



Logius
*Ministerie van Binnenlandse Zaken en
Koninkrijksrelaties*

Translatiespecificatie

Digikoppeling 3.0

17/11/2014

Versie 1.1.1, Definitief

Colofon

Logius Postbus 96810
Servicecentrum: 2509 JE Den Haag

t. 0900 555 4555 (10 ct p/m)
e. servicecentrum@logius.nl

Documentbeheer

Datum	Versie	Auteur	Opmerkingen
08/10/2013	1.0	Logius	-
11/06/2014	1.1	Logius	Redactionele wijzigingen
17/11/2014	1.1.1	Logius	Titel en koptekst gecorrigeerd

Inhoud

1	Inleiding.....	4
1.1	Doel en doelgroep.....	4
1.2	Digikoppeling documentatie	4
1.3	Leeswijzer	5
1.3.1	Status.....	5
1.3.2	Gehanteerde begrippen.....	5
1.4	Digikoppeling	5
1.4.1	Samenvatting	5
1.4.2	Architectuurprincipes	5
1.4.3	Koppelvlakstandaarden	6
1.5	Translatiespecificatie	6
2	Algemene architectuur	7
2.1	Inleiding	7
2.2	Vertaaldienst.....	7
2.2.1	Doel	7
2.2.2	Digikoppeling profielen	8
2.2.3	Uitgangspunten en eisen	8
2.2.4	Component overview	9
3	Implementatie	11
3.1	Mapping en routing	11
3.1.1	Protocol lifecycle berichten	11
3.1.2	Berichtmapping.....	11
3.1.3	Contractmapping.....	15
3.1.4	Service & Action mapping.....	16
3.1.5	Conversationmapping	16
3.2	Persistency	17
3.2.1	Endpoint persistency	17
3.2.2	Message persistency.....	18
3.3	Security.....	18
3.3.1	Certificaten	18
3.3.2	Point-to-Point	18
3.3.3	End-to-End security	18
3.3.4	Authenticatie en autorisatie	19
3.4	Foutafhandeling.....	19
3.5	Monitoring	19
4	Bijlage 1: Voorbeeld scenario melding inclusief terugmelding	21

1 Inleiding

1.1 Doel en doelgroep

Dit document beschrijft de specificaties voor vertaling tussen WUS en ebMS door vertaaldiensten die specifiek voor Digikoppeling 3.0 worden ingericht.

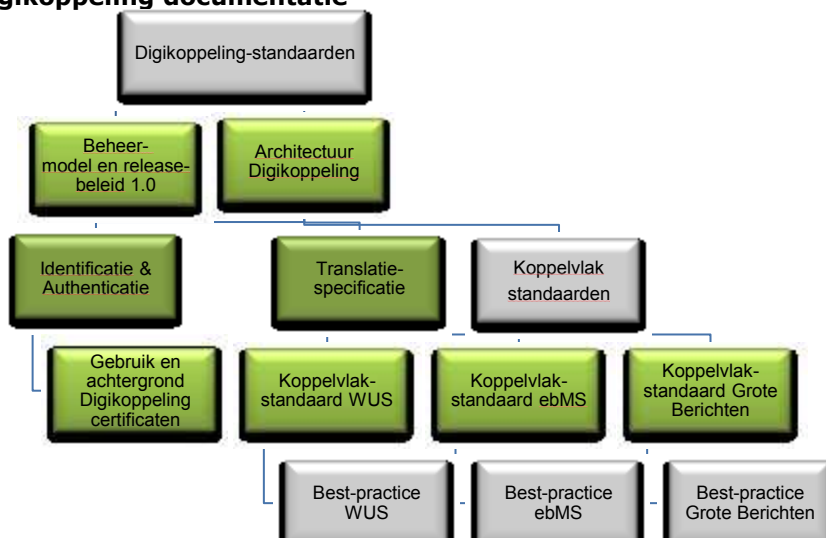
Vanaf Digikoppeling 3.0 is het mogelijk om meldingen in zowel ebMS als WUS te implementeren. De afnemers van deze meldingendiensten kunnen zelf bepalen welke protocol men wilt gebruiken. Aanbiedende partijen dienen daarom beide protocollen te ondersteunen om dit mogelijk te maken.

Aanbiedende partijen hebben de keus om beide protocollen 'in house' te ondersteunen of men kan gebruik maken van een vertaaldienst. De uniforme inrichting van vertaaldiensten bepaalt in hoge mate de uitwisselbaarheid van 'in house' ondersteuning en deze vertaaldienst(en). Dit document specificeert in detail de voorschriften aan vertaling tussen WUS en ebMS. De eisen aan vertaaldiensten zijn gespecificeerd in de Architectuur Digikoppeling 3.0.

Dit document is bedoeld voor ontwikkelaars en architecten die een vertaaldienst gaan realiseren. Zie onderstaande tabel bij welke taken dit document ondersteunt.

Afkorting	Rol	Taak	Doelgroep?
[MT]	Management	Bevoegdheid om namens organisatie (strategische) besluiten te nemen.	Nee
[PL]	Projectleiding	Verzorgen van de aansturing van projecten.	Nee
[A&D]	Analyseren & ontwerpen (design)	Analyseren en ontwerpen van oplossings-richtingen. Het verbinden van Business aan de IT.	Ja
[OT&B]	Ontwikkelen, testen en beheer	Ontwikkelt, bouwt en configureert de techniek conform specificaties. Zorgen voor beheer na ingebruikname.	Ja

1.2 Digikoppeling documentatie



Figuur 1: Opbouw documentatie Digikoppeling

- Alle groene gekleurde documenten vallen onder het beheer zoals geformaliseerd in het *Beheermodel en release-beleid 1.0*.

- Een overzicht van alle Digikoppeling documentatie is opgenomen in *Bijlage A: Bronnen*.
- Alle goedgekeurde documenten zijn te vinden op de website van Logius, www.logius.nl/digikoppeling.

1.3 Leeswijzer

Hoofdstuk 1 bevat een aantal algemene inleidende onderwerpen.
Hoofdstuk 2 bevat het algemene architectuur en de bijbehorende uitgangspunten
Hoofdstuk 3 bevat de implementatie richtlijnen onderverdeeld in specifieke onderwerpen als mapping, routing en security

1.3.1 Status

Dit document is de definitieve versie van de .

1.3.2 Gehanteerde begrippen

Waar in dit document WUS wordt genoemd, is dat inclusief WSRM.
De definities van begrippen die binnen het Digikoppeling documentatie gehanteerd worden zijn in **Bijlage B: Begrippenlijst** van het document 'Architectuur Digikoppeling 3.0' opgenomen.

