Student: Lyva Van Damme

Stagebegeleider: Koen Verhoeyen

Stagebedrijf: Delaware

Stagementor: Nicolas Pauwelyn

Academiejaar 2018-2019

Stageverslag

Bachelor toegepaste informatica

Enterprise messaging

Enterprise messaging toe gepast op een business case

Stageverslag 0

Bachelor toegepaste informatica 0

Enterprise messaging 0

Enterprise messaging toe gepast op een business case 0

Voorwoord 2

Delaware 3

Historiek 3

Geografische ligging 3

Situering stage binnen Delaware 3

Stageopdracht: Partner notification framework on SAP S/4 HANA Cloud using SAP Cloud Platform Enterprise Messaging Queueing system on SAP Cloud Platform Suite with SAP Cloud Platform Integration with graphic user interface made with Hypertext markup language and Hypertext preprocessor. 5

Beginsituatie 5

Doel van de opdracht 6

Uitwerking 6

Eindresultaat 8

Business doelstellingen 15

Persoonlijke doelstellingen 15

Verklarende woordenlijst 17

Bibliographies 18

Voorwoord

Dit stageverslag werd geschreven tijdens mijn stagetraject bij Delaware.

Ik zou graag Delaware bedanken voor de kansen die ik kreeg binnen het bedrijf en het team. Het was een leerrijke ervaring. Het was een zeer interessante opdracht. Het had uitdagingen.

Ik wil graag mijn ouders bedanken om mij bij te staan bij deze nieuwe ervaring.

Ook wil ik graag mijn broer en zus bedanken om mij te steunen wanneer nodig en hun positieve ingesteldheid.

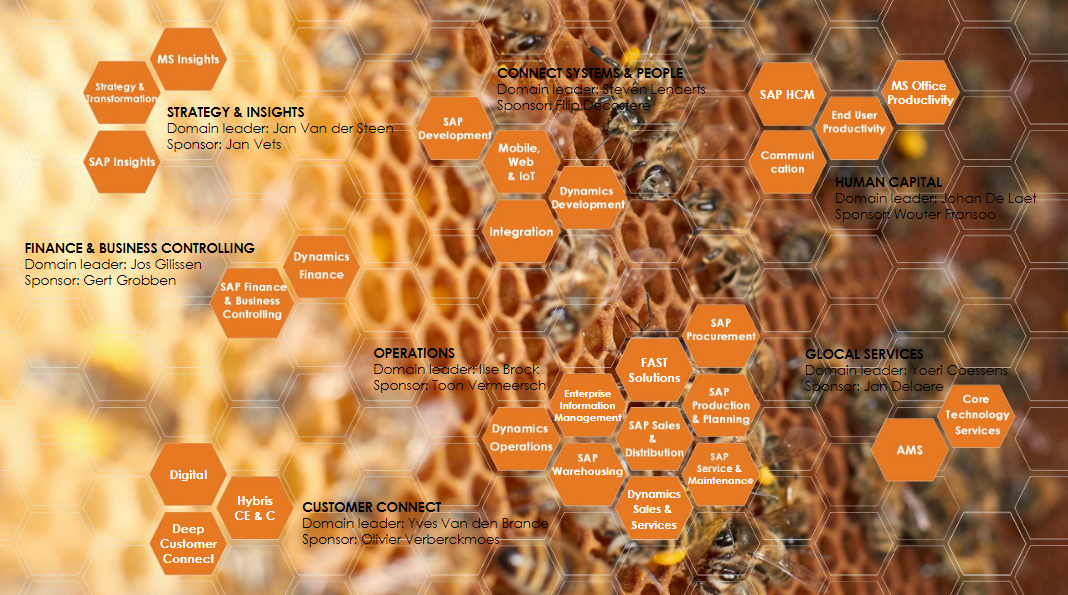
Delaware

Historiek

Geografische ligging

Delaware heeft vestigingen en Europa, Amerika en Azië. In het totaal zijn er 13 landen waarin ze gevestigd zijn. Er zijn 24 kantoren over alle continenten. Er zijn kantoren gevestigd in de Verenigde Staten van Amerika. In Azië zijn ze gevestigd in China. De meeste kantoren bevinden zich in Europa. In België hebben ze er vier: Gent, Antwerpen, Kortrijk en Waver. Er zijn ook kantoren te vinden Nederland, Groothertogdom Luxemburg en Frankrijk.

Situering stage binnen Delaware



Hierop is de structuur van Delaware te zien. Delaware koos ervoor om hun bedrijf zo voor te stellen. Hiermee proberen ze aan te tonen dat de ene afdeling niet beter is dan de ander.



Ik doe mijn stage binnen het SAP-integratie team. Dit is gesitueerd binnenin de solution Integration. Wat valt onder het domein Connect Systems & People.

Ik koos voor dit bedrijf als stageplaats omdat in eerste instantie de stage-opdracht mij aansprak. Daarna ben ik ook gaan polsen bij medestudenten die al stage gedaan hadden. En ik kreeg veel positieve reacties op de sfeer binnen Delaware. Ook dit sprak mij aan. Na een gesprek, met iemand binnen Delaware, over mijn stage, was ik helemaal overtuigd dat dit bedrijf mij zeker goed zou liggen.

Stageopdracht: Partner notification framework on SAP S/4 HANA Cloud using SAP Cloud Platform Enterprise Messaging Queueing system on SAP Cloud Platform Suite with SAP Cloud Platform Integration with graphic user interface made with Hypertext markup language and Hypertext preprocessor.

Beginsituatie

Enterprise Messaging is een nieuwe technologie binnen SAP Cloud Platform. Er was een business case uitgeschreven waarop deze technologie moest worden toegepast. Er was nog niets opgezegd. Er was wel wat documentatie over Enterprise Messaging zelf. Die was te vinden op de wiki pagina van Delaware.

