

NEDU/EDSN Ontwerpkeuzes

Webservices en berichtdefinities

Naam: NEDU/EDSN Ontwerpkeuzes

Versie: 3.9

Kenmerk:

Status: Goedgekeurd

Datum: 02-04-2021

Auteur: EDSN

Colofon

Versiebeheer

Versie	Naam	Datum	Status	Distributie
2.0	Ontwerpkeuzes	02-12-2011	DEFINITIEF	EDSN
2.01	Ontwerpkeuzes	16-04-2012	CONCEPT	EDSN
0.1	NEDU/EDSN Ontwerpkeuzes 2013	01-06-2012	CONCEPT	EDSN
0.2	NEDU/EDSN Ontwerpkeuzes 2013	18-07-2012	CONCEPT	EDSN
0.3	NEDU/EDSN Ontwerpkeuzes 2013	20-09-2012	CONCEPT	EDSN
0.4	NEDU/EDSN Ontwerpkeuzes 2013	30-11-2012	CONCEPT	EDSN
0.5	NEDU/EDSN Ontwerpkeuzes 2013	05-12-2012	CONCEPT	TC NEDU, EDSN
0.6	NEDU/EDSN Ontwerpkeuzes 2013	07-12-2012	CONCEPT	EDSN
3.0	NEDU/EDSN Ontwerpkeuzes 2013	28-02-2013	DEFINITIEF	NEDU, EDSN
3.1	NEDU/EDSN Ontwerpkeuzes 2013	14-06-2013	DEFINITIEF	NEDU, EDSN
3.2	NEDU/EDSN Ontwerpkeuzes 2013	14-04-2014	CONCEPT	EDSN
3.3	NEDU/EDSN Ontwerpkeuzes 2013	21-05-2014	TER VASTSTELLING	TC NEDU
3.3	NEDU/EDSN Ontwerpkeuzes 2013	11-07-2014	VASTGESTELD	NEDU, EDSN
3.4	NEDU/EDSN Ontwerpkeuzes 2013	22-06-2015	TER VASTSTELLING	TC NEDU
3.5	NEDU/EDSN Ontwerpkeuzes 2013	11-12-2015	CONCEPT	TC NEDU
3.5	NEDU/EDSN Ontwerpkeuzes 2013	18-12-2015	VASTGESTELD	NEDU, EDSN
3.6	NEDU/EDSN Ontwerpkeuzes	07-07-2016	TER VASTSTELLING	TC NEDU
3.6	NEDU/EDSN Ontwerpkeuzes	22-07-2016	VASTGESTELD	NEDU, EDSN
3.7	NEDU/EDSN Ontwerpkeuzes	22-03-2018	CONCEPT	TC NEDU
3.7	NEDU/EDSN Ontwerpkeuzes	20-04-2018	VASTGESTELD	NEDU, EDSN
3.8	NEDU/EDSN Ontwerpkeuzes	07-05-2020	CONCEPT	TC NEDU
3.8	NEDU/EDSN Ontwerpkeuzes	01-06-2020	DEFINITIEF	NEDU, EDSN
3.9	NEDU/EDSN Ontwerpkeuzes	02-04-2021	GOEDGEKEURD	TC NEDU

Wijzigingshistorie

Versie	Wijzigingen	Datum
3.1	Toelichting categorieën van Logische Data Types (LDT's) toegevoegd (zie pag. 21). Tabel met technische fouten aangevuld met SOAP Fault 5 "Time-out" (zie pag. 48). Toelichting tekenset bij CSV: ASCII (Extended ASCII conform ISO 8859-1) (zie pag. 62).	14-06-2013
3.2	Toelichting opgenomen over de notatie van "datum" en "datum/tijd" conform W3C DTF Note ISO 8601 (TC021) in paragraaf 4.5. Invoering TC023 (TLS 1.0) opgenomen in paragraaf 5.7.	14-04-2014
3.3	Toelichting opgenomen over versiebeheer en bekrachtiging van Ontwerpkeuzes door ALV NEDU of TC NEDU.	21-05-2014
3.4	Toelichting opgenomen over de notatie van "datum" en "datum/tijd" conform W3C DTF Note ISO 8601 (TC021) in paragraaf 8.5. Technische foutmeldingen (SOAP Fault) toegevoegd als gevolg van TC025.	09-03-2015
	SOAP Fault sequentiediagrammen toegevoegd bij technische foutmeldingen.	30-04-2015
3.5	Toelichting opgenomen over retry actie bij SOAP Fault als gevolg van update TC025. Toelichting bijgewerkt voor de categorieën van Logische Data Types (LDT's) als gevolg van TC028.	11-12-2015
	Verwijzingen naar BFI verwijderd als gevolg van uitfasen en uitbouwen BFI per SR2015.	18-12-2015
3.6	Naam document gewijzigd van "NEDU/EDSN Ontwerpkeuzes 2013" in "NEDU/EDSN Ontwerpkeuzes". Toelichting paragraaf 4.5: LDT categorie (TC028) in servicebeschrijvingen. Hoofdstuk 9: procesplaat EDSN Berichtencentrum bijgewerkt. Invoering TC024 (TLS 1.2) en invoering TC027 (PKI-overheid "private root" certificaten) opgenomen in paragraaf 5.7 Beveiliging.	07-07-2016
3.7	Paragraaf 5.5: zendende partij kan ook de unieke naam van het betreffende systeem zijn. IC048: toegevoegd bij paragraaf 4.5 TC032: toegevoegd bij paragraaf 4.5, onderdeel EDT TC036: toegevoegd bij paragraaf 5.7	22-03-2018
3.8	Paragraaf 4.3: naast de reeds opgestelde informatiemodellen (LDM/LOM) worden ook referentiemodellen zoals ebIX® en IEC CIM aangewend.	07-05-2020
3.9	Verwijzing naar richtlijnen voor het ontwerpen van REST API's opgenomen in de inleiding en in de tabel met de gerefereerde documenten.	09-03-2021

Gerefereerde documenten

Code	Document	Eigenaar	Datum	Status
API01	API strategie	TC NEDU	16-12-2020	Vastgesteld ALV NEDU
API02	API ontwerprichtlijnen	TC NEDU	16-12-2020	Vastgesteld ALV NEDU
API03	API URI richtlijnen	TC NEDU	16-12-2020	Vastgesteld ALV NEDU
BPSE	Basic Profile Version 1.1 Final	WS-I	10-04-2006	Final WS-I Specification
DPM	Diverse DPM's voor het Marktmodel	EDSN	2012	Vastgesteld ALV NEDU
ENTSO	The Harmonised Electricity Market Role Model	ENTSO-E	2011-01	ENTSO-E Specificatie
EXML	Extensible Markup Language (XML) 1.0 (Fifth Edition)	W3C	26-11-2008	W3C Recommendation
IC048	Eenduidige datum/periode bepaling in berichten	IC-K	04-03-2009	Vastgesteld ALV NEDU
SOAP	Simple Object Access Protocol (SOAP) 1.1	W3C	08-05-2000	W3C Note
TC001	Security berichtenverkeer	TC NEDU	13-11-2009	Vastgesteld ALV NEDU
TC003	Scheiding technische en logische foutmeldingen berichtenverkeer	TC NEDU	08-07-2009	Vastgesteld ALV NEDU
TC006	Toepassing XML Schema validatie berichten	TC NEDU	26-05-2010	Vastgesteld ALV NEDU
TC007	Idempotentie	TC NEDU	03-02-2011	Vastgesteld ALV NEDU
TC008	Versie mechanisme berichten	TC NEDU	26-05-2010	Vastgesteld ALV NEDU
TC014	Functionele compositie berichten	TC NEDU	29-09-2010	Vastgesteld ALV NEDU
TC016	Gebruik van codes in gegevensuitwisseling	TC NEDU	28-09-2011	Vastgesteld ALV NEDU
TC017	Granulariteit van berichten	TC NEDU	28-09-2011	Vastgesteld ALV NEDU
TC018	Beveiliging berichtenverkeer via centrale P4	TC NEDU	06-7-2011	Vastgesteld ALV NEDU
TC020	Semantische annotaties	TC NEDU	28-09-2011	Vastgesteld ALV NEDU
TC021	Vastlegging datum/tijd conventies	TC NEDU	28-09-2011	Vastgesteld ALV NEDU
TC023	Beveiligde verbindingen	TC NEDU	27-04-2013	Vastgesteld ALV NEDU
TC024	Overgang op TLS1.2	TC NEDU	15-01-2015	Vastgesteld ALV NEDU
TC025	Aanvullende maatregelen voor verbeteren robuustheid	TC NEDU	09-12-2015	Vastgesteld ALV NEDU
TC027	Beveiligingscertificaten 2016	TC NEDU	28-10-2015	Vastgesteld ALV NEDU
TC028	Veldlengtes in berichten	TC NEDU	28-10-2015	Vastgesteld ALV NEDU
TC032	Domeinwaarde controle in berichten	TC NEDU	18-01-2017	Vastgesteld ALV NEDU
TC036	Berichtinhoud beveiliging TMR	TC NEDU	24-05-2017	Vastgesteld ALV NEDU
UNDR	XML Naming and Design Rules Technical Specification 3.0	UN/CEFACT	17-12-2009	UN/CEFACT Specification
WSDL	Web Services Description Language (WSDL) 1.1	W3C	15-03-2001	W3C Note
XSDL	XML Schema Part 0: Primer Second Edition XML Schema Part 1: Structures Second Edition XML Schema Part 2: Datatypes Second Edition	W3C	28-10-2004	W3C Recommendation

Gebruikte afkortingen

Afkorting	Omschrijving
ACC	Aggregate Core Component
ADC	Afdrachtcontrole systeem
API	Application Programming Interface
BC	EDSN Berichtencentrum
CA	Certificate Authority
CCL	Core Components Library
CCMA	Core Component Message Assembly
CCTS	UN/CEFACT Core Components Technical Specification
CRL	Certificate Revocation List
CSV	Comma Separated Value
DEN	CCTS Dictionary Entry Names
EA	Enterprise Architect
GUI	Graphical User Interface (grafische gebruikersinterface)
HTTPS	Hypertext Transfer Protocol Secure
LDM	Logisch Data Model
LOM	Logisch Object Model
LV	Leverancier
MDD	Meetdata distributie
MV	Meetverantwoordelijke
NLV	Nieuwe leverancier in betreffende proces
OLV	Oude leverancier in betreffende proces
OMG	Object Management Group
PKI	Public Key Infrastructure
PRIM	Primitieve datatypes
PUD	Portaal Uitwissel Dienst
PV	Programmaverantwoordelijke
REST	REpresentational State Transfer
RNB	Regionale netbeheerder
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
TC	Technische Commissie NEDU
TLS	Transport Layer Security
UML	Unified Modeling Language
UMM	UN/CEFACT Modeling Methodology
UPCC	UML Profile for Core Components
URI	Uniform Resource Identifier
URN	Uniform Resource Name
UUID	Universally Unique Identifier
WSDL	Web Services Description Language
XML	eXtensible Markup Language
XSD	XML Schema Definition

Versiebeheer van document NEDU/EDSN Ontwerpkeuzes is op basis major en minor versie updates. Major versies worden vastgesteld in de ALV NEDU, minor versies worden goedgekeurd in de TC NEDU.

Inhoud

1	Inleiding	7
2	Leeswijzer	8
3	Doel en scope	9
4	Functionele architectuur	10
4.1.	Inleiding	10
4.2.	Applicatie-omgeving.....	10
4.3.	Logisch Object Model	12
4.4.	Tooling	17
4.5.	Datatypes.....	18
4.6.	Berichten.....	24
4.7.	Enveloppering	24
4.8.	Tagged values.....	26
4.9.	Services	27
4.10.	Asynchrone Batchverwerking	28
4.11.	Processen	30
5	Technische berichtenarchitectuur	31
5.1.	Communicatieprotocol	31
5.2.	Synchroniteit en batch verwerking	31
5.3.	Transactionaliteit	34
5.4.	Idempotentie	36
5.5.	EDSN Header	38
5.6.	Autorisatie.....	44
5.7.	Beveiliging.....	45
5.8.	CRL check	46
5.9.	Standaard Request, Response, Afwijzing en Foutsituaties	47
5.10.	Uitwisselen van datasets	59
6	Webservices	61
6.1.	Inleiding	61
6.2.	Ontwerp	61
6.3.	Interface granulariteit	61
6.4.	Synchrone vs. asynchrone operaties.....	62
6.5.	Aanlever vs. opvraag operaties.....	62
6.6.	Statefull vs. Stateless Interfaces	62
6.7.	Beschrijving webservice.....	62

7	XML Schema's	64
7.1.	Principes	64
7.2.	Namespace	64
7.3.	Namespace prefix	64
7.4.	Annotatie	65
7.5.	Versionering	65
7.6.	Character Encoding	67
7.7.	XML declaratie	67
7.8.	complexType name	68
8	CSV berichtstructuur	69
8.1.	Inleiding	69
8.2.	Afspraken CSV	69
8.3.	Indeling CSV	69
8.4.	Naamgeving CSV-bericht	69
8.5.	Datum en datum/tijd notatie in CSV-bericht	70
8.6.	Voorbeeld CSV	70
9	Werkwijze EDSN Berichtencentrum	71

1 Inleiding

De partijen in de Nederlandse energiemarkt, verenigd binnen de NEDU, hebben besloten alle gegevensuitwisseling tussen marktpartijen in het kader van de keten- of marktprocessen te standaardiseren en te baseren op XML of CSV. Hiertoe is het Berichtencentrum opgericht bij EDSN.

In samenwerking met de Technische Commissie NEDU (TC) is een werkmethode in het Berichtencentrum ingericht, die zoveel mogelijk is gericht op het bieden van constante kwaliteit in opgeleverde berichtdefinities.

In nauw overleg met de TC zijn en worden de te hanteren standaards bepaald. Deze standaards gelden voor de huidige werkwijze. Technologieën en standaarden voor gegevensuitwisseling zijn voortdurend in beweging. Waar de stand der techniek het mogelijk maakt en de bedrijfsvoering van de partijen in de Nederlandse energiesector dit vereist, zullen nieuwe standaards voorgesteld worden. Gezien het in toenemende mate internationaliseren van de Nederlandse energiemarkt en van de daarin betrokken partijen zal bij het voorstellen van standaards terdege rekening gehouden worden met internationale en Europese ontwikkelingen.

De webservices worden gedocumenteerd in de W3C Web Services Description Language (WSDL) en de W3C XML Schema Definition Language (XSD) conform de W3C XML Recommendation [EXML, XSDL].

Zowel de WSDL als de XSD documenten worden gegenereerd vanuit UML class modellen van binnen de sector vastgestelde communicatieprocessen, zoals vastgelegd in de Detail Proces Modellen [DPM]. UML staat voor de Unified Modeling Language, een Object Management Group specificatie, en is een internationaal toegepaste en door tools geadopteerde standaard voor beschrijving van processen en datastructuren.

Deze door het EDSN Berichtencentrum ontwikkelde en beheerde UML (informatie)modellen zijn opgezet volgens de UN/CEFACT Modeling Methodology en de UN/CEFACT Core Components Technical Specification (CCTS). De UMM methodologie wordt pragmatisch, dus niet strikt gevolgd. Het EDSN Logisch Object Model (LOM) kent wel de UMM stereotypen maar niet de UMM afleiding ("basedOn"). In plaats daarvan wordt het "derivedFrom" afleidingsprincipe gehanteerd waardoor de UML class modellen, van de berichten, een relatie blijven houden met het EDSN LOM. Dit in tegenstelling tot de UMM afleiding "basedOn" waarbij er geen relatie meer is tussen de berichtmodellen en het EDSN Logisch Object Model. De basis voor het EDSN LOM in huidige stijl is geïnitieerd door het issuedocument TC014.

De WSDL en XSD documenten zijn samengesteld op basis van UN/CEFACT Naming & Design Rules 3.0 [UNDR] en WSI Basic Profile 1.1 [BPSE].

Naast de berichtdefinities (de WSDL en XSD documenten) publiceert EDSN voor die berichtdefinities, waarvoor EDSN een ondersteunende applicatie in de communicatieketen van het betreffende marktproces biedt, tevens zogenaamde Servicebeschrijvingen. Deze Servicebeschrijvingen dienen als aanvulling op de berichtdefinities om het gebruik van de berichten en de velden in die berichten beter in de functionele en logische context te kunnen plaatsen. In veel gevallen kunnen deze Servicebeschrijvingen worden gezien als verbindende schakel tussen de betreffende DPM en de berichtdefinities. De Servicebeschrijvingen worden gepubliceerd op *mijnEDSN*.

Dit document, de NEDU/EDSN Ontwerpkeuzes, is opgesteld voor het ontwerpen en ontwikkelen van SOAP webservices en XML berichtdefinities. Voor het ontwerpen en ontwikkelen van REST API's¹ heeft de NEDU TC de ontwerprichtlijnen voor REST API's opgesteld:

1. API strategie [API01];
2. API ontwerprichtlijnen [API02];
3. API URI richtlijnen [API03].

¹ Een REST API (ook bekend als RESTful API) is een applicatie-programmeerinterface (API of web-API) die voldoet aan de richtlijnen van de REST-architectuurstijl en interactie met RESTful-webservices mogelijk maakt. REST staat voor REpresentational State Transfer.

2 Leeswijzer

Onderhavig document bevat de volgende hoofdstukindeling:

- Hoofdstuk 3: Doel en scope NEDU/EDSN Ontwerpkeuzes;
- Hoofdstuk 4: Functionele architectuur van de EDSN services;
- Hoofdstuk 5: Technische berichtenarchitectuur;
- Hoofdstuk 6: Webservices;
- Hoofdstuk 7: XML Schema's;
- Hoofdstuk 8: CSV berichtstructuur;
- Hoofdstuk 9: Werkwijze EDSN Berichtencentrum.

Om een beeld te geven van de 'wereld' waarin de berichten worden toegepast, wordt in hoofdstuk 4 een korte impressie van de diverse aandachtgebieden gegeven waarin de berichten worden toegepast. Tevens wordt hierbij een aantal verschillende soorten van communicatie besproken.

De werkmethode van het Berichtencentrum is kort beschreven in hoofdstuk 9. Als basis wordt een informatiemodel gemaakt van de benodigde informatie uitwisseling tussen partijen en/of systemen, zoals de binnen ketenprocessen in de markt uit te wisselen gegevens structuren. Dit basis informatiemodel is op aanvraag in digitale vorm beschikbaar voor de marktpartijen.

In het vervolg van dit document zijn ter verduidelijking van de structuren de keuzes toegelicht die in de berichtmodellering en -generatie zijn gemaakt.

3 Doel en scope

Dit document beschrijft de uitgangspunten en keuzes die leidend zijn bij het opzetten en ontwikkelen van de EDSN (informatie)modellen en bij het vormgeven van het WSDL en XSD genereringsproces (het maken van de berichten). Tevens wordt in dit document een aantal aspecten van de technische interface (de opzet van de webservices) behandeld. Dit laatste heeft een inleidende functie. Voor gedetailleerde protocolspecificaties wordt verwezen naar de betreffende documentatie.

De EDSN modellen worden uitgedrukt in UML. UML is een erg flexibele taal. Bij het modelleren van systemen en processen met behulp van UML moeten tal van keuzes gemaakt worden. UMM en CCTS perken de keuzes in, maar bieden nog steeds veel vrijheid, zoals bijvoorbeeld bij de naamgeving van objecten. Bij het opzetten van de EDSN webservices zijn samen met de TC keuzes gemaakt. Deze keuzes zijn zoveel mogelijk vastgelegd in dit document en worden doorgevoerd in de diverse modellen en berichtdefinities.

4 Functionele architectuur

4.1 Inleiding

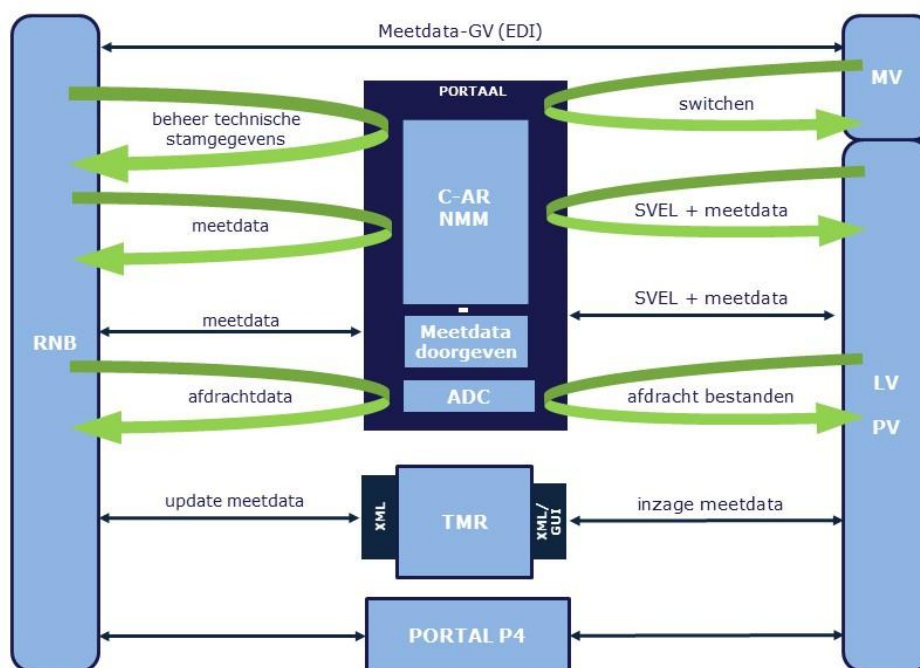
Dit hoofdstuk geeft een korte impressie van de functionele architectuur van een aantal EDSN services en van de diverse procesclusters en omgevingen waarvoor berichtdefinities door het Berichtencentrum zijn of worden gemaakt.

Als basis voor de berichten, die tussen applicaties in de energiesector worden uitgewisseld, wordt per applicatie-omgeving of ketenprocescluster een Logisch Object Model (LOM) opgesteld. Deze LOM's zijn gebaseerd op een EDSN(/sector)-breed Logisch Data Model (LDM). Vanuit de LOM's worden de UML class modellen van de berichtdefinities gegenereerd. Deze UML class modellen worden geautomatiseerd omgezet in XML Schema (XML berichtdefinities). De XML Schema's worden gepubliceerd op *mijnEDSN* en *mijnNEDU* ter implementatie door de marktpartijen en in veel gevallen door EDSN in ondersteunende applicaties.

4.2 Applicatie-omgeving

In het kader van de uitrol van het nieuwe marktmodel Stroomopwaarts, worden de nieuwe en aangepaste berichtdefinities voor de klantgerichte ketenprocessen uitgerold conform de werkwijze van het Berichtencentrum, zoals beschreven in dit document.

De klantgerichte berichten die gericht zijn op het aansluitingenregister, zoals de switch en verhuisprocessen, worden verlegd naar het Centrale Aansluitingen Register (C-AR). Het C-AR gebruikt het Portaal om de berichten te ontvangen/versturen en te behandelen. Het Portaal controleert de berichtenstroom op technische consistentie. Bovendien zorgt het Portaal ervoor dat alle uitgewisselde berichten opgeslagen worden, dat autorisatie en authenticatie kloppen en dat de berichten eventueel herhaald worden aangeboden in geval van verstoringen. Leveranciers, PV/Shippers en Meetverantwoordelijken communiceren met het C-AR (en met andere applicaties, zoals MDD) via het Portaal. Ook de Regionale Netbeheerders communiceren via het Portaal om informatie over hun aansluitingen in het centrale aansluitingenregister te onderhouden (van aanleg tot en met sloop).



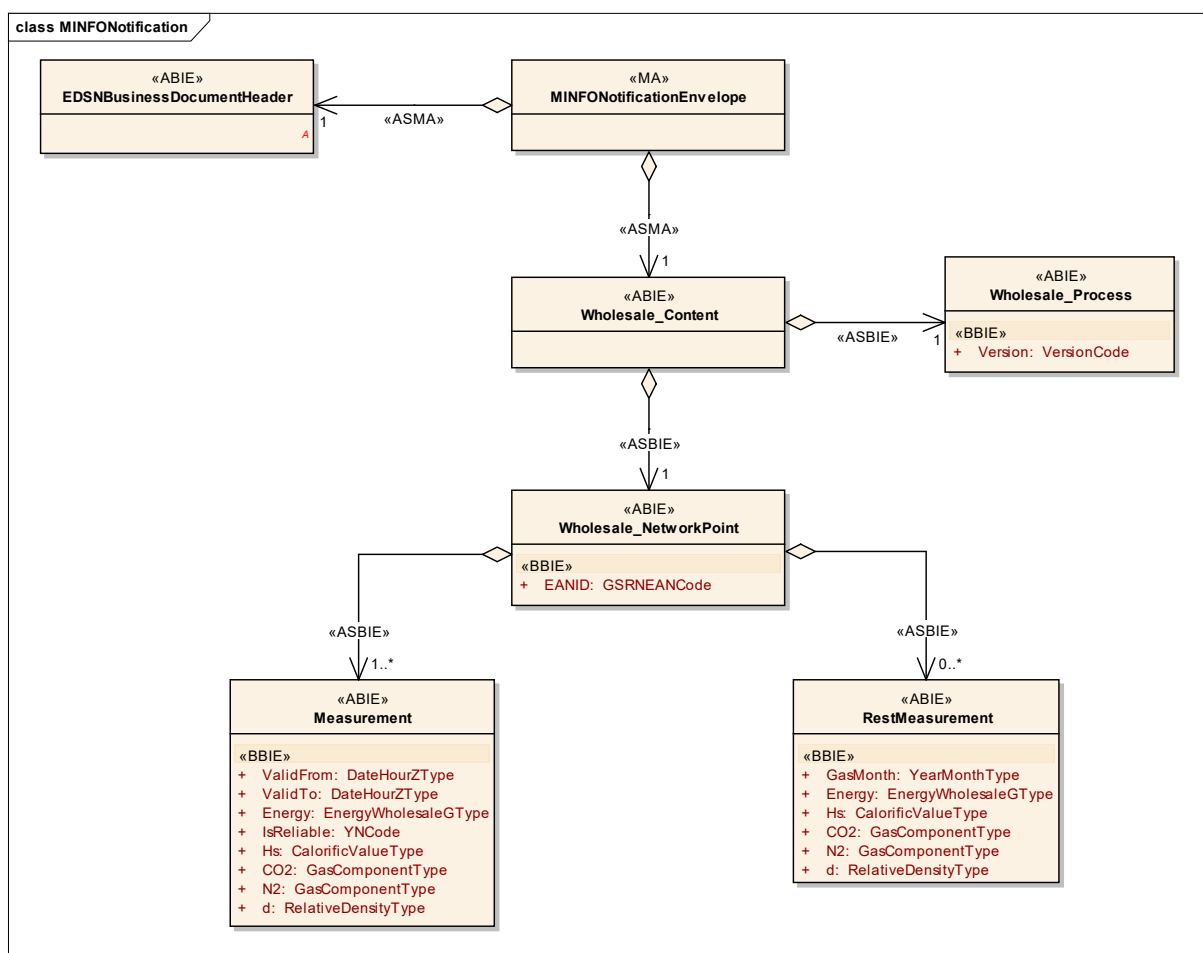
Figuur 1: Uitsnede functionele systeemarchitectuur bij invoering marktmodel Stroomopwaarts

Markpartijen (leveranciers, programmaverantwoordelijken, meetverantwoordelijken) interacteren met het Portaal via Webservices (XML) of via een GUI.

De regionale netbeheerders bieden afdrachtdata aan vanuit het centrale AfDracht Controle Systeem (ADC) die door leveranciers moet worden opgehaald en verwerkt en zonodig aangepast terug geleverd.

Het Toegankelijk Meetdataregister (TMR) biedt de registerfunctie voor de inzage in (vastgestelde, historische) meetdata. Voor meetdata distributie en transactiebewaking wordt een separate module MeetData Distributie (MDD) geoperationaliseerd.

Andere ketenprocessen en applicaties buiten getoonde domein, ook in de klantgerichte processen, zijn of worden eveneens voorzien van de nieuwe standaard voor de berichtdefinities, soms zonder gebruik te maken van de centrale Portaal voorziening. Bijvoorbeeld is dit het geval voor berichtenuitwisseling voor Allocatie en Reconciliatie gas, waarvan een voorbeeld berichtdefinitie in Figuur 2 is opgenomen.



Figuur 2: MINFONotification, berichtdefinitie voor het uitwisselen van metingen op een netwerkpunt

4.3 Logisch Object Model

De informatiemodellen worden als LDM (Logisch Data Model) c.q. LOM (Logisch Object Model) opgezet. Voor iedere ketenprocescluster of applicatie wordt een LOM opgesteld. Zo zijn er het LOM C-AR voor de netbeheerder-specifieke interactie met C-AR en het LOM Portaal, MDD en PUD voor klantgerichte marktprocessen met de interactie van de marktpartijen met het C-AR en aanverwante applicaties. Deze informatiemodellen sluiten in beginsel optimaal aan bij de datamodellen van de achterliggende centrale applicaties. De verschillende LOM's zijn alle onderdeel van het EDSN Logisch Data Model (LDM). De modellen worden opgezet met gebruikmaking van CCTS Core Components, ook om ze te zijner tijd in te kunnen brengen in het internationale standaardisatieproces voor communicatie rondom energielevering.

