



Aan : Arend van Beek, Huub van Vliet, Hans Boer, Tom Horeweg, Jos Wolf

Van : Menno van den Berg, Alan Saadou

Cc : Nanne Osinga, Jules de Ruijter, Marielle Copier, Colin 't Hart, Patrick, Remco Vossen

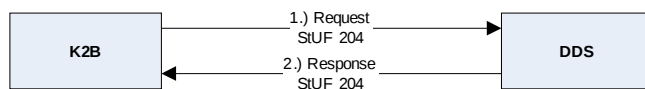
Datum : 20-05-2009

Versie : 1.03

Betreft: Testplan werking GUC + Gegevensmagazijn

1 Uitgangssituatie

Productie
omgeving PZR

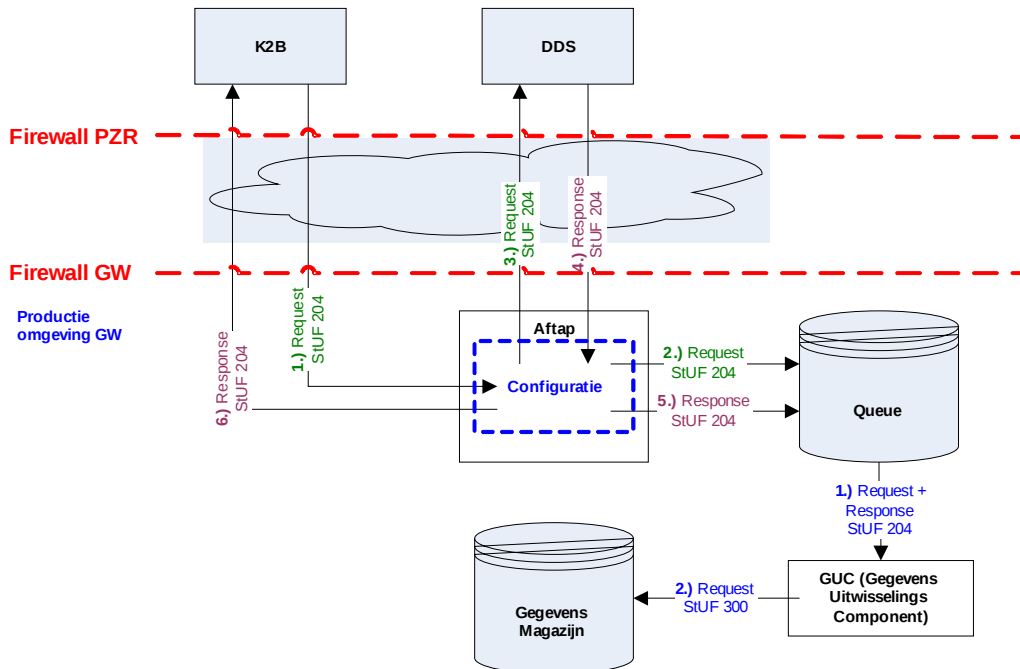


Figuur 1 - Huidige werking koppeling K2B – DDS

Binnen Key2Burgerzaken is de term "Bericht" bekend als "Kennisgeving" en de term "Response" als "Bevestiging". De termen Bericht en Response worden in dit testplan gehanteerd. Een bericht kan een toevoeging, wijziging of een correctie bevatten. De StUF berichten worden via SOAP over http verzonden. Het proces dat afgebeeld is in Figuur 1 is een synchroon proces; de response komt dus via dezelfde verbinding terug.

2 Gewenste situatie

Productie
omgeving PZR



Figuur 2 – Gewenste werking koppeling K2B – DDS

In bovenstaande figuur is de gewenste situatie op de productie omgevingen van PZR en GW weergegeven. Deze situatie zal eerst op de integratieomgeving, testomgeving en acceptatieomgeving worden getest alvorens een oplossing zal worden doorgezet naar de productie omgeving.