Doel van de opdracht

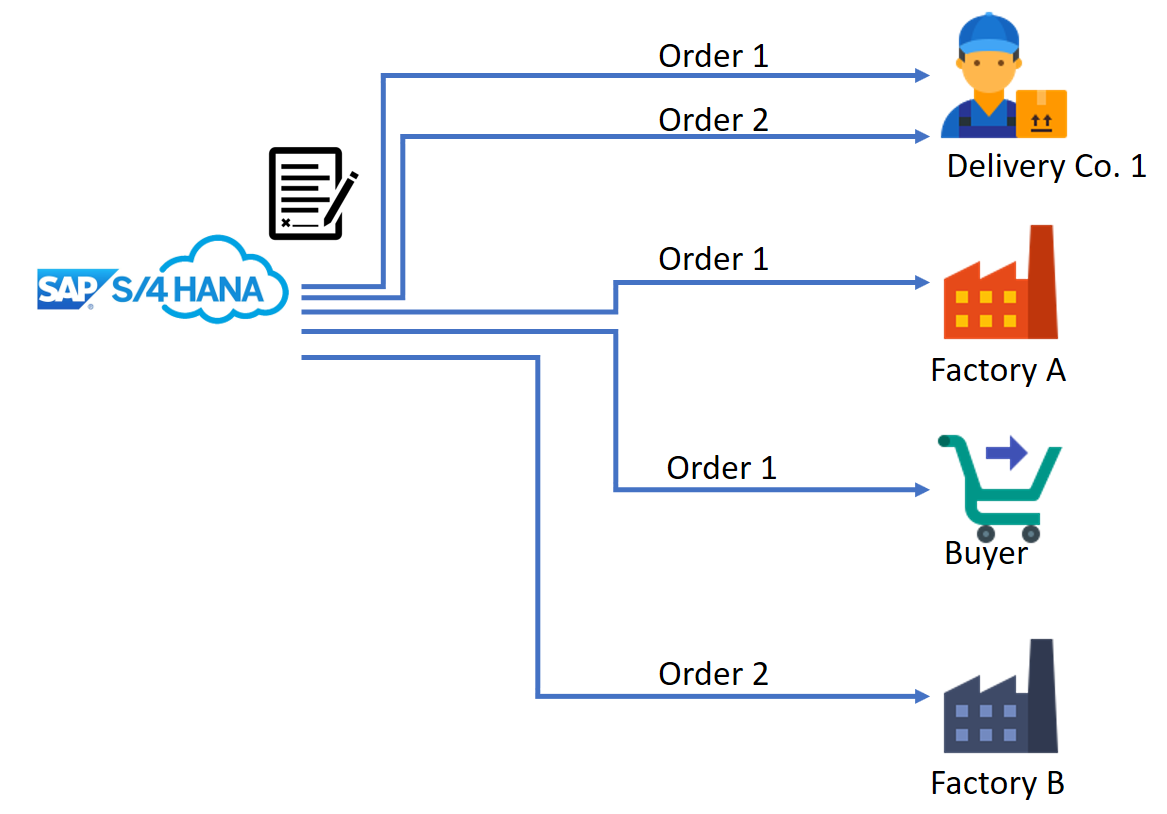
Het doel van deze opdracht is om een demo op te zetten die volledig werkt. Dit is de opdracht die ze mij hebben doorgestuurd bij de aanvang van mijn stage.

Een productiebedrijf ontvangt verkooporders die dienen te worden opgevolgd door 3e partijen. Deze partners kunnen zich via een API/ Webpagina inschrijven om hun verkooporder op te volgen.

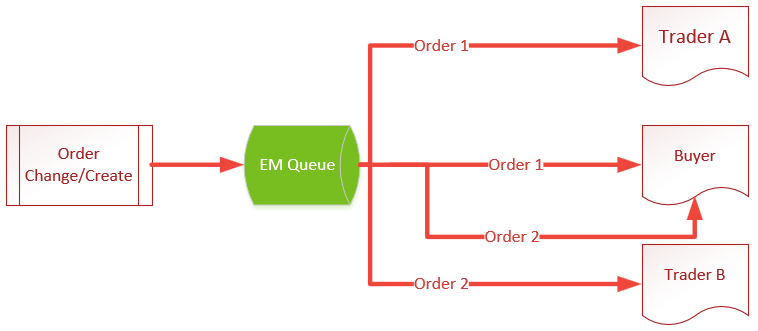
Elke aanpassing aan een verkooporder wordt doorgegeven vanuit SAP S/4H naar de partners via een event, dat naar een Message Queue gaat. De bedoeling is dat elke partner zich kan registeren op 1 of meerdere queues. Het aanmaken, opvullen & verwijderen van een verkooporder queue dient dynamisch te gebeuren, zonder dat er manuele interactie nodig is.

De opgezette oplossing moet robuust, dynamisch en herbruikbaar zijn voor andere projecten (vb. productieorders, facturatie, etc.).

Het proces is als volgt:



Een al kleine architecturale uitwerking hieronder:



Uitwerking

Eerst moest ik leren werken met CPI. Het SAP Cloud Platform Integration wordt gebruikt om Flow te maken. Daarnaast maak ik ook gebruik van S/4 HANA, bij een verandering of creatie van een order, wordt er een proces getriggerd. En als laatste maak ik gebruik van SAP Cloud Enterprise Messaging.

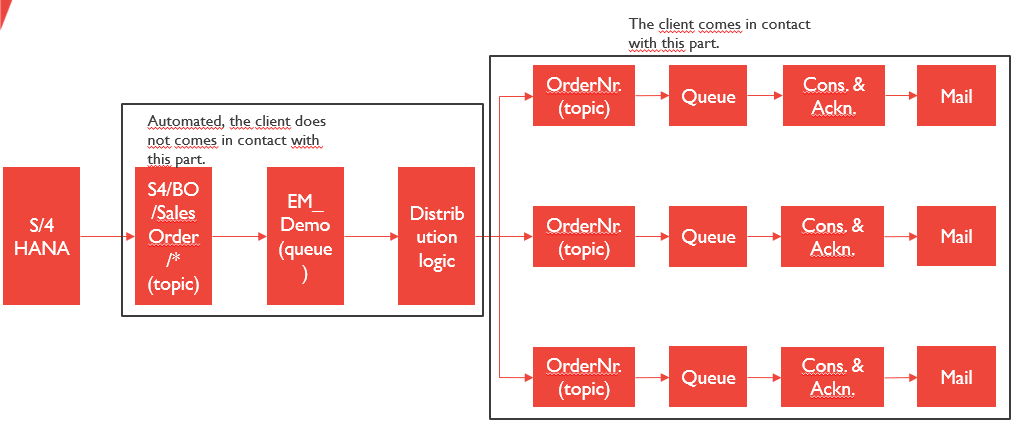
**De volledige uitwerking van het proces**

De klant meldt zich aan voor de registratie. Er moet een emailadres en bedrijfsnaam meegegeven worden. Met die data kan er dan een queue gemaakt worden. De queue is een onderdeel van Enterprise Messaging.