1.4 Digikoppeling

1.4.1 Samenvatting

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

Digikoppeling biedt de mogelijkheid om op een sterk gestandaardiseerde wijze berichten uit te wisselen tussen serviceaanbieders (service providers) en serviceafnemers (service requesters of consumers).
Om digitale berichten uit te wisselen moeten organisaties op drie niveaus afspraken maken:

- Over de inhoud en betekenis van berichten (payload en eventuele bijlagen): de structuur, semantiek, waardebereiken enzovoort.
- Over de logistiek (envelop): transportprotocollen (HTTP), messaging (SOAP), adressering, beveiliging (authenticatie en encryptie) en betrouwbaarheid.
- Over het transport (netwerk): de protocollen van de TCP/IP stack (TCP voor Transport, IP voor Netwerk) en de infrastructuur, bijvoorbeeld Diginetwerk of Internet.

Digikoppeling richt zich dus uitsluitend op de logistieke laag. Deze afspraken zijn uitgewerkt in de koppelvakstandaarden en andere voorzieningen.

1.4.2 Architectuurprincipes

De koppelvakstandaarden dienen te leiden tot een maximum aan interoperabiliteit met een minimum aan benodigde ontwikkelingspanning.
Daarom is gekozen voor bewezen interoperabele internationale standaarden.

De architectuurprincipes van Digikoppeling zijn:

1. **Interoperabiliteit:** De interoperabiliteit van diensten is mogelijk door het gebruik van bewezen interoperabele internationale standaarden.
2. **Standaardoplossingen:** Het gebruik van standaardoplossingen is mogelijk, met een minimum aan ontwikkelingspanning of maatwerk.
3. **Veiligheid en vertrouwelijkheid:** Gegevens worden veilig uitgewisseld conform de eisen van de toepasselijke wet en

regelgeving. Wanneer berichten met persoonsgegevens verstuurd worden, moet de serviceafnemer nagaan of de uitwisseling voldoet aan de wet- en regelgeving (met name de WBP).

4. **Betrouwbaarheid:** Berichtuitwisseling is betrouwbaar indien nodig.
5. **Ontkoppeling:** De ontkoppeling van diensten wordt mogelijk door de verantwoordelijkheid van de logistieke laag, de transportlaag en de bedrijfsproceslaag strikt te scheiden.

Betrouwbaarheid is een eis voor meldingen. Daarnaast dienen vertaaldiensten te zorgen voor foutafhandeling en monitoring.

1.4.3

Koppelvlakstandaarden

Digikoppeling maakt berichtenuitwisseling mogelijk op basis van de ebXML/ebMS en WUS families van standaarden, inclusief de daaraan verwante standaarden.

Aan te sluiten overheidsorganisaties hebben aangegeven op een uniforme manier (één stekker) te willen aansluiten op Digikoppeling. Organisaties die beschikken over eigen middleware (ESB, broker) kunnen de aansluiting op Digikoppeling in het algemeen realiseren via adapters in de eigen middleware. Deze adapters worden door vele ICT-leveranciers aangeboden of kunnen zelf worden ontwikkeld op basis van de Digikoppeling Koppelvlakstandaard WUS en de Digikoppeling Koppelvlakstandaard ebMS.

1.5

Translatiespecificatie

De Translatiespecificatie is opgesteld op basis van de eisen die beschreven zijn in het document 'Architectuur Digikoppeling 3.0'. De nadruk van de Translatiespecificatie ligt op de werking van een vertaaldienst en hoe de protocolvertaling tot stand komt. De details van de protocollen staan beschreven in de betreffende koppelvlakstandaarden van Digikoppeling. Digikoppeling maakt gebruik van bepaalde profielen om een set van standaarden af te spreken. De Translatiespecificatie benoemt de ondersteunde profielen waarvoor de werking van de vertaaldienst van toepassing is.

NOOT: De Translatiespecificatie biedt technische details om de ontwikkelaar zoveel mogelijk informatie te geven om een vertaaldienst te kunnen ontwikkelen. De Translatiespecificatie is echter implementatie- en techniek-onafhankelijk opgezet; de specificatie doet geen uitspraken over technologie- of implementatie-keuzes. In plaats daarvan dient een implementatie partij zelf de specifieke implementatie documentatie op te stellen waarin de gemaakte keuzes zijn vastgelegd.

2 Algemene architectuur

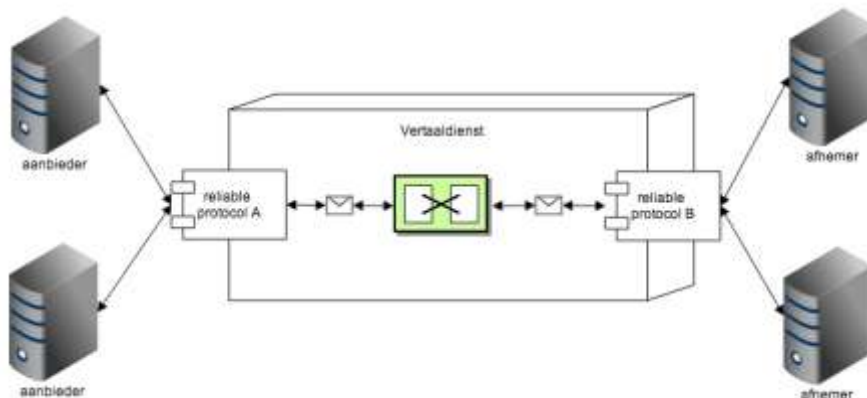
2.1 Inleiding

Vanaf Digikoppeling 3.0 is het mogelijk om een meldingendienst te baseren op de WUS standaard en/of de ebMS standaard. De afnemers kunnen zelf bepalen of zij de berichten willen ontvangen in ebMS of WUS. Een aantal aanbieders zullen zelfstandig beide protocollen implementeren maar een aantal zal daar niet voor kiezen.

Mede daarom is de behoefte ontstaan voor vertaaldiensten die de aanbieders kunnen ontlasten van het bieden van beide protocollen. Vertaaldiensten zijn nodig om een protocolvertaling te kunnen maken tussen ebMS en WUS zodat:

1. aanbieders de mogelijkheid hebben zich te richten op één protocol implementatie naar keuze;
2. afnemers de vrije keuze hebben welke van de twee protocollen zij implementeren.

De vertaaldienst wordt uitgevoerd namens de aanbiedende partijen. Afnemers kunnen daarentegen ook berichten via dezelfde weg terug sturen. Dit is nodig in het geval van het asynchroon versturen van een functionele terugmelding. Een vertaaldienst kan zowel meerdere aanbieders aansluiten als meerdere afnemers. Het volgende diagram toont de schematische werking van de data stromen:



Los van de gekozen technologie dienen de vertaaldiensten te voldoen aan een set van randvoorwaarden. Deze bieden kaders aan de implementatie partijen die een vertaaldienst gaan inrichten en zorgen voor zekerheden over wat hun klanten mogen verwachten. In de volgende paragrafen wordt dit in detail verder uitgewerkt.

