

Microservice Integration Patterns on Order-to-Cash proces in SAP

Onderzoeksvoorstel Bachelorproef

Lyva Van Damme¹

Samenvatting

Microservice is een nieuwe manier om services te maken. Deze manier kan de huidige integration patterns vervangen bij het maken van een stabielere applicatie. Uit onderzoek moet blijken hoe deze technologie kan toegepast worden op het order-to-cash proces in SAP. SAP is een veel gebruikte software. We gaan kijken hoe microservices een order-to-cash proces beïnvloedt. Zodat de services geen hinder ondervinden bij het falen of vertragen van één of meerdere services.

Sleutelwoorden

Onderzoeksdomein. E-business — ERP-Systems — SAP

Co-promotor

Nicolas Pauwelyn² (Nicolas.Pauwelyn@delaware.pro)

Contact: ¹ lyva.vandamme.y7102@student.hogent.be;

Inhoudsopgave

1	Introductie	1
2	Literatuurstudie	1
2.1	Waarom microservices gebruiken	1
2.2	Principes voor Microservices Integration	2
2.3	Order-to-cash in SAP	2
3	Methodologie	2
4	Verwachte resultaten	2
5	Verwachte conclusies	2
	Referenties	2

1. Introductie

Hoe Microservice integration patterns een order to cash proces in SAP beïnvloedt. Dit onderwerp werd gekozen omdat deze nieuwe technology een interessante invloed kan hebben op de order-to-cash proces. Dit is een manier om connecties te maken met de databank. In de meeste software wordt gebruik gemaakt van één grote databank of meerdere databanken die in staan zijn om meerde services te voorzien van data. Bij microservice integration patterns wordt voor elke service een aparte databank opgesteld. Niet voor elke kleine service kan een aparte databank gemaakt worden. De kost zou hoog kunnen oplopen. Er worden meestal grote stukken opgedeeld in kleinere. Dit heeft als voordeel om de performantie te verhogen. Maar de belangrijkste vraag is namelijk: Hoe microservice integration patterns een order-to-cash in SAP beïnvloedt.

2. Literatuurstudie

Over het onderwerp "Microservice Integration Patterns on Order-to-Cash proces in SAP", is er nog niets verschenen. De meeste informatie komt uit artikels die meer uitleg geven over microservices en artikels met een uitgebreide beschrijving over wat order-to-cash inhoudt.

2.1 Waarom microservices gebruiken

In het artikel van Gunaratne, 2018 werd besproken hoe je een microservice werkt. En waarom deze gebruikt worden. Volgens dit artikel zijn microservices een goede, nieuwe techniek die op lange termijn huidige SOA's kan vervangen.

Atrash, 2018 beschrijft waarom je deze techniek kunt gebruiken, in de plaats van de kleinere SOA-services. Ook hier wordt verwezen naar de belangrijkheid van de requirements van de business. Door de grootte van deze services, is er de mogelijkheid om te caching.

Devoteam, 2018 legt uit waarom microservices een kleine-SOA is. Een microservice omvat bepaalde, aanvullende, concepten omliggend deze 'kleinere services' en dit is waar ze beginnen met aantonen van de verschillen.

Is de integratie van microservices wel mogelijk? Deze vraag beantwoordt door Van Bart, 2018. Software wordt meestal nog geïmplementeerd in de 3-tiers manier. Ook wel monolithic genoemd. De applicatie is uit een alleenstaande unit gemaakt. Eén verandering heeft een impact op de volledige applicatie. Is dit dan een geldige reden om voor microservices te kiezen? Dat hangt af van wat je applicatie percies nodig heeft. Niet alle applicaties worden er beter van om een microservice te implementeren. Een microservice bestaat er uit om op zichzelf te werken. Dit wordt uitgelegd in het volgende deel.

2.2 Principes voor Microservices Integration

De principes van microservices integration werden uitgelegd in volgend artikel Aradheye, 2018 In het artikel wordt op een duidelijke manier uitgelegd hoe microservices worden gebruikt. Microservices worden het best opgesteld aan de hand van business units. Een microservice wordt benaderd vanuit de business requirements. Deze hebben, volgens dit artikel, een betere performantie dan de huidige gebruikte techniek. Microservices worden opgedeeld in verschillende klassen. Bijvoorbeeld:

- één voor klantendata
- één voor bestellingen
- één voor "wil-ik"lijsten

De belangrijkste eigenschappen van microservices zijn:

- Microservice bestaat uit meerdere componenten
- Gemaakt voor de business
- Microservice maakt gebruik van simpele routing
- Een microservice is gedecentraliseerd
- Een microservice werkt zelfstandig

2.3 Order-to-cash in SAP

Order-to-cash is een van de vele processen in SAP die vast gelegd zijn. Dit proces legt uit hoe men van een bestelling naar de inning van het geld gaat. Er zijn verschillende versies van hoe het proces gaat. Volgens Akthar, 2018 verloopt het proces als volgt: er wordt een order geplaatst dan wordt die bestelling geleverd, daarna wordt er een factuur opgesteld en als laatste wordt het geld geïnd. Het proces op zich is niet ingewikkeld. OpenSAP, 2018 geeft veel meer uitleg over wat er achter de schermen gebeurt. Bij dit proces wordt de financiële kant van SAP aangesproken, ook de verkoop en distributie alsook de stock worden aangesproken. Een gedetailleerder proces houdt in dat er een sales order gemaakt wordt. Dan wordt de stock bekeken. Afhankelijk van de beschikbaarheid van de goederen kan er een levering gepland worden. Zijn de goederen beschikbaar, dan kan er na de planning van de levering, effectief geleverd worden. Als volgt wordt er een factuur opgemaakt. Als laatste komt dan het betalingsproces.

3. Methodologie

In dit werk gaan we onderzoeken hoe we Microservices kunnen integreren in SAP. De services die ze nu gebruiken vergelijken met microservices. Kunnen microservices de werking van SAP versnellen en performanter maken bij fouten? Eerst willen we het volledige systeem van SAP en microservices los van elkaar begrijpen. Daarna gaan we alles opdelen in kleinere services om die dan te implementeren in het gekozen pattern. Er zal uitgebreid getest worden. Ook op performantie wordt er grondig getest. Hoe zullen de andere services reageren wanneer er één uitvalt. De gekozen methode is opzetten en testen. Het onderzoek zal uitwijzen of microservices een goede keuze is voor grote, complexe software pakketten. Er zal ook onderzocht worden wat de impact is op de reactiesnelheid van de services.

4. Verwachte resultaten

De microservices zouden moeten zorgen voor een stabielere applicatie. Microservices zijn "Designed for failure". Ook

zou er uit dit project moeten komen dat andere microservices blijven draaien bij het uitvallen van één ervan. Ik denk dat voor sommige delen van SAP deze techniek een verbetering kan zijn. Maar alles in microservices omzetten is te complex. Bij enkele microservices zal ook de kost berekend worden.

5. Verwachte conclusies

De conclusie die we uit dit onderzoek kunnen trekken: microservices integration patterns zijn voordeliger en gebruiksvriendelijker dan het huidige systeem is voor de mensen die de software gaan gebruiken. Bij fouten aan een service, zal het platform nog beschikbaar zijn. Het risico is wel dat door de relatief nieuwe techniek, er enkele dingen niet zullen lopen zoals we zouden willen. Het is mogelijk dat we maar tot een gedeeltelijke conclusie komen.

Referenties

- Akthar, J. (2018). What order-to-cash cycle controls in SAP ensure compliance?
- Aradheye, Y. (2018). Principles for Microservices Integration. *DZone*.
- Atrash, M. A. (2018). Why microservice architecture? *Devoteam*.
- Devoteam. (2018). Microservices: just another word for 'tiny-SOA'? *Devoteam*.
- Gunaratne, I. (2018). Wiring Microservices, Integration Microservices & APIs. *ContainerMind*.
- OpenSAP. (2018). Order-to-cash overview. *Order-to-cash overview*.
- Van Bart, A. (2018). Microservices and integration: Doing IT right? *devoteam*.