

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

## Onderzoeksvoorstel Bachelorproef

Lyva Van Damme<sup>1</sup>

### 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. Dit zal een vergelijkende studie zijn.

### Sleutelwoorden

Onderzoeksdomein. E-business — ERP-Systems — SAP

### Co-promotor

Nicolas Pauwelyn<sup>2</sup> (Nicolas.Pauwelyn@delaware.pro)

Contact: <sup>1</sup> lyva.vandamme.y7102@student.hogent.be;

## Inhoudsopgave

1	Introductie	1
2	Literatuurstudie	1
2.1	Wat zijn microservices?	1
2.2	Waarom microservices gebruiken	1
2.3	Principes voor Microservices Integration	2
2.4	Order-to-cash in SAP	2
2.5	Kyma	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 technologie een interessante invloed kan hebben op de order-to-cash proces. Dit is een manier om een proces robuuster te maken. 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. Dit is maar een klein deeltje van een microservice. De microservices moeten voldoen aan business requirements. SAP zelf heeft ook al veel ondernomen omtrend microservices. Eén van hun oplossingen is Kyma. Maar de belangrijkste vraag is namelijk: Hoe microservice integration patterns een order-to-cash in SAP beïnvloedt. Deze bachelorproef zal voor het grootste deel een theoretische vergelijking zijn. Omdat deze

studie als bachelorproef dient, is er maar beperkte tijd en resources om een onderzoek te doen.

## 2. Literatuurstudie

Over het onderwerp "Microservice Integration Patterns on Order-to-Cash proces in SAP", zijn er nauwelijks thesissen te vinden. De meeste informatie komt uit artikels die meer uitleg geven over microservices en artikels met een uitgebreide beschrijving over wat order-to-cash inhoudt. Voor wie denkt dat microservices iets nieuws is, zit er een beetje naast. Grote bedrijven zoals Netflix, Twitter, Amazon en facebook maken al gebruik van deze technologie. Ciber, 2018

### 2.1 Wat zijn microservices?

Het bouwen van aparte functies/modules met hun eigen interface en methoden. Deze manier van werken is in het voordeel van Agile. Bij Agile wordt er gewerkt met deeltjes software opleveren en opgeleverde software, daar wordt er zo goed als niks meer aan veranderd. Microservices worden onderverdeeld aan de hand van business requirements. Deels zorgen microservices ervoor dat er beter moet worden samengewerkt met de business.

### 2.2 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 monolithisch 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 precies 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.

### 2.3 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

### 2.4 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.

### 2.5 Kyma

Kyma is een open-source project van SAP. Het is vooral gebaseerd op Kubernetes. Op deze manier kan je oplossingen in de Cloud maken. Kyma is speciaal omdat zij zo

goed als alle oplossingen op één plek hebben. Zij hebben een application connector. Kyma is serverless. Ze maken service management eenvoudiger. Kyma, 2019 Volgens het artikel van Semerdzhiev, 2018 is er meer nood aan openheid en een modernere architectuur. Dat is ook de reden waarom Kyma een open source project is. Kyma ondersteund container-based werken (zoals docker) alsook cloud-native apps.

## 3. Methodologie

In dit werk gaan we onderzoeken op welke manier microservices een invloed zou kunnen hebben op een order-to-cash proces in SAP. De services die ze nu gebruiken vergelijken met microservices. Kunnen microservices de werking van SAP versnellen en performanter maken bij fouten? Welke messaging manier zou het beste kunnen zijn. Eerst willen we het volledige order-to-cash proces verstaan. Dan gaan we gaan onderzoeken welke verschillende mogelijkheden er zijn in verband met microservices. Kyma, PaaS.io of zijn er nog andere die een grote rol kunnen spelen. Ook moet er gekeken worden welke manier van communiceren tussen de microservices het beste is. Voor dit onderzoek zal er veel literatuur studie gedaan worden.

## 4. Verwachte resultaten

Naar de gelezen literatuur kijkende, zou Kyma eigenlijk de beste oplossing moeten zijn. Deze is namelijk zelf afkomstig van SAP. Dit zou een goede oplossing moeten zijn. Maar zijn microservices wel haalbaar in een order-to-cash proces in SAP. Eerst zal dit duidelijk moeten worden vooraleer we gaan kijken naar welke microservice de best mogelijk noden opvult.

## 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? *SAP*.
- Aradheye, Y. (2018). Principles for Microservices Integration. *DZone*.
- Atrash, M. A. (2018). Why microservice architecture? *Devoteam*.
- Ciber. (2018). Microservices: het einde van de SAP spaghetti? <https://www.ciber.nl/blog/deel-2-microservices-het-einde-van-de-sap-spaghetti>.
- Devoteam. (2018). Microservices: just another word for 'tiny-SOA'? *Devoteam*.
- Gunaratne, I. (2018). Wiring Microservices, Integration Microservices & APIs. *ContainerMind*.

- Kyma. (2019, februari 13). What is kyma? <https://kyma-project.io/>.
- OpenSAP. (2018). Order-to-cash overview. *Order-to-cash overview*.
- Semerdzhev, K. (2018, juli 24). Introducing project Kyma. <https://kyma-project.io/blog/introducing-project-kyma/>.
- Van Bart, A. (2018). Microservices and integration: Doing IT right? *devoteam*.