In de komende hoofdstukken gaan we in op de stappen die tot deze situatie zullen leiden. Bij elke stap zullen we de teststrategie uiteenzetten. De tests zullen we uiteenzetten als volgt:

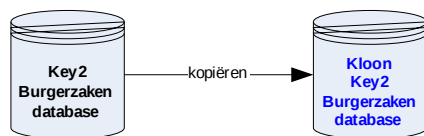
- Tests op de Test en Acceptatieomgeving
- Tests op de Productieomgeving

Aan de hand van de tests bij de desbetreffende component kan men controleren dat de component volgens specificatie werkt.

3 Snapshot van de K2B omgeving

Op tijdstip "t" moet een kopie van de K2B database gemaakt worden. Dit noemen we de K2B kloon. Deze database zal gebruikt worden om de initieel proxy van informatie te voorzien. Het doel van de kloon is het maken van een snapshot van de data zoals deze op tijdstip 't' aanwezig is in K2B. Het is dan ook belangrijk dat tijdens het maken van de kloon er geen mutaties plaatsvinden op de data. Alle berichten die mogelijk in een queue staan in de K2B omgeving om verstuurd te worden naar DDS moeten ook afgehandeld worden voordat de kloon gemaakt kan worden.

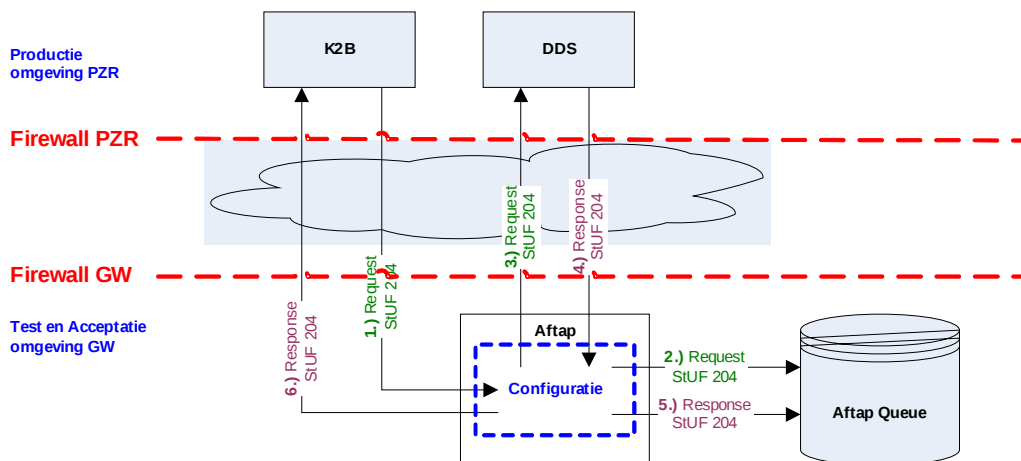
Test, Acceptatie en
Productie omgeving
PZR



Figuur 3 – Maken kloon K2B database

Deze stap wordt bij PZR uitgevoerd en moet intern getest worden op een correcte werking.

4 Aftappen van de berichtenuitwisseling tussen K2B en DDS



Figuur 4 - Aftappen berichtuitwisseling K2B - DDS

In bovenstaande figuur wordt het aftapmechanisme voor de berichtenuitwisseling tussen Key2Burgerzaken en DDS weergegeven. Door de berichtenuitwisseling af te tappen (stap 2 in de figuur) van de productie omgeving vanaf tijdstip "t" worden vanaf tijdstip "t" alle door Key2Burgerzaken verzonden StUF2.04 berichten naar DDS verwerkt in DDS en tegelijkertijd opgeslagen in de queue van de aftap. Het doel is om de berichten die in de queue zijn opgeslagen uiteindelijk in te lezen in het Gegevensmagazijn. Deze inleesactie zal gebeuren nadat de Gegevensmagazijn initieel gevuld is (zie latere stappen).

4.1 Aftap proxy tests

Het functioneel gedrag van de aftap proxy is als volgt:

- Het aftappen moet transparant zijn. De proxy mag niets wijzigen aan de inkomende of uitgaande berichten om de communicatie tussen K2B en DDS niet te verstoren.
- De aftap proxy moet de inkomende berichten vanuit K2B en de ontvangen bevestigingen van DDS verbatim opslaan in de queue.
- De aftap proxy moet goed omgaan met foutsituaties (zoals een fout die gerapporteerd wordt door de DDS omgeving).

Ten opzichte van de huidige situatie (zie Figuur 1) zijn de aftap proxy en de bijbehorende database toegevoegd. De werking van de aftap proxy in deze stap is afhankelijk van de volgende punten:

- De firewall van PZR moet zodanig geconfigureerd zijn dat de aftap proxy bereikbaar is vanaf de K2B omgeving.
- De firewall van GW en PZR moet zodanig geconfigureerd zijn dat DDS bereikbaar is vanaf de omgeving van de aftap proxy.



- ~~Requests vanuit de K2B omgeving moeten naar de URL van de aftap proxy worden gestuurd in plaats van rechtstreeks naar DDS.~~
- Correcte configuratie van de database waar de queue wordt bijgehouden en de configuratie van de aftap proxy in het algemeen.

Deze afhankelijkheden worden getest door het controleren van de werking van de proxy. De tests verschillen per omgeving (Test, Acceptatie en Productie).

4.1.1 Test & Acceptatie

In de test en acceptatie omgeving tests we de aftap proxy op twee manieren:

- Functionele tests met Fitnesse
- Handmatige tests

Deze testwijzen staan hierna beschreven.

4.1.1.1 Functionele tests met Fitnesse

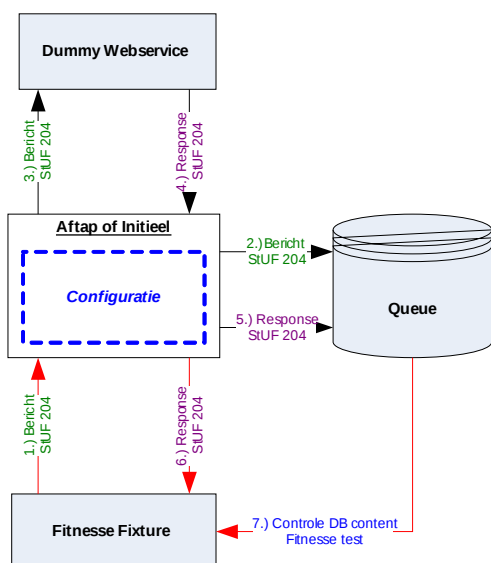
De aftap proxy zal in de test en acceptatie omgeving functioneel getest worden middels het uitvoeren van geautomatiseerde testgevallen in Fitnesse. De tests hebben als scope:

- Normaal gedrag van de aftap proxy bij gebruik tussen een K2B en een DDS omgeving (zowel de K2B als de DDS omgeving zijn gesimuleerd).
- Gedrag van de aftap proxy bij een exception binnen de webservice.
- Gedrag van de aftap proxy bij het aanroepen van een method die niet wordt aangeboden door de webservice.
- Gedrag van de aftap proxy bij het ontvangen en verwerken van een bericht dat diakrieten bevat.
- Gedrag van de aftap proxy bij grote berichten.

Voor het uitvoeren van deze tests is het volgende nodig:

- Een K2B omgeving om berichten te versturen. Deze rol wordt door de Fitnesse Fixtures ingevuld. Deze zullen berichten opsturen naar de te testen aftap proxy.
- Een dummy web service die gebruikt wordt om bepaalde situaties te simuleren (bijvoorbeeld: een exception die in de webservice optreedt). De webservice simuleert ook een DDS omgeving die ontvangen berichten bevestigt middels een response volgens het StUF 204 formaat. Dit is een custom webservice die in een Mule instance wordt gehost.

In Figuur 5 is te zien hoe de test is opgezet. De te testen onderdelen in deze opzet zijn de aftap (of Initieel) proxy en de database waarin de queue wordt bijgehouden.



Figuur 5 - Aftap proxy tests in Fitnesse

De fixtures hebben de volgende opzet. Achter de punten staat het nummer uit de figuur waarin de actie wordt uitgevoerd:

1. Stuur een bericht naar de aftap proxy (1).
2. Controleer dat de verstuurde request in de fixture gelijk is aan de request zoals de aftap proxy deze heeft opgeslagen in de queue (na 6 en 7, binnen Fixture).
3. Controleer dat de ontvangen response gelijk is aan de verwachte response in de fixture (na 6, binnen Fixture).
4. Controleer dat de ontvangen response gelijk is aan de response zoals de aftap proxy deze heeft opgeslagen in de queue (7).

4.1.1.2 Handmatige tests

De aftap proxy zal in test en acceptatie ook handmatig getest worden. De test is van belang om te controleren dat de koppeling met de testomgeving in orde is. Functioneel is de aftap proxy in de vorige stap getest middels Fitnesse fixtures. Voor de handmatige tests moet de aftap proxy opgezet worden zoals in Figuur 4. Hierna zal men vanaf de K2B omgeving een kennisgeving uit doen gaan. Zoals in Figuur 4 te zien is, zal de kennisgeving via de aftap proxy bij DDS aankomen. De response van DDS gaat via diezelfde aftap proxy weer terug naar K2B. Door handmatig de database van de aftap proxy te bekijken kan gecontroleerd worden dat de aftap proxy goed werkt. In de request tabel moet een bericht te zien zijn die te maken heeft met de kennisgeving die verstuurd is vanuit K2B. In de response tabel moet een bericht te zien zijn die afkomstig is van DDS en die een bevestiging bevat van de request.

4.1.2 Productie

Als de aftap proxy in productie getest moet worden willen we de staat van de productie database op geen enkele wijze beïnvloeden.

Fitnesse tests zijn niet geschikt omdat deze tests de database beïnvloeden: aan het begin van de test wordt de database opgeschoond en de sequence waarden gaan omhoog tijdens de test. Aan het eind van de tests blijven er ook een aantal records achter.



De handmatige tests uit paragraaf 4.1.1.2 zijn beter geschikt in deze omgeving. De database mag aan het eind van deze handmatige tests echter geen testdata bevatten. De volgende acties zijn nodig om in productie de koppeling met de aftap proxy te testen en de database in een correcte staat terug te brengen na de test:

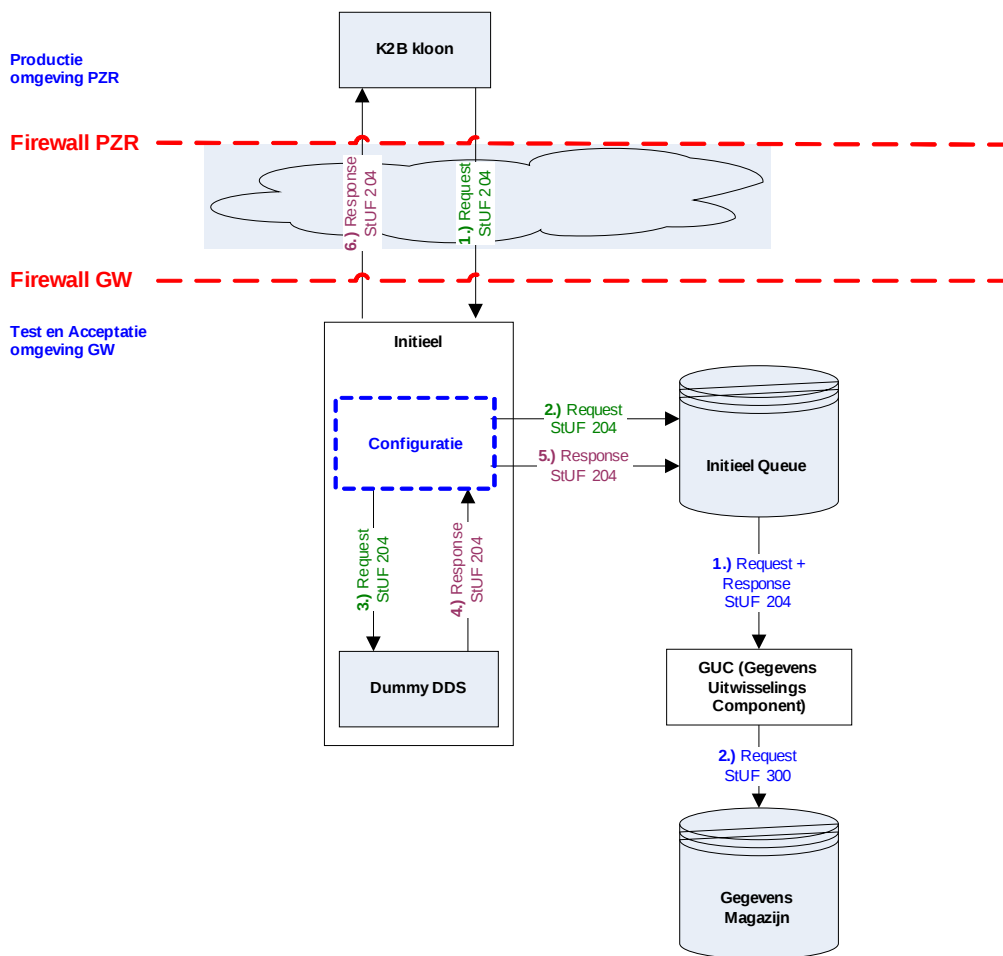
1. Voer in K2B een actie uit die een kennisgeving naar DDS tot gevolg zal hebben (bijv. nieuwe burger)
2. Als het bericht naar DDS is verstuurd voert men een correctie uit waarbij de actie gecorrigeerd wordt (bijv. verwijder burger). Dit bericht heeft als doel dat de K2B en DDS omgevingen in dezelfde staat teruggebracht worden als deze was voor stap 1 (afgezien van eventuele historie en auditing op die 2 systemen).
3. De K2B omgeving moet een succesvolle communicatie met DDS rapporteren.
4. Controleer dat de verwachte requests en responses aanwezig zijn in de database van de aftap proxy.
5. Verwijder alleen deze requests en responses uit de database van de aftap proxy.

Hiermee is de aftap proxy op productie getest zonder de database te beïnvloeden. Het enige spoor van deze tests is de sequence in de aftap proxy database die opgehoogd is door de berichtwisseling.

Als de koppeling met de proxy niet goed werkt verliest K2B geen data omdat deze niet aangekomen kennisgevingen nogmaals zal versturen totdat DDS een response terugstuurt.

5 Gegevensmagazijn initieel vullen aan de hand van de K2B kloon

Het doel van deze stap is om de gegevens uit de K2B kloon af te tappen en de afgetapte gegevens te importeren in het Gegevensmagazijn. In deze stap gebruiken we de in hoofdstuk 3 gemaakte kopie van de database van K2B. Een tweede K2B omgeving (in Figuur 6 als K2B kloon aangeduid) moet worden opgezet rond deze database om de berichten naar de initieel proxy te sturen. De situatie is afgebeeld in Figuur 6.



Figuur 6 - Initieel vullen via K2B kloon

De initieel queue die in Figuur 6 is afgebeeld kan ten opzichte van de aftap proxy (zie hoofdstuk 4) in dezelfde schema in de database staan. De initieel proxy gebruikt andere tabellen dan de aftap proxy, het is dus mogelijk dat de initieel proxy en de aftap proxy in dezelfde schema hun queues bijhouden.

De dummy DDS, die in Figuur 6 te zien is, stuurt dezelfde responses terug die de echte DDS zou terugsturen. De dummy DDS is onderdeel van de initieel proxy en kan desgewenst uitgezet worden in de



~~configuratie om de proxy gebruik te laten maken van een DDS kloon omgeving. Het gebruik van de dummy DDS brengt de volgende voordelen met zich mee:~~

- K2B heeft enkel een response nodig waarin staat dat het bericht ontvangen is. Een echte DDS omgeving zou de data verwerken en de response terugsturen. De verwerkingstijd is in dit proces (initieel) verloren tijd omdat de verwerking van de gegevens in DDS niet van belang is. De dummy DDS in Figuur 6 verwerkt geen data maar genereert direct een response waardoor de verwerking van de initieel data veel sneller kan verlopen.
- Als een DDS kloon omgeving niet nodig is bespaart dit resources aan de kant van PZR en tijd in het algemeen.
- Er hoeven geen poorten geopend te worden naar een DDS kloon omgeving, de dummy DDS draait namelijk binnen de Mule instance.

Hierna beschrijven we hoe we dit onderdeel in de test, acceptatie en productie omgeving zullen testen.

5.1 Initieel vullen tests

De werking van de initieel proxy in deze stap is afhankelijk van de volgende punten:

- De firewall van PZR moet zodanig geconfigureerd zijn dat de initieel proxy bereikbaar is vanaf de K2B kloon omgeving.
- Requests vanuit de K2B omgeving moeten naar de URL van de aftap proxy worden gestuurd in plaats van rechtstreeks naar DDS.
- Correcte configuratie van de database waar de queue wordt bijgehouden en de configuratie van de aftap proxy in het algemeen.
- Correcte configuratie van de GUC component.

Deze afhankelijkheden worden getest door het controleren van de werking van de proxy en de GUC component. De tests verschillen per omgeving (Test, Acceptatie en Productie).

De te testen onderdelen zijn:

- De initieel proxy (inclusief dummy DDS)
- De database waarin de initieel queue wordt bijgehouden
- De GUC component voor import van gegevens uit de initieel queue naar het Gegevensmagazijn
- Het Gegevensmagazijn

5.1.1 Test & acceptatie

Op de test en acceptatie omgeving wordt op de volgende wijze getest:



- Middelste Fitnessse tests

- Door handmatig te testen

5.1.1.1 Fitnessse tests

De Fitnessse tests beschouwen het systeem als bestaande uit twee onderdelen:

- De initieel proxy bestaande uit de proxy, dummy DDS en de database.
- De GUC component bestaande uit de component zelf en het Gegevensmagazijn

De scheiding tussen deze systemen ligt bij de database van de initieel proxy waar de queue wordt bijgehouden. De initieel proxy schrijft daarin data weg die asynchroon verwerkt wordt door de GUC component.

De initieel proxy wordt in de Fitnessse tests op dezelfde wijze getest als dat bij de aftap proxy is gedaan (zie paragraaf 4.1.1.1). Het functioneel gedrag is namelijk identiek aan de aftap proxy. De dummy DDS die onderdeel uitmaakt van de initieel proxy wordt ook getest in de Fitnessse tests.

De GUC component wordt getest in een eigen test suite (DVGM test suite). Deze test suite maakt geen gebruik van de initieel proxy en de queue. De Fitnessse fixtures gebruiken de file in service van de GUC component om de data aan te leveren aan de GUC component. Daarbij wordt door de test een bestand met een StUF 2.04 bericht geplaatst in een bepaalde directory waarna de GUC component de data oppakt en deze verwerkt in het Gegevensmagazijn. De fixture zal controleren dat de gegevens correct zijn geïmporteerd door de GUC component.

Als we ervan uit gaan dat het systeem correcte berichten krijgt (StUF 2.04 berichten al dan niet in een SOAP envelope) kan het systeem de volgende soorten berichten ontvangen:

- Toevoeging
- Wijziging
- Verwijdering
- Correctie

Het DVGM testsuite in Fitnessse heeft als doel om per entiteit in het domein (persoon, adres ...) deze 4 berichtsoorten te controleren op een correcte verwerking.

Verder wordt het GUC component getest op het gedrag bij de volgende situaties:

- Het bericht is geen XML bericht



-
- Het bericht is niet een ongeldig XML bericht (bevat fouten)
 - Het bericht is een correct XML bericht maar is geen StUF 2.04 bericht
 - Het bericht is een geldig StUF 2.04 bericht maar bevat geen geldige data

5.1.1.2 Handmatige tests

De handmatige tests zijn bedoeld om het hele proces te testen en om te controleren dat de configuratie van de initieel proxy en de GUC component in orde is. De tests hebben dezelfde opzet als de handmatige tests die voor de aftap proxy zijn gedaan (zie paragraaf 4.1.1.2). Het verschil met die tests is als volgt:

- In plaats van de K2B omgeving wordt de K2B kloon omgeving gebruik om de request te versturen.
- De handmatige controle bestaat ook uit het bekijken van de database van het Gegevensmagazijn om vast te stellen dat het proces van begin tot eind correct verloopt.
- K2B kloon moet goed kunnen omgaan met de responses van de initieel proxy. Deze responses zijn gegenereerd door de dummy DDS. Bij een correcte werking zal K2B de responses op dezelfde wijze verwerken als waren ze afkomstig uit een rechtstreekse verbinding met de echte DDS omgeving. K2B zal hierover feedback geven.

5.1.2 Productie

Omdat de K2B kloon geen raakvlakken heeft met de productie K2B omgeving en de productie DDS omgeving kan op de productie omgeving van de K2B kloon op een iets andere wijze getest worden. We zijn nog steeds beperkt tot handmatige tests omdat we niets willen wijzigen aan de productie database. We kunnen echter direct werken met productie data.

Op de K2B kloon omgeving moet een actie uitgevoerd worden waardoor er 1 bericht naar de initieel proxy wordt verstuurd. Hierna zullen we op dezelfde wijze als bij de handmatige tests in test en acceptatie (zie paragraaf 5.1.1.2) de werking van het initieel vullen als geheel testen.

Na de test hoeven we geen data te verwijderen. Bij een geslaagde test kan aan K2B kloon de opdracht worden gegeven om alle data op te sturen naar de initieel proxy. K2B heeft eerder geregistreerd welk bericht al is verwerkt door de initieel proxy tijdens de test zal het bericht niet nogmaals verzenden.

Na het geven van de opdracht aan K2B om alle berichten te versturen volstaat het om de log van de GUC component periodiek in de gaten te houden om fouten tijdig te kunnen detecteren. Het gedrag van het systeem is ook te zien aan de queue van de initieel proxy. Bij een correcte verwerking van het bericht wordt in de queue aangegeven wat de tijd was van de verwerking. Door een query uit te voeren op de request tabel, die telt hoeveel berichten een dequeue_timestamp anders dan 'null' hebben staan, kan bepaald worden of er requests verwerkt worden.



6 Koppeling van de GUC component aan de aftap queue

Als de K2B kloon omgeving (zie Figuur 6) alle berichten heeft opgestuurd naar de initieel proxy en als daarna de GUC component alle records uit de initieel queue heeft geïmporteerd in het Gegevensmagazijn maken wij een backup van de initieel queue database. De database bevat namelijk een volledig snapshot van de data zoals deze aanwezig was in K2B op tijdstip "t".

In deze situatie kunnen we de GUC component veilig ontkoppelen van de initieel queue en koppelen aan de aftap queue (zie Figuur 4). Wanneer dat is gebeurd krijgen we de situatie zoals deze eerder in Figuur 2 - Gewenste werking koppeling K2B – DDS is weergegeven.

Door de configuratie van de GUC component te wijzigen, koppelen we deze aan de aftap queue en worden de berichten uit die queue ingelezen en verwerkt in het Gegevensmagazijn. Na de wijziging van de configuratie moet de Mule instance waarin de GUC component draait worden herstart.

Om te controleren dat alles naar behoren werkt doen we dezelfde handmatige controle zoals deze gedaan is bij de in paragraaf 5.1.1.2.