

Issue TC007 Idempotentie

Status:	Vastgesteld		
Versienummer:	1.0	Versiedatum:	3 maart 2011
Documentnaam:	TC007 Idempotentie		

Behandeld door de Technische Commissie

Geaccepteerd door de Issue Commissies:	IC-Klant: 3 februari 2011	IC-WE:	IC-WG:
--	------------------------------	--------	--------

Raakt aan:	Kleinverbruik	Grootverbruik	Wholesale
Elektriciteit	X	X	X
Gas	X	X	X

Impact op keten	Ja/nee
berichten	nee
processen	nee
systemen	ja
overigen	nee

Vastgesteld door NEDU op 2 maart 2011

Invoering in markt per eerstvolgende implementatie gebaseerd op de EDSN xml berichtenstandaard

Contact	
Technische Commissie	TC@edsn.nl

Situatieschets:

De betrouwbaarheid en robuustheid van de ketenprocessen en speciaal de berichten uitwisseling daarin is een belangrijke voorwaarde voor de werking van de geliberaliseerde markt. Om te kunnen zekeren dat verzoeken bij de ontvanger aan komen (en verwerkt worden) is het van belang dat een verzoek bij uitblijven van een tijdige response nogmaals gesteld kunnen worden. Daartoe dient het ontvangende en verwerkende systeem wel om te kunnen gaan met herhaalde (identieke) verzoeken en mag het verzoek niet leiden tot (onterechte) functionele verwerking. Door deze robuustheid in te richten, zoals als standaard is gedaan op het CAR/EMP van EDSN, voorkomen we dat er veel onderlinge discussie ontstaat over vermeend missende berichten of foutieve (onterechte) verwerkingen.

Deze robuustheid kunnen we realiseren met behulp van het begrip "Idempotentie". Hiermee wordt bedoeld dat een systeem of stelsel van processen en systemen om kan gaan met herhaalde meldingen/verzoeken/opdrachten, zodanig dat het antwoord op het herhaalde verzoek hetzelfde is als het originele antwoord en tevens dat het systeem niet ontregeld wordt door herhaalde verzoeken (bijvoorbeeld doordat de verzoeken meerder keren ten onrechte worden uitgevoerd en tot onherkenbare effecten leiden).

Vraag:

Tot op welk niveau willen we in de ketenprocessen de idempotentie inrichten?

Anders gesteld willen we het mogelijk maken dat op een (technisch) zelfde vraag ook hetzelfde antwoord komt – waarbij het ontvangende functionele systeem slechts eenmaal acteert (of muteert) (niet verstoord raakt door herhaalde vragen) en waardoor het vragende functionele en technische systeem door kan vragen tot een antwoord is ontvangen (kortom: 'er verdwijnen geen berichten').

Overwegingen

De uitwisseling van berichten kan worden ontleed in een functionele en een technische component:

- Functioneel: de business omgeving bepaalt de inhoud van het bericht en het gewenste effect (verwerking) van de berichtuitwisseling.
- Technisch: dit zijn functioneel gezien passieve componenten, die er voor zorgen dat het bericht van het ene functionele systeem naar het andere komt.

Om op functioneel niveau (in de business applicaties en processen) in staat te zijn om herhaalde opdrachten te versturen en te ontvangen (idempotent te zijn), moet er veel extra ingericht worden en moeten er veel afspraken tussen al de functionele, logische en technische componenten in de hele keten gemaakt worden. Vaak is het eenvoudiger om een functionele interactie tussen systemen (marktpartijen) idempotent an sich te maken. Bijvoorbeeld in een bericht (verzoek of opdracht) om een bepaalde waarde van een item in te stellen kan men veiliger het commando geven door de gewenste waarde mee te geven dan een commando te ontwikkelen die de waarde met een percentage verlaagt: meerdere keren dezelfde waarde instellen levert hetzelfde resultaat, terwijl

meerdere keren de opdracht geven een waarde met 10% te verlagen decimeert de waarde. De meeste van onze ketenprocessen zijn qua karakter functioneel redelijk idempotent, zoals een switch en verhuizing: als een leverancier meerdere keren dezelfde functionele switchopdracht geeft zal deze over het algemeen maar eenmaal worden uitgevoerd en de vervolg opdrachten als onzinnig worden afgewezen.

Daarnaast is het voor een functionele omgeving wel erg aantrekkelijk te weten dat als een opdracht (bericht) is verzonden dat dan gegarandeerd ook een antwoord (responsbericht) terug komt (los van de inhoud). Dit is technische idempotentie, die relatief eenvoudig op een eenduidige manier is in te regelen: de berichtverwerkingseenheid van de functionele omgeving dient (identiek) hetzelfde bericht net zolang (binnen redelijke tijdsgrenzen) te versturen tot er een antwoord is ontvangen. De ontvanger dient hiertoe in te regelen dat een eenmaal ontvangen bericht uniek kan worden geïdentificeerd zodat eventueel herhaalde identieke berichten dezelfde respons naar de indiener geven zonder functioneel meerder malen te worden verwerkt.

Oplossing:

Functionele idempotentie alleen inregelen voor die processen die het daadwerkelijk nodig hebben.
Technische idempotentie standaard voor alle processen en berichten inregelen.

Het inregelen van de technische idempotentie betekent voor de zender dat hij zelf kan inregelen dat er gegarandeerd een respons op zijn verzoek komt: een herhaald verzonden verzoek (middels een bericht) wordt gegarandeerd beantwoord met hetzelfde antwoord alsof het verzoek de eerste maal zou zijn behandeld door de ontvanger/bewerker. Het inrichten van de technische idempotentie betekent voor de ontvanger dat zijn functionele (business)omgeving niet wordt 'lastig' gevallen met dezelfde opdrachten, die al in uitvoering zijn dan al zijn uitgevoerd.

Er kunnen zich voor de zender twee situaties voordoen bij het mislukken van een communicatie:

1. De initiator van de communicatie kan niet direct verbinding maken met de aangeroepen service en doet een (beperkt) aantal pogingen alsnog het contact tot stand te brengen (retry binnen dezelfde (technische) transactie). Hiervoor wordt gebruik gemaakt van de in te richten technische idempotentie-robustheid;
2. Situatie 1 (hierboven) heeft niet binnen het maximaal aantal gestelde pogingen of tijdspanne tot succes geleid. De initiator sluit de transactie en probeert op een later tijdstip opnieuw. (functionele retry van dezelfde logische (business) transactie middels een nieuwe technische transactie, met een nieuw uniek MessageID).

Een voorbeeld van simpele vorm van technische idempotentie (situatie 1) uit de dagelijkse praktijk is het simpelweg thuis verzenden van een email bericht: De gebruiker klikt (functioneel) eenmaal op verzenden, waarna het (technische) systeem meerdere keren zal proberen het bericht bij de ontvangende server te krijgen –tot er een ontvangstbevestiging van de provider ontvangen wordt (bij kleine en korte verstoringen en minder bandbreedte in de lijnverbinding wordt en herzendpoging ondernomen). Pas als het na enige minuten echt niet lukt zal de gebruiker een melding krijgen dat het bericht niet kan worden verzonden.

Op alle verzonden berichten dient binnen een zekere termijn (enkele seconden tot enkele minuten) tenminste een technische response van de ontvanger zijn ontvangen. Is dit niet het geval dan kan de zender op basis van het gezonden bericht, met identiek dezelfde inhoud en hetzelfde unieke MessageID in de header van het bericht zonder problemen hetzelfde verzoek opnieuw insturen

(zonder dat de functionele business laag (back-end) dit hoeft te weten). Dit kan net zolang worden herhaald tot er een technisch respons is ontvangen. Uiteraard zal er een zekere redelijkheid in acht moeten worden genomen. Als na een langere periode geen respons komt dan is er mogelijk een persistente storing (lijnverbinding ligt er langdurig uit, systemen down, etc.).

De ontvanger van berichten richt een structuur in dat een herhaald ontvangen bericht (op basis van het zelfde MessageID, berichtsoort en zender) dezelfde response krijgt: was er nog niet eerder een bericht ontvangen dan zal het als nieuw bericht worden doorgegeven aan de functionele omgeving. Was dit echter al gedaan (dan is er een ook een response gegenereerd en afgegeven aan de indiener) dan wordt hetzelfde response (kopie nu weer worden afgegeven aan de inzender).

