e·overheid

BOUW MEE AAN BETERE DIENSTVERLENING

Service-Georienteerde Architectuur

Dossier Architectuur

e·overheid

BOUW MEE AAN BETERE DIENSTVERLENING

Service-Georienteerde Architectuur

Dossier Architectuur

Auteur	Paul Oude Luttighuis
Versie	1.0
Status	Definitief
Den Haag,	01-11-2010

Voorwoord

Voorwoord bij de eerste versie
De al overvolle informatiekundige en informatietechnologische supermarkt van Haarlemmer-oliën en zilveren kogels heeft ondertussen een complete afdeling gekregen met de naam SOA (Service-Oriented Architecture) of — om negatieve associaties in het Nederlands taalgebied te vermijden — SGA (Service-Georiënteerde Architectuur). Is SOA de zoveelste technologie-gedreven hype waarbij ICT-gebruikersorganisaties, en dus ook de e-overheid, eigenlijk beter hun schouders kunnen ophalen?

Dat zou onverstandig zijn, al is het maar omdat vrijwel de gehele softwaremarkt – de multinationale ICT-grootmachten voorop — hierop heeft ingezet. Maar er zijn nog andere en belangrijkere redenen: de ontwikkeling in softwareland passen bij een parallelle ontwikkeling in economieën en overheden, waar de verhoudingen binnen en tussen organisaties zich meer en meer in termen van diensten laten karakteriseren. Daarnaast ruilen softwareapplicaties hun puur instrumentele rol in bedrijfsprocessen in voor een prominentere rol als hoofduitvoerder. SOA is dus niet alleen een technische aangelegenheid; het is een kwestie van organisatie en bedrijfsvoering, zeker in een e-overheid die zich ten doel stelt dienstverlener te zijn voor burger en bedrijf.

Daarom kiest ook de Nederlandse Overheids Referentie Architectuur (NORA) voor SOA. Voortbouwend op deze keuze wil het boekje in uw handen een verdieping geven aan het begrip SOA. Daarbij wil het niet beweren de enig geldige of denkbare invulling te geven van het begrip, maar het vooral zo invullen dat het de e-overheid helpt haar niet geringe ambities sneller en beter te laten realiseren. Kenniscentrum hoopt dat u waarde en genoegen put uit dit boekje en nodigt u uit uw op- en aanmerkingen kenbaar te maken.
Den Haag, augustus 2007

Voorwoord bij de tweede versie
In drie jaar gebeurt veel. Niet zoveel dat service-oriëntatie zijn geldigheid heeft verloren. Sommigen hebben SOA weliswaar dood verklaard, maar dat betrof vooral een smalle technologische opvatting van de term, waaraan dit dossier zich bij voorbaat al onttrok. Wel is de e-overheid voortgeschreden, met nieuwe namen aan het firmament, al dan niet als vervanger van oude. Ook heeft NORA een gewijzigde opzet gekregen. Dat was voldoende reden om een bescheiden actualisering van dit dossier door te voeren.

Wat ook niet is veranderd, is de uitnodiging uw op en aanmerkingen te delen, momenteel met RENOIR, het ICTU-programma waarin in april 2009 het voormalige Kenniscentrum is opgegaan.

Enschede, november 2010

Inhoud

1	**Introductie**	**7**
1.1	Achtergrond en aanleiding	7
1.2	Waar gaat SOA over?	7
1.3	Doelgroep en leeswijzer	10
1.4	Wat staat er niet in dit dossier?	10
2	**Waarom SOA?**	**13**
2.1	Wie kan zonder?	13
2.2	De e-overheid	13
2.3	De valkuil van de metafoor	14
2.4	Eerst afspraken, dan werking	14
2.5	Andere overheden en de software-industrie	15
2.6	Het instrument en de professional	15
3	**SOA als ontwerpstijl**	**17**
3.1	Bril en mal	17
3.1.1	De gegevens-georiënteerde stijl	17
3.1.2	De werkstroom-georiënteerde stijl	18
3.1.3	De service-georiënteerde stijl	19
3.1.4	De stijlen op een rij	21
3.2	Groot en klein	21
3.3	Wegwijs in het dienstenwoud	22
3.4	Principes, patronen en antipatronen	24
3.5	Diensten en ketens	26
3.6	Diensten en gebeurtenissen	28
3.7	Andere definities van SOA	28
3.8	Betekenis en context	32
4	**Publiceren en afspraken in een SOA**	**33**
4.1	De SOA-cyclus	33
4.2	Registers en overeenkomsten	34
4.3	De dienstpublicatie	35
4.4	De dienstverleningsovereenkomst	38
5	**Standaarden voor SOA**	**39**
5.1	Interoperabiliteit	39
5.2	Wegwijs in het standaardenwoud	40
5.3	Generieke standaarden voor SOA	41
5.3.1	Design-time standaarden voor SOA	41

5.3.2 Agreement-time standaarden voor SOA 44
5.3.3 Run-time standaarden voor logistiek en transport voor een SOA 45
5.4 Domein-specifieke standaarden in een SOA 45