Objecten in het LDM en LOM zijn gemodelleerd als UML class modellen. Classes, die in verschillende LOM's voorkomen, worden generiek op bovenliggend LDM-niveau gedefinieerd. In het betreffende LOM wordt dan een afgeleide van die generieke class gemaakt, die eventueel met nog extra attributen wordt aangevuld. Deze werkwijze, geheel ondersteund door het gebruikte tool, garandeert consistentie van het model over de verschillende toepassingsdomeinen heen.

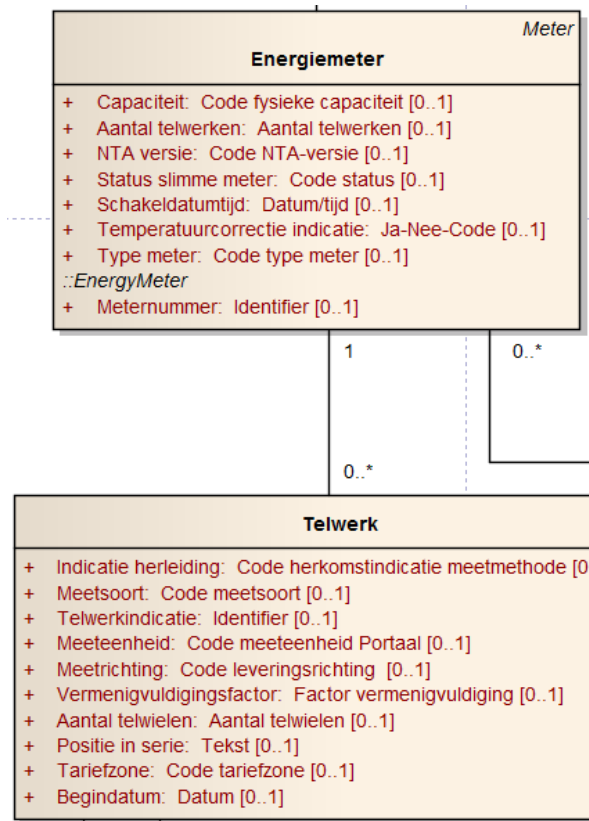
Daarnaast worden referentiemodellen, zoals ebIX® en IEC CIM, toegevoegd aan het LDM voor objecten die reeds in deze referentiemodellen zijn gedefinieerd en daardoor niet (nogmaals) in een LOM gedefinieerd behoeven te worden. Objecten die al in het LDM zijn opgenomen zouden bij ingrijpende wijzigingen aan de informatie uitwisseling mogelijk vervangen kunnen worden door objecten uit de ebIX® en IEC CIM modellen.

In het LDM en LOM wordt de relatie tussen twee objecten weergegeven zonder daarbij aan te geven of ene object wel/niet ondergeschikt is aan het andere object. Eveneens wordt in de relatie tussen twee objecten door de zgn. cardinaliteit weergegeven hoeveel objecten van het ene type in relatie kunnen staan met het objecten van andere type.

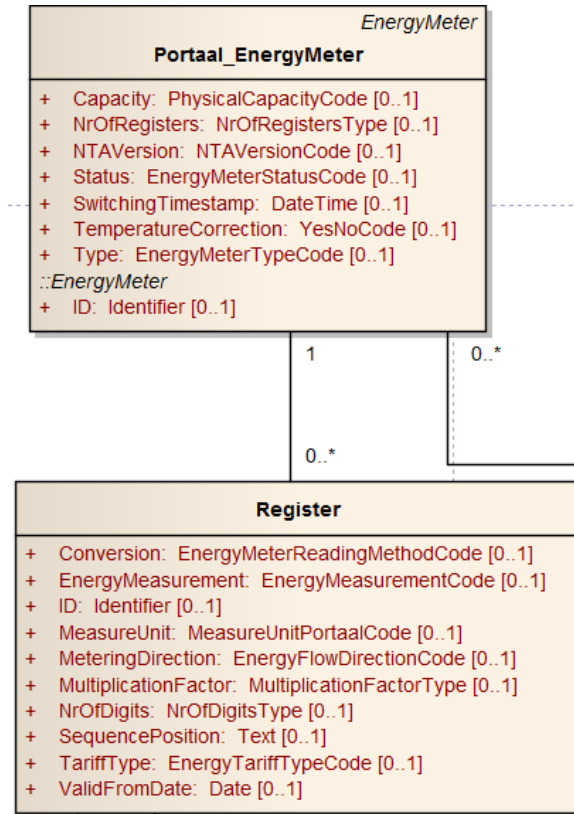
Het LOM bevat alléén de informatie die men binnen een bepaald toepassingsdomein communiceert. Het bevat heel nadrukkelijk niet alle informatie die wordt gebruikt (in de applicaties) in dat specifieke domein. Een model beperkt zich tot de informatie die vereist is in de uitwisseling, dus die voorkomt in de berichten tussen informatie leverende en informatie afnemende partijen.

Van een LOM wordt een functionele representatie (Figuur 3) en een technische representatie (Figuur 4) opgesteld. Feitelijk zijn dit twee representaties of "views" op hetzelfde LOM. In de functionele representatie wordt de Nederlandstalige functionele terminologie getoond en in de technische representatie wordt de technische naamgeving (veelal Engels) getoond.

Codd's normaalvormen (Codd, 1970) staan aan de basis van de modellering. Voor een object geldt dat een andere periode, tijd, granulariteit, status, versie, (bron)systeem, proces of eenheid niet een reden is om een object of informatiemodel te splitsen; een ander doeldomein is dat wel. Het hanteren van Normalisatieregels is vastgelegd in TC017.



Figuur 3: Uitsnede uit het LOM Portaal, functionele representatie



Figuur 4: Uitsnede uit het LOM Portaal, technische representatie

Naamgeving

De naamgeving van de berichtdefinities en de samenstellende objecten is belangrijk om consistentie over de hele sector te kunnen borgen. Namen kunnen, o.a. om reden van consistentie, bestaan uit meer dan één woord. Gespecialiseerde objecten krijgen de naam van het object waarop ze zijn gebaseerd, voorafgegaan door één of meer “Qualifiers”. Qualifiers zijn woorden die van de naam worden gescheiden door een underscore (_)².

Voorbeeld:

In de UN/CEFACT Core Components Library (CCL) is een Aggregate Core Component (ACC) object “Address.Details” opgenomen. Dit internationale adres heeft 24 attributen. Ten behoeve van de ketenprocessen in de Nederlandse energiesector is een object “EDSN_Address” gedefinieerd met 6 attributen. De “.Details” is niet in de naam van dit object opgenomen om de naam van dit object niet langer te maken.

De ‘officiële’ CCTS ‘Dictionary Entry Names’ (DEN) zijn in het UML class model opgenomen als naam van het object (class). UML Profile for Core Components (UPCC) schrijft voor dat eventuele spaties uit de DEN’s worden verwijderd om er UML namen van te maken. In de UML namen (en in de XML tag-namen) wordt dus CamelCase³ toegepast. De underscores worden echter gehandhaafd, om het onderscheid tussen multi-woord namen en gekwalificeerde namen (specialisaties) te behouden. Voorbeeld:

- UML naam: EDSN_Geographical Coordinate
- XML-tag naam: EDSN_GeographicalCoordinate

De spatie in “Geographical Coordinate” is in de XML-tag naam verwijderd.

Namen van objecten en attributen worden in de Engelse taal gesteld. Bovendien krijgt ieder object en ieder attribuut een (veelal Nederlandstalige) business term. Deze business term wordt als alias bij het object (class) opgenomen. Documentatie voor technische doeleinden bevat de Engelse termen, functionele documentatie bevat de (Nederlandstalige) business termen. In elke servicebeschrijving is een cross-reference opgenomen tussen Nederlandse termen, bijvoorbeeld zoals die in de C-AR attributenlijst en in de DPM’s voor de marktprocessen zijn opgenomen, en de Engelstalige implementatie in het informatiemodel (LOM) en in de berichten. Zie onderstaande tabel:

Fysieke eigenschappen (MPPhysicalCharacteristics)

Gegeven	Datatype	XML element	Multipliciteit	Omschrijving
Allocatiemethode	EnergyAllocationMethodCode	AllocationMethod	0..1	Allocatie methode van de aansluiting (EDT).
Capaciteitstariefcode	GLNEANCode	CapTarCode	0..1	Tariefcode (LDT).
Contractcapaciteit	Amount	ContractedCapacity	0..1	Grootverbruik: capaciteit van de aansluiting (LDT).
Leveringsrichting	EnergyFlowDirectionCode	EnergyFlowDirection	0..1	Leveringsrichting van de energie (EDT).
Afrekenmaand	InvoiceMonthType	InvoiceMonth	0..1	Factuurmaand (LDT).

Namen van attributen zijn specifiek binnen het object waartoe het attribuut behoort.

² UMM hanteert een underscore met spatie (“_”) als scheiding tussen de qualifiers. In het EDSN LDM en LOM wordt alleen een underscore (“_”) gehanteerd.

³ Toepassen van CamelCase betekent dat samengestelde woorden of zinnen, waar de woorden normaliter met een spatie gescheiden worden, als één woord worden geschreven door de spatie weg te laten en de daarop volgende letter te vervangen door de corresponderende hoofdletter.

Definities

In de Notes van een object wordt een zo goed mogelijke definitie van dat object opgenomen. Naast de definitie kunnen ook toelichtingen of procesinformatie in de Notes staan.

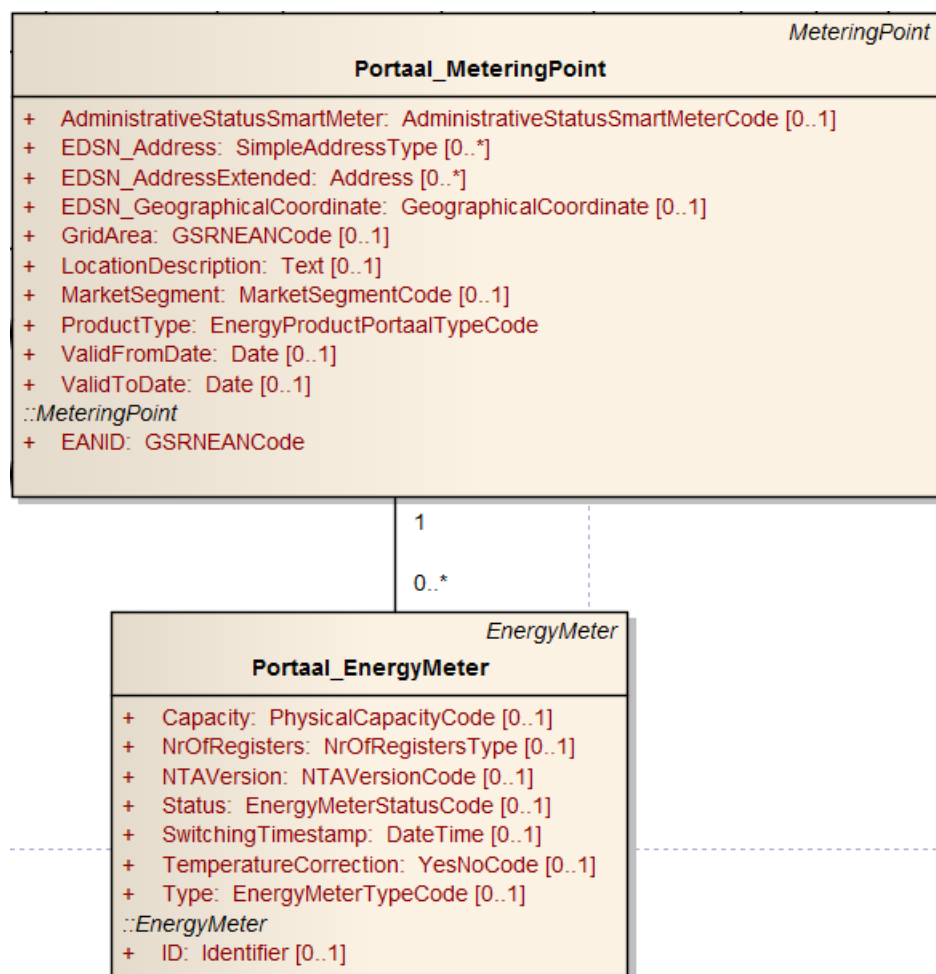
Aansluiting

In de meeste LOM's staat het begrip "MeteringPoint" (Aansluiting) centraal. De meeste gegevenselementen die in de huidige vooral klantgerichte ketenprocessen worden uitgewisseld zijn attributen van de aansluiting. Teneinde de lengte van de namen enigszins beperkt te houden wordt een dergelijk veel voorkomende term afgekort: MeteringPoint (MP). Verder zijn, om overzicht te houden in het werken met de vele attributen van de aansluiting, de attributen van de aansluiting gegroepeerd in verschillende objecten, zoals bijvoorbeeld:

- MPCommercialCharacteristics met de gegevens, die voor de commerciële kant van de processen van belang zijn;
- MPPhysicalCharacteristics met de meer technische gegevens van de aansluiting.

Bovendien kunnen aansluitingen naar verschillende gezichtspunten gegroepeerd zijn. Hiertoe is het object MeteringPointGroup gedefinieerd.

Daarnaast bestaat een aantal objecten die aan MeteringPoint gerelateerd zijn, zoals EnergyMeter. Onderstaand Figuur 5 met een uitsnede van het LOM, technische representatie, met de relatie tussen MeteringPoint en EnergyMeter.



Figuur 5: Uitsnede LOM, technische representatie

Datering van mutatie

Van veel gegevensentiteiten, zoals een aansluiting, wordt in de centrale systemen de datering van de mutatie bijgehouden (hoe de waarden van de verschillende attributen in de tijd zijn (of zullen zijn) gewijzigd). In het LOM worden daarom mutatiegegevens in het Mutation object vastgelegd (zie Figuur 6). Strikt genomen bevat het Mutation object geen statische gegevens, maar procesgegevens. Het betreft voor de processen rond C-AR, o.a.:

- MutationTimeStamp: Datum/tijd van het mutatiebericht;
- MutationReason: Reden van de mutatie;
- MutationDate: Datum effectuering mutatie marktproces.

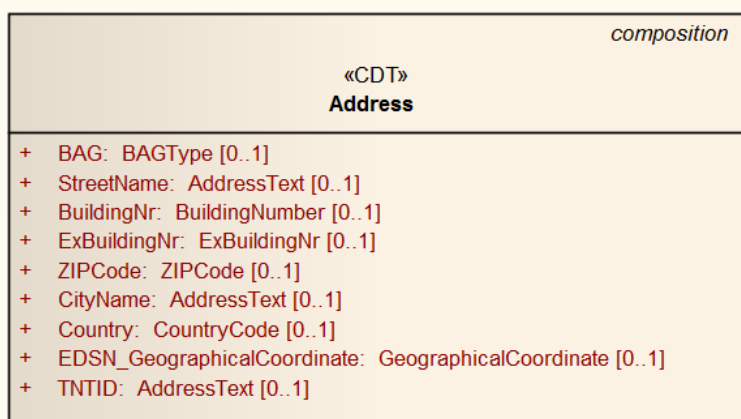
Deze mutatiegegevens (die worden bijgehouden in de database) kunnen indien gewenst worden opgenomen in berichten.



Figuur 6: Mutation object, functionele representatie

Adres- en locatiegegevens

We kennen in het algemeen meerdere adres- of locatieaanduidingen voor een aansluiting, zoals een post- of fysiek adres en/of een geo-coördinaat en/of een andere plaatsaanduiding. Een contractant kan daarnaast een (ander dan de aansluiting) adres hebben. Adressen kunnen geïdentificeerd worden m.b.v. een code, zoals een TNT- of een BAG identificatienummer. Adressen worden vastgelegd in een standaard structuur (zie Figuur 7).



Figuur 7: EDSN adresstructuur, technische representatie

Markttrollen

Het model kent de volgende markttrollen, zoals die binnen de ketenprocessen in de Nederlandse energiesector worden gebruikt:

- Netbeheerder (Grid Operator);
- Leverancier (Balance Supplier);
- PV/Shipper (Balance Responsible Party);
- Meetverantwoordelijke (Metering Responsible Party).

De specifieke verantwoordelijkheden per rol zijn voor de verschillende processen apart gedocumenteerd. Let wel dat een marktrol binnen processen verschillende rollen kan spelen (zie bijvoorbeeld het geharmoniseerde rollenmodel van Ebix-EntsoE [ENTSO]). De EAN nummers van de partijen die de rollen vervullen zijn opgenomen als attribuut van de desbetreffende objecten.

Technische objecten

In de LOM's zijn een aantal zogenaamde technische objecten gedefinieerd, deze zijn logisch gezien niet nodig maar zijn behulpzaam bij het uitwisselen van de gegevens of het aansturen van de processen of systemen. Voorbeelden van deze technische objecten:

Afwijzing-object



Figuur 8: Object Afwijzing

Resultaat-object (van een verwerking of aanvraag)



Figuur 9: Object Resultaat

4.4 Tooling

Het LDM en de LOM's worden ontwikkeld met gebruikmaking van de applicatie Enterprise Architect (EA) van SparxSystems. Dit is een modelleringstool, specifiek gericht op UML modellering.

Voor het genereren van de berichtmodellen (XSD's) uit de LOM's wordt gebruik gemaakt van een "Add-In" voor Enterprise Architect. Deze Add-In, de "LUXE-tool", is ontwikkeld door Sander Huizinga van de Gasunie en de Rijks Universiteit Groningen. Deze Add-In wordt nog verder ontwikkeld en kan inmiddels ook de webservices (WSDL's) genereren. De Add-In is ook voor andere partijen verkrijgbaar.

4.5 Datatypes

Alle in de modellen gebruikte datatypes zijn gebaseerd op primitieve datatypes (PRIM), die gedefinieerd zijn in de XML Schema specificatie van W3C [EXML, XSDL]. In de modellen worden de volgende PRIM datatypes gebruikt:

Datatype	Gebruik
base64Binary	Binaire gegevens (bijv. afbeeldingen)
boolean	Indicatoren (ja/nee, aan/uit)
composition	Samengestelde gegevens, zoals Adres
date	Datums
dateTime	Datumtijd stempels
decimal	Getallen
enumeration	Opsommingen, codelijsten
integer	Gehele getallen
string	Teksten
time	Tijdaanduidingen
xml	Tekst, opgemaakt als XML string

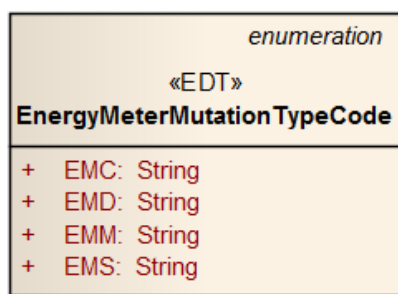
De hiervan afgeleide datatypes in de modellen zijn in drie groepen ondergebracht:

- EDT – Enumeratie Data Type;
- CDT – Complex Data Type;
- LDT – Logisch Data Type.

Deze drie groepen van afgeleide datatypes worden in het LOM gedefinieerd en worden als separaat document aan de marktpartijen voor de implementatie van de berichten ter beschikking gesteld als aanvulling op de servicebeschrijving.

EDT

Veel attributen kennen een aantal vooraf gedefinieerde en toegestane set van waardes of codes. Deze codes zijn in het model opgenomen als enumeraties. Codes worden dus in het model zelf onderhouden. Enumeraties zijn afgeleid van de primaire datatype 'enumeration'. In Figuur 10 is een voorbeeld opgenomen van een lijst, bestaande uit vier codes (EMC, EMD, EMM en EMS). De betekenissen van deze codes (Meter Change, Meter Dismount, Meter Mount en Meter Swap) zijn in de notes van de enumeratie in het model opgenomen. Deze enumeraties worden overgenomen in de XML Schema's.

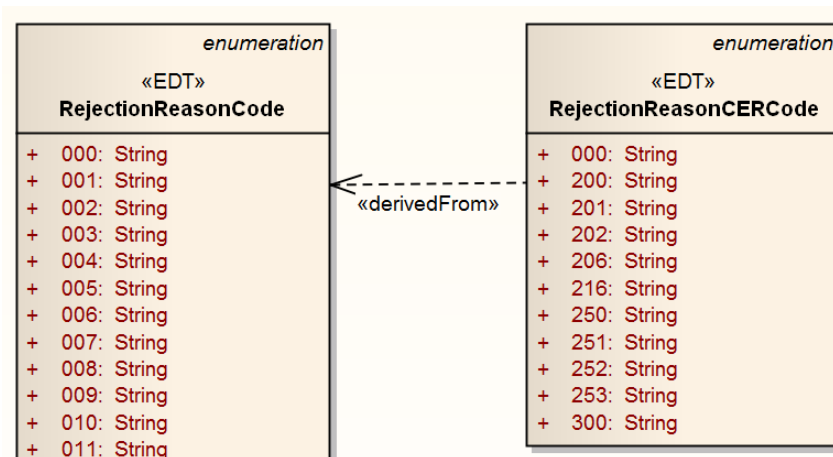


Figuur 10: EDT EnergyMeterMutationCode

Er moet onderscheid worden gemaakt tussen codes en identificators (Identifiers). Identificators (zoals EAN codes en referentienummers) worden niet in het model geënumereerd, dus niet opgesomd zoals de codes in Figuur 10. Ze staan dus ook niet als enumeraties in het XML Schema. Identificatoren dienen door de verwerkende applicaties gevalideerd te worden.

Codes met een restrictief karakter worden dus wel geënumereerd in het model. Daarmee worden de beschikbare en geldige waarden vastgelegd. Vervolgens kunnen deze in de syntax controle van de berichten worden geverifieerd (of de code logisch zinvol is dient vervolgens in de verwerkende applicatie te worden bepaald).

Afwijssreden (RejectionReasonCode) heeft een code-enumeratie die niet proces-specifiek is. Niet iedere afwijssreden is relevant in ieder proces. Het is de verantwoordelijkheid van de zendende partij om alleen (in de context) betekenisvolle afwijssredenen te communiceren. Wel wordt er een afleiding (“derivedFrom”) opgesteld uit de basislijst met afwijssredenen voor één of meerdere processen met als doel het reduceren van de hoeveelheid afwijssredenen voor die processen. Zie Figuur 11.



Figuur 11: EDT RejectionReasonCERCode als afleiding uit de basislijst RejectionReasonCode

Om “wildgroei” van codelijsten in te dammen zijn twee maatregelen vastgelegd in TC016:

Interne vaststelling

Vaststellen van de normen en richtlijnen voor het inrichten van codelijsten die gebruikt worden binnen de energiesector. Belangrijke elementen daarin zijn:

- Eenduidig gebruik van codes, limitatieve opsomming van toegestane waarden;
- Uniek gebruik van een code over alle code lijsten en gebruiksgebieden heen;
- Lokale coderingen afstemmen met internationale codestandaarden;
- Voorkomen van dubbele code voor een zelfde functionele betekenis.

Een voorstel voor dergelijke normen en richtlijnen met bijbehorende voorbeelden zijn te vinden in appendix 2 van TC016.

Externe normalisatie

Omdat coderingen en berichtstandaarden aan elkaar verbonden zijn is het van belang dat in de Nederlandse energiemarkt wordt gestreefd naar een nationaal beheer en ontwikkeling van gebruikte berichtstandaarden door één specifieke groep of instantie (aanbeveling) of als officieel erkende taak voor een bestaande groep. Deze instantie zal de bijhorende code lijsten beheren, ontwikkelen en afstemmen met internationale ontwikkelingen.

Waar mogelijk wordt gebruik gemaakt van leesbare codes (“mnemonics”) in plaats van numerieke codes. Uitzondering hierop zijn reeds bestaande enumeraties (legacy).

Afwijssreden (RejectionReasonCode) is gedefinieerd als een reeks van nummers (legacy). Nieuwe processen maken gebruik van deze, reeds bestaande reeks, van nummers. Eventuele nieuwe afwijssredenen worden aan deze reeks toegevoegd, zie Figuur 11.

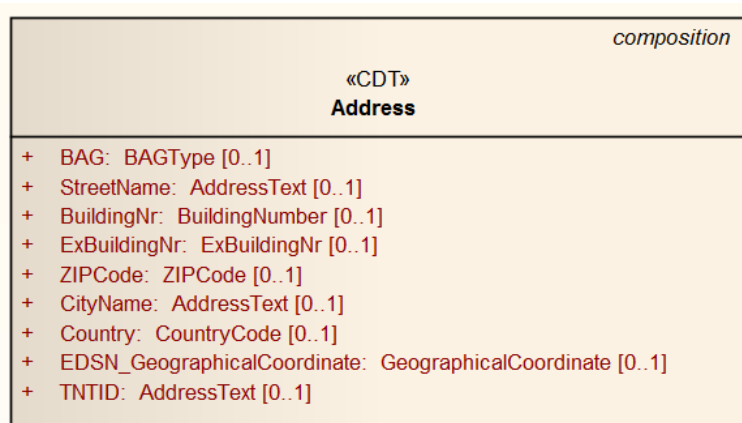
Als gevolg van TC032 worden enumeraties (EDT's) uit de berichtdefinities verwijderd en vervangen door strings (LDT's):

Issue TC032		Domeinwaarde controle in berichten		
Status	Vastgesteld			
Versienummer	1.0	Versiedatum	18 januari 2017	
Documentnaam	TC032 Domeinwaarden in berichten v1.0.doc			
Behandeld door de Technische Commissie				
Raakt aan	Kleinverbruik	Grootverbruik	Wholesale	
Elektriciteit	X	X	X	
Gas	X	X	X	
Impact op keten	Ja/Nee	Mogelijke impact marktrollen en EDSN		Ja/Nee
Codewijziging	Nee	LV		Ja
Berichten	Ja	PV		Ja
Ketenprocessen	Nee	MV		Ja
		RNB		Ja
		LNB		Ja
		EDSN		Ja
Vastgesteld door NEDU op	14 december 2016	Contact		
Invoering in markt	direct na vaststelling in ALV	Technische Commissie		

Het in berichtdefinities vervangen van enumeraties door strings zal plaatsvinden zodra een berichtdefinitie als gevolg van een NEDU issue moet worden aangepast.

CDT

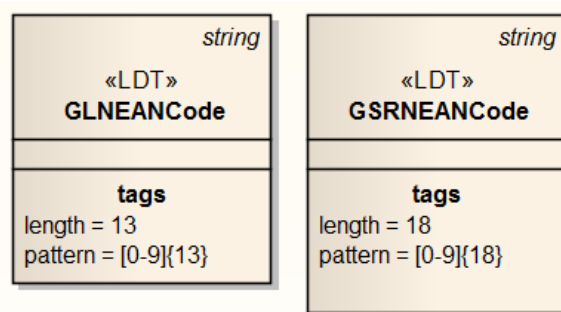
Complex Data Types (CDT's) zijn afgeleid van de primaire datatype "composition". CDT's worden gebruikt wanneer bepaalde gestructureerde informatie, die uit meerdere (vaste) samenstellende "attributen" bestaat, vaker hergebruikt wordt. Zo zijn voor adres, geografische coördinaten of eenvoudige afwijzingen CDT's gedefinieerd. Zie Figuur 12 voor adres CDT.



Figuur 12: CDT Address

LDT

Logische Data Types (LDT's) zijn afgeleid van de eerder genoemde primaire datatypen, niet zijnde een 'composition' of een 'enumeration'. Een LDT is een vaste, vooraf gedefinieerde set van eigenschappen voor een attribuut zoals minimale lengte, maximale lengte, patroon. Zo kennen we bijvoorbeeld verschillende EAN codes die middels een LDT worden gedefinieerd. In Figuur 13 zijn twee LDT's opgenomen met een lengte en een patroon (XML Schema Regular Expression). Merk op dat de "EAN-13 Check Digit" algoritme, die op de cijferreeks van de EAN-13 codes geldt, niet in het patroon is opgenomen omdat dergelijke algoritme niet goed mogelijk zijn met XML Schema Regular Expression.



Figuur 13: LDT GLNEANCode en GSRNEANCode

Formaat-restricties, zoals maximale lengte of bepaalde patronen, die in de XML weergave moeten worden gerepresenteerd als 'facets', worden in het model opgenomen als 'tagged values'.

Categorieën van Logische Data Types (LDT's)

Vanuit het oogpunt van maximale lengte, als eigenschap van een attribuut, worden drie categorieën van Logische Data Types (LDT's) onderkend:

1. LDT's waarvan de lengte wordt bepaald door (inter)nationale standaardisering of regelgeving zoals W3C Date Time Format (ISO 8601), Ebix/ENTSO-E codes, EAN, MPM's/DPM's, issues NEDU, etc.;
2. LDT's waarvan de lengte wordt bepaald door functionaliteit. Voorbeelden hiervan zijn stand van telwerk, verbruik, meternummer, dossiernummer, NAW gegevens;
3. LDT's waarvan de lengte niet wordt bepaald door categorie 1 of 2. Voor deze attributen is de lengte veelal niet eenduidig bepaald of (nog) niet vastgesteld. Betreft doorgaans attributen met datatype "Text" en bevatten in de regel een omschrijving of een toelichting. Voorbeelden hiervan zijn tekstuele

toelichting op het resultaat van een aanvraag, tekstuele toelichting op foutmelding, tekstuele toelichting op het gebruik van een aansluiting.