Deze structuur geldt voor elke interactie tussen twee partijen. Als er sprake is van een tussenpersoon (zoals EDSN tussen marktpartijen in 1 keten) dan is de technische idempotentie tussen alle twee interacties in geregeld.

Deze faciliteit is primair bedoeld voor het technisch opnieuw proberen (retry) te versturen van berichten. De MessageID is 'geldig' tot het moment dat de initiator van een bericht de (technische) ontvangstbevestiging van de ontvanger van het bericht verkregen heeft. De ontvanger van het bericht dient de ontvangen MessageID een beperkte periode te bewaren. Dit om te kunnen onderkennen dat een verzoek mogelijk herhaald wordt ingediend. Voor de technische idempotentie wordt deze standaard waarde ingesteld op 1 uur. Deze waarde is op basis van inzichten en ervaringen muteerbaar bij gezamenlijke marktafspraak. Voor alle processen en berichten wordt deze standaard ingesteld en alleen voor die processen en berichten die specifiek andere eisen stellen zal dit afwijkend moeten worden ingeregeld.

Voorbeelden:

1. Een enkelvoudig bericht dat synchroon wordt afgehandeld dient in de respons een functioneel antwoord te ontvangen. Wordt het verzoek herhaald verzonden dan:
 - Ontvangt de ontvanger dit voor de eerste keer dan gaat hij dit gelijk verwerken en geeft antwoord (functionele afwijzing of respons);
 - Had de ontvanger het bericht al ontvangen en verwerkt dan stuurt deze het eerder gezonden responsbericht met de eerder geproduceerde functionele inhoud, zonder de functionele back-end nogmaals dezelfde vraag te stellen.
2. Een batchverzoek wordt asynchroon afgehandeld, hetgeen inhoudt dat de in de batch aangeboden transacties later separaat verwerkt worden en de indiener kan later de reacties ophalen en ontvangen (vaak in meerdere verschillende berichten). Wordt een batch ingediend, dan dient hier altijd een technische respons ("ontvangen en ik ga de inhoud verwerken" – of "bericht ontvangen maar niet te verwerken"(waarna de indiener de fout dient te herstellen om de verzoeken correct in te dienen)). Wordt een batch verzoek herhaald ingediend dan:
 - Wordt de batch verwerkt als de ontvanger dit bericht voor de eerste keer krijgt;
 - Wordt door de ontvanger hetzelfde eerder afgezonden technische response bericht gezonden als het bericht al wel eerder was ontvangen en de respons verzonden.
3. Het ophalen van functionele responses ("staan er nog antwoorden op eerder via batch ingediende verzoeken klaar?") wordt functioneel dagelijks meerdere keren herhaald met functioneel verschillende resultaten. Ontvangt de zender van het verzoek om responses geen reactie dan dient hij identiek hetzelfde verzoek (zelfde message-ID) nogmaals in:
 - Had de ontvanger het verzoek nog niet eerder ontvangen dan zal hij het functioneel laten verwerken en een functionele response geven (een batch vol antwoorden);

- Had de ontvanger van het verzoek al een reactie gegeven (een batch vol antwoorden) dan zal hij deze reactie (het betreffende bericht) opnieuw sturen.

Enige kenmerken:

- Voor de unieke en herhaalbare berichtuitwisseling op technisch niveau wordt een uniek nummer op EDSN Business Document Header niveau meegegeven, de MessageID. Wordt een bericht (verzoek) met hetzelfde MessageID herhaald ingediend, dan wordt het toch maar één keer door de ontvangende (functionele) applicatie verwerkt;
- Zou bij het ophalen van (batch) responses idempotentie niet ondersteund worden dan zou het onmogelijk zijn een eenmaal verstrekt antwoord nogmaals op te leveren. Om dit toch mogelijk te maken (idempotent), dient de aanvragende partij middels een verzoek met herhaalde MessageID een eenmaal verstrekt antwoordbericht nogmaals te kunnen opvragen;
- De ontvanger van een verzoek kan het MessageID hanteren om te bepalen of een verzoek 'hetzelfde' is als een eerder ontvangen verzoek en op basis daarvan besluiten het herhaald ontvangen verzoek niet nogmaals in behandeling te nemen. Het is hierbij wel van belang dat de ontvanger ter beantwoording (ontvangstbevestiging) hetzelfde antwoord geeft als bij beantwoording het eerst ontvangen verzoek;
- De ontvanger van een verzoek zal bij herhaald ontvangen van dezelfde MessageID nogmaals het laatst opgeleverde antwoord opleveren. Bij ontvangst van een nieuwe MessageID zal de ontvanger het bericht eerst functioneel verwerken en vervolgens het passende antwoord opleveren;
- Na het ontvangen van een afwijzing is het onzinnig hetzelfde bericht (zelfde MessageID) nogmaals te zenden. Er is een logische response en na functionele beoordeling en mogelijk aanpassing kan een nieuw verzoek worden ingediend (met een nieuw MessageID).

De EDSN Business Document Header is in alle nieuwe XML berichten ingevoerd en is gestandaardiseerd voor alle EDSN Business Object 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. In de EDSN Business Document Header zijn de MessageID, CreationTimestamp, ReceiverID (onderdeel van Destination) en SenderID (onderdeel van Source) verplicht.

De MessageID dient voor elk bericht uniek te zijn behalve voor een (bewust) herhaald verzoek. Om de ID uniek te maken wordt de UUID of GUID gebruikt, een globale standaard om op basis van de lokale omgeving en tijd een (per zender) uniek nummer te genereren.

Enige implementatie opmerkingen:

Het is zeer wenselijk dat alle partijen de voorgestelde technische idempotentie gaan invoeren.

In de klantgerichte processen is EDSN veelal in de rol van ontvanger: marktpartijen zijn aan de zendende kant, zij vragen verzoeken aan of dienen opdrachten in.

De implementatie aan zendende kant:

- Deze is zeer gewenst, echter niet verplicht maar wel een gemiste kans bij niet implementeren:
Door de herzend actie wel in te richten "op de uitgaande poort" na uitblijven van een tijdig respons bericht kan aan de

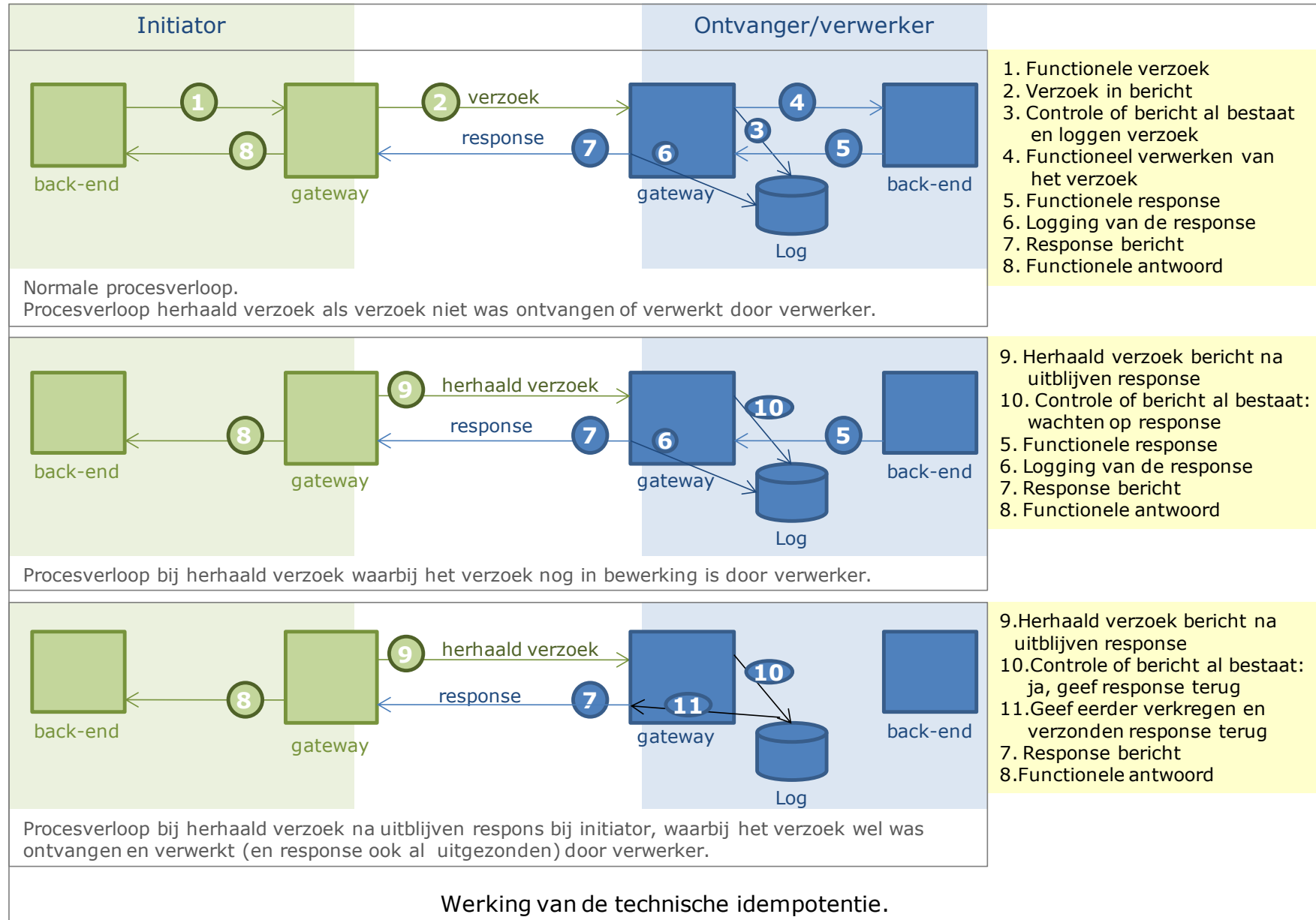
TC007: Idempotentie

functionele systemen van de eigen organisatie een zekere garantie van verwerking worden afgegeven en hoeft er niet in de keten gebeld te worden aangaande 'missende berichten'.

- De keuze voor het niet inrichten van de idempotentie houdt wel in dat men de keten niet zondermeer mag lastig vallen met vragen over vermeend missende berichten.

De implementatie aan ontvangende kant:

- Deze is verplicht.
Het biedt duidelijkheid en meerwaarde aan de ketenpartners en garandeert unieke verwerking van de aangeboden verzoeken (en daarmee betrouwbaarder systemen en registraties).



Invoeringsstrategie:

Zo snel mogelijk in te regelen om de mechanismen te kunnen toepassen meteen vanaf de invoering van de nieuwe XML berichten (zoals bij het nieuwe marktmodel en de P4) en nieuwe centrale systemen.

De voorgestelde technische idempotentie is al op CAR/EMP gerealiseerd.

Versieoverzicht:

- 0.1 Eerste aanzet door Ed van der Pijl en Roelof Dijkstra na bespreking in overleg 13 november 2009;
- 0.2 Tweede versie met inbreng van Roelof Dijkstra en Martijn van Glabbeek;
- 0.3 Aangepaste versie waarin commentaar van Roelof en Martijn deels is opgenomen. (11-jan 2010);
- 0.4 Aangepaste versie door Roelof Dijkstra en Ed van der Pijl. Afsplitsing van onderwerpen. TC007 bevat nu uitsluitend idempotentie o.b.v. MessageID. Gebruik van optionele velden in de EDSN Business Document Header is opgenomen in nieuw issue TC013. Dit om goedkeuring van deelissues niet van elkaar afhankelijk te laten zijn;
- 0.5 Aanpassingen door Ed van der Pijl. Verduidelijken scenario's;
- 0.6 Volledig herschreven door Gerrit Fokkema na keuze in de TC van 27 augustus 2010 voor uitsluitend technische idempotentie in te richten;
- 0.7 Toevoeging van platen met procesverloop n.a.v. bespreking in TC van 22 oktober 2010 en. Tevens bewaartijd (instelbaar) op 1 uur
- 0.8 Paar kleine redactionele verbeteringen na vergadering TC van 19 november 2010 en ter beoordeling naar IC-Klant;
- 0.91-0.93 Na commentaar van de Sectorplanningscommissie en de IC-Klant additionele toelichtingen na bespreking in de TC vergadering van 17 december 2010 en die van 14 januari 2011;
- 0.94 Benadrukking van gewenstheid tot implementatie door alle marktpartijen met akkoord door de IC-Klant op 3 februari 2011
- 1.0 Na vaststelling door de ALV op 2 maart 2011.