6 Servicebussen 49

7 Hoe krijgt de e-overheid een SOA? 53
7.1 Ontwerpstijl 53
7.2 Publiceren en afspraken 54
7.3 Standaardiseren 55
7.4 Servicebussen 56
7.5 Valkuilen 56

8 Afkortingen 59

9 Document informatie 61

1 Introductie

1.1 Achtergrond en aanleiding

NORA is de referentie-architectuur voor de Nederlandse e-overheid[1]. Zij stelt zich tot doel de inhoudelijke samenhang te bevorderen tussen de grote hoeveelheid programma's, projecten en voorzieningen die de e-overheid rijk is en nog gaat krijgen. NORA biedt ontwerp-principes, modellen en afspraken voor het (her)inrichten van de e-overheid, geordend in een raamwerk. Zij maakt daarbij twee architecturale basiskeuzes. Service-georiënteerde Architectuur (SOA) is één van die keuzes.

Hoewel deze keuze op hoofdlijnen door velen wordt gedeeld, spelen er nog allerlei belangrijke vragen over:

- wat SOA precies moet inhouden: vele definities, afbakeningen en modellen doen de ronde, waarvan sommige ook wezenlijk verschillen;
- wat SOA vermag: het is zaak SOA te ontdoen van mythische proporties zonder te vervallen in onderschatting;
- welke stappen de Nederlandse overheid kan zetten om SOA op een passende wijze te omarmen.

Dit dossier wil een stap uitgebreider aandacht besteden aan deze vragen dan NORA zelf kan doen, maar toch ook zelf handzaam blijven.

1.2 Waar gaat SOA over?

Afbakening

De e-overheid is een steeds veranderend stelsel van werkzame onderdelen: organisaties (gerangschikt in sectoren en overheidslagen), afdelingen, applicaties, voorzieningen, infrastructuur en mensen. De werking van deze onderdelen is deels gegoten in software, maar voor een belangrijk deel ook niet. SOA gaat over beide soorten onderdelen[2].

SOA gaat over software én over niet geautomatiseerde bedrijfsprocessen

SOA gaat over de interactie tussen de werkzame onderdelen van de elektronische overheid

De onderdelen van de e-overheid hebben interactie: burgers en bedrijven nemen diensten af van de overheid, organisaties en hun afdelingen werken samen in ketens en netwerken, organisaties en hun mensen gebruiken applicaties, applicaties gebruiken infrastructuur.

[1] http://www.e-overheid.nl/thema/referentiearchitectuur/de NORA/de NORA.xml

[2] Soms wordt om deze reden gesproken over de *Service-Oriented Enterprise*.

SOA gaat over de manier waarop we spreken over deze interactie. SOA doet dat in termen van diensten, van dienstverlening. SOA gaat dus ook over interoperabiliteit in de e-overheid, een hoofddoelstelling van NORA.

SOA gaat over de rand én de binnenkant van de elektronische overheid

SOA gaat dus niet alleen over de randen van de e-overheid, waar diensten worden geleverd aan burger en bedrijf, maar zeker ook over de binnenkant: de manier waarop overheidsorganisaties, hun afdelingen, applicaties en voorzieningen zich tot elkaar verhouden. SOA gaat dus ook over keten- en netwerksamenwerking, over koppelvlakken, over integratie en over standaarden. SOA ziet de e-overheid als een gemeenschap van onderdelen die elkaar diensten verlenen.

SOA gaat over tactische, operationale en technische aspecten van de elektronische overheid

Dienstverlening is niet alleen een operationele en technische aangelegenheid. Het vindt plaats op basis van zakelijke afspraken tussen de werkzame onderdelen van de e-overheid. SOA gaat daarmee ook over tactische aspecten van de e-overheid, namelijk hoe deze afspraken eruit zien en hoe zij tot stand komen.

SOA gaat over subsidiariteit

Een grote uitdaging in de e-overheid is om haar werkzame onderdelen samen te laten werken zonder daarbij de eigen manoeuvreerruimte van de onderdelen te zeer geweld aan te doen. Daarvoor zijn losse koppelingen nodig, maar weer niet zo los dat de samenwerking aan waarde of kwaliteit inboet. Het dienstenbegrip van SOA wil een antwoord zijn op dit dilemma: een dienstenafspraak regelt precies dat wat op het koppelvlak nodig is, maar laat de interne aangelegenheden aan de individuele onderdelen zelf. In NORA wordt deze eis het subsidiariteitsprincipe genoemd.

Elementen

Maar hoe regelt SOA dit allemaal? Op welke elementen moet de e-overheid zich richten om op enig moment te kunnen beweren dat zij een service-georiënteerde architectuur heeft? Dat zijn er vier.

SOA is een ontwerpstijl

SOA gaat over ontwerpen, over architectuur, niet alleen van software, maar ook van bedrijfsprocessen. In de SOA-ontwerpstijl speelt het dienstbegrip de centrale rol. Daarmee onderscheidt deze stijl zich van andere zoals de gegevens-georiënteerde en werkstroom-georiënteerde manier van ontwerpen.