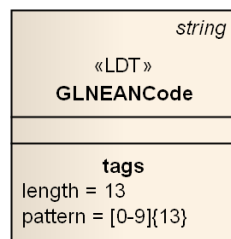
Deze LDT categorie zal in de servicebeschrijvingen bij de berichtdefinities worden getoond. Bij nieuwe servicebeschrijvingen zal deze categorie altijd worden getoond. Bij bestaande servicebeschrijvingen zal deze categorie alleen bij een update van de berichtdefinities worden getoond. Tevens zal deze LDT categorie worden getoond in de lijst van LDT attributen welke geregeld voor de sector wordt gepubliceerd op *mijnEDSN*.

Categorie 1 en 2 LDT's

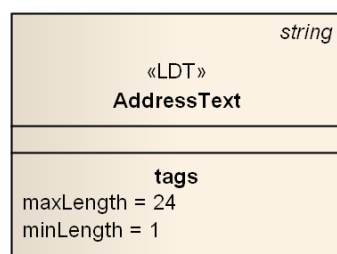
Voor berichten uitwisseling in de sector wordt de regel gehanteerd om de sanity-controles (lees: de syntax check van het bericht) zoveel mogelijk op de front-end (bij de voordeur) uit te voeren en daarmee het back-end (de verwerkende systemen) te ontlasten.

Categorie 1 en 2 LDT's zijn van belang voor de technische en functionele afhandeling van een XML bericht. Voor die categorie 1 en 2 LDT's wordt de maximale lengte, die voor elk specifiek categorie 1 en 2 LDT van toepassing is, gedefinieerd in de XML berichtdefinitie. Daarmee wordt de back-end zo veel mogelijk ontlast van onzinnige verwerking van niet-correcte berichten.

Deze maximale lengte voor categorie 1 en 2 LDT's wordt conform "W3C XML Schema Part 2: Datatypes Second Edition" als "length" of "maxLength" gedefinieerd in de XML berichtdefinitie. Indien de maximale lengte als eigenschap van een LDT altijd een vast gegeven is, dan zal "length" worden gehanteerd. Zie Figuur 14, hier is voor de EAN13 code als formaat-restrictie "length = 13" gedefinieerd. Indien de maximale lengte als eigenschap van een LDT altijd een variabel gegeven is, dan zal "maxLength" worden gehanteerd. Zie Figuur 15, hier is voor het adres als formaat-restrictie "maxLength = 24" gedefinieerd.



Figuur 14: Vaste lengte voor EAN13



Figuur 15: Variabele lengte voor adres

Door het definiëren van de maximale lengte voor categorie 1 en 2 LDT's in de XML berichtdefinitie, is men gehouden aan deze maximale lengte. Validatie van een XML bericht, met een categorie 1 of 2 LDT waarvan de maximale lengte niet conform de XML berichtdefinitie is, resulteert in een XML validatie fout en dus een afwijzing van het XML bericht door de front-end.

Een wijziging van de lengte van een categorie 1 of 2 LDT leidt tot een wijziging van het XML Schema, of XML Schema's, waarin dit LDT is opgenomen.

Categorie 3 LDT's

Categorie 3 LDT's zijn niet van belang voor de technische afhandeling van een XML bericht. Voor deze categorie 3 LDT's is de maximale lengte op 2000 karakters gezet ("maxLength = 2000").

Door het definiëren van deze maximale lengte voor categorie 3 LDT's in de XML berichtdefinitie, is men gehouden aan deze maximale lengte. Validatie van een XML bericht, met een categorie 3 LDT waarvan de lengte groter is dan 2000 karakters resulteert in een XML validatie fout en dus een afwijzing van het XML bericht door de front-end.

Een wijziging van een eigenschap van een categorie 3 LDT leidt niet tot een wijziging van het XML Schema (mits binnen de maximale lengte).

Datum

Datum notatie in de XML berichten moet conform de W3C DTF Note ISO 8601. Hierdoor wordt de eenduidige interpretatie van datum internationaal gewaarborgd.

Datum YYYY-MM-DD, bv: 2012-11-28. Geen tijdzone bij datum!

Datum/tijd

Datum/tijd notatie in de XML berichten moet conform de W3C DTF Note ISO 8601. Hierdoor wordt de eenduidige interpretatie van datum en tijd internationaal gewaarborgd.

Er zijn meerdere tijdsdefinities, maar meest gangbaar en binnen de berichten gebruikt is de UTC. Bij gegevensuitwisseling is een tijdsdefinitie zonder referentiepunt een incomplete tijdsdefinitie. De definitie van een datum/tijd is pas compleet en ondubbelzinnig met een offset van UTC:

```
2011-03-31T10:15:30+02:00 (UTC offset van +02:00 uur);
2011-03-31T08:15:30Z (UTC) .
```

De regel is om altijd de offset tot UTC verplicht te stellen. Het verplicht maken van de offset ten opzichte van UTC geeft de ontvangende partij de mogelijkheid zonder misinterpretatie over zomer/wintertijd en locatie de datum/tijd te verwerken.

De notatie van datum/tijd conform de W3C DTF Note ISO 8601 is vastgelegd in TC021.

Eenduidige datum/periode bepaling in berichten

In IC048 is eenduidig vastgelegd hoe datumvelden in berichten geïnterpreteerd dienen te worden en qua verwerking in de diverse ketenprocessen en interne systemen dienen te worden behandeld, onder meer om te voorkomen dat als gevolg van verschillende interpretaties er in een aantal gevallen een dag teveel of een dag te weinig aan een partij wordt toegerekend (zoals bij afrekeningen en allocatie).

Concreet: Een startdatum is een "vanaf"-datum is (inclusief de datum zelf). Een einddatum is een "tot"-datum (exclusief de datum zelf).

Bij een periode, die gemarkeerd wordt door een begin- en einddatum, is de dag van de begindatum de eerste dag die tot de periode behoort. De dag voorafgaand aan de genoemde einddatum is de laatste dag die tot de periode behoort.

Bv. het uur van 15.00 - 16.00 (UTC) op 27 februari 2018:

startdatum, de "vanaf"-datum:	2018-02-27T15:00Z
einddatum, de "tot"-datum:	2018-02-27T16:00Z

Bv. de gehele maand februari 2018:

startdatum, de "vanaf"-datum:	2018-02-01T00:00Z
einddatum, de "tot"-datum:	2018-03-01T00:00Z

De strekking van dit issue is inmiddels ook in de MPM's en DPM's verwerkt.

4.6 Berichten

Een tussen applicaties uit te wisselen document wordt in XML gerepresenteerd als een bericht. Iedere berichtdefinitie wordt gemodelleerd als een UML class model. Deze berichtdefinities worden opgeslagen in packages met als stereotype CCMA (Core Component Message Assembly). Zij worden met de LUXE-tool op basis van het LOM gegenereerd. Een bericht is een afgeleide subset van het betreffende LOM. Nadat een CCMA gegenereerd is, zijn aan de objecten, attributen en associaties stereotypes toegekend.

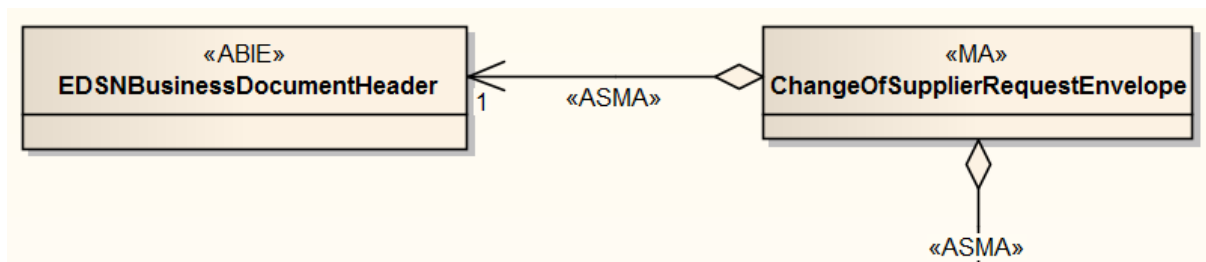
De root van een bericht is gestereotypeerd als «MA» (Message Assembly). Objecten in het bericht zijn gestereotypeerd als «ABIE» (Aggregate Business Information Entities), attributen als «BBIE» (Basic Business Information Entities) en associaties als «ASMA» (ASsociation Message Assembly) of «ASBIE» (ASsociation Business Information Entities). Bovendien zijn voor de associaties de richting en cardinaliteiten aangegeven.

Indien informatie impliciet is aan de processtep, wordt die informatie niet expliciet in een attribuut weergegeven. Sommige berichten zijn met betrekking tot structuur en inhoud procesonafhankelijk. Deze berichten hebben dan "Proces ID" attribuut, (of zoals de Engelse term luidt: Process Type) voor het opnemen van procesinformatie. Bijvoorbeeld bij stamgegevens:

- opleveren stamgegevens op aanvraag;
- opleveren stamgegevens als gevolg van een wijziging door een netbeheerder;
- opleveren stamgegevens als gevolg van een wijziging door een marktproces.

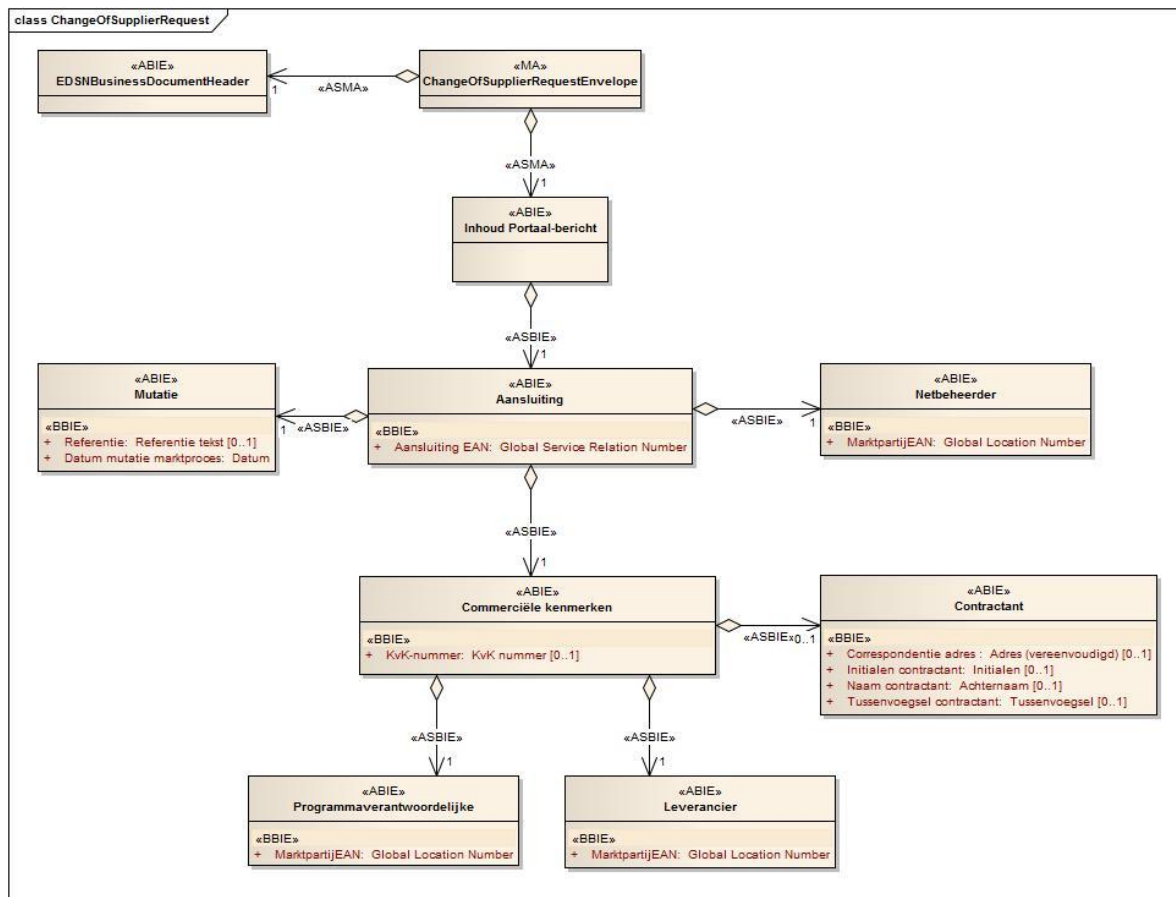
4.7 Enveloppering

Ieder document is als "Envelope" met als stereotype "MA" gedefinieerd, zie Figuur 16. In XML-termen is de envelop het root-element van het bericht (afgezien van de SOAP envelop, die daar weer omheen zit). De envelop bevat de EDSN Business Document Header, waarin functionele header-attributen zijn opgenomen (in tegenstelling tot de SOAP Header, die technische attributen bevat). Enveloppen zijn genoemd naar transactie-activiteiten.



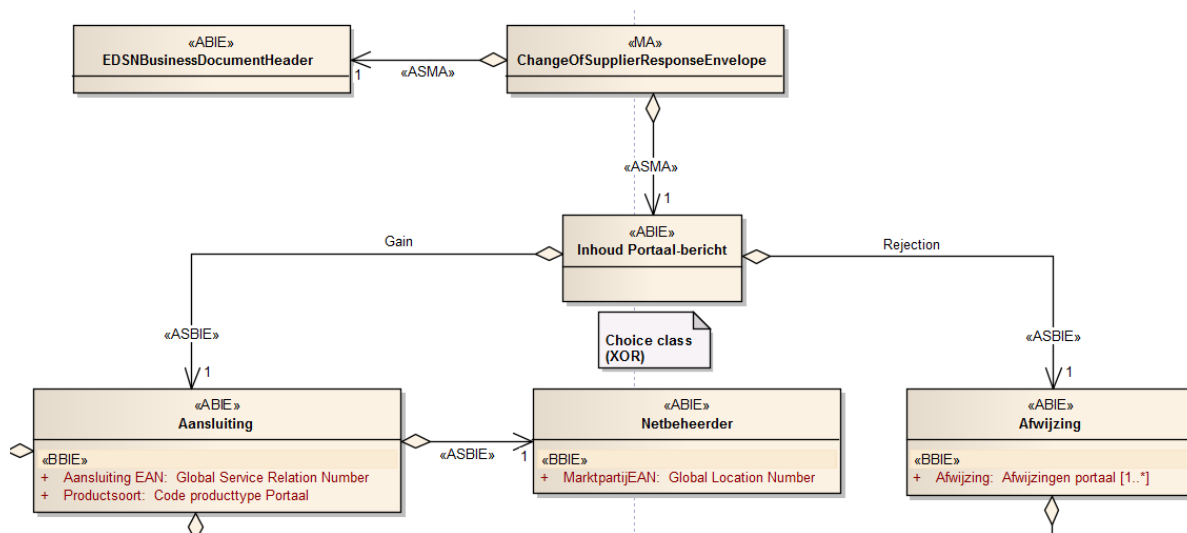
Figuur 16: Enveloppering ChangeOfSupplier

De envelop is geassocieerd met een of meer inhoudelijke objecten (ABIEs), die in de meeste gevallen subsets zijn van (gebaseerd zijn op) de objecten in het LOM. Zo verwijst een envelop in veel gevallen naar een subset van Aansluiting (MeteringPoint), zoals bijvoorbeeld de envelop van een "ChangeOfSupplier" switchbericht. Zie Figuur 17.



Figuur 17: ChangeOfSupplier switchbericht

Een webservice handelt een interactie tussen twee partijen af en kent een requesting en een responding activity. Iedere activity bevat een eigen envelop. Wanneer bij de responding activity een keuze wordt gemaakt (bijvoorbeeld: “success” of “failure”), wordt deze op XSD (webservice) niveau als “XSD:Choice” toegevoegd. Zie Figuur 18.



Figuur 18: ChangeOfSupplier responding activity met choice

4.8 Tagged values

Aan de berichtdefinities, gemodelleerd als een UML class model en gestereotypeerd als CCMA, worden kenmerken toegekend zoals versie, datum, status, organisatie. Deze kenmerken zijn als zgn. “tagged values” aan elke CCMA toegevoegd.

De kenmerken worden gebruikt voor de XML namespace. De XML namespace is een W3C aanbeveling voor toevoegen van context aan XML elementen en zorgt ervoor dat de naam van een object in een berichtdefinitie uniek is (ook over versies van het object). In paragraaf 7.2 wordt de XML namespace nader toegelicht.

Onderstaande het overzicht van “tagged values” en de toepassing/functie van elke “tagged value”.

Tagged value in CCMA	Waarde	Toepassing/functie
coreCompenentsDocumentation	True of False	Wel / niet opnemen van CCTS metadata conform de CCTS specificatie in de XML berichtdefinitie (XSD): - Core Component Registry Class: - VersionID - UniqueID - Core Component Common Information: - DictionaryEntryName - Definition - BusinessTermName Het opnemen van CCTS metadata is de standaard oplevering. Heeft geen functie voor de XML namespace.
hierarchy	esdn	Onderdeel van de namespace
majorVersion	0..9	In version attriboot van het <xsd:schema element>, eveneens als version informatie in de XML namespace. Als “Schema version/revision” informatie in XML Schema.
minorVersion	0..9	In version attriboot van het <xsd:schema element>. Als “Schema version/revision” informatie in XML Schema. Heeft geen functie voor de XML namespace.
namespaceScheme	0 of 1	0 = namespace als URL 1 = namespace als URN XML namespace definitie wordt standaard als URN opgenomen in de XML Schema’s. Dit is conform het advies van UN/CEFACT [UNDR].
organization	nedu	Onderdeel van de XML namespace. Als “Schema agency” informatie in XML Schema.
publishingDate	[yyyy-mm-dd]	Als “Schema date” informatie in XML Schema. Heeft geen functie voor de XML namespace.
revision	0..9	Als “Schema version/revision” informatie in XML Schema. Heeft geen functie voor de XML namespace.

Tagged value in CCMA	Waarde	Toepassing/functie
semanticAnnotations	True of False	Wel / niet opnemen van semantische annotaties (W3C Recommendation 28 August 2007 - Semantic Annotations for WSDL and XML Schema). Het opnemen van semantische annotaties is de standaard oplevering. Heeft geen functie voor de XML namespace.
status	draft of standard	Onderdeel van de XML namespace.

4.9 Services

Een bericht is onderdeel van een Service of transactie. Een Service is in het model een package dat als «Service» is gestereotypeerd. Een Service kan worden geïmplementeerd als Webservice.

Services worden ingedeeld naar variatie van de vorm (patroon, synchroon, asynchroon), niet naar variatie van de inhoud. Een andere periode, granulariteit, status, versie, (bron)systeem, proces of eenheid van de informatie is geen reden om een andere service te maken; een ander doeldomein is dat wel. Elke service heeft één doeldomein (en dus gebaseerd op één informatiemodel) en het gebruik van het afgeleide bericht blijft beperkt tot dat doeldomein.

Een service moet schaalbaar zijn. Informatie moet in principe d.m.v. parameters kunnen worden geselecteerd, of in delen (per “record”) kunnen worden opgevraagd.

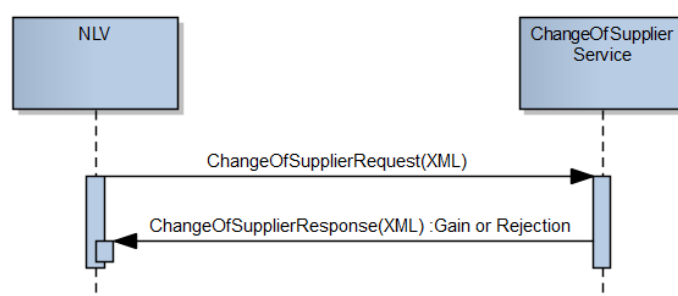
Services worden gemodelleerd volgens een patroon. Voor berichtinteracties onderkent UN/CEFACT verschillende patronen:

- Notification / Acknowledgement
- Information Distribution
- Commercial Transaction
- Request / Confirm
- Request / Response
- Query / Response

Een transactie patroon kent twee acties, een vragende actie en een reagerende actie. Elke actie wordt uitgevoerd door een geautoriseerde partij.

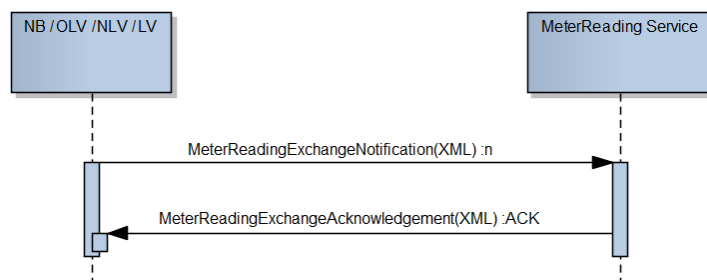
De meeste service transacties zijn als “Request-Response” patroon gedefinieerd. Enkele transacties zijn als “Notification-Acknowledgement” gedefinieerd.

Het “Request-Response” patroon wordt gebruikt om een transactie te starten of om informatie op te vragen. Veelal bevat het “Request” bericht inhoudelijke gegevens, soms alleen een identificatie van de aansluiting waarover informatie wordt opgevraagd en soms alleen proces-administratieve gegevens zoals datum-tijd en referentienummer, indien die samen met het berichttype voldoende duidelijk maken welke response wordt verlangd of welke gegevens worden opgevraagd. Het “Response” bericht bevat soms inhoudelijke gegevens en altijd administratieve gegevens. Het “Response” bericht kan een afwijzing bevatten van het request (met de reden van afwijzing). Het “Response” bericht dient tevens als ontvangstbevestiging voor het “Request” bericht.



Figuur 19: “Request-Response” patroon

Het “Notification-Acknowledgement” patroon wordt gebruikt wanneer de initiërende partij informatie aanlevert die ter kennisgeving of ter verwerking naar de ontvanger wordt gezonden, en alleen een bevestiging, het “Acknowledgement” bericht, van die verwerking (of van de ontvangst) terug verwacht.



Figuur 20: “Notification-Acknowledgement” patroon

4.10 Asynchrone Batchverwerking

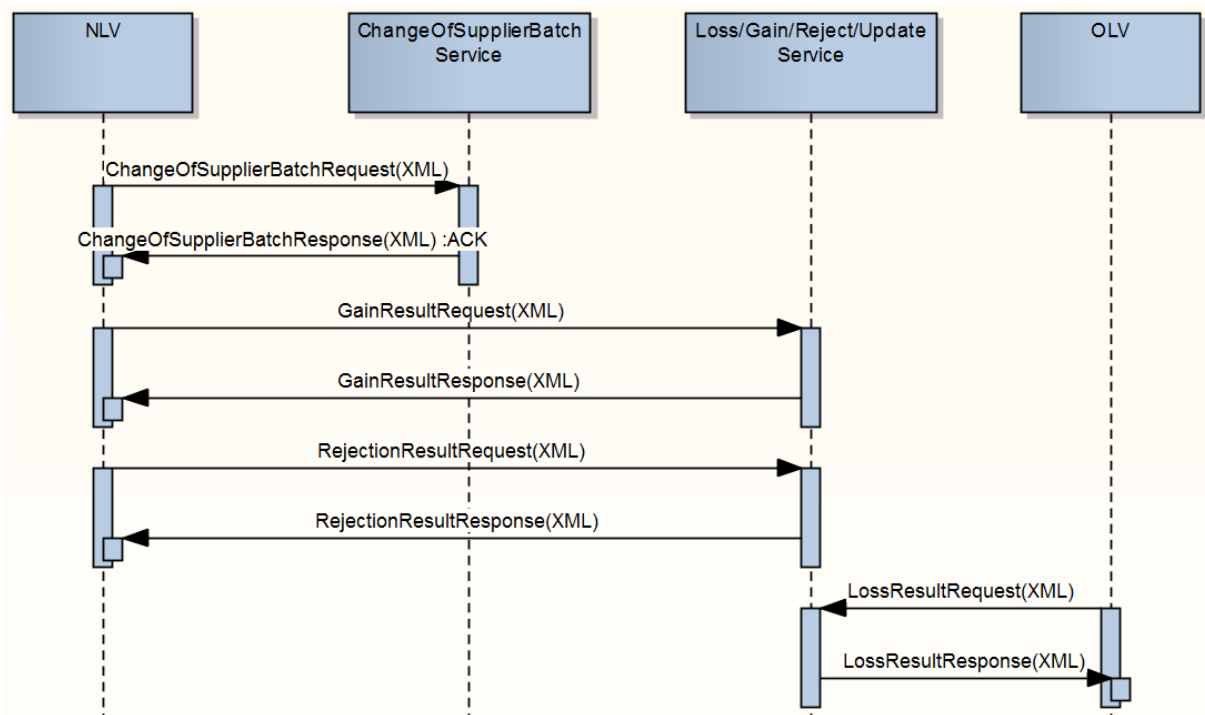
Voor gevallen waarin verzoeken om informatie of verwerking niet direct afgehandeld kunnen of hoeven worden zijn (functioneel asynchrone) webservices gedefinieerd voor batchverwerking. In een aanvragende transactie wordt de verwerking of de informatie aangevraagd. Het synchrone responsebericht dat volgt op de aanvraag functioneert dan uitsluitend als ontvangstbevestiging. Het functionele antwoord op de aanvraag wordt vervolgens (normaal binnen een werkdag) verkregen middels een tweede webservice transactie die expliciet om de responses vraagt. De aanvrager neemt dan (opnieuw) het initiatief om de antwoorden op te vragen. Een batch kan meerdere aanvragen van (in principe) hetzelfde type (of proces) bevatten, zoals switches voor verschillende aansluitingen.

Het functionele antwoord op een aanvraag kan positief of negatief zijn. Beide kunnen in aparte webservices worden aangeboden, of gebundeld in één webservice. Gelijksortige antwoorden (bijvoorbeeld Gains) kunnen ook zijn gebundeld over verschillende soorten aanvragen (bijv. Supplier Switch en Inhuizing). Als regel geldt dat een batch gelijksortige berichten bevat. De functionele antwoorden, op aanvragen die in één request batch zijn aangeboden, hoeven niet middels één volgende response batch opgeleverd te worden. De antwoorden worden individueel teruggegeven, waardoor een response batch functionele antwoorden op meerdere aanvragen kan bevatten (en soms ook van meerdere soorten aanvragen, zoals bij de Gain en Loss antwoorden).

Als voorbeeld kan dienen de batchverwerking van leveranciersswitches. Allereerst kan een aantal switchverzoeken in één envelop worden ingediend. De totale omvang van een dergelijk batchverzoek is gemaximeerd (momenteel tot 5 MB). Bij grotere omvang dienen meerdere batchverzoeken te worden ingediend.

Het synchrone antwoord op ingediende batchverzoeken bestaat uit een (vrij technische) ontvangstmelding van de batch (de melding dat de switch-meldingen in goede orde en verwerkbaar zijn ontvangen – en vervolgens asynchroon zullen worden verwerkt - of als afwijzing omdat de hele batch om de opgegeven reden(en) niet kan worden geaccepteerd voor verwerking).

Bij positieve respons kunnen (enige tijd bijvoorbeeld een halve dag later) zowel de zogenaamde Gains en Losses worden opgehaald die het gevolg zijn van de succesvolle verwerking van de leveranciersswitches als ook de Rejections van (functioneel) ongeldige verzoeken. Gains, Losses en Rejections zijn van verschillend type, dit wordt, in dit voorbeeld van batchverwerking van leveranciersswitches, middels drie verschillende webservice operaties geregeld.



