



Deel 2 – Microservices: het einde van de SAP spaghetti?



30-11-2018

Microservices: het einde van de SAP spaghetti?

In deze tweedelige blog leggen we uit hoe het gebruik van microservices de klassieke problemen van SAP ERP producten zouden kunnen verhelpen. Dit doen we door het laatste aanbod van SAP op dit gebied te belichten, SAP S/4HANA. Ook bespreken we hoe microservices daarnaast nieuwe kansen kunnen bieden voor SAP en haar partner-ecosysteem. In deel 1 bespreken we de blind spots in de huidige S/4HANA propositie. In deel 2 stellen we microservices voor als oplossing en werken we een blik

propositie. In deel 2 stellen we microservices voor als oplossing en werpen we een blik op de agenda van SAP op dit gebied.

Microservices in een notendop

Microservices zijn in een rap tempo populair geworden. Dit komt doordat toonaangevende bedrijven zoals Netflix, Twitter en Amazon laten zien hier wereldwijd succesvol mee te zijn. Daarnaast heeft de ontwikkeling van microservices een vlucht genomen door de groei van Cloud leveranciers en de opkomst van container-based technologieën zoals Docker en Kubernetes. Hoewel er sprake is van een hype, biedt het gebruik van microservices mogelijkheden om de problemen rondom het opschalen van complexe, monolithische toepassingen het hoofd te bieden.

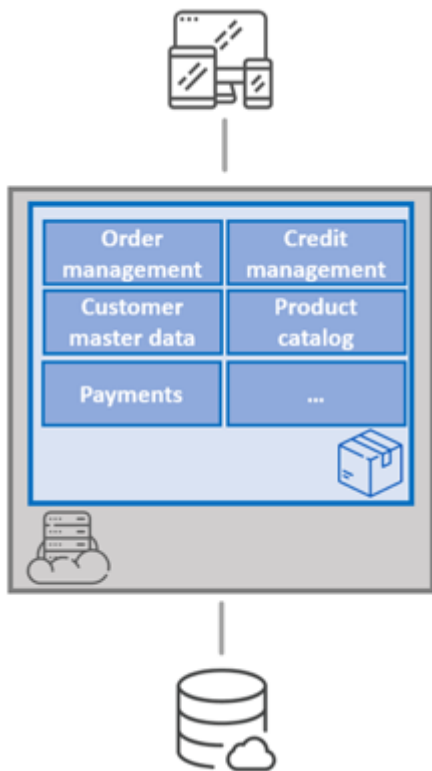
Maar wat zijn microservices nu eigenlijk? In abstracte zin is een microservice te beschrijven als een onafhankelijk inzetbare en schaalbare toepassingen die een functie of taak binnen een organisatie vertegenwoordigd. Deze functie-specifieke toepassingen opereren daarmee autonoom van andere functies die binnen een organisatie kunnen bestaan. In de meeste pure vorm heeft elke microservice dan ook zijn eigen database, een eigen gebruikersinterface en wordt deze ontsloten via een API.



De microservice 'verpakking' voor een bedrijfsfunctie met daarin een gebruikersinterface, toepassingslogica, database en (RESTful) API

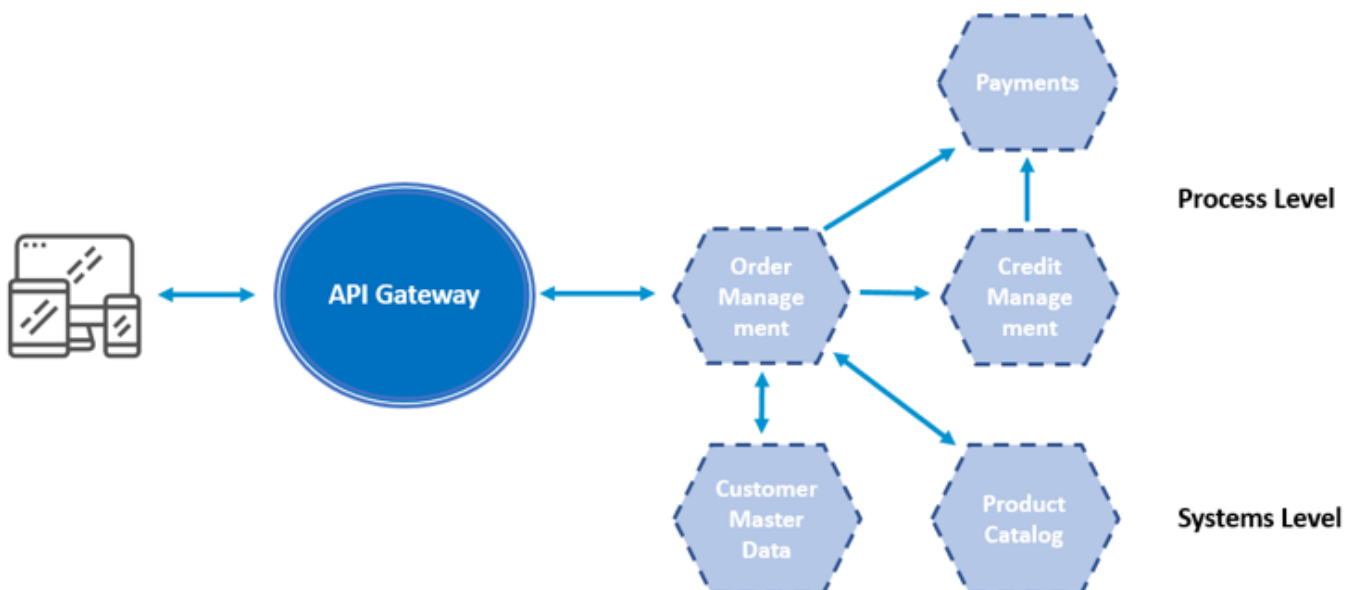
Ter illustratie, in het voorbeeld van een webshop, kunnen de orderinvoer en het debiteurenbeheer als aparte bedrijfsfuncties worden afgebeeld, elk vertegenwoordigd door verschillende afdelingen binnen een organisatie. In een monolithische omgeving zou onze applicatie er als volgt uit kunnen zien:

monolithische omgeving zou onze applicatie er als volgt uit kunnen zien.



Webshop als monoliet

In het geval van microservices ziet de architectuur er wat anders uit. In plaats van een monolithische eenheid, functioneert het geheel in onafhankelijke, maar gekoppelde bedrijfsfuncties. Zo zouden er in dit geval afzonderlijke microservices aangelegd kunnen worden voor de orderinvoer en het debiteurenbeheer.



Webshop als microservice architectuur

In dit voorbeeld wordt de order invoer service benaderd via de centrale API gateway, waar deze op zijn beurt de onderliggende service met betrekking tot het klantenbestand raadpleegt. Kredietlimieten worden voor klanten per order

gecontroleerd. Dit laat zien dat microservices zowel sequentieel als hiërarchisch geordend kunnen zijn, enkelvoudig of in een samenstelling van meerdere onderliggende services.

Maar wat is hier het nut van? Uit zowel technisch als bedrijfskundig oogpunt hebben microservices voordelen te bieden.

Geen eenheidsworst – Elke microservice kan onafhankelijk worden ontwikkeld, getest, ingezet en opgeschaald, afhankelijk van de vereisten voor die specifieke service. One-size-fits-all afhankelijkheden zijn niet van toepassing.

Altijd up-to-date – Het autonome team, verantwoordelijk voor de ontwikkeling en beheer van de microservice, kan zelf besluiten over de te gebruiken technologieën. Nieuwe trends, platformen en technologieën kunnen op microservice niveau worden ingebed, zonder het functioneren van andere microservices te verstoren.

Commerciële kansen – Omdat elke individuele microservice is ontsloten via een API, kunnen prijsmodellen op serviceniveau worden geïmplementeerd. Facturering kan plaatsvinden op basis van daadwerkelijk gebruik van een microservice.

Risicovrij uitbesteden – Het uitbesteden van ontwikkel- en beheerwerkzaamheden is eenvoudig. Dit omdat het offshore team enkel toegang krijgt tot aangewezen microservices. Daarmee kan worden voorkomen dat buitenstaanders toegang krijgen tot het 'geheime recept' van het bedrijf. Met andere woorden: het is met microservices dus makkelijker om de toegang van derde partijen tot het intellectueel eigendom van de organisatie (die meer en meer in werkende software wordt omgezet) te managen.

Elimineer vendor lock-in – Met microservices worden de afnemers van die diensten in de bestuurdersstoel gezet om de voor hen relevante services te consumeren. Microservices kunnen worden geconsumeerd door middel van breed toegepaste standaarden (REST/JSON), waardoor nieuwe microservices makkelijk te integreren zijn zonder dat er diepgaande kennis nodig is over hoe de services technisch in elkaar steken of welke technologieën worden gebruikt.

Versnel staffing – Nieuwe teamleden kunnen snel op de hoogte worden gebracht, omdat één microservice maar een beperkte functionele scope betreft.

Fout isolatie – Technische fouten zoals geheugenlekken zijn beperkt tot die specifieke microservice, aangezien andere microservices onafhankelijk daarvan worden uitgevoerd.

Microservices maken zo een hogere flexibiliteit en productiviteit mogelijk. Daarnaast stellen ze bedrijven in staat om op een granulair niveau organisatiemiddelen te managen, afhankelijk van de daadwerkelijk benodigdheden. Een microservice biedt daarmee een afgestemde functie-als-een-service, in plaats van de traditionele

koelkastbenadering. Hoewel veelbelovend, werpt het gebruik van microservices ook een aantal uitdagingen op. Dit zijn vooral vragen die voortkomen uit het introduceren van een gedistribueerde computeromgeving, waarbij elke microservice een keten van afhankelijk services kan hebben. Het gaat buiten de context van dit artikel om hier uitvoerig bij stil te staan, maar de oplossingen hiervoor zijn ofwel aanwezig in de vorm van tooling (i.e.: service mesh brokers), dan wel in de vorm van nieuwe concepten (i.e.: event-gedreven architecturen).

On-topic: hoe kan deze discussie worden teruggebracht naar ERP en SAP?

Agile ERP: van tegenstelling tot werkelijkheid

Stel je voor dat, in de plaats van één grote monoliet, een ERP suite bestaat uit een uitgebreide catalogus van microservices. Elke microservice vertegenwoordigd een bedrijfsfunctie en wordt standaard aangeboden op basis van de geldende industriële best practices voor het betreffende bedrijfsproces. Elke service kan afzonderlijk worden geïmplementeerd, ontwikkeld en worden opgeschaald, in de cloud of on-premise, afhankelijk van de vereisten van de organisatie.

Elk van deze microservices kan individueel worden geconsumeerd door klanten, waarbij de prijzen zijn gebaseerd op het daadwerkelijk gebruik van de microservice. Omdat niet alle bedrijven binnen één best practice template te drukken zijn, kunnen organisaties ervoor kiezen om een service uit te breiden of aan te passen wanneer differentiatie nodig is. Afhankelijk van de uitgegeven machtigingen voor die microservice, kunnen de aanpassingen worden teruggevoegd in de standaard of als volledig nieuwe service binnen de catalogus worden gepubliceerd, waarin deze direct bruikbaar is voor andere consumenten in het ecosysteem. Daarbij kan elke service worden ontwikkeld in een variëteit aan programmeertalen en technologieën, wat betekent dat elk ontwikkelteam direct met deze microservices uit de voeten kan en de functionaliteit in de organisatie kan gaan implementeren.

Op deze manier kunnen de inspanningen voor het doorontwikkelen van de ERP suite worden verdeeld over het ecosysteem van gebruikers, terwijl het collectief als geheel profiteert van continue gepubliceerde verbeteringen. Op deze manier wordt dubbel werk voorkomen en past de suite zich op een natuurlijke manier aan aan de laatste trends en innovaties.

De doorlooptijd van implementaties is kort, wat niet alleen betekent dat het management een duidelijk grip houdt op kosten, maar ook op de ontwikkeling van organisatieprocessen, waarbij IT en business steeds meer met elkaar verweven zijn. De implementatieteams zijn georganiseerd als autonome, multidisciplinaire eenheden waarin IT-experts nauw samenwerken met de kennishouders uit de business.

De op gebruik gebaseerde prijsmodellen verlagen de TCO van de ERP suite, terwijl de gerealiseerde functionele waarde gemaximaliseerd wordt. Zo kunnen zelfs multinationale organisaties met een veelheid van complexe processen hun wendbaarheid waarborgen en bedrijfsresultaten stimuleren. Dit in tegenstelling tot bedrijven die vasthouden aan de verzonken kosten in rigide, ouderwetse ERP monolieten, die hen als een blok aan het been achter zullen houden.

Klinkt goed! Wat nu?

Als we terug in de realiteit stappen, kunnen we dan van een SAP klant verwachten dat zij zelf het pionierswerk gaan leveren om de gehele ERP suite te herschrijven in een microservice architectuur? Natuurlijk niet. Er is in dit verhaal een duidelijk rol weggelegd voor de gevestigde ERP leveranciers. Het is in deze rol dat spelers zoals SAP als product leverancier kunnen dienen. En waar implementatie partners zoals Experis Ciber, op basis van jarenlange ervaringen uit verschillende branches en industrieën, zouden moeten samenwerken om dit te vermarkten. Het zou in dat licht interessant zijn wanneer een ERP leverancier voor een open-source benadering zou durven kiezen en daarmee voor maximale transparantie gaat.

Op dit punt moet benadrukt worden dat een microservice-architectuur niet altijd beter is dan een monolithische. Zoals altijd is de beste benadering afhankelijk van de situatie en het probleem dat moet worden opgelost. Monolithische architecturen hebben namelijk door de tijd al lang en breed bewezen in veel gevallen prima te werken. Dit geldt ook voor de ERP markt, nu en in de toekomst. Wanneer een bedrijf zich kan schikken naar de grenzen van de best-practice monoliet, maar daarnaast op andere manieren strategische differentiatie weet aan te brengen, dan zou dit een succesvolle strategie kunnen zijn.

Het zou daarom verstandig zijn dat ERP leveranciers microservices op een complementaire manier gaan aanbieden, waarbij klanten zelf kunnen beslissen welke vorm het beste bij hen past. Bovendien zal voor klanten de keuze voor microservices onderdeel moeten zijn van een breder actieplan, waarbij het gebruik van microservices geen doel op zich is, maar een logisch gevolg van de bedrijfsvoering. Voor organisaties die naar microservices willen bewegen is het daarom van belang een integrale, langetermijnvisie te ontwikkelen om het applicatielandschap stapsgewijs te ontwarren, waarbij ERP slechts een deel van de puzzel vormt.

SAP's agenda

Kijkend naar de activiteiten van SAP, werd er in het verleden binnen het CRM domein al geëxperimenteerd met het gebruik van microservices middels een project genaamd Yaas.io. Dit project werd eerder dit jaar echter stilzwijgend geannuleerd. Hoewel

gehuld in speculatie, betekende dit toch niet het einde van microservices voor SAP.

Meer recentelijk heeft SAP namelijk het Kyma project gelanceerd. Kyma is een bundeling van vooraanstaande open-source technologieën die het vooruitstrevende klanten makkelijk moet maken om microservices te integreren in het landschap. De betrokkenheid van SAP binnen dit project geeft een bemoedigend signaal voor het gebruik van microservices en we verwachten dan ook op korte termijn hier de eerste resultaten van terug te zien in het product portfolio van SAP.

Deel je mening!

Vanuit Experis Ciber zien wij microservices als een interessante ontwikkeling die een hoop nieuwe kansen biedt. Kansen die, op dit moment misschien nog, onbenut worden gelaten door de gevestigde spelers. Kijkend naar ERP in het bijzonder, is de grootste aanstaande innovatie daarmee misschien niet van technische, maar van architectonische aard. En als dit niet door de grote spelers wordt opgepakt, zou dit wel eens een gat in de markt voor nieuwe spelers kunnen zijn!

Wat is jullie microservice roadmap? Wil je meer weten over microservices, of heb je vragen over andere trends en emerging technologies, dan mogen jullie ons contacteren op LinkedIn, door op onze namen hieronder te klikken. We komen ook graag in contact met je via mail of telefoon.

Bron infographics: [SAP NL](#)

Martijn Hulshof & Robin Day

Business Thema's

Partnerships & Co-creation >

IT-Trends

API-First Economy > Cloud First > Data & Analytics > Security by Design >