SuwiML

Transactiestandaard 4.0

Datum	Versienummer	Auteur	Opmerking
15-12-2017	20171215-02	S. Hadiutomo T. Zwaan	Definitief

Inhoudsopgave

1. Inleiding

1.1. Doel

De SuwiML Transactiestandaard beschrijft hoe de gestructureerde elektronische informatie-uitwisseling in de Suwi keten is ingericht. Dit document geeft gezamenlijk afgesproken richtlijnen waarmee Suwi partijen eenvoudiger nieuwe gegevensuitwisselingen kunnen opzetten. Er worden allerlei aspecten rondom webservices en berichtenverkeer behandeld. De Transactiestandaard dient gebruikt te worden bij alle projecten waarbij gegevensuitwisseling tussen verschillende partijen plaats gaat vinden. De Transactiestandaard dient ook als naslagwerk in het geval van incidenten en problemen bij bestaande elektronische uitwisselingen. De SuwiML Transactiestandaard is bedoeld voor technisch georiënteerde architecten, technisch ontwerpers, programmeurs en applicatie- beheerders.

Dit document biedt ***toepassing-onafhankelijke*** richtlijnen. De Transactiestandaard richt zich op de ***koppelvlakken*** tussen de applicaties van de verschillende partijen. De Suwi Keten gebruikt een ***contract-first*** benadering: bij een nieuwe informatie-uitwisseling worden de koppelvlak specificaties eerst afgesproken en vastgelegd. Vervolgens zijn de partijen zelf verantwoordelijk voor hoe zij er een applicatie voor optuigen. Het platform en de ontwikkelomgeving voor de applicatie is voor iedere partij vrij te kiezen. Als het uiteindelijke berichtenverkeer tenminste maar aan de afgesproken koppelvlak specificaties voldoet. In de tekst worden wel nog aanwijzingen gegeven voor de wijze waarop de applicaties operationeel met de berichten om dienen te gaan. Waar mogelijk worden in de tekst voorbeelden gegeven.

Afspraak 1: Afspraken worden expliciet vastgelegd in genummerde afspraken.

1.2. Afspraken

1.3. Verschillen met 3.1

Met deze nieuwe versie van de Transactiestandaard is de voornaamste verandering het verdwijnen van de SuwiML-Header bij de synchrone berichten. De aanleiding voor deze aanpassing zijn de wensen om het gegevensverkeer betrouwbaarder te maken en te laten voldoen aan de landelijke standaard Digikoppeling. Om webservices te laten voldoen aan de landelijke standaard Digikoppeling is het noodzakelijk om de Suwi-specifieke header (SuwiML-Header) te laten vervallen. Daarnaast zal ten behoeve van integriteit en betrouwbaarheid van de gegevens-uitwisselingen het Digikoppeling-profiel 2W-be-S worden toegepast. Daarmee wordt invulling gegeven aan de wens om end-to-end secure gegevens uit te wisselen.

Doordat de SuwiML-Header komt te vervallen zullen de functies die aan de Header toegekend waren op een andere manier geïmplementeerd worden. Het betreft hier de functies identificatie/authenticatie, autorisatie routering en verantwoording (logging).

Via Suwinet worden miljoenen berichten met privacygevoelige gegevens uitgewisseld. Overheid, burgers, bronhouders en afnemers rekenen er op dat dit betrouwbaar en veilig gebeurt. Dit houdt in dat die berichten op de juiste plek aankomen, afkomstig zijn van een geautoriseerde afnemer, de identiteit van de afnemer beter kan worden vastgesteld, onderweg tijdens de transport niet afgeluisterd of gelezen of aangepast worden door onbevoegdheden, etc. Dit wordt “end-to-end security” genoemd. In het kort betreft het de maatregelen door het toepassen van de security-elementen (WS-Security 1.1) in de Header van de berichten, zoals beschreven is in het Digikoppeling-profiel 2W-be-S. Een bijkomstig voordeel is dat de keten W&I hiermee ook voldoet aan de landelijke standaard Digikoppeling. Dit heeft een positief effect op de interoperabiliteit.

Deze Transactiestandaard bevat 3 aspecten van beveiliging:

1. Beveiliging op transportniveau (netwerk) op basis van TLS. Dit zorgt ervoor dat de gegevens niet onderweg gelezen kunnen worden door onbevoegden.
2. Elektronische ondertekening van een bericht. Met deze handtekening kan door de ontvanger de identiteit van de zender vastgesteld wordt. Daarnaast wordt door de ondertekening het mogelijk om te controleren of de inhoud van het bericht ongewijzigd aangekomen is.
3. Encryptie van de gegevens. Als ook getransporteerde gegevens beschermd moeten worden, kan er voor gekozen worden om de gegevens te versleutelen. Of de gegevens, die getransporteerd worden, versleuteld moeten worden is afhankelijk van de mate vertrouwelijkheid of gevoeligheid. Het besluit of gegevens versleuteld moeten worden, wordt door de bron van de gegevens bepaald. Op dit moment is dit nog niet noodzakelijk.

Om end-to-end security te kunnen doorvoeren moeten alle op Suwinet aangesloten bronhouders en afnemers hun webservices worden aangepast, zoals het gebruiken van PKI-Overheid certificaten t.b.v. TLS-verbinding, elektronische ondertekening van berichten en eventueel door versleuteling inhoud van berichten. De koppelvlakken zullen worden voorzien van de security-elementen. Tevens zal het aanpassingen vereisen op onderdelen die de authenticatie en autorisatie en de logging ondersteunen.

Voor het autoriseren van organisatie of onderdelen van organisatie zal gebruik gemaakt worden van het OIN en sub-OIN’s, welke in WS-Addressing element ‘wsa:From’ opgenomen zal worden. Door het gebruik te maken van sub-OIN wordt fijnmaziger autorisatie mogelijk, bijv. op het niveau van organisatieonderdeel, afdeling, applicatie, wettelijke grondslag, etc. Dit houdt in dat pér sub-OIN een aparte PKI-Overheid certificaat wordt toegepast.

Omdat er nog geen duidelijkheid is over welk protocol toegepast zal worden voor asynchrone berichten en de gegarandeerde levering (Reliability), blijft voor asynchrone gegevensuitwisseling Transactiestandaard 3.1 van toepassing. Dat is de reden waarom in deze versie alleen synchrone bevragingen worden beschreven.

1.4. Doorvoeren van een nieuwe versie

Deze nieuwe versie van de Transactiestandaard staat in het kader van het verhogen van het beveiligingsniveau van de gegevensuitwisseling. Om het gewenste beveiligingsniveau te behalen zoals beschreven in de nieuwe Transactiestandaard is het aan te raden om alle koppelvlakken binnen een bepaalde periode aan te passen. En niet zoals gebruikelijk alleen een koppelvlak aan te passen als er een functionele wijziging doorgevoerd moet worden. Het is dan namelijk niet aan te geven wanneer het vereiste beveiligingsniveau bereikt is. Immers de zwakste schakel in een keten bepaalt de sterkte.

1.5. Historie

In het regeerakkoord van 1998 werd de uitvoeringsstructuur van de publieke arbeidsvoorziening en de sociale zekerheid ingrijpend gewijzigd. De nieuwe structuur kreeg vorm in Kabinetsbesluiten in 1999 en 2000. Onder andere werd daar besloten tot de oprichting van het CWI. De automatisering van de nieuwe structuur werd onder andere vormgegeven door centrale componenten als het Suwi Gegevensregister (SGR) en het bijbehorende XML vocabulaire SuwiML. Om gestructureerde informatie-uitwisseling met behulp van SuwiML Berichten van de grond te krijgen werd er in 2001 al een eerste versie van de SuwiML Transactiestandaard geschreven. Ook in 2001 kreeg CWI de opdracht om de afzonderlijke eenheid Bureau Keteninformatisering Werk en Inkomen (BKWI) op te richten. Het BKWI kreeg taken rond ontwikkeling, beheer en onderhoud van gemeenschappelijke afspraken en voorzieningen, bijvoorbeeld ook van het SGR, SuwiML en de Transactiestandaard.

In 2003 ging de Suwinet Inkijk applicatie een belangrijke rol spelen bij het ontsluiten van grote hoeveelheden gegevens uit de bron-systemen van UWV, CWI (heden is CWI een onderdeel van UWV) en Gemeentelijke Sociale Diensten. De techniek van het ontsluiten gebeurde conform de Transactiestandaard. In 2004 en 2005 werden er Elektronische Keten Berichten (EKB's) gemaakt waarmee dossiers overgeheveld werden van de éne naar de andere Suwi Partij. De inzichten die voortkwamen uit die trajecten vonden hun weerslag in versie 2.0 van de Transactiestandaard.

Versie 3.0 is enerzijds geïnspireerd door het Digitaal Klant Dossier (DKD) programma dat in 2007 in de Suwi Keten heeft gelopen. Anderzijds hebben er internationaal ontwikkelingen plaatsgevonden op het gebied van standaardisatie van elektronische informatie-uitwisseling. Met deze nieuwe versie sluit de Transactiestandaard ook weer beter bij die ontwikkelingen aan.

Versie 3.1 bevat wijzigingen met betrekking tot het meer conformeren aan Digikoppeling en het niet meer leveren van Envelope-schema’s bij de koppelvlak-specificaties.

Versie 4.0 kent zijn oorsprong in de onderzoeken die gedaan zijn in 2010 m.b.t te veiligheid van gegevensuitwisseling binnen het Suwidomein. Daarbij is ook gekeken welke eisen er gelden vanuit het Normenkader die te maken hebben met uitwisseling van gegevens.
Tevens is de behoefte geuit in in KArWel om aansluiting te zoeken bij de landelijke standaarden.

1.5.1 Document historie

Versie	Datum	Auteur
1.0	20-12-2001	Mark Backer, Henk Gingnagel, Arjan Loeffen, Astrid Hackenberg
1.1	29-10-2002	Paul Vriend ism Mark Vlems
2.0	09-01-2004	Paul Vriend ism Mark Vlems, Paul Schlotter, Eduard Renger
3.0	16-03-2009	Dirk Temme
3.1	26-06-2013	Shinta Hadiutomo en Tonkie Zwaan
4.0	15-12-2017	Shinta Hadiutomo en Tonkie Zwaan

De datum is de goedkeuringsdatum door de Werkgroep XML (WGX).

2. Positionering

In de Suwi-keten hebben we te maken met de gedistribueerde systemen van de verschillende Suwi partijen, en met die van externe bronnen, basisregistraties en bedrijfsleven. De SuwiML Transactiestandaard (en de SuwiML Berichtstandaard) tracht een invulling te geven aan gestructureerde gegevensuitwisseling tussen die gedistribueerde systemen. Uiteraard gebeurt iets dergelijks in allerlei sectoren, en er zijn dan ook vele manieren om die problematiek aan te pakken. Om niet iedereen het wiel opnieuw te laten uitvinden is er standaardisatie nodig. Standaarden geven houvast en enige zekerheid voor de toekomst. In dit hoofdstuk een overzicht van de verschillende standaardisatie-initiatieven, en de relatie ervan tot de SuwiML Transactiestandaard.

2.1. Internationale standaarden

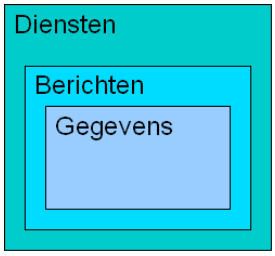
Als resultaat van gezamenlijke internationale samenwerkingsverbanden ontstaan er breed geaccepteerde internationale standaarden, ook op het gebied van gestructureerde gegevensuitwisseling. Sommige van die standaarden hebben inmiddels aanzienlijke draagkracht in de internationale gebruikersgemeenschap, en in de industrie die in bijbehorende software moet gaan voorzien. In de SuwiML Transactiestandaard willen we ons waar mogelijk conformeren aan dit soort internationale standaarden. Voorbeelden van belangrijke en relevante standaardisatie- organisaties zijn W3C, WS-I en OASIS. De SuwiML Transactiestandaard is met name gebaseerd op het WS-I Basic Profile 1.1, en de daarin gerefereerde W3C standaarden SOAP 1.1 en WSDL 1.1.

Afspraak 2: De SuwiML Transactiestandaard conformeert zich aan de Requirements in het WS-I Basic Profile 1.1.

Dit betekent dat er in de SuwiML Transactiestandaard hooguit aanscherpingen van het WS-I Basic Profile 1.1 gedaan zullen worden.

2.2. Nationale standaarden

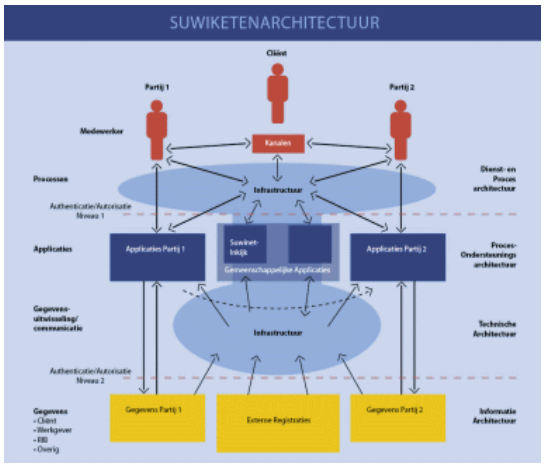
In de Suwi-keten achten we het ook zinvol om ons aan de principes uit de Nederlandse Overheid Referentie Architectuur (NORA, versie 3.0) te houden. Een van de basis uitgangspunten van de NORA (zie §4.3.2) is dat er voor een Servicegerichte Architectuur (SOA) gekozen wordt. De technieken die daarbij behulpzaam kunnen zijn worden in de Suwi-keten voor een deel (XML, SOAP) al sinds enkele jaren toegepast. De gegevensuitwisseling en de ontwikkeling daarvan is in onze sector echter tot nu toe voornamelijk Bericht-geörienteerd geweest.



De SuwiML Transactiestandaard beschrijft in feite de servicebus van de Suwi-keten. De toepassing van een zogenaamde servicebus in de scope van overheden en andere sectoren is Digikoppeling (voorheen OverheidsServiceBus). In deze versie van de SuwiML Transactiestandaard conformeren we ons aan het document Koppelvlakstandaard WUS Digikoppeling versie 3.5,

onderdeel van Digikoppeling 2.0. Voor de implementatie binnen onze sector maken wij gebruik van de keuzevrijheden, die de Digikoppeling standaard biedt. Formeel blijft de SuwiML Transactiestandaard een sectorale standaard. Dit betekent dat indien er aanleiding is de Suwiketen mag afwijken van de landelijke standaard.