Figuur 21: Batchverwerking

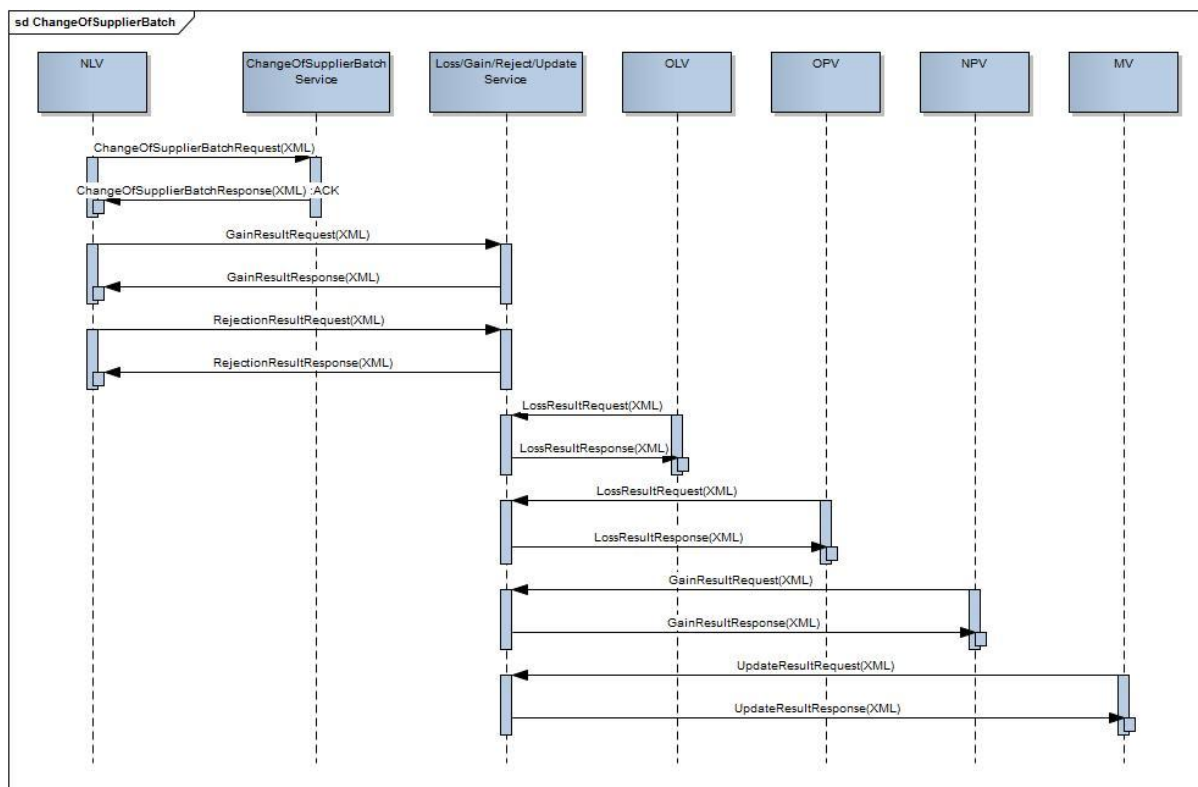
Indien de omvang van de Gain, Loss en/of Rejection, als resultaat van de batchverwerking, de omvang van 5 MB overschrijdt, zal de webservice de relevante berichten (GainResultResponse, RejectionResultResponse en/of LossResultResponse) beperken tot 5 MB. De aanvrager neemt dan (opnieuw) het initiatief om de resterende resultaten van de batchverwerking op te vragen door het (opnieuw) insturen van een GainResultRequest, RejectionResultRequest en/of LossResultRequest.

4.11 Processen

Webservice transacties, zoals leveranciersswitch, vinden plaats binnen een bepaald proces, zoals de Leveranciersswitch. Welke transacties in welke volgorde in de verschillende processen in de Nederlandse energiesector worden uitgevoerd (en dus welke webservices worden aangeroepen) is beschreven in de betreffende Detail Proces Modellen (DPM's), Functionele Ontwerpen of Issues, en tevens gespecificeerd in bijhorende Servicebeschrijvingen.

In het model is dit aspect van de webservices opgenomen in zogenaamde sequentiediagrammen. In sequentiediagrammen worden de activiteiten voor de verschillende actoren (marktrollen) in het procesverloop, zoals die in de procesbeschrijvingen staan beschreven, afgebeeld op de te gebruiken webservices.

In Figuur 22 staat een voorbeeld van een sequentiediagram.



Figuur 22: Sequentiediagram Leveranciersswitch

5 Technische berichtenarchitectuur

5.1 Communicatieprotocol

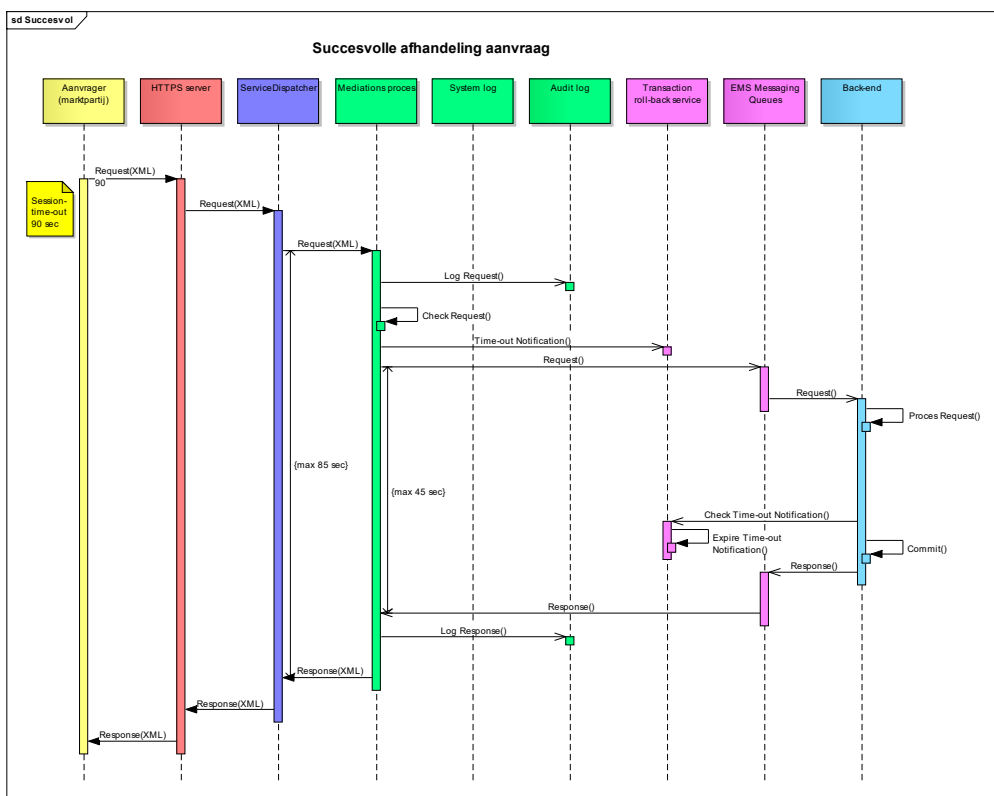
Het communicatieprotocol voor de interfaces met de gecentraliseerde systemen is gebaseerd op HTTPS 1.1. De berichten worden op basis van SOAP 1.1 [SOAP] uitgewisseld. De beschrijving van de webservices wordt gedaan met behulp van WSDL 1.1 [WSDL]. Voor iedere berichtuitwisseling wordt een operatie (WSDL operation) beschreven. Zo'n operatie is onderdeel van een WSDL. Operaties worden binnen de WSDL gegroepeerd binnen een zogenaamd porttype (WSDL portType).

Elk porttype heeft een unieke naam om in de binding (WSDL binding), waar het transport protocol wordt gedefinieerd, te kunnen verwijzen naar het porttype en de operatie van die betreffende porttype. In de service definitie van de WSDL wordt de binding en URL (het adres) als service gedefinieerd. Een WSDL kan meerdere services bevatten.

De WS-I.org, een organisatie die zich bezig houdt met de standaardisering van webservices gebruik, heeft een aantal richtlijnen vastgelegd van gedefinieerde keuzes binnen de mogelijkheden van de diverse WS-* standaarden⁴. Zowel voor berichtenuitwisseling als gebruik van WS-Security aspecten zijn zogenaamde implementatierichtlijnen in de vorm van "basic profiles" gedefinieerd. T.a.v. het gebruik van HTTPS, SOAP en WSDL wordt in lijn met Basic Profile 1.1 Second Edition gewerkt [BPSE].

5.2 Synchroniteit en batch verwerking

Alle webservices zijn service-technisch gezien synchrone services. Dat wil zeggen dat er altijd direct een antwoord (technisch (een SOAP Fault) of functioneel (een XML response)) komt op de aanvraag. Figuur 23 betreft een sequentiediagram met een XML response op de aanvraag.

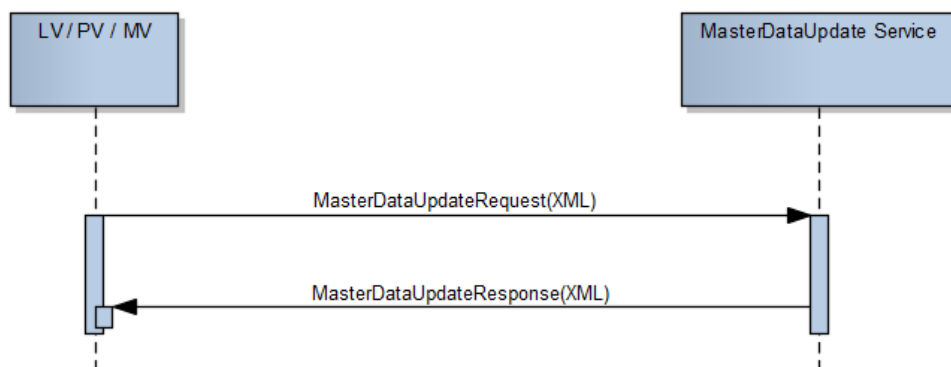


Figuur 23: Sequentiediagram succesvolle afhandeling aanvraag

⁴ WS-ReliableMessaging en WS-Addressing worden niet toegepast.

Logisch-functioneel gezien is er onderscheid tussen synchrone en asynchrone verwerking van berichten. Bij synchrone verwerking worden de berichten direct functioneel verwerkt in de achterliggende, verwerkende applicatie(s). Het resultaat van de verwerking wordt teruggegeven in dezelfde synchrone service. Bij asynchrone (of 'batch') verwerking worden berichten 'geparkeerd' en (iets) later functioneel verwerkt. In de service wordt (na acceptatie-controle, zoals de syntaxcheck) alleen een 'ontvangstbevestiging' synchroon teruggegeven. Het resultaat van de functionele verwerking wordt later klaargezet en moet door de organisatie, die het bericht heeft aangeleverd, actief opgehaald worden. Hiertoe wordt opnieuw een synchrone service aangeroepen.

Asynchrone verwerking wordt toegepast indien meervoudige berichten of batches (zoals meerdere switchverzoeken) tegelijkertijd worden aangeboden ter verwerking. In sommige gevallen kunnen echter ook enkelvoudige berichten asynchroon worden verwerkt. Voor de meeste webservices zullen er zowel enkelvoudige als batch varianten beschikbaar worden gesteld. Sommige webservices kennen (zoals voor het ophalen van informatie) geen asynchrone varianten. Bijvoorbeeld opvragen gewijzigde stamgegevens van aansluitingen, zie Figuur 24. In het synchrone antwoord (MasterDataUpdateResponse) op het ingediende verzoek (MasterDataUpdateRequest) kunnen de gewijzigde stamgegevens van 0 of meer aansluitingen worden geretourneerd.

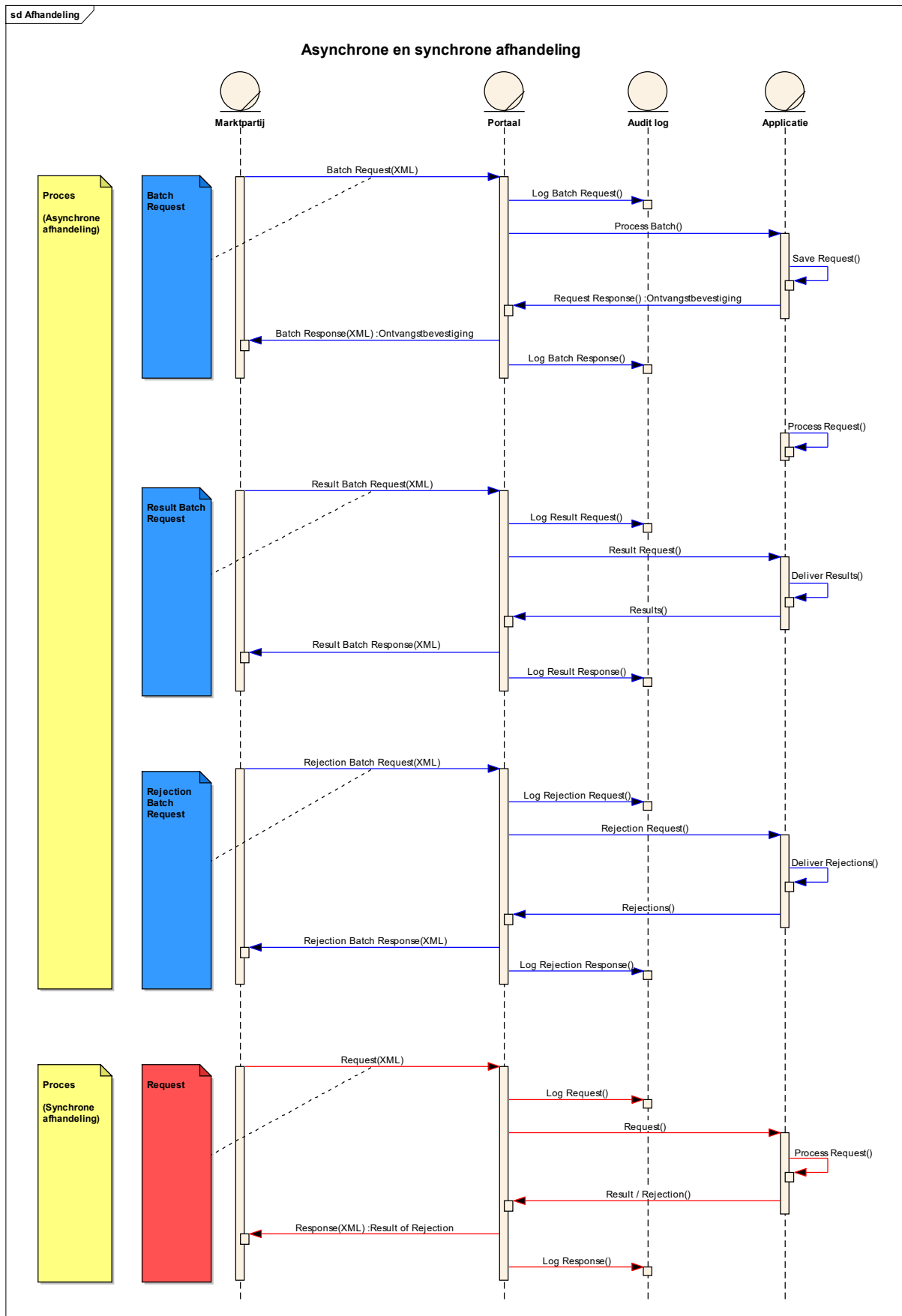


Figuur 24: Opvragen gewijzigde stamgegevens van aansluitingen

In de meeste gevallen is er een aparte webservice voor het ophalen van (positieve) resultaten en een aparte webservice voor het ophalen van afwijzingen en functionele foutmeldingen.

In Figuur 25 is met behulp van een sequentiediagram aangegeven hoe de asynchrone (blauw) en de synchrone (rood) verwerking eruit ziet.

Sommige services worden gebruikt voor het uitwisselen van berichten tussen marktpartijen. In die gevallen wordt er geen informatie uitgewisseld met EDSN systemen, maar worden berichten 'geparkeerd' op het Portaal om opgehaald te worden door de ontvangende marktpartij. De functie van het Portaal beperkt zich voor deze services tot een 'brievenbus'.



Figuur 25: Asynchrone en synchrone verwerking sequentiediagram

5.3 Transactionaliteit

Transactionaliteit gaat over het volledig verwerkt zijn van operaties op een doelsysteem (zoals in geval van ketenprocessen op C-AR tot en met de gebruikte database).

Van iedere webservice-operatie geserviceerd door EDSN wordt zowel de aanvraag als het antwoord gelogd. Dit loggen gebeurt door het Portaal, buiten de database transactie op de applicatie. Voor enkelvoudige, synchrone berichten wordt (bijvoorbeeld binnen het aansluitingenregister) de aanvraag meteen volledig transactioneel verwerkt. Dus de aanvraag wordt of geheel verwerkt of in z'n geheel niet verwerkt (lees: afgewezen).

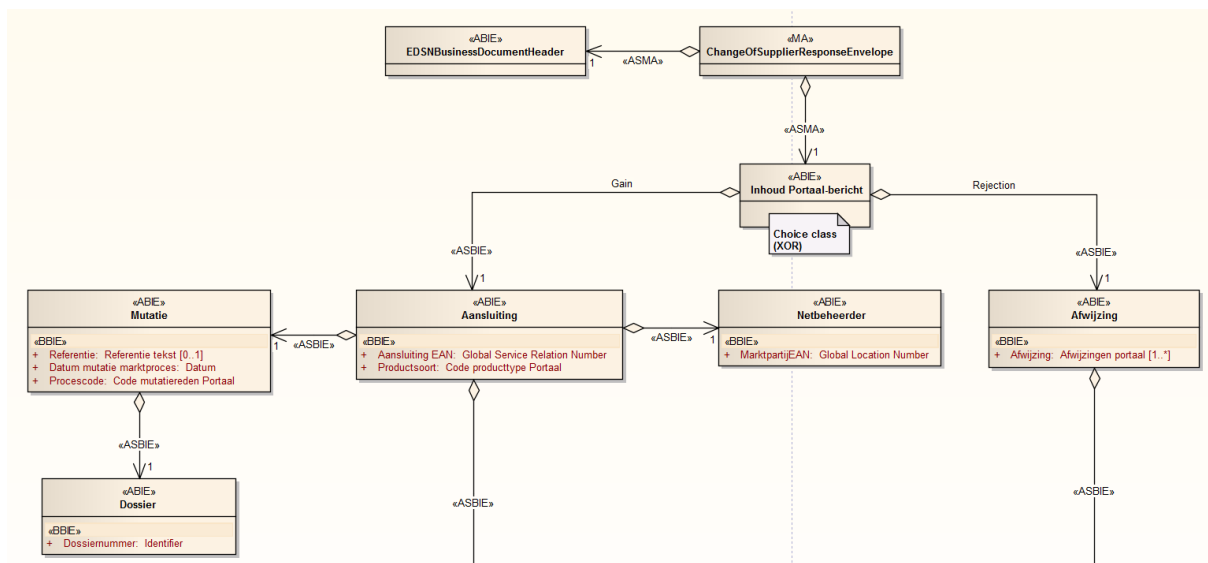
Bij batches met meervoudige berichten wordt ieder functioneel bericht in de batch later afzonderlijk transactioneel verwerkt door de betreffende applicatie. Voordat de berichten aan de applicatie worden aangeboden wordt in het acceptatieproces van de batch eerst een XML Schema validatie uitgevoerd. Bij de XML Schema validatie wordt de batch in z'n geheel geaccepteerd of afgekeurd voor verdere verwerking. Dit wordt synchroon terug gemeld in de aanbiedende service.

Het verwerken van de functionele berichten uit de batch in de betreffende applicatie vindt vervolgens (asynchroon) separaat plaats. De enkelvoudige berichten uit een batchbericht worden één voor één als transactie verwerkt. Per individueel functioneel bericht komt er een antwoord (logische afwijzing / OK bericht). De antwoorden kunnen, weer als batch, middels een separate aanroep worden opgehaald. De antwoordbatch kan antwoorden op meerdere ingediende batch verzoeken bevatten.

Portaal dossier ID

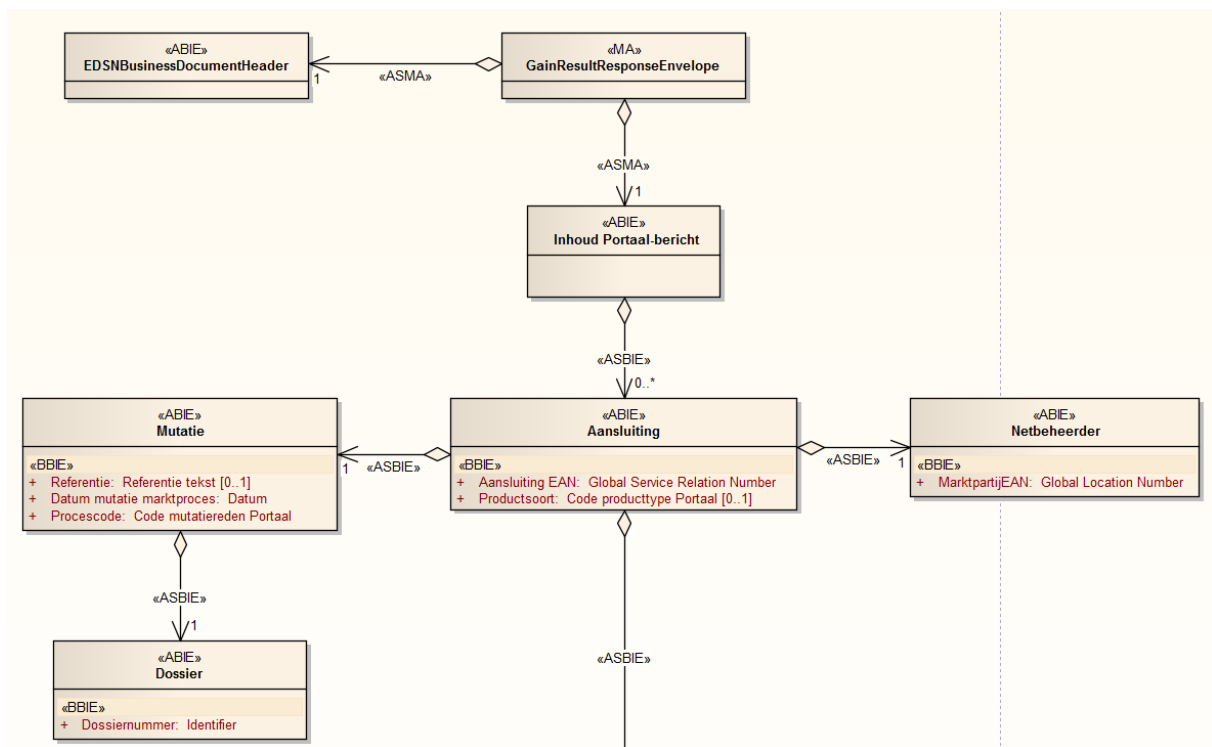
Het Portaal identificeert alle berichten die binnen een transactie worden uitgewisseld met een Portaal dossier ID. Dit Portaal dossier ID wordt teruggegeven in het eerste antwoord nadat een bericht ter initiatie van een proces is ingestuurd. In een synchrone webservice is dat het synchrone antwoord. In batchprocessen zit het antwoord in het functionele response bericht, dat (later) opgehaald moet worden. Het Portaal dossier ID kan worden gebruikt voor de traceerbaarheid van de transacties en mutaties en om de datering van mutaties van de transactie in te zien. Het Portaal dossier ID is gerelateerd aan MeteringPoint (Aansluiting).

In een synchrone webservice waarbij er een transactie heeft plaatsgevonden op één MeteringPoint (Aansluiting), wordt het Portaal dossier ID geretourneerd in de synchrone response. Zie Figuur 26.



Figuur 26: Portaal dossier ID (Dossiernummer) geretourneerd in de synchrone response

In batchprocessen wordt in de response (bv. Gain) per individueel bericht uit de batch (dus meestal per MeteringPoint (Aansluiting)) een Portaal dossier ID geretourneerd. Zie Figuur 27.



Figuur 27: Portaal dossier ID (Dossinummer) geretourneerd, per MeteringPoint (Aansluiting), in de GainResultResponse

5.4 Idempotentie

Soms kan zich de situatie voordoen dat een transactie uitgevoerd wordt, zonder dat de aanvrager het uiteindelijke resultaat (correct) ontvangt. Dit kan gebeuren als bijvoorbeeld een antwoord van een webservice later komt dan de afgesproken tijdslimiet, of dat zich tussen transport niveau (HTTP en/of TCP/IP) en de applicatie logica (in een gedistribueerde omgeving) fouten voordoen die niet binnen een transactie kunnen worden afgehandeld. Daartoe is er een mechanisme geïmplementeerd dat zorgt dat in dergelijke gevallen de “berichten niet verdwijnen”, maar alsnog kunnen worden opgevraagd.

De aanvrager zal in zo’n geval dezelfde transactie nogmaals moeten initiëren. Normaal gesproken rijst dan de vraag wat er met die herhaalde transactie zou moeten gebeuren. Nogmaals verwerken, een foutmelding geven (‘is al verwerkt’), verwerken omdat deze nooit eerder was aangekomen of hetzelfde antwoord nogmaals geven? In het laatste geval wordt gesproken van idempotentie.

Idempotentie is vooral van belang bij transactie patronen waarbij de ontvangende partij een Create, Delete of Update aanvraag moet afhandelen. Immers, een herhaalde (identieke) aanvraag voor een Create, Delete of Update kan leiden tot ongewenste situaties in de business applicatie(s) van de ontvangende partij.

Idempotentie wordt ondersteund op het niveau van de webservice, het niveau waarop over HTTP(S) in SOAP enveloppen verpakte berichten (requests, responses, notifications en acknowledgements) worden uitgewisseld. Idempotentie speelt binnen een synchrone webservice. Om idempotentie in algemene termen te kunnen ondersteunen, zijn er twee voorwaarden:

1. een webservice aanvraag moet herkend worden als ‘duplicaat’ (duplicate detection);
2. het oorspronkelijke antwoord moet zijn opgeslagen.

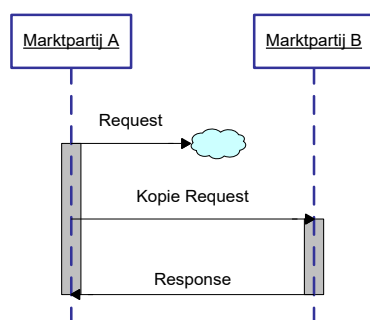
Door in de header van het bericht een uniek Message ID mee te geven kan aan de eerste voorwaarde worden voldaan. De Message ID in de header is daarom verplicht. Het veld RepeatedRequest dat voorkomt in de header moet niet worden meegegeven. Dit veld heeft, doordat idempotentie is geïmplementeerd conform de twee gestelde voorwaarden, geen functie in ondersteuning van idempotentie.

Aan de tweede voorwaarde kan worden voldaan door bij het ontwerp en ontwikkeling van de business applicatie rekening te houden met de eis dat een antwoord op een aanvraag enige tijd (minimaal één dag) moet worden opgeslagen. Dit is een aspect van de Quality of Service van de webservice die de aanvraag moet afhandelen.

In twee herkenbare situaties speelt idempotentie een rol.

Situatie 1

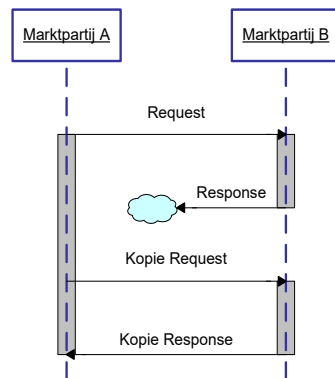
Een request van marktpartij A wordt niet door marktpartij B ontvangen (zie Figuur 28). Marktpartij A, die geen ontvangstbevestiging heeft ontvangen, kan concluderen dat Marktpartij B de request mogelijk niet heeft ontvangen. Hij zal na enige tijd (bijvoorbeeld minimaal 120 seconden) opnieuw een kopie van de request (met zelfde Message ID) uitsenden. Marktpartij B krijgt de aanvraag maar één keer en heeft dus niet te maken met idempotentie. Marktpartij A moet zich ervan bewust zijn dat een kopie van de request is gestuurd (en als er alsnog een antwoord komt op de oorspronkelijke aanvraag mag dit niet tweemaal worden verwerkt).



Figuur 28: Idempotentie bij vrager: Request was niet ontvangen

Situatie 2

Een response van marktpartij B wordt niet door marktpartij A ontvangen (zie Figuur 29). Marktpartij A zal net als in situatie 1 concluderen dat de request mogelijk niet door Marktpartij B is ontvangen. Marktpartij A zal daarom na enige tijd (bijvoorbeeld minimaal 120 seconden) opnieuw een kopie van de request uitsenden. Idempotentie is voor deze situatie wel van belang voor marktpartij B. De kopie request moet door marktpartij B wel op identieke wijze worden beantwoord, maar niet functioneel voor de tweede keer worden verwerkt in de backend.