Eens de queue is aangemaakt, kan de klant op zijn orders gaan subscriben. Een order wordt voorgesteld als een topic. Een topic is een verdeler van berichten. Op een topic wordt een bericht gepost. Dat topic stuurt dan het bericht door naar alle queues die luisteren naar hem. Dit alles gebeurt via queue subscription

Gebeurt er een verandering in S/4HANA, dan triggert dit een heel proces. Bijvoorbeeld: verandert er iets aan salesorder 55 dan wordt die verandering meteen gestuurd naar een algemeen topic, S4/BO/SalesOrder/\*. De queue EM\_Demo luistert naar dat topic. Om de 10 minuten wordt er een proces gestart dat alle berichten van op EM\_Demo ophaalt. Er wordt naar elk bericht apart gekeken, het ordernummer wordt bekeken en opgeslaan. Het bericht wordt dan verder doorgestuurd naar een topic met het ordernummer. Die stuurt dan het bericht door naar de gesubscribde queues. Het bericht dat op de queue staat, wordt opgehaald en doorgemaild naar de klant.

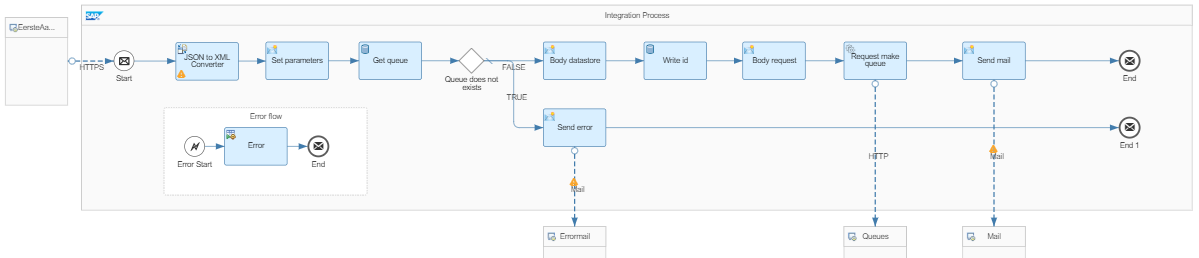
**Een beter beeld van de architectuur**



Eindresultaat

# De flows in CPI

## Een queue aanmaken

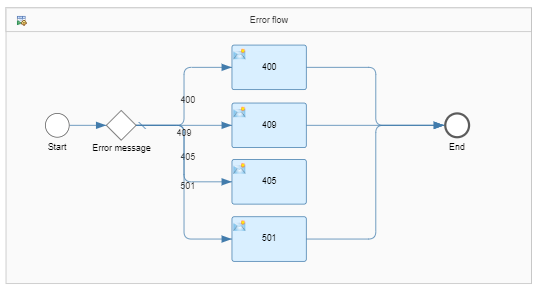


Als eerste komt er een request binnen met volgende elementen in de body:

* Een emailadres,
* De naam van de queue.

De body wordt in een JSON-format doorgestuurd. Maar CPI werkt enkel met XML-format. Er moet dus eerst omgevormd worden. Dan wordt er gecontroleerd of die queue al niet bestaat in het systeem. Dit gebeurt om te voorkomen dat er gedupliceerde namen in het systeem zouden zitten. Indien de naam al eens voorkomt, wordt er een mail gestuurd met de mededeling dat de queue naam al bestaat en ze een andere naam moeten kiezen.

Bestaat de naam nog niet? Dan worden de gegevens in de body gezet en weggeschreven naar de databank. Dan wordt er een request gemaakt om die queue aan te maken. Eens dit gelukt is, wordt er een e-mail verzonden met de confirmatie van de aangemaakte queue.



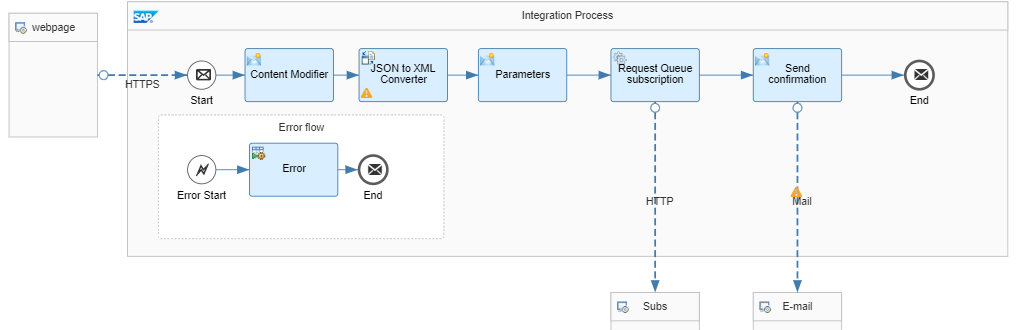
Er is ook een error flow aanwezig in het proces, die is verder uitgewerkt in een sub proces.

Zoals te zien is, kunnen er vier soorten foutmeldingen zijn. Een fout 400 wil zeggen dat er het request slecht was. Er werd niet voldaan aan de verplichte parameters.

Een fout 409 komt voor wanneer er een conflict gebeurt. Zoals dubbele namen voor de queue.

Een fout 501 wil zeggen dat de manier van de request sturen niet de correcte is en dus niet ondersteund wordt.

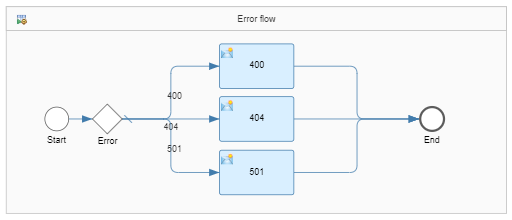
## Een queue subscription aanmaken



Er komt een request binnen met volgende elementen in de body:

* Emailadres,
* Queue naam,
* Ordernummer.

De content modifier zorgt ervoor dat er een root element rond de inhoud komt te staan. Anders loopt het proces vast. Dan wordt het bericht opnieuw van format omgevormd. De parameters die nodig zijn om de request te maken, worden opgevuld. Daarna wordt de request gemaakt om de subscription te maken. Hierna wordt er ook een confirmatie mail gestuurd naar de klant.

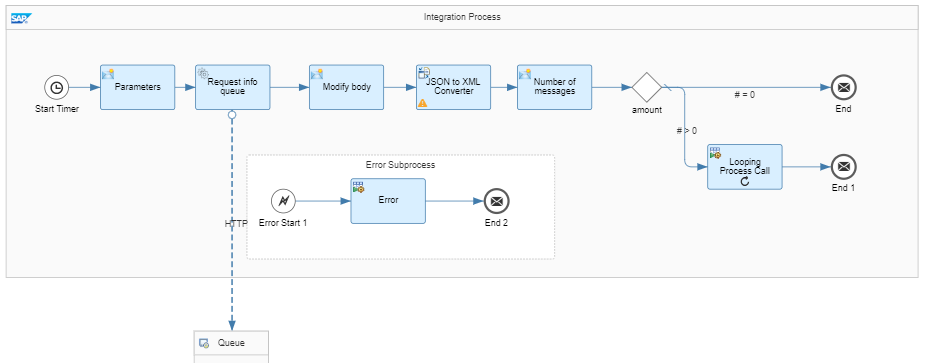


Een fout 400 is ook hier een slechte request.