2.2 Vertaaldienst

2.2.1 Doel

Het doel van een vertaaldienst is het leveren van een eenduidige oplossing die op een betrouwbare en voorspelbare manier een vertaling maakt tussen WUS en ebMS.

2.2.2

Digikoppeling profielen

Vertalingen vinden uitsluitend plaats voor meldingen. Digikoppeling 3.0 biedt voor meldingen 3 profielen aan per protocol, in feite zes mogelijke profielen. Dit zijn:

	WUS	ebMS
Betrouwbaar	Digikoppeling 2W-R	Osb-rm
Betrouwbaar en ondertekend	Digikoppeling 2W-R-S	Osb-rm-s
Betrouwbaar, ondertekend en versleuteld	Digikoppeling 2W-R-SE	Osb-rm-e

De vertaaldienst hanteert de bovenstaande tabel als profiel-mapping tabel.

2.2.3

Uitgangspunten en eisen

Voor het bepalen van de architectuur van de vertaaldienst zijn de uitgangspunten en eisen gevolgt die zijn beschreven in het document 'Architectuur Document Digikoppeling 3.0'. De onderstaande eisen zijn hieruit overgenomen.

De vertaaldienst:

- Legt afspraken vast in serviceovereenkomsten (inclusief bewerkersovereenkomsten en aansluitvoorwaarden), en contracten met serviceaanbieders en serviceafnemers.
- Heeft een eigen identiteit, dus een eigen identificatienummer (identificeert de organisatie) en een eigen PKIoverheid certificaat (authenticeert).
- Voert periodiek risicoanalyses uit op zijn complete dienstverlening en neemt passende maatregelen op het gebied van informatiebeveiliging.
- Maakt gebruik van de Translatiespecificatie om een fijnmazig verband te configureren tussen de servicedefinities (WSDL of CPA) van de koppelvlakstandaarden waartussen de protocolvertaling plaatsvindt.
 - Geeft de identiteit van verzender, ontvanger door in de berichtheaders.
 - Zorgt ervoor dat het eindresultaat op applicatieniveau hetzelfde is als wanneer dat bericht zonder vertaaldienst zou zijn verstuurd.
 - Voldoet aan de afgesproken beschikbaarheid en performance-eisen.
 - Voorziet in foutafhandeling.
 - Voorziet in sleutelbeheer en beveiliging van sleutels.
- Voorkomt dat berichten verloren gaan door berichten tijdelijk op te slaan:
 - Een bericht is pas betrouwbaar afgeleverd als de ontvanger een ontvangstbevestiging heeft gestuurd en die door de vertaaldienst is ontvangen. Een vertaaldienst moet deze ontvangstbevestiging afwachten voordat het bericht verwijderd mag worden.
 - Houdt bij of er voor meldingen technische ontvangstbevestigingen zijn aangekomen en voorziet in een signalering naar zowel verzender als ontvanger als het bericht niet bij de ontvanger is aangekomen (er is geen ontvangstbevestiging ontvangen van de ontvanger). Afhankelijk van afspraken kan deze signalering ook periodiek en/of via schriftelijke rapportages plaatsvinden.

De volgende functionele eisen worden gesteld aan vertaaldiensten in het algemeen:

- Een vertaaldienst ontvangt, vertaalt en verstuurt berichten:
 - Kan een eventuele ondertekening hiervan valideren.
 - Kan een eventuele versleuteling hiervan ontcijferen.
 - Kan een berichtheader lezen.
 - Kan een bericht omzetten naar een ander protocol.
 - Kan een bericht genereren.
 - Kan een bericht ondertekenen.
 - Kan een bericht versleutelen.
 - Kan een bericht valideren.
 - Kan een bericht versturen.
 - Kan berichtvolgorde handhaven¹.
- Een vertaaldienst heeft logging, monitoring en een audittrail ingericht en:
 - Registreert individuele berichten in de audittrail op datum en tijdstip/sequence of conversation id/message id/ontvangstbevestiging, evt. foutcodes, en indien beschikbaar de elektronische handtekening..
 - Voorziet in timestamping van alle transacties.
 - Stelt de audittrail van het knooppunt beschikbaar aan de betrokken partijen in de keten.
 - Voorziet in autorisatie voor toegang tot de audittrail. De audittrail zelf is niet wijzigbaar.
 - Voorziet in een exportmogelijkheid voor het exporteren van logfiles, audittrails en andere rapportages².
 - Neemt maatregelen om te waarborgen dat de audittrail niet kan worden gewijzigd.
- Een vertaaldienst ondersteunt bewaartermijnen:
 - Opgeslagen berichten worden opgeslagen voor zolang als nodig is om te waarborgen dat het bericht correct is verzonden en ontvangen en dit te kunnen controleren.
 - De audittrail wordt opgeslagen voor zolang als nodig is voor de auditdoeleinden zoals die met partijen zijn overeengekomen.

Naast de bovenstaande eisen zijn de volgende management en beheerfuncties belangrijk bij het inrichten van een vertaaldienst:

- Voorziet in beheerfuncties voor het beheren van de vertaaldienst.
- Voorziet in configureerbare management rapportages.
- Voorziet in autorisatie van beheerders (role-based access control).
- Voorziet in back-up en restore functionaliteit.

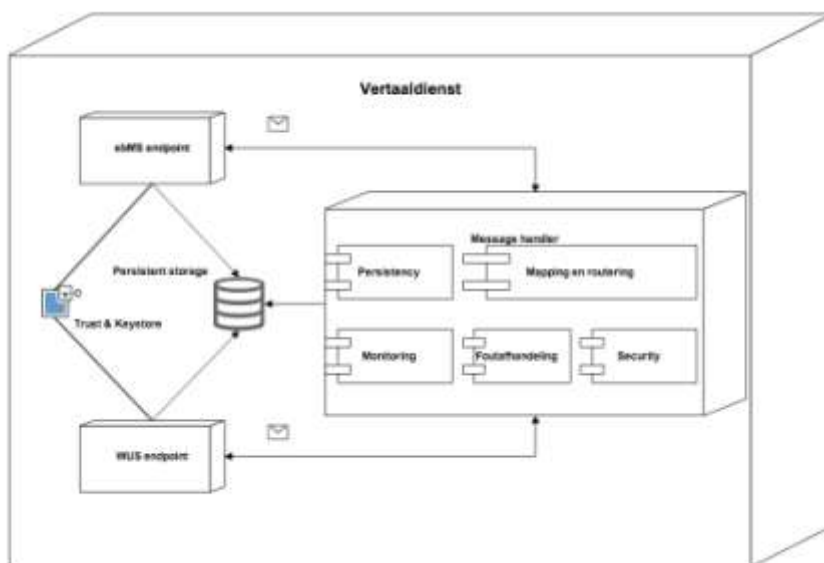
2.2.4

Component overview

In deze paragraaf worden een aantal logische componenten benoemd die benodigd zijn om een vertaaldienst op te zetten.