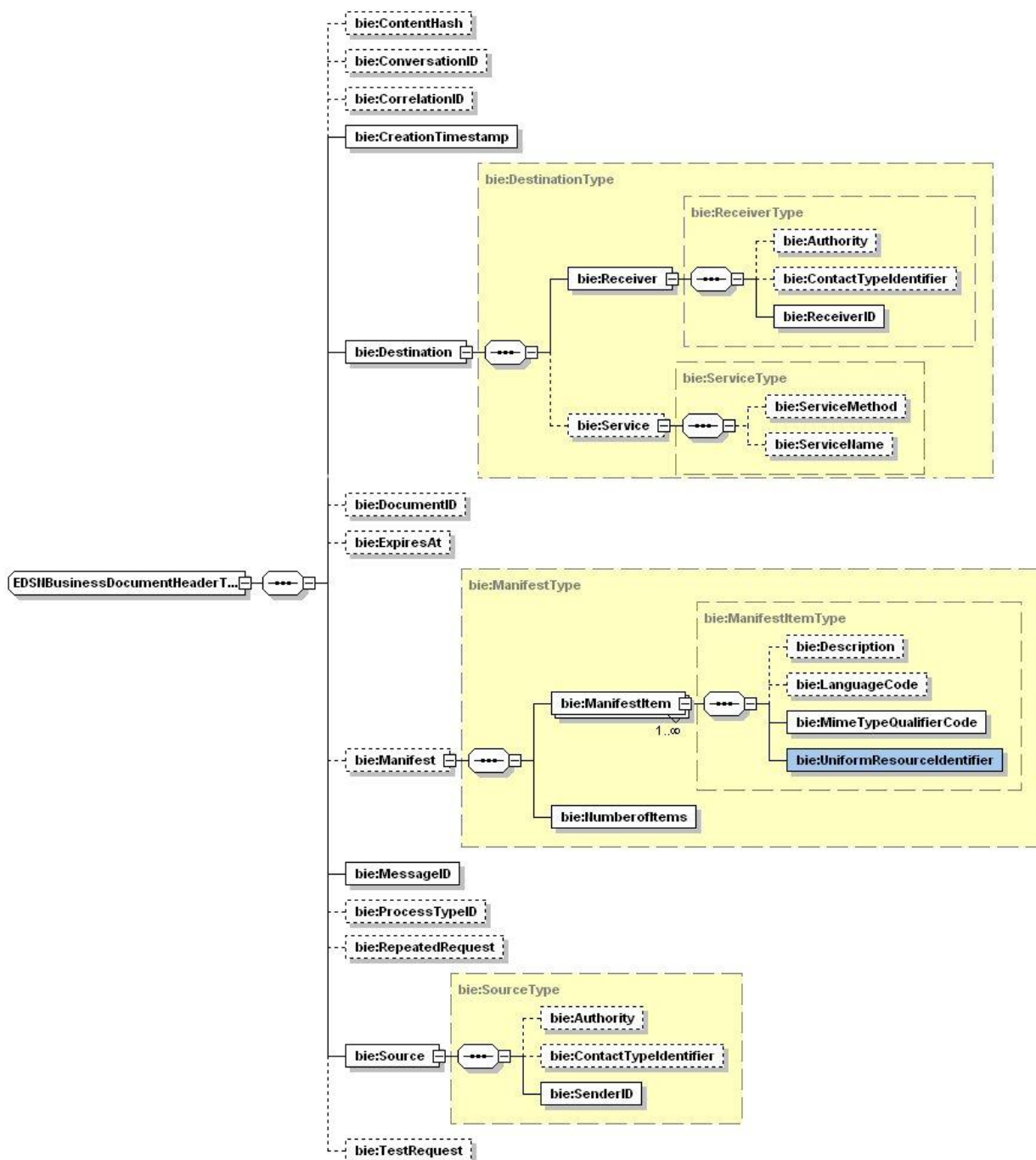


Figuur 29: Idempotentie bij responder: Response was niet ontvangen

Deze faciliteit is primair bedoeld voor technisch opnieuw proberen (retry). Verzonden berichten moeten daarom door de afzender minimaal een afgesproken periode worden opgeslagen om opnieuw verzonden te kunnen worden.

5.5 EDSN Header

De EDSN Business Document Header is gestandaardiseerd voor alle EDSN Business Document definities. De EDSN Business Document Header is gebaseerd op de OAGIS 9.x ApplicationArea, de UN/CEFACT Standard Business Document Header, de GS1 Standard Business Document Header en de Gasunie Business Document Header.



Figuur 30: EDSN Business Document Header

In de EDSN Business Document Header zijn:

- MessageID, CreationTimestamp, ReceiverID (onderdeel van Destination) en SenderID (onderdeel van Source) verplicht;
- overige velden zijn optioneel en worden alleen gebruikt indien er tussen de partijen in de communicatieketen duidelijke afspraken zijn gemaakt en gedocumenteerd over het gebruik van die betreffende velden.

Identifiers

De volgende identifiers worden toegepast in de EDSN Business Document Header om de diverse business document instanties te kunnen identificeren in de context van een business proces waarin deze business document instanties worden uitgewisseld.

Identifier	Omschrijving
ConversationID	De ConversationID kan gevuld worden door de aanvrager. Wordt een bericht binnen de communicatieketen doorgestuurd, dan blijft het ConversationID intact. Indien een bericht opnieuw wordt verzonden, dan wordt het oorspronkelijke ConversationID meegezonden. Wanneer bij een proces verschillende marktpartijen betrokken zijn, hebben alle berichten tussen die partijen die deel uitmaken van dat functionele proces dezelfde ConversationID. De ConversationID is optioneel. Wordt toegepast bij georchestreerde business processen waarbij meerdere partijen betrokken zijn. De ConversationID kan pas worden gebruikt als er in de keten duidelijke afspraken over zijn gemaakt. In de context van het Portaal en C-AR worden processen geïdentificeerd door een uniek Transactiedossinummer, dat wordt toegekend door het centrale systeem. Dit Transactiedossinummer wordt in alle berichten behorende bij het proces teruggekoppeld.
CorrelationID	De CorrelationID kan gevuld worden door de aanvrager. Indien gevuld, dient de ontvangende partij (ook een volgende knooppunt is een partij) dit CorrelationID te vermelden in antwoorden aan zijn aanvrager. Dit zorgt ervoor dat de aanvrager altijd antwoord bij de juiste vraag kan passen. Merk hierbij op dat batchgewijze verzoeken met een webservice synchroon antwoord van ontvangst geven (met hetzelfde CorrelationID), maar dat de functionele antwoorden van de verschillende verzoeken in de batch alle als onafhankelijke antwoorden in verschillende opvragingen worden opgehaald zonder vermelding van de CorrelationID uit het verzoek.
DocumentID	Dit is een identifier om een bericht (business document instantie) op functioneel niveau te identificeren. De inhoud van deze identifier is door de zender vrij te definiëren. De DocumentID is uniek voor ieder bericht. Deze identifier is optioneel.
ProcessTypeID	Dit is een identifier om een bericht (business document instantie) op proces niveau te identificeren. Deze identifier is optioneel. Door gebruik te maken van deze ProcessTypeID voor alle berichten die van toepassing zijn voor een bepaald proces kan door "business activity monitoring" inzichtelijk worden gemaakt hoe de afhandeling van de berichten voor dat proces verloopt. De ProcessTypeID kan pas worden gebruikt als er in de keten duidelijke afspraken over zijn gemaakt.
MessageID	Elk bericht tussen twee partijen heeft een uniek MessageID (elk knooppunt in de communicatieketen dient hierbij als partij beschouwd). De MessageID is hierbij de technische identificatie van een bericht. Als een (functioneel) bericht meerdere schakels in een keten doorloopt, dan krijgt dit bericht telkens een nieuw MessageID tussen elke twee knooppunten. Echter, indien een bericht wordt herhaald omdat er geen reactie is ontvangen op een eerder gestuurd bericht, moet de zender hetzelfde MessageID gebruiken van het eerder gestuurde bericht (zie idempotentie, paragraaf 5.4). Het MessageID moet uniek zijn. Met uitzondering van XML berichten voor Allocatie en Reconciliatie Wholesale Gas, wordt daarom de UUID (Universally Unique Identifier) voor de MessageID toegepast. Een UUID wordt weergegeven in 32 hexadecimale cijfers ingedeeld over vijf groepen, in ongelijk aantal en gescheiden door koppeltekens: 8-4-4-4-12. Een UUID is dus opgebouwd uit 36 tekens: 32 hexadecimale cijfers en vier koppeltekens. Bijvoorbeeld: 10f01640-8594-11e2-9e96-0800200c9a66 UUID is gespecificeerd in RFC 4122.

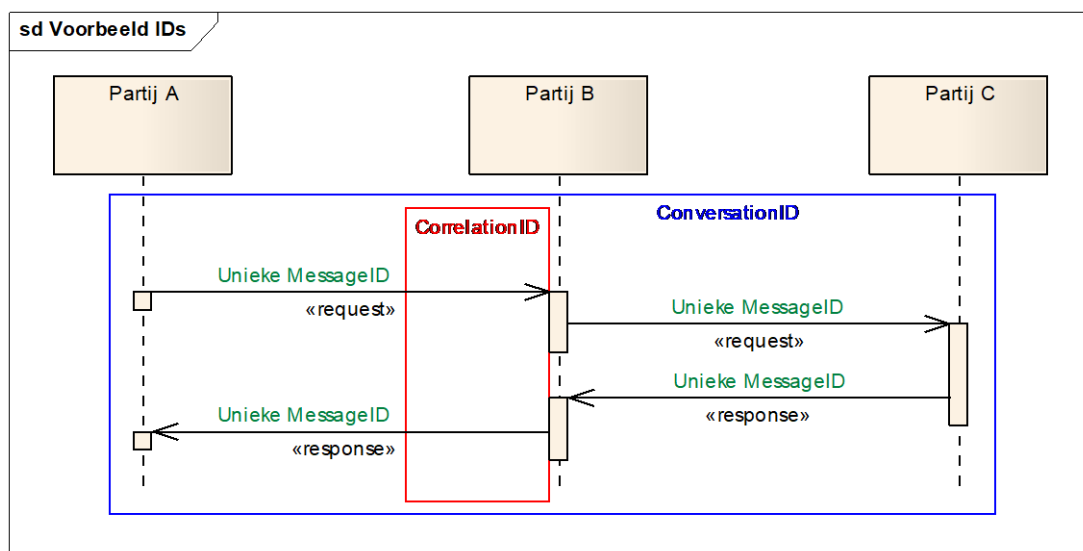
Voor correct gebruik van deze identifiers, zoals in onderstaand voorbeeld is weergegeven, is het noodzakelijk dat tussen alle partijen in de communicatieketen strikte afspraken over het gebruik (en correct teruggeven) van de identifiers zijn gemaakt.

Vooralsnog zijn er strikte afspraken over het toepassen van de verplichte identifier MessageID en de optionele identifier CorrelationID:

- MessageID: technische en unieke identificatie (UUID) van een bericht;
- CorrelationID: identificatie in het bericht door de aanvrager, dit CorrelationID dient de ontvangende partij als CorrelationID te vermelden in de antwoorden naar de aanvrager.

De overige identifiers kunnen toegepast worden indien daar in de communicatieketen duidelijke afspraken over zijn gemaakt.

In Figuur 31 is geïllustreerd hoe de verschillende identifiers gebruikt kunnen worden in de context van een proces.



Figuur 31: Scope van identifiers

In Figuur 31 initieert partij A een proces door middel van een Request aan partij B. Partij A vult de ConversationID en de CorrelationID. Partij B verwerkt de Request en stuurt een Request aan partij C. Deze Request krijgt dezelfde ConversationID zoals toegekend door partij A maar een nieuwe CorrelationID. Zowel de (door partij A toegekende) ConversationID als de (door partij B toegekende) CorrelationID worden meegegeven in de Response van partij C. De daarop volgende Response van partij B aan partij A bevat de oorspronkelijke ConversationID en de oorspronkelijke, door partij A toegekende, CorrelationID.

Inhoud en doel van MessageID, CorrelationID en ConversationID zijn technisch van aard.
inhoud en doel van DocumentID en ProcessTypeID zijn functioneel van aard.

TestRequest

De TestRequest indicator is een optioneel veld om aan te kunnen geven dat de betreffende business document instantie een test document is en door het ontvangende systeem in een voorgedefinieerde test modus afgehandeld moet worden. Echter, het aangeven of een ontvangend systeem een business document instantie in test modus moet afhandelen wordt veelal kenbaar gemaakt door gebruik te maken van separate testcertificaten.

RepeatedRequest

RepeatedRequest is een optioneel veld om aan te kunnen geven dat het een herhaling betreft van een eerder gestuurde business document instantie. Dit is voor de performance van het ontvangende systeem van belang. Dit systeem kan dan de controle of een business document instantie al een keer eerder is gestuurd, achterwege laten. De herhaald verstuurd instantie moet als idempotent met het origineel worden beschouwd.

CreationTimestamp

CreationTimestamp is een verplicht datum/tijd veld dat vastlegt wanneer de business document instantie (het bericht) was gecreëerd. Datum/tijd conform W3C DTF Note ISO 8601.

ExpiresAt

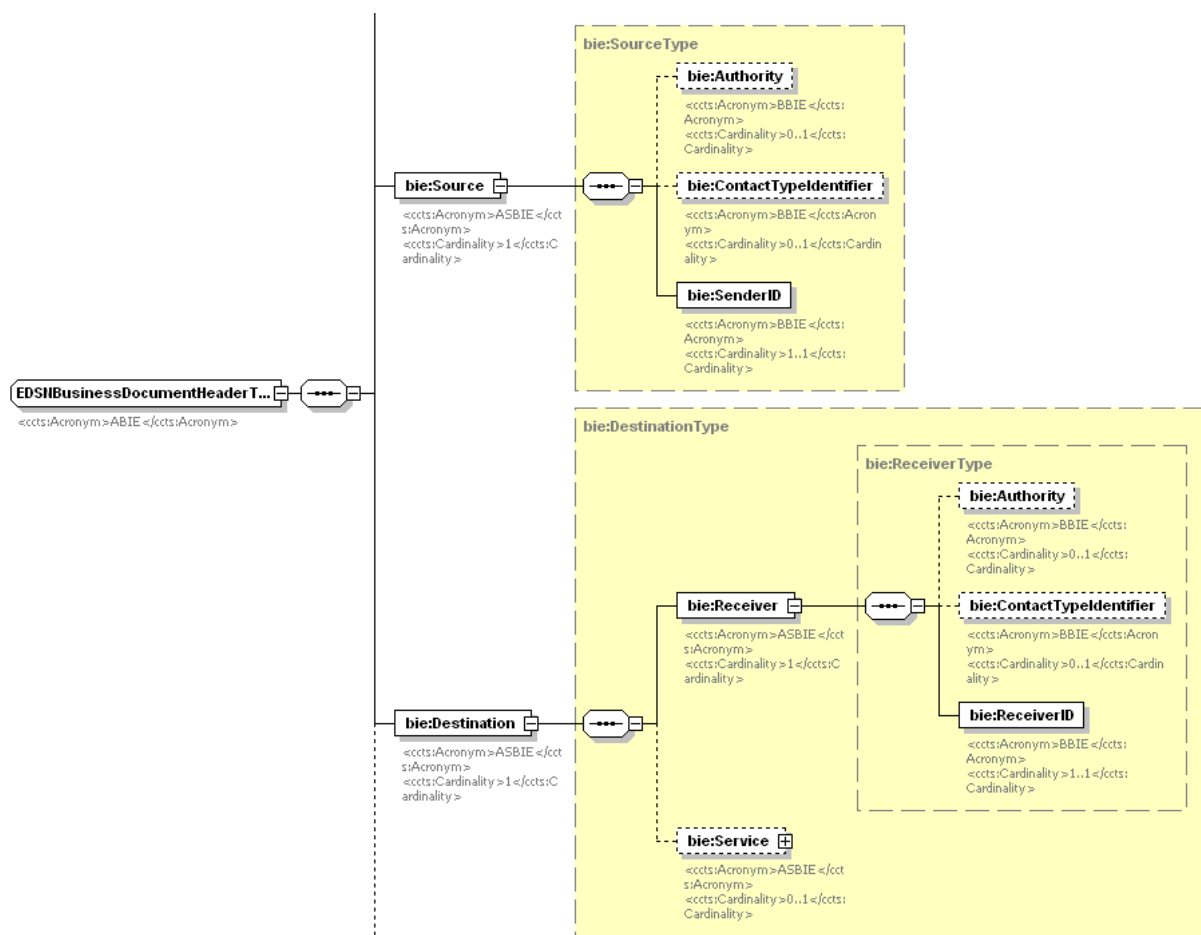
ExpiresAt is een optioneel datum/tijd veld om de uiterste geldigheidsdatum aan te geven. Tot en met deze datum mag de betreffende business document instantie afgehandeld worden, daarna niet meer! Dit mechanisme is of wordt niet in alle systemen geïmplementeerd (de meeste niet). Datum/tijd conform W3C DTF Note ISO 8601.

ContentHash

ContentHash is een optioneel veld waarmee het ontvangende systeem de integriteit van een business document instantie kan controleren. De specifieke toepassing zal vastgelegd worden zodra dit veld gebruikt gaat worden.

Source en Destination

Zender en ontvanger worden in Source en Destination gedefinieerd.



Figuur 32: Source en Destination in de EDSN header

SenderID en ReceiverID zijn verplicht. Overige velden zijn optioneel.

Veld	Omschrijving
Source > SenderID	Identificatie marktpartij (EAN ID). (Datatype: string).
Source > Authority	Omschrijving van de kwalificatie van de marktpartij identifier (SenderID). Kwalificatie kan bv. EAN.UCC of GLN zijn. (Datatype: string).
Source > ContactTypeIdentifier	Rol zendende marktpartij. Rol kan bv. leverancier of netbeheerder zijn. Zie onderstaande paragraaf "Identificatie marktpartij of organisatie in EDSN Business Document Header". In voorkomende gevallen waar de zendende partij een systeem is: de unieke naam van het betreffende systeem. Dit dient separaat en gedetailleerd te worden beschreven in desbetreffende interface specificatie c.q. servicebeschrijving. (Datatype: string).
Destination > Service > ServiceName	Naam van de service. (Datatype: string).
Destination > Service > ServiceMethod	Gewenste functionaliteit van de service. Specificatie als: Requesting Service Transaction of Responding Service Transaction. (Datatype: string).
Destination > Receiver > ReceiverID	Identificatie marktpartij (EAN ID). (Datatype: string).
Destination > Receiver > Authority	Omschrijving van de kwalificatie van de marktpartij identifier (ReceiverID). Kwalificatie kan bv. EAN.UCC of GLN zijn. (Datatype: string).
Destination > Receiver > ContactTypeIdentifier	Rol ontvangende marktpartij. (Datatype: string).

Identificatie marktpartij of organisatie in EDSN Business Document Header

Alle XML Schema's worden door het Berichtencentrum voorzien van de standaard EDSN Business Document Header. De vier verplichte velden in deze header zijn:

- MessageID;
- CreationTimestamp;
- SenderID;
- ReceiverID.

Indien een verzoek naar de centrale systemen van EDSN wordt gestuurd:

SenderID gevuld met de EAN13 code van de partij die het verzoek instuurt.

ReceiverID gevuld met de EAN13 code van EDSN (8712423010208).

In het, door de centrale systemen, geretourneerde bericht wordt:

SenderID gevuld met de EAN13 code van EDSN (8712423010208).

ReceiverID gevuld met de EAN13 code van de partij die het geretourneerde bericht ontvangt.

Specifiek voor het Portaal moet in de EDSN Business Document Header van de XML berichten kenbaar worden gemaakt of een verzoek afkomstig is van een organisatie of van een marktpartij. Een organisatie kan werken met meerdere marktpartijen (lees EANcodes of labels) binnen dezelfde rol. Een organisatie kan op deze wijze de verwerking over meerdere marktpartijen bundelen. Bij organisaties die slechts werken met één marktpartij vallen deze twee niveaus feitelijk samen. Deze identificatie moet als volgt worden opgenomen in het veld ContactTypeIdentifier van de EDSN Business Document Header:

Partijrol	Organisatie	Marktpartij
RNB	DDM_O	DDM_M
LV	DDQ_O	DDQ_M
PV	DDK_O	DDK_M

MV	MDR_O	MDR_M
EDSN	EDSN	

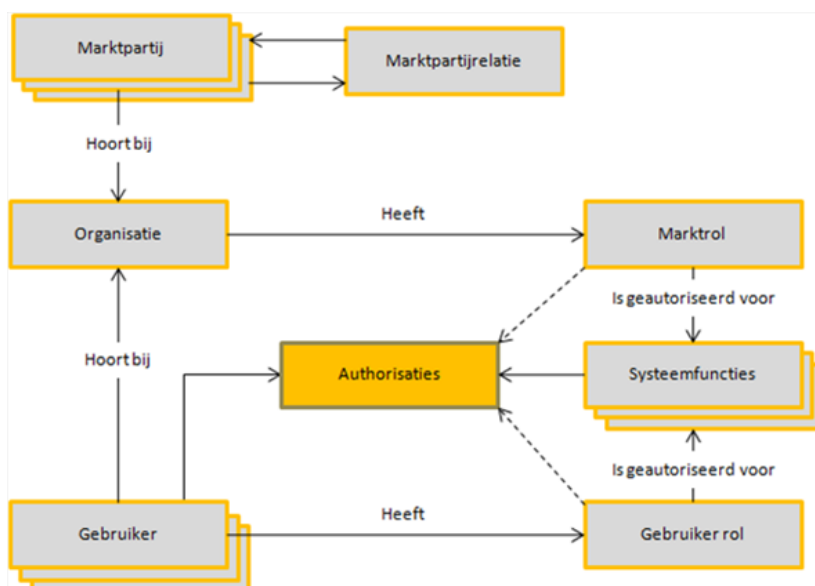
Deze toevoeging geeft aan of de betreffende afzender/ontvanger een organisatie respectievelijk marktpartij is. Hieronder volgt een voorbeeld van fragment van de EDSN Business Document Header van een verzoek naar de centrale systemen van EDSN.

```
<EDSNBusinessDocumentHeader>
.
<Receiver>
  <ContactTypeIdentifier>EDSN</ContactTypeIdentifier>
  <ReceiverID>8712423010208</ReceiverID>
</Receiver>
.
<Source>
  <ContactTypeIdentifier>DDQ_O</ContactTypeIdentifier>
  <SenderID>8714252008029</SenderID>
</Source>
.
</EDSNBusinessDocumentHeader>
```

Figuur 33: ContactTypeIdentifier in EDSN BDH

Het toevoegen van organisatie of marktpartij identificatie in veld ContactTypeIdentifier van de EDSN Business Document Header is vereist voor de structuur van organisatie met daaronder één of meerdere marktpartijen. Het doel van deze structuur is om organisaties de mogelijkheid te bieden om voor meerdere EAN codes binnen dezelfde marktrol vanuit hetzelfde (systeem)account te kunnen werken.

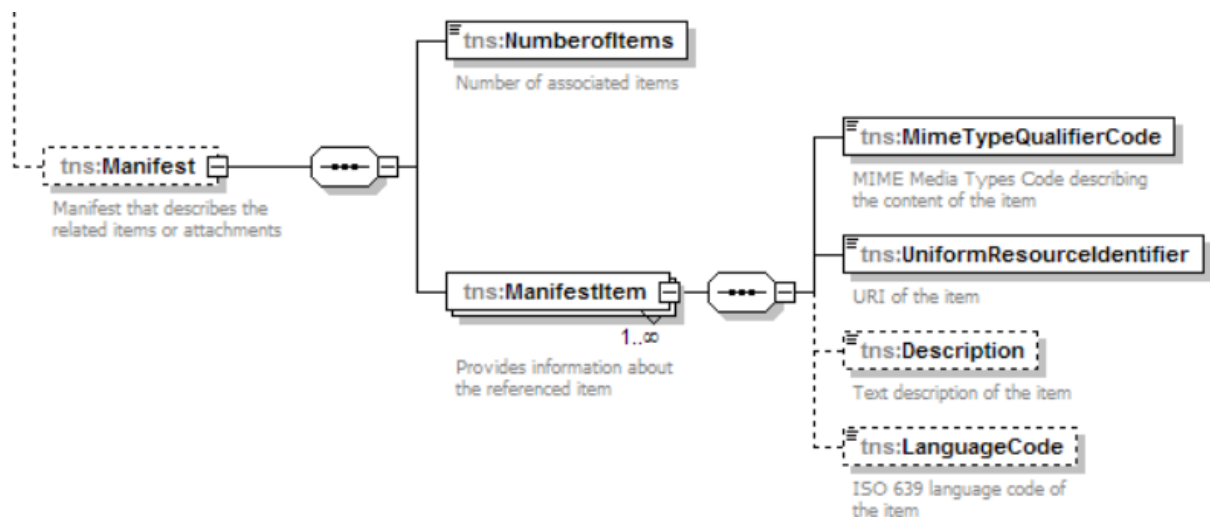
In het datamodel is de autorisatie van gebruikers en rollen gedefinieerd. Iedere gebruiker hoort bij één organisatie en iedere organisatie heeft precies één marktrol. Een organisatie kan één of meerdere marktpartijen onder zich hebben. Gebruikers zijn dus direct gekoppeld aan een organisatie, maar kunnen uit naam van zo'n marktpartij handelen door de EANcode van die marktpartij te gebruiken in de aanvraag



Figuur 34: Datamodel autorisaties van het EDSN Portaal

Definitie Manifest

Het Manifest is een optioneel veld dat de aan een business document instantie gerelateerde attachments beschrijft. Dit mechanisme is niet geïmplementeerd.



Figuur 35: Manifest

Veld	Omschrijving
Manifest > NumberOfItems	Aantal attachments gerelateerd aan de business document instantie.(Datatype: integer).
Manifest > ManifestItem > MimeTypeQualifierCode	MIME Media Type code van de attachment, code conform registratie IANA (www.iana.org). (Datatype: string).
Manifest > ManifestItem > UniformResourceIdentifier	Identificer voor locatie en naam van een attachment. (Datatype: string).
Manifest > ManifestItem > Description	Vrij te definiëren veld met tekstuele informatie behorend bij de attachment. (Datatype: string).
Manifest > ManifestItem > LanguageCode	ISO 639 taal code van attachment. (Datatype: string).

5.6 Autorisatie

Autorisatie is de verantwoordelijkheid van de marktpartij zelf. De naam van de gebruiker wordt meegegeven in de EDSN Business Document Header en op dat niveau afgehandeld. Indien gewenst kunnen systemen als gebruiker worden bestempeld.

Organisaties zijn zelf verantwoordelijk voor de interne authenticatie van hun gebruikers en onderdelen. Onder de vleugels van een concern (organisatie) kan bijvoorbeeld een batch gevuld worden met berichten die afkomstig zijn uit meerdere van zijn bedrijfsonderdelen (marktpartij met een specifieke marktrol). Binnen de berichten worden de bedrijfsonderdelen en rollen onderscheiden en geïdentificeerd met de EAN GLN codes, die de organisatie aan die onderdelen heeft toegelend.

5.7 Beveiliging

Voor het beveiligen van de berichtenuitwisseling in de sector wordt gebruik gemaakt van beveiligingscertificaten. Dit betreft “private root” PKI-overheid certificaten welke worden uitgegeven door een Certified Authority (CA)⁵. De volgende beveiligingscertificaten zijn van toepassing:

1. een servercertificaten voor beveiliging op transportniveau (datacommunicatie) en authenticatie (signing);
 2. een vertrouwelijkheidcertificaten voor versleuteling van de uitgewisselde informatie.
1. Op transportniveau wordt voor beveiliging gebruik gemaakt van servercertificaten in de Transport Layer Security (TLS)⁶ op de HTTPS verbinding. De authenticiteit van de server wordt, middels een servercertificaat eenzijdig gecontroleerd, terwijl de client onbekend blijft. Voor tweezijdige authenticatie wordt een clientcertificaat (“private root” PKI-overheid) gebruikt welke weer door de server wordt gecontroleerd. Voor een servercertificaat dient de uitgever van dit certificaat en de naam van de server gecontroleerd te worden, voor het gebruik van PKI-overheid dienen de public keys verspreid/geladen worden om de communicatie op transportniveau te kunnen vertrouwen. Er wordt op deze manier een beveiligde verbinding opgezet. In het algemeen wordt eenzijdige authenticatie van de server gebruikt, zeker in B2B scenario's. Door EDSN samen met de CA is er een procedure opgesteld die wordt gebruikt om de cliënt certificaten veilig uit te wisselen. Het servercertificaat kan ook worden gebruikt voor authenticatie (signing) van de content.

Beveiliging op transportniveau (datacommunicatie) en authenticatie (signing) is geïmplementeerd voor: Portaal, PIR, TMR, PP4, CSS, AS2 Teststraat, MCV.

2. De integriteit van berichten zelf wordt geregeld door gebruik te maken van versleuteling (payload encryption) zoals vastgelegd in de WS-Security standaard. Hiervoor worden vertrouwelijkheidcertificaten gebruikt, deze PKI certificaten worden eveneens door een Certified Authority (CA) uitgegeven en zijn een ander soort certificaat dan bij 1. benoemde servercertificaten. Voor versleuteling dient het public key uit het vertrouwelijkheidscertificaat van de ontvanger gebruikt te worden, de ontvanger is de enige die het kan decoderen met het private key uit het vertrouwelijkheidscertificaat.

Versleuteling van de uitgewisselde informatie is niet geïmplementeerd in de centrale systemen. Was voorheen alleen geïmplementeerd voor TMR. Als gevolg van issue TC036 is de versleuteling van TMR informatie uitgefaseerd.

De EAN controle wordt op serviceniveau gedaan door de ‘SenderId’ uit de inhoud van bericht te vergelijken met het Subject.serialNumber veld uit het beveiligingscertificaat van hetzelfde bericht (of het beveiligingscertificaat van de verbinding). Het inrichten van de autorisatie-controle (op functie en op data) is aan de ontvanger. Een bedrijfsonderdeel kan op deze wijze (in een uitwijkscenario) eventueel gebruik maken van de verbinding van een ander bedrijfsonderdeel; tenslotte is de verbinding vertrouwd en de berichten worden nog steeds namens de juiste marktpartij getekend.