SOA gaat over het publiceren van het dienstenaanbod en over het maken van dienstenafspraken

Zoals gezegd gaat SOA ook over het maken van afspraken tussen de werkzame onderdelen van de e-overheid. Zulke afspraken regelen de inhoud, de uitvoering, de voorwaarden en de kwaliteit van de dienstverlening. Daarbij gaat SOA er bovendien van uit dat het dienstenaanbod binnen een zekere gemeenschap wordt gepubliceerd in een dienstenregister. Kortom: SOA gaat over open en zakelijke architecturen.

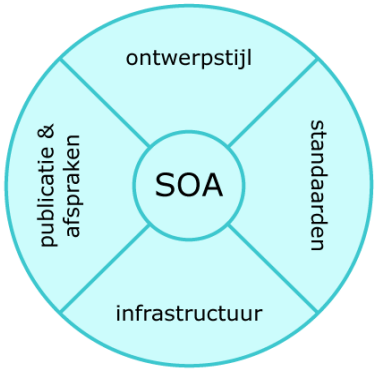
SOA gaat over standaarden

Omdat SOA vooral speelt op koppelvlakken tussen de werkzame delen van de elektronische overheid, is standaardisatie een belangrijk element van SOA. Niet alleen maken service-georiënteerde architecturen gebruik van zekere technische SOA-basisstandaarden, ook het ontwikkelen van gegevensinhoudelijke standaarden vindt service-georiënteerd plaats.

SOA gaat over infrastructurele voorzieningen

Zowel voor het operationele aspect — de feitelijke levering van diensten — als voor het tactische aspect — de publicatie van het dienstenaanbod, maken service-georiënteerde architecturen gebruik van infrastructurele voorzieningen, zoals portalen, servicebussen en dienstenregisters.

Deze vierdeling zal in dit boekje regelmatig terugkeren en visualiseren we daarom in het zogenaamde SOA-wiel.



Het SOA-wiel

Omdat dit dossier service-oriëntatie op deze manier afbakent en positioneert, maakt het geen onderscheid tussen de termen *dienst* en *service*. Het zal steeds de eerste term gebruiken, behalve in eigennamen en in de samenstellingen *service-oriëntatie* en *servicebus*, omdat die gemeengoed zijn.

1.3 Doelgroep en leeswijzer

Dit boekje wil een toegankelijke inleiding in service-georiënteerde architecturen zijn voor iedereen die verantwoordelijkheid draagt voor de processen of dienstverlening, voor informatiemanagers en voor architecten van de informatiehuishouding van de e-overheid.

Voor hen die zich oriënteren op service-georiënteerde architecturen en zoeken naar argumenten, bespreekt hoofdstuk 2 redenen voor het kiezen voor een service-georiënteerde architectuur.

Zij die meer informatie zoeken over één of meer van de elementen van het SOA-wiel, kunnen dat vinden in hoofdstuk 3. Dit hoofdstuk laat zien:

- hoe de service-georiënteerde ontwerpstijl zich onderscheidt van andere stijlen;
- hoe kan worden omgegaan met het beschrijven en publiceren van diensten en het maken van dienstverleningsafspraken;
- welke standaarden bestaan voor SOA's en hoe service-georiënteerde semantische standaardisatie werkt
- welke infrastructuur (servicebussen) gebruikt kan worden voor SOA's.

Ten slotte bespreekt dit hoofdstuk enkele andere typeringen van SOA die wel gehoord worden.

Voor hen die zich oriënteren op de feitelijke invoering van SOA binnen de e-overheid, biedt hoofdstuk 4 aanknopingspunten voor migratie naar een SOA, voor elk van de vier elementen van het SOA-wiel.

1.4 Wat staat er niet in dit dossier?

Het is goed om op deze plaats ook de verwachtingen enigszins te temperen. Dit dossier biedt geen kant-en-klaar handboek voor het ontwikkelen, implementeren, publiceren, kiezen en afspreken van diensten in de e-overheid. Het biedt zelfs geen klant-en-klaar setje van SOA-standaarden die de e-overheid zou moeten hanteren.

Een kant-en-klaar handboek vraagt een veel grotere omvang en veel meer detail dan redelijkerwijs in een dossier als dit zou passen. Bovendien zou het nu al aanbrengen van dit detail kunnen leiden tot een onnodig grote en chaotische discussie in de e-overheid. Liever beperkt dit dossier zich tot een eerste invulling van de door NORA gepresenteerde hoofdlijn. Als deze hoofdlijn gedragen wordt, is een volgende detaileringsstap aan de orde, gebruik makend van reeds beschikbare fragmenten.

Het is bovendien goed om zich rekenschap te geven van het feit dat de internationale standaardisatiewereld er nog niet uit is. Op belangrijke onderdelen bestaan zeker goede en open standaarden, maar vaak ook concurrerende. Bovendien worden sommige aspecten van SOA nog niet door een volwassen standaard afgedekt.

