1. In de WSDL worden een aantal XML-schema's aangeropen, zoals BodyAction, BodyReaction en FWI. Vervolgens worden de aangeropen operaties beschreven. De structuur van de input- en output- en foutberichten worden vervolgens voor iedere operatie vastgelegd. Naast de WSDL van het koppelvlak wordt er in de map "impl" een implementatie WSDL meegeleverd.

Voor een nadere beschrijving van de WSDL in de SuwiML zie de Transactiestandaard.

Hieronder eerst een afbeelding van de 'basis' WSDL gevolgd door de WSDL uit de map "impl".



Afbeelding 15 Inhoud WSDL



Afbeelding 16 De BKWI WSDL uit de map "impl"

De WSDL uit de implementatie-map importeert de basis WSDL. De WSDL die verspreid wordt in de map IMPL, bevat een algemene URL bij het soap:address van de service. Deze mag door de implementator zelf aangepast worden. Beschrijf zoveel mogelijk een generieke conventie bij de annotaties van de URL's. De naamgeving van de URL moet binnen de context van het koppelvlak passen. Voorbeeld BKWI soap:adress:

<https://suwibroker-integratietest.suwinet.nl/suwibroker/soap/GSDDossierReintegratie-v0100>.

3.6. FWI

Fouten, Waarschuwingen en Informatie kunnen door de antwoordende services gecommuniceerd worden op basis van het FWI-schema. Technische fouten zullen als SOAP fault geleverd worden. Fouten van functionele aard moeten wel op basis van dit schema samengesteld worden.

Zodra er wel een element uit de body geleverd kan worden in het bericht, kan alleen een waarschuwings- of informatie-blok van een FWI opgenomen worden.

Een functionele fout wordt dan gecommuniceerd zonder dat er een ander element uit de body (payload) geleverd wordt. Dit geldt ook voor de mogelijkheden “NietsGevonden”, “NietGerechtigd” en “Niet van Toepassing”. Deze laatste genoemde elementen worden leeg geleverd.

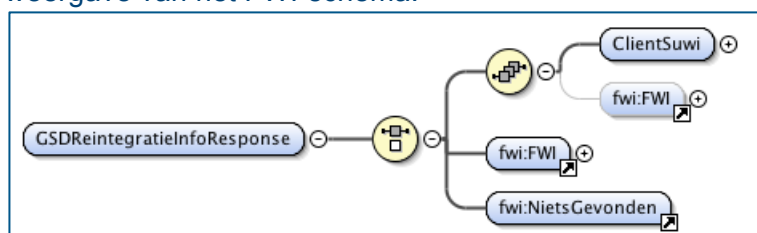
Hieronder volgt een algemene beschrijving in welke situaties een Fout, waarschuwing of informatie geleverd wordt.

Fout	Het betreft de situatie waarbij de achterlichtende dienst de verwerking niet goed heeft kunnen uitvoeren. Denk hierbij bijvoorbeeld aan een Burgerservicenummer dat niet voldoet aan de 11-proef.
Waarschuwing	Waarschuwingen kunnen betrekking hebben op de inhoud van de levering (bijvoorbeeld delen van informatie is niet geleverd) of over de dienstverlening op zich (deze service is tot 31-12-2016 beschikbaar).
Informatie	De levering is goed gegaan, maar de leverancier levert extra informatie over de dienstverlening (Bijvoorbeeld: Morgen om 03.00 uur is de service in onderhoud.)
Niets gevonden	Het opvraag proces is goed gegaan, maar heeft geen resultaat opgeleverd. Bijvoorbeeld de zoek sleutel hebben niet tot een resultaat geleid.
Niet gerechtigd	De optie “NietGerechtigd” wordt toegepast bij het verwerken van meldingen in het kader van beheer op whitelists. De zender van een verzoek mag de opgegeven whitelist niet bewerken.
Niet van toepassing	Deze optie is toegevoegd aan de FWI voor het bericht Whitelist-toets die alleen door Suwinet Inkijk wordt gebruikt. “Niet van toepassing” betekent dat er geen whitelist toegepast hoeft te worden voor de bevraging.

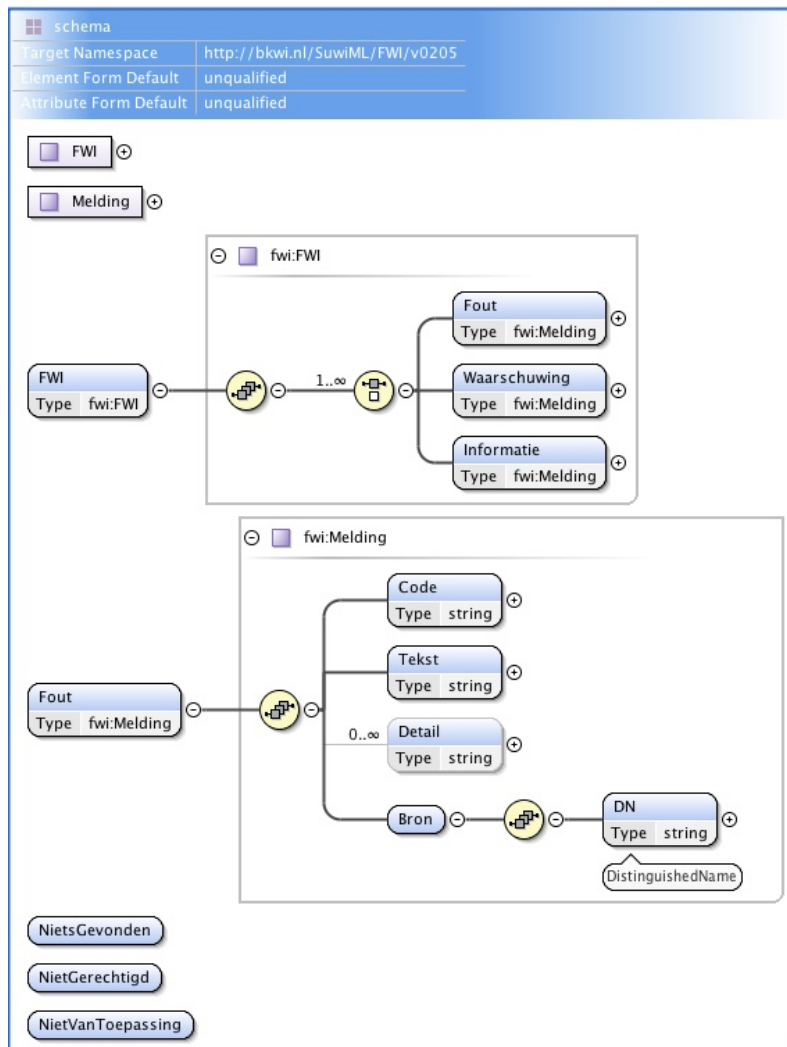
De waarden “Niet gerechtigd” en “Niet van toepassing” worden alleen gebruikt in koppelvakken voor ondersteunende services zoals Whitelisting en Filtering. Dus niet bij koppelvakken waar het gaat om gegevensleveringen van een bron.

In de Transactiestandaard wordt verder invulling gegeven over hoe omgegaan moet worden in bepaalde situaties met betrekking tot foutafhandeling ten aanzien van technische fouten.

Hieronder de grafische weergave van het FWI binnen een Body-response gevolgd door een weergave van het FWI-schema.



Afbeelding 17 Weergave FWI in body-response



Afbeelding 18 weergave FWI-schema

3.7. Versiebeheer koppelvlaakspecificatie

In principe zal een koppelvlaakspecificatie pas worden aangepast als er een functionele//inhoudelijke aanleiding voor is. De kern van de aanpassing betreft het deel Diensten van de koppelvlaakspecificatie. Dan wordt de versie minor met "01" opgehoogd.

Indien noodzakelijk om van de andere bouwstenen, zoals de versie van Basisschema, de versie van FWI ook aan te passen, in verband met wijzigingen, dan wordt de versie major met "01" opgehoogd en de versie minor op "01" teruggezet.

Een uitzondering op het principe dat een koppelvlak-specificatie alléén wordt aangepast als er een functionele aanleiding voor is, betreft de grote/majeur aanpassingen, bijv. het in de toekomst toepassen van technische beveiligingsmaatregelen, zoals signing en/of encryptie. Deze majeure aanpassingen moeten voor alle koppelvlak-specificaties worden doorgevoerd, ondanks inhoudelijk niets gewijzigd is.

Naast het versienummer wordt ook een buildnummer opgenomen in de naam van een koppelvlak. Het buildnummer is echter geen onderdeel van het versienummer. Er kunnen meer dan één versie van een koppelvlak naast elkaar beschikbaar zijn. Maar nooit van één versie meerdere buildnummers.

3.8. Identificatie van namespaces

De naam van het koppelvlak of de service bepaalt de identificatie van de namespaces. Alle functies worden gegroepeerd binnen de context, zodat deze een unieke betekenis en werking hebben. De conflicten moeten worden voorkomen door middel van de eenduidigheid in de naamgeving van de namespace. Een namespace moet uniek zijn en moet de context identificeren. Bovendien vormt de naam van het koppelvlak/de service een onderdeel van de wsaw:Action en het soap-Adres.

4. Afspraken en principes m.b.t. gebruik

4.1. Validatie

Een SuwiML berichtschema is een formele beschrijving van een bericht conform SGR/SuwiML. De zendende partij heeft de verplichting om een, volgens het vastgestelde SuwiML berichtschema, valide bericht te versturen. De ontvanger moet ieder binnenkomend bericht valideren tegen het bijbehorende vastgestelde SuwiML berichtschema alvorens het intern verder te verwerken.

Validatie zorgt er voor dat ieder bericht wordt afgeleverd met informatie die voldoet aan de standaard. Het geeft daarmee garanties dat de het bericht technisch verwerkt kan worden door de afnemer. De afnemer hoeft alleen inhoudelijke controles uit te voeren op de geleverde gegevens. Dit geldt ook voor codes die de afnemer tegen de codelijsten kunnen valideren.

Afspraak 7: De structuur en inhoud van een SuwiML bericht wordt gevalideerd aan de hand van het SuwiML Berichtschema dat eraan ten grondslag ligt.

Volgens Afspraak 7 moeten de SuwiML berichten (automatisch) tegen de gerelateerde SuwiML berichtschema's en/of WSDL worden gevalideerd. Om nu te voorkomen dat de verzender van het bericht de vrijheid heeft te verwijzen naar een willekeurig SuwiML berichtschema op een willekeurige locatie, is het verplicht om bij de implementatie van iedere afzonderlijke gegevensuitwisseling ervoor te zorgen dat tegen een vooraf vastgesteld en op een specifieke locatie geplaatst SuwiML berichtschema en/of WSDL wordt gevalideerd. Dit ongeacht of er een verwijzing is in het 'xsi:SchemaLocation' statement in het bericht. Parsers die inkomende berichten automatisch verwerken moeten de verwijzing in het 'xsi:schemaLocation' statement dus feitelijk negeren, en gebruik maken van het vooraf vastgestelde en op een specifieke locatie geplaatste SuwiML berichtschema en/of WSDL. Zie ook § 5.5 van de Transactiestandaard.

4.2. Geen waarde voor element

Een element in de payload van een bericht kan in sommige gevallen geen waarde hebben of geen waarde gegeven worden. Als een waarde niet geleverd wordt kan dit verschillende redenen hebben. Zo kan de betekenis van het niet hebben van de waarde overlidensdatum bij een persoon betekenen dat deze persoon leeft. Het kan ook voorkomen dat een optioneel element niet geleverd kan worden, omdat een proces nog niet volledig is afgerond. Zodra dit proces wel volledig is afgerond wordt het gegeven alsnog geleverd. Er kan dan geconcludeerd worden dat de informatie die op een bepaald moment geleverd wordt de

actuele stand van zaken weergeeft.

Als de bron het gegeven niet kan leveren, wordt deze niet opgenomen in het berichtschema. Als de afnemer niet geautoriseerd is voor een gegeven, dat wel bij de bron beschikbaar is, wordt dit gegeven niet opgenomen in het berichtschema van de afnemer (zie 5.2.1 Bericht op maat).

Lege velden / waarden in een SuwiML bericht

In het geval de verzender van een SuwiML berichtinstantie voor bepaalde velden in het bericht geen waarde beschikbaar heeft, kunnen zich de volgende situaties voordoen:

SuwiML berichtschema	SuwiML berichtinstantie waarde beschikbaar	SuwiML berichtinstantie geen waarde beschikbaar
<i>veld is optioneel</i>	In het geval een veld optioneel is volgens het SuwiML berichtschema en er is een waarde beschikbaar, dan <u>moet</u> dit veld in het bericht <u>opgenomen</u> worden met een <u>geldige waarde</u> . Het veld <u>mag</u> dan dus <u>niet achterwege</u> <u>blijven</u> . NB. een ongeldige waarde zal vanzelfsprekend tot een foutmelding leiden bij de samenstelling en validatie van het bericht.	In het geval een veld optioneel is volgens het SuwiML berichtschema en er is geen waarde beschikbaar, dan <u>moet</u> dit veld in het bericht <u>achterwege</u> <u>blijven</u> . Het veld <u>mag</u> dan dus <u>niet</u> met een open- en close-tag <u>worden</u> <u>opgenomen</u> <u>zonder feitelijke waarde</u> .

SuwiML berichtschema	SuwiML berichtinstantie waarde beschikbaar	SuwiML berichtinstantie geen waarde beschikbaar
<i>veld is verplicht</i>	<p>In het geval een veld verplicht is volgens het SuwiML berichtschema, dan <u>moet</u> dit veld altijd in het bericht <u>opgenomen worden met een geldige waarde</u>. Het veld <u>mag</u> dan dus <u>in geen geval achterwege blijven</u>.</p> <p>NB. een ongeldige waarde zal vanzelfsprekend tot een foutmelding leiden bij de samenstelling en validatie van het bericht.</p>	<p>In het geval een veld verplicht is volgens het SuwiML berichtschema, dan moet dit veld altijd in het bericht <u>opgenomen worden met een geldige waarde</u>. Het veld <u>mag</u> dan dus <u>in geen geval achterwege blijven</u>.</p> <p>NB. het ontbreken (niet beschikbaar zijn) van een geldige waarde zal vanzelfsprekend tot een foutmelding leiden bij de samenstelling en validatie van het bericht.</p> <p>NB. als voor dit veld in het SGR een waarde is gedefinieerd die aangeeft dat het veld leeg/onbekend is, dan dient bij het ontbreken (niet beschikbaar zijn) van een geldige waarde, deze indicatie "leeg/onbekend" te worden gebruikt in het bericht, bijvoorbeeld bij "Geslacht" de waarde "0" indien "onbekend".</p>

Tabel 2 Regels hoe om te gaan met lege waarden.

In het algemeen geldt dat wat ook in een SuwiML berichtinstantie wordt gestopt, het altijd, ook bij foutsituaties, moet voldoen aan het bijbehorende SuwiML berichtschema. De zender heeft de verplichting dit te garanderen, de ontvanger moet dit controleren.

Kort samengevat komt het er op neer dat als er geen waarde beschikbaar is bij de verzender en het veld optioneel is, het veld weggelaten moet worden. Is het veld verplicht dan moet in het SGR een specificatie zijn opgenomen voor een (fictieve) waarde die "geen waarde" betekent. Deze fictieve waarde wordt dan in het verplichte veld opgenomen. Het omgaan met geen waarde is de verantwoordelijkheid van de informatie modelleur en deze heeft per geval de vrijheid om een passende oplossing te kiezen. Hieronder volgt een voorbeeld van een schema en de uitwerking daarvan in een bericht.


```

<element name="Inkomstenperiode" maxOccurs="unbounded">
  <complexType>
    <sequence>
      <element name="DatBlkp" type="sml:Datum"/>
      <element name="DatElkp" type="sml:Datum" minOccurs="0"/>
      <element name="CdSrtlkv" type="sml:CdSrtInkomstenverhouding" minOccurs="0"/>
      <element name="CdTypeArbeidscontract" type="sml:CdTypeArbeidscontract" minOccurs="0"/>
      <element name="CdInvloedVerzekeringsplicht" type="sml:CdInvloedVerzekeringsplicht" minOccurs="0"/>
      <element name="IndLoonheffingskortingToegepast" type="sml:StdIndJN" minOccurs="0"/>
      <element name="IndRegelmatigArbeidspatroon" type="sml:StdInd" minOccurs="0"/>
      <element name="IndPersoneelsleningNietInLoon" type="sml:StdIndJN" minOccurs="0"/>
      <element name="IndLoonIsMedeAowAlleenstaande" type="sml:StdIndJN" minOccurs="0"/>
      <element name="IndLoonInclusiefWajongUitkering" type="sml:StdIndJN" minOccurs="0"/>
      <element name="IndVakantiebonnenToegepast" type="sml:StdIndJN" minOccurs="0"/>
      <element name="CdAardlkv" type="sml:CdAardlkv" minOccurs="0"/>
      <element name="CdIncidInkomstenvermindering" type="sml:CdIncidInkomstenvermindering" minOccurs="0"/>
    </sequence>
  </complexType>
</element>

```

Afbeelding 19 SuwiML Schema

De ellipsen geven aan waar gegevens die optioneel zijn volgens schema, maar niet geleverd worden. Als er voor een koppelvlak gegevens zijn waarvoor een partij niet geautoriseerd is, worden deze gegevens niet opgenomen in het schema.

```

<Inkomstenperiode>
  <DatBlkp>00011030</DatBlkp>
  <DatElkp>00011030</DatElkp>
  <CdSrtlkv>00</CdSrtlkv>
  <IndLoonheffingskortingToegepast>1</IndLoonheffingskortingToegepast>
  <IndRegelmatigArbeidspatroon>0</IndRegelmatigArbeidspatroon>
  <IndPersoneelsleningNietInLoon>0</IndPersoneelsleningNietInLoon>
  <IndLoonIsMedeAowAlleenstaande>0</IndLoonIsMedeAowAlleenstaande>
  <IndLoonInclusiefWajongUitkering>0</IndLoonInclusiefWajongUitkering>
  <IndVakantiebonnenToegepast>1</IndVakantiebonnenToegepast>
  <CdAardlkv>01</CdAardlkv>
</Inkomstenperiode>

```

Afbeelding 20 Voorbeeld SuwiML bericht

5. Principes

5.1. Bepalen naam van een koppelvlak

De naam van een koppelvlak bestaat uit verschillende onderdelen, die op basis van een aantal regels wordt bepaald. De onderdelen zijn:

- Deel 1 de naam of afkorting van een:
 - Organisatie (bijv. Kadaster, RDW, DUO enz.)
 - Suwipartij (GSD, UWV en SVB)
 - Basisregistratie (bijv. BRP, BROI enz.)
- Deel 2
 - de beschrijving van inhoud bericht (bijv. DossierInkomsten, DossierReintegratie, InfoFilterGeboortedatum enz.)
 - Generieke functionaliteit (bijv. Invordering, Codelijsten, FilterToets, Whitelist enz.)
- Deel 3
 - Versienummer (incl. buildnummer)

Als meer dan 1 partij of organisatie hetzelfde bericht levert wordt deel 1 niet opgenomen in de naamgeving van het koppelvlak. Voorbeeld: Bijstandsregelingen.

Bij een koppelvlak die generieke functionaliteit beschrijft, wordt deel 1 ook niet opgenomen in de naamgeving van het koppelvlak.

De naamgeving van een koppelvlak wordt, als de leverancier een ketenpartij is, altijd afgestemd met de ketenpartij. Of als meer dan één ketenpartij dezelfde gegevens leveren, wordt de naamgeving met die ketenpartijen afgestemd. Zoals bijvoorbeeld GSD en SVB zijn de leveranciers/bronnen van het koppelvlak Bijstandsregelingen.

5.2. Proportionaliteit en doelbinding

'Bericht-op-maat' en *granulariteit* zijn maatregelen om de doelbinding en proportionaliteit te waarborgen.

5.2.1 Bericht-op-maat

Voor diverse toepassingen van inlezen en voorinvullen, zijn specifieke berichten-op-maat gedefinieerd en ontwikkeld. Dit komt omdat de (huidige) dossierberichten meer informatie verschaffen, dan de afzonderlijke afnemer nodig heeft voor de uitvoering van de wettelijke taak (doelbinding). Het principe van 'bericht-op-maat' kan tevens een middel zijn voor het toekennen van de autorisatie/rol van de afnemer, ondanks dat een bericht en de autorisatie totaal twee verschillende componenten binnen de gegevensuitwisseling.

Op dit moment zijn er verschillende oplossingsrichtingen beschikbaar om het principe 'bericht-op-maat' uit te werken, bijvoorbeeld:

- de bronhouder maakt een 'bericht-op-maat' op basis van de autorisatiematrix en stuurt het 'bericht-op-maat' rechtstreeks naar de afnemer via de Suwi-Broker.
 - Voorbeeld: het bericht van RDW dat bedoeld is voor de digitale diensten van de gemeenten.
- Suwi-Broker ontvangt het 'maximaal' bericht van de bronhouder, vult de gegevens in het van tevoren gedefinieerde 'bericht-op-maat' of filtert de gegevens die niet geautoriseerd zijn voor de afnemer en levert het aan de afnemer.
 - Voorbeeld: op basis van het Inkomsten-bericht van UWV worden diverse berichten-op-maat gemaakt voor GSD, voor Gemeentelijke Belastingen, voor SNG, etc.
- Suwi-Broker genereert een 'bericht-op-maat' op basis van het 'maximaal' bericht aan de hand van de autorisatiematrix van de bronhouder.
 - Voorbeelden: berichten van BRP-sets.

Kortom, de inhoud van een 'bericht-op-maat' wordt bepaald door de afstemming over doelbinding en proportionaliteit tussen de bronhouder en de afnemer van de gegevens.

5.2.1.1 Naamgeving van bericht-op-maat

De naam van een bericht-op-maat is opgebouwd door de naam van het bronbericht, waarbij de naam van de Bron vermeld staat, aangevuld met de naam van de afnemer. Tevens is het mogelijk dat een bericht-op-maat voor meer dan één organisatie bestemd is. In dat geval wordt de naam aangevuld met een beschrijving van de gedeelde taak van die organisaties. Aanvullend op een naam wordt bij leveringen t.b.v. uitvoering van wetten, die niets te maken hebben met de SUWI wetgeving, de naam of afkorting van de desbetreffende wet opgenomen.

Voorbeeld: UWVDossierInkomstenGSD ==> UWV is de Bron, GSD is de afnemer en het gaat over DossierInkomsten.

In sommige gevallen wordt de naam van het bericht-op-maat additioneel nog aangevuld met de context/functie van het gebruik.

Voorbeeld: SBRDossierUWVDienstWerkgever ==> SBR is de Bron tevens waar het over gaat, UWV is de afnemer en de gegevens worden gebruikt door UWV t.b.v. de Dienst Werkgever.

Voorbeeld: UWVDossierInkomstenInvordering ==> UWV is de bron en het gaat over DossierInkomsten met de toevoeging van de gedeelde taak van een aantal organisaties.

5.2.2 Granulariteit

Granulariteit houdt in het opdelen van de uit te wisselen informatie in logische delen. De redenen hiervoor zijn o.a. het kunnen hergebruiken van de logische delen voor andere toepassingen en een verbetering van de performance. De granulaire opdeling heeft ook een positief effect op het kunnen ondersteunen van berichten-uitwisseling, die meer recht doet aan de proportionaliteit van de informatievragen.

De aanleiding om granulariteit in berichten toe te passen was in eerste instantie een optimalisatieslag te maken in het berichtengebruik via Suwinet (GeVS).

De bronberichten kunnen worden opgeknipt naar:

- gegevenssoorten, zoals Inkomstenverhouding, Uitkeringsverhouding, Re-integratie, Persoon-gerelateerde gegevens, Adresgegevens, etc.
- de uniforme processen (functionele opdeling)
- de bronsystemen, zoals Polis, WW, WIA/Wajong, ZW, etc.

De huidige voorbeelden:

- UWV heeft een aantal granulaire berichten, zoals Inkomsten-bericht, Werknemersverzekeringen-bericht, Arbeidsverleden-bericht, Re-integratie-bericht.
- GSD heeft een Bijstandsregelingen-bericht en het Re-integratie-bericht.
- SVB heeft een Bijstandsregelingen-bericht en een dossierbericht voor AKW, ANW, AOW en Remigratiewet.

De criteria om berichten te granuleren dienen altijd bepaald te worden van zowel de gegevensleveringskant (leverancier van gegevens) als de gegevensbehoefte-kant (afnemer van gegevens), al dan niet in relatie tot de doelbinding.

De veranderde inhoud van het berichten als gevolg van granulariteit, heeft gevolgen voor welke informatie geleverd en getoond kan worden. Dit zou binnen de huidige werkwijze en ketenstandaarden uitvoerbaar moeten zijn.

5.2.3 Principes van granulariteit

Het in de keten gehanteerde principe is dat een bericht-op-maat slechts wordt samengesteld door één (1) bronbericht. Als gevolg van de granulariteit van een bronbericht is het mogelijk dat er een aantal nieuwe berichten-op-maat gedefinieerd moeten worden.

Voor de afnemers is dit principe bedoeld om de voordelen van meer proportionele leveringen

uit te nutten.

Afspraak 8: Bericht-op-maat wordt altijd samengesteld op basis van 1 bronbericht.

De uitgangspunten zijn:

- Eén bericht-op-maat op basis van één bronbericht.
- De foutafhandeling moet zuiver blijven; d.w.z. dat de afnemer geïnformeerd moet worden op basis van de situatie van iedere service.
- Suwi Broker, namens de bron, levert het bericht-op-maat aan de afnemer(s).
- Bij een wijziging/wegvallen van een bronbericht, blijft het afgesproken bericht-op-maat geleverd, mits het vervangende bronbericht te mappen is op het bericht-op-maat.
- De leverancier van het bronbericht hoeft het bericht-op-maat niet te implementeren en te beheren.
- De afnemer van een bericht-op-maat kan in eigen tempo het bericht implementeren en hoeft minder berichtversies te implementeren, zolang de autorisatie nog geldig is.
- Deze oplossing biedt de meeste flexibiliteit als het gaat om ontkoppeling van bronnen en afnemers en geeft de bron voldoende waarborg dat de afnemer niet meer gegevens ontvangt dan waar de doelbinding voor bestaat.
- Binnen een synchrone berichtuitwisseling (bevragingen) worden de Request en Response altijd afgehandeld binnen 1 HTTP-sessie.

5.2.4 Genericeit van berichten

In het kader van de gegevensuitwisseling van één en dezelfde wetgeving/taak, wordt altijd één generieke berichtdefinitie gemaakt. Ook als meerdere bronnen dezelfde gegevens leveren.

Het bericht bevat een aantal optionele elementen, die conform een bepaalde afspraak, gevuld worden door de bronnen. Het kan zijn dat één bronhouder bepaalde elementen niet vult, omdat ze bepaalde elementen niet registreren. Met als reden dat die elementen voor die bronhouder niet van toepassing zijn. Aan iedere bronhouder wordt een aparte lijst samen met de berichtspecificatie meegegeven, waarin wordt aangegeven welke gegevens door welke bronhouder geleverd moeten worden.

Hiermee wordt bereikt dat interoperabiliteit in de gegevenslevering, waarbij meer dan één bron dezelfde wetgeving/taak-informatie uitvoert, wordt gewaarborgd.

Afspraak 9: Een generiek bericht is gebaseerd op de uitvoering van één en dezelfde wetgeving of taak.

5.3. Integriteit van de gegevensuitwisseling

Om de integriteit van gegevens te waarborgen wordt een bericht voorzien van een handtekening, zoals bij “end-to-end security” wordt toegepast. De ontvanger van een request of response kan dan controleren of het bericht ongewijzigd is. Zie hiervoor de SuwiML Transactiestandaard.

6. Referenties

[iso]

International Organization for Standardization (ISO)
Method for Development of messages – ISO/NP 17113 (Health Informatics)
Information technology - Metadata registries - ISO/IEC 11179
<http://www.iso.org>

[oasis]

Organization for the Advancement of Structured Information Standards (OASIS)
<http://www.oasis-open.org>

[w3c]

World Wide Web Consortium (W3C)
<http://www.w3c.org>

[protocols]

W3C Protocols
<http://www.w3.org/protocols>

[soap]

Simple Object Access Protocol SOAP/1.1, W3C Note 08 May 2000
<http://www.w3.org/TR/2000/NOTE-SOAP-20000508>

[web services]

<https://www.w3.org/2011/07/wspas-pr.html>

[wsdl]

Web Services Description Language 1.1, W3C Note 15 March 2001
<https://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[xml]

Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26 November 2008

<https://www.w3.org/TR/2008/REC-xml-20081126/>

[xml-namespaces]

Namespaces in XML 1.0 (Third Edition), W3C Recommendation 8 December 2009

<https://www.w3.org/TR/2009/REC-xml-names-20091208/>

[xml-schema]

XML Schema Part 0: Primer Second Edition, W3C Recommendation 28 October 2004

<https://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>

XML Schema Part 1: Structures Second Edition, W3C Recommendation 28 October 2004

<https://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>

XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 28 October 2004

<https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

XML Schema Definition Language (XSD): Component Designators, W3C Candidate Recommendation 19 January 2010

<https://www.w3.org/TR/2010/CR-xmlschema-ref-20100119/>

[uml]

Unified Modelling Language (UML)

<http://www.uml.org/>

[umm]

United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) – Modelling Methodology (UMM)

http://www.unece.org/cefact/umm/umm_index.htm