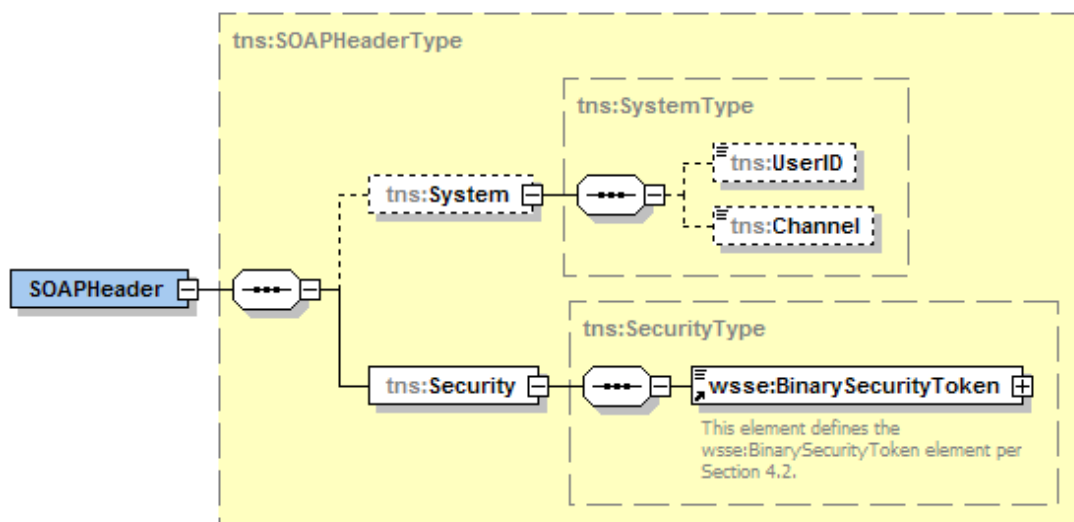
Voor beveiliging op transportniveau (datacommunicatie), authenticatie en versleuteling beveiliging van de berichten wordt gebruik gemaakt van WS-Security⁷. Hiertoe wordt het beveiligingscertificaat, of een verwijzing naar een elders opgeslagen beveiligingscertificaat, opgenomen in de SOAP header. Indien naar een beveiligingscertificaat wordt verwezen wordt een <wsse:SecurityTokenReference> gebruikt. Het certificaat zelf wordt geplaatst in een <wsse:BinarySecurityToken>. Autorisatie informatie, zoals een UserID, wordt eveneens in de SOAP header geplaatst, zie Figuur 36. Voor een goede werking dient men eerst, indien van toepassing, compressie toe te passen en daarna signing en mogelijk encryptie. Overigens wordt compressie om deze redenen niet toegepast op voorschrift van de TC NEDU.

De procedure voor het verwerven van een beveiligingscertificaat is te vinden op *mijnEDSN*.

⁵ CA is KPN. Deze CA heeft a.g.v. invoering van TC027 (per 15 juni 2016) de vereiste “private root” PKI-overheid servercertificaten opgeleverd aan de partijen in de sector.

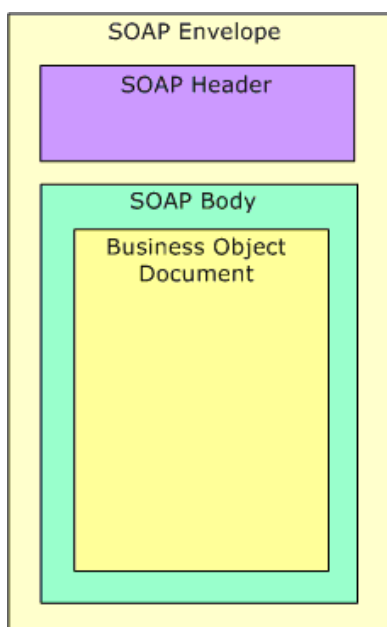
⁶ In lijn met TC024: op transportniveau TLS 1.2 waarbij terugval op TLS 1.0 of lager per 22 juni 2016 niet meer wordt ondersteund.

⁷ WS-Security zoals hier beschreven is nog niet voor alle centrale systemen al geïmplementeerd.



Figuur 36: SOAP Header

Figuur 37 toont de positie van de SOAP Header in de SOAP Envelope. Het Business Object Document (met XML payload) zit in de SOAP Body.



Figuur 37: Positie SOAP Header in de SOAP Envelope

Beveiliging van berichtenverkeer is vastgelegd in TC001, TC018, TC024 en TC027.

5.8 CRL check

Een Certificate Revocation List (CRL) is een lijst met certificaatsnummers welke door de betreffende Certificate Authority (CA) herroepen zijn, niet meer geldig zijn en niet meer te vertrouwen zijn voor gebruikers. Deze paragraaf is gericht op het gebruik van de CRL check bij de afhandeling van het berichtenverkeer.

Een CRL wordt periodiek aangemaakt en bijgewerkt. De CRL wordt altijd uitgegeven (vaak elke 24 uur) door de Certificate Authority (CA) die zich alleen toespitst op de door hem uitgegeven certificaten. Deze CRL's kunnen worden geraadpleegd door applicaties met PKI functionaliteit.

We kennen grofweg twee soorten gebruik van de CRL:

1. een gebruiker die op de GUI van een website gebruik maakt van een certificaat, waarbij de geldigheid wordt geverifieerd bij het aanmelden;
2. systemen die onderling gebruik maken van certificaten in gegevensuitwisseling binnen geautomatiseerde processen (B2B = Business to Business).

Het gebruik benoemd bij 1. is geen onderdeel van dit document. Voor 2. is onderstaande toelichting van toepassing.

Als tijdens een interactie tussen systemen (B2B) de benodigde CRL niet beschikbaar is, kunnen de systemen de geldigheid van de gebruikte certificaten niet real-time testen. Het gevolg is dan dat de software het certificaat niet vertrouwd en dus de betreffende berichten niet worden geaccepteerd en verwerkt. Dit is voor geautomatiseerde verbindingen en voor tijdkritische processen in het bijzonder niet wenselijk. Er zijn mogelijkheden om de online beschikbaarheid van de CRL's te omzeilen door een lokale kopie op te slaan, hier kleven echter ook weer nadelen aan.

Aan de B2B communicatie liggen meestal (contractuele) verplichtingen voor de betrokken partijen aan ten grondslag. Indien een digitaal certificaat ingetrokken dient te worden kan dat twee oorzaken hebben: het contract is beëindigd of het certificaat is gecompromitteerd. In het laatste geval moet er overleg plaatsvinden hoe en wanneer het certificaat wordt vervangen en/of dat de communicatie tijdelijk wordt gestopt of op een andere wijze plaatsvindt; dus niet alleen vertrouwen op een technische CRL check. Als het contract of de verplichting eindigt worden de rechten en procesverloop in de back-end systemen van de betrokken partijen aangepast: de terugtrekkende partij mag bijvoorbeeld alleen nog historische gegevens opvragen. Indien een partij geen toegang meer mag hebben tot het proces en/of de gegevens worden eerst de rechten op de back-end systemen verwijderd, daarna het certificaat voor de berichtencommunicatie en als laatste wordt –indien van toepassing- gemeld bij de CA dat het certificaat ingetrokken moet worden.

Het raadplegen van de CRL zou de continuïteit van het berichtenverkeer niet moeten beïnvloeden, hoe partijen dit technisch implementeren is aan deze partijen zelf (CRL check uit, melding van revocation aan de systeembeheerder, etc.).

Aanbeveling: Partijen die B2B communicatie in geautomatiseerde processen onderhouden, moeten een procedure hebben waarin naast de bewaking van contractuele aspecten ook het intrekken van digitale certificaten een onderdeel is.

5.9 Standaard Request, Response, Afwijzing en Foutsituaties

De standaard request en response zal altijd de EDSN Business Document Header bevatten. Deze header is ook de basis voor een functionele bevestiging als antwoord op batchaanvragen, alsmede als verzoek voor het opvragen van batchresultaten. Zo'n functionele bevestiging fungeert als generieke response.

Request

Voorbeeld van een standaard batch request structuur (vraag alle (gereedstaande) Loss berichten op):

```
<LossResultRequest>
  <EDSNBusinessDocumentHeader>
    <CreationTimestamp>2009-12-17T09:30:47Z</CreationTimestamp>
    <Destination>
      <Receiver>
        <ReceiverID>8712423010208</ReceiverID>
      </Receiver>
    </Destination>
    <MessageID>55264bc0-309f-11de-9acb-0002a5d5c51b</MessageID>
    <Source>
      <SenderID>000000000000</SenderID>
    </Source>
  </EDSNBusinessDocumentHeader>
</LossResultRequest>
```

Response

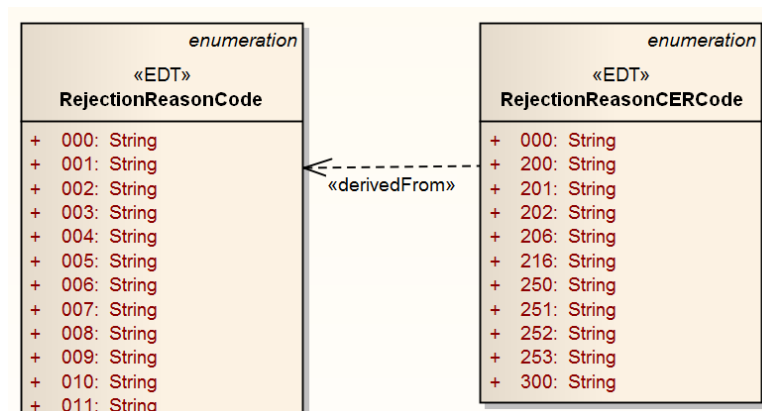
Voorbeeld van een standaard response structuur (functioneel 'OK' bericht (ontvangstbevestiging) op een batch verzoek):

```
<EndOfMeteringResponsibleBatchResponse>
  <EDSNBusinessDocumentHeader>
    <CreationTimestamp>2009-11-10T11:38:34Z</CreationTimestamp>
    <Destination>
      <Receiver>
        <ReceiverID>000000000000</ReceiverID>
      </Receiver>
    </Destination>
    <MessageID>be557b20-309f-11de-90d7-0002a5d5c51b</MessageID>
    <Source>
      <SenderID>8712423010208</SenderID>
    </Source>
  </EDSNBusinessDocumentHeader>
</EndOfMeteringResponsibleBatchResponse>
```

Functionele afwijzing

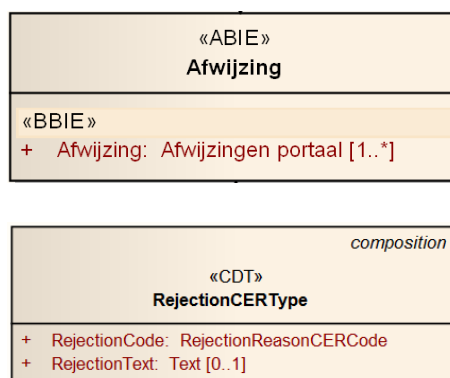
Er wordt onderscheid gemaakt tussen functionele fouten in de vorm van afwijzingen (Rejection) en technische fouten. Het gebruik van dit onderscheid is vastgelegd in TC003.

Functionele fouten zijn semantische fouten, zoals incorrecte combinaties van domeinwaarden of andere validatie en/of data autorisatie fouten. Deze fouten worden teruggekoppeld als een Rejection bericht, waarin een afwijsreden (RejectionReasonCode) en een optionele toelichting is opgenomen.



Figuur 38: EDT RejectionReasonCERCode als afleiding uit de basislijst RejectionReasonCode

Afwijzing (Rejection) structuur (1..*) voor functionele fouten:



Figuur 39: Afwijsreden RejectionReasonCERCode en optionele toelichting

Technische fouten

Technische fouten ontstaan als een aanvraag/melding niet verwerkt kan worden als gevolg van technische redenen, zoals het niet voldoen van het aangeboden XML bericht aan het XML Schema van de interface, of andere (onverwachte) systeem technische problemen

Indien zich een technische fout voordoet, zal een SOAP Fault teruggegeven worden. Een SOAP Fault is één van de drie berichtdefinities waarmee een request/reply operatie in een WSDL is beschreven (naast input en output ook een fault). De berichtdefinitie van deze algemene SOAP Fault zal een ErrorCode, een ErrorText en ErrorDetails bevatten. In de SOAP Fault komen deze drie velden slechts éénmaal voor.

De volgende technische fouten worden gebruikt:

Error-Code	ErrorText	ErrorDetails	Opmerking	Advies
1	Database fout	<geen>	Een transactie kan niet worden doorgevoerd vanwege het feit dat zich een database probleem voordoet.	Service-request (d.w.z. hetzelfde verzoek) opnieuw aanbieden met een <u>nieuw</u> UUID. Indien deze “retry” faalt, dan escaleren ⁸ .
2	XML Schema validatie fout	<details van XML Schema validatie fout>	Het aangeboden XML bericht valideert niet tegen het bijbehorende XML Schema.	Het aangeboden XML bericht is op syntax afgewezen en dient door de aanvrager gecontroleerd, aangepast en gevalideerd te worden tegen het corresponderende XML Schema. Na aanpassing de service-request opnieuw aanbieden met een <u>nieuw</u> UUID.
3	Systeem fout	<geen>	Het systeem is niet in staat om het bericht te verwerken.	Service-request (d.w.z. hetzelfde verzoek) opnieuw aanbieden met <u>nieuw</u> UUID. Indien deze “retry” faalt, dan escaleren ¹⁰ .
4	Security fout	<geen>	Een aangeboden transactie wordt om security redenen niet verwerkt.	Controle security door aanvrager. Na controle en opheffen van security tekortkoming de service-request opnieuw aanbieden met een <u>nieuw</u> UUID.
5 ⁹	Time-out	<geen>	Een aangeboden transactie wordt vanwege een time-out niet verwerkt.	De service-request opnieuw aanbieden met een <u>nieuw</u> UUID. Het feit dat de eerste aanvraag resulteerde in een time-out kan wijzen op een verminderde performance van de back-end. Het is een “best practice” om in het geval van een time-out melding, het bericht nog één of twee keer opnieuw aan te bieden (met nieuw UUID). Indien de aanvraag alsnog niet succesvol kan worden afgehandeld, is de aanbeveling om een bepaalde periode te wachten alvorens het opnieuw aan te bieden (met nieuw UUID) ¹⁰ .

⁸ Geadviseerd wordt om in dat geval contact op te nemen met EDSN Beheer.

⁹ SOAP Fault 5 “Time-out” is niet voor alle systemen al geïmplementeerd.

¹⁰ Deze periode zal in de “Quality of Service” specificatie van de betreffende service beschreven worden.

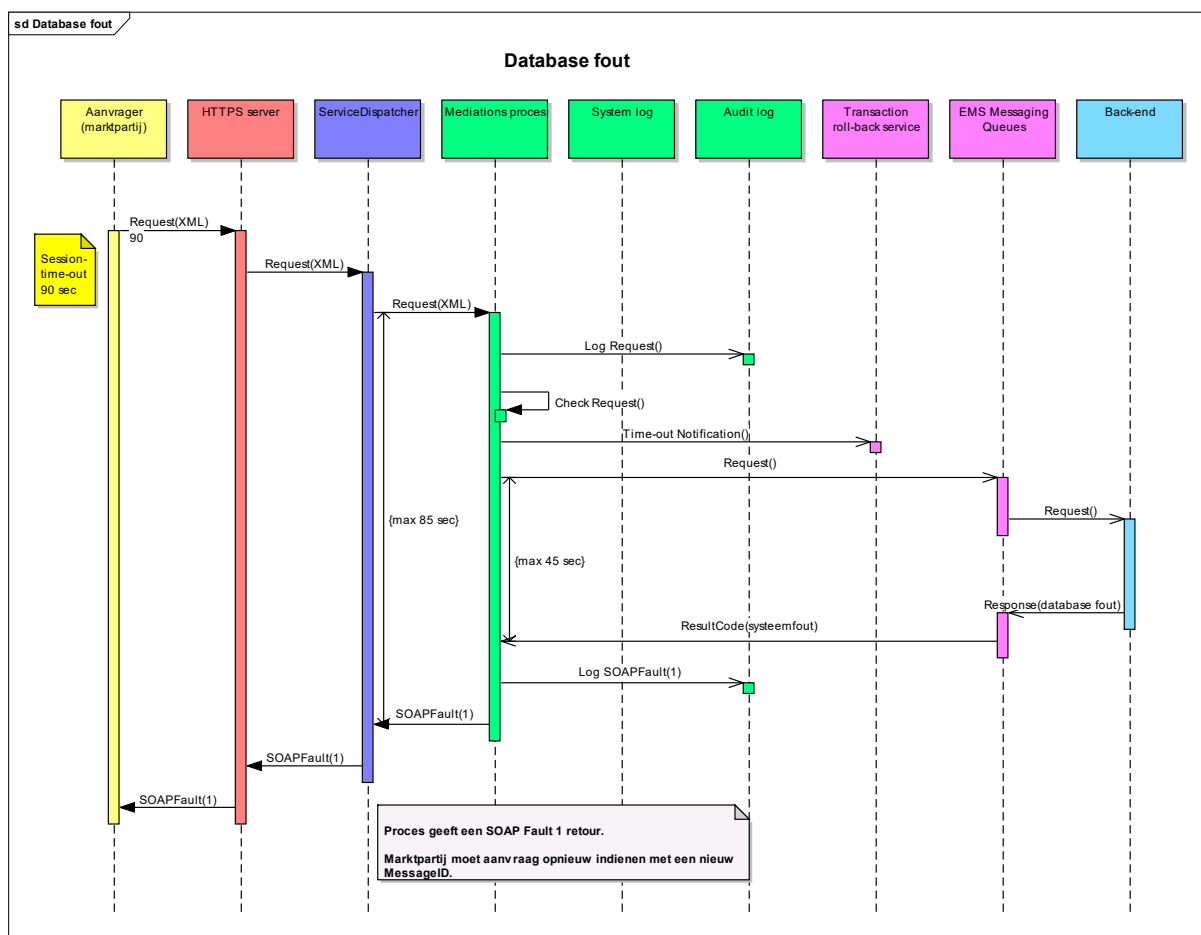
Error-Code	ErrorText	ErrorDetails	Opmerking	Advies
				Indien deze “retry” faalt, dan escaleren ¹¹ .
6	Einde sessie	<geen>	Het antwoord op de aangeboden transactie kan niet worden geretourneerd vanwege einde sessie service dispatcher.	Service-request opnieuw aanbieden met het <u>originele</u> UUID. Aanvrager ontvangt de eventueel gelogde melding uit de audit log (door de ingestelde idempotentie). Indien deze “retry” faalt en opnieuw resulteert in een SOAP Fault 6, dan escaleren ¹³ .

¹¹ Geadviseerd wordt om in dat geval contact op te nemen met EDSN Beheer.

De technische fouten worden conform onderstaande scenario's afgehandeld:

SOAP Fault 1 – Database fout

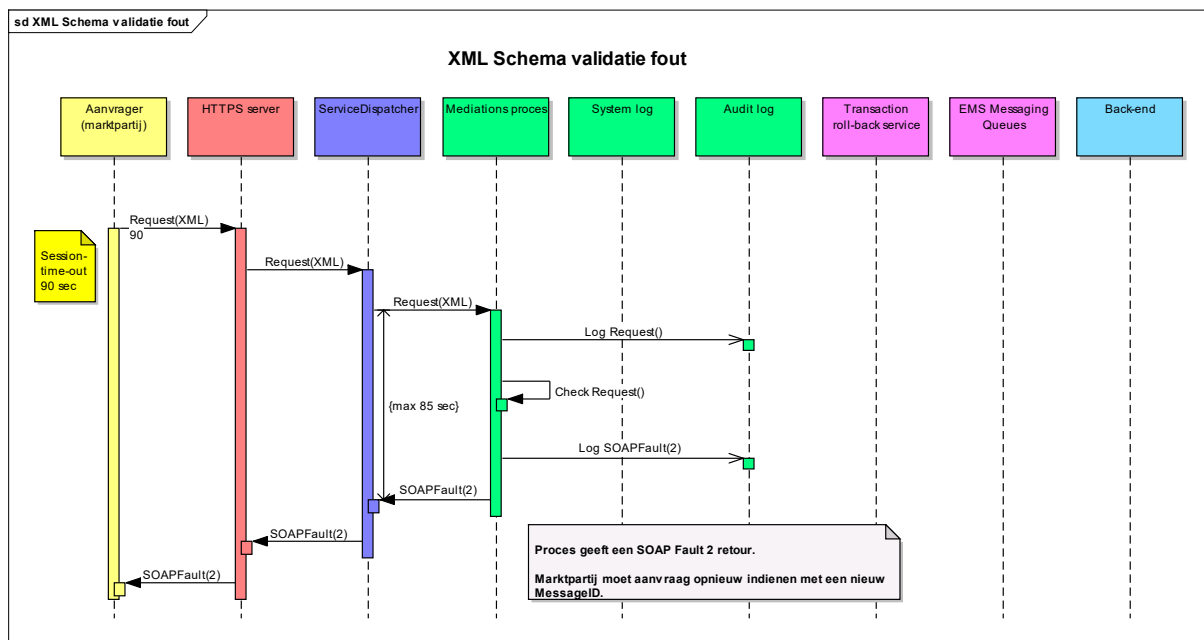
Database fout	
Omschrijving	Een transactie kan niet worden doorgevoerd vanwege het feit dat zich een database probleem voordoet.
Preconditie(s)	1. De service-request wordt ingediend door een aanvrager die hiervoor is geautoriseerd en geauthentiseerd; 2. De service-request is valide.
Resultaat	Aanvrager ontvangt een SOAP Fault 1 retour. Service-request is niet geëffectueerd in back-end.
Verloop	Aanvrager moet service-request (d.w.z. hetzelfde verzoek) opnieuw indienen met een <u>nieuw</u> MessageID (UUID). Indien deze “retry” faalt, dan escaleren. Geadviseerd wordt om in dat geval contact op te nemen met EDSN Beheer.



Figuur 40: Database fout – SOAP Fault 1

SOAP Fault 2 - XML Schema validatie fout

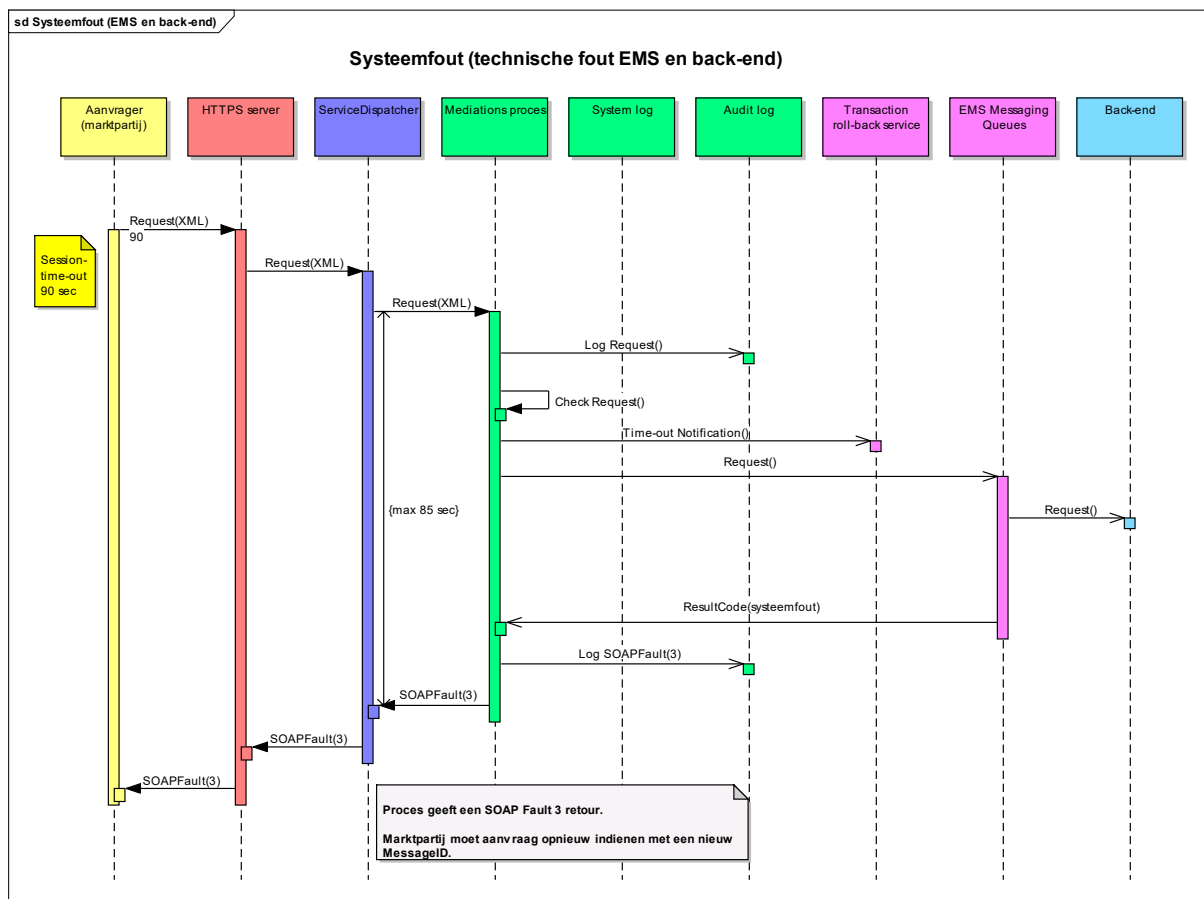
XML Schema validatie fout	
Omschrijving	Aangeboden XML bericht valideert niet tegen het bijbehorende XML Schema.
Preconditie(s)	1. De service-request wordt ingediend door een aanvrager die hiervoor is geautoriseerd en geauthentiseerd.
Resultaat	Aanvrager ontvangt een SOAP Fault 2 retour. Het aangeboden XML bericht is op syntax afgewezen en dient door de aanvrager gecontroleerd, aangepast en gevalideerd te worden tegen het corresponderende XML Schema. Service-request is niet geeffectueerd in back-end.
Verloop	Aanvrager moet de (aangepaste) service-request opnieuw indienen met een <u>nieuw</u> MessageID (UUID).



Figuur 41: XML Schema validatie fout – SOAP Fault 2

SOAP Fault 3 - Systeemfout

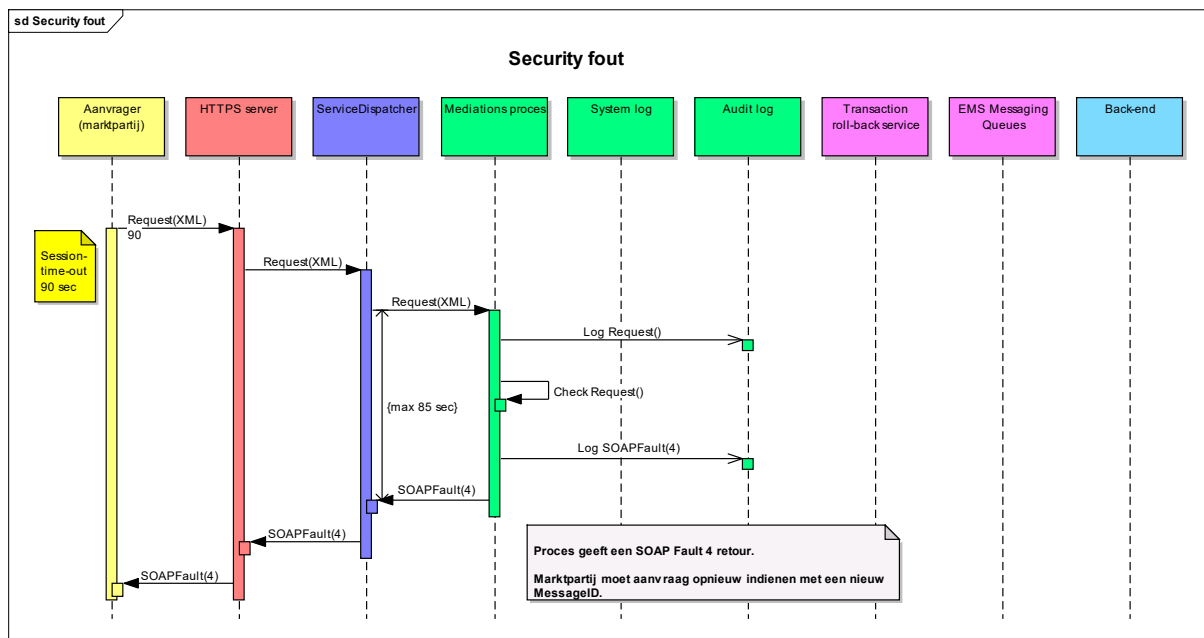
Systeemfout	
<u>Omschrijving</u>	Service-request wordt door back-end niet correct afgehandeld.
<u>Preconditie(s)</u>	1. De service-request wordt ingediend door een aanvrager die hiervoor is geautoriseerd en geauthentiseerd; 2. De service-request is valide.
<u>Resultaat</u>	Aanvrager ontvangt een SOAP Fault 3 retour. Service-request is niet geëffectueerd in back-end.
<u>Verloop</u>	Aanvrager moet service-request (d.w.z. hetzelfde verzoek) opnieuw indienen met een nieuw MessageID (UUID). Indien deze "retry" faalt, dan escaleren. Geadviseerd wordt om in dat geval contact op te nemen met EDSN Beheer.



Figuur 42: Systeemfout – SOAP Fault 3

SOAP Fault 4 - Security fout

Security fout	
Omschrijving	Een aangeboden service-request wordt om security redenen niet verwerkt.
Preconditie(s)	1. De service-request is valide.
Resultaat	Aanvrager ontvangt een SOAP Fault 4 retour. Controle security door aanvrager. Service-request is niet geeffectueerd in back-end.
Verloop	Aanvrager moet na opheffen van de security tekortkoming het service-request opnieuw indienen met een <u>nieuw</u> MessageID (UUID).

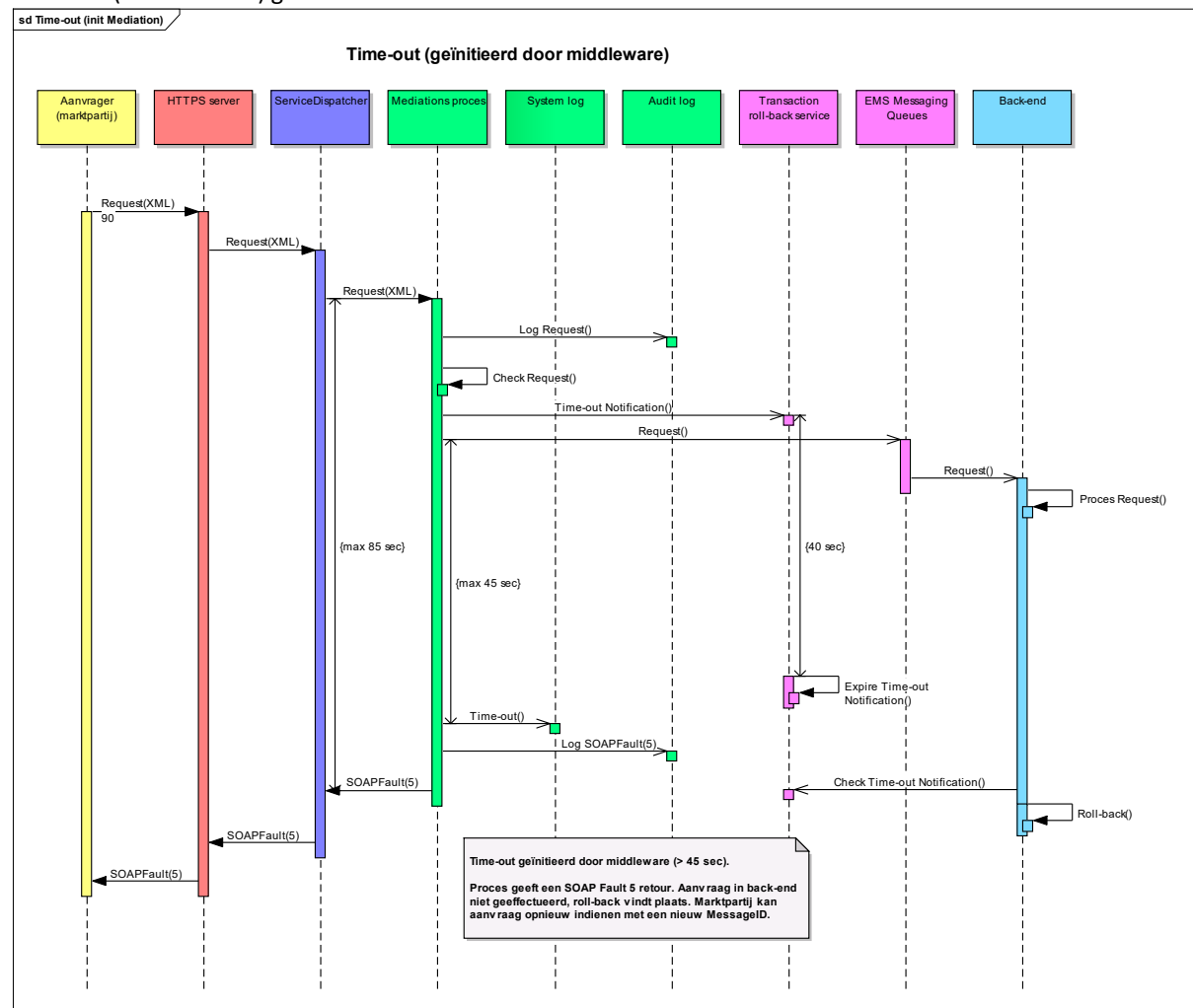


Figuur 43: Security fout – SOAP Fault 4

SOAP Fault 5 - Time-out back-end (middleware/applicatie)

Time-out back-end (middleware/applicatie)	
Omschrijving	Service-request wordt door back-end niet binnen de ingestelde time-out afgehandeld.
Preconditie(s)	1. De service-request wordt ingediend door een aanvrager die hiervoor is geautoriseerd en geauthentiseerd; 2. De service-request is valide.
Resultaat	Aanvrager ontvangt een SOAP Fault 5 retour. Service-request is niet geëffectueerd in back-end (roll-back mechanisme).
Verloop	Aanvrager moet service-request opnieuw indienen met een <u>nieuw</u> MessageID. Het is een "best practice" om in het geval van een time-out melding, het bericht nog één of twee keer opnieuw aan te bieden (met <u>nieuw</u> UUID). Indien de aanvraag alsnog niet succesvol kan worden afgehandeld, is de aanbeveling om een bepaalde periode te wachten alvorens het opnieuw aan te bieden (met <u>nieuw</u> UUID)¹². Indien deze "retry" faalt, dan escaleren. Geadviseerd wordt om in dat geval contact op te nemen met EDSN Beheer.

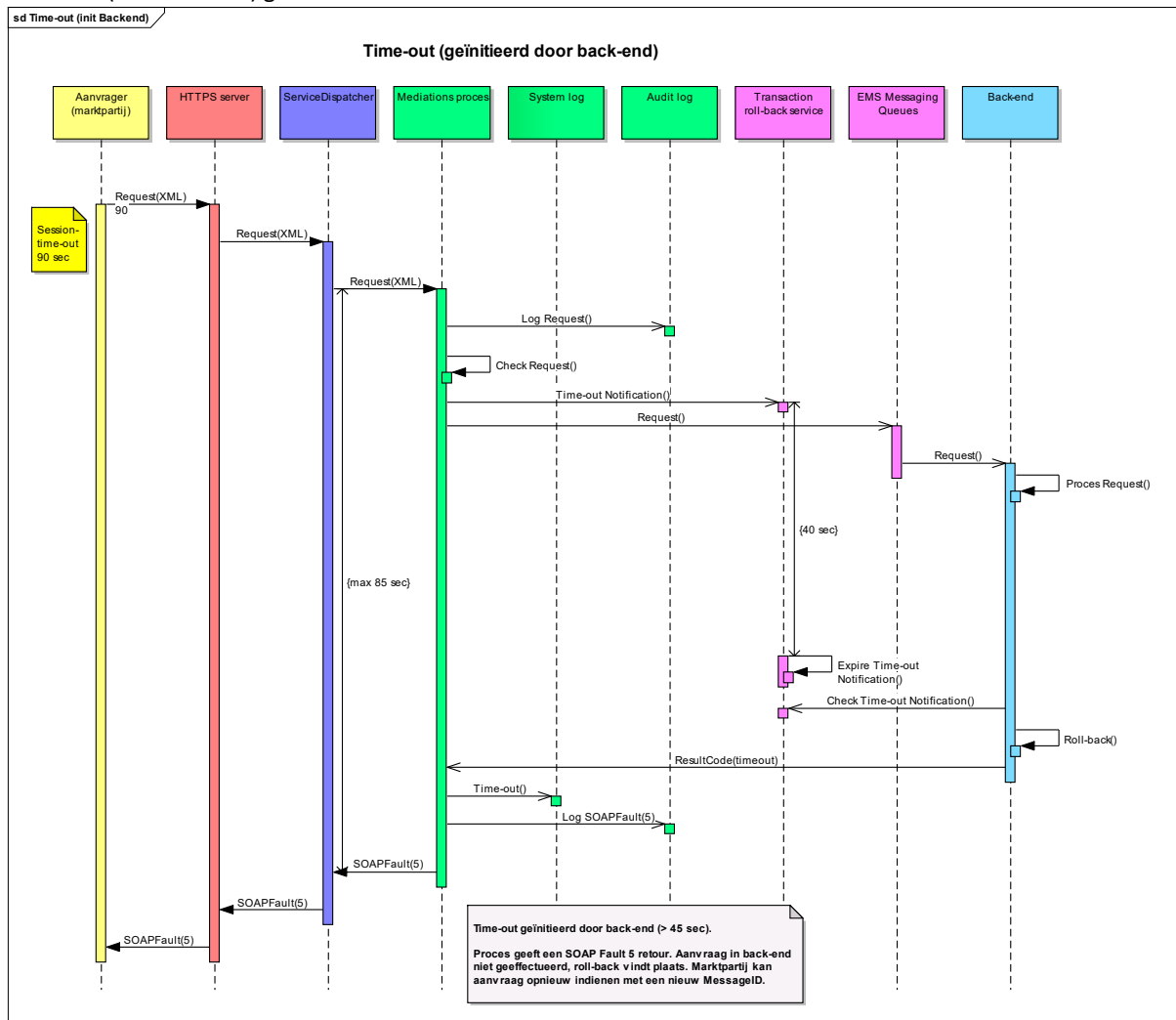
Time-out (SOAP Fault 5) geïnitieerd door middleware



Figuur 44: Time-out (middleware) – SOAP Fault 5

¹² Deze periode zal in de "Quality of Service" specificatie van de betreffende service beschreven worden.

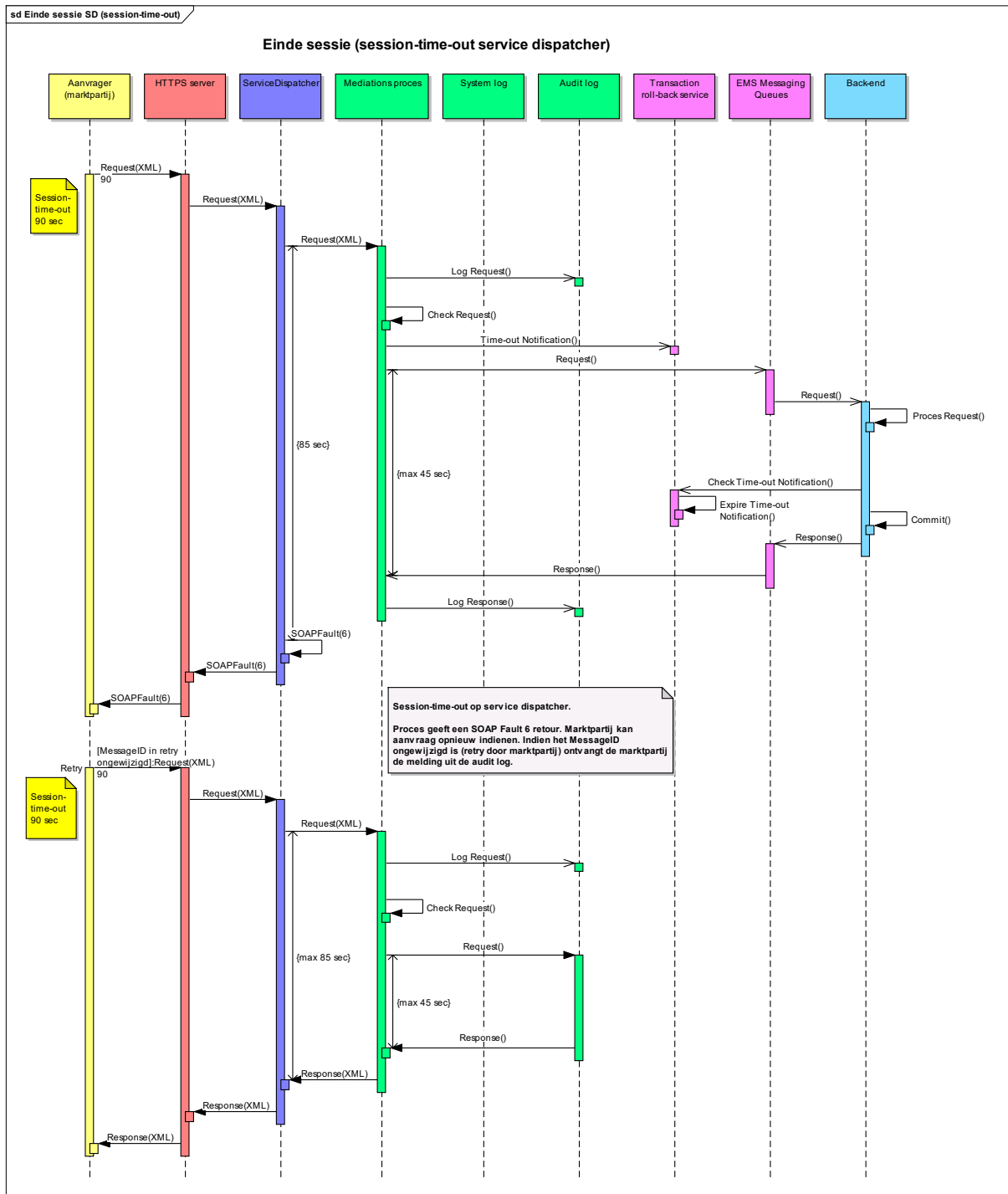
Time-out (SOAP Fault 5) geïnitieerd door back-end



Figuur 45: Time-out (backend) – SOAP Fault 5

SOAP Fault 6 - Time-out front-end (service dispatcher)

Time-out front-end (service dispatcher)	
Omschrijving	Front-end ontvangt binnen de ingestelde time-out geen service-response van het back-end.
Preconditie(s)	1. De service-request wordt ingediend door een aanvrager die hiervoor is geautoriseerd en geauthentiseerd; 2. De service-request is valide.
Resultaat	Aanvrager ontvangt een systeemfout (SOAP Fault 6) retour. Service-request in back-end kan nog operationeel zijn en de binnen de time-out van het back-end afgehandeld worden. De service-response wordt in de audit log geplaatst.
Verloop	Aanvrager moet de service-request opnieuw indienen met <u>ongewijzigd</u> MessageID (UUID). De aanvrager ontvangt de eventueel gelogde service-response uit de audit log (door de ingestelde idempotentie). Indien deze “retry” faalt en opnieuw resulteert in een SOAP Fault 6, dan escaleren. Geadviseerd wordt om in dat geval contact op te nemen met EDSN Beheer.



Figuur 46: Session-time-out – SOAP Fault 6

Referentie

Marktpartijen kunnen zelf een referentie in de berichten meegeven, die teruggegeven wordt in antwoordberichten. Deze referentie kan worden gebruikt om in de eigen administratie berichten die bij één procesinstantie horen bij elkaar te houden of te herleiden naar een brongegeven. Deze referentie is als attribuut Referentie (ExternalReference) in class Mutatie opgenomen.

Bij berichten met een centrale verwerking wordt de referentie alleen gecommuniceerd met de indiener van de referentie. De toepassing van referentie is voor de systemen MDD (MeetData Distributie) en PUD (Portaal UitwisselDienst) anders van opzet. Deze systemen zijn een doorgeefluik en sturen de informatie door inclusief de referentie. De ontvangende partij zal bij MDD en PUD dus de External Reference ontvangen van de sturende partij.

Validatie

De ontvangende laag (zoals in veel gevallen het EDSN Portaal) controleert alleen op consistentie van de services en berichten met de WSDL's en de XSD's. Bovendien wordt gecheckt op geldige (XML-) syntax, autorisatie en authenticatie. Domeinwaarden (bijv. in berichten gebruikte codes) worden alleen door de portal gecontroleerd als deze domeinwaarden expliciet in de syntaxdefinitie van de berichten zijn opgenomen. Is dat niet het geval dan vindt de domeinwaarden controle plaats in de achterliggende applicatie.

Uitgangspunt is om in alle XML Schema's de domeinwaardenspecificaties op te nemen voor die attributen die niet ISO-gespecificeerd zijn. Bijvoorbeeld attributen met een beperkt aantal toegestane, stabiel gedefinieerde domeinwaarden (soorten, statussen, eenheden, aanduidingen, categorieën, etc.)

Door opname van domeinwaarden voor dit soort attributen kunnen en dienen partijen voorafgaand aan de uiteindelijke verzending van berichten een domeinwaardvalidatie uitvoeren. Met als doel de finale controle op dekkend zijn van vertalingen van lokaal (-IT domein per marktpartij) naar gemeenschappelijk (EDSN) formaat. Deze validatie beperkt de uitval van berichten in de verdere schakels in het proces.

Niet zinvol gebruik van domeinwaarden dient te worden vermeden. Bijvoorbeeld attributen met een omvangrijke en/of sterk aan wijzigingen onderhevige domeinwaardenlijst, zoals (EAN code van) aansluitingen en marktpartijen.

Het opnemen van domeinwaarden door het EDSN Berichtencentrum in de XML Schema's, conform bovenstaand uitgangspunt, is vastgelegd in TC006.

Identificatie van de gebruiker en (marktpartij binnen) de organisatie

De identiteit van de gebruiker en marktpartij die de webservice operatie aanroept wordt in de Portaal logging vastgelegd. Die identiteit van de marktpartij is niet vastgelegd in de SOAP Body, maar wordt bepaald uit de context van het bericht.

De marktpartij binnen de organisatie voor wie de gebruiker een webservice operatie aanroept zal veelal in de bericht inhoud als EAN 13 code zijn opgenomen. Voor de netbeheerder interface is dit eveneens de EAN 13 code van de netbeheerder (GridOperator). De bijbehorende organisatie wordt aan een enkelvoudige transactie binnen een bericht gekoppeld.

5.10 Uitwisselen van datasets

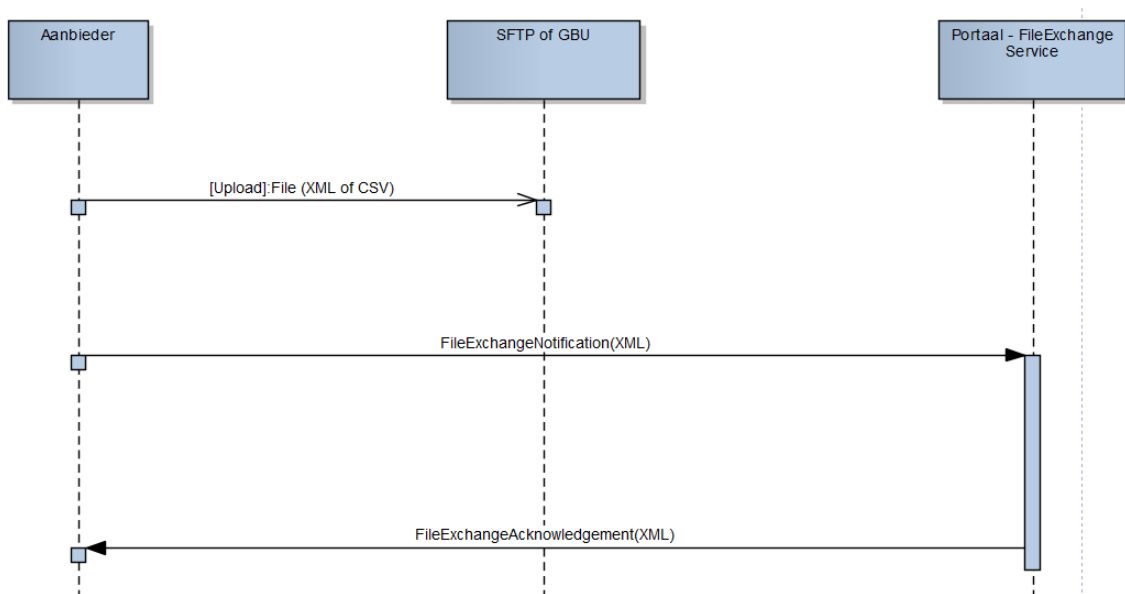
In sommige gevallen moeten grote hoeveelheden data worden uitgewisseld tussen partijen. Voor het uitwisselen van deze datasets is de service "Uitwisseling datasets" (FileExchange) ontwikkeld. Deze service biedt partijen de mogelijkheid om via verschillende kanalen datasets aan te bieden en op te halen.

Zodra een dataset beschikbaar wordt gesteld, dient de aanbieder een notificatie met meta-informatie over de aangeboden dataset voor de ontvanger aan te maken en te zenden. In deze notificatie is onder andere informatie opgenomen over het gebruikte uitwisselkanaal. Deze aanbieder kan een partij zijn, die een dataset voor een centrale applicatie/verwerking aanbiedt, of omgekeerd kan de aanbieder een, onder het Portaal liggende, centrale applicatie zijn die een dataset aanbiedt aan een partij.

Voor de uitwisseling biedt service “Uitwisseling datasets” twee verschillende kanalen:

1. Generieke Bestandsuitwisseling (GBU), een module binnen Portaal, waarmee marktpartijen via een GUI of via een webservice bestanden (middels SOAP with Attachment) kunnen uitwisselen;
2. SFTP, waarbij marktpartijen via een separate server met SSH File Transfer Protocol (SFTP) bestanden kunnen aanbieden of ophalen.

De notificatie met meta-informatie over de aangeboden dataset en de dataset (XML en/of CSV) zijn ontkoppeld. Dit is in lijn met het Claim-Check patroon.



Figuur 47: Aanbieden dataset en notificatie met meta-informatie gescheiden

Voor de uit te wisselen bestanden zullen, conform afspraken in de sector, CSV en/of XML berichtdefinities gelden. Deze berichtdefinities zijn beschreven in de servicebeschrijving waar het uit te wisselen bestand onderdeel van uit maakt.

Deze service FileExchange is beschreven in servicebeschrijving Uitwisseling datasets.

6 Webservices

6.1 Inleiding

Systemen binnen EDSN worden volgens een service georiënteerde architectuur gerealiseerd. Het doel hierbij is de flexibiliteit in dienstverlening te vergroten, door 'loosely coupled' systemen. Eén van de belangrijkste doelstellingen bij het ontwerpen van webservice interfaces is het snel mogelijk maken van (nieuwe) business processen. Tevens moet de systeemintegratie met zowel de eigen systemen als systemen van de ketenpartners worden ondersteund.

Binnen de context van servicegeoriënteerde architectuur, is het ontwerp van de webservice interface van groot belang. Een slecht ontworpen webservice interface kan de ontwikkeling van applicaties, die deze services gebruiken, nodeloos complex maken. Vanuit een business perspectief kan een slecht ontworpen webservice interface een negatief effect hebben op flexibiliteit van business processen. Daarentegen kunnen goed ontworpen webservice interfaces de ontwikkelsnelheid van applicaties en de flexibiliteit van business processen verbeteren.

In de volgende paragrafen worden enkele belangrijke best-practices beschreven die EDSN hanteert bij de bij het ontwerpen van de webservice interfaces¹³.

6.2 Ontwerp

Voor het ontwerpen en ontwikkelen van webservices kunnen er drie verschillende aanpakken worden onderscheiden:

- **Bottom up:** Hierbij wordt eerst de logica geïmplementeerd en wordt binnen de ontwikkelomgeving op basis van de code de interfacedefinitie gegenereerd;
- **Top down:** Hierbij wordt eerst de interfacedefinitie met behulp van WSDL-definities en XML Schema Definities (XSD's) ontworpen en wordt er vervolgens in de ontwikkelomgeving een raamwerk voor implementatie van de logica gegenereerd;
- **Meet in the Middle:** In deze aanpak wordt tegelijkertijd begonnen met het ontwerpen van de interface (in WSDL en XSD's) en de implementatie in de code. Vervolgens wordt de interfacedefinitie en de logica zo goed mogelijk op elkaar afgestemd en waar nodig aangepast.

Hoewel de "Bottom-up" aanpak erg geliefd is bij ontwikkelaars en het de snelheid van ontwikkeling van individuele services kan bevorderen, is deze aanpak niet aan te raden omdat het onvoldoende een integrale aanpak is en het veelal de aansluiting met het bedrijfsproces mist. Daardoor worden datatypes niet integraal gedefinieerd en kan het bedrijfsprocessen inflexibel maken.

Best practice: Gebruik een "Top down" en "Meet in the middle" ontwerp- en ontwikkelaanpak in plaats van "Bottom-up". Ontwerp de webservice interface op basis van XML Schema Definities (XSD's) en WSDL-definities en genereer op basis hiervan het coderaamwerk.

6.3 Interface granulariteit

Een webservice interface zal over het algemeen meer dan één operatie moeten bevatten! Operaties zouden functioneel gegroepeerd moeten worden in webservice interfaces.

Een groot aantal webservice interfaces met alleen één operatie, illustreert een slecht interface-ontwerp. Hierbij is er een groot risico dat functionaliteit meervoudig wordt geïmplementeerd en er onvoldoende hergebruik gemaakt wordt van bestaande functionaliteit.

Anderzijds illustreert een zeer klein aantal webservice interfaces met een groot aantal operaties per webservice interface eveneens een slecht interface-ontwerp. Daarbij zijn de services moeilijk te begrijpen waardoor de vertaling naar een goed gebruik binnen het bedrijfsproces moeilijk is.

¹³ Gebaseerd op IBM Best Practices for Service Interface Design in SOA (<http://www.ibm.com/developerworks/architecture/library/ar-servdsgn1/>).

Best practice: Een webservice interface zou over het algemeen meer dan één operatie moeten bevatten. Operaties dienen functioneel gegroepeerd te worden in webservice interfaces. Procesonafhankelijke operaties dienen functioneel gegroepeerd te worden in separate webservice interfaces ter voorkoming van redundantie.

6.4 Synchronische vs. asynchrone operaties

Een webservice interface kan op basis van functionele groepering meerdere operaties bevatten. Het is echter om twee redenen niet aan te raden om synchrone en asynchrone operaties te combineren in één webservice interface:

- Asynchrone operaties gaan anders om met retourberichten en foutafhandeling dan synchrone operaties;
- Asynchrone operaties vereisen op systemen veelal andere instellingen voor de verwerking.

Het combineren van synchrone en asynchrone operaties in één webservice interface leidt tot complexe interface definities en slecht te 'tunen' services.

Best practice: Combineer geen synchrone en asynchrone operaties in één webservice interface. Wanneer beide typen gewenst zijn, creëer dan twee separate webservice interfaces.

6.5 Aanlever vs. opvraag operaties

In een “aanlever” bericht wordt informatie aangeleverd die afgehandeld moet worden. De operatie retourneert een bevestiging ontvangst van het aangeleverd bericht. Deze operatie wordt gekenmerkt als een ingaande operatie.

In een “opvraag” bericht wordt informatie opgevraagd. De operatie retourneert de opgevraagde informatie. Deze operatie wordt gekenmerkt als een uitgaande operatie.

Deze beide type operaties mogen niet gecombineerd worden in één webservice interface omdat aan ontvangende en verwerkende systemen andere eisen worden gesteld aan de verwerking van informatie door deze operaties. Dit leidt tot een verschil in service ontwerp waarbij de “aanlever” service voor het grootste deel asynchroon kan worden opgezet (de bevestiging volgt als het bericht verwerkbaar is en voordat het daadwerkelijk verwerkt is) en de “ophaal” service synchroon is (alle gegevens dienen binnen de afgesproken termijn direct geretourneerd te worden). Dit verschil in ontwerp vereist ook verschil in deployment inrichting waarbij vaak voor de “ophaal” service meer systeemresources gealloceerd dienen te worden.

Best practice: Combineer geen ingaande en uitgaande operaties in één webservice interface.

6.6 Statefull vs. Stateless Interfaces

Informatieuitwisseling met webservices kan 'statefull' of 'stateless' zijn. Van 'statefull' is sprake wanneer de aangeroepen webservice informatie bewaart en deze in volgende aanroepen door de client van invloed is. Bij 'stateless' is er geen directe afhankelijkheid tussen verschillende aanroepen van een webservice. Binnen de Service Georiënteerde Architectuur is het gewenst om services 'stateless' te laten zijn. 'Stateless' services kunnen eenvoudig atomair door clients aangeroepen worden, zonder dat daarbij rekening moet worden gehouden met eerdere aanroepen. Bij de aanroep moet alle informatie meegeleverd worden die nodig is voor een volledige atomaire verwerking van de aanroep.

Best practice: Definieer webservice interfaces alleen 'stateless'. De aanroep moet dan alle informatie bevatten die vereist is voor een volledige atomaire verwerking van de aanroep.

