



Ministerie van Binnenlandse Zaken en
Koninkrijksrelaties

Best Practice WUS Digikoppeling 2.0

Versie 1.3

Datum 24 maart 2011
Status Definitief

Colofon

Projectnaam	Digikoppeling 2.0
Versienummer	1.3 Definitief
Organisatie	Servicecentrum Logius Postbus 96810 2509 JE Den Haag T 0900 555 4555 servicecentrum@logius.nl
Bijlage(n)	0

Inhoud

Colofon	2
Inhoud	3
Inleiding	4
1.1 <i>Doel en doelgroep</i>	4
1.2 <i>Opbouw Digikoppeling documentatie</i>	4
1.3 <i>Digikoppeling (voorheen OverheidsServiceBus)</i>	4
1.3.1 <i>Doel en scope van Digikoppeling</i>	4
1.3.2 <i>Uitwisseling binnen Digikoppeling</i>	5
1.4 <i>Opbouw van dit document</i>	5
1.5 <i>Gehanteerde terminologie: Digikoppeling Glossary</i>	5
2 Werkwijze/Aanbevelingen/Best Practices	6
Bijlage 1 - Opdeling WSDL in WSDL 1.1	10
Bijlage 2 – Foutmeldingen	11

Inleiding

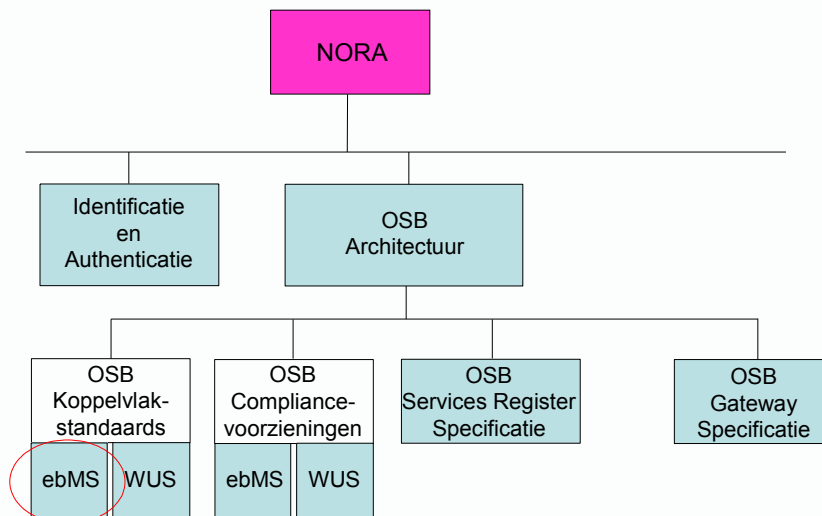
1.1 Doel en doelgroep

Alle Digikoppeling (voorheen OverheidsServiceBus) webservices die op WUS gebaseerd zijn, moeten conformeren aan Digikoppeling Koppelvlakstandaard WUS. Dit document is een aanvulling hierop. Het heeft als doel ontwikkelaars te adviseren en te informeren over de huidige werkwijze bij het toepassen van Digikoppeling Koppelvlakstandaard WUS – deze informatie geldt dus alleen voor de WUS-variant.

Het document is bestemd voor ontwikkelaars van webservices, die zijn aangesloten op Digikoppeling. Het gaat hierbij om zowel service providers als service requesters (clients).

1.2 Opbouw Digikoppeling documentatie

Digikoppeling is beschreven in een set van documenten. Deze set is als volgt opgebouwd:



figuur 1 – Opbouw documentatie Digikoppeling

1.3 Digikoppeling (voorheen OverheidsServiceBus)

1.3.1 Doel en scope van Digikoppeling

Voor de Overheid als geheel is interoperabiliteit tussen een groot aantal serviceaanbieders en serviceafnemers van essentieel belang. Die grootschalige interoperabiliteit wordt bereikt door een sterke standaardisatie van het koppelvlak tussen de communicatiepartners.