¹ Het is mogelijk om berichtvolgorde op protocolniveau te regelen, maar dit is doorgaans een ingewikkelde oplossing. Het is aan te raden om berichtvolgorde aan te geven door middel van berichtvolgnummers of iets vergelijkbaars. Dit kan door partijen onderling worden afgesproken.

² Indien gewenst moet het mogelijk zijn om de audittrail via een derde partij te laten verlopen, opslaan, of auditen.



In het diagram zijn de volgende componenten benoemd:

Component	Beschrijving
ebMS endpoint	ebMS endpoint is het component dat de volledige afhandeling doet van het ebMS protocol. Dit component dient volledig compliant te zijn met de ebMS Koppelvlakstandaard Digikoppeling 2.0.
WUS endpoint	WUS endpoint is het component dat de volledige afhandeling doet van het WUS protocol. Dit component dient volledig compliant te zijn met de WUS Koppelvlakstandaard Digikoppeling 3.0.
Trust & Keystore	Trust en keystore componenten worden gebruikt om de certificaten te plaatsen voor zowel Point-to-Point als End-to-End beveiliging. Zie hoofdstuk 'Beveiliging' voor meer informatie over dit onderwerp.
Persistent storage	Persistent storage is het component dat voor de opslag van data zorgt. Zie hoofdstuk 'Persistency' voor meer informatie over dit onderwerp.
Message handler	De message handler vormt het hart van de vertaaldienst implementatie. Het bevat alle functionele aspecten dat een vertaaldienst dient te leveren. De volgende logische functionele componenten zijn gedefinieerd: <ul style="list-style-type: none"> - Persistency - Mapping en routing - Monitoring - Foutafhandeling - Security <p>Alle functionele componenten worden verder beschreven onder hoofdstuk 'Implementatie'</p>

Tabel 1: Componenten vertaaldienst.

3 Implementatie

In dit onderdeel worden de relevante implementatie werkzaamheden meer in detail beschreven voor de realisatie van een vertaaldienst.

3.1 Mapping en routing

Een vertaaldienst transformeert feitelijk een ingaand bericht naar een van te voren bepaalde uitgaand berichtformaat. De transformatie wordt vaak gedaan door middel van een mapping schema. Een vertaaldienst voor meldingen daarentegen heeft extra complexiteit omdat het een reliable endpoint biedt. Reliability functionaliteit gaat namelijk samen met lifecycle berichten die zorgen voor zaken als sequentie-nummering en bevestigingen. Deze worden door de vertaaldienst zelf afgehandeld en dus niet vertaald.

Een vertaaldienst dient op verschillende niveaus informatie te mappen om uiteindelijk een bericht te vertalen en te routeren. De headers van een bericht zijn namelijk gedeeltelijk dynamisch en statisch bepaald. De mapping voor het dynamische deel, wordt beschreven in het paragraaf 'Bericht mapping'. Het statische deel wordt beschreven in de paragraaf 'Contract mapping'. Door de mapping, waar mogelijk op, contract niveau te bepalen, wordt de complexiteit voor het dynamisch vertalen en routeren van berichten verlaagd. Het streven is dus om het dynamische vertalen van berichtinformatie te minimaliseren.

3.1.1 Protocol lifecycle berichten

Betrouwbaar berichtenverkeer komt tot stand door het gebruik van acknowledgement berichten. De verzender kan hiermee bepalen of een bericht succesvol is ontvangen. Zowel ebMS als WS-RM maken gebruik van deze methodiek om de betrouwbaarheid te garanderen. Hoewel de methodiek gelijk is, zijn de implementatie verschillen groot tussen de twee protocollen. WS-RM maakt bijvoorbeeld gebruik van een aparte 'createSequence' bericht om een sequence af te spreken terwijl dit voor ebMS niet nodig is. Daarnaast kent WS-RM het concept 'piggy-backing' terwijl ebMS dit concept niet kent.

Een vertaaldienst biedt feitelijk twee onafhankelijke endpoints aan die protocol-specifiek betrouwbaar verkeer afhandelen. WS-RM maakt gebruik van een createSequence om een sequence tussen twee endpoints vast te leggen. De sequence is nodig voor WS-RM om de betrouwbaarheid functionaliteit te bieden tussen twee endpoints. Deze wordt intern door endpoint afgehandeld en wordt niet vertaald. Na een succesvol createSequence request, wordt het daadwerkelijk bericht betrouwbaar verstuurd. Dit bericht wordt na ontvangst vertaald en doorgestuurd via de andere reliable endpoint.

Omdat de endpoints in de vertaaldienst onafhankelijk van elkaar werken, worden de acknowledgement berichten van beide protocollen ook niet vertaald.

3.1.2 Berichtmapping

Dit onderdeel beschrijft in detail de mapping van dynamische data tussen WUS en ebMS berichten. Het dynamische deel van het bericht wordt bepaalt op basis van het binnenkomend bericht. Deze data wordt danwel direct 'gemapped' of getransformeerd overgezet naar het vertaalde bericht.

In de onderstaande tabellen wordt een overzicht gegeven van header gegevens die binnen komen en wat er dynamisch overgezet wordt. Elementen die niet dynamisch overgezet worden, zijn aangemerkt als 'Geen mapping nodig' met een beschrijving erbij. Bij de gegevens die getransformeerd moeten worden, staan een verwijzing naar de betreffende transformatie omschrijving. Deze zijn opgenomen in een transformatietabel aan het eind van deze paragraaf. De inhoud van het bericht wordt in principe ongewijzigd doorgezet naar het vertaalde bericht met inachtneming van de beveiligingseisen. Additioneel dient de vertaaldienst altijd geoptimaliseerde berichten te versturen zoals deze beschreven zijn in de desbetreffende koppelvlakstandaarden (WUS MTOM, ebMS SwA).