Ten slotte: voor drie onderwerpen geldt dat zij, ook in een SOA, van groot belang zijn, maar in dit dossier toch niet of nauwelijks aan de orde komen. Het gaat om: beveiliging, autorisatie en semantiek. Deze onderwerpen worden wel in het denkraam geplaatst, maar niet uitgewerkt.

2 Waarom SOA?

2.1 Wie kan zonder?

- Herkent u zich in de volgende kenschetsen?
- U hoeft zich geen zorgen te maken over koppelingen met andere organisaties en/of systemen. U bent en blijft de enige meester over uw bedrijfsprocessen en systemen.
- U hoeft zich geen zorgen te maken over toekomstige veranderingen aan bedrijfsprocessen of informatiesystemen. Immers, “Wie dan leeft, die dan zorgt” en “Beheer is voor later”.
- U heeft niets te maken met in- of uitbesteding van bedrijfsprocessen of informatiesystemen.
- U bouwt uw bedrijfsprocessen en systemen vanaf de bodem zelf op, zonder gebruik te maken van functies of componenten van derden.
- Het is niet nodig om scherp te zijn over wat uw bedrijfsproces of informatiesysteem biedt; men komt toch wel naar u toe: u bezit een wettelijke taak.
- U hebt de volledige vrijheid bedrijfsprocessen en zelfs uw klanten aan te passen, waar technologie dat vraagt. Het is geen probleem dat elke nieuwe migratie meer tijd kost.

Als u een speler bent in de e-overheid is de kans klein dat u op meerdere van deze vragen volmondig “ja” kunt antwoorden.

2.2 De e-overheid

De ambities en omstandigheden van die e-overheid laten zich immers juist karakteriseren door:

- klant-gedreven dienstverlening, waarvan geëist wordt[3]:
  - betrouwbaarheid
  - pro-activiteit
  - transparantie
  - toegankelijkheid
  - vindbaarheid
  - noodzakelijkheid
- terugkerende wijzigingen in bedrijfsprocessen door:
  - nieuwe maatschappelijke kwesties, prioriteiten of wetgeving
  - verschuiving van taken en in- en uitbesteding
  - efficiëntie-overwegingen
- terugkerende wijzigingen in softwaresystemen door
  - veranderende wensen en eisen
  - voortschrijdende technologie

[3]Zie hiervoor de tien basisprincipes van NORA. http://www.e-overheid.nl/images/stories/architectuur/nora-katern-strategie.pdf

- vergaande informatiekoppelingen
  - tussen overheid enerzijds en onderneming en burger anderzijds
  - tussen en binnen overheidsorganisaties

Voorwaar een uitdaging voor hen die verantwoordelijk zijn voor inrichting en beheer van bedrijfsprocessen en informatiesystemen. Zij hebben immers te maken met een elektronische overheid, die in haar omvang en complexiteit niet ineens te begrijpen of beschrijven is en waarvan de onderdelen bovendien “op eigen houtje” en in eigen tempo blijven veranderen.

2.3 De valkuil van de metafoor

Wie meebouwt aan de e-overheid, vergelijke haar daarom liever niet met een integraal ontworpen informatiefabriek, een zachtjes snorrend buizenstelsel waardoor informatie stroomt. Dat is een gevaarlijke metafoor omdat:

- informatie geen product is, maar een middel waarmee dienstverlening wordt bedreven — of nog beter, waarmee waarde[4] wordt gecreëerd;
- de e-overheid niet de integrale procesmatige beheersing van stromen kan en mag kennen die wel nodig is in een chemische fabriek; de overheid bestaat uit onderdelen met eigen zeggingskracht;
- niemand het overzicht, in breedte en detail, over het gehele elektronische overheidsstelsel kan hebben, laat staan houden.

Deze metafoor is overigens vaak wel voldoende in overzichtelijker (deel)situaties, waar sprake is van een duidelijke centrale regie en waarin efficiëntie-overwegingen prevaleren boven wendbaarheid in de toekomst.

Waakzaamheid is dus geboden bij beelden van de elektronische overheid, waarin gedacht wordt in termen van stromen (gegevensstromen of werkstromen), van informatie als ware het een product, of in termen van “informatielogistiek”. Zij verraden een “loodgietersblik” op de e-overheid, die essentiële eigenschappen en randvoorwaarden over het hoofd dreigt te zien. Een loodgieter, hoe onmisbaar ook, is geen architect.

2.4 Eerst afspraken, dan werking

Maar wat dan wel? Een passender metafoor moet ruimte bieden aan min of meer autonome spelers binnen de e-overheid, die niettemin steeds een werkend stelsel moeten vormen, hoe veranderlijk ook. Daarom is het belangrijk vooral te kijken naar de uitvoerings*afspraken* tussen deze spelers (organisaties en systemen). In SOA staan juist deze afspraken voorop, niet de werking van de onderdelen. De werking van een organisatie of een informatiesysteem is er slechts om aan de afspraken te voldoen.

[4] Maatschappelijke of private waarde.