6.7 Beschrijving webservice

Webservices worden beschreven in de Web Service Definition Language (WSDL). De WSDL documenten worden met de LUXE-tool gegenereerd vanuit de LOM's in het EDSN EA model.

Als voorbeeld volgt de WSDL voor de melding van een leveranciersswitch:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="urn:nl:edsn:service:changeofsupplier:trans:1"
xmlns:bm3="urn:nl:edsn:service:changeofsupplierrequest:env:1"
xmlns:bm4="urn:nl:edsn:service:changeofsupplierresponse:env:1" name="ChangeOfSupplier"
targetNamespace="urn:nl:edsn:service:changeofsupplier:trans:1"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

  <wsdl:types>
    <xs:schema targetNamespace="urn:nl:edsn:service:changeofsupplier:trans:1"
elementFormDefault="qualified">
      <xs:import namespace="urn:nl:edsn:service:changeofsupplierrequest:env:1"
schemaLocation="../../Schemas/EDSN/PortaalDocs/ChangeOfSupplierDocs_1/ChangeOfSupplierRequestEnvelope.xsd"/>
      <xs:import
namespace="urn:nl:edsn:service:changeofsupplierresponse:env:1"
schemaLocation="../../Schemas/EDSN/PortaalDocs/ChangeOfSupplierDocs_1/ChangeOfSupplierBatchResponseEnvelope.xsd"/>
    </xs:schema>
  </wsdl:types>

  <wsdl:message name="RequestMsg">
    <wsdl:part name="ChangeOfSupplierRequestEnvelope"
element="bm3:ChangeOfSupplierRequestEnvelope"/>
  </wsdl:message>

  <wsdl:message name="PositiveResponseMsg">
    <wsdl:part name="ChangeOfSupplierBatchResponseEnvelope"
element="bm4:ChangeOfSupplierBatchResponseEnvelope"/>
  </wsdl:message>

  <wsdl:portType name="ResponderChangeOfSupplierRespondingActivity">
    <wsdl:operation name="ChangeOfSupplierRequest">
      <wsdl:input message="tns:RequestMsg"/>
      <wsdl:output message="tns:PositiveResponseMsg"/>
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="ResponderChangeOfSupplierRespondingActivityBinding"
type="tns:ResponderChangeOfSupplierRespondingActivity">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="ChangeOfSupplierRequest">
      <soap:operation soapAction="ChangeOfSupplierRequest" style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

  <wsdl:service name="ResponderChangeOfSupplierRespondingActivityService">
    <wsdl:port name="ResponderChangeOfSupplierRespondingActivityPort"
binding="tns:ResponderChangeOfSupplierRespondingActivityBinding">
      <soap:address
location="http://localhost:7001/ResponderChangeOfSupplierRespondingActivity"/>
    </wsdl:port>
  </wsdl:service>

</wsdl:definitions>
```

Een WSDL beschrijft altijd een (technisch-synchrone) webservice. De XML Schema's van request en response (of van notification en acknowledgement) worden vermeld, alsmede de operations en de poorten waar die operations worden uitgevoerd.

7 XML Schema's

7.1 Principles

De XSDs en WSDLs worden samengesteld op basis van NDR 3.0 [UNDR] en WS-I Basic Profile 1.1. Berichten worden geconsolideerd tot aan het principe van "1 Bericht = 1 XML-schema".

De XML Schema's (XSD's) voor de berichtdefinities worden met de LUXE-tool gegenereerd. Dit gebeurt conform de NDR 3.0 [UNDR]. Deze XSD's hebben de zogenaamde "Venetian Blind" structuur. Dat betekent dat ieder element als type apart wordt gedefinieerd. Datatypes kunnen worden hergebruikt in verschillende elementen.

Per bericht wordt er een XSD uitgegeven waarin alle gebruikte types zijn gedefinieerd. De XSD's zijn dus zelfvoorzienend. De keuze van de Technische Commissie om de berichten op deze manier te definiëren is ingegeven om de informatie uitwisseling in de marktprocessen nietodeloos afhankelijk van (realtime) inspecties en raadplegingen in XSD's elders te maken.

De XML tagnamen worden automatisch afgeleid uit de in het UML class model gebruikte namen. Deze zijn opgemaakt in CamelCase. Waar qualifiers in de namen zijn verwerkt komen ook underscores voor om die qualifiers van de rest van de naam te scheiden.

In het UML class model kunnen Choice constructies worden gebruikt. Bij het genereren van het XML-schema wordt een choice-group gemaakt met als kinderen de geassocieerde objecten van die choice-group.

7.2 Namespace

De XML namespace is een W3C aanbeveling voor toevoegen van context aan XML elementen. De XML namespace wordt door EDSN toegepast voor het toevoegen van proces context, versie informatie en status aan de XML elementen in de XML berichtdefinities. De XML namespace moet een Uniform Resource Identifier (URI) zijn, hetgeen de keuze laat tussen een Uniform Resource Name (URN) of een Uniform Resource Locator (URL). UN/CEFACT adviseert de URN voor de XML namespace [UNDR].

De XML namespace wordt afgeleid van de aan het UML class model toegevoegde "tagged values" en wordt als volgt opgebouwd conform de richtlijn van UN/CEFACT [UNDR]:

```
urn:<organization>:<hierarchy>:<schematype>:<proces>:<majorVersion>:<status>
```

Dit resulteert in de volgende XML namespace in de berichtdefinitie:

```
urn:nedu:edsn:data:changeofsupplierrequest:1:draft
```

Hierbij is "changeofsupplierrequest" de naam van de berichtdefinitie zoals deze als UML class model met typing CCMA (Core Component Message Assembly) is gemodelleerd.

In paragraaf 4.8 worden de aan het UML class model toegevoegde "tagged values" nader toegelicht.

7.3 Namespace prefix

De XML namespace wordt in de XML berichtdefinitie gedefinieerd en wordt voorzien van een zgn. prefix. Deze prefix is feitelijk een "placeholder" voor de namespace. De prefix is een lokaal gedefinieerde variabele in de XML berichtdefinities en in het XML bericht. Dat is ook zo expliciet beschreven in de W3C documentatie:

"Note that the prefix functions only as a placeholder for a namespace name. Applications SHOULD use the namespace name, not the prefix, in constructing names whose scope extends beyond the containing document".

Voorbeeld 1

Namespace: urn:nedu:edsn:data:changeofsupplierrequest:0:standard

Prefix: ccma1

```
<?xml version="1.0" encoding="UTF-8"?>
<ccma1:ChangeOfSupplierRequestEnvelope
xsi:schemaLocation="urn:nedu:edsn:data:changeofsupplierrequest:0:standard
ChangeOfSupplierRequest_0p92.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccma1="urn:nedu:edsn:data:changeofsupplierrequest:0:standard">
  <ccma1:Portaal_Content>
    <ccma1:Portaal_MeteringPoint>
      <ccma1:EANID>871694840000040168</ccma1:EANID>
      <ccma1:GridOperator_Company>
        <ccma1:ID>8716948000003</ccma1:ID>
      </ccma1:GridOperator_Company>
    </ccma1:Portaal_MeteringPoint>
  </ccma1:Portaal_Content>
</ccma1:ChangeOfSupplierRequestEnvelope>
```

Voorbeeld 2

Namespace: urn:nedu:edsn:data:changeofsupplierrequest:0:standard

Prefix: ccma8

```
<?xml version="1.0" encoding="UTF-8"?>
<ccma8:ChangeOfSupplierRequestEnvelope
xsi:schemaLocation="urn:nedu:edsn:data:changeofsupplierrequest:0:standard
ChangeOfSupplierRequest_0p92.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ccma8="urn:nedu:edsn:data:changeofsupplierrequest:0:standard">
  <ccma8:Portaal_Content>
    <ccma8:Portaal_MeteringPoint>
      <ccma8:EANID>871694840000040168</ccma8:EANID>
      <ccma8:GridOperator_Company>
        <ccma8:ID>8716948000003</ccma8:ID>
      </ccma8:GridOperator_Company>
    </ccma8:Portaal_MeteringPoint>
  </ccma8:Portaal_Content>
</ccma8:ChangeOfSupplierRequestEnvelope>
```

De XML namespace is in beide bovenstaande voorbeelden identiek. De prefix is een lokaal gedefinieerde variabele in het XML bericht. Deze kan afwijken, maar heeft vanuit XML perspectief geen impact.

7.4 Annotatie

In het model is aanvullende documentatie opgenomen, zoals de Nederlandse termen, de definities en de CCTS Dictionary Entry Names (DEN).

De tagged value “semanticAnnotations = True” in het Enterprise Architect berichtmodel zorgt voor het toevoegen van de W3C Semantic Annotations in de XML Schema’s voor ontwikkel- en testdoeleinden. Met deze informatie kunnen informatie analisten en ontwikkelaars zonder misinterpretatie herkennen welke definities in een XML Schema zijn gebruikt en repeterende mappings automatiseren. Het toevoegen van semantische annotaties is vastgelegd in TC020.

De tagged value “coreCompenentsDocumentation = True” in het Enterprise Architect berichtmodel zorgt voor het toevoegen van CCTS metadata conform de CCTS specificatie. Hiermee wordt verwezen naar metadata in de berichtenbibliotheek, zoals de ‘officiële’ naam en de definitie van een element.

Formaat-restricties, die in het UML class model zijn weergegeven als “tagged values”, worden in de XSD’s opgenomen als facets. Facets zijn de W3C formaat restricties, zoals die in het UML-model zijn vastgelegd, doorvertaald naar W3C XML Schema length, minLength, maxLength, pattern, enumeration, maxInclusive, maxExclusive, minExclusive, minInclusive, totalDigits en fractionDigits.

7.5 Versionering

Conform de UN/CEFACT XML Naming and Design Rules [UNDR] is de versionering op de berichten opgedeeld in twee soorten versies: minor en major.

Minor versionering

Minor versies garanderen compatibiliteit van een berichtdefinitie met voorgaande en opvolgende minor versies van die berichtdefinitie binnen dezelfde major versie. Minor versionering zoals gedefinieerd door de UN/CEFACT zorgt voor 'backward' en 'forward' compatibiliteit.

Minor versies worden gecreëerd indien er 'backward' compatibele aanpassingen aan de berichtdefinitie moeten worden doorgevoerd zoals het toevoegen van nieuwe, optionele, elementen.

Kenmerken van minor versie updates:

- 'upward' compatibiliteit (oude clients kunnen de service, met de gewijzigde berichtdefinitie, aanroepen);
- 'backward' compatibiliteit binnen dezelfde major versie (nieuwe clients en services ondersteunen tevens de oude versie);
- ongewijzigde namespaces en endpoints.

De eerste twee kenmerken van minor versie updates zijn voor de gegevensuitwisseling tussen de energiemarktpartijen niet van toepassing. Al deze partijen implementeren dezelfde versies van de berichtdefinities.

In release management wordt de afweging gemaakt of een wijziging als minor update kan worden uitgebracht of dat impact vereist dat er een major update moet worden uitgebracht. Deze afweging wordt gemaakt conform de regels in UN/CEFACT XML Naming and Design Rules [UNDR].

[R A735]	Minor versioning MUST be limited to declaring new optional XML content, extending existing XML content, or refinements of an optional nature.	1
----------	--	---

Major versionering

De major versionering wordt gebruikt indien de berichtdefinitie significant wijzigt. Er is geen 'backward' of 'forward' compatibiliteit met andere major versies. Indien een XML instantie, gebaseerd op een oudere major versie, gevalideerd gaat worden tegen het XML Schema van een nieuwe major versie, kunnen foutmeldingen verwacht worden.

Major versies worden gecreëerd indien er 'non-backward' compatible aanpassingen aan de berichtdefinitie moeten worden doorgevoerd zoals het wijzigen van namen van elementen en wijzigen van cardinaliteit.

Kenmerken van major versie update:

- geen 'backward' of 'upward' compatibiliteit met voorgaande major versie;
- gewijzigde namespaces;
- gewijzigde namen van berichtdefinities;
- significante wijzigingen in de berichtdefinitie.

Deze versionering staat toe dat verschillende major versies van services naast elkaar kunnen worden aangeboden. In principe zal er echter op enig moment één major versie van een webservice worden ondersteund.

De implementatie van de versionering wordt conform NDR 3.0 [UNDR] uitgewerkt.

Versionering wordt geïmplementeerd in het version attribuut van het xsd:schema element:

```
<xsd:schema ... version="<major versie>.<minor versie>">
```

De major versie wordt bovendien weergegeven in de namespace:

```
urn:nedu:edsn:data:<proces>:<major versie>:<status>
```

Voorbeeld: urn:nedu:edsn:data:changeofsupplierrequest:1:draft

7.6 Character Encoding

XML is een tekst gebaseerd uitwisseling formaat. Echter, teksten kunnen in verschillende talen geschreven worden, met uitzonderlijke tekens, van Japanse Kanji tot en met cyrillische tekens in het Grieks. Hiervoor is de bestaande ASCII tekenset/encoding, met maar 128 mogelijke tekens, te beperkt. Binnen de XML specificatie is hierop ingespeeld en gekozen voor de –vrije- mogelijkheid om zelf de encoding te kiezen en deze aan te geven in de XML declaratie (zie paragraaf 7.7).

Als deze XML declaratie ontbreekt in een XML bericht, zoals mogelijk is binnen de XML 1.0 standaard, of de encoding is niet opgenomen in de XML declaratie, is per default een XML bericht UTF-8 of UTF-16. De XML parser moet dat dan afleiden aan de aanwezigheid van een Byte-Order-Mark (BOM), zoals gesteld in the UTF specificaties.

De XML 1.0 standaard stelt:

- UTF-8 en UTF-16 encoding moeten ondersteund worden door XML parsers;
- het is niet verplicht, maar ondersteuning van meer encodings, zoals de “westerse” ISO-8895-1, wordt wel aangemoedigd.

Daar de XML standaard per definitie aangeeft dat UTF-8 en UTF-16 op z’n minst ondersteund moeten worden, ligt het voor de hand te kiezen uit één van deze twee encodings. De keuze uit één van deze twee encodings geeft de zekerheid dat elk XML verwerkend systeem, framework of programmeertaal hiermee overweg kan.

Daarbij is gekozen voor UTF-8 op basis van onderstaande overweging:

- zowel UTF-8 als UTF-16 kunnen alle tekens ;gerepresenteerd worden. Dus dat is niet de limiterende factor. UTF-8 gebruikt echter 1 byte per teken of coderingskenmerk, UTF-16 2 bytes;
- een standaard ASCII document kan één-op-één als UTF-8 worden ingelezen;
- UTF-16 is default voor tekst manipulaties binnen MS Windows omgevingen en .NET toepassingen (alhoewel ze ook prima met UTF-8 kunnen converteren naar- en van UTF 16);
- UTF-16 XML berichten zijn per definitie groter dan UTF-8 berichten (tot 2x zo groot);
- XSD’s, die de structuur van XML berichten beschrijven, zijn in UTF-8 formaat, [R88E2 UN/CEFACT: Every XML Schema MUST use UTF-8 encoding]. Veel software die berichten aanmaken aan hand van een XSD, nemen de encoding over van de XSD, ook al is dat niet vastgelegd als standaard. Zie paragraaf 7.7;
- conversie van UTF-x formaat naar “native” formaat van applicatie kost extra tijd. Er zijn meer software applicaties die “native UTF-8” gebruiken dan “UTF-16”;
- XML berichten voor koppelvlakstandaard WUS Digikoppeling, Logius, gedefinieerd voor uitwisseling met de Nederlandse overheid, hebben altijd UTF-8 encoding.

Met de keuze voor UTF-8 als standaard encoding kunnen ontvangers van EDSN XML berichten er van uit gaan dat ze geen rekening hoeven te houden met andere encodings in EDSN XML berichten.

7.7 XML declaratie

NDR3 [UNDR] schrijft voor dat XML Schema en XML berichten conform W3C XML Schema 1.0 zijn:

[R 8059]	All XML Schema design rules MUST be based on the W3C XML Schema 1.0 Recommendation: XML Schema Part 1: Structures Second Edition and XML Schema Part 2: Datatypes Second Edition .	1
----------	--	---

De XML declaration is in W3C XML Schema 1.0 Recommendation een SHOULD. Deze XML declaratie zou men dus achterwege kunnen laten. Echter, conform NDR3 [UNDR] moet de XML declaratie wel opgenomen worden:

[R 9EDA]	Every UN/CEFACT XML Schema File must start with an XML declaration that specifies the xml version and encoding being used.	1
[R AABF]	Every UN/CEFACT XML Schema File must use XML version 1.0.	1
[R 88E2]	Every UN/CEFACT XML Schema File MUST use UTF-8 encoding.	1

EDSN Berichtencentrum neemt in XML Schema en XML berichten de onderstaande XML declaration op:

```
<?xml version="1.0" encoding="UTF-8"?>
```

7.8 complexType name

In de lijn van NDR 3.0 [UNDR] moet de naam van type definities (complexType name) uniek zijn in het XML Schema. Dat is te realiseren door:

1. aan elke complexType name een additionele code toe te voegen;
2. aan elke complexType name additionele naamgeving toevoegen.

Optie 2 wordt nu toegepast omdat daarmee een complexType name als type definitie een logische naam krijgt en daarmee een XML Schema wat beter leesbaar is voor technici. Echter, deze constructie kan leiden tot zeer lange complexType name.

Berichtencentrum volgt de TC aanbeveling om, ter voorkoming van zeer lange complexType name in de XML Schema's, gestandaardiseerde afkortingen (mnemonische codes) te gebruiken.

Zonder gestandaardiseerde afkorting:

```
<xsd:element name="BalanceSupplier_Company" type="ccma:MoveInResponseEnvelope_Portal_Content_Portal_MeteringPoint_MPCommercialCharacteristics_BalanceSupplier_Company" sawsdl:modelReference="urn:nedu:edsn:lom:portaal:model:1:standard#BalanceSupplier_Company">
```

Met gestandaardiseerde afkorting:

```
<xsd:element name="BalanceSupplier_Company" type="ccma:MoveInResponseEnvelope_PC_PMP_MPCC_BalanceSupplier_Company" sawsdl:modelReference="urn:nedu:edsn:lom:portaal:model:1:standard#BalanceSupplier_Company">
```

Via natuurlijk proces zullen XML Schema's met langere complexType name "uitsterven" bij nieuwe releases. Tot die tijd is er, daar waar nodig en zinvol, en variant beschikbaar met verkorte complexType name.

8 CSV berichtstructuur

8.1 Inleiding

Een CSV-bericht bestaat uit één of meer aantal regels (=records), gescheiden door regeleinde. Elk record bestaat uit velden, gescheiden door een ander teken, meestal een komma. Een ander scheidingsteken is ook mogelijk¹⁴.

Een echte standaard voor CSV is er niet, in de praktijk wordt veelal RFC 4180 als uitgangspunt gehanteerd. Dit uitgangspunt, met als doel afspraken over indeling en formaat voor CSV voor de Nederlandse energiemarkt vast te leggen, is eveneens gehanteerd voor de NEDU/EDSN Ontwerpkeuzes.

Elk CSV-bericht wordt, gelijk de XML berichtmodellen, gemodelleerd in Enterprise Architect. Daarbij worden de in de XML berichtmodellen gehanteerde namen overgenomen in de CSV berichtmodellen.

8.2 Afspraken CSV

- Het einde van een CSV record (regel) wordt aangegeven door een Carriage Return (CR)/Line Feed (LF);
- De laatste CSV record (regel) moet ook afgesloten worden met een Carriage Return (CR)/Line Feed (LF);
- In een CSV record mag geen Carriage Return (CR)/LineFeed (LF) voor komen;
- Velden worden gescheiden door een scheidingsteken, dit scheidingsteken is komma (,)¹³;
- Spaties voorafgaand of volgend op het scheidingsteken worden genegeerd;
- Ieder veld wordt ingesloten in dubbele aanhalingstekens ("), bijvoorbeeld "EUR";
- Indien een dubbel aanhalingsteken (") voor komt in een veld, dan moet deze voorafgegaan worden door additioneel dubbel aanhalingsteken;
- Als tekenset wordt ASCII gehanteerd¹⁵.

8.3 Indeling CSV

De eerste regel van een CSV is de EDSN Business Document Header, afgesloten met een CR/LF. Elke volgende regel is een record, afgesloten met een CR/LF.

8.4 Naamgeving CSV-bericht

Voor de naamgeving van de uit te wisselen bestanden wordt de volgende conventie gehanteerd:

<naam bericht>_<zender>_<ontvanger>_<JJJJMMDD>_<VOLGNR>.<EXTENSIE>

waarbij:

<naam bericht> = naam van het bericht;

<zender> = EAN13-code van de zender;

<ontvanger> = EAN13-code van de ontvanger;

<JJJJMMDD> = datum waarop het bestand is aangemaakt;

<VOLGNR> = het volgnummer binnen de combinatie van dit type bestand en aanmaakdatum. Het volgnummer bestaat altijd uit een oplopend getal van twee cijfers, startend met "01";

<EXTENSIE> = bevat de waarde "csv".

De naamgeving is niet hooflettergevoelig (niet case sensitive).

Voorbeeld: "ContractRenewal_8712423010383_8712423010208_20120801_01.csv".

Gedetailleerde informatie over naamgeving van CSV-bericht is opgenomen in elke servicebeschrijving waar uitwisseling van berichten op basis van CSV van toepassing is.

¹⁴ Voor C-ARM RG0.2 informatie verstrekking is de | (pipe) als scheidingsteken gedefinieerd.

¹⁵ Is in lijn met RFC 4180. In geval van diakrieten moet dat Extended ASCII (ISO 8859-1) zijn.

8.5 Datum en datum/tijd notatie in CSV-bericht

Datum

Datum notatie in de CSV berichten moet conform de W3C DTF Note ISO 8601. Hierdoor wordt de eenduidige interpretatie van datum internationaal gewaarborgd.

Datum YYYY-MM-DD, bv: 2012-11-28. Geen tijdzone bij datum!

Datum/tijd

Datum/tijd notatie in de CSV berichten moet conform de W3C DTF Note ISO 8601. Hierdoor wordt de eenduidige interpretatie van datum en tijd internationaal gewaarborgd.

Er zijn meerdere tijdsdefinities, maar meest gangbaar en binnen de berichten gebruikt is de UTC. Bij gegevensuitwisseling is een tijdsdefinitie zonder referentiepunt een incomplete tijdsdefinitie. De definitie van een datum/tijd is pas compleet en ondubbelzinnig met een offset van UTC:

2011-03-31T10:15:30+02:00 (UTC offset van +02:00 uur);

2011-03-31T08:15:30Z (UTC).

De regel is om altijd de offset tot UTC verplicht te stellen. Het verplicht maken van de offset ten opzichte van UTC geeft de ontvangende partij de mogelijkheid zonder misinterpretatie over zomer/wintertijd en locatie de datum/tijd te verwerken.

De notatie van datum/tijd conform de W3C DTF Note ISO 8601 is vastgelegd in TC021.

8.6 Voorbeeld CSV

Regel	Voorbeeld
1	"2012-03-23T18:23:55Z","1234567890","871687120052440131","871687120052440179"
2 .. n	"871687120052440179","ELK","871687120052440063","2010-03-23","56234578","459865","45","50889,78","5024,33","19"

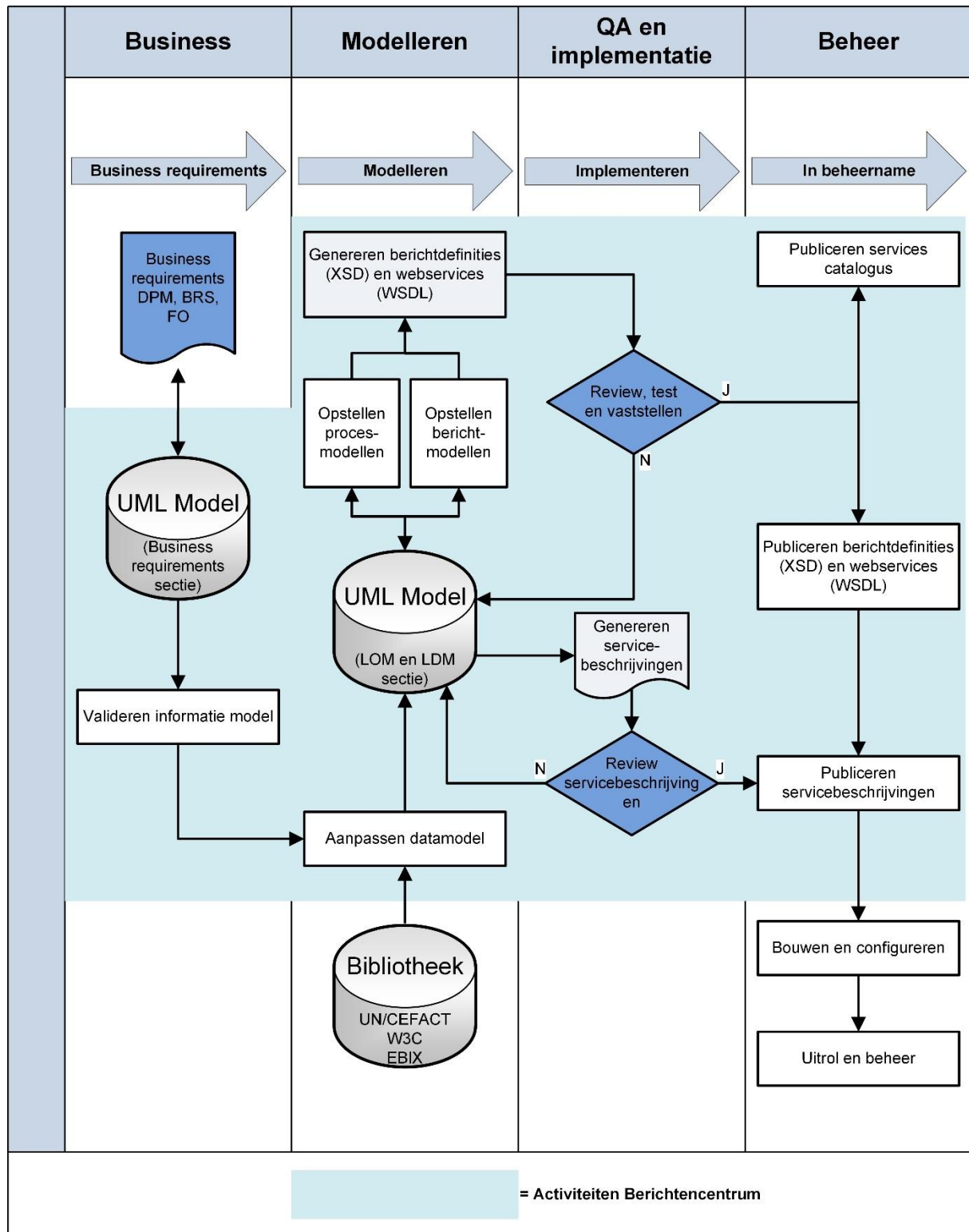
9 Werkwijze EDSN Berichtencentrum

Het Berichtencentrum produceert de gewenste bericht- en procesdefinities op basis van functionele input. Functionele input kan worden aangeleverd in de vorm van een Detail Proces Model (DPM), een Business Requirement Specification (BRS), een issuedocument of een Functioneel Ontwerp (FO). Vooraf wordt door het Berichtencentrum beoordeeld of de (meestal per procescluster of langs systeem grenzen) aangeleverde functionele informatie compleet en consistent is, door de requirements met behulp van Enterprise Architect (EA) te modelleren. Dit requirementsmodel bestaat uit weergaven van het berichtuitwisselproces en van de per bericht uit te wisselen gegevens. Modelelementen uit andere processen worden zoveel mogelijk hergebruikt. De volgorde van interacties in de tijd wordt weergegeven in UML Sequence diagrammen. Bedrijfsprocessen worden vanuit een generiek perspectief (zowel voor business als IT mensen herkenbaar) beschreven in Business Process Modelling Notatie (BPMN).

De requirementsmodellen worden verwerkt in een Logisch Object Model (LOM), gebruikmakend van generieke objecten op bovenliggend LDM-niveau, tot een consistent informatiemodel van het betreffende toepassingsdomeinen. Vanuit het LOM worden de berichtmodellen gegenereerd.

De berichtdefinities en webservices om de berichten in de betreffende (markt)processen mee uit te wisselen worden vervolgens vanuit het LOM en de berichtmodellen gegenereerd. De berichtdefinities worden door de functionele betrokkenen gereviewd, door ze samen met de requirementsmodellen te presenteren. Vervolgens worden ze met andere applicaties (XML Spy, SOAtest en SOAPui) getest.

In Figuur 48 staat een schematisch overzicht van de werkwijze. De activiteiten met een lichtblauwe achtergrond zijn activiteiten van het Berichtencentrum.



Figuur 48: Werkwijze Berichtencentrum

De berichtdefinities worden gepubliceerd op *mijnEDSN* en *mijnNEDU*. Daarnaast staat ook het model op aanvraag ter beschikking van de marktpartijen. Bovendien worden de berichtdefinities zowel technisch als functioneel gedocumenteerd in de Servicebeschrijvingen.