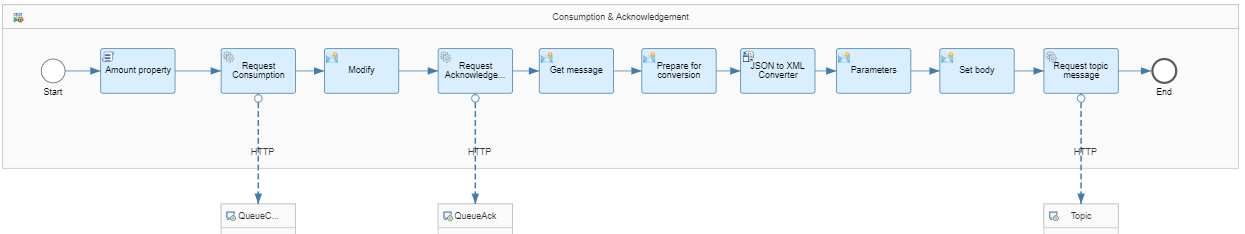
Bij een 404 fout werd de queue niet gevonden. Er kan een schrijffout opgetreden zijn of de queue moet eerst nog aangemaakt worden.

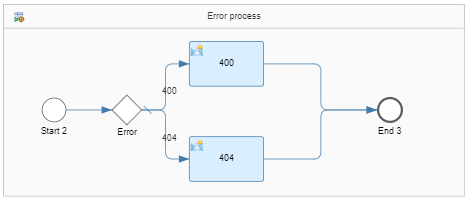
Bij een 501 fout is de manier van de request niet ondersteund.

## Ophalen van berichten van de queue EM\_Demo



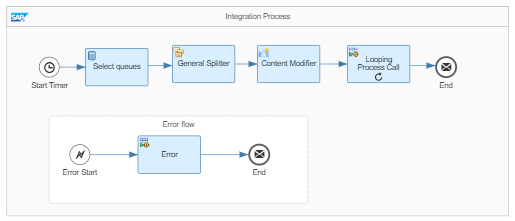
Elke 10 minuten wordt dit proces aangeroepen. De parameters van de queue worden ingevuld. Dit proces is statisch en niet dynamisch omdat het proces gefikst is op één queue. Er wordt info van die queue opgevraagd. De request geeft een JSON-format body terug, die ook omgevormd wordt naar XML. Het aantal berichten wordt opgehaald. Zijn er geen berichten op de queue dan eindigt het proces. Staan er berichten op de queue, dan start er een proces dat zich blijft herhalen tot wanneer alle berichten van de queue zijn gehaald. Het loop proces is te zien op onderstaande foto.

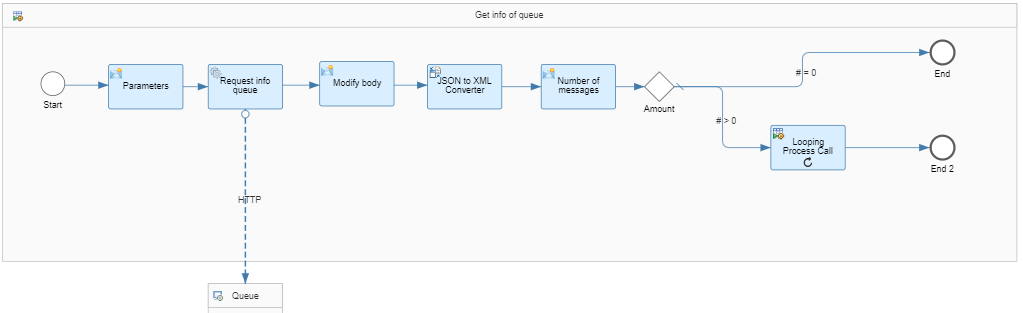


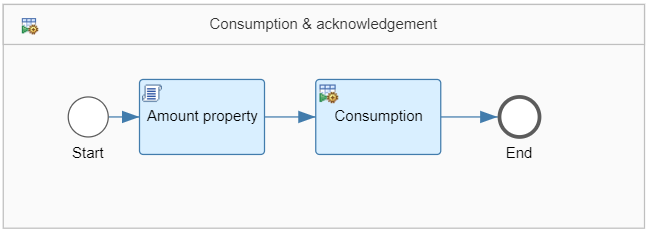
Het aantal berichten is opgeslaan in een variabele. Aan het begin van het proces, wordt dat aantal altijd vermindert met één. Zodat wanneer deze op nul komt te staan, het proces ook stopt. Anders gaat het proces in fout en faalt het. Nadat het aantal verminderd is, wordt het eerste bericht opgehaald en ook meteen verwijdert van de lijst. Zodat het effectieve aantal ook klopt met de variabele. De inhoud van het bericht wordt opgehaald en ook weer omgezet naar XML. Dan wordt het ordernummer opgehaald. Het ordernummer is de naam van het order specifieke topic. Het bericht wordt dan gepusht naar dat topic. Bij een fout dat 400 teruggeeft, is er sprake van een bad request. Bij een fout dat 404 teruggeeft, is de queue niet gevonden.

## Ophalen van berichten van queues van bedrijven

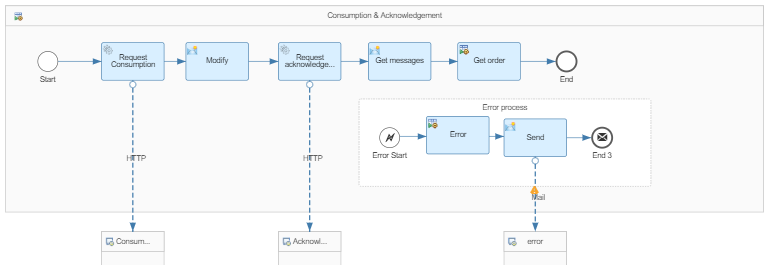
Dit proces wordt elke 10 minuten gerund. Dit proces lijkt heel hard op dat van ‘Ophalen van berichten van EM\_Demo’.