Een belangrijke vraag is vervolgens wat er in zo'n afspraak zou moeten staan. Daarbij is een dilemma dat de afspraak:

- strak en compleet genoeg moet zijn om het stelsel te laten functioneren. Het stelsel moet niet ineenstorten, omdat de afspraken waarop het is gebouwd niet rijk genoeg zijn of teveel ruimte laten.
- los genoeg moeten zijn om de spelers flexibel en autonoom te laten blijven, zowel in het inrichten van hun interne werking alsook het koppelen met andere partijen.

Het antwoord op dit dilemma heet in SOA de *dienst* of de *dienstverleningsovereenkomst*. Eigenlijk geeft SOA dus ook een belangrijk antwoord op de vraag hoe de e-overheid het zo gewaardeerde subsidiariteitsbeginsel[5] kan realiseren. Verderop in dit boekje gaan we dieper in op wat er allemaal in zo'n afspraak vermeld staat.

Zo appelleert SOA aan een lonkend perspectief waarin:

- tegelijkertijd zowel de *samenhang* van het hele e-overheidsstelsel als de *beweeglijkheid* van zijn onderdelen wordt gewaarborgd;
- de *transparantie* en *zakelijkheid* van het e-overheidsstelsel wordt bevorderd. Er wordt immers gestuurd op basis van heldere uitvoeringsafspraken en niet op basis van interne werking van een proces of van technologie;
- het *hergebruik* van overheidsdiensten en informatiesystemen wordt bevorderd.

2.5 Andere overheden en de software-industrie

Hierboven hebben we het over de voordelen van SOA gehad, geredeneerd vanuit de belangen van de (elektronische) overheid. Iets minder belangrijk, maar niet te veronachtzamen, is het feit dat in de wijde omgeving van de e-overheid grootschalig wordt ingezet op service-oriëntatie in informatie-architecturen. Dat geldt niet alleen voor de internationale software-industrie, maar ook voor collegaoverheden in andere landen.

2.6 Het instrument en de professional

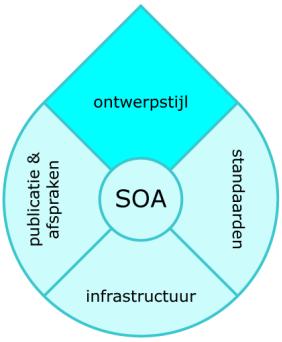
Toch blijft SOA een instrument. En, wil een instrument goed werken, moet het niet alleen naar de letter, maar ook naar de geest worden toegepast. Er bestaan ook "slechte" architecturen, die op de keper beschouwd wel service-georiënteerd zijn. Het is als met muziekinstrumenten. Het is niet moeilijk om met een Stradivarius schandelijk gekrijs voort te brengen, maar voor "goede" muziek zijn wel goede instrumenten nodig. Tussen het instrument en het resultaat staat nog steeds het vakmanschap van de professional.

[5] Referentie naar het betreffende NORA-principe.

3 SOA als ontwerpstijl

3.1 Bril en mal

Menigeen bedient zich — vaak ongemerkt — van een primaire manier van kijken naar de e-overheid, van een voorkeursbril. Dat is een goede zaak: de elektronische overheid laat zich niet via één bril begrijpen en elk perspectief kan belangrijke inzichten met zich meebrengen. Het wordt anders als deze manier van kijken niet alleen maar de bril is, maar ook de mal waarin de e-overheid wordt ontworpen en gerealiseerd. Niet alle mallen presteren namelijk gelijk als het gaat om de kwaliteit, de beheerbaarheid, de efficiëntie, de toekomstvastheid en de transparantie van het hele stelsel. Daarom is het belangrijk stil te staan bij de manier van kijken.



We noemen hier drie ontwerpstijlen. Niet dat er niet meer zijn, of dat er geen mengvormen bestaan, maar deze drie zijn te zien als stereotype ontwerpstijlen, die bovendien aanwijsbaar gangbaar zijn binnen de huidige praktijk van de e-overheid.

3.1.1 De gegevens-georiënteerde stijl

De gegevens-georiënteerde stijl ziet overheidsorganisaties en informatiesystemen vooral als houders van informatie. Registratie en opslag zijn de hoofdfuncties van de onderdelen van de e-overheid. Voor zover deze onderdelen informatie uitwisselen, doen zij dat om elkaar “op de hoogte te brengen” van een wijziging in de (echte of administratieve) werkelijkheid of om elkaar te bevragen.

In deze stijl wordt vaak gewerkt met gegevens-georiënteerde modellen zoals ERD’s en DFD’s. Als er sprake is van berichtenverkeer worden berichtenschema’s vaak rechtstreeks uit gegevensschema’s afgeleid. De bedoeling van de berichten wordt vaak uitgedrukt in termen van CRUD[6] – de vier klassieke database-operaties. Typisch vindt men daar dus bijvoorbeeld lees- of bevragingsberichten en schrijf- of mutatieberichten.

Deze stijl wordt vaak ook impliciet gehanteerd in de wereld van de documentaire informatievoorziening, omdat in deze wereld “het document” het ordenende begrip is. Een document is ook een gegevenseenheid.

[6] Create, Read, Update, Delete.