Deze communicatie vindt plaats in het domein van Digikoppeling, en daarbij worden Digikoppeling koppelvlakstandaarden toegepast. Dat is een zeer beperkte set van standaarden waaruit onder gedefinieerde omstandigheden gekozen kan worden.

Digikoppeling biedt de mogelijkheid om op die sterk gestandaardiseerde wijze berichten uit te wisselen tussen serviceaanbieders (service providers) en serviceafnemers (service requesters of clients). Digikoppeling richt zich (in elk geval voorlopig) uitsluitend op uitwisselingen tussen overheidsorganisaties.

1.3.2 *Uitwisseling binnen Digikoppeling*

De uitwisseling tussen service providers en - requesters is in drie lagen opgedeeld:

- Inhoud: deze laag bevat afspraken over de inhoud van het uit te wisselen bericht, dus de structuur, semantiek en waardebereiken . Digikoppeling houdt zich niet met de inhoud bezig, "heeft geen boodschap aan de boodschap".
- Logistiek: op deze laag bevinden zich de afspraken betreffende transportprotocollen (HTTP), messaging (SOAP), adressering, beveiliging (authenticatie en encryptie) en betrouwbaarheid. Dit is de laag van Digikoppeling.
- Transport: deze laag verzorgt het daadwerkelijke transport van het bericht.

Digikoppeling richt zich uitsluitend op de logistieke laag. Deze afspraken komen in de koppelvlakstandaarden en andere voorzieningen.

1.4 **Opbouw van dit document**

Hoofdstuk 1 bevat een aantal algemene inleidende onderwerpen. Hoofdstuk 2 bevat aanbevelingen, werkwijzes en Best Practices.

1.5 **Gehanteerde terminologie: Digikoppeling Glossary**

Voor de definities die binnen het Digikoppeling project gehanteerd worden, zie de 'Digikoppeling Glossary' die via Digikoppeling website te vinden is.

2 Werkwijze/Aanbevelingen/Best Practices

Deze paragraaf bevat de aanbevelingen t.a.v. het omgaan met servicedefinities.

Nr	Omschrijving
WB001	<p>In WSDL 1.1 is een Authoring Style advies (zie bijlage 1) opgenomen: "separate the definitions in three documents: data type definitions, abstract definitions, and specific service bindings". Dit advies, met name het apart beschrijven van de "specific service bindings" (WSDL onderdelen Binding en Service) wordt overgenomen. Dit advies is mede gebaseerd op het beoogde gebruik van UDDI.</p>
WB002	<p>Voor de specificatie van zaken die buiten het bereik van de WSDL vallen (TLS en WS-Security) wordt aanbevolen om in de WSDL van een service "documentation elements" (<wsdl:documentation> of <!--xxx -->) op te nemen die de eisen ten aanzien van metadata verwoorden, of een verwijzing naar betreffende documenten bevat.</p>
WB003	<p>Voor communicatie binnen het Digikoppeling WUS kanaal wordt de UCS karakterset (ISO/IEC 10646) gebruikt. Deze karakterset omvat (is superset van) de set Unicode, Latin (ISO/IEC 8859-x) en de GBA karakterset.</p> <p>Bij berichtenverkeer speelt de gebruikte karakterset een belangrijke rol. Een serviceafnemer kan in de vraag aanroep karakters gebruikt hebben die niet door de serviceaanbieder ondersteund worden. Voor goede communicatie is het dus belangrijk dat hiervoor afspraken gelden. UCS is de meest uitgebreide karakterset. Toepassen van UTF-8 zorgt er voor dat efficiënt omgegaan wordt met het aantal bytes in het bericht.</p>
WB004	<p>Versie aanduiding</p> <p>Er zijn een aantal elementen waaraan een versie aanduiding moet worden toegevoegd. Dit zijn: WSDL/namespace WSDL/Service naam WSDL/PortType WSDL/Type(s) (XSD) namespace</p> <p>Er zijn een aantal manieren om de versie van een service aan te duiden. De meest gangbare zijn "Major.Minor", "Enkelvoudige versie" (bijv V1) en "YYYY/MM".</p> <p>Het voorstel is om voor zowel de XSD als de WSDL de Enkelvoudige versie aanduiding te gebruiken.</p> <p>Waarom gebruiken we geen major.minor? Er zijn verschillende mogelijkheden om Minor wijzigingen backward compatible te houden, deze worden echter als erg omslachtig beschouwd en/of ze vereisen speciale tooling ondersteuning. Daarom wordt voorgesteld geen onderscheid tussen major en minor te maken, en dus alleen met enkelvoudige versies te werken. Dit heeft als resultaat dat de WSDL en XSD namespace dus alleen de "Enkelvoudige" aanduiding, zoals _v1 krijgt.</p> <p>Een aantal voorbeelden:</p> <p>WSDL namespace "http://wus.osb.gbo.overheid.nl/wsdl/compliance-v1" XSD namespace "http:// wus.osb.gbo.overheid.nl/xsd/compliance/xsd/compliance-v1". Service naam "OSBComplianceService_v1" PortType "IOSBComplianceService_v1"</p> <p>De aanduiding YYYY/MM slaat ook op een enkelvoudige versie, d.w.z. zonder onderscheid tussen major en minor. Die aanduiding kan dus ook gebruikt worden. Het lijkt echter aan te bevelen om versies van webservices aan te duiden met (oplopende) versie nummers, omdat communicatie daarover iets eenduidiger is.</p>

