

OSB versie 1

OSB Koppelvlak Standaard WUS

Datum: 5 september 2007
DocumentVersie: 0.92



Documentbeheer

Documenthistorie

datum	versie	auteur	opmerking
10 januari 2007	0.1	E.Reinhoud	
17 januari 2007	0.2	E.Reinhoud	Uitbreiding en wijziging nav commentaar Paul
23 januari 2007	0.21	E.Reinhoud	Verdere aanvullingen
24 januari 2007	0.3	E.Reinhoud	Wijziging nav commentaar Paul
25 januari 2007	0.4	E.Reinhoud	WS-I BP toegevoegd
28 januari 2007	0.5	E.Reinhoud Paul Schlotter	Document geherstructureerd
28 februari 2007	0.57	E.Reinhoud Paul Schlotter	commentaar Pim van der Eijk, Dirk Temme, OSB team Profiel 2 Verwijderd Bijl. met details uit BP1.1 verwijderd
25 april 2007	0.58	E.Reinhoud	WSDL van compliancevoorziening aangepast Bijlage 2
29 mei 2007	0.91	E. Reinhoud	Commentaar van Kees Boogaards verwerkt. Aanpassingen WSDL in Bijlage 2
8 augustus 2007	0.92	E. Reinhoud	Zie release note Diverse tekstuele aanpassingen.

Goedkeuring

datum	versie	goedgekeurd door	opmerking



Inhoudsopgave

1 Inleiding	4
1.1 Doel en Doelgroep.....	4
1.2 Opbouw OSB documentatie.....	4
1.3 De OverheidsServiceBus OSB.....	4
1.4 Opbouw van dit document.....	6
1.5 Gehanteerde terminologie.....	6
2 Koppelvlakstandaard WUS	7
2.1 Inleiding.....	7
2.2 Gehanteerde standaarden.....	8
2.3 Voorschriften.....	9
2.4 Foutafhandeling.....	15
2.5 Werkwijze/Adviezen/best practices.....	15
2.6 Open aandachtspunten.....	16
3 OSB profielen WUS	18
3.1 WUS1.....	18
4 Voorbeelden	19
Bijlage 1 Opdeling WSDL in WSDL 1.1	24
Bijlage 2: Voorbeeld WSDL	26
2.1: osb-voorbeeld.xsd	26
2.2: osb-voorbeeld-abstract-definition.wsdl	28
2.3: osb-voorbeeld-servicebinding.wsdl	30



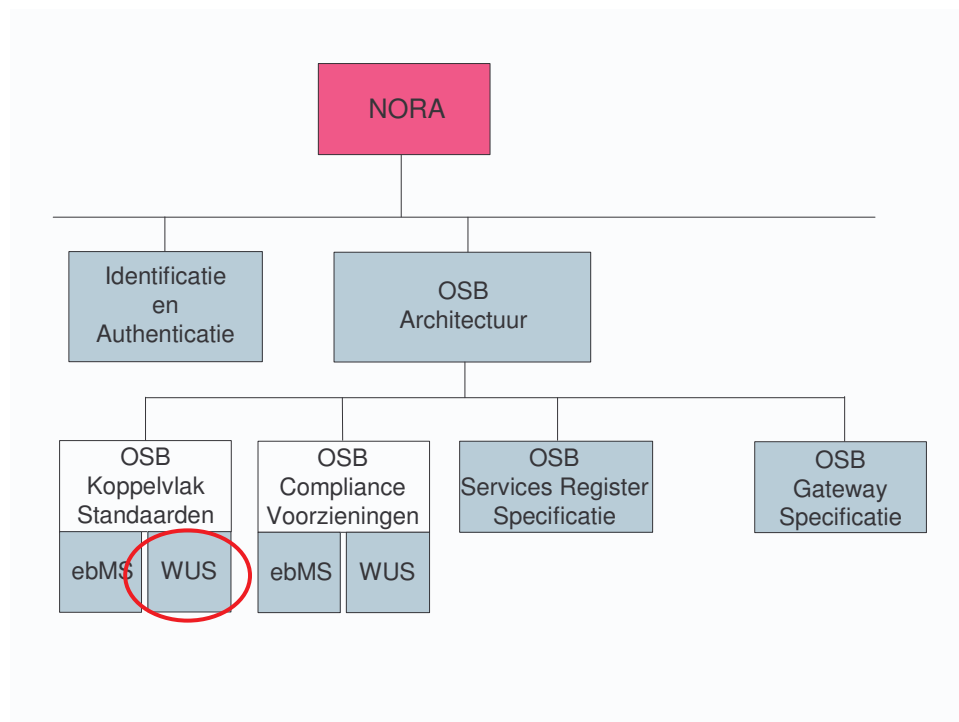
1 Inleiding

1.1 Doel en Doelgroep

Dit document beschrijft de OSB-koppelvlakstandaard gebaseerd op de WUS-familie. Alle WUS-gebaseerde OSB web services dienen te conformeren aan deze koppelvlakstandaard. Het document is bestemd voor ontwikkelaars van webservices, zowel serviceproviders als servicerequesters (clients), die zijn aangesloten¹ op de OSB.

1.2 Opbouw OSB documentatie

De OSB is beschreven in een set van documenten. Deze set is als volgt opgebouwd.



Dit document beschrijft de WUS variant van de Koppelvlakstandaards.

1.3 De OverheidsServiceBus OSB

Deze paragraaf bevat zeer beknopt een aantal hoofdpunten uit de overige documentatie.

Doel en scope van de OSB

¹ Met de term “aangesloten op de OSB” wordt bedoeld dat gewerkt wordt conform het stelsel van afspraken van de OSB.



De Overheidsservicebus biedt de mogelijkheid om op een sterk gestandaardiseerde wijze berichten uit te wisselen tussen serviceaanbieders (Service Providers) en serviceafnemers (Service Requesters).

De OSB richt zich nu uitsluitend op uitwisselingen tussen overheidsorganisaties.

De uitwisseling tussen Service Providers en Requesters wordt in drie lagen opgedeeld:

- **Inhoud:** Op deze laag worden de afspraken gemaakt over de inhoud van het uit te wisselen bericht, dus de structuur, semantiek, waardebereiken etc.
De OSB houdt zich niet met de inhoud bezig, “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.
- **Transport:** deze laag verzorgt het daadwerkelijke transport van het bericht.

De OSB richt zich nu uitsluitend op de Logistieke laag. Deze afspraken landen in de koppelvlakstandaarden en andere voorzieningen.

Leidend principe (requirement)

De koppelvlakstandaarden dienen te leiden tot een maximum aan interoperabiliteit met een minimum aan benodigde ontwikkelinspanning.

Daarom wordt gekozen voor bewezen interoperabele internationale standaarden.

De OSB maakt berichtenuitwisseling mogelijk op basis van de ebXML/ebMS en WUS families van standaarden incl. de daarbij behorende andere standaarden.

De voor de OSB vereiste interoperabiliteit van de WUS standaarden van OASIS en W3C wordt gebaseerd op de profielen (en tests) van WS-I. De interoperabiliteit van ebMS is gebaseerd op de standaard ebMS versie 2 (ISO standaard), en de tests/certificering van Drummond.



1.4 Opbouw van dit document

Hoofdstuk 1 bevat een aantal algemene inleidende onderwerpen.

Hoofdstuk 2 bevat de kern van de standaard. Deze is onderverdeeld naar onderwerpen/gebieden: WSDL, WS-Addressing, naamgeving, beveiliging (TLS). Het identificeert de gekozen internationale (WS-I) profielen die dienen als fundament voor de OSB koppelvlakstandaard WUS. Die keuzen, de nadere invullingen voor de OSB binnen de ruimte van die standaarden/profielen, en specifieke aandachtspunten bij de keuzen vormen tezamen de “voorschriften” per onderwerp. Ook worden hier best practices beschreven.

Hoofdstuk 3 definieert het resulterende OSB WUS profiel 1;

Hoofdstuk 4 bevat voorbeelden waarin de voorschriften uit hoofdstuk 2 zijn gerealiseerd in concrete service.

Bijlage 1 bevat een best practice die is overgenomen uit de Standaard WSDL 1.1.

Bijlage 2 bevat een voorbeeld xsd en WSDL.

Bijlage 3 bevat informatie over de verschillende versies van WS-Addressing, ondersteund door diverse toolkits.

Status

Deze versie van dit document bevat nog een aantal open punten. Die punten zijn opgenomen in paragraaf “Open aandachtspunten”. Deze versie is vooral bedoeld voor het afstemmingsproces met betrokkenen. Dat afstemmingsproces, tezamen met ervaringen uit een aantal pilots resp. eerste implementaties zal leiden tot de vast te stellen versie 1.0 van de OSB koppelvlakstandaard WUS.

1.5 Gehanteerde terminologie

Zie OSB Glossary.



2 Koppelvlakstandaard WUS

2.1 Inleiding

Doel van de Koppelvlakstandaard WUS is het eenduidig en bruikbaar definiëren van het koppelvlak voor WUS. Er zijn daarvoor de volgende onderwerpen relevant: WSDL met adressering en naamgeving, Beveiliging, resulterende berichtheaders, Compliance voorzieningen.

WSDL

Een web service wordt deels formeel en automatisch verwerkbaar gedefinieerd (beschreven – description) door een WSDL. Deze WSDL geeft een beschrijving van de eisen die ten aanzien van de communicatie gesteld worden. Het is een abstracte definitie van de web service. De web service communiceert feitelijk middels SOAP berichten, die gegenereerd worden op basis van de WSDL. Adressering en naamgeving zijn specifieke aandachtsgebieden.

Beveiliging

In een WSDL worden echter niet de beveiligingsaspecten beschreven. Dat geldt voor beveiliging conform WS-Security en conform TLS. Dit wordt veroorzaakt doordat er nog geen goede (interoperabele) oplossing is om dergelijke definities te koppelen. Er is wel een standaard in ontwikkeling, namelijk de WS-PolicyAttachment, die beoogt om policies (beschrijving van bijvoorbeeld de WS-Security vereisten) te koppelen aan een WSDL. Momenteel ontbreekt hier echter de volwassenheid van de standaard en de benodigde software ter verwerking. Daarom zijn in deze OSB koppelvlakstandaard ook aparte paragrafen opgenomen betreffende beveiliging.

Resulterende berichtheaders

Deze standaard beschrijft per definitie aan welke eisen een WSDL resp een TLS implementatie moet voldoen. De praktijk leert dat dergelijke eisen vaak erg abstract zijn, en dus gebaat bij voorbeelden. Voorbeelden zijn gewenst zowel voor een mogelijk resulterende WSDL, en eveneens voor de resulterende berichten, met name de headers.

Compliance Voorzieningen (zie apart document)

Voor de OSB koppelvlakstandaard WUS zijn ook compliance voorzieningen beschikbaar gesteld, zodat een ontwikkelaar kan testen “of het werkt”. Deze compliance voorzieningen zijn gedefinieerd aan de hand van de WSDL’s die via de OSB web site beschikbaar zijn. De volgende compliance voorzieningen zijn beschikbaar:

- OSB Service Requester Compliance Voorziening; wordt gebruikt om een ontwikkelde client te testen
- OSB Service Provider Compliance Voorziening; wordt gebruikt om een ontwikkelde Service te testen
- OSB WUS WS-I Compliance voorziening; wordt gebruikt om te toetsen of de uitwisseling voldoet aan de WS-I regels.



2.2 Gehanteerde standaarden

In dit hoofdstuk worden de verschillende gehanteerde standaarden en versies van de onderliggende internationale standaarden vermeld.

2.2.1 Overwegingen

Primair wordt gekozen voor de interoperabele profielen van WS-I.

De OSB kiest voor standaards die algemeen interoperabel beschikbaar zijn, d.w.z. interoperabel geïmplementeerd zijn in het grootste deel van de (ontwikkel)tools. De kans daarop is groter bij “final” standaarden dan bij draft. De OSB kiest daarom voor WS-I standaarden met status final. Alleen indien er een zeer dringende reden is, wordt voor (delen uit) een standaard met status Draft gekozen.

De status van de diverse “deliverables” van deze WS-I organisatie is (begin 2007) als volgt:

WS-I Basic Profile 1.1 heeft de status “final”; in deze standaard is nog een beperkt aantal onderwerpen opgenomen. Deze zijn gebaseerd op onderliggende standaarden als WSDL 1.1, SOAP 1.1 en UDDI.

WS-I Basic Profile 1.2 is nu (sinds 17-10-2006) een “working group draft”. In deze versie zijn zaken als WS-addressing en MTOM (attachments) opgenomen.

WS-I werkt aan Basic Profile 2.0, deze heeft echter nog geen officiële status. Ook in deze versie zal expliciet niet met WSDL 2.0 gewerkt worden.

WS-I Basic Security Profile (BSP) 1.0 is beschikbaar met status final sinds 30 maart 2007; BSP 1.0 verwijst naar BP 1.0 en 1.1.

BSP 1.1 is inmiddels ook beschikbaar als working group draft.

Er wordt gewerkt aan WS-I Reliable and Secure Profile (RSP). Hiervan is nog geen draft beschikbaar.

2.2.2 Resulterende beslissingen t.a.v. Standaarden

t.a.v. standaarden (alleen lijst van gekozen versies met eventueel gevolgen?)

Gekozen wordt in OSB versie 1.0:

- WS-I BP 1.1; deze is gebaseerd op onderliggende standaarden: SOAP 1.1, WSDL 1.1, XML 1.0 (Second Edition);
- WS-I BP 1.2 voor de te kiezen onderdelen van WS-Addressing;
- TLS 1.0 of SSLv3.0, conform de aanbevelingen in WS-I BSP 1.0 voor beveiliging op Transport/kanaal niveau;
- WS-I Simple SOAP Binding Profile Version 1.0.



Bovenstaande standaarden zijn gebaseerd op diverse onderliggende standaarden. Gevolg is dat de OSB WUS standaard versie 1.0 gebruik maakt van de volgende set van standaarden:

Standaarden	Gevolg van o.a.
HyperText Transfer Protocol 1.1 (RFC2616)	WS-I Basic Profile 1.1
SOAP 1.1	WS-I Basic Profile 1.1
WSDL 1.1	WS-I Basic Profile 1.1
XML 1.0 (Second Edition)	WS-I Basic Profile 1.1
XML Schema Part 1: Structures	WS-I Basic Profile 1.1
XML Schema Part 2: Data types	WS-I Basic Profile 1.1
TLS 1.0 (RFC2246)	WS-I Basic Profile 1.1
HTTP over TLS Transport Layer Security (RFC2818)	WS-I Basic Profile 1.1
Internet X.509 Public Key Infrastructure Certificate and CRL Profile (RFC2459)	WS-I Basic Security Profile 1.0
The Secure Sockets Layer Protocol Version 3.0	WS-I Basic Security Profile 1.0
WS-Addressing	BP 1.2 (Wordt ondanks "Draft" versie toch gebruikt, zie paragraaf 2.3.2)

2.3 Voorschriften

De inhoud van de OSB standaard vloeit voort uit bovengenoemde standaarden, dan wel vormen een nadere invulling daarvan (een keuze uit toegestane mogelijkheden in die standaarden).

Reeds eerder is aangegeven dat de complete set van voorschriften in de huidige WUS wereld een aantal onderwerpen betreft, die niet of slechts gedeeltelijke beschreven kunnen worden in een WSDL, d.w.z. een formeel definitiedocument.

In de paragrafen hierna worden die onderdelen achtereenvolgens behandeld:

- WSDL
- WS-Addressing
- Namespace
- TLS



2.3.1 Service definitie WSDL

Voorschriften als gevolgen van de keuze voor BP 1.1

Nr	Omschrijving
WW001	Voor de SOAP berichten wordt SOAP 1.1 en “document-literal binding” gehanteerd (BP1.1)
WW002	Door het opleggen van het SOAP style type “document/literal” zal de inhoud van de berichten beschreven worden door XML en geen afgeleide daarvan. Dit houdt in dat er niet een eigen mapping mag worden geïntroduceerd voor encoding types zoals bijvoorbeeld bij SOAPencoding het geval is. Kortom de datatypes moeten voldoen aan de XML Schema Part 2: Datatypes (Volgens WS-I BP1.1 http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/)
WW003	<p>Bij document –literal mag het SOAP “body” element slechts 1 XML element bevatten. Hierbinnen kunnen eventueel wel meerdere elementen opgenomen worden. Het is ook bijvoorbeeld mogelijk om meerdere elementen van het type {http://www.w3.org/2001/XMLSchema}base64Binary op te nemen binnen dit eerste element. Daarmee ondersteunt deze koppelvakstandaard het versturen van attachments met binaire data impliciet. Het geoptimaliseerd versturen (MTOM) van een dergelijk bericht met meerdere base64 elementen wordt nog niet ondersteund.</p> <p>Door binnen het SOAP XML bericht een element van het type {http://www.w3.org/2001/XMLSchema}base64Binary op te nemen, kan invulling worden gegeven aan het kunnen versturen van binaire data en conformiteit aan de OSB Koppelvakstandaard WUS. Deze oplossing kan ook gekozen worden indien er bijvoorbeeld XML/XBRL etc. data verzonden moet worden.</p>
WW004	<p>BP 1.1 stelt eisen aan het “PortType” van een WSDL. Hierbij mogen de “parts” van de “messages” alleen een “element” bevatten (geen “parts” die een “type” attribuut gebruiken). “R2204 A document-literal binding in a DESCRIPTION MUST refer, in each of its soapbind:body element(s), only to wsdl:part element(s) that have been defined using the element attribute.” Er is geen voorbeeld bij WS-I, maar een voorbeeld kan zijn:</p> <pre> <element name="TradePriceRequest"> <complexType> <all> <element name="tickerSymbol" type="string"/> </all> </complexType> </element> ... <message name="GetLastTradePriceInput"> <part name="body" element="xsd1:TradePriceRequest"/> </message> </pre>

Aanvullende voorschriften (dus specifieke OSB-invulling binnen de ruimte van een bovengenoemde standaard)



Nr	Omschrijving
WS001	In OSB vs 1.0 kunnen meerdere operaties per web service gedefinieerd worden.
	<p>Overweging: Indien er slechts 1 operatie per web service gedefinieerd zou mogen worden door een serviceprovider, is de transformatie in de OSB-Gateway waarschijnlijk eenvoudiger.</p> <p>1 operatie per service brengt echter zowel voor een serviceprovider als voor een servicerequester extra werk met zich mee met name bij wijzigingen.</p> <p>De huidige inschatting is dat de transformatie in de Gateway relatief eenvoudig is op te lossen, en een eventuele besparing daar niet opweegt tegen de nadelen voor Serviceproviders en -requesters.</p>
WS002	In OSB WUS versie 1 wordt de SOAPAction in WSDL gevuld met een lege string of de vermelding kan weggelaten worden, of dezelfde vulling als de wsa:Action. In de HTTP Header van het bericht moet de SOAPAction een lege string met quotes zijn (“”) of een waarde gelijk aan de WS-Addressing action (wsa:action) (nadere invulling van de mogelijkheden in WS-I Basic Profile 1.1).
	<p>Overweging: De SOAPAction moet in SOAP 1.1 toegevoegd worden, en gelijk zijn aan de request URI. In SOAP 1.2 zal het veld vervallen (deprecated).</p> <p>Een lege string voor de SOAPAction houdt in dat het bericht bestemd is voor de locatie zoals aangegeven door de http request-URI.</p>
WS003	De OSB WUS ondersteunt alleen de zogenaamde “request/response” berichtenuitwisseling (zie WSDL 1.1 specificatie paragraaf “2.4 Port Types”).
	<p>Overweging: De WSDL specificatie biedt een viertal manieren voor de interactie tussen de partij die het bericht verstuurt en de partij die het bericht ontvangt. “Request/response” en “one-way” zijn de enige twee met een binding; daarmee vervallen de overige in OSB WUS v1. One-way vereist reliability en wordt daarom in OSB vs niet met WUS ondersteund.</p>
WS004	De mogelijkheid voor toevoegen van “ attachments” worden nog niet geaccepteerd.
	<p>Overweging: De standaarden hiervoor zijn nog in beweging: BP 1.1 kent een uitbreiding, attachment profile, gebaseerd op SwA; BP 1.2 gaat uit van MTOM/XOP; MTOM ondersteunt SwA backwards compatible, maar heeft meer functionaliteit.</p>
WS005	De WSDL bevat slechts één “portType” per WSDL bestand.
	<p>Overweging: Het Porttype element is de beschrijving van de abstracte web service. Indien er meerdere zouden zijn, zijn beschrijf je dus eigenlijk meerdere web services in 1 wsdl (best practice 1 ws 1 wsdl). Vanuit uddi moeten publishers hun web services registreren, bij het plaatsen van meerdere porttypes/bindings moet er met xpointers vanuit de UDDI tModels naar de juiste binding worden verwezen. Dit brengt meer complexiteit met zich mee bij UDDI registraties.</p> <p>Hetzelfde geldt dus eigenlijk voor bindings, 1 wsdl 1 binding. (Zie boek “Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI” blz ... en “http://publib.boulder.ibm.com/infocenter/adiehelp/index.jsp?topic=/com.ibm.etools.ctc.ea.b.doc/ref/rhtwsdl.html”)</p>



WS006	OSB ondersteunt alleen UTF-8
	Overweging: BP 1.1 stelt dat berichten gecodeerd kunnen worden volgens UTF-8 én UTF-16. BP1.1 schrijft het gebruik voor van UDDI vs 2.0. UDDI ondersteunt alleen UTF-8. Vervolg onderzoek is nog nodig naar het “door elkaar gebruiken van UTF-8 en UTF16. Gezien die onzekerheden en mogelijke problemen bij legacy systemen bij het gebruik van UTF-16, wordt alleen voor het gebruik van UTF-8 gekozen.
WS007	In de header zijn geen eigen velden (header blocks) toegestaan. De header bevat alleen de in het betreffende profiel vastgestelde velden, die dus uitsluitend gedefinieerd zijn in het betreffende WS-I profiel(resp de onderliggende OASIS/W3C standaarden).
	Overweging: Definiëren van “eigen” velden ondermijnt de interoperabiliteit. Voorlopig worden alleen WS-Addressing headervelden toegestaan.

2.3.2 WS-Addressing

Voorschriften als gevolg van het toepassen van WS-Addressing

Nr	Omschrijving
WA001	<p>OSB WUSgebruikt de volgende velden uit WS-Addressing:</p> <ul style="list-style-type: none"> • wsa:To • wsa:Action • wsa:MessageID • wsa:RelatesTo <p>De communicatie binnen het OSB domein is voor een deel afhankelijk van de toepassing van WS-Addressing velden. Doordat er meerdere WS-Addressing specificaties zijn, die ondermeer verschillende namespaces kunnen hebben is er voor gekozen om alleen de specificatie van 2005/08 (http://www.w3.org/TR/2005/CR-ws-addr-core-20050817/) of van 2006/05 (http://www.w3.org/TR/ws-addr-core/) verplicht te stellen in de berichten binnen het OSB domein. Hieronder wordt de toepassing van de verschillende velden toegelicht.</p> <p>Er is gekozen voor een zo klein mogelijke subset uit de WS-Addressing standaard, die echter wel de gewenste functionaliteit ondersteunt, en zo veel mogelijk gedefinieerd kan worden in de WSDL.</p> <p>wsa:To Dit wordt gebruikt voor de routing van het bericht. De opsteller van het bericht heeft de waarde die hiervoor gebruikt wordt uit de WSDL (service address location) gehaald van de betreffende aan te roepen web service van de Service Provider.</p>



	<p>Voor het response bericht gelden geen verdere beperkingen buiten de WS-Addressing specificatie gezien het synchroon verkeer betreft.</p> <p>wsa:Action</p> <p>Deze waarde wordt gebruikt om een specifieke operatie aan te roepen. Ook deze waarde is terug te vinden in de WSDL van de betreffende aan te roepen web service van de Service Provider. Dit veld is verplicht en moet in het bericht worden opgenomen.</p> <p>wsa:MessageId</p> <p>De waarde hiervan kan door de service requester of provider zelf ingevuld worden zolang dit een waarde is die aan de onderliggende specificatie voldoet. Het volgende format heeft de voorkeur: UUID@URI UUID volgens http://www.ietf.org/rfc/rfc4122.txt URI is een anyURI volgens http://www.w3.org/2001/XMLSchema</p> <p>wsa:RelatesTo</p> <p>Dit element komt alleen voor in de SOAP header van het response bericht. Het bevat de waarde van de wsa:MessageId van het requestbericht.</p> <p>Het is toegestaan om overige WS-Addressing velden op te nemen in de berichten, hierbij geldt wel de beperking dat de waarde voor deze velden het routerings mechanisme niet verstoort. Het WUS koppelvlak in het OSB domein is bedoeld voor synchrone communicatie en er mag dus bijvoorbeeld niet op basis van het WS-Addressing ReplyTo veld naar een ander kanaal een response verwacht worden. Derhalve moet, indien het bericht andere velden dan hierboven bevat, de waarde “http://www.w3.org/2005/08/addressing/anonymous” of “http://www.w3.org/2005/08/addressing/none” aan deze velden toegekend worden.</p>
	<p>Overweging</p> <p>Beoogd voordeel: WS-Addressing kan gebruikt worden om aan een aantal eisen tegemoet te komen die mogelijk voor het OSB WUS domein gaan gelden. Zo kan bij aanwezigheid van intermediairs (bijv de OSB gateway, maar ook “Gateways” bij organisaties) de te maken route van het bericht gedefinieerd worden.</p> <p>Potentieel nadeel: Partijen hoeven zonder gebruik van WS-Addressing alleen over de technische kennis te beschikken om een SOAP header te kunnen lezen of schrijven, er is immers geen soapenv:Header. Bij toepassing van WS-addressing moet er door partijen gebruik worden gemaakt van tools die misschien niet allemaal dezelfde versie implementeren. Ook kan het gebruik van een dergelijk tool complexer zijn dan bij dan een “kale OSB” SOAP header.</p>



2.3.3 Naamgeving, namespaces

Er worden geen aanvullende eisen gesteld aan de namespaces.

In de header mogen alleen velden voorkomen conform OASIS/W3C standaarden (gebaseerd op de daar gedefinieerde namespaces) of servicespecifieke namen, gedefinieerd door de provider met een namespace gedefinieerd door de provider.

Ook aan de naamgeving van services, operations etc worden vanuit de OSB geen eisen gesteld.

Met betrekking tot het verkrijgen van eenduidigheid in de WSDL bestanden is het verplicht dat xml namespace prefixen volgens de WSDL 1.1 specificatie gebruikt worden. (<http://www.w3.org/TR/wsdl> "1.2 Notational Conventions").

De prefix "osb" is gereserveerd voor mogelijk toekomstig gebruik.

2.3.4 TLS

Deze beveiliging zorgt ervoor dat het bericht is beveiligd tijdens het transport van versturende naar de ontvangende partij. Hierbij gelden de volgende voorschriften:

Nr	Omschrijving
WT001	Authenticatie in WUS profiel 1 op transportniveau gebeurt op basis van TLS V1.0 resp SSL V3.0.
WT002	De te gebruiken certificaten voldoen aan de eisen van PKI.Overheid (PvE xx).
WT003	Voor vercijfering wordt minimaal 128 AES of 3DES gebruikt, TLS_RSA_WITH_AES_128_CBC_SHA / TLS_RSA_WITH_3DES_EDE_CBC_SHA (de CSPRNG dient aan FIP 18602 te voldoen). INDIEN SSLv3 dan ook : SSL_RSA_WITH_AES_128_CBC_SHA
WT004	De geldigheid van het certificaat wordt getoetst met betrekking tot de geldigheidsdatum en de CRL.
WT005	De betreffende CRL dient zowel voor de versturende als ontvangende partij te benaderen zijn.
WT006	Voor communicatie over HTTPS wordt port 443 gebruikt.
WT007	Binnen een TLS sessie kunnen meerdere berichten verstuurd worden.
WT008	Voor de TLS sessie geldt een maximale duur (XXX min), na het verloop hiervan wordt de verbinding verbroken (socket time out / connection time out XXX?)
WT009	De inhoud van de identificerende velden in het certificaat dienen te voldoen aan de afspraken als gesteld in de functionele eisen Authenticatie OSB.



2.4 Foutafhandeling

Er worden momenteel nog geen eisen gesteld aan de foutafhandeling anders dan in de onderliggende specificaties vermeld wordt, zoals WS-I Basic Profile 1.1.

2.5 Werkwijze/Adviezen/best practices

Deze paragraaf bevat enige adviezen t.a.v. het omgaan met service definities.

Nr	Omschrijving
WB001	In WSDL 1.1 is opgenomen een Authoring Style advies (zie bijlage 1): “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.
WB002	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.
WB003	Voor communicatie binnen het OSB 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. 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.
WB004	Versies van OSB WUS protocol resp van berichten, dus versienummering van webservices en/of operaties; vereist waarschijnlijk naamgevingafspraken. Het aanduiden van versies is bijvoorbeeld mogelijk door een datum op te nemen in de namespace van de WSDL, het “location” attribuut van het service element en de namespace in de payload XSD. Een voorbeeld hiervan is de WSDL in bijlage 2
WB005	Voor het onderscheid tussen testservices en productieservices heeft het de voorkeur dat deze op aparte machines komen met een eigen DNS naam (en dus verschillende PKI overheid certificaten).
WB006	Gebuijk van document/literal wrapped style. Naast de opmerking in 2.3.1 item 3 dat voor document literal de body maar 1 element mag bevatten, heeft het de voorkeur 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. (zie http://www-128.ibm.com/developerworks/webservices/library/ws-whichwsdl/)
	Wsd definition



<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>	<pre> bericht: ... soap:envelope> <soap:body> <myMethod> <x>5</x> <y>5.0</y> </myMethod> </soap:body> </soap:envelope> ... </pre>
---	--

2.6 Open aandachtspunten

Deze versie van de OSB koppelvlakstandaard WUS biedt een basis om te gebruiken bij de pilots. Ervaringen uit de pilots zullen meegenomen naar de definitieve OSB koppelvlakstandaard 1.0 versie. Momenteel zijn er een aantal vragen onderkend:

- Beveiliging van berichten middels WS-Security.
De huidige inschatting, gemaakt in afstemming met leden van de klankbordgroep OSB, is dat WS-Security niet noodzakelijk is, en wel een extra niveau van



complexiteit veroorzaakt.

Als de noodzaak hiervoor anders blijkt te worden, zou dat kunnen leiden tot een wel opnemen van WS-Security.

- Foutafhandeling van berichten

De opmerkingen over standaard foutafhandeling in BP 1.1 lijken als generieke richtlijn te voldoen. Extra algemeen geldige OSB voorschriften of best-practices zouden dan overbodig zijn.

Ervaringen worden verzameld, die kunnen leiden tot extra voorschriften of best practices.

- Aanscherpen naamgeving

Het vrij laten van naamgeving, ook voor nieuwe versies van webservices lijkt te voldoen. Extra algemeen geldige OSB voorschriften of best-practices zouden dan overbodig zijn.

Ervaringen worden verzameld, die kunnen leiden tot extra voorschriften of best practices.



3 OSB profielen WUS

Op de OSB wordt gewerkt met zog “Profielen”. Een profiel is een gedefinieerde bundeling van functionaliteit en daarmee van voorschriften. In de huidige OSB versie is slechts één profiel aanwezig “OSB WUS Profiel 1” genaamd.

3.1 WUS1

Beveiliging

Dit profiel maakt voor de beveiliging gebruik van 2-zijdig TLS, conform de in hoofdstuk 2 gedefinieerde eisen.

Headerblocks

Voor addressing worden ondermeer de volgende headers gedefinieerd:

```
...
<soapenv:Envelope xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<wsa:To> {http://www.w3.org/2001/XMLSchema}anyURI </wsa:To>
<wsa:MessageID>UUID@ {http://www.w3.org/2001/XMLSchema}anyURI
</wsa:MessageID>
<wsa:Action> {http://www.w3.org/2001/XMLSchema}anyURI </wsa:Action>
</soapenv:Header>
...
```



4 Voorbeelden

4.1.1 Beschrijving OSB WUS Profiel 1

Kenmerken van OSB WUS profiel 1:

- Dit profiel ondersteunt geen reliable messaging.
- Dit profiel maakt alleen voor beveiliging (authenticatie en encryptie) gebruik van HTTP over TLS.

De voorbeelden zijn gebaseerd op de WSDL van bijlage 2.

Een aantal aspecten die met de OSB standaard WUS samenhangen worden in de WSDL gedefinieerd. Zo wordt “Wrapped style” voor een deel bepaald door de structuur van de inhoud van het bericht. Deze inhoud wordt beschreven door het “schema” in de WSDL. Een voorbeeld van een dergelijk schema wordt weergegeven bij “Werkwijze/Adviezen/best practices” bij itemnummer WB006.

In onderstaande figuur 4a “WSDL abstract defintion” en 4b “WSDL Binding defintie” worden andere belangrijke elementen van een WSDL weergegeven. Het SOAPAction element wordt met een waarde aangegeven die gelijk is aan de wsa:Action. De SOAPAction van een operatie wordt aangegeven in het “binding” element van de WSDL, dit is met rood aangegeven.

De bepaling dat een web service communiceert volgens document literal style wordt eveneens aangegeven in het “binding” element van de WSDL.

Figuur 4c geeft het resulterende SOAP request “on the wire” en figuur d geeft het bijbehorende reponsebericht.



```

<wsdl:message name="pingRequest">
  <wsdl:part name="parameters" element="osbxsd:ping"/>
</wsdl:message>
<wsdl:message name="pingResponse">
  <wsdl:part name="parameters" element="osbxsd:pingResponse"/>
</wsdl:message>
<wsdl:message name="toUpperCaseRequest">
  <wsdl:part name="parameters" element="osbxsd:toUpperCase"/>
</wsdl:message>
<wsdl:message name="toUpperCaseResponse">
  <wsdl:part name="parameters" element="osbxsd:toUpperCaseResponse"/>
</wsdl:message>
<wsdl:message name="TestFout">
  <wsdl:part name="parameters" element="osbxsd:fout"/>
</wsdl:message>
<wsdl:portType name="IOSBVoorbeeldService">
  <wsdl:operation name="ping">
    <wsdl:input message="tns:pingRequest" wsa:Action="
http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSBVoorbeeldService/pingRequest/"/>
    <wsdl:output message="tns:pingResponse" wsa:Action="
http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSBVoorbeeldService/pingResponse/"/>
    <wsdl:fault name="TestFoutMelding" message="tns:TestFout"/>
  </wsdl:operation>
  <wsdl:operation name="toUpperCase">
    <wsdl:input message="tns:toUpperCaseRequest" wsa:Action="
http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSBVoorbeeldService/toUpperCaseRequest/"/>
    <wsdl:output message="tns:toUpperCaseResponse" wsa:Action="
http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSBVoorbeeldService/toUpperCaseResponse/"/>
    <wsdl:fault name="TestFoutMelding" message="tns:TestFout"/>
  </wsdl:operation>
</wsdl:portType>

```

...

Figuur 4a, WSDL abstract definition



Onderstaande figuur 4b geeft het binding en service element binnen de WSDL weer.

```

...
<wsdl:binding name="OSBVoorbeeldServiceBinding" type="osb:IOSBVoorbeeldService">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>

  <wsdl:operation name="ping" >
    <soap:operation soapAction="
http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSBVoorbeeldService/
pingRequest "/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="TestFoutMelding">
      <soap:fault name="TestFoutMelding" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="toUpperCase" >
    <soap:operation soapAction="
http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSBVoorbeeldService/
toUpperCaseRequest "/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="TestFoutMelding">
      <soap:fault name="TestFoutMelding" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="OSBVoorbeeldService">
  <wsdl:port name="OSBTestp1" binding="tns:OSBVoorbeeldServiceBinding">
<wsdl:documentation>
  <wsi:Claim conformsTo="http://ws-i.org/profiles/basis1.1/">
  </wsi:Claim conformsTo="http://ws-i.org/profiles/basis1.1/">
  </wsdl:documentation>
  <soap:address location="https://voorbeelden.overheidsservicebus.nl/services/
OSBVoorbeeldService"/>
  </wsdl:port>
</wsdl:service>
...

```

Figuur 4b WSDL binding definitie



In Figuur 4c wordt een voorbeeld van een bericht (“on the wire”) dat voldoet aan profiel 1 en de beschreven WSDL weergegeven. Het bericht is op te delen in verschillende lagen, een “SOAP Header”, een “SOAP Envelope” en een “SOAP Body”. De figuur toont tevens de HTTP gegevens.

```

POST http://test.services...../axis/services/OSBTestp1 HTTP/1.1
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.3
Host: 127.0.0.1:
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: "http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSBVoorbeeldService/ toUpperCaseRequest"
Content-Length: 363
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<wsa:To> http://local.org-b.nl/services/OSBVoorbeeldService </wsa:To>
<wsa:ReplyTo>
<wsa:Address>http://www.w3.org/2005/08/addressing/anonymous</wsa:Address>
</wsa:ReplyTo>
<wsa:MessageID>urn:uuid:A60CE28BB6C5C6053D11773522750002@org-b.nl
</wsa:MessageID>
<wsa:Action>http://service.voorbeeld.osb.gbo.overheid.nl/200706/
osbvoorbeeldservice/IOSBVoorbeeldService/toUpperCaseRequest</wsa:Action>
</soapenv:Header>
<soapenv:Body>
<ns1:toUpperCase xmlns:ns1="http://service.voorbeeld.osb.gbo.overheid.nl/200706/osb-
voorbeeld.xsd">
<ns1:data>in</ns1:data>
</ns1:toUpperCase>
</soapenv:Body>
</soapenv:Envelope>

```

Figuur 4c, SOAP request volgens profiel 1



```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<wsa:To> http://www.w3.org/2005/08/addressing/anonymous </wsa:To>
<wsa:ReplyTo>
<wsa:Address>http://www.w3.org/2005/08/addressing/none</wsa:Address>
</wsa:ReplyTo>
<wsa:MessageID>urn:uuid:A60CE28BB6C5C6053D11773522750002@services.orgx.nl
</wsa:MessageID>
<wsa:Action>http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/
IOSBVoorbeeldService/toUpperCaseResponse</wsa:Action>
</soapenv:Header>
<soapenv:Body>
<ns1:toUpperCaseResponse
xmlns:ns1="http://service.voorbeeld.osb.gbo.overheid.nl/200706/osb-voorbeeld.xsd">
<ns1:dataUit>in</ns1: dataUit >
</ns1:toUpperCaseResponse>
</soapenv:Body>
</soapenv:Envelope>

```

Figuur 4d, SOAP response volgens profiel 1



Bijlage 1 Opdeling WSDL in WSDL 1.1

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

2.1.2 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.

<http://example.com/stockquote/stockquote.xsd>

```
<?xml version="1.0"?>
<schema targetNamespace="http://example.com/stockquote/schemas"
  xmlns="http://www.w3.org/2000/10/XMLSchema">

  <element name="TradePriceRequest">
    <complexType>
      <all>
        <element name="tickerSymbol" type="string"/>
      </all>
    </complexType>
  </element>
  <element name="TradePrice">
    <complexType>
      <all>
        <element name="price" type="float"/>
      </all>
    </complexType>
  </element>
</schema>
```

<http://example.com/stockquote/stockquote.wsdl>

```
<?xml version="1.0"?>
<definitions name="StockQuote"

  targetNamespace="http://example.com/stockquote/definitions"
  xmlns:tns="http://example.com/stockquote/definitions"
  xmlns:xsd1="http://example.com/stockquote/schemas"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
```




```

xmlns="http://schemas.xmlsoap.org/wsdl/">
<import namespace="http://example.com/stockquote/schemas"
  location="http://example.com/stockquote/stockquote.xsd"/>
<message name="GetLastTradePriceInput">
  <part name="body" element="xsd1:TradePriceRequest"/>
</message>
<message name="GetLastTradePriceOutput">
  <part name="body" element="xsd1:TradePrice"/>
</message>
<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>
</definitions>

```

http://example.com/stockquote/stockquoteservice.wsdl

```

<?xml version="1.0"?>
<definitions name="StockQuote"
  targetNamespace="http://example.com/stockquote/service"
  xmlns:tns="http://example.com/stockquote/service"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:defs="http://example.com/stockquote/definitions"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://example.com/stockquote/definitions"
    location="http://example.com/stockquote/stockquote.wsdl"/>
  <binding name="StockQuoteSoapBinding"
    type="defs:StockQuotePortType">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastTradePrice">
      <soap:operation
        soapAction="http://example.com/GetLastTradePrice"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>

```



Bijlage 2: Voorbeeld WSDL

2.1: osb-voorbeeld.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://service.voorbeeld.osb.gbo.overheid.nl/200706/osb-voorbeeld.xsd"
targetNamespace="http://service.voorbeeld.osb.gbo.overheid.nl/200706/osb-voorbeeld.xsd"
elementFormDefault="qualified">
  <xsd:element name="ping">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="berichtIn" type="xsd:string"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="pingResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="berichtUit" type="xsd:string"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="toUpperCase">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="data" type="xsd:string" minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="toUpperCaseResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="dataUit" type="xsd:string"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="fout">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="foutMelding" type="xsd:string"
nillable="true"/>

```



```

nillable="true"/>
    <xsd:element name="foutMeldingCode" type="xsd:string"
    </xsd:sequence>
    </xsd:complexType>
    </xsd:element>
</xsd:schema>
```



2.2: osb-voorbeeld-abstract-definition.wSDL

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://service.voorbeeld.osb.gbo.overheid.nl/200706"
xmlns:osbxsd="http://service.voorbeeld.osb.gbo.overheid.nl/200706/osb-voorbeeld.xsd"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:wSA="http://www.w3.org/2005/08/addressing"
targetNamespace="http://service.voorbeeld.osb.gbo.overheid.nl/200706">
  <wsdl:types>
    <xsd:schema elementFormDefault="qualified">
      <xsd:import
namespace="http://service.voorbeeld.osb.gbo.overheid.nl/200706/osb-voorbeeld.xsd"
schemaLocation="./osb-voorbeeld.xsd"/>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="pingRequest">
    <wsdl:part name="parameters" element="osbxsd:ping"/>
  </wsdl:message>
  <wsdl:message name="pingResponse">
    <wsdl:part name="parameters" element="osbxsd:pingResponse"/>
  </wsdl:message>
  <wsdl:message name="toUpperCaseRequest">
    <wsdl:part name="parameters" element="osbxsd:toUpperCase"/>
  </wsdl:message>
  <wsdl:message name="toUpperCaseResponse">
    <wsdl:part name="parameters" element="osbxsd:toUpperCaseResponse"/>
  </wsdl:message>
  <wsdl:message name="TestFout">
    <wsdl:part name="parameters" element="osbxsd:fout"/>
  </wsdl:message>
  <wsdl:portType name="IOSBVoorbeeldService">
    <wsdl:operation name="ping">
      <wsdl:input message="tns:pingRequest" wSA:Action="
http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSBVoorbeeldServ
ice/pingRequest"/>
      <wsdl:output message="tns:pingResponse" wSA:Action="
http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSBVoorbeeldServ
ice/pingResponse"/>
      <wsdl:fault name="TestFoutMelding" message="tns:TestFout"/>
    </wsdl:operation>
    <wsdl:operation name="toUpperCase">
      <wsdl:input message="tns:toUpperCaseRequest"
wSA:Action="http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSB
voorbeeldService/toUpperCaseRequest"/>

```



```
<wsdl:output message="tns:toUpperCaseResponse" wsa:Action="
http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice/IOSBvoorbeeldServ
ice/toUpperCaseResponse"/>
  <wsdl:fault name="TestFoutMelding" message="tns:TestFout"/>
</wsdl:operation>
</wsdl:portType>
</wsdl:definitions>
```



2.3: osb-voorbeeld-servicebinding.wSDL

```

<?xml version="1.0" encoding="UTF-8"?>
<wSDL:definitions xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
xmlns:soap="http://schemas.xmlsoap.org/wSDL/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:osb="http://service.voorbeeld.osb.gbo.overheid.nl/200706" xmlns:WSI="http://ws-
i.org/schemas/conformanceClaim/"
xmlns:tns="http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice"
targetNamespace="http://service.voorbeeld.osb.gbo.overheid.nl/200706/osbvoorbeeldservice"
>
    <wSDL:import namespace="http://service.voorbeeld.osb.gbo.overheid.nl/200706"
location="/osb-voorbeeld-abstract-definition.wSDL"/>
    <wSDL:binding name="OSBVoorbeeldServiceBinding"
type="osb:IOSBVoorbeeldService">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wSDL:operation name="ping">
            <soap:operation/>
            <wSDL:input>
                <soap:body use="literal"/>
            </wSDL:input>
            <wSDL:output>
                <soap:body use="literal"/>
            </wSDL:output>
            <wSDL:fault name="TestFoutMelding">
                <soap:fault name="TestFoutMelding" use="literal"/>
            </wSDL:fault>
        </wSDL:operation>
        <wSDL:operation name="toUpperCase">
            <soap:operation/>
            <wSDL:input>
                <soap:body use="literal"/>
            </wSDL:input>
            <wSDL:output>
                <soap:body use="literal"/>
            </wSDL:output>
            <wSDL:fault name="TestFoutMelding">
                <soap:fault name="TestFoutMelding" use="literal"/>
            </wSDL:fault>
        </wSDL:operation>
    </wSDL:binding>
    <wSDL:service name="OSBVoorbeeldService">
        <wSDL:port name="OSBTestp1" binding="tns:OSBVoorbeeldServiceBinding">
            <soap:address
location="https://voorbeelden.overheidsservicebus.nl/services/OSBVoorbeeldService"/>
        </wSDL:port>
    </wSDL:service>
</wSDL:definitions>

```



</wsdl:service>
</wsdl:definitions>



Bijlage 3: WS-Addressing overwegingen

Onderstaande informatie is in deze versie opgenomen, omdat het deel uitmaakt van de te voeren discussie over dit onderwerp.

Versies:

Er zijn een aantal WS-Addressing versies bekend, overzicht versies te vinden op <http://www.w3.org/2002/ws/addr/>.

Deze specificaties worden door verschillende software implementaties ondersteund, zie overzicht (deze is ad hoc opgesteld en komt niet van een internationale organisatie):

Addressing spec.	ActiveSOAP	Axis 2	WSE 2.0	WSE 3.0	XFire
March 2003.	X				
March 2004			X		
August 2004		X		X	X
August 2005		X			X
May 2006		X			

(<http://blog.springframework.com/arjen/archives/2006/07/22/ws-addressing-needs-a-phone-book/>)

Uit dit overzicht blijkt dat de versie van augustus 2004 en augustus 2005 het meest geïmplementeerd worden. Een aantal andere implementaties zoals IBM Websphere ondersteunen WS-addressing ook maar maken het uitermate moeilijk te achterhalen welke versie dat precies is.

In de OSB WUS specificatie wordt gesteld dat implementaties hiervan aan WS-I Basic Profile 1.1 dienen te voldoen. Deze versie van het BP ondersteunt het gebruik van WS-Addressing niet. Bij validatie op confirmatie aan het WS-I BP1.1 van de OSB WUS implementaties zou het mogelijk kunnen zijn dat hiermee rekening moet worden gehouden. Verder is de WS-Addressing versie 1.0 Recommendation 9 May 2006 (<http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>) wel opgenomen in de WS-I BP 1.2 draft versie. Terug kijkend naar het overzicht lijkt het er op dat dit nog maar door weinig implementaties wordt ondersteund.

Conclusie:

Beoogd is dat in de voorgenomen pilots getest wordt in hoeverre de inzet van WS-Addressing een probleem resp voordelen oplevert. Dit kan bruikbare informatie opleveren ivm welke versies er op het moment door verschillende software pakketten ondersteund worden en of deze interoperabel zijn. Een aandachtspunt blijft dat BP1.2 WS-Addressing versie may 2006 voorschrijft voor de complete implementatie.