De gegevens-georiënteerde stijl kan volstaan in omgevingen die zich hoofdzakelijk richten op registratie en opslag, maar voor andere omgevingen is hij niet rijk genoeg. Vooral is het in deze stijl moeilijk om te spreken over de handelingen die op of met de gegevens worden gepleegd. Gegevensmodellen zijn zo goed als statische modellen: zij drukken uit wat er zoal kan *bestaan*, maar niet wat er zoal kan *gebeuren*: de dynamische aspecten.

Dynamische aspecten bepalen grotendeels de context en dus de bedoeling van uitgewisselde gegevens. Omdat de gegevens-georiënteerde stijl deze bedoeling niet goed kan uitdrukken, hebben gegevens-georiënteerd ontworpen berichtenstandaarden de neiging:

- hetzij een grote (moeilijk beheerbare) hoeveelheid verschillende berichttypen te bevatten, namelijk voor elke gebruikscontext één
- hetzij te werken met één groot sjabloonbericht, dat echter weer een te ruim jasje is, waarbinnen iedere toepasser zijn eigen specifieke keuzes kan maken, ten koste van interoperabiliteit.

De puur gegevensgeoriënteerde stijl schiet vooral ook tekort als de activiteiten en processen die op gegevens worden uitgevoerd onderwerp zijn van ontwerp.

3.1.2 De werkstroom-georiënteerde stijl

Juist deze activiteiten en processen waren het onderwerp van een ontwikkeling die zich in de eerste helft van de jaren 1990 begon af te tekenen. De blik richtte zich op de manier waarop uitvoerende activiteiten binnen organisaties kunnen worden gerangschikt en geoptimaliseerd. Als belangrijkste aanleiding daarvoor gold de behoefte om in sterk verzuilde organisaties de activiteiten zodanig over de zuilen heen af te stemmen en te sturen dat de efficiëntie en de prestaties voor de eindklant konden worden verbeterd. De trend die hieruit ontstond en nog steeds leeft heeft verschillende namen: zoals BPR, BPE en BPM.

Oorspronkelijk werd daarbij een ruim procesbegrip gehanteerd: een samenhangende set van activiteiten. Gaandeweg echter maakte deze plaats voor een specifieke vorm van samenhang, namelijk de tijdsvolgorde. Proces werd procedure, of werkstroom. Tegenwoordig wordt nog vaak het woord “proces” gebruikt waar “werkstroom” wordt bedoeld.

In deze stijl spelen werkstroommodellen de hoofdrol; soms heten zij activiteitendiagrammen, zoals in UML. Werkstromen regelen, zoals een software engineer het zou kunnen zeggen, de control flow. Omdat de werkstroom-georiënteerde stijl gezien kan worden als een uitbreiding op de gegevens-georiënteerde, vinden we de gegevens- en documentmodellen vaak terug als een laag onder de werkstroommodellen. Van activiteiten wordt aangegeven op welke gegevens of documenten zij werken.

Als het gaat om gegevensuitwisseling, bedient deze stijl zich van bijvoorbeeld MSD's, waarin de tijdsvolgorde tussen verschillende berichtenuitwisselingen wordt weergegeven, maar ook wel van werkstroommodellen die zich uitstrekken dwars door meerdere organisaties.

Werkstromen dienen in deze gevallen vaak als belangrijke context voor het ontwerpen van berichtenschema's. Zo suggereert de werkstroom bijvoorbeeld vaak welke berichten al voorafgingen of wat een ontvangende partij gaat doen met de inhoud van het bericht. Werkstromen zijn bij uitstek het domein van (operationele) managers; zij beschrijven immers de activiteiten en de sturing daarvan. Zij gaan over uitvoering en over besturing.

Maar daar zit meteen ook de beperking. Een stijl die te sterk op werkstromen leunt, herbergt een aantal gevaren:

- werkstroomverkokering, waarin het steeds inwilligen van werkstroom-specifieke wensen leidt tot beperkt hergebruik en samenwerking tussen werkstromen
- omdraaiing van uitvoering en doel: het precieze doel, effect of resultaat van een werkstroom blijft vaak onbeschreven, terwijl al wel de inrichting van de werkstroom wordt ontworpen
- vermenging van uitvoering en besturing: omdat werkstromen over uitvoering gaan, is besturing vaak impliciet. Vaak wordt centrale sturing van de hele werkstroom geïmpliceerd. Werkstromen missen een handvat voor afbakening van uitvoeringsverantwoordelijkheid.

De werkstroom-georiënteerde stijl deelt met de gegevens-georiënteerde stijl dat zij een loodgietersblik hanteert. Daarom ook zijn ze relatief eenvoudig te combineren.

3.1.3 De service-georiënteerde stijl

Architectenland voert met enige regelmaat de discussie wat belangrijker is: gegevens of processen. Het antwoord van service-georiënteerde architecturen is: geen van beide of — positiever geformuleerd — allebei. Noch gegevens, noch werkstromen kunnen standaard als startpunt van ontwerp worden gekozen, om redenen die hierboven al zijn genoemd.

Maar wat dan wel? Bij het zoeken naar een antwoord op deze vraag komen we terecht bij een aantal klassieke ontwerpuitdagingen die professionals al jaren bezighouden en waar in de loop der jaren steeds wisselende antwoorden op zijn gegeven[7]:

- Hoe kunnen we ontwerpen zo opdelen in onderdelen dat we de verantwoordelijkheden voor de onderdelen kunnen verdelen, zonder dat men onderling te afhankelijk wordt?
- Hoe kunnen we eenheden van ontwerp kiezen die hergebruik eenvoudiger maken?

[7] Zoals object-georiënteerd ontwerpen of component-gebaseerd ontwerpen.

Dit zijn niet alleen technische ontwerpuitdagingen, het zijn ook relevante vragen voor bedrijfsprocessen, vanuit bedrijfskundig of bedrijfseconomisch perspectief. Immers, we zouden de twee vragen ook als volgt kunnen stellen:

- Hoe kunnen we taken verdelen tussen verschillende, meer of minder autonome partijen, zonder de samenhang in het totale proces gevaar loopt?
- Hoe kunnen we de taak van een partij zo kiezen, dat zoveel mogelijk andere partijen daarvan gebruik kunnen maken?

In service-georiënteerde architecturen is deze “eenheid van ontwerp” de dienst. Over de exacte definitie van dit begrip is men het nog steeds niet eens[8]. Wij zullen diensten zien als software- of procescomponenten die:

- een afgebakende en precies omschreven verantwoordelijkheid hebben om een prestatie voor hun omgeving te leveren;
- daarvoor een expliciete scheiding aanbrengen tussen hun zichtbare deel (het koppelvlak) en hun verborgen deel (de implementatie); zowel het koppelvlak als de binnenkant kent een gegevens- en een werkstroomaspect.
- voor het waarmaken van hun verantwoordelijkheid — het leveren van hun dienst — op hun beurt gebruik mogen maken van andere diensten; diensten kunnen dus worden samengesteld tot nieuwe diensten.

Bij diensten horen twee rollen: aanbieders en afnemers. Je zou kunnen zeggen dat de aanbieder de “eigenaar” is, maar dat is niet in de geest van service-georiënteerd denken. Immers, een dienst is geen eigendom, maar een verantwoordelijkheid. Service-georiënteerd denken is denken van buiten naar binnen, niet van binnen naar buiten.

Voorbeelden van diensten zijn:

- een dienst om uit een adres een postcode uit te rekenen;
- een dienst om een vergunningsaanvraag af te handelen;
- een dienst om een verhuizing in de GBA te verwerken;
- een dienst om het laatst genoten jaarinkomen van een werknemer te raadplegen;
- een dienst om een verdachte aan te houden.

Uit dat laatste voorbeeld blijkt dat diensten zich niet hoeven te beperken tot strikt informatiekundige handelingen.

[8] Voor een definitie van OASIS, zie http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm. Voor een definitie van The Open Group, zie http://opengroup.org/projects/soa/doc.tpl?gdid=10632.

3.1.4 De stijlen op een rij

Onderstaande tabel karakteriseert de drie stijlen in kernwoorden.

Stijl Eigenschap	Gegevens-georiënteerd	Werkstroom-georiënteerd	Service-georiënteerd
Aspect	Informatie	Uitvoering	Verantwoordelijkheid
Hoofdrelatie		Tijdsvolgorde	Functioneel
Probleemsoort	Registratie	Sturing	Dienstverlening
Werkwoord	Weten	Doen	Bieden

Overigens gebruiken we hier karikaturen van de drie stijlen om hun verschil duidelijk te maken. In de werkelijkheid komen genuanceerdere versie van de stijlen voor. Bovendien zijn deze stijlen geen rechtstreekse concurrenten. Eerder dragen zij elk hun steentje bij in een gebalanceerde ontwerpbenadering. Niettemin is de principiële volgorde: dienst vóór werkstroom, werkstroom vóór data[9].

3.2 Groot en klein

Ook al willen service-georiënteerde architecturen een antwoord geven op vragen van herbruikbaarheid en soepele taakverdeling, het enkel aanreiken van het dienst-concept is daarvoor slechts een beperkte stap. Dat biedt immers alleen de denkvorm, niet de inhoud. Een belangrijke vraag daarbij is hoe “groot” een dienst moet zijn[10]. Daarop is geen vast antwoord te geven. Het hangt helemaal af van de wensen van de aanbieders en de afnemers. Wel willen we hier drie manieren noemen om van kleine diensten grotere te maken.

Samenstellen. Hiervan is sprake als een aanbieder van een dienst gebruik maakt van onderliggende deeldiensten, in beginsel onzichtbaar voor de afnemer. Een voorbeeld van een samengestelde dienst is als een vergunningsaanvraag door één partij wordt afgehandeld die daartoe op de achtergrond stapsgewijs collega-organisaties of voorzieningen raadpleegt of aanspreekt.

[9] Verderop zal blijken dat ook dienst niet het hoogste is in deze hiërarchie: semantiek (betekenis) is van een nog hogere orde.

[10] Dit probleem heet ook wel *granulariteit* (korreligheid). Grote granulariteit betekent kleine diensten en omgekeerd.

Bundelen. Hiervan is sprake als een gebundeld pakket diensten ineens wordt aangeboden in bijvoorbeeld:

- een contract (op basis van belangen- of waardebalans tussen de spelers)
- een applicatie (op basis van inhoudelijke of functionele samenhang)
- een portaal (op basis van gebruiker of gebruikersgroep).

Een voorbeeld van gebundelde diensten is als naast de vergunningsafhandeling ook inzicht in de voortgangsstatus als dienst wordt aangeboden[11].

Generaliseren. Hiervan is sprake als een ingebakken aspect van een dienst wordt losgelaten en vrij gekozen mag worden. Een voorbeeld van een gegeneraliseerde dienst is als specifieke vergunningsdiensten worden samengenomen in een generieke, zoals bij de omgevingsvergunning. Maar ook personalisatie is een mooi voorbeeld. Een gepersonaliseerde dienst is een specifiek toegesneden versie, op basis van een afnemersprofiel, van een generiekere dienst.

Samengestelde en samengenomen diensten zijn over het algemeen specifieker en dus minder herbruikbaar dan “enkelvoudige”, maar ze komen wel meer tegemoet aan de uiteindelijke wensen van de afnemers. Gegeneraliseerde diensten zijn meer herbruikbaar dan specifieke, maar de afnemer moet in het algemeen meer werk zelf doen. Bovendien kan het zijn dat niet elke afnemer in staat mag worden gesteld de dienst in zijn volle genericiteit af te nemen, bijvoorbeeld als deze daardoor zou kunnen gaan beschikken over teveel informatie.

Dit illustreert de continue balans die moet worden gevonden tussen de maat van afnemer en de maat van de aanbieder als het gaat om de vraag hoe groot een dienst moet zijn. Granulariteit en herbruikbaarheid van diensten is geen uitkomst van een waardevrije ontwerpkundige exercitie, maar van een spel tussen belangen, kosten, waarde en — toch ook — architectureel vakmanschap. Dat spel wordt alleen maar ingewikkelder als er sprake is van een dienst die door velen gebruikt moet kunnen worden.

3.3 Wegwijs in het dienstenwoud

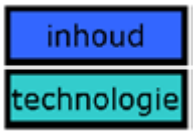
Maar hoe moeten dan de juiste diensten worden geïdentificeerd, met de juiste eigenschappen en de juiste maatvoering? Op het gevaar af de lezer opnieuw teleur te stellen: ook hiervoor bestaat geen vast antwoord. Toch zijn er wel handvatten te geven. Het is mogelijk om op een aantal manieren diensten te classificeren en te groeperen.

[11] Bij enkelvoudige diensten is sprake van een asymmetrie in rollen: er is een aanbieder en een afnemer van de dienst. Bij samengenomen diensten kan dit onderscheid verdampen, omdat binnen hetzelfde dienstenpakket de rol kan wisselen. Zo is het goed mogelijk in een enkel contract twee diensten samen te nemen, waarbij de aanbieder van de ene dienst de afnemer van de andere is en omgekeerd.

Dat kan bijvoorbeeld op omvang: van groot naar klein of van grof naar fijn. Dat is echter een vormonderscheid dat weinig concrete houvast biedt. Een dergelijke classificatie suggereert een gecontroleerde functionele decompositie (top-down) die zelden haalbaar en nog minder vaak wenselijk is. Service-georiënteerde architecturen willen ook bottom-up en middle-out benaderingen mogelijk maken.

Daarom zijn inhoudelijke classificaties van meer nut. Bekend zijn de volgende manieren van onderscheid.

Tussen inhoud en technologie. Inhoudelijke diensten hebben inhoudelijke betekenis voor de gebruiker of de business, terwijl technologische diensten generieke “kale” basisprestaties leveren, met een technologisch platform of infrastructuur. De verhouding is een gebruiksrelatie: inhoudelijke diensten gebruiken technologiediensten. De visualisatie van dit onderscheid is meestal gelaagd.



Tussen voorkant en achterkant. Dit onderscheid is soms terug te vinden onder te termen *front-end* en *back-end* of *front-office* en *back-office*. Voorkantdiensten zijn afgebakend, gedefinieerd en georganiseerd rondom de afnemers en hun vragen. Achterkantdiensten zijn afgebakend, gedefinieerd en georganiseerd rondom inhoudelijke thema’s of specialismen. Ook hier is de verhouding tussen de twee idealiter een gebruiksrelatie: voorkantdiensten gebruiken achterkantdiensten. De visualisatie van dit onderscheid is meestal geschakeld.

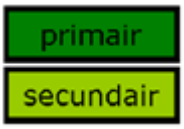
Tussen besturende en bestuurde diensten. Besturende diensten beïnvloeden of bepalen de werking van bestuurde diensten. Let wel, het gaat hier om een besturingsrelatie, niet om een gