

Eisen en richtlijnen aan services

Soa en de kwaliteit van services

Service oriented architecture (soa) is een architectuurstijl waarbij applicaties van elkaar ontkoppeld worden door services. Een service is een duidelijk gedefinieerde functie die een applicatie aanbiedt aan andere applicaties. De interface van de service is daarbij ontkoppeld van de implementatie. Andere typische eigenschappen van services zijn hun onderling losse koppeling, het feit dat ze los van andere services kunnen worden ingezet en dat ze toestandloos zijn. Het streven is om elke functionaliteit maar door één service aan te laten bieden waardoor alle redundantie in het applicatielandschap wordt geëlimineerd. De beoogde voordelen hiervan zijn meervoudig. Het verhoogt de flexibiliteit in het applicatielandschap, waardoor nieuwe applicaties snel kunnen worden toegevoegd en bestaande applicaties snel aan elkaar kunnen worden gekoppeld door gebruik te maken van bestaande services. De vaak vele specifieke koppelingen tussen applicaties worden vervangen door een kleiner aantal generieke koppelingen, waardoor ontwikkeling en onderhoud minder tijd en geld kosten. Ook ontstaat er expli-

Veel organisaties hebben een omvangrijke verzameling applicaties gecreëerd die overlappen en vaak op allerlei manieren aan elkaar zijn gekoppeld. Wijzigingen zijn daardoor moeilijk door te voeren. Dit leidt ertoe dat de flexibiliteit van de organisatie afneemt en de kosten van ontwikkeling en onderhoud van de applicaties relatief erg hoog zijn.

Danny Greefhorst, Jan Miedema, Hans Rijks

ciete aandacht voor het eenmalig definiëren van functionaliteit, waarbij overlappende en overvloedige applicaties kunnen worden geconsolideerd. De onderhoudskosten van het applicatielandschap kunnen daarbij drastisch afnemen.

Door de genoemde voordelen krijgt service oriented architecture tegenwoordig veel aandacht bij bedrijven, met name als kostenbesparingsinitiatief. De focus ligt daarbij echter vaak op het realiseren van individuele services, waarbij technologie een belangrijke rol speelt. Hoewel dit een logische eerste stap is naar het invoeren van een service oriented architecture, is het zeker niet de meest waardevolle manier. Beter is het indien er gefocust wordt op die integratie die vanuit de bedrijfsoptiek de meeste toegevoegde waarde heeft.

De basis

Het is duidelijk dat de services zelf een zeer belangrijke rol spelen in een service oriented architecture. In feite hangt het succes van de gehele architectuur af van de kwaliteit van deze services. Een nadere blik op de services leert dat er allerlei soorten services zijn die applicaties aan elkaar aanbieden, maar die onderling van elkaar verschillen. Het probleem is alleen dat er geen overeenstemming bestaat over de precieze soorten services en hun naam. Figuur 1 geeft een model weer waarin de belangrijkste soorten softwareservices zijn beschreven. Allereerst is er sprake van services die verantwoordelijk zijn voor het aansturen van langlopende bedrijfsprocessen. Deze noemen we hier *workflowservices*. Vanuit het proces wordt er typisch gebruik gemaakt van *applicatieservices*, die specifieke use-cases ondersteunen. Deze

Samenvatting

Service oriented architecture is een visie op architectuur waarbij applicaties netjes van elkaar ontkoppeld zijn. Door het invoeren van een service oriented architecture kunnen kosten worden bespaard en kan de time-to-market van nieuwe processen en applicaties worden verkort. Het definiëren van goede services is uiteindelijk een belangrijke sleutel voor het slagen van de architectuur. De auteurs beschrijven in dit artikel welke eisen er gesteld worden aan goede services en geven richtlijnen om aan die eisen te voldoen.

applicatieservices kunnen gebruikmaken van onderliggende services. *Bedrijfslogicaservices* bieden toegang tot de bedrijfsregels, die de essentie van de bedrijfsvoering beschrijven. *Gegevensservices* bieden toegang tot de bedrijfsgegevens, die een belangrijke waardebron voor organisaties zijn. De onderste laag van services zijn de *technische services*, die door de software-infrastructuur worden geboden. Indien een service een andere service aanroept om zijn functie uit te voeren, is er sprake van een samengestelde service, wat in feite een stuk proceslogica is. Op alle genoemde lagen kunnen samengestelde services voorkomen. Daarnaast kunnen alle lagen binnen een applicatie bestaan en worden aangeroepen om applicaties met elkaar te koppelen.

Eisen

Zoals al eerder aangegeven bepalen de services bepalen voor een groot deel de kwaliteit van de service oriented architecture. Maar wat is dan een goede service? Om op deze vraag een antwoord te kunnen geven worden een aantal eisen beschreven die door verschillende belanghebbenden gesteld zouden moeten worden. In feite gaat het om de servicekwaliteit die de afnemer van de aanbieder vraagt en de kwaliteitseisen die de aanbieder stelt om een effectieve en efficiënte exploitatie van de service mogelijk te maken.

1. Aansluiting op bedrijfsvoering

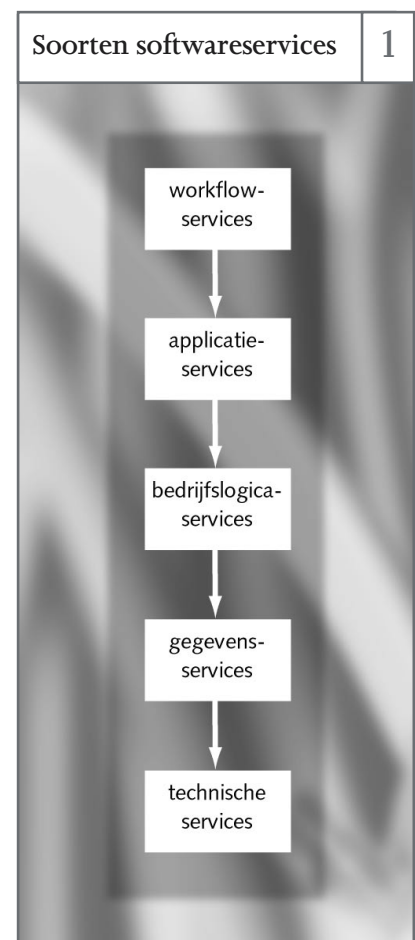
Een goede service biedt een optimale ondersteuning voor de huidige en toekomstige bedrijfsactiviteiten van de organisatie die de service gebruikt. Dat betekent dat een service doet wat er vanuit het proces van hem wordt verlangd. Een ander aspect van aansluiting bij de bedrijfsvoering is het herkenbaar zijn van de service voor de

medewerkers in de business. Dit betekent dat de service gebaseerd is op de terminologie van de business en in eenheden redeneert die de business herkent. Dit voorkomt dat de business wordt opgezaagd met verwarrende terminologie, waardoor deze zijn werk niet goed kan uitvoeren. Ook moeten services aansluiten bij de doelstellingen van de organisatie; alle voor de doelstellingen benodigde services dienen beschikbaar te zijn. Ten slotte is van belang dat services geleverd moeten kunnen worden door de bestaande bedrijfsapplicaties, die niet zomaar vervangen kunnen worden. Het is duidelijk dat onderzoek of afstemming nodig is om te voorzien in de aansluiting bij de bedrijfsvoering. Wil er in een kwaliteitssysteem enig zicht zijn op dit aspect, dan zal de aanbieder dus moeten kunnen aantonen in hoeverre dit onderzoek heeft plaatsgevonden. De eis tot het aansluiten bij de bedrijfsvoering is met name relevant voor de services boven in figuur 1 (workflow-services en applicatieservices).

2. Hergebruik

Services dienen bruikbaar te zijn in de processen van verschillende gebruikende partijen, dus in diverse contexten. Dat kan eigenlijk alleen als ze contextonafhankelijk worden gedefinieerd. Door het hergebruiken van services wordt het dubbel ontwikkelen van functionaliteit voorkomen en kan snel een nieuwe applicatie worden gedefinieerd. Er zijn twee belangrijke kanttekeningen te plaatsen bij de mogelijkheid tot hergebruik. Ten eerste is het te hergebruiken zijn relatief ten opzichte van de context waarbinnen de service is gedefinieerd. Een service kan prima te hergebruiken zijn binnen een applicatie, maar daarbuiten geen relevantie hebben. Een andere kante-

kening is dat het een bewuste keuze kan zijn om een service specifiek op maat te maken voor één afnemer, waarbij hergebruik geen uitgangspunt is. Aan het andere eind van het spectrum bevinden zich standaardservices die beschikbaar worden gesteld aan een groot aantal afnemers, zonder daarbij de precieze eisen van de afnemers te kennen. De aanbieder van standaardservices zal uitgaan van zijn beeld van de markt, al dan niet ondersteund door feitelijk marktonderzoek. Vooral in deze categorie is de inkoop of het aanbieden van standaard 'off-the-shelf'-services relevant. Hergebruik is met name haalbaar voor de services onder in figuur 1, maar heeft potentieel meer



toegevoegde waarde in de hogere lagen van de figuur.

3. Stabiel

Services dienen zodanig gedefinieerd te zijn dat hun interface niet of zeer weinig wijzigt, aangezien een wijziging in de interface van een service veelal ook tot wijzigingen leidt in alle afnemers van de service. Bedenk daarbij dat het primaire doel van een service oriented architecture nu juist is om de flexibiliteit van de organisatie te verhogen door de impact van wijzigingen te verlagen. Als er dan wijzigingen nodig zijn, dienen afnemers hiervan zo weinig mogelijk te merken. Bij wijzigingen is *backwards compatibility* dus een eis. Een aangepaste service moet dus zijn oude functionaliteit voor een bepaalde tijd blijven ondersteunen zodat afnemers de kans hebben in eigen tempo hun systemen aan te passen.

4. Eenduidig gedefinieerd

De definitie van een service en zijn interface dient helder te zijn voor de aanbieder en afnemer van de service. Deze helderheid is een noodzakelijkheid om goede afspraken met elkaar te maken en te voorkomen dat er misinterpretaties ontstaan. Dit betekent niet alleen dat er strikte afspraken moeten zijn over de naamgeving en syntaxis van de inhoud van de interface, maar dat er vooral helderheid dient te zijn over de betekenis. De betekenis van de service als geheel, evenals die van de gegevenselementen daarbinnen, moet helder zijn gedefinieerd.

5. Kwaliteit

Buiten de specifieke eerdere eisen moeten services voldoen aan de kwaliteitseisen van de afnemer. Dit betekent onder meer een adequate performance,

beschikbaarheid en beveiliging. Merk op dat het gaat om adequaat; maximaal is niet in alle gevallen adequaat. Een probleem daarbij is dat niet alle afnemers dezelfde kwaliteitseisen hebben. Toch dient een service al zijn afnemers tevreden te stellen. Dit betekent dan ook een afstemming van de kwaliteit van de dienstverlening (*quality of service*) tussen de aanbieder en de afnemer van de service. Idealiter verzorgt de aanbieder van de service de kwaliteit door deze in de infrastructuur in te regelen.

Richtlijnen

Deze eisen zijn te vertalen naar richtlijnen. Door het toepassen van een juiste combinatie van richtlijnen kunnen kwalitatief hoogwaardige services gedefinieerd worden. De eisen en daarmee ook de richtlijnen om aan deze eisen te voldoen zijn afhankelijk van de laag waarin de services zich bevinden (zie figuur 1).

Het top-down afleiden van services

Om services goed aan te laten sluiten bij de bedrijfsvoering is er een top-down analyse van bedrijfsprocessen nodig. Dit betekent dat de bedrijfsprocessen zullen moeten worden gede-composeerd in subprocessen, processtappen en handelingen. Processen op al deze niveaus zijn kandidaatservices. Een soortgelijke top-down analyse kan gedaan worden op de doelstellingen van de organisatie, bijvoorbeeld op basis van *key-process indicators* (kpi's). Deze kpi's kunnen vertaald worden naar subdoelen en uiteindelijk naar services die daarvoor benodigd zijn. Als een organisatie bijvoorbeeld tot doel heeft om de cross-selling te verhogen, kan een subdoel zijn dat er inzicht in de productafname benodigd is, en daar is dan een service voor nodig om voor een klant alle afgenomen producten te bepalen. Van belang bij deze top-down benadering is dat er nadrukkelijk wordt gekeken naar logisch bij elkaar horende eenheden van werk. Bedenk dat services die geoptimaliseerd zijn voor een proces minder goed te hergebruiken zijn. Hoewel het identificeren van services primair top-down zou moeten plaatsvinden, kan de definitie van een service uiteindelijk wel worden beïn-

vloed door beperkingen in bestaande bedrijfsapplicaties.

Granulariteit

In de context van services wordt er vaak gesproken over granulariteit ofwel de korrelgrootte van de services. Er zijn twee belangrijke aspecten aan servicegranulariteit: het proces en de gegevens. Wil een standaardservice voor een aantal gebruikende partijen gemeenschappelijk bruikbaar zijn, dan betekent dit dat voor al die partijen hetzelfde deel van het proces wordt ondersteund. Aangenomen dat de processen en de context bij de gebruikende partijen verschillend zijn, dan zal het gemeenschappelijke deel over het algemeen klein zijn, wat dus leidt tot een relatief fijnkorrelige service met betrekking tot het proces. Vanuit een gegevensperspectief is het verstandig zoveel mogelijk potentieel relevante gegevens op te nemen, waarmee de granulariteit met betrekking tot gegevens wordt vergroot. Bedenk wel dat wanneer een subset van de gegevens wezenlijk anders verwerkt moet worden, het weinig voordeel heeft voor de aanbieder om er één service van te maken. Ook wanneer de gegevens niet gemeenschappelijk voor alle afnemers van de service zijn, valt het te overwegen om er meerdere services voor te definiëren. Verder zal een service waar veel complexiteit in zit slechter te wijzigen zijn, waardoor onderhoud en beheer duurder wordt.

Gehele bedrijfsobjecten

Een bedrijfsobject is een voor de business herkenbaar object, zoals klant en order, en beslaat typisch meerdere databasetabellen. Een algemeen streven zou moeten zijn om een service zoveel mogelijk te laten communiceren in gehele bedrijfsobjecten. Deze bedrijfsobjecten definiëren nu eenmaal logisch bij elkaar horende gegevens. Een dergelijke service is beter te hergebruiken doordat er niet verschillende services hoeven te worden gemaakt voor afnemers met een net iets andere gegevensbehoefte. Deze richtlijn is vooral relevant voor meer gegevensgerichte services die gericht zijn op het raadplegen van gegevens. Voor het

vastleggen of wijzigen van gegevens kunnen er ook andere natuurlijke eenheden zijn voor het consistent en compleet kunnen registreren.

Multi-grained services

Het nadeel van het communiceren in gehele bedrijfsobjecten is dat deze vaak meer gegevens kunnen bevatten dan noodzakelijk is in een specifieke processtap. Dit kan een negatieve impact op de performance hebben aangezien meer gegevens moeten worden opgehaald en uitgewisseld. Een mogelijke optimalisatie is het voor de afnemer mogelijk maken om een deelverzameling van de gegevens te selecteren. Om de service toch generiek en herbruikbaar te houden kan dit een dynamisch te bepalen gegevensverzameling zijn. Services die werken volgens dit mechanisme staan ook wel bekend als *multi-grained services*. Deze richtlijn is vooral relevant voor services die gegevens raadplegen.

»De kwaliteit van een service oriented architecture staat of valt met de kwaliteit van de services«

Domeinspecifieke standaarden

Domeinspecifieke standaarden bieden standaard proces- of gegevensmodellen voor zowel horizontale (personeel, financieel) als verticale domeinen (verzekeren, telecommunicatie). Door gebruik te maken van domeinspecifieke standaarden voor het definiëren van services en gegevens wordt de betekenis eenduidig gedefinieerd. Ook zal de service stabiel en beter te hergebruiken worden doordat de standaard gedefinieerd is vanuit een bredere optiek dan een specifieke toepassing en doordat deze gebaseerd is op meer kennis en ervaring. Het communiceren in volledige bedrijfsobjecten wordt eenvoudiger door gebruik te maken van een standaardinformatiemodel waarin de granulariteit van bedrijfsobjecten is gestandaardiseerd. Voor veel

domeinen zijn reeds standaardinformatiemodellen beschikbaar of in ontwikkeling. In de praktijk blijkt alleen dat deze nog niet altijd volwassen genoeg zijn of onvoldoende aansluiten bij de bedrijfsvoering. In dat geval kan een externe standaard als basis dienen voor een eigen te ontwikkelen standaard.

Uitbreidbaarheid van services

Er zijn mechanismen en technologieën die het mogelijk maken servicedefinities uit te breiden zonder het bestaande interfacecontract te verbreken (denk aan bijvoorbeeld XML). Door gebruik te maken van deze mechanismen wordt ervoor gezorgd dat een servicedefinitie maximaal hergebruikt kan worden en stabiel is voor bestaande afnemers. De nieuwe servicedefinitie is daarmee dus backwards compatible geworden. Nadeel van XML is de overhead die het kan veroorzaken in de datatransfer, die daarmee de performance kan beïnvloeden. Uitbreidbaarheid is echter ook te

realiseren bij het gebruik van *fixed record*-formaten, door het opnemen van lege velden die naderhand kunnen worden gevuld. Deze richtlijn is goed bruikbaar voor services die gegevens raadplegen. Als toegevoegde gegevens nodig zijn voor verwerking, dient de interfacewijziging adequaat gecommuniceerd te worden. In veel gevallen kunnen oudere versies van de service nog wel worden ondersteund; door het meegeven van een versienummer bij het aanroepen van een service kan de juiste implementatie worden geselecteerd.

Het realiseren van meerdere implementaties van services

In een aantal gevallen kunnen de kwaliteitseisen die afnemers aan een service stellen zodanig extreem zijn of van andere afnemers afwijken, dat het verstandig is om meerdere implemen-

taties van een service-interface te realiseren. Bijvoorbeeld, een zeer strikte performance-eis voor een service die wereldwijd gebruikt wordt kan ertoe leiden dat er meerdere implementaties van dezelfde service nodig zijn die dichter bij de afnemers zijn gesitueerd. Indien er afnemers zijn met sterk verschillende performance-eisen, kan het nodig zijn om twee verschillende implementaties van de service te realiseren. De ene kan voor performance geoptimaliseerd zijn, bijvoorbeeld door van duurdere infrastructuur gebruik te maken. De andere kan geheel op kosten geoptimaliseerd zijn en gebruikmaken van goedkope infrastructuur. Op deze wijze wordt de service geheel toegespitst op de eisen en beperkingen (bijvoorbeeld kosten) van de afnemer en worden de afnemers in wezen van elkaar geïsoleerd. Om wildgroei van services te voorkomen kunnen de verschillende implementaties wel worden verborgen achter een samengestelde service. Door het meegeven van parameters bij het aanroepen van de samengestelde service kan de juiste (snelste, goedkoopste, hoogste kwaliteit, enzovoort) implementatie worden geselecteerd.

Toestandloos

Een toestandloze service is een service die niets van de procescontext onthoudt nadat hij is aangeroepen. Hij is daarmee dus niet afhankelijk van eerdere aanroepen. Applicatiedata wordt opgehaald uit de database of aangeboden via de interface maar wordt niet doorgegeven van aanroep naar aanroep. De belangrijkste motivatie voor het maken van toestandloze services is het vergroten van de performance en schaalbaarheid, dit doordat locks op resources worden voorkomen en afnemers niet op elkaar hoeven te wachten.

Het verbergen van de implementatie

Afnemers zouden geen kennis moeten hebben van de implementatie van een service. Dit betekent het afschermen van de gehanteerde technologie, maar ook van de interne realisatie van een service. Aan de parameters van een service zou je dus niet mogen zien in

welke programmeertaal de service is gerealiseerd of wat het interne datamodel van de service provider is. Merk op dat middleware een belangrijke rol kan spelen bij het verbergen van de implementatie van services. Middleware biedt allerlei mechanismen om applicaties van elkaar te ontkoppelen. Daardoor hoeft de aanroeper van een service geen kennis te hebben van het platform, de taal, het protocol, de locatie, de naam en het gegevensformaat van de aanbieder van de service. Ook kan middleware voor ontkoppeling in tijd zorgen, waarbij een afnemer van een service niet hoeft te wachten op een antwoord. Bij het ontwerp van de afnemer moet hier overigens expliciet rekening mee worden gehouden.

De kwaliteit van services

In het algemeen dient de kwaliteit van de geleverde services te worden bewaakt. Ook hier kan middleware worden ingezet. Zo kan middleware gebruikt worden voor het beïnvloeden en monitoren van de kwaliteit van de dienstverlening. Een afnemer kan via een policy aangeven welke eisen hij stelt aan de dienstverlening, waarna de middleware op zoek kan gaan naar de juiste instantie van de service. Ook kan de middleware met toegevoegde waarde-diensten additionele kwaliteit realiseren, bijvoorbeeld door het redundant uitvoeren van services.

Governance-structuur

Een service oriented architecture vraagt om expliciete organisatorische aandacht en inrichting. Zo dient het beschrijven, publiceren, beschikbaar stellen en hergebruiken van services organisatorisch goed te worden belegd. Ook dient er aandacht te zijn voor strategievorming rondom de architec-

tuur, conceptbewaking en het informeren van gebruikers. Zoals de genoemde richtlijnen aangeven, is het niet eenvoudig om tot goede services te komen zonder daarbij soms concessies te moeten doen. Temeer omdat veel van dergelijke concessies vaak vanuit de optiek van een project plaatsvinden, is het verstandig het beheer van de servicedefinities en de controle daarvan te beleggen bij een separaat organisatieonderdeel. Deze afdeling beoordeelt de servicedefinities op hun kwaliteit en kijkt vooral naar het aspect 'herbruikbaarheid'. Deze afdeling heeft de autoriteit om servicedefinities bij te stellen en ze te veralgemeniseren, zodat zij breder toepasbaar zijn. Het moge duidelijk zijn dat dit niet geldt voor services die bewust als maatwerk zijn gerealiseerd. Voor deze services geldt dat de afnemer grotendeels de servicedefinitie bepaalt. Naast het inrichten van de genoemde afdeling moet vooral het informeren van gebruikers niet worden vergeten. Daarbij moeten de gehanteerde concepten en de daaruit voortvloeiende consequenties goed worden uitgelegd aan zowel serviceafnemers en serviceleveranciers als architecten en ontwerpers van applicaties.

Conclusie

De kwaliteit van een service oriented architecture staat of valt met de kwaliteit van de services. In dit artikel is een overzicht gegeven van de eisen die gesteld zouden moeten worden aan services. Daarnaast zijn richtlijnen gepresenteerd waarmee aan de eisen kan worden voldaan. Aangezien zowel de eisen als de richtlijnen voor een deel conflicteren, is het helder dat er geen volledige automatische vertaling bestaat van bedrijfsdoelen naar services. Het definiëren van goede services is vooral

een kwestie van het toepassen van de juiste richtlijnen en het afwegen van de verschillende krachten die daarbij een rol spelen. Slechte services zullen gewoonweg niet of onvoldoende worden hergebruikt, waardoor de initiële investering niet wordt gerechtvaardigd. De kwaliteit van een service oriented architecture staat of valt daarmee met de kwaliteit van de ontwerpers en het gebruik van de services.

Reviewer Guido van Humbeeck

Links

Nederlands Architectuur Forum: www.naf.nl/
 W3C XML Protocol Working Group:
www.w3.org/2000/xml/Group/
 Web Services Interoperability Organization:
www.ws-i.org/

Danny Greefhorst

is werkzaam als Senior IT Architect bij IBM Business Consulting Services. E-mail: greefhorst@nl.ibm.com.

Jan Miedema

is werkzaam als Informatie Architect en Consultant bij ING. E-mail: jan.miedema@mail.ing.nl.

Hans Rijks MBA

is werkzaam als Informatie Adviseur bij het Algemeen Burgerlijk Pensioenfonds. E-mail: hans.rijks@abp.nl.

Dit artikel is tot stand gekomen naar aanleiding van een discussie in de werkgroep Service Oriented Architecture van het Nederlands Architectuur Forum. De inhoud van dit artikel is dan ook mede gebaseerd op de kennis en ervaring van de leden van de werkgroep, die afkomstig zijn van verschillende bedrijven. De auteurs bedanken de leden dan ook voor hun bijdrage aan en review van het artikel.