ebMS	WUS	Beschrijving
eb:From/eb:PartyId	wsa:From/wsa:Address	Zie transformatie 'TRANS1'
eb:From/eb:Role	-	Geen mapping nodig 'Role' heeft uitsluitend een betekenis in de ebMS-contexten blijft daar toepasbaar. WUS heeft geen 'Role' definitie om aan te geven of het een service requestor of provider is
eb:To/eb:PartyId	wsa:To/wsa:Address	Zie transformatie 'TRANS1'
eb:To/eb:Role	-	Geen mapping nodig Zie eb:From/eb:Role
eb:CPAId	-	Geen mapping nodig CPAId is specifiek voor ebMS om de contract instantantie te communiceren. WUS kent geen CPA in dergelijke vorm.
eb:ConversationId	-	Zie paragraaf 'Conversation Mapping'
eb:Service	-	Zie paragraaf 'Service & Action mapping'
eb:Action	-	Zie paragraaf 'Service & Action mapping'
eb:MessageData/eb:MessageID	wsa:MessageID	MessageID binnen ebMS is functioneel gelijk aan de MessageID voor WUS.
eb:MessageData/eb:TimeStamp	-	Geen mapping nodig TimeStamps worden ook in WUS gebruikt maar deze zijn onafhankelijk van de ebMS timestamp waarde.
eb:DuplicateElimination	-	Geen mapping nodig Ontdubbeling configuratie wordt binnen WUS niet door de databericht bepaald maar van te voren op service niveau ingericht.

ebMS	WUS	Beschrijving
-	wsa:ReplyTo/Address	Dit adres dient gevuld te worden met de URI van de vertaaldienst inclusief de OIN van de vertaaldienst.
eb:refToMessageId	wsa:RelatesTo	RefToMessageId Element wordt gebruikt om op protocol niveau de relatie aan te geven tussen 2 berichten. Binnen WUS wordt hiervoor de wsa:RelatesTo gebruikt.

Tabel 2: ebMS naar WUS

WUS	ebMS	Beschrijving
wsa:Action	eb:Action	Zie transformatie 'TRANS2'
wsa:MessageID	eb:MessageId	Waarde van MessageID dient gelijk te zijn aan de messageID
wsa:To/wsa:Address	eb:To/eb:PartyId	Zie transformatie 'TRANS3'
wsa:ReplyTo/wsa:Address	-	Geen mapping nodig ReplyTo concept binnen ebMS is niet geïmplementeerd op bericht niveau maar op contract niveau. De waarde van de replyTo wordt daarom niet vertaald.
wsa:From/wsa:Address	eb:From/eb:PartyId	Zie transformatie 'TRANS3'
wsa:RelatesTo	eb:RefToMessageId	RefToMessageId Element wordt gebruikt om op protocol niveau de relatie aan te geven tussen 2 berichten.
wsrc:Sequence/wsrc:Identificer	-	Geen mapping nodig Sequence identificer is gecombineerd met MessageNumber uniek binnen een sessie. Met deze unieke identificatie ondersteund WS-RM de reliability functionaliteit. De
wsrc:Sequence/wsrc:MessageNumber		

WUS	ebMS	Beschrijving
		waarde van deze elementen zijn protocol gebonden en worden ook niet door vertaald naar ebMS.
-	eb:conversationId	Zie 'Conversation Mapping'

Tabel 3: WUS naar ebMS

Transformatie naam	Beschrijving
TRANS1	ebMS werkt met een partyId om een partij te identificeren. Hierbij is het belangrijk om op te merken dat een eb:To meerdere partyId elementen kan bevatten. Digikoppeling schrijft voor dat enkel de partyId met type 'urn:osb:oin' wordt gebruikt. De transformatie dient uiteindelijk een partyId om te zetten naar een geldige wsa:Address waarde met een OIN als querystring parameter toegevoegd. PartyId's die afwijken van dit 'urn:osb:oin' formaat worden genegeerd.
TRANS2	De action waarde binnen ebMS dient getransformeerd te worden naar een geldige URI waarmee een WUS action wordt geïdentificeerd. De URI dient opgezocht te worden door middel van de bijbehorende WSDL.
TRANS3	WUS maakt gebruik van WS-addressing om routeer informatie op te nemen. Digikoppeling WUS schrijft voor dat enkel het element 'Address' gebruikt mag worden. Dit adres bevat voor meldingen een extra querystring parameter wat het 'oin' bevat. De transformatie van WUS naar ebMS partyId houdt in dat de partyID met attribuut 'urn:osb:oin' gevuld wordt met de waarde van OIN zoals deze in de parameter wordt meegegeven.

Tabel 4: Transformatie mapping

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://www.oasis-
open.org/committees/ebxml-msg/schema/envelope.xsd">
  <SOAP-ENV:Header
    xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd">
    <eb:MessageHeader
      xmlns:eb="http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd"
      SOAP-ENV:mustUnderstand="1" eb:version="2.0">
      <eb:From>
        <eb:PartyId eb:type="urn:osb:bin">[TRANS1]</eb:PartyId>
        <eb:Role>SR</eb:Role>
      </eb:From>
      <eb:To>
        <eb:PartyId
eb:type="urn:osb:bin">[Envelope/Header/To]</eb:PartyId>
        <eb:Role>SP</eb:Role>
      </eb:To>
      <eb:CPAId>[generatedId]</eb:CPAId>
      <eb:ConversationId>[Envelope/Header/MessageId]</eb:ConversationId>
      <eb:Service
eb:type="urn:osb:services">[Envelope/Header/Action]</eb:Service>
      <eb:Action>[Envelope/Header/Action]</eb:Action>
    </eb:MessageHeader>
  </SOAP-ENV:Header>
</SOAP-ENV:Envelope>

```

```

        <eb:MessageData>
            <eb:MessageId>[dynamicallyGenerated]</eb:MessageId>
            <eb:Timestamp>[dynamicallyGenerated]</eb:Timestamp>
        </eb:MessageData>
        <eb:DuplicateElimination /> <!-- wordt dit element standaard
toegevoegd of alleen als je dubbele berichten moet verwijderen? -->
    </eb:MessageHeader>
    <eb:AckRequested
        xmlns:eb="http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd"
        SOAP-ENV:mustUnderstand="1" eb:signed="false" eb:version="2.0"
        soap-env:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH" />
    </SOAP-ENV:Header>
    <SOAP-ENV:Body
        xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd">
        <eb:Manifest
            xmlns:eb="http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd"
            eb:version="2.0">
            <eb:Reference eb:id="Payload-0" xlink:href="cid:Payload-0"
                xlink:type="simple" />
        </eb:Manifest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Figuur 2: Voorbeeld: ebMS bericht

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>
        <Action xmlns="http://www.w3.org/2005/08/addressing">[TRANS2]
        </Action>
        <MessageID xmlns="http://www.w3.org/2005/08/addressing">[eb:ConversationID]
        </MessageID>
        <To xmlns="http://www.w3.org/2005/08/addressing">[TRANS3]
        </To>
        <From xmlns="http://www.w3.org/2005/08/addressing">[TRANS3]
        </From>
        <ReplyTo xmlns="http://www.w3.org/2005/08/addressing">
            <Address>[TRANS4]</Address>
        </ReplyTo>
        <ns2:Sequence soap:mustUnderstand="1"
            xmlns:ns2="http://docs.oasis-open.org/ws-rx/wsrn/200702"
            xmlns:ns3="http://www.w3.org/2005/08/addressing">
            <ns2:Identifier>urn:uuid:4a98dfel-b3cf-4672-ae8f-b3eeffe98dd2
            </ns2:Identifier>
            <ns2:MessageNumber>1</ns2:MessageNumber>
        </ns2:Sequence>
    </soap:Header>
    <soap:Body>
        [body]
    </soap:Body>
</soap:Envelope>

```

Figuur 3: Voorbeeld WUS bericht

3.1.3

Contractmapping

Zowel ebMS als WUS werken met digitale contracten waarbij in grote mate de afspraken technisch zijn vastgelegd. Binnen ebMS is dit de CPA en binnen WUS is dit de WSDL. Een belangrijk verschil tussen een CPA en WSDL is dat een CPA een 1-op-1 afspraak is en de WSDL een 1-op-n. Functioneel hebben beide contracten hetzelfde doel, namelijk: technisch vastleggen hoe de communicatie eruit ziet. Op protocol niveau worden de belangrijkste afspraken hierin vastgelegd. Om een bericht succesvol te vertalen tussen beide protocollen, dienen de contracten functioneel gelijkwaardig worden opgezet.

Binnen een CPA is het ook gebruikelijk om zowel het verkeer van A naar B als van B naar A te beschrijven in 1 CPA. Hiermee definieer je feitelijk 2 endpoints binnen 1 CPA, namelijk wat endpoint A kan ontvangen en wat endpoint B kan ontvangen. Een WSDL bevat daarentegen de informatie van 1 endpoint. Dat zou dus inhouden dat een CPA met 2 partijen zou leiden tot 2 WSDL contracten. Het is uiteraard mogelijk om alle actions in

1 WSDL vast te leggen en deze gedeeltelijk te implementeren. Dit is afhankelijk per vertaaldienst implementatie hoe hiermee om te gaan. Randvoorwaarde is dat het een functioneel gelijkwaardig resultaat oplevert conform Digikoppeling standaarden. Een vertaaldienst implementatie hanteert de volgende relaties met betrekking tot de contracten:

1. CPA heeft een bijbehorende WSDL en vice versa
2. Elke CanSend en CanReceive binnen een CPA heeft een operation in de WSDL met enkel een input definitie (het gaat immers om een asynchrone melding)
3. Zowel de CPA als de WSDL maken gebruik van dezelfde XSD schema definitie
4. CPA en WSDL dienen gelijk waardige Digikoppeling profielen te hanteren zoals deze beschreven staan in hoofdstuk 2.2.2 'Digikoppeling profielen'
5. Volgordelijkheid wordt door beide protocollen ondersteund en is optioneel. Indien volgordelijkheid binnen 1 protocol is aangezet, dient dit ook voor de andere protocol te gebeuren.

3.1.4 *Service & Action mapping*

De vertaaldienst dient een bericht van protocol A naar protocol B om te zetten en vervolgens door te sturen naar juiste endpoint. Om dit te kunnen doen, dienen er afspraken te worden gemaakt hoe de services van endpoint in beide protocollen genoemd worden en hoe deze herleid kunnen worden.

Een belangrijk verschil tussen WUS en ebMS is dat ebMS 1 endpoint adres beschikbaar stelt waar verschillende service berichten binnen komen terwijl WUS in feite 1 service per endpoint aanbiedt. Een bericht dat op een ebMS endpoint binnen komt, moet op header niveau gecontroleerd worden voor welke service deze bestemd is. De vertaaldienst dient de service vervolgens naar de juiste WUS endpoint te routeren. De implementatie dient een endpoint te mappen op basis van de 'To/PartyId' en de 'Service' naam.

Om berichten voor WUS te ontvangen dient een vertaaldienst een endpoint per service beschikbaar te stellen waar de berichten binnen kunnen komen. De mapping van deze berichten naar de juiste ebMS endpoint dient ook te gebeuren op basis van de 'To/Address'. Binnen WUS bevat het adres zowel de OIN als de servicenaam. Hiermee kan de bijbehorende ebMS endpoint bepaald worden. De wijze waarop dit gebeurt is implementatie afhankelijk.

Voor zowel WUS als ebMS bevat een service 1 of meer actions. Het bericht dat ontvangen wordt bevat de informatie welke action aangeroepen dient te worden. WUS actions hebben een standaard notatie van een volledige URI terwijl dit voor ebMS niet is gestandaardiseerd. Vertaaldienst implementaties dienen een interne mapping op te zetten tussen de ebMS action namen en de action URI's van WUS.

3.1.5 *Conversationmapping*

In de bericht mapping is al aangegeven dat de conversationId binnen ebMS niet vertaald zal worden. ebMS maakt gebruik van conversationId om aan te geven dat de berichten onder dezelfde conversatie vallen. Hiermee is het mogelijk om op 'long running transactions' met meerdere berichten te realiseren onder dezelfde conversatie. Binnen WUS is er geen equivalent aanwezig voor conversationId. Daarentegen wordt de correlatie enkel gerealiseerd door de combinatie MessageID en RelatesTo. Door een bericht te voorzien van een RelatesTo, is het mogelijk om een bericht te