WB005	<p>Voor het onderscheid tussen test- en productieservices heeft het de voorkeur dat deze twee op aparte machines komen met een eigen DNS naam (en dus met verschillende PKI overheid certificaten).</p> <p>Aan de locatie (uri) van de service is daardoor te zien of het om een productie- of een testservice gaat.</p>
WB006	<p>Gebruik van document/literal wrapped style. In Digikoppeling Koppelvlakstandaard WUS staat de bij voorschrift WW003 dat bij de document literal style de body maar 1 element mag bevatten. Het wordt sterk aangeraden dat dit element de operatie naam bevat voor een bepaald bericht. Deze wordt dus door de xsd beschreven en bevat een beschrijving van de payload. Door deze methode te gebruiken wordt de interoperabiliteit verhoogd, met name tussen Microsoft en andere omgevingen. (zie http://www-128.ibm.com/developerworks/webservices/library/ws-whichwsdl/)</p> <p>Wsd definition</p> <p>...</p> <pre> <types> <schema> <element name="myMethod"> <complexType> <sequence> <element name="x" type="xsd:int"/> <element name="y" type="xsd:float"/> </sequence> </complexType> </element> <element name="myMethodResponse"> <complexType/> </element> </schema> </types> <message name="myMethodRequest"> <part name="parameters" element="myMethod"/> </message> <message name="empty"> <part name="parameters" element="myMethodResponse"/> </message> <portType name="PT"> <operation name="myMethod"> <input message="myMethodRequest"/> <output message="empty"/> </operation> </portType> ... </pre> <p>bericht:</p> <p>...</p> <pre> soap:envelope> <soap:body> <myMethod> <x>5</x> <y>5.0</y> </myMethod> </soap:body> </soap:envelope> ... </pre>

WB007	<p>Met betrekking tot het verkrijgen van eenduidigheid in de WSDL bestanden, wordt sterk aangeraden dat xml namespace prefixen volgens de WSDL 1.1 specificatie gebruikt worden. (http://www.w3.org/TR/wsdl) "1.2 Notational Conventions").</p>
WB008	<p>Naamgeving conventies</p> <p>Over het algemeen moet de service naam een goede weerspiegeling zijn van de context waarin de service wordt gebruikt. De operatienamen die door deze service ondersteund worden, moeten passen binnen de context van de service en overdadige lange tekststrings moet worden voorkomen.</p> <p>Namespace</p> <p>Het heeft de voorkeur dat een namespace wordt opgebouwd op basis van de domeinnaam van de web service.</p>
WB009	<p>Foutafhandeling</p> <p>Bij berichtenuitwisseling tussen een service requester en service provider kunnen fouten optreden. Het is van belang dat er nagedacht wordt over het nut van het communiceren van een bepaalde foutmelding. De beschrijving van een of andere interne fout bij een service provider zal in het algemeen niet interessant zijn voor een requester; het terugmelden van de specifieke oorzaak heeft dan geen zin.</p> <p>Als de oorzaak van de fout bij de service requester ligt, dan dient de foutmelding er op gericht te zijn dat de service requester op basis van de foutmelding de fout kan achterhalen en actie ondernemen.</p> <p>Voor communicatie in het Digikoppeling domein volgens de Koppelvlakstandaard WUS zijn vier verschillende fouttypen te onderkennen: protocolfouten, transportfouten, functionele fouten en technische fouten.</p> <p>Protocol- en transportfouten (dus TLS of HTTP fouten).</p> <p>Protocol- en transportfouten zijn in het algemeen in het protocol gedefinieerd. Wijzigingen daarin - dus aanpassing van standaard software - zijn niet wenselijk. Protocol- en transportfouten worden daarom niet beschreven. De hier beschreven foutmeldingen hebben betrekking op situaties waarin het requestbericht door de web service is ontvangen. Deze kan het echter niet goed verwerken en stuurt daarom een foutmelding terug.</p> <p>Functionele fouten</p> <p>Functionele fouten zijn in het kader van Digikoppeling moeilijk te standaardiseren. Deze zullen voor veel organisaties verschillen en ook het communiceren van de foutmelding zal niet altijd eenduidig zijn. Dit is weliswaar iets wat om aandacht vraagt, maar het valt buiten de scope van Digikoppeling.</p> <p>Technische fouten</p> <p>Voor technische foutmeldingen kan een standaard bericht gedefinieerd worden. In de SOAP specificatie is de SOAP Fault beschreven die je hiervoor goed kunt gebruiken.</p> <p>Communiceren van een fout via een SOAP Fault heeft een aantal voordelen: Uitzonderingen op een consistente manier afgehandeld worden; De SOAPFault wordt beschreven in de SOAP specificatie; De verschillende elementen waaruit een SOAP Fault is opgebouwd biedt de mogelijkheid tot het communiceren van uitgebreide informatie; De FaultCode kan aanduiden of de fout was veroorzaakt door Client of Server.</p> <p>Een aantal nadelen zijn:</p> <p>Soapfaults kunnen geen binding (HTTP) gerelateerde fout communiceren. In dat geval wordt over het onderliggende protocol de fout gecommuniceerd Bij een SOAPFault bericht mag geen additionele data toegevoegd worden</p> <p>Het 'detail' element van de SOAP Fault is bedoeld om applicatie specifieke foutmeldingen te</p>

	<p>communiceren die gerelateerd zijn aan het SOAP 'Body' element. Het moet aanwezig zijn in de SOAP Fault indien de fout werd veroorzaakt door het 'Body' element. Indien er geen 'detail' element in de SOAP Fault aanwezig is, dan betekent dit dat de fout niet is ontstaan bij het verwerken van het 'body' element.</p> <p>Voor een web service in het Digikoppeling domein moeten foutmeldingen gedefinieerd worden. Neem die foutmeldingen op in de publicatie van de service in het Digikoppeling Service Register.</p> <p>Mogelijke technische foutmeldingen worden in bijlage 2 weergegeven. Een aantal hiervan zijn overgenomen uit de StUF3.00 specificatie.</p>
WB010	<p>Voorschrift WA001 van Digikoppeling Koppelvlakstandaard WUS heeft betrekking op WS-Addressing headers. Hier volgen een aantal Best Practices voor de invulling hiervan.</p> <p>MessageID Vulling volgens "UUID@URI" of "GUID@URI" heeft de voorkeur. UUID of GUID volgens http://www.ietf.org/rfc/rfc4122.txt URI is een anyURI volgens http://www.w3.org/2001/XMLSchema De URI kan de domeinnaam zijn van Digikoppeling messagehandler of de web service namespace.</p>

Bijlage 1 - Opdeling WSDL in WSDL 1.1

Onderstaande tekst is een citaat uit de WSDL 1.1 Standaard. Het vormt een door Digikoppeling overgenomen best practice bij het schrijven van duidelijke WSDL's.

Authoring Style (© W3C Note 15 March 2001)

"The use of the import element allows the separation of the different elements of a service definition into independent documents, which can then be imported as needed. This technique helps writing clearer service definitions, by separating the definitions according to their level of abstraction. It also maximizes the ability to reuse service definitions of all kinds. As a result, WSDL documents structured in this way are easier to use and maintain. Example 2 below shows how to use this authoring style to define the service presented in Example 1. Here we separate the definitions in three documents: data type definitions, abstract definitions, and specific service bindings. The use of this mechanism is of course not limited to the definitions explicitly presented in the example, which uses only language elements defined in this specification. Other types of definitions based on additional language extensions can be encoded and reused in a similar fashion. "*

Ofwel:

Wijze van notatie (© W3C Note 15 March 2001)

"Door gebruik van het importelement wordt het mogelijk, de diverse elementen van een servicedefinitie te scheiden in afzonderlijke documenten, die dan naar behoefte geïmporteerd kunnen worden. Met behulp van deze techniek kun je duidelijker servicedefinities schrijven, omdat de definities gescheiden worden al naar gelang hun abstractieniveau. Het vergroot ook het vermogen om allerlei soorten servicedefinities nogmaals te gebruiken. Het gevolg is, dat WSDL documenten die op deze manier gestructureerd zijn, gemakkelijker te gebruiken en te onderhouden zijn. In het 2e voorbeeld hieronder is te zien hoe deze manier van schrijven gebruikt wordt om de service in het 1e voorbeeld te definiëren. Hier scheiden we de definities in drie documenten: gegevenstype definities, abstracte definities en specifieke service bindings. Natuurlijk is het gebruik van dit mechanisme niet beperkt tot de definities uit het voorbeeld, dat alleen taalelementen gebruikt die in deze specificatie gedefinieerd zijn. Andere typen definities, die op meerdere taalextensies gebaseerd zijn, kunnen op dezelfde manier gecodeerd en hergebruikt worden."*

* Het voorbeeld waar deze tekst van de WSDL 1.1 standaard naar verwijst wordt is niet in overeenstemming met het Digikoppeling Koppelvlakstandaard WUS profiel (zie voor details CRF 12). Om een indruk te krijgen van de opdeling (authoring style) van een WSDL wordt verwezen naar Bijlage 1 van Digikoppeling Koppelvlakstandaard WUS 1.1.

Bijlage 2 – Foutmeldingen

Digikoppeling Code	Beschrijving	SOAPFault faultcode	SOAPFault faultstring	SOAPFault detail
0001	Het bericht heeft een invalide envelope namespace (voldoet niet aan de SOAP 1.1 specificatie)	VersionMismatch		
0002	Het requester systeem is niet geautoriseerd voor deze bevraging	Client		
0003	Het requestbericht heeft een SOAPAction die niet voldoet aan wat koppelvlakstandaard WUS voorschrijft	Client		
0004	Element binnen de berichtbody is niet conform de XSD	Client		Details van element(en) die fout veroorzaken
0005	WS-Addressing header "To" ontbreekt	Client		
0006	WS-Addressing header "Action" ontbreekt	Client		
0007	WS-Addressing header "MessageID" ontbreekt	Client		
0008	WS-Addressing header "RelatesTo" ontbreekt	Client		
0007	Binnen het SOAP body element bevinden zich meer dan 1 subelementen	Client		
0008	Bericht is niet volgens document-literal style	Client		
0009	Bericht is niet volgens UTF-8 Karaktercodering.	Client		
0010	Bericht bevat headers anders dan WS-Addressing headers	Client		
0011	Bericht bevat 1 of meerdere WS-Addressing headers die niet voorgeschreven worden door koppelvlakstandaard en bevat een waarde anders dan voorgeschreven	Client		
0050	Proces voor afhandelen bericht	Server		

	geeft fout			
0051	Het antwoordende systeem is niet in staat de bevraging af te handelen binnen de connectie time out.	Server		
0100	Opslaan bericht niet mogelijk	Client /Server		