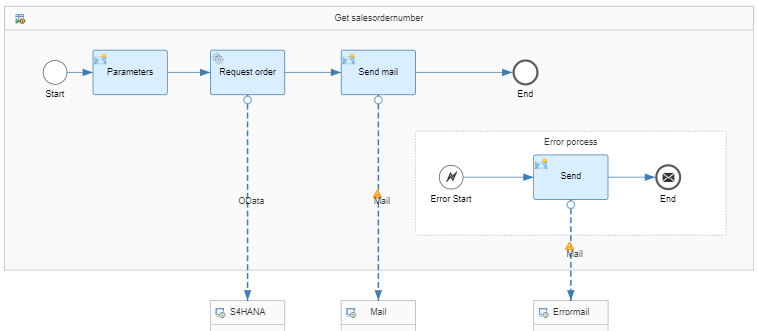
In dit proces worden eerst alle queue’s opgehaald. Dit wordt doorgestuurd als een XML-format. De hele datastore wordt teruggegeven in XML-format. Die moeten dan opnieuw opgesplitst worden. Dit gebeurt via de ‘General Splitter’. De content modifier wijst dan een queue naam en een emailadres toe aan variabelen. Die worden dan in een loop gestoken. Voor elke rij in de datastore wordt de loop toegepast.



Als eerste wordt er een header toegevoegd die noodzakelijk is voor de request. Dan wordt er info van die queue opgehaald. Het antwoord moet worden opgehaald. Het antwoord wordt teruggegeven in een JSON-format en moet dus omgevormd worden naar een XML-format. Het aantal berichten dat op de queue staat, wordt bekeken. Staan er geen berichten op de queue dan wordt de loop beëindigt voor deze queue. Staat er een of meerdere berichten op de queue dan wordt er een andere loop aangeroepen. Ook hierbij, zijn alle berichten van de queue gehaald, stopt de loop voor deze queue.

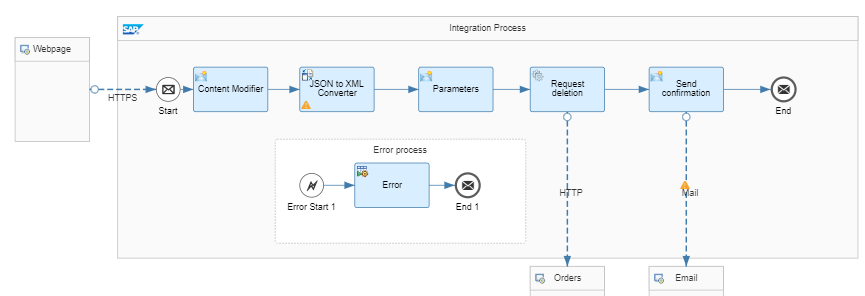
De variabele ‘amount’ wordt verminderd met één nog voordat het bericht van de queue wordt gehaald. Zodat wanneer ‘amount’ op nul staat, de eerste loop wordt beëindigd voor die queue.

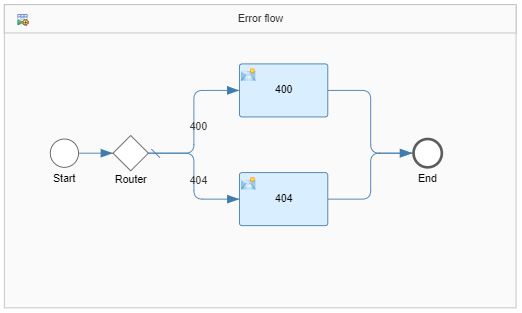


Als eerste wordt het bericht op de queue gelezen en de id van het bericht wordt teruggegeven. Het bericht wordt opgeslaan in een variabele. Het bericht wordt dan van de queue verwijdert. Het bericht dat op de queue stond wordt uit de variabele gehaald. Dan wordt er een subproces aangeroepen om de gegevens van het order op te halen.

Het proces om de gegevens van het order op te halen, gaat als volgt. De parameters worden ingesteld met het ordernummer. Dan wordt er een request gemaakt om van dat specifiek order de gegevens op te halen. De gegevens worden meteen via mail doorgestuurd naar de klant. De gegevens zijn te vinden in de bijlage die wordt meegestuurd.

## Het verwijderen van een queue subscription

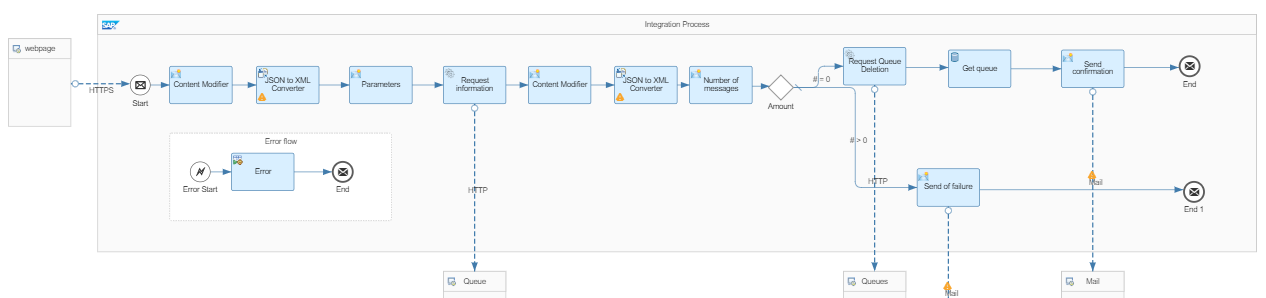


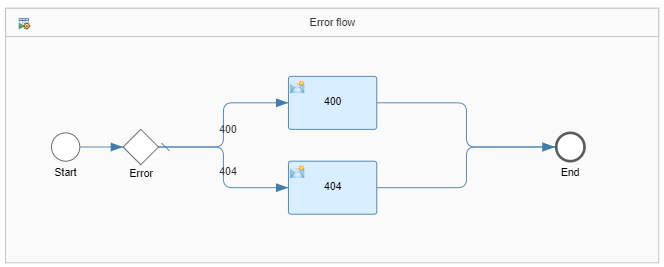
Er komt een request binnen met de queue naam en ordernummer. De ‘content modifier’ plaatst een root element rond de JSON body. Dan wordt het JSON bericht omgezet naar een XML bericht. De parameters worden dan opgevuld zodat die dynamisch gebruikt kunnen worden. Na het verwijderen van de subscription wordt er een email verzonden.

Bij een fout 400 was de request slecht. Bij een 404 werd de queue niet gevonden.

Loopt er iets fout bij de request, dan wordt de mail niet verstuurd.

## Het verwijderen van een queue



Een request laat die proces van start gaan. Volgende gegevens worden verwacht: emailadres en queue naam. Er wordt eerst een root element rond de ingestuurde body geplaatst. Dan wordt de body van JSON naar XML omgezet. Eerst wordt er informatie opgehaald van die queue. Het antwoord wordt omgezet naar XML. Het aantal berichten wordt opgehaald. Staan er nog berichten op queue dan kan deze niet verwijdert worden. Er moet gewacht worden tot de volgende keer dat het proces van consumption en acknowledgement gerund heeft. Staan er geen berichten meer op de queue. Dan wordt de queue verwijdert. Ook wordt de queue verwijdert uit de datastore. Zodat bij de volgende consumption en acknowledgement geen fout kan gebeuren omdat er een queue niet gevonden werd.

Een fout 400 komt er wanneer er een slechte request is gemaakt. Een fout 404 is als de queue niet gevonden wordt.

# De trigger in S/4 HANA en de verdeling van de berichten

Bij een verandering aan een salesorder wordt dit als bericht gepost op de topic S4/BO/SalesOrder/\*. De queue EM\_Demo luistert als enige queue naar die topic. Hierop wordt het proces ‘Ophalen van berichten van de queue EM\_Demo’ toegepast.

Om ervoor te zorgen dat de verandering of creatie werd doorgestuurd. Moest er een communication arrangement worden gemaakt. Volgend stappenplan werd gevolgd en werkte, (Panda, 2018). Na de eerste keer kwamen er fouten naar boven bij de trigger. Het probleem kon ik niet alleen oplossen en ik kreeg dus hulp van collega’s.

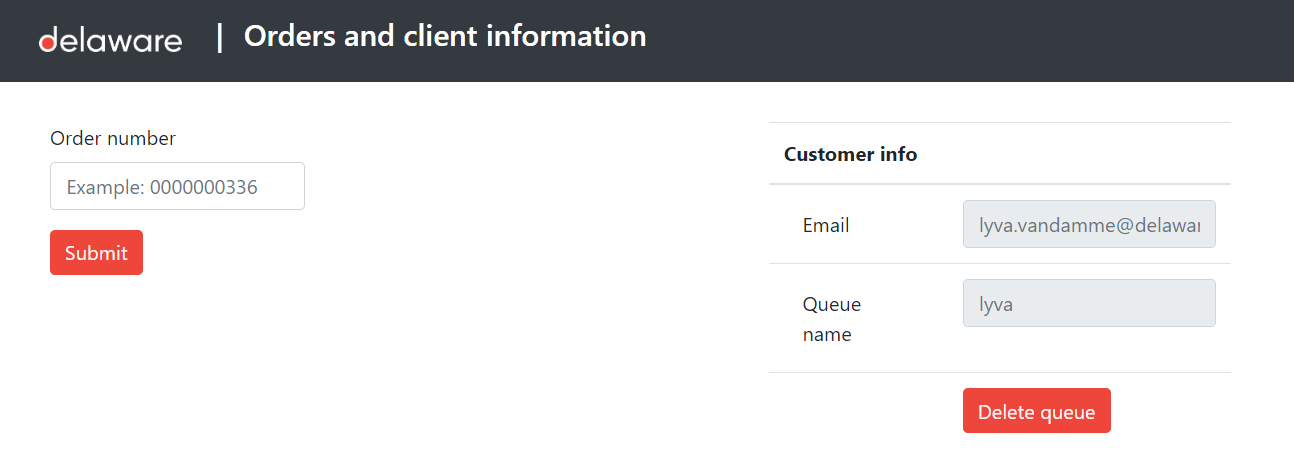
# De grafische interface

## Aanmaken queue

Dit is het eerste scherm dat een klant krijgt. Er moet een bedrijfsnaam meegegeven worden. Dat is dan de naam van de queue de achterliggend gemaakt wordt. Ook moet er een emailadres meegegeven worden. Dit is om mails te sturen voor bevestiging bij het aanmaken van een queue, het subscriben op een topic, het verwijderen van een subscription en het verwijderen van een queue.

Bij het vergeten invullen van één van de twee velden komt er een error tevoorschijn.

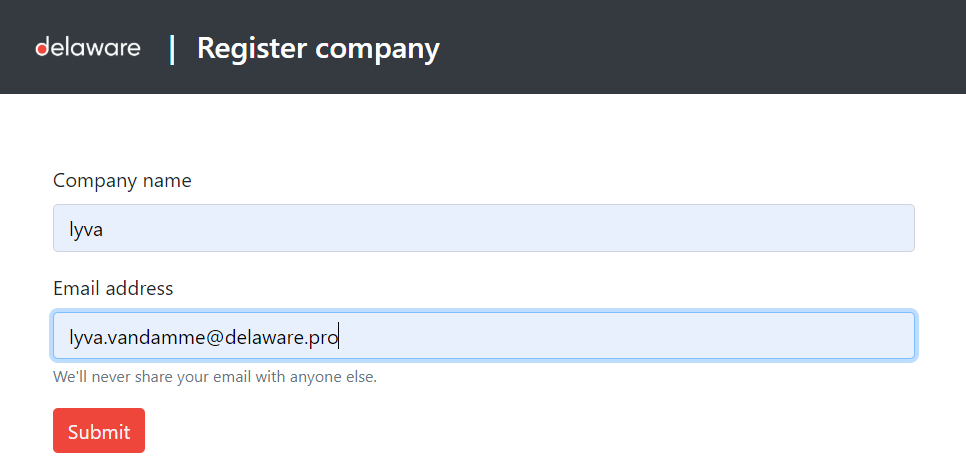
## Subscriben op een topic



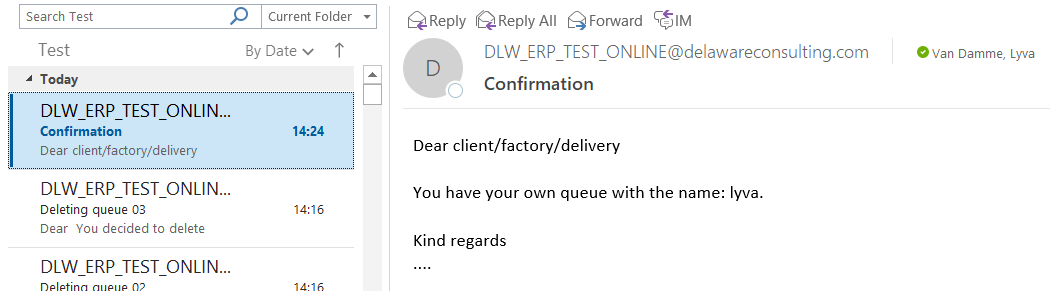
Als ze het vorige formulier correct ingevuld is, worden ze doorgestuurd naar dit scherm. Hun bedrijfsnaam en email zijn automatisch ingevuld en kunnen ook niet aangepast worden. Om meldingen te ontvangen van orders, moet er een ordernummer ingevuld worden. Wordt er geen ordernummer ingevuld, dan komt er een foutmelding. Er wordt dan geen request gemaakt. Is het order nummer correct ingevuld dan wordt er een subscription aangemaakt.

# Een voorbeeld

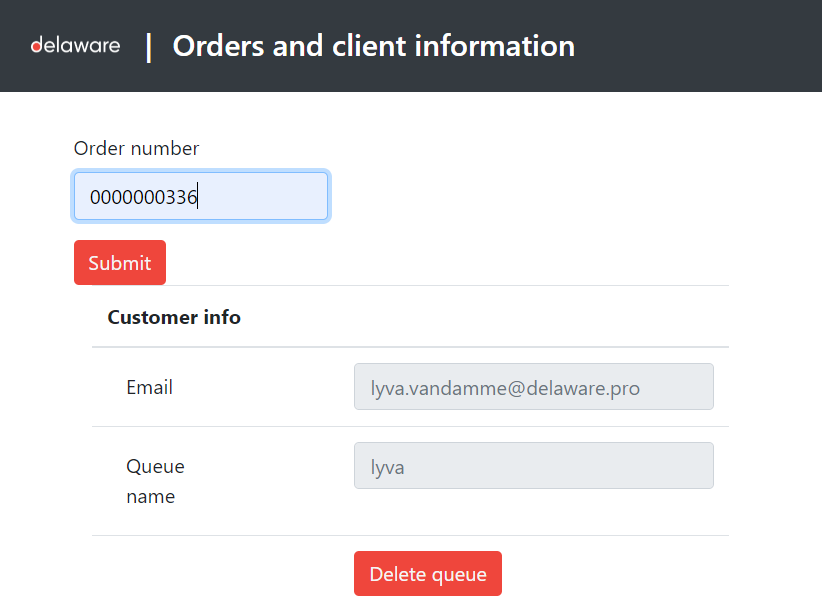
## Aanmaken queue

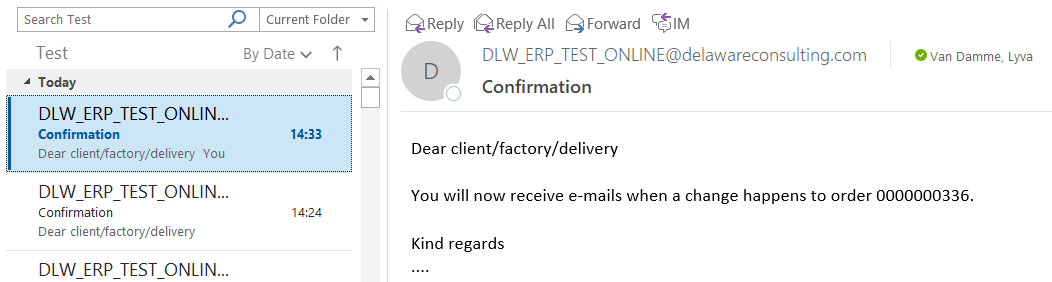


De bevestigingsemail:

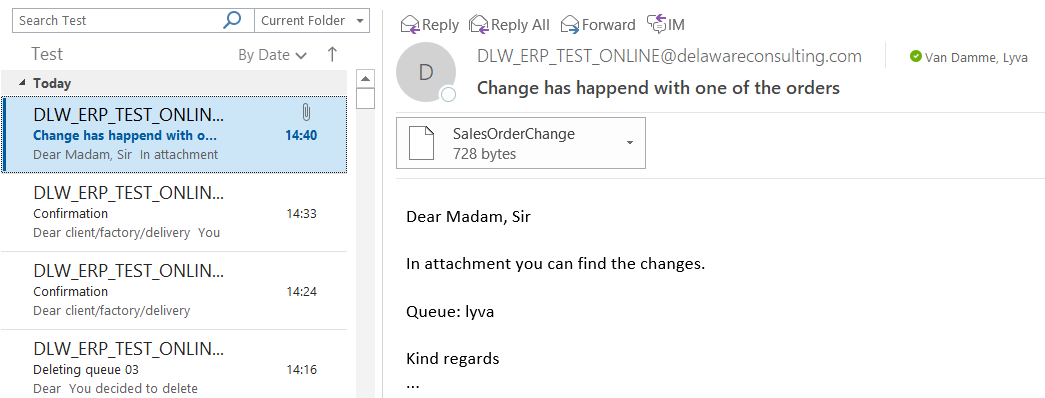


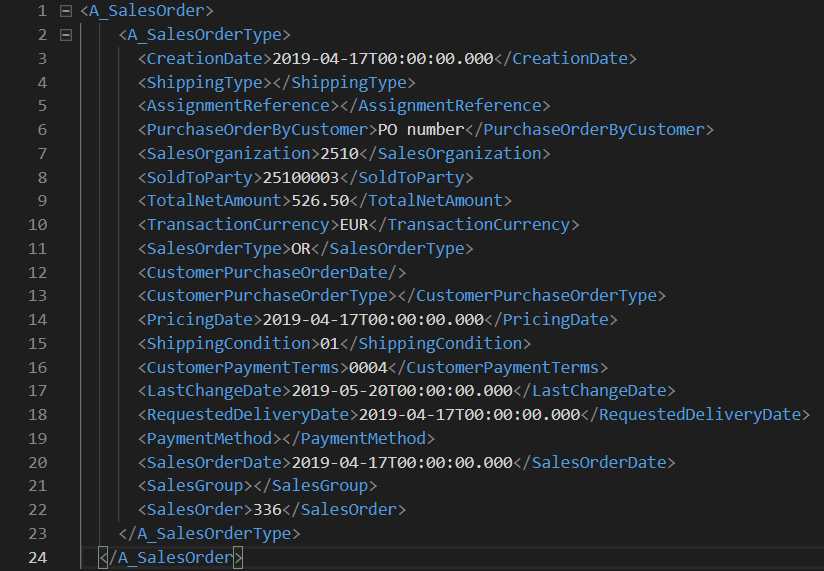
## Subscriben op topic





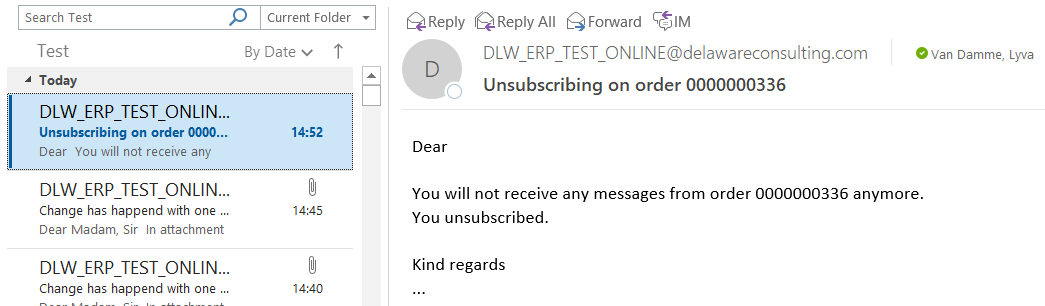
## Ontvangen van mail over verandering op order 0000000336