correleren aan een ander bericht. Indien meerdere berichten heen en weer gestuurd worden, wordt feitelijk een keten van MessageID's opgebouwd.

Hoewel een vertaaldienst niet expliciet de conversationId vertaald, dient het wel de conversationId correct te vullen voor een ebMS bericht. Om dit te bewerkstelligen dient een vertaaldienst voor een eb:MessageId ook de bijbehorende conversationId te persisteren (bijvoorbeeld via een dynamische tabel). Bij een WUS bericht dat een RelatesTo element bevat, wordt het RelatesTo ID gebruikt om de bijbehorende ebMS message en conversationId op te halen. Indien een conversationId gevonden wordt, dient deze gebruikt te worden als waarde voor conversationId. In het geval dat er geen conversationId gevonden wordt (b.v. voor een WUS bericht dat naar ebMS wordt vertaald), dient de MessageID gebruikt te worden als conversationId.

Ter verduidelijking een voorbeeldscenario:

- De verzender verstuurt een ebMS bericht naar de vertaaldienst met messageId 1 en conversationId A
- de vertaaldienst slaat het ebMS bericht op met messageId 1 en conversationId A
- het ebMS bericht wordt vertaald naar WUS waarbij het bericht via WUS aan de afnemer wordt aangeboden
- de afnemer stuurt een terugmelding via WUS terug naar de vertaaldienst met MessageID 2 en RelatesTo 1
- de vertaaldienst zoekt het conversationId op door middel van de waarde van RelatesTo
- de vertaaldienst stuurt het ebMS bericht terug naar de verzender met de gevonden conversationId

3.2 **Persistency**

De betrouwbaarheid van een systeem wordt voor een groot deel bepaald door de manier waarop het systeem omgaat met onverwachte systeem- of netwerk-uitval. Voor netwerkuitval biedt een reliable protocol een oplossing maar bij een systeemuitval is o.a. persistency nodig. Persistency zorgt er namelijk voor dat de 'state' van een systeem niet verloren gaat als het onverwacht uitvalt. Omdat een vertaaldienst een belangrijke schakel is in een betrouwbare keten, is het noodzakelijk dat implementaties hier maatregelen voor treffen.

3.2.1 *Endpoint persistency*

Een vertaaldienst ondersteunt reliable meldingen. Een belangrijk kenmerk van reliable verkeer is dat er controle plaatsvindt of een bericht ontvangen is en indien dit niet het geval is het bericht opnieuw verstuurd wordt. Om dit te bewerkstelligen wordt de state bijgehouden voor elk bericht dat verstuurd wordt naar een endpoint. Een vertaaldienst implementatie dient met endpoint persistency ervoor te zorgen dat deze state niet verloren gaat bij systeem uitval. Zodra het systeem weer operationeel is, dient het berichtenverkeer automatisch door te gaan, alsof het systeem niet down is geweest.

Deze eis geldt niet alleen voor het ontvangen van berichten maar ook het versturen daarvan. Een vertaaldienst implementatie bevat zowel een ebMS adapter als een WUS WS-RM oplossing. Afhankelijk van de oplossing dient een persistence storage gekoppeld te worden zodat zowel de state van ebMS als de WUS oplossing bewaard blijft. De exacte configuratie is implementatie specifiek.

3.2.2 *Message persistency*

Message persistency gaat met name over hoe te waarborgen dat een bericht tijdens de afhandeling niet verloren gaat. Een protocol zorgt er namelijk voor dat een bericht reliable aankomt bij een applicatie. De vertaaldienst implementatie dient elk bericht direct te persisteren na ontvangst. Dit om het risico te minimaliseren dat tijdens het vertalingsproces het systeem uitvalt en daarmee ook het bericht verloren gaat. Welke vorm van persistentie gebruikt wordt, is implementatie afhankelijk. Na verwerking mag het bericht verwijderd worden.

3.3 **Security**

3.3.1 *Certificaten*

Digikoppeling ondersteunt zowel point-to-point als end-to-end security via de verschillende profielen. Een belangrijk middel om dit te realiseren is het gebruik van PKI overheidscertificaten. Een vertaaldienst is een belangrijke schakel in deze keten en dient te voldoen aan de Point-to-Point en End-to-End security eisen.

3.3.2 *Point-to-Point*

Vertaaldienst implementaties dienen in het bezit te zijn van eigen PKI certificaten waarmee de TLS verbinding wordt opgebouwd tussen de endpoints. TLS verbinding wordt opgebouwd tussen enerzijds 'aanbieder en vertaaldienst' en anderzijds 'vertaaldienst en afnemer'.

3.3.3 *End-to-End security*

Vertaaldienst implementaties moeten berichten vertalen en versturen in een ander protocol. Om dit te kunnen realiseren, is het nodig voor een vertaaldienst om ook berichten kunnen verwerken die versleuteld zijn en ondertekend zijn. Dit houdt in dat een vertaaldienst implementatie de certificaat inrichting dient te hebben om:

- het bericht van de aanbieder te valideren op handtekening (public key aanbieder)
- het bericht van de aanbieder te ontcijferen (decrypten) (private key vertaaldienst)
- het bericht naar de afnemer te versleutelen (public key afnemer)
- het bericht naar de afnemer te ondertekenen (private key vertaaldienst)

Concreet houdt het in dat indien indien een versleuteld bericht binnen komt, het volgende van toepassing is:

- Berichtinhoud die versleuteld is dient opnieuw versleuteld te worden met de public key van de partij die het bericht zal ontvangen
- Attachments die versleuteld zijn, dienen opnieuw versleuteld te worden met de public key van de partij die het bericht zal ontvangen

Vertaaldienst implementaties handelen de End-to-End security functionaliteit af met eigen certificaten. Dit houdt in dat het versleutelen en ondertekenen van berichten met eigen certificaten zal gebeuren. Met de aanbieder dient expliciete afspraken gemaakt te worden dat een vertaaldienst implementatie dit mag doen. In principe ondertekend een vertaaldienst implementatie het bericht namens een aanbieder en in het geval van terugmeldingen, ook namens de afnemer. Dit vereist de nodige organisatorische afspraken.

3.3.4 *Authenticatie en autorisatie*

Zoals aangegeven is een belangrijk verschil tussen de CPA van ebMS en de WSDL van WUS dat een CPA een 1 op 1 relatie is. Partijen die met elkaar kunnen praten worden expliciet met een partyInfo in een CPA vastgelegd terwijl dit met WUS niet het geval is. Authenticatie bij ebMS is dus niet alleen op certificaat niveau geregeld maar ook op CPA niveau. De vertaaldienst implementatie dient een aparte notificatie proces in te richten om authenticatie problemen te registreren en de betrokken partijen te notificeren. Het proces dient de scenario af te dekken dat een WUS bericht correct op de vertaaldienst binnen komt, terwijl deze niet afgeleverd kan worden omdat de betreffende partyInfo niet via een CPA is ingeladen.

Zowel WUS als ebMS werkt met een OIN als identificatie kenmerk. Echter aan de hand van enkel een OIN is niet te bepalen of een OIN geautoriseerd is om een service aan te roepen. Een OIN kan namelijk gekoppeld zijn aan verschillende services. Een vertaaldienst dient per OIN een lijst bij te houden welke services een OIN kan bereiken en welke endpoint daarbij hoort.

3.4 **Foutafhandeling**

Fout afhandeling binnen een vertaaldienst is extra belangrijk omdat het een vertaling uitvoert tussen twee verschillende reliable protocollen. Een reliable protocol garandeert namelijk een succesvol aflevering bij een endpoint. In het geval van een vertaaldienst moet het bericht nog vertaald en doorgestuurd worden naar de volgende endpoint. Foutafhandeling binnen dit proces dient eenduidig en transparant te zijn voor alle partijen. Technische fouten die de succesvolle aflevering bij een vertaaldienst belemmeren dienen door de verzender opgelost te worden.

Een belangrijk aspect van foutafhandeling is het inrichten van een notificatie proces. Een vertaaldienst implementatie dient een notificatie proces ingeregeld te hebben. De exacte werking en mogelijkheden van dit proces kan per vertaaldienst verschillen.

Een vertaaldienst implementatie foutafhandeling dient te voldoen aan de volgende eisen:

- alle fouten dienen via het notificatie proces gemeld te worden aan de verzendende partij
- alle fouten dienen opgeslagen te worden en traceerbaar te zijn op bijbehorende messageId
- vertaaldienst dient een mogelijkheid te hebben om fout berichten opnieuw door te zetten of te laten verwijderen door een beheerder
- foutafhandeling dient rekening te houden met de volgorde van berichten indien volgordelijkheid binnen de service is aangezet.

In het geval van volgordelijkheid moet rekening gehouden worden met de scenario dat bericht vertaling procesmatig geblokkeerd dient te worden voor berichten die bij hetzelfde conversationId (ebMS) of sequenceNumber (WUS) horen.

3.5 **Monitoring**

De vertaaldienst is een belangrijke schakel in de betrouwbare berichtsketen voor meldingen. Om een protocol vertaling succesvol uit te voeren zijn meerdere handelingen nodig. Een vertaaldienst dient de volgende maatregelen te nemen met betrekking tot monitoring:

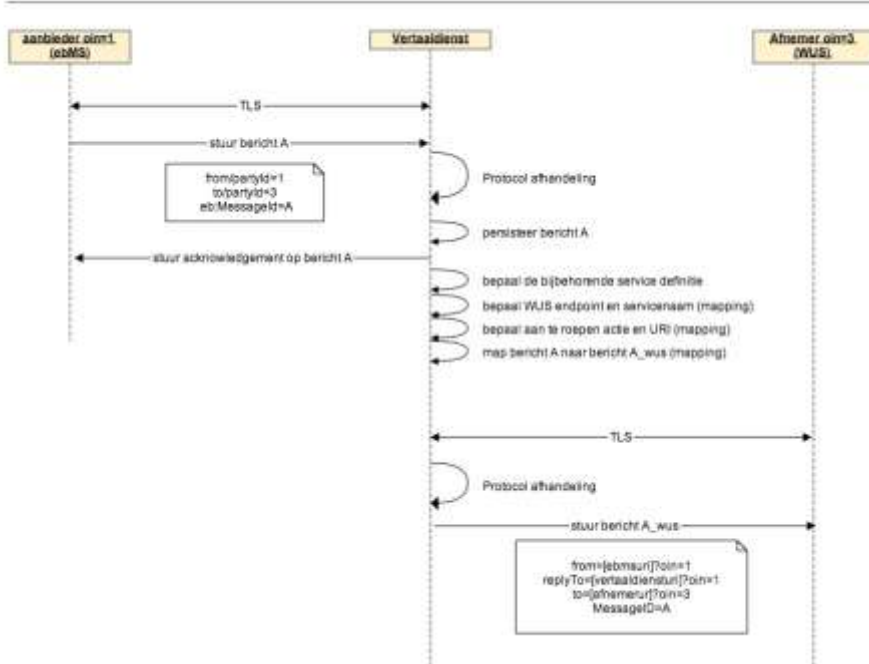
- de vertaaldienst dient een audittrail bij te houden die opvraagbaar is op messageId,

- de audittrail bevat enkel informatie dat op message header niveau is meegegeven,
- de audittrail bevat minimaal de informatie om te bepalen wanneer een bericht verstuurd is, welke routeer informatie is meegegeven en indien beschikbaar de elektronische handtekening,
- de vertaaldienst dient de monitoring informatie te beveiligen zodat deze alleen beschikbaar is voor de betrokken partijen,
- de vertaaldienst dient het systeem te monitoren op system resources en de interne programma status,
- het monitoring systeem dient notificatie mogelijkheden te bevatten die betrokken partijen kan notificeren bij systeem uitval.

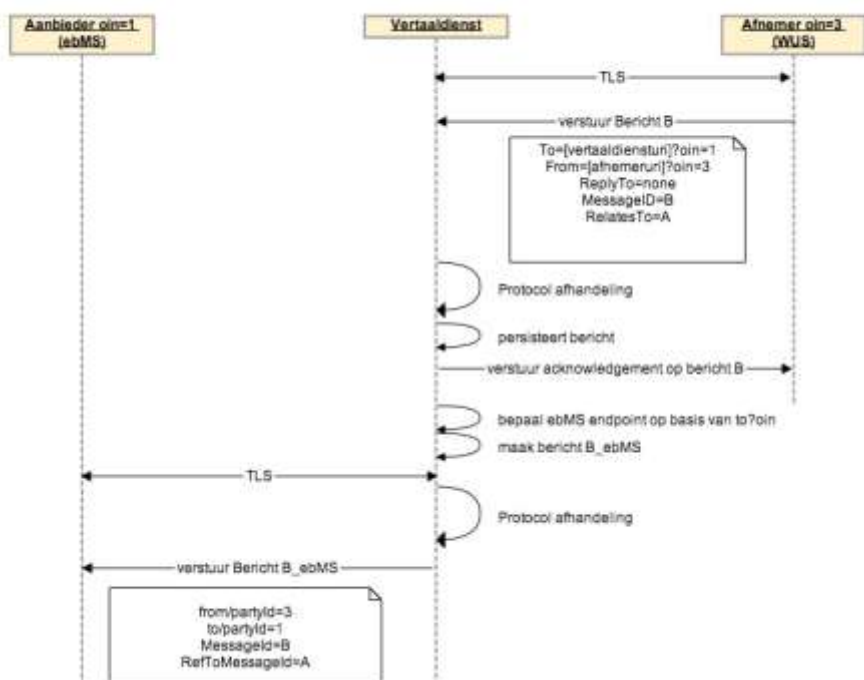
4 Bijlage 1: Voorbeeld scenario melding inclusief terugmelding

In deze bijlage wordt door middel van twee sequentie diagrammen inzicht gegeven hoe een vertaaldienst implementatie deze verwerkt.

Voorbeeld scenario



Voorbeeld scenario (terugmelding)



In de sequence diagrammen wordt na ontvangst en voor verzending van een bericht de protocol afhandeling gedaan. Dit proces is wat een adapter

aan functionaliteit levert. Dit omvat het afhandelen van de reliable messaging berichten en zaken rondom security (signing & encryption).