2.3. Standaarden in de keten

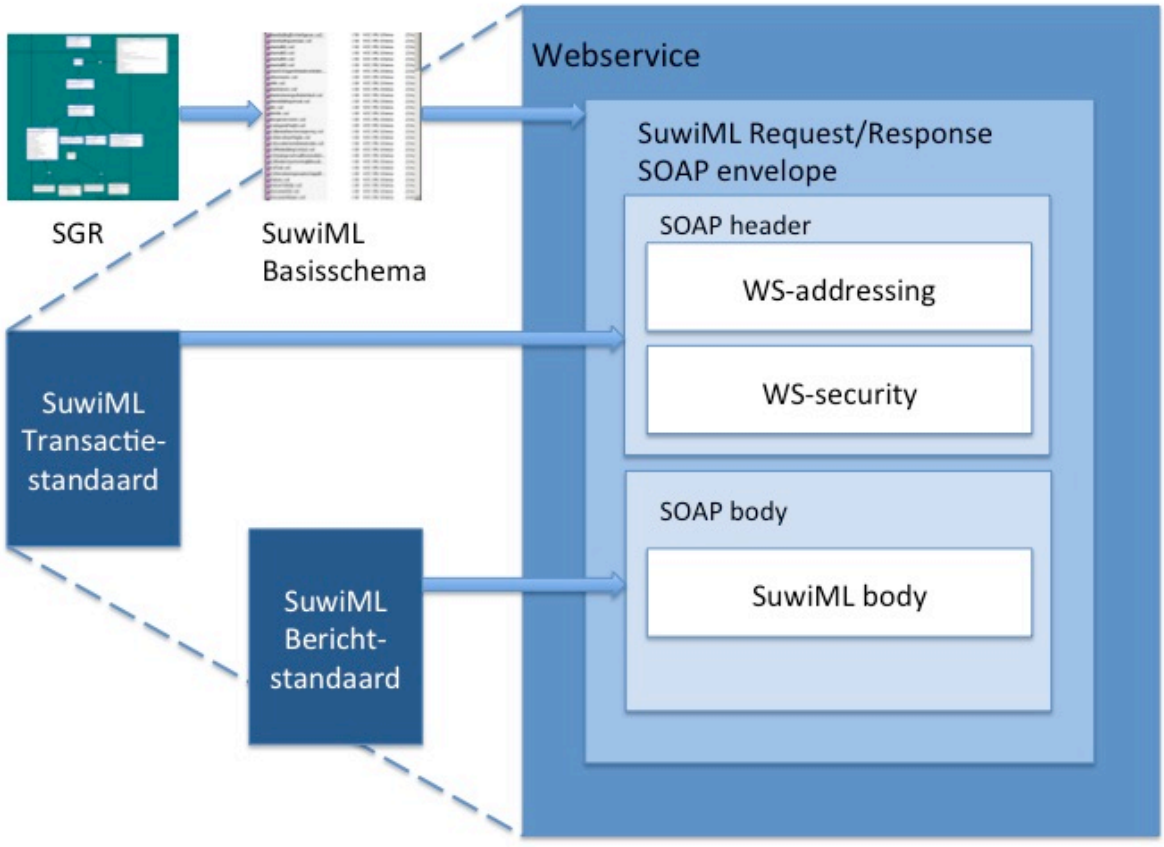


Naast de NORA is er de Ketenarchitectuur Werk en Inkomen (KArWeI). KArWeI versie 2.5 bevat beschrijvingen over Bedrijfsarchitectuur, Informatiearchitectuur en Technische architectuur. De SuwiML Transactiestandaard bevat een technische uitwerking van het Technische Architectuur gedeelte, en dan met name van de in de Suwi-ketenarchitectuur onderkende sessielaag, de transactielaag en de netwerklaag.

In de Suwi-keten kennen we naast de SuwiML Transactiestandaard ook het Suwi Gegevensregister (SGR), het SuwiML Basisschema, en de SuwiML Berichtstandaard. 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.

SuwiML is de vertaling in XML van het SGR. Alle entiteiten en attributen uit het SGR hebben een SuwiML XML-tag en datatype. Alle SuwiML bouwstenen tezamen vormen het SuwiML Basisschema. De SuwiML Berichtstandaard is een document met technische richtlijnen voor het definiëren van het inhoudelijke deel van SuwiML berichten. Tenslotte geeft de SuwiML Transactiestandaard dan technische richtlijnen voor het definiëren van de stuurinformatie in de SuwiML berichten, en voor het beschrijven van de koppelvlakken van de bijbehorende webservices.

In Afbeelding 1 worden de relaties tussen de verschillende onderdelen van SGR/SuwiML, noodzakelijk voor de opbouw van een SuwiML dienst, schematisch weergegeven. Het SuwiML basisschema is het XML synoniem voor het Suwi Gegevensregister en is van invloed op alle onderdelen van een SuwiML bericht. De SuwiML transactiestandaard schrijft de SOAP envelope structuur voor. De SuwiML berichtstandaard schrijft de opbouw van de SuwiML body voor.



Afbeelding 1 Opbouw berichten vanuit de Suwi standaarden

2.4. Verschillen met Digikoppeling

In de versie van de Transactiestandaard maken we gebruik van het document Koppelvlakstandaard WUS Digikoppeling versie 3.5, onderdeel van Digikoppeling 2.0.
Het genoemde document biedt een aantal vrijheden met betrekking tot de implementatie ervan. Hieronder worden de verschillen opgesomd:

1. De SuwiML Transactiestandaard zal in eerste instantie alleen het WUS profiel 2W-be-S ondersteunen. Als er de behoefte is om de inhoud (payload) van het bericht te beschermen tegen inzage door derden of op tussenstations, dan kan (optioneel) de inhoud van een bericht worden versleuteld (WUS profiel 2W-be-SE).
2. Het WS-Addressing element ‘wsa:From’ wordt verplicht gesteld (zowel voor request als response). Dit in tegenstelling tot Digikoppeling waar het element optioneel is.
3. De vulling van het element ‘wsa:From’ moet met de originele resource URI gevuld worden inclusief een OIN of subOIN.
4. In het WS-Addressing element ‘wsa:To’ zal zowel bij request als response de specifieke URI van respectievelijk het endpoint of ontvanger opgenomen moeten worden inclusief het OIN of subOIN.

Daar waar de Transactiestandaard verschilt van Digikoppeling, zal dit ook in verderop in de beschrijving uitgewerkt worden. Een SuwiML koppelvlakspecificatie moet zonder problemen goedgekeurd worden door de Compliancevoorziening van Logius.

2.5. Keuzes SuwiML i.r.t. gebruik Basic Profile 1.1, SOAP 1.1, WSDL 1.1

In de SOAP 1.1 standaard en ook in het Basic Profile 1.1 is het gebruik van een SOAP Header optioneel. In de Suwi keten is er echter een set aan stuurinformatie afgesproken. Op zijn minst gaat er met ieder bericht een MessageID mee in de SOAP Header. Het MessageID is onmisbaar bij het snel verhelpen van problemen. Bovendien dient vanwege het privacy-gevoelige karakter de informatie-uitwisseling in de Suwi Keten tot en met op bericht niveau gemonitord te kunnen worden.
De standaard SOAP 1.1 en ook Basic Profile1.1 laten de vulling van de SOAP Header geheel vrij. De SuwiML Transactiestandaard perkt die vrijheid in door slechts het gebruik van WS-Addressing in de Header toe te staan.

De WSDL 1.1 standaard kent vier patronen van bericht-uitwisseling: One Way, Request – Response, Sollicit – Response, en Notification. Het Basic Profile 1.1 beperkt de keuze tot One Way of Request – Response. De SuwiML Transactiestandaard verkiest voor alle uitwisselingen het Request – Response patroon. Ieder Bericht wordt dus beantwoord met een Response Bericht. De naam 'Request' betekent niet dat er in het heengaande Bericht alleen maar een vraag gesteld mag worden. We gebruiken zo'n 'Request' ook om een Melding of een Signaal of een Trigger te versturen.

Het Request – Response patroon kan wat WSDL 1.1 betreft zowel synchroon (Response in het HTTP back-channel van het Request) als asynchroon (Response in een nieuwe HTTP connectie) geïmplementeerd worden. De SuwiML Transactiestandaard verkiest voor alle uitwisselingen de synchrone variant, om daarmee optimale duidelijkheid te verschaffen over wat er precies in het back-channel van HTTP connecties verstuurd dient te worden.

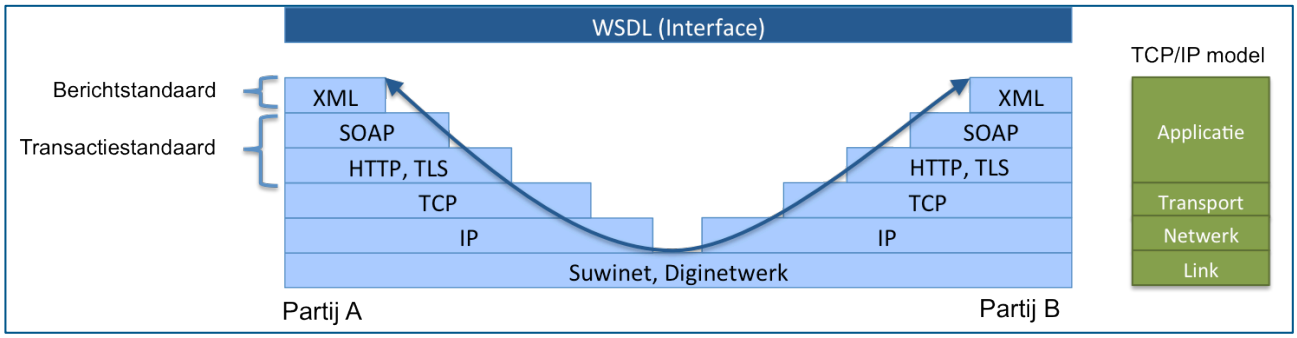
De WSDL 1.1 standaard schrijft voor hoe het gebruik van SOAP 1.1 in een WSDL file beschreven kan worden. De WSDL standaard staat daarbij enkele keuzes toe. Bijvoorbeeld het attribuut 'style' kan de waardes 'rpc' of 'document' bevatten. De SuwiML Transactiestandaard verkiest voor alle uitwisselingen de waarde 'document'. En volgens WSDL 1.1 kan het attribuut 'use' de waardes 'literal' of 'encoded' bevatten. Het Basic Profile 1.1 en de SuwiML Transactiestandaard verkiezen voor alle uitwisselingen de waarde 'literal'.

3. Onderliggende technische basis standaarden

Om tot gestructureerde informatie-uitwisseling tussen verschillende systemen te komen zijn er technische afspraken nodig op verschillende niveaus. Die afspraken worden vastgelegd in protocollen en aanvullend in bijvoorbeeld onderliggende SuwiML Transactiestandaard. In dit Hoofdstuk beschrijven we de basis ingrediënten, de Technische Standaarden waar we gebruik van maken en waar de verdere afspraken in deze Transactiestandaard over zullen gaan.

3.1. Verschillende lagen

Het lagen-model uit Afbeelding 2 toont wat er gebeurt bij informatie-uitwisseling tussen twee partijen. Laag voor laag wordt bij de ene partij de boodschap steeds verder ingepakt totdat het pakket klaar is om over de infrastructuur verzonden te worden. Bij de andere partij wordt het hele pakket dan weer laag voor laag uitgepakt totdat de originele boodschap overblijft.



Afbeelding 2 Technische niveaus en protocollen

De Transactiestandaard beschrijft hoe de lagen 'SOAP', 'HTTP' en 'TLS' ingevuld dienen te worden. De afspraken op de lagen 'SOAP' en 'HTTP' vinden hun weerslag in 'WSDL' beschrijvingen van de diensten van de partijen.

3.2. De XML laag

De bovenste laag is de XML laag, ook wel genoemd de 'Payload'. Op bedrijfs-applicatieniveau is bepaald dat er gegevens moeten worden uitgewisseld tussen verschillende applicaties / systemen. Hier is bekend welke gegevens uitgewisseld gaan worden en voor welke applicatie(s) de gegevens bestemd zijn.

De inhoudelijke functionele informatie die tussen partijen wordt uitgewisseld wordt vormgegeven en gestructureerd met behulp van XML. De XML variant van de Suwi-keten heet SuwiML. In de SuwiML Berichtstandaard staat tot in detail beschreven hoe met behulp van SuwiML de bericht-inhoud samengesteld en vastgelegd wordt. De Transactiestandaard doet daar verder geen uitspraken over.

3.3. De SOAP laag

De volgende laag is de SOAP laag, ook wel genoemd de berichtlaag. Hier wordt bepaald hoe de gegevens uitgewisseld moeten worden. De XML met de inhoudelijke boodschap wordt voorzien van nog wat extra XML-tags conform het SOAP 1.1 protocol. Een SOAP bericht bestaat uit een SOAP Envelope met daarin een SOAP Body en daarin weer de inhoudelijke XML boodschap uit de laag er boven. De SOAP Envelope bevat ook een SOAP Header met stuurinformatie op het gebied van Bericht-Identificatie, Routering en Adressering. Tevens kan/zal deze laag ook informatie bevatten voor het ondertekenen van berichten om de onweerlegbaarheid van de uitwisseling te waarborgen. De berichten kunnen onderweg niet aangepast worden. Functionaliteit

zorgt het voor dat de gegevens en identiteit van de zender of ontvangen onderweg niet aangepast worden. Hiermee wordt invulling gegeven aan eis (**GeVS 15.4 BIR/BIG 10.9.1(~) BIR/BIG 12.2.3**) dat maatregelen worden getroffen ten behoeve van de integriteit van gegevensleveringen.

De precieze invulling van de SOAP Header wordt beschreven in Hoofdstuk 5 SuwiML Berichten.

Afspraak 3: Ieder SuwiML Bericht heeft een SOAP Header met stuurinformatie

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsa:Action>http://bkwi.nl/SuwiML/Diensten/NaamService/Aanvraag</wsa:Action>
    <wsa:From>https://applicatie.afnemer.nl/dienst?oin=xxxxxxxxxxxxxxxxxxxx</wsa:From>
    <wsa:To>https://suwibroker-productie.suwinet.nl/suwibroker/soap/NaamService-
v0100?oin=xxxxxxxxxxxxxxxxxxxx</wsa:To>
    <wsa:MessageID>uuid: ca4270a9-091e-4194-a662-717ddf468bba</wsa:MessageID>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <smls:AanvraagInfo xmlns:smls="http://bkwi.nl/SuwiML/Diensten/NaamService">
      <Burgerservicenr>123456789</Burgerservicenr>
    </smls:AanvraagInfo>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Afbeelding 3 Structuur van een SuwiML SOAP-Envelope (voorbeeld)

3.3.1 WS-Security

In deze versie van de Transactiestandaard wordt het gebruikt van WS-Security elementen geïntroduceerd. Met WS-Security worden zaken m.b.t. integriteit en vertrouwelijkheid mogelijk om in te regelen. Integriteit betreft enerzijds het kunnen vaststellen dat het bericht onderweg niet aangepast is en dat het bericht van een te verifiëren vertrouwde partner afkomstig is (onweerlegbaarheid). Dit gebeurt door het bericht te voorzien van een XML-Signature gebaseerd op een PKI-Overheid certificaat. Om end-to-end secure gegevens uit te wisselen is deze optie verplicht en binnen het Suwidomein als er tussen aanvrager en endpoint één of meer tussenstation(s) bevinden.

De andere optie betreft de mogelijkheid zeer vertrouwelijk informatie, de XML payload, te encrypten. Deze optie wordt toegepast als geen elke partij die zich bevind tussen zender en ontvanger de inhoud van het bericht mag inzien. Immers, als de lijnverbinding met TLS beveiligd is, is de payload op bij tussenstation(s) nog steeds inzichtelijk. Op het moment dat de payload ge-encrypt moet worden, zullen niet transparante tussenstation(s) geen bewerkingen kunnen uitvoeren of over de payload kunnen rapporteren. Vooralsnog is de gegevensuitwisseling binnen het Suwidomein te categoriseren als risicoklasse 2 en is encrypten van de payload niet vereist. Dit is de reden waarom deze optie binnen WS-Security niet beschreven wordt in deze versie van de Transactiestandaard.

Afspraak 4: Om end-to-end secure gegevens binnen Suwidomein uit te wisselen is signing verplicht bij een situatie als er tussen aanvrager en endpoint één of meer tussenstation(s) bevinden.

3.4. De http laag

De HTTP laag verzorgt de bezorging van de berichten. Het SOAP bericht wordt naar een HTTP adres gestuurd. HTTP is de naam van het Transport protocol. HTTP informatie wordt meegestuurd met het bericht in de vorm van HTTP parameters. Bijvoorbeeld in de parameter Content-Type wordt een indicatie gegeven van de karakterset die gebruikt is in het bericht.

```
POST /axis2/services/VoorbeeldService HTTP/1.1
Content-Length: 1301
Content-Type: text/xml; charset=utf-8
SOAPAction: ""
User-Agent: Jakarta Commons-HttpClient/3.1
Host: 127.0.0.1:8081

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header>
      ...
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      <AanvraagInfo xmlns="http://bkwi.nl/SuwiML/Diensten/NaamService">
         <Burgerservicenr>123456789</Burgerservicenr>
      </AanvraagInfo>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Afbeelding 4 Voorbeeld van een SOAP bericht met daarboven de HTTP parameters

Het hangt van de gebruikte ontwikkel-tools af of en in welke mate er nog invloed mogelijk is op de HTTP parameters. De parameter SOAPAction wordt echter vastgelegd door de WSDL beschrijving van een webservice. Zie daarvoor ook 4.1.3 De WSDL Binding.

3.5. De TLS laag

Een HTTP connectie wordt versleuteld met behulp van tweezijding TLS (Transport Layer Security). Voor websites en webservices waarbij privacy-gevoelige of andere vertrouwelijke informatie verstuurd wordt (dus ook voor bijna alle SuwiML Diensten) is adequate versleuteling een must. TLS is een breed geaccepteerde en geïmplementeerde standaard daarvoor. Tussenstation(s) met een logistieke functie of afnemers en kunnen op basis van de verbinding geïdentificeerd worden.

Afspraak 5: De verbinding tussen twee partijen moet versleuteld worden d.m.v 2-zijdige TLS conform de richtlijnen van NCSC.

De service provider installeert daartoe een X509 server-certificaat op zijn webserver. De URL waar het request heen gestuurd wordt begint dan met de protocolnaam https://.... Om de informatie-uitwisseling ongestoord automatisch te laten verlopen moet de service requester in zijn component bekend maken dat de service provider vertrouwd kan worden. Dat kan door de publieke sleutel van het certificaat van de service provider in een keystore op te slaan.

Een service provider zal van een service requester verlangen dat die zich identificeert. Dat kan óók met behulp van een X509 certificaat. De service requester installeert een X509 client-certificaat en de service provider installeert de bijbehorende publieke sleutel in zijn keystore. Dit levert dan een zogenaamde 'dubbelzijdige' TLS connectie op. Zeker in het geval van vertrouwelijke informatie dient een service provider de identiteit van de service requester te controleren (authenticatie) en

ook moet gecontroleerd worden of die partij recht heeft op toegang tot de gevraagde dienst (autorisatie). TLS voorziet in een mechanisme daarvoor.

In de praktijk wordt de authenticatie en autorisatie van het TLS protocol afgehandeld in aparte hardware (bijvoorbeeld een Content Server Switch) of in aparte software die zich als eerste toegangspoort in de webserver installeert. De ontwikkelaar van de webservices en de SOAP berichten hoeft zich daar dan als het goed is geen zorgen meer over te maken.

3.6. WSDL en XML schema

Conform het WS-I Basic Profile 1.1, zie http://www.ws-i.org/Profiles/BasicProfile-1.1.html, worden de koppelvlakken van Suwi webservices beschreven met behulp van Webservices Description Language (WSDL) 1.1 files. WSDL is net als SuwiML en SOAP ook een gestructureerde XML variant. Een WSDL beschrijving van een webservice laat precies zien hoe de lagen 'SOAP' en 'HTTP' uit bovenstaande Afbeelding 2 ingevuld zijn voor die webservice.

Afspraak 6: De koppelvlak specificaties van een SuwiML Webservice worden vastgelegd in en bepaald door een WSDL beschrijving

De WSDL file is dus 'leading'. Documenten als deze Transactiestandaard en het WS-I Basic Profile dienen 'slechts' ter ondersteuning. Vanuit de WSDL file wordt precies de structuur van de berichten vastgelegd. Hiermee sluiten we aan op de internationale W3C-standaarden en het gebruik van deze standaarden binnen ontwikkel platforms. Door deze methode te hanteren wordt het mogelijk om elementen van deze standaard vanuit de WSDL aan te roepen voor het genereren van koppelvlakken. Denk hierbij aan bijvoorbeeld WS-Addressing functionaliteiten.

In de WSDL file worden 'import's gedaan van een aantal XML Schema files. De XML Schema files beschrijven de afzonderlijke bouwstenen die betrokken zijn bij de webservice. Het betreft bouwstenen uit het SuwiML Basisschema, en bouwstenen voor herbruikbare componenten zoals Headers en Foutafhandeling.

4. SuwiML webservices

Iedere afzonderlijke webservice interface / gegevensuitwisseling, wordt volgens SGR/SuwiML vastgelegd door een technische beschrijving in een WSDL file met bijbehorende SuwiML bericht-schema's. Een webservice wordt uniek geïdentificeerd door middel van de targetNamespace van de WSDL file. In die URI dient ook het versienummer opgenomen te worden. De webservice URI komt ook terug als namespace in de requests die naar de webservice gestuurd worden en de responses die door de webservice terug gestuurd worden.

4.1. De WSDL file van een SuwiML webservice

In de WSDL wordt vastgelegd hoe het request er uit moet zien dat door de zender (de client) verstuurd zal worden, èn hoe dan het bijbehorende response van de ontvanger (de server) er uit zal zien.

Indien er een fout optreedt tijdens de gegevensuitwisseling, bijvoorbeeld als de ontvanger niet in staat is de request te verwerken en/of te beantwoorden, dan wordt een foutmelding geretourneerd. Hoofdstuk 6 Foutafhandeling beschrijft in detail op welke wijze de foutafhandeling plaatsvindt binnen de verschillende lagen.

4.1.1 De WSDL PortType

Het centrale onderdeel van een WSDL file is de PortType. Met de PortType wordt de interface van de webservice beschreven, in termen van een of meer Operations voorzien van Input en Output parameters.

Afspraak 7: Iedere Operation van een SuwiML webservice heeft zowel een Input als een Output

```
<wsdl:definitions ... xmlns:smls="http://bkwi.nl/SuwiML/Diensten/NaamService">
   ...
   <wsdl:message name="Aanvraag">
      <wsdl:part name="parameters" element="smls:AanvraagInfo"/>
   </wsdl:message>
   <wsdl:message name="Levering">
      <wsdl:part name="parameters" element="smls:AanvraagInfoResponse"/>
   </wsdl:message>
   ...
   <wsdl:portType name="TestInfo">
      <wsdl:operation name="AanvraagInfo">
         <wsdl:input message="Aanvraag"/>
         <wsdl:output message="Levering"/>
      </wsdl:operation>
   </wsdl:portType>
   ...
</wsdl:definitions>
```

Afbeelding 5 XML weergave van een PortType in WSDL

Aan de XML weergave in Afbeelding 5 is te zien dat in de Operation 'AanvraagInfo' voor de Input verwezen wordt naar het Element 'smls:AanvraagInfo' en voor de Output naar het element

'smls:AanvraagInfoResponse'. Die elementnamen zullen in het berichtenverkeer terug komen als de eerste tag in de SOAP Body van een Request en van een Response.

4.1.2 Document – literal wrapped stijl

Een document literal wrapped inrichting van een webservice zorgt ervoor dat de XML requests en responses optimaal gemapped kunnen worden op een object structuur die vrij is van overbodige Request en Response objecten. Op dit moment is er met de document literal wrapped style en met bijbehorende afspraken ook de beste kans op succes bij het automatisch interpreteren van een WSDL door de verschillende beschikbare SOAP toolkits, bijvoorbeeld die van Microsoft .Net en Apache Axis2. Zie bijvoorbeeld http://pzf.fremantle.org/2007/05/handlign.html en http://www.ibm.com/developerworks/webservices/library/ws-whichwsdl/. Bovendien ondersteunt Digikoppeling dit advies.

Afspraak 8: Iedere SuwiML webservice heeft een 'Document – Literal Wrapped' interface

Om een 'document-literal wrapped' interface te krijgen dienen enkele richtlijnen gevolgd te worden. Voor de PortType en de bijbehorende Messages in een WSDL geldt het volgende:

- Iedere Message heeft slechts één Message Part met de naam 'parameters';
- Ieder Message Part verwijst naar een 'element', niet naar een 'type';
- De naam van het Message Part element in de Input is gelijk aan de naam van de Operation;
- De naam van het Message Part element in de Output is gelijk aan de naam van de Operation, met Response' er aan toegevoegd.

4.1.3 De WSDL Binding

De WSDL 1.1 standaard is op zichzelf niet gebonden aan het gebruik van SOAP, maar er is wel in de WSDL 1.1 standaard een SOAP Binding opgenomen. Deze SOAP Binding gebruikt de SOAP 1.1 (zie http://www.w3.org/TR/soap/) standaard. Het WS-I Basic Profile 1.1 (en ook deze versie van de SuwiML Transactiestandaard) stelt het gebruik van deze SOAP 1.1 Binding verplicht. Als gevolg daarvan zullen de berichten van en naar WS-I Basic Profile 1.1 conforme webservices opgesteld zijn in het SOAP 1.1 formaat. Waar het WS-I Basic Profile 1.1 nog wat vrijheid overlaat voor de 'style' van de binding, kiest de SuwiML Transactiestandaard dus voor de 'document – literal' style. Daarom heeft het attribuut 'style' van een SOAP Binding de waarde 'document'.

In de SOAP Binding wordt ook de waarde van de HTTP parameter SOAPAction vastgelegd. Aangezien er met de introductie van WS-Addressing in de SOAP Headers een veld <wsa:Action> opgenomen kan worden, verliest de HTTP parameter SOAPAction zijn waarde. In de WSDL moet er echter wel iets over gezegd worden. Voor SuwiML webservices kiezen we er voor om de SOAPAction leeg te laten: <soap:operation soapAction=””/>, zie Afbeelding 6

```
<wsdl:definitions ... xmlns:smls="http://bkwi.nl/SuwiML/Diensten/NaamService">
   ...
   <wsdl:binding name="TestBinding" type="smls:TestInfo">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsaw:UsingAddressing wsdl:required="true"/>
      <wsp:UsingPolicy wsdl:Required="true"/>
      <wsp:Policy wsu:Id="myPolicy">
         <wsp:ExactlyOne>
            <wsp:All>
               <sp:SecurityToken>
                  <sp:TokenType>sp:X509v3</sp:TokenType>
               </sp:SecurityToken>
               <sp:UsernameToken/>
               <sp:SignedParts>
                  <!-- Beschrijft welke onderdelen van het bericht header en/of body gesigneerd worden.--
>
                  <sp:Body/>
                  <sp:Header Name="To"/>
                  <sp:Header Name="From"/>
                  <sp:Header Name="FaultTo"/>
                  <sp:Header Name="ReplyTo"/>
                  <sp:Header Name="MessageID"/>
                  <sp:Header Name="RelatesTo"/>
                  <sp:Header Name="Action"/>
               </sp:SignedParts>
      ...
      </wsp:Policy>
      <wsdl:operation name="TestPersoonsInfo">
         <soap:operation soapAction=""/>
         <wsdl:input>
            <soap:body use="literal"/>
         </wsdl:input>
         <wsdl:output>
            <soap:body use="literal"/>
         </wsdl:output>
         <wsdl:fault name="AanvraagNietOk">
            <soap:fault name="AanvraagNietOk" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
   </wsdl:binding>
   ...
</wsdl:definitions>
```

Afbeelding 6 De XML weergave van een SOAP binding

Afspraak 9: Voor SuwiML webservices is de HTTP parameter SOAPAction leeg: “”

In de Binding dient ook informatie over benodigde stuurinformatie vastgelegd te worden. Om aan te sluiten bij de Digikoppeling-standaarden maken we gebruik van WS-Addressing.

Afspraak 10: SuwiML webservices maken gebruik van WS-Addressing

In Afbeelding 6 blijkt het gebruik van WS-Addressing uit het element <wsaw:UsingAddressing .../>. Voor verdere informatie over het gebruik van WS-Addressing, zie 4.8.1 Ondersteuning voor WS-Addressing.

Voorheen waren er in de Suwi-keten zelf-gedefinieerde headers om stuurinformatie met de berichten mee te sturen. Mocht er voor een bepaalde webservice nog gebruik gemaakt worden van die Suwi-specifieke stuurinformatie, dan dient dat in de WSDL koppelvlak specificaties van de webservice kenbaar gemaakt te worden met het element <soap:header .../>.

Afspraak 11: Ten behoeve van het ondertekenen van onderdelen in berichten wordt in de WSDL gebruik gemaakt van WS-Policy.

Een WSDL geeft een beschrijving van de eisen die ten aanzien van de communicatie gesteld worden. Een WSDL kan onder andere bestaan uit meerdere schema definities in aparte XSD's en policy definities. Gezamenlijk vormt dit een abstracte definitie van de webservice. De webservice communiceert feitelijk door middel van SOAP berichten, die gegenereerd worden op basis van de WSDL. Voor de authenticatie en encryptie wordt gebruikgemaakt van WS-Security.
Een contract wordt voor een Digikoppeling WUS koppelvlak gedefinieerd door een WSDL. De WSDL 1.1 specificatie op zich biedt geen mogelijkheden om het gebruik van WS-Security aan te geven. Conform de Digikoppeling-standaard is de WSDL- specificatie wel extensible en er zijn standaarden waarin het gebruik van WS-Security kan worden aangegeven.
Dit zijn:

- WS-Policy (versie 1.2 / 1.5),
- WS-PolicyAttachment (versie 1.2/1.5),
- WS-SecurityPolicy 1.1

Afspraak 12: Eventueel gebruik van SuwiML stuurinformatie voor een webservice wordt in de WSDL koppelvlak specificaties van die webservice vastgelegd

4.1.4 De WSDL Service

Waar de WSDL PortType de functionele aspecten van de koppelvlak definitie (vrije vertaling van 'interface') van een webservice beschrijft, staat de WSDL Service voor een concrete implementatie van die koppelvlak definitie. De WSDL Binding vormt de verbinding tussen de twee. De WSDL Service bestaat uit één of meer endpoints, voorzien van een URL http(s) adres, waarop de implementatie(s) bereikbaar zijn.

```
<wsdl:definitions ... xmlns:smls="http://bkwi.nl/SuwiML/Diensten/NaamService">
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
…
   <wsdl:service name=""NaamService">
      <wsdl:port name="BrokerProductie" binding="smls:TestBinding">
         <soap:address location="https://suwibroker-
productie.suwinet.nl/suwibroker/soap/NaamService"/>
      </wsdl:port>
      <wsdl:port name="BrokerIntegratietest" binding="smls:TestBinding">
         <soap:address location="https://suwibroker-
integratietest.suwinet.nl/suwibroker/soap/NaamService"/>
      </wsdl:port>
      <wsdl:port name="BrokerBTO" binding="smls:TestBinding">
         <soap:address location="https://suwibroker-bto.suwinet.nl/suwibroker/soap/NaamService"/>
      </wsdl:port>
   </wsdl:service>

</wsdl:definitions>
```

Afbeelding 7 Verschillende implementaties met hun URL's

4.2. Bevragingen versus Meldingen

Vanwege het vervallen de ondersteuning van WSRM in Digikoppeling, wordt opnieuw overwogen welk protocol voor meldingen binnen de keten Werk & Inkomen gebruikt zal worden. Tot die tijd wordt verwezen naar de Transactiestandaard 3.1 verwezen voor de beschrijving en richtlijnen m.b.t. meldingen.
Zodra de keuze bekend is, zal dit document weer aangepast worden.

4.3. Brokers en andere tussenstations

Er kunnen verschillende redenen zijn om een bepaalde service op meerdere plaatsen aan te bieden. Er wordt bijvoorbeeld vaak gebruik gemaakt van een Broker bij wijze van centrale toegangspoort tot allerlei achterliggende services. Deze services worden dan dus zowel geleverd door de achterliggende systemen als ook door de broker, namens de achterliggende services. Een dergelijke opzet kan voordelen bieden op het gebied van hergebruik, caching en het inregelen van autorisaties. Zowel UWV, als UWV Werkbedrijf als de centrale Suwinet Inkijk omgeving maken gebruik van een Broker (UWV gebruikt de Klantbeeldserver (KBS), Systeem Integratie Platform (SIP), UWV Werkbedrijf een Oracle ESB, en Suwinet Inkijk de Suwi Broker). Het Sectorloket van het Inlichtingenbureau fungeert ook als een soort Broker voor de achterliggende services (die voor Bijstandsregelingen en GSD Dossier Re-integratie) van de Gemeentelijke systemen. Het kan ook voorkomen dat een service op verschillende netwerken aangeboden wordt, bijvoorbeeld op het besloten Suwinet. Tenslotte kan een partij er voor kiezen om alle zorgen op het gebied van connectiviteit, monitoring, beveiliging, enz. uit handen te geven en het transport te laten verzorgen door een derde partij.

We onderkennen verschillende type van tussenstations. Ten behoeve van de authenticatie en autorisatie onderkennen hierbij ook verschillende rollen.

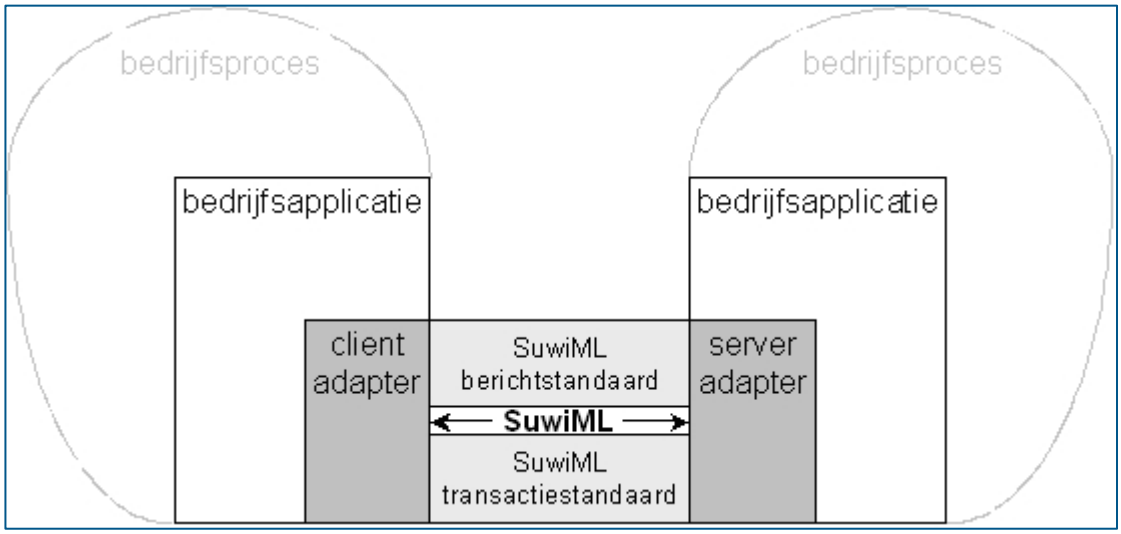
Transparant tussenstation
Een transparant tussenstation heeft als rol berichten te routeren zonder een bewerking op header bericht te doen. Een transparant tussenstation kan het bericht afleveren bij een ander tussen station of het endpoint waar het koppelvlak geïmplementeerd is.

Niet transparant tussenstation
Een niet transparant tussenstation kenmerkt zich op twee verschillen de manier van een transparant tussenstation. Te weten:

- Als een transparant tussenstation het bericht in SuwiML moet omzetten voor een achterliggende webservice of applicatie. Een dergelijk tussenstation zorgt ervoor dat de juiste identiteit en handtekening voor het doel (autorisatie) van de bevraging meegegeven wordt.
- Als een applicatie meerdere doelen (autorisaties) van één of meer organisaties moet bedienen. Onder het algemeen wordt deze situatie als een SAAS-oplossing betiteld. Een dergelijke situatie kan ook voorkomen bij een samenwerkingsverband van meer dan 1 organisatie. De SAAS-oplossing.

Een endpoint is een locatie waar een specifiek koppelvlak afgenomen kan worden. De SuwiBroker is een dergelijk endpoint voor koppelvlakken. De Suwibroker treedt op als endpoint voor bronnen van gegevens binnen de het Suwidomein. En als SAAS-oplossing voor bronnen van gegevens van buiten het Suwidomein.

4.4. SOAP adapters



Afbeelding 8 Schematisch beeld van het gebruik van SuwiML bij de gegevensuitwisseling tussen client en server (voorbeeld)

In de praktijk komt het vaak voor dat de SOAP communicatie afgehandeld wordt door een speciale SOAP Adapter, terwijl de functionele inhoud van het SOAP bericht bestemd is voor een achterliggende andere bedrijfsapplicatie. De communicatie tussen de adapter en de bedrijfs-applicatie zal vaak geen SOAP zijn en misschien zelfs geen XML. De adapter transformeert van het interne protocol naar het protocol van de keten, en andersom. Wanneer een partij over een broker beschikt dan zal die broker de rol van adapter op zich nemen. De SuwiML Transactiestandaard gaat over de communicatie tussen de broker / adapter van de client partij en die van de server partij, en niet over de interne communicatie bij een partij tussen de broker / adapter en de bedrijfsapplicatie.

4.5. Verschillende versies

Voor iedere SuwiML webservice kunnen meerdere versies in omloop zijn. Echter iedere request-response-fout set wordt door precies één en dezelfde SuwiML webservice- specificatie bepaald. Het versienummer wordt opgenomen in de targetnamespace van de WSDL en komt ook terug in de namespace van de body van het Request en het Response. Bij ieder request van een Client applicatie is dus voor de Server applicatie herkenbaar voor welke versie van de webservice het request bedoeld is.

In de naam van het .zip pakket met de WSDL en de bijbehorende XML Schema files staat naast het versie-nummer nog een build-nummer. Het build-nummer wordt ***niet*** opgenomen in de targetnamespace en in de namespace van de body van het Request en het response. Gedurende het ontwikkel- en testtraject van een nieuwe (versie van een) webservice kan naar aanleiding van de bevindingen het build-nummer nog veranderen.

4.6. SOAP toolkits

Iedere partij bouwt zijn eigen implementatie van de SuwiML webservices die zij wensen te gebruiken. Zij kunnen dat doen met zelf te kiezen tools en ontwikkelomgevingen.

4.7. Stuurgegevens

In een WSDL koppelvlak-beschrijving wordt meegegeven welke soort stuurgegevens er van belang zijn voor de webservice. In vorige versies van de SuwiML Transactiestandaard werd er alleen de eigen SuwiML Header gebruikt. Inmiddels worden de stuurgegevens opgenomen in de elementen van WS-Addressing. In deze en toekomstige versies van de SuwiML Transactiestandaard worden de onderdelen uit de SuwiML Header vervangen door soortgelijke headers van de relevante WS-* en/of andere algemeen geldende standaarden.

4.7.1 Ondersteuning voor WS-Adressing

De standaard Web Services Addressing 1.0 - Core (W3C Recommendation 9 May 2006) bepaalt welke velden er in de header van een bericht meegestuurd dienen te worden ten behoeve van adequate adressering van het bericht. In de specificatie "Web Services Addressing 1.0 - WSDL Binding" (W3C Candidate Recommendation 29 May 2006) wordt aangegeven hoe in een WSDL beschrijving het gebruik van WS-Addressing gestuurd kan worden.

Het element <wsaw:UsingAddressing wsdl:required="true"/> wordt in iedere WSDL toegevoegd als child van het element <wsdl:Definitions> / <wsdl:Binding>, bij wijze van indicatie dat het gebruik van WS-Addressing verplicht is. De prefix wsaw staat voor de namespace http://www.w3.org/2006/05/addressing/wsdl.

Ieder <wsdl:Input> element in een WSDL beschrijving krijgt een attribuut wsaw:Action. Iedere client implementatie dient de waarde van het attribuut wsaw:Action te gebruiken als waarde voor het element <wsa:Action> in elk bijbehorend request.

Ieder <wsdl:Output> element in een WSDL beschrijving krijgt ook een attribuut wsaw:Action. Iedere server implementatie dient de waarde van het attribuut wsaw:Action te gebruiken als waarde voor het element <wsa:Action> in elk bijbehorend response.

Zie verder 5.2 Adressering voor de gevolgen van WS-Addressing voor de afzonderlijke Berichten.

4.8. Ondersteuning voor binaire bestanden

Niet alle data leent zich er voor om in XML formaat verstuurd te worden. Er zijn webservices denkbaar, ook in de Werk en Inkomen keten, die bij de te versturen data een bestand zoals een CV, een Diploma of een Foto zouden willen meesturen. In de WSDL van een dergelijke webservice kan dat aangegeven worden door het gebruik van elementen van het type "xmime:base64Binary".

Bijvoorbeeld:

```
<element name="Foto" xmime:expectedContentTypes="image/jpeg" type="xmime:base64Binary"/>
<element name="CV" xmime:expectedContentTypes="application/pdf" type="xmime:base64Binary"/>
```

Per mee te leveren bestand dient dus ook het mime-type (image/jpeg, image/png, application/pdf, ...) van het bestand vermeld te worden.

In de koppelvlak specificaties van de SuwiML webservices zal het meeleveren van binaire bestanden ook op deze manier worden vormgegeven.

In de WSDL wordt voor het Request van de operation “Upload” verwezen naar het element “smls:Upload”. En voor de het Response van de operation “DownloadCV” wordt verwezen naar het element “smls:DownloadCV”. Als voorbeeld wordt het element “smls:Upload” gedefinieerd in de XML Schema file BodyAction.xsd:

```
<<xs:element name="Upload">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="Burgerservicenr" type="sml:Burgerservicenr"/>
         <xs:element name="Foto" type="xmime:base64Binary"
                        xmime:expectedContentTypes="image/jpeg, image/png"/>
         <xs:element name="CV" type="xmime:base64Binary"
                        xmime:expectedContentTypes="application/pdf"/>
         <xs:element name="Diploma" type="xmime:base64Binary"
                        xmime:expectedContentTypes="application/pdf"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Afbeelding 9 Voorbeeld van elementen van het type "xmime:base64Binary"

Ook het element “smls:DownloadCV”, dat wordt gedefinieerd in de XML Schema file BodyReaction.xsd, bevat een dergelijk subelement van het type “xmime:base64Binary”. Met het attribuut xmime:expectedContentTypes wordt aangegeven wat voor type binair bestand er met dit element verstuurd mag worden.

De aanwijzing xmime:expectedContentTypes="..." in de WSDL dwingt ***niet*** af dat er in de praktijk ook daadwerkelijk altijd een bestand van dat type verstuurd wordt. Er zou in de praktijk misbruik gemaakt kunnen worden van een dergelijke webservice door een (mogelijk kwaadaardig) bestand van een ander type te voorzien van een extensie die door de webservice geaccepteerd wordt. De partijen die de koppelvlakspecificaties implementeren dienen zich daarvan bewust te zijn, de risico's in te schatten en eventuele voorzorgsmaatregelen te nemen.

De aanwijzingen in de WSDL zeggen niets over de maximale grootte van de bij te voegen bestanden. Ook voor dat aspect is het dus van belang dat de partijen die de webservice implementeren de grenzen van hun eigen systemen kennen, en dat ze eventueel voorzorgsmaatregelen nemen. Aan de HTTP parameter “Content-Length” is bijvoorbeeld wel te zien hoe groot een inkomend bericht is.

Zie verder 5.7 Berichten met Binaire Bestanden voor een bespreking van de gevolgen van dit soort koppelvlak specificaties voor de bijbehorende XML berichten.

5. SuwiML berichten

Een SuwiML Bericht is een request gericht aan een SuwiML webservice ofwel een response afkomstig van een SuwiML webservice. Een SuwiML Bericht bestaat uit:

- Een SOAP Envelope
- Met in de SOAP Envelope
    - Een SOAP Header met stuurgegevens conform deze SuwiML Transactiestandaard, inclusief de WS-Security-elementen t.b.v. signing (elektronische handtekening).
    - En een SOAP Body met de applicatie-specifieke bericht-inhoud conform de bij de service horende XML-Schema's, en opgebouwd volgens de SuwiML Berichtstandaard.

5.1. Gemeenschappelijk deel

De SOAP Header wordt volgens de richtlijnen van deze Transactiestandaard gevuld met de stuurgegevens ten behoeve van adressering, routering, bericht-identificatie. Voor de stuurinformatie worden de geaccepteerde internationale standaarden gebruikt. Het betreft de standaard WS-Addressing 1.0 voor de tags <wsa:MessageId>, <wsa:Action>, <wsa:To> en <wsa:From>.

5.2. Adressering

Voor stuurinformatie met betrekking tot Adressering maken alle nieuwe services verplicht gebruik van de WS-Addressing standaard, zie http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/. In '4.8.1 Ondersteuning voor WS-Addressing' werd besproken hoe het gebruik van WS-Addressing in de WSDL kenbaar wordt gemaakt.

Tabel 1 Implementatierichtlijnen WS-Addressing elementen voor een synchroon request

Tag	Verplichtingen: V/C/O	Vulling	Omschrijving	Bron
<wsa:MessageID>	Verplicht	URI ter identificatie van een enkel request: http://afzender/ someuniquestring	Unieke identificatie van het bericht t.b.v. traceerbaarheid.	Te bepalen door de afzender
<wsa:From>	Verplicht	URI van de vragende webservice of applicatie: https://applicatie.afnemer.nl/dienst?oin=xxxxxxxxxxxxxxxxxxx	Autorisatie van de requester inclusief (sub)OIN	Het volledige adres van de oorspronkelijk webservice die het request verstuurd
<wsa:To>	Verplicht	URI ter identifcatie van het endpoint van de webservice: https://suwibroker-productie.suwinet.nl/suwibroker/soap/naamservice-v0100?oin=xxxxxxxxxxxxxxxxxxx	Endpoint van het request inclusief (sub)OIN	wsdl:definitions / wsdl:service/ wsdl:port/ soap:address/ @location

<wsa:Action>	Verplicht	URI ter identificatie van het input-bericht zoals gedefinieerd in de WSDL	Unieke identificatie zoals bepaald in de WSDL voor het Request, Response en Fout.	wsdl:definitions / wsdl:portType / wsdl:operation / wsdl:input / @wsaw:Action

In deze versie van de Transactiestandaard gebruiken we de de volgende elementen in de SOAP Headers: <wsa:MessageID>, <wsa:Action>, <wsa:From>, <wsa:To> en <wsa:RelatesTo>.

5.2.1 Toepassing in de synchrone gegevensuitwisseling

Dit gedeelte beschrijft hoe WS-Addressing-elementen in de synchrone gegevensuitwisseling binnen de keten worden geïmplementeerd.

Met een verwijzing in de WSDL naar het regel: <wsaw:UsingAddressing wsdl:required="true"/>, hoeven de optionele elementen niet expliciet te worden uitgeschreven. Als de toolkit de optionele WSA-elementen automatisch genereert, die niet onderdrukt kunnen worden, dan gelden de voorschriften conform Digikoppeling. Hierbij geldt wel de beperking dat de waarde voor deze elementen het routeringsmechanisme niet verstoort.

Tabel 2 Implementatierichtlijnen WS-Addressing elementen voor een synchroon Response op een synchroon Request

Tag	Verplichtingen: V/C/O	Vulling	Omschrijving	Bron
<wsa:MessageID>	Verplicht	URI ter identificatie van een enkel response: http://beantwoorder/someother uniquestring	Unieke identificatie van het bericht t.b.v. traceerbaarheid.	Te bepalen door de zender van het Response
<wsa:From>	Verplicht	URI van de antwoordende webservice of applicatie: https://suwibroker-productie.suwinet.nl/suwibroker/soap/naamservice-v0100?oin=xxxxxxxxxxxxxxxxxxx	t.b.v. logging.	Het volledige adres van de oorspronkelijk webservice die het response verstuurd (wsa:To uit request)
<wsa:To>	Verplicht	URI ter identifcatie van het endpoint van de webservice: https://applicatie.afnemer.nl/dienst?oin=xxxxxxxxxxxxxxxxxxx	t.b.v. logging	Wordt gevuld met het adres uit de From van het request
<wsa:Action>	Verplicht	URI ter identificatie van het output-bericht zoals gedefinieerd in de WSDL	Unieke identificatie zoals bepaald in de WSDL voor het Request, Response en Fout.	wsdl:definitions / wsdl:portType / wsdl:operation / wsdl:output / @wsaw:Action

<wsa:RelatesTo>	Verplicht	URI ter identificatie van het bijbehorende request: http://afzender/someuniquestring	Een verwijzing naar de Identificatie van het bericht uit het bijbehorende Request	<wsa:MessageId> uit het Request

Note: Alle elementen WS-Addressing zijn volgens de standaard optioneel behalve <wsa:Action> en mogen in willekeurige volgorde in de header voorkomen. De implementatie bepaalt of elementen gebruikt moeten worden of optioneel zijn.

Mocht er in een bepaalde implementatie de optionele elementen toch geleverd worden, dan worden deze in de verwerking genegeerd. Het gedrag zal dan ook hetzelfde zijn als wanneer deze elementen niet in de headers van de berichten opgenomen zouden zijn.

5.3. Identificatie, authenticatie en autorisatie

Zoals eerder is aangegeven worden in deze versie van Transactiestandaard, versie 4.0, wijzigingen opgenomen in het kader van end-to-end security voor bevragingen.
Door het vervallen van de SuwiML header, moet er gebruik gemaakt worden van andere identificerende gegevens om de autorisatie toe te passen. De basis voor de autorisatie is het (sub-)OIN. Voor het authenticatieproces wordt van het PKI-overheid certificaat gebruikgemaakt. Hier mee wordt de elektronische handtekening gezet.

5.3.1 OIN, sub-OIN's, PKI-overheid certificaat, etc.

Het Organisatie-Identificatienummer (OIN) is een onderdeel van de Digikoppeling standaard, die wordt gebruikt die wordt gebruikt in elektronische berichtuitwisseling door of met de overheid. Digikoppeling verplicht de opname van het OIN in het PKIoverheid-certificaat zodat organisaties kunnen worden geïdentificeerd en geauthentiseerd. Daarmee is het OIN een randvoorwaarde voor veilig digitaal verkeer. Het OIN wordt gebruikt op verschillende manieren en met vijf doeleinden: de identificatie, authenticatie en autorisatie van organisaties of organisatie- onderdelen, en de routering en de adressering van berichten naar organisaties, organisatie- onderdelen en voorzieningen. Om deze doelen te kunnen bereiken is het mogelijk om onder een OIN sub-OIN's te definiëren en aan te maken. Met sub-OIN's is het mogelijk om binnen een organisatie verschillende niveaus van autorisatie te erkennen. Hierdoor blijft fijnmazig autoriseren mogelijk.

Voor meer informatie over OIN zie bijlage 1.

5.3.2 Identificatie van partijen / componenten / applicaties

Het identificeren van partijen valt uiteen in twee delen:
1. identificeren van afnemer (organisatie, afdeling/organisatie-unit, applicatie, etc.)
2. het identificeren van transparante- (IB-Broker, SAAS-applicatie, etc.) en niet-transparante tussenstations (SuwiBroker)

Welke route een bericht aflegt, is niet relevant. Voor de gegevensuitwisseling tussen 2 webservices maken we gebruik van het Digikoppeling profiel 2W-BE-S.
2W= 2 weg (dubbelzijdige) TLS-verbinding, BE= Best Effort, S=Signed. Dit betekent dat de verzendende partij het bericht signeert met zijn private (geheime) sleutel, de ontvangende partij

ontcijfert de elektronische handtekening (signing) met de publieke sleutel van de verzendende partij en de verbinding vindt plaats op basis van een TLS-PKI-Overheid certificaat.

5.3.3 Autorisatie

Door het sub-OIN wordt het mogelijk op verschillende niveaus te onderkennen. De doelbinding waarvoor de gegevens nodig zijn is daar leidend in. Op welke wijze de autorisatie ingericht wordt, wordt niet bepaald door de Transactiestandaard. Om een beeld van niveaus van autorisatie te geven wat mogelijk is noemen we een aantal voorbeelden.

Voorbeelden van autorisatie niveaus:

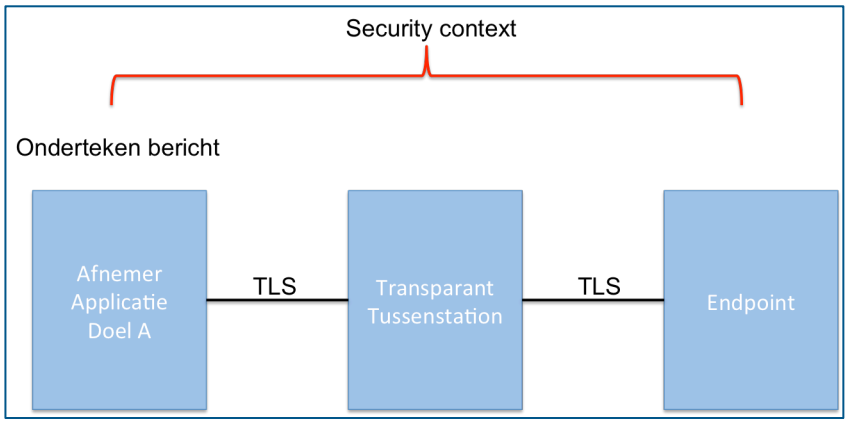
- Applicatie van een organisatie of organisatieonderdeel - GSD applicatie Amersfoort
- Afdeling van een organisatie – Afdeling Handhaving UWV
- Uitvoering van een doel – Re-integratie

Afspraak 13: Vertrouwelijke informatie wordt alleen verstrekt aan geautoriseerde partijen voor specifieke doelen.

5.3.4 Uitwerking 2W-be-S met 1 of meer tussenstations

In deze paragraaf wordt een uitwerking gegeven hoe om te gaan met tussenstations. Zie paragraaf 4.3 voor de beschrijving van de verschillende type tussenstations. Bij alle afbeeldingen wordt één tussenstation weergegeven. In werkelijkheid kunnen er meer dan één tussenstation voorkomen.

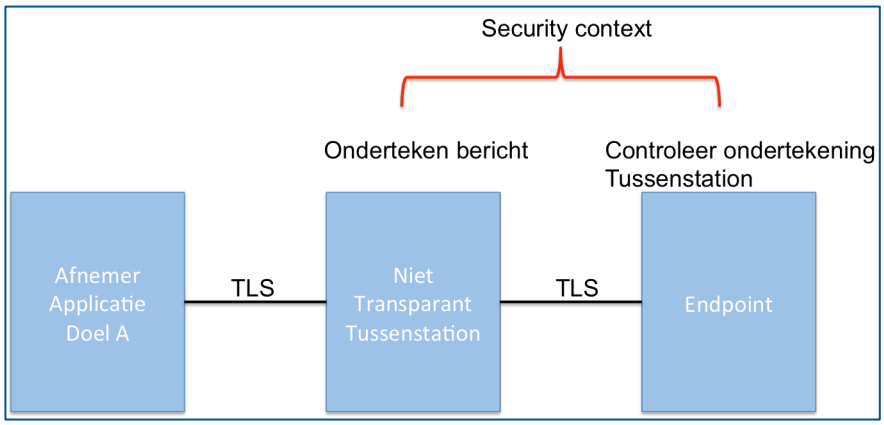
Bij het aansluiten van een afnemer zal duidelijk moeten worden om welke situatie het gaat.
De 2-zijdige TLS verbinding in onderdeel van de beschrijving van deze situatie. En als er meer dan één tussenstation aanwezig is moet elke verbinding van de afnemer tot en met endpoint voorzien worden van TLS. De verbinding wordt altijd tot stand gebracht met PKI-overheid certificaten.



Afbeelding 10 Afnemer met transparant tussenstation

Kenmerken verbinding met een transparant tussenstation:

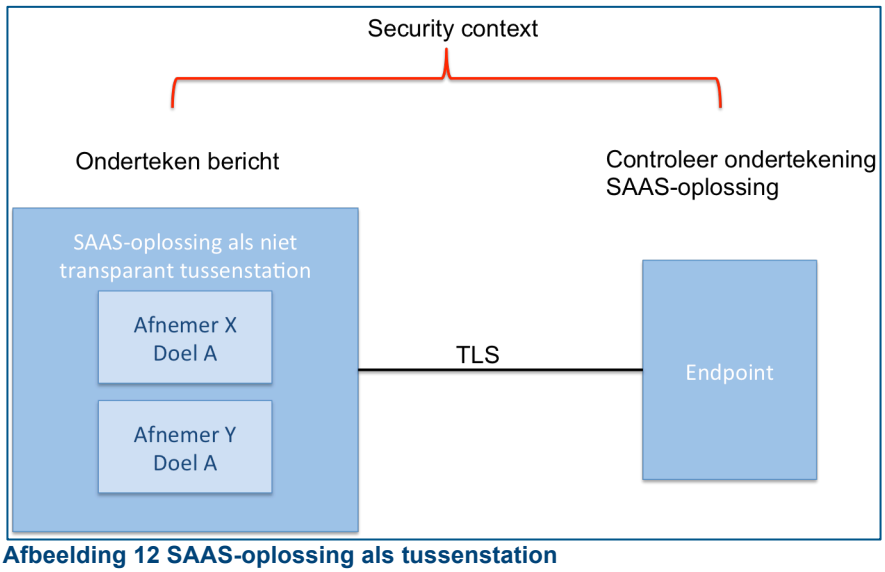
- De applicatie/webservice ondertekend het bericht.
- Elke afzonderlijke afnemer heeft zijn eigen identiteit in de vorm van een sub-OIN.
- Van de afnemer tot aan het endpoint worden berichten uitgewisseld op basis van SuwiML.
- Het transparant tussenstation routeert het bericht naar het endpoint of evetueel naar een vonden transparant tussenstation. Het bericht kan en mag niet aangepast worden onderweg.



Afbeelding 11 Afnemer met niet transparant tussenstation

Kenmerken verbinding met een niet transparant tussenstation:

- Het niet transparante tussenstation ondertekent het bericht namens de applicatie.
  - Als het tussenstation zich binnen de afnemer in één omgeving bevindt dan zijn er geen aanvullende voorwaarden van toepassing.
  - Als het tussenstation als een SAAS-oplossing afgenomen, moeten daar mogelijk aparte aanvullende afspraken gemaakt worden m.b.t. de beveiliging en de ondertekening van berichten.
- Elke afzonderlijke afnemer heeft zijn eigen identiteit in de vorm van een sub-OIN.
- Vanaf het tussenstation naar het endpoint wordt het bericht uitgewisseld op basis van SuwiML.
- Vanaf de applicatie/webservice naar kan een intern protocol gebruikt worden, maar uiteraard ook SuwiML. Het tussenstation kan een SuwiML-adapter genoemd worden. Naast de SuwiML omzetting kan het tussenstation nog andere functionaliteiten bevatten.
- Een niet transparant tussenstation kan ook via één of meer transparante tussenstations het endpoint bevragen.



Afbeelding 12 SAAS-oplossing als tussenstation

Ander dan een de niet transparante tussenstation bevat een SAAS-oplossing naast de webservices om berichten op te vragen ook meer business-functies. Bijvoorbeeld een applicatie die door een marktpartij via een private cloud aan meer dan één afnemer wordt aangeboden. Of denk aan een constructie waarbij meer gemeenten 1 pakket gebruiken.

Kenmerken verbinding met een SAAS-oplossing:

- Het niet transparante tussenstation ondertekent het bericht namens de afnemer.
- Elke afzonderlijke afnemer heeft zijn eigen identiteit in de vorm van een sub-OIN.
- Een SAAS-oplossing kan ook via één of meer transparante tussenstations het endpoint bevragen.

5.4. Andere stuurgegevens

De SuwiML Transactiestandaard zal zich in dat geval conformeren aan het WS-I Basic Security Profile, zie http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html. Op dit moment is die behoefte nog niet gesignaleerd en wordt volstaan met versleuteling en identificatie op transport-niveau door middel van TLS.

5.5. Valideren van een inkomend Request

Het valideren van een bericht dat binnenkomt bij een adapter (of een broker, of een applicatie die SOAP 'praat') gaat enigszins anders dan voorheen. In de huidige service-georiënteerde opzet moeten de volgende stappen doorlopen worden:

I. Parse het inkomende request
II. Verifieer dat het een SOAP bericht betreft door te valideren tegen het XML-Schema http://schemas.xmlsoap.org/soap/envelope/.
III. Verwerk de informatie uit de WS-Addressing elementen in de header. Hier wordt geen gebruik gemaakt van een schema, waarin vastgelegd is welke elementen moeten/mogen voorkomen. Hiervoor wordt gebruik gemaakt van het WS-Addressing schema zoals is voorgeschreven door W3C. De service moet zelf de implementatierichtlijnen zoals beschreven in 5.2 implementeren.
IV. Neem het eerste element uit de SOAP Body. Verifieer dat de namespace van dat element voorkomt in het lijstje van webservices dat de adapter ondersteunt
V. Verifieer dat de naam van dat element een van de operations is van de desbetreffende webservice.
VI. Valideer het element met alles wat er onder zit tegen het schema dat in de WSDL geïmporteerd wordt
VII. Verwerk de informatie uit de body.

Bij iedere bovengenoemde stap kan het proces fout gaan. Zie Hoofdstuk 6 Foutafhandeling voor de gewenste afhandeling van de diverse foutsituaties.

5.6. Diakrieten, karaktersets en encoderingen

Binnen de berichtenuitwisseling wordt gebruik gemaakt van de UTF-8 encodering en de Unicode karakterset. Uiteraard gelden hierbij dat de door SOAP gereserveerde karakters conform de standaard ge-escape worden

Afspraak 14: Er gelden geen beperkingen aan de te gebruiken karakters anders dan dat ze tot de Unicode karakterset moeten behoren.

Afspraak 15: Bij het versturen van een Bericht dient de UTF-8 encodering gebruikt te worden.

5.7. Berichten met binaire bestanden

Zoals besproken in 4.9 Ondersteuning voor binaire bestanden dienen binaire bestanden in een bericht vervoerd te worden met behulp van elementen van het type “xmime:base64Binary”. Base64 conversie is een oude en zeer bekende techniek die in alle programmeertalen geïmplementeerd is. Met Base64 kun je iedere binaire file representeren als een string met daarin alleen maar 'veilige' karakters.

Zie bijvoorbeeld het volgende willekeurige met Paint gemaakte plaatje:  . De Base64 string voor dit plaatje is als volgt:

/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEB
AQEBAQEBAQEBAQEBAQEBAQEBAQH/2wBDAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEAQEBAQEBAQEBA
QEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/wAARCAAWACcDASIAAhEBAxEB/8QAGQABAAMBAQAAAAAAAAA
AAAAAAAclCQYK/8QAJRAAAgICAQMFAQEBAAAAAAAABgcFCAQJAwIVFgABChcYExQS/8QAFAEBAAAAAAAAAAAAAAAAAA
AAAP/EABQRAQAAAAAAAAAAAAAAAAAAAD/2gAMAwEAAhEDEQA/ALdlhBPXQ+znW3Xrt/gKirXSDaGwjqJmgNaVItpn1Try/Vh
YuANMQyCUhVhcH6LWVPbFyBR0sSKB2SuSNTLPFwTtRcJnM4xvwDZ2+PyXUVlimqz/ANv8qdVPYFN4lty9ncH49KZrjEB1rXYFC
wFPZb0jU2RYXKv3iss8il0nZddQjaRMRizy9ehdny6FzeSbl6ftDSZ5PadMatMhF1Aa9mhDSCUOWYZWHP8A4qRK6e04xVnV0Idw5
KVvqYU2atIwFfNLPMPDXouKzz4OsIYtY/NBgSqjC9HGo5mgGzrWlWequxKvSkm601AqqP8AL+E2Ebobn2i2EIkmyUSzLRPpQPQ
wLnvYSr6/Zov/AFzYxPDTFylOQc5jX5RYBE9Q1OvEf63QZ1jDV9Tm83cp81BsLKTvyHrYIlxd9g1TsMt2W96j0DXuNZ9U627lyLFH6
9VrRVu62m1UnAM42Jgk0375h8Kj/us8h4LQ6ZrOaxjJj542Ps6zauQV66dUyH9r8CxCtP2HsPagJtyx9RDZUgdVXDCwnEs17sSzkH1
WVts+nAsq7M1ArUJUZxa3MtcB14YNdDbjKJBdwSlHT7KDOrRWcvBABJDr01gL6uTWv+9ozKRUlvAsJmAiu+r5jaEPyFg5kalVXkgo
T5shcaqyoqhcQuBmR5JMIasBUxU+wwy4rrUnTMEJqGC2lT26tvgTWfWCGr5XhP7XspfWMjtqD9Yoq9ZEPebP5ka66zQYUsizJOB+
iwSijBRh8U8MBKit0o47DmyWyHBye+dkQob/ACe1+WRutdB44drb27nqxORS0goLjEBCAuii1Sypw85c1tgPFLTOaHUA4XUqfdPwx
SHkecmxk9OiRygciWtCLKSwtG88MmOt6zg2CQT6T9v3FVyPfS/132asBUCoJSsrHNL5EGyfLBE+CLSwFy5c6NJ1lWCWy9+8/NP+
MVGRleJXu0wn5hoR7zSgoTQpJaQvT70G31sdQVBav6XrLDn5HqAcO2u+sByQn6X/ACclxlpmDTUlUiPA+9O/dhIC0fYBAWj/AJ/3fz
Wbl4ojy/8AZ5TJSeP3bmgCQr0064PC0Is7m2v382i9gfGGYR+9A1MkaWl2yRy+7V3wmYYNmII3e9+/sDk8c6IiNyhKbA1YAKyJWiLT
SdVylUYULYj09BL9vaDv3gNKTV6IW3SBx18bm1+5DrBVfYrXsaOOOHCp/ILaVb7Mj3Nh5144IJeg+DZJsYjIhiw4Cm5HFKsZcsvller
CJQQNMgaMapX6LnEhR5UNGkAGm2pSDdnTLFt700mNOu4T0FQG7Vba2Wl5rJCQ3aETFuogAmtKZEbSQ643l0Y6FRkTke7EV
EMSGPWLhr09B6HtcdYPybfq8668aqAK96qBroNe30pqH+LlZzdxdGzeC/2z6t+6nv39gcnjn8JU/8ALo7ug5jig745h+K9zmHp6eg//9k

Afbeelding 13 Voorbeeld van een Base64 string

Een dergelijke Base64 string kan in een XML bericht worden opgenomen omdat er geen teken-combinaties in kunnen voorkomen die de XML structuur in de war kunnen sturen. Een bericht ziet er dan als volgt uit:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
   <SOAP-ENV:Header>
     ...
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      <smls:Upload xmlns:smls="http://bkwi.nl/SuwiML/Diensten/NaamService">
         <Burgerservicenr>000000000</Burgerservicenr>
         <Foto
xmime:contentType="image/jpeg">/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQ
EBAQEBAQEBAQEBAQEBAQEB
AQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/2wBDAQEBAQEBAQEBAQEBAQEBAQEBAQEB
AQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/wAARCAAWACcDASIA...
hR5UNGkAGm2pSDdnTLFt700mNOu4T0FQG7Vba2Wl5rJCQ3aETFuogAmtKZEbSQ643I0Y6FRkTke7
EVEMSGPWLhr09B6HtcdYPybfq8668aqAK96qBroNe30pqH+LlZzdxdGzeC/2z6t+6nv39gcnjn8J
U/8ALo7ug5jig745h+K9zmHp6eg//9k=</Foto>
      </smls:Upload>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Afbeelding 14 Voorbeeld van een Upload bericht met een binair bestand

Merk op dat ook in het bericht (net als in het XML Schema met het attribuut xmime:expectedContentTypes met het attribuut xmime:contentType="image/jpeg" aangegeven wordt wat het type is van het meegeleverde binaire bestand.

Een aantal gangbare webservices-toolkits, waaronder Axis2 en .Net, heeft de mogelijkheid om voor berichten met binaire bestanden gebruik te maken van de techniek MTOM (Message Transmission Optimization Mechanism). MTOM maakt gebruik van MIME, bekend van attachments van email. Wanneer MTOM gebruikt wordt, dan komt er, in de plaats van het element dat de base64Binary string bevat, een verwijzing-element

```
<xop:Include href="cid:1.A91D6D2E3D7AC4D580@apache.org"
      xmlns:xop="http://www.w3.org/2004/08/xop/include">
```

in het Bericht. Het Bericht is dan een Multipart MIME Bericht geworden. In het eerste MIME part zit de SOAP Envelope, met de XML payload en daarin een of meer <xop:Include ...> elementen. Het attribuut href="..." van een <xop:Include ...> element verwijst dan naar een ander MIME part waar het desbetreffende binaire bestand in zit. Ter illustratie, de MTOM-versie van het voorbeeld van Afbeelding 15 is als volgt:

```
Content-Type: Multipart/Related; type="application/xop+xml"; boundary="----
=_Part_0_1744155.1118953559416"
Content-Length: 3453
SOAPAction: ""

------=_Part_1_4558657.1118953559446
Content-Type: application/xop+xml; type="text/xml"; charset=utf-8

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xmime="http://www.w3.org/2005/05/xmlmime" xmlns:xop="http://www.w3.org/2004/08/xop/include">
   <SOAP-ENV:Header>
      …
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      <smls:Upload xmlns:smls="http://bkwi.nl/SuwiML/Diensten/NaamService">
         <Burgerservicenr>000000000</Burgerservicenr>
         <Foto>
            <xop:Include href="cid:5aeaa450-17f0-4484-b845-a8480c363444@example.org" />
         </Foto>
      </smls:Upload>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

------=_Part_1_4558657.1118953559446
Content-Type: image/jpeg
Content-ID: <5aeaa450-17f0-4484-b845-a8480c363444@example.org>

   ... binary data ...

   ------=_Part_1_4558657.1118953559446
```

Afbeelding 15 Bericht met een binair bestand in MTOM formaat

Het gebruik van MTOM heeft voordelen zoals een betere scheiding tussen XML payload en bijgeleverde bestanden, minder belasting voor de XML parser, en beter zicht op de grootte van de afzonderlijke bijgeleverde bestanden.

Maar, het eventuele gebruik van MTOM is een ***keuze*** in de implementaties. In de koppelvlak specificaties van een webservice wordt er geen uitsluitsel over gegeven.

6. Foutafhandeling

Een webservice moet conform de koppelvlak specificaties reageren op inkomende requests. In de WSDL is ook ruimte voor het specificeren van voor de hand liggende foutsituaties. Fouten kunnen optreden op verschillende niveaus, grofweg overeenkomend met de lagen uit Afbeelding 2. Enkele veel voorkomende fout-situaties worden onder besproken. Er wordt getoond hoe een Axis2 implementatie in die gevallen zal reageren. De precieze reactie bij andere implementaties zal net iets anders zijn, maar er wel op lijken.

Het is raadzaam dat de besproken situaties ook in het test-traject van een webservice applicatie (client of server) aan bod komen. Er dient gecontroleerd te worden of er in deze situaties door de applicatie het gewenste gedrag vertoond wordt. En door het te testen zullen dergelijke situaties sneller herkend worden wanneer ze onverhoopt in de Productie-omgeving optreden.

6.1. Foutafhandeling in de WSDL

Wanneer er ten tijde van het opstellen van de specificaties voor een webservice bepaalde uitzondering-situaties onderkend worden die een aparte afhandeling vergen, dan kan dat in de WSDL vermeld worden.

In de WSDL is een generieke Fault 'AanvraagNietOk' opgenomen, maar indien gewenst kan daar ook een specifiekere fout-situatie naast vermeld worden. De partijen die de webservice implementeren dienen ook de in de WSDL vermelde Fout-situaties te implementeren, zowel in de Client-applicatie (hoe reageert de applicatie als er zo'n Foutmelding terug komt) als in de Server applicatie (in de bedoelde uitzondering-situatie dient de gespecificeerde Foutmelding terug gestuurd te worden).

```
<wsdl:definitions ... xmlns:fwi="http://bkwi.nl/SuwiML/FWI/v0203" ...>
   <wsdl:types>
      <xs:schema targetNamespace="http://bkwi.nl/SuwiML/FWI/v0203">
         <xs:include schemaLocation="../../FWI/v0203/FWI.xsd"/>
      </xs:schema>
      ...
   </wsdl:types>
   ...
   <wsdl:message name="AanvraagNietOk">
      <wsdl:part name="parameters" element="fwi:Fout"/>
   </wsdl:message>
   <wsdl:portType name="TestInfo">
      <wsdl:operation name="AanvraagInfo">
         ...
         <wsdl:fault name="AanvraagNietOk" message="smls:AanvraagNietOk"
          wsaw:Action="http://bkwi.nl/SuwiML/Diensten/NaamService/Fout"/>
      </wsdl:operation>
      ...
   </wsdl:portType>
   <wsdl:binding name="TestBinding" type="smls:TestInfo">
      ...
      <wsdl:operation name="AanvraagInfo">
         ...
         <wsdl:fault name="AanvraagNietOk">
            <soap:fault name="AanvraagNietOk" use="literal"/>
         </wsdl:fault>
      </wsdl:operation>
      ...
   </wsdl:binding>
</wsdl:definitions>
```

Afbeelding 16 Specificatie van de Foutmelding 'AanvraagNietOk' in de WSDL

In de <wsdl:message>'s voor de Foutmeldingen wordt verwezen naar het Element <fwi:Fout>. Dat Element wordt gespecificeerd in het FWI.xsd schema, zie http://bkwi.nl/SuwiML/FWI. Met het FWI.xsd schema maken we hergebruik van een vaste structuur voor SuwiML Foutmeldingen. FWI staat voor 'Fouten, Waarschuwingen en andere Informatie'. Het element <fwi:Fout> bevat sub-elementen 'Code', 'Tekst', 'Detail' en 'Bron'.

Er wordt door de Server-applicatie een AanvraagNietOk Foutmelding gegenereerd om meer gedetailleerde informatie omtrent de precieze fout-situatie in een SOAP Fault te kunnen terugkoppelen.

6.2. Fouten in de HTTP Headers

Als alles goed gaat dan zien de HTTP Headers van het Response er als volgt uit:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Mon, 20 Oct 2008 08:26:11 GMT

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   ...
</soapenv:Envelope>
```

Afbeelding 17 Een succesvolle HTTP Response

De HTTP Code 200 duidt op een succesvolle HTTP dialoog. Als er een andere HTTP Code terug gestuurd wordt dan is er iets speciaal aan de hand.
Als de hostnaam van het verzoek fout is, dan zal het verzoek niet bij de bedoelde webserver uitkomen.

```
POST /axis2/services/NaamService HTTP/1.1
Content-Type: text/xml
User-Agent: XML Spy
SOAPAction: ""
Host: foutehost.nl
```

Afbeelding 18 Verkeerde hostnaam in het HTTP Request

Het verzoek kan dan niet getransporteerd worden en in de software van de client zal een “UnknownHostException: foutehost.nl” optreden.

Als de hostnaam wel goed is, maar de server staat uit of heeft het te druk, dan zal er in de software van de client een “ConnectException: Connection timed out” optreden. Een dergelijke time- out verschijnt ook wanneer de hostnaam naar een verkeerd IP-Adres vertaald wordt.

Als de hostnaam van het verzoek goed is maar het pad van de webservice is fout (in de eerste regel van de HTTP Header van het Request staat dan bijvoorbeeld “POST /foutpad/NaamService HTTP/1.1” dan zal de webserver een HTTP Code 404 terugsturen: “HTTP/1.1 404 Not Found”, eventueel samen met een HTML pagina met een toelichting.

6.3. Foutmeldingen van Digikoppeling

I. Gebruik de technische foutmeldingen en categorisering volgens de lijst Digikoppeling.

II. De lijst met technische fouten van Digikoppeling worden verplicht overgenomen door elke leverende service en/of intermediair.

III. Elimineer doublures in eigen lijsten met foutmeldingen ten opzichte van de lijst van Digikoppeling. Domein- of standaardspecifieke technische foutmeldingen worden met een verwijzing naar het domein/standaard in de faultcode opgenomen.

IV. Functionele fouten mogen nooit in de SOAPfault worden opgenomen.

V. Zorg voor een juiste categorisering van meldingen, zodat fouten op een juiste manier verwerkt kunnen worden.

Verschillende fout-categorieën zijn:

1. syntax fouten, zowel op structuur als op inhoudelijke waarden cq. domein check
2. inhoudelijke fouten, waaronder ook protocol fouten, leidend tot onbestaanbare en/of inconsistente situaties
3. fouten doordat een service niet (onvoldoende QoS) beschikbaar is, waaronder ook time-out en autorisatie problemen/fouten.

Per categorie kan op hoofdlijnen een procedure voor de foutafhandeling gedefinieerd worden.

1. bij syntax fouten dient zo mogelijk aangegeven te worden welk element fout is (zoals in foutmeldingen 0005 t/m 0008 aangegeven staat)
2. Zo mogelijk aangeven waarom. Bij protocol fouten aangeven wat de status is en wat wel verwacht wordt, bij inhoudelijke fouten aangeven dat het bericht vanwege inconsistentie niet verwerkt kan worden. (dit is eigenlijk geen transport/koppelvlak probleem , maar veeleer een business probleem met een bijbehorende afhandelingsprocedure, vgl. de terugmelding in het stelsel).

VI. Houd rekening met hoe meldingen bij gebruikers terecht komen. De faultstring kan gebruikt worden om meldingen rechtstreeks aan eindgebruikers door te geven.

VII. Technische foutmeldingen moeten eenduidig worden geïnterpreteerd.

6.3.1 Afspraken voor vulling van SOAPfault elementen

Vul bij technische fouten de SOAPfault conform onderstaande tabel. Gebaseerd op SOAPfault 1.1 van W3C http://www.w3.org/TR/2000/NOTE-SOAP-20000508/ - _Toc478383507.

Element	Standaardisatie afspraak
faultcode	• Bevat de defaultwaarden VersionMismatch, MustUnderstand, Client en Server. • De specifieke foutcode die hoort bij de technische fout (format: <afkortingbron><codering van fout>. • De korte omschrijving van de fout (bijvoorbeeld: de Digikoppeling omschrijving uit de lijst)
faultstring	• De eigen meer gedetailleerde beschrijving van de foutsituatie. • De eigen beschrijving zoveel mogelijk geschikt maken voor het kunnen presenteren aan gebruiker. • De ontvanger is niet verplicht deze tekst over te nemen.
faultactor	• Bevat een URI van de antwoordende service. • Vul de faultactor in met de URI van de bron van de oorzaak, indien het SOAP-bericht langs een tussenstation gaat.
faultdetail	• Alleen detailinformatie opnemen als de technische fout betrekking heeft op de body van het bericht.

Foutcodes
Lijst van technische foutmeldingen met classificatie naar fout-categorieën

Code	Omschrijving	Categorie	toelichting
0001	Invalide soap envelope	1	Voldoet niet aan verwachte syntax. Structuur van de envelope matcht niet met wat er verwacht wordt
0002	Niet geautoriseerd	3	Service niet beschikbaar (QoS). Door gebrek aan bevoegdheden.

0003	Invalide soapaction	1	De inhoud leidt niet tot een voltooide actie, is niet gedefinieerd of onbegrijpelijk. Protocol fout
0004	Niet conform xsd	1	Voldoet niet aan verwachte syntax
0005	WS-Addressing header “to” ontbreekt	1	Ontbreekt of voldoet niet aan verwachte syntax
0006	WS-Addressing header “action” ontbreekt	1	Ontbreekt of voldoet niet aan verwachte syntax
0007	WS-Addressing header “messageID” ontbreekt	1	Ontbreekt of voldoet niet aan verwachte syntax
0008	WS-Addressing header “relates to” ontbreekt	1	Ontbreekt of voldoet niet aan verwachte syntax
0009	Niet volgens UTF	1	Voldoet niet aan verwachte characterset
0010	Headers anders dan WSA-headers	1	Voldoet niet aan verwachte syntax
0011	Header andere waarde dan voorgeschreven	1	Voldoet niet aan verwachte spec/waarde
0051	Service niet beschikbaar	3	Service niet beschikbaar (QoS). Door gebrek aan resources/ verwerkingscapaciteit.

Categorieën
Bij gegevensuitwisseling kunnen er fouten optreden door verschillende oorzaken. Fouten kunnen in één van de volgende categorieën ingedeeld worden:

1. syntax fouten, hebben betrekking op de structuur van de berichten (XSD) en standaarden zoals WSA en SOAP
2. inhoudelijke fouten, hebben betrekking op inhoudelijke verwerking en zijn context/domein of sector specifiek en worden niet binnen DK gestandaardiseerd
3. protocolfouten, hebben betrekking op TLS of HTTP
   fouten doordat een service niet (onvoldoende QoS) beschikbaar is, waaronder ook time-out en autorisatie problemen/fouten.

Per categorie kan op hoofdlijnen een procedure voor de foutafhandeling gedefinieerd worden.

1. bij syntax fouten dient zo mogelijk aangegeven te worden welk element fout is (zoals in foutmeldingen 0005 t/m 0008 aangegeven staat)
2. Zo mogelijk aangeven waarom. Bij inhoudelijke fouten aangeven dat het bericht vanwege inconsistentie niet verwerkt kan worden. (dit is eigenlijk geen transport/koppelvlak probleem, maar veeleer een business probleem met een bijbehorende afhandelingsprocedure, vgl. de terugmelding in het stelsel).

7. Logging

De beheerders van een systeem hebben taken op het gebied van monitoring, voorkomen en herstellen van incidenten, en het signaleren van problemen. Niet alleen voor hun eigen partij, maar ook voor andere partijen waarmee zij elektronisch informatie uitwisselen. Het vastleggen van informatie rondom de uitwisselingen maakt het mogelijk dat zij aan diagnose en foutherstel kunnen werken. Bovendien dienen er rapportages gemaakt te kunnen worden op basis van de gelogde informatie. De gelogde informatie dient ook een bepaalde tijd bewaard te worden zodat het verloop van de informatie-uitwisseling achterhaald kan worden wanneer er later een vermoeden bestaat van misbruik.

Onderscheid moet worden gemaakt tussen het loggen tijdens de gegevensuitwisseling en het loggen tijdens de gegevensverwerking. Dit Hoofdstuk heeft betrekking op het loggen tijdens de gegevensuitwisseling. Het loggen tijdens de gegevensverwerking is een zaak van de afzonderlijke partijen zelf.

7.1. Logging ten behoeve van diagnostiek en foutherstel

Voor eventuele diagnostiek en foutherstel is het van belang dat achteraf in de logging van een systeem door beheerders nagekeken kan worden hoe het berichtenverkeer verlopen is. Het loggen kan echter op meer of minder uitgebreide manier gebeuren. Afzonderlijke onderdelen die minimaal al dan niet in combinatie met elkaar gelogd zouden kunnen worden zijn de volgende:

- De HTTP statuscode
- De URL waar een bericht naar toe gestuurd werd
- Het IP-Adres waar een bericht vandaan kwam
- De identificerende gegevens van het PKI-Overheid certificaat (serienummer) a.g.v. de TLS-verbinding (het OIN)
- De identificerende gegevens van het PKI-Overheid certificaat (serienummer) a.g.v. de signing (het OIN)
- De identiteit (OIN of sub-OIN) van het From-element WS-Addressing
- De (overige) inhoud van de WSA-elementen: MessageId, Action, To, From, RelatesTo
- De inhoud van de SOAP Body voor zover het de zoeksleutels betreft van het Request
- Datum en tijdstip van ontvangst van een bericht
- Datum en tijdstip van verzenden van een bericht
- Datum en tijdstip van ondertekening
- Het bereiken van een time-out bij het wachten op een Response

Al te enthousiaste grootschalige logging vormt een bedreiging voor de schaalbaarheid van een systeem. Het is daarom ook wenselijk dat de koppelvlak specificaties zo zijn opgesteld dat er niet al te zware berichten verstuurd worden, en dat eventuele binaire bestanden met behulp van MTOM in een apart MIME attachment meegestuurd worden, zodat de binaire bestanden zelf niet gelogd hoeven te worden.

Maar, in ieder geval is het van belang dat de logging van alle partijen samen zo is ingericht dat, indien nodig, in gezamenlijk overleg tussen de Beheerders van de verschillende bij een bepaald bericht betrokken partijen, met de Logs kan worden achterhaald wat het traject, end-to-end, van het Bericht geweest is en wat er onderweg allemaal mee gebeurd is.

Omdat veel SuwiML berichten vertrouwelijke persoonsgegevens van burgers bevatten is voor de “inhoud van de SOAP Body” de Wet Bescherming Persoonsgegevens[1] (WBP) van toepassing. De WBP stelt onder andere dat:

- persoonsgegevens gedeeld mogen worden als dat vereist is op grond van een wettelijk voorschrift, in het kader van een publiekrechtelijke taak of als er daarvoor een gerechtvaardigd belang is
- persoonsgegevens alleen verwerkt mogen worden voor welbepaalde, uitdrukkelijk omschreven en gerechtvaardigde doeleinden
- persoonsgegevens niet langer bewaard mogen worden dan noodzakelijk is voor de doeleinden waarvoor zij zijn verzameld of worden gebruikt

Voor de logging van de inhoud van de SOAP Body maken we onderscheid tussen

- Sleutelwaardes (zoals een BSN) in een Bericht
- Vertrouwelijke informatie (zoals persoonsgegevens van klanten / burgers)
- Niet-vertrouwelijke informatie (zoals de inhoud van een ontvangstbevestiging)

In verband met de WBP dienen systemen die 'slechts' tot doel hebben om informatie dóór te sturen (zoals een Broker of een SOAP Adapter), of om overzichten te tonen (zoals Suwinet Inkijk), voor wat betreft het loggen van vertrouwelijke informatie in de berichten een veel kortere bewaartermijn te hanteren dan voor stuurinformatie, sleutelwaardes en andere niet-vertrouwelijke informatie.

Bovenstaande overwegingen leiden tot de volgende afspraken:

Afspraak 16: Alle partijen loggen zoeksleutels, de inhoud van de stuurinformatie, het tijdstip, en het adres waar een Bericht naar toe gaat of vandaan komt.

Afspraak 17: Stuurinformatie, sleutelwaardes en andere niet-vertrouwelijke informatie wordt tenminste bewaard volgens de gangbare wet en regelgeving.

7.2. Management Informatie

Naast het bieden van mogelijkheden voor diagnostiek en foutherstel dient iedere partij ook op verzoek Management Informatie te kunnen leveren. In de diverse Service Level Agreements worden daarover afspraken gemaakt. Het betreft het kunnen leveren van aantallen berichten, fouten die optraden, Time-Outs, Response-tijden en Down-tijden. Bovendien zal op gezette tijden uitgerekend moeten worden in hoeverre door het systeem aan de afspraken in de SLA's voldaan is.

[1] Met ingang van 25 mei 2018 zal de WBP (Wet Bescherming Persoonsgegevens) worden vervangen door AVG (Algemene Verordening Gegevensbescherming)

8. Afsluiting

8.1. Groeipad

Het is wenselijk dat de SuwiML Transactiestandaard doorgroeit naar de relevante nationale en internationale standaarden. Er worden nu ook al wenselijke vervolgstappen gezien die in volgende versies van de SuwiML Transactiestandaard beschreven zullen worden. Mogelijke vervolgstappen kunnen zijn:

- Invoeren ondersteuning voor de betrouwbaarheid in de vorm van Reliability
- Asynchrone bericht-uitwisseling
- Eventuele REST-full webservices

Of de rol van de SuwiML Transactiestandaard geheel overgenomen zal worden door bijvoorbeeld de standaarden van Digikoppeling, of door een set van WS-* standaarden, of door een WS-I Basic Profile, valt nog te bezien. De SuwiML Transactiestandaard zal de komende jaren nog wel van waarde blijven, om de afhankelijkheid van externe ontwikkelingen te beperken, om een eigen tempo te kunnen bepalen, en om de mogelijkheid te behouden eigen keuzes in de Suwi keten te maken.

Bijlage 1 Organisatie Identificatienumer (OIN)

Informatie geciteerd uit de notities van Logius m.b.t. het nieuwe OIN-beleid (d.d. 23 juni 2016).

Functies OIN

Functie	Definitie	Toelichting functie OIN
Identificatie (Identificeren)	Het bekend maken van de identiteit van personen, organisaties of IT-voorzieningen. (Bron: *NORA 3.0 Principes voor samenwerking en dienstverlening)*	Het OIN-(sub)nummer is het identificerende nummer voor organisaties t.b.v. digitaal verkeer met de overheid.
Authenticatie (Authenticeren)	Het aantonen dat degene die zich identificeert ook daadwerkelijk degene is die zich als zodanig voorgeeft: ben je het ook echt? Authenticatie noemt men ook wel verificatie van de identiteit. (Bron: *NORA 3.0 Principes voor samenwerking en dienstverlening)*	Het OIN-(sub)nummer wordt opgenomen in het subject serial number veld van het PKIoverheid certificaat.
Autorisatie (Autoriseren)	Het proces van het toekennen van rechten voor de toegang tot geautomatiseerde functies en/of gegevens in ICT voorzieningen.	Het feit dat een organisatie over een OIN-(sub)nummer beschikt zegt niets over enige autorisatie op gegevens of informatie. Dit is voorbehouden aan de verstrekkende partij die dit zelf beoordeelt. Partijen die voorzieningen aanbieden onderhouden zelf autorisatie lijsten waarin het OIN van geautoriseerde organisaties kan worden opgenomen.
Adresseren	Het aangeven van de ontvangende partij (en de verzendende partij) in het bericht.	Digikoppeling schrijft b.v. voor dat het OIN (sub)nummer wordt gebruikt in de header voor adressering.
Routeren	Het doorsturen van een bericht aan de geadresseerde partij bijvoorbeeld via een routeringsregel of tabel.	Routering vertaalt het logische adres – het OIN (sub)nummer – naar een fysiek endpoint (url).

Implementatie van het nieuwe OIN-beleid heeft de volgende eigenschappen:

- Er is meer flexibiliteit door het introduceren van subOIN's.
- De verantwoordelijkheid voor subOIN's wordt expliciet bij de rechtspersoon belegd zodat de verantwoordelijkheid om bij organisatiewijzigingen te muteren of te verwijderen zo dicht mogelijk bij de bron ligt.
- Het aanmaakproces van subOIN's wordt verder geautomatiseerd.

- De identiteit van de subOIN-houder en die van de verantwoordelijke rechtspersoon worden via de COR publiekelijk ontsloten door een portal en webservice.
- Rechtspersonen ingeschreven in het Handelsregister beschikken automatisch over een OIN.
- Er is geen dubbele registratie door koppeling aan het Handelsregister.
- De nummersystematiek voor OIN wijzigt niet.
- De reeds uitgegeven OIN's en subOIN's blijven geldig.
- De toepassing van het OIN in het berichtenverkeer wijzigt niet.

Organisaties en organisatieonderdelen die niet in een aangesloten overheidsregister voorkomen komen mogelijk in aanmerking voor een OIN-(sub)nummer. Deze worden geregistreerd in het OIN-subnummerregister.

De lengte van het OIN is 20 posities, omdat dit wordt opgenomen in het subject serial number field van het PKIoverheid certificaat; dit veld bestaat uit 20 posities.

De OIN-systematiek blijft ongewijzigd. Het OIN is opgebouwd uit de volgende elementen:

Element	Lengte	Waarde
Prefix	8 posities	Zie Prefix tabel
Hoofdnummer	8 of 9 posities	Identificerend nummer[7] uit een register. Als het hoofdnummer een KvK nummer is, is het hoofdnummer 8 posities lang.
Suffix	3 of 4 posities	Als het hoofdnummer 9 posities heeft dan is de suffix 000. Als het hoofdnummer 8 posities heeft dan is de suffix 0000.

Een OIN-subnummer is een betekenisloos nummer dat wordt gegeneerd tijdens de registratie. Betekenisloos houdt in dat het OIN-subnummer zelf geen aanwijsbare relatie heeft met het OIN van de OIN-houder. De relatie is alleen te raadplegen via de COR (Centrale OIN Raadpleegvoorziening).

De keuze voor een betekenisloos nummer is vanwege een aantal redenen:

- Gebruikmaken van een suffix heeft een beperking tot 999 volgnummers. Dit lijkt een voldoende groot aantal maar vanwege de regel dat eenmaal ingetrokken OIN-subnummers nooit hergebruikt mogen worden is het bereiken van deze limiet niet ondenkbaar
- Het is niet onmogelijk dat organisatieonderdelen wijzigen van een juridische verantwoordelijke, of dat een samenwerkingsverband van samenstelling wijzigt. Met een volgnummerconstructie wordt de ontkoppeling van rechtspersoon en OIN-subnummerhouder onmogelijk.)

De prefix verwijst naar het OIN-subnummerregister.

Element	Lengte	Waarde
Prefix	8 posities	00000004
Hoofdnummer	9 posities	Gegenereerd nummer
Suffix	3 posities	000

Een aangesloten overheidsregister krijgt een prefix (per uniek nummer) als het register wordt toegevoegd aan het OIN-stelsel. Dit wordt ook een OIN-subnummer register genoemd. De prefix tabel wordt als aparte lijst beheerd door de Beheerder van de COR en wordt gepubliceerd op de website van Logius.

Prefix	Identificerend nummer	Bron
00000001	RSIN	Handelsregister
00000002	Fi-nummer	Het fiscaal nummer wordt verstrekt door de Belastingdienst aan de organisatie zelf[8] (alleen voor organisaties die niet in het Handelsregister staan).
00000003	KvK nummer	Handelsregister[9]
00000004	OIN-subnummer	OIN-subnummer register
00000005	Nog niet in gebruik	
00000006	Nog niet in gebruik	
00000007	BRIN nummer	De Basisregistratie Instellingen (BRIN) is een register van onderwijsinstellingen dat door DUO wordt beheerd in opdracht van het Ministerie van OCW.
00000008	Buitenlandse nummers	Buitenlandse nummers (registratie via CSP)
00000009	Nog niet in gebruik	
00000099	Test OIN's	Elke organisatie mag een test OIN gebruiken mits voorzien van deze prefix.

Volgorde identificerende nummers

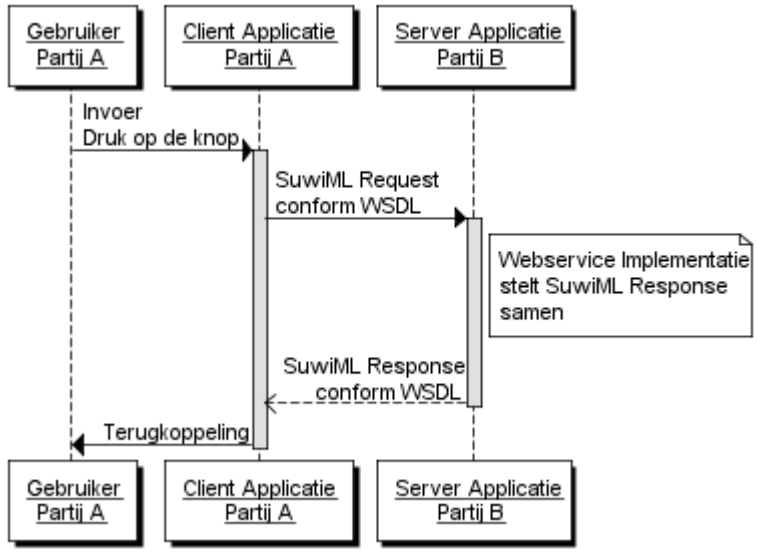
Bij het toekennen van een hoofdnummer wordt de onderstaande volgorde gehanteerd:

- RSIN – identificerend nummer van de rechtspersoon of samenwerkingsverband in het Handelsregister, is identiek aan het fiscale nummer10.
- KvK nummer – het inschrijvingsnummer van de onderneming in het Handelsregister. Bij bedrijfsoverdracht houden ondernemingen niet meer hun KvK-nummer, maar krijgen zij een nieuw KvK- nummer.
- BRIN nummer – het identificerend nummer van onderwijsinstellingen in het BRIN register

Bijlage 2 Scenario's en Sequence diagrammen

In deze bijlage wordt een overzicht geboden van alle Bericht-uitwisseling scenario's die in de Suwi keten voorkomen. De scenario's worden in Sequence Diagrammen weergegeven. In en bij de Sequence Diagrammen worden instructies en aandachtspunten vermeldt die bij het implementeren van de koppelvlak specificaties ter harte genomen dienen te worden. Alle besproken scenario's worden door deze versie van de Transactiestandaard ondersteund.

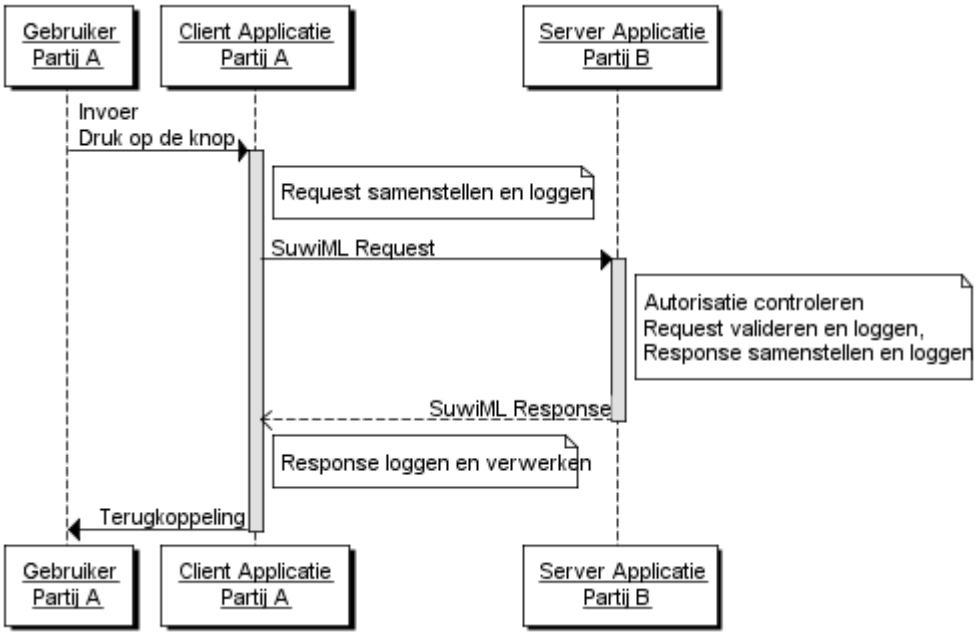
Raadplegingen



Afbeelding 1 Meest eenvoudige SuwiML Request - Response scenario

In de basis bieden SuwiML webservices een eenvoudige methode voor communicatie tussen een Client Applicatie van de ene partij (Partij A) en een Server Applicatie van een andere partij (Partij B), zie Afbeelding 1. Deze Transactiestandaard gaat met name over het koppelvlak tussen de Client Applicatie van Partij A en de Server Applicatie van Partij B.

Bijna alle SuwiML webservices leveren privacy-gevoelige informatie. Voor alle Services waarbij vertrouwelijke informatie getransporteerd wordt zijn er extra maatregelen noodzakelijk op het gebied van Autorisatie en Logging, zie Afbeelding 2. De Client Applicatie en de Server Applicatie dienen beiden daarvoor te zorgen.

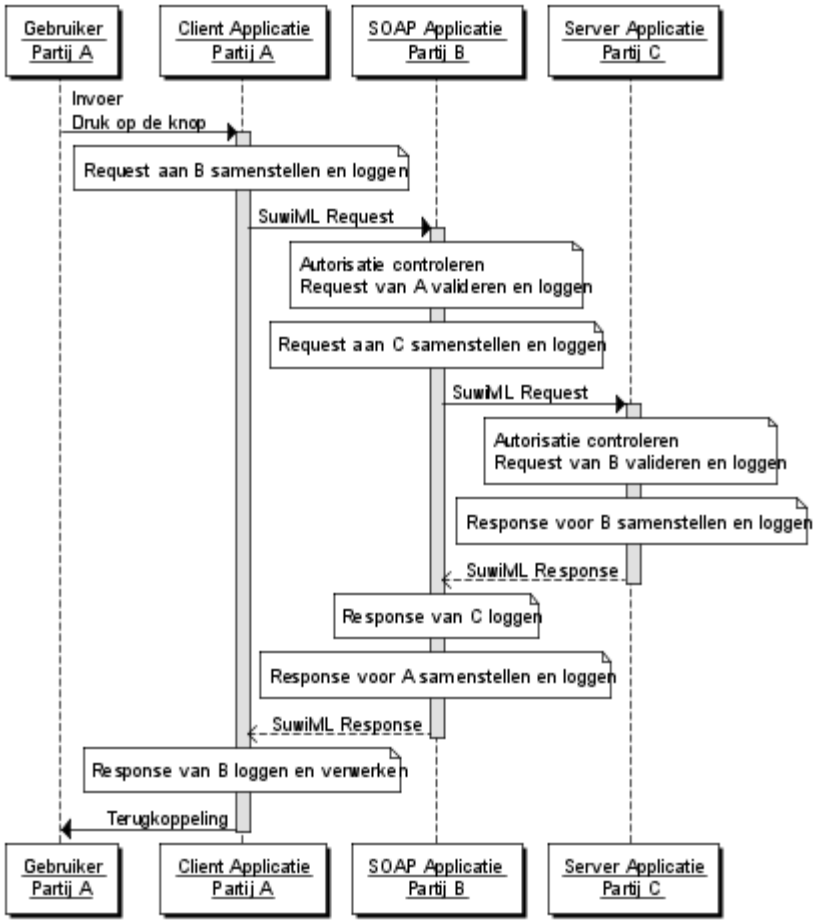


Afbeelding 2 Zorg voor Logging, Validatie, Autorisatie

Het kan zijn dat Partij B voor het samenstellen van het Response informatie nodig heeft van nog andere Partijen. Op basis van het Request van Partij A wordt door Partij B een Request naar Partij C gestuurd, zie Afbeelding 3.

In het scenario van Afbeelding 3 is het vaak zo dat het koppelvlak tussen Partij A en Partij B niet precies hetzelfde is als het koppelvlak tussen Partij B en Partij C. Bekende voorbeelden uit de Suwi praktijk zijn de Suwi Broker en de Broker (Sectorloket) van het Inlichtingenbureau. Beiden spelen de rol van Partij B in Afbeelding 3. De Requests die door Inlezende applicaties naar de Suwi Broker gestuurd worden zijn vragen van de vorm “Geef mij alle info voor dit Burgerservicenummer van alle achterliggende Partijen”. De Requests die door de Suwi Broker naar het Sectorloket van het Inlichtingenbureau gestuurd worden zijn vragen van de vorm “Geef mij alle info voor dit Burgerservicenummer van alle Gemeenten waar deze Persoon bekend is”. En de Requests die door het Sectorloket naar een bepaalde Gemeente gestuurd worden zijn van de vorm “Geef mij alle info van uw Gemeente over de persoon met dit Burgerservicenummer”. Het Response van de Gemeente bevat alleen de informatie van die Gemeente over de persoon in kwestie. Het Response van het Sectorloket bevat alle info van alle Gemeenten waar de persoon bekend. En het Response van de Suwi broker bevat alle info van de verschillende Gemeenten maar ook de info van de andere achterliggende Partijen. De koppelvlakken tussen de verschillende partijen zijn dus verschillend, ook al ziet de vraag er steeds bijna hetzelfde uit.

Wanneer er sprake is van verschillende koppelvlakken, dan dient dit ook in de koppelvlak specificaties tot uiting gebracht te worden. Er dienen aparte WSDL beschrijvingen gemaakt te worden voor de verschillende koppelvlakken.



Afbeelding 3 Een SuwiML Request van Partij A dat door partij B doorgestuurd wordt naar Partij C