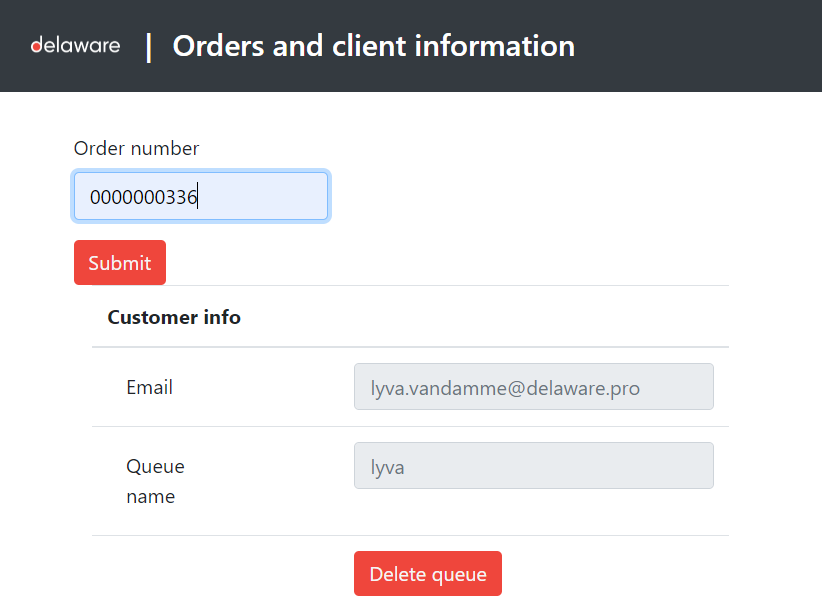
## Verwijderen van queue subscription

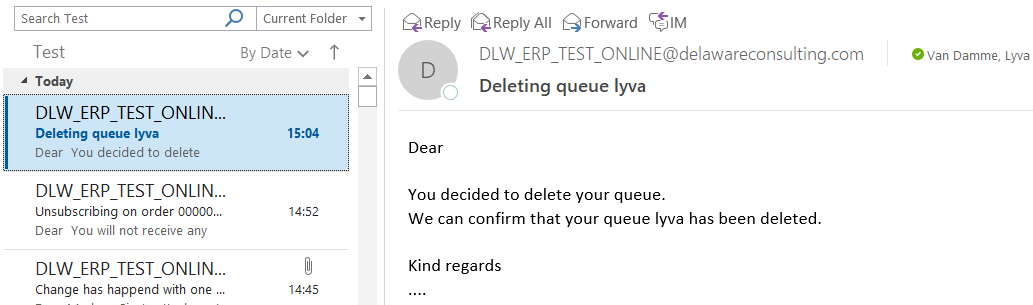
Het verwijderen van een queue subscription kan niet via de grafische interface. Dit gebeurt via POSTMAN.



## Verwijderen van de queue

Het verwijderen kan via de interface. De knop ‘delete queue’ verwijdert de queue van de klant. De pagina verandert naar de startpagina.





Business doelstellingen

Het volgende wordt verwacht na de afronding van mijn stage:

* Werkende demo
* Architectuur plaatje van de gemaakte oplossing
* Documentatie van de opzet
* Documentatie van het gebruik met test cases

Een grafische interface is geen must have maar een nice to have.

Persoonlijke doelstellingen

Persoonlijk wil ik graag mezelf tot het uiterste duwen en nieuwe dingen op zelfstandige basis leren. Ook wil ik werken aan mijn zelfstandig leren en verder denken. Eindreflectie

In het begin was het niet gemakkelijk om met CPI te werken. Het is een systeem waar wij, op HoGent, niet met leren werken. Ik heb altijd logica leren omzetten in code en niet in CPI. Dat was een eerste obstakel. Leren om de logica om te zetten in low-code development.

Er moest veel opzoek werk gedaan worden over het onderwerp. Enterprise messaging

Verklarende woordenlijst

|  |  |
| --- | --- |
| **Woord** | **Uitleg** |
| Cloud Platform Integration | Het cloud-integratie platform van SAP. |
| IFlow | Een grafische voorstelling van een interface, gebaseerd op BPMN. |
| Queue | Een wachtrij waarbij First-in First-out wordt toegepast. |
| Topic | Subscribers in een topci-based systeem ontvangen alle berichten die naar de topic gestuurd werden. Een topic kan gezien worden als een verdeler. Er kunnen meerdere mensen een bericht sturen naar de verdeler. De verdeler stuurt ze dan naar de subscribers. Een verstuurd bericht wordt niet opgeslaan bij een topic. |
| Queue subscription | Dit is een term binnen Enterprise Messaging. Een queue gaat subscriben op een topic. Dus verdeeld de topic zijn bericht dan slaat een gesubscribde queue die op. En wacht tot wanneer het bericht er wordt afgehaald. |
| S/4 HANA | Dit is een business intelligence suite die gebaseerd is op SAP hun operationele databasesysteem en in het geheugen computerplatform. |
| Consumption | Het lezen van een bericht op de queue. |
| Acknowledgement | Het ophalen en verwijderen van een bericht dat op de queue staat. |
| JSON | Javascript Object Notation.  Het is een manier om data te structureren. Dit wordt vaak gebruikt om data tussen een server en webpagina te sturen. |
| XML | Extensible Markup Language.  Het is een opmaak taal. Het is een verzameling van regels die leesbaar zijn voor mensen als voor een machine. |

Bibliographies

Panda, P. (2018). AP Cloud Platform Enterprise Messaging – Making S/4HANA Event Notification Easy. *https://blogs.sap.com/2018/12/18/sap-cloud-platform-enterprise-messaging-making-s4hana-event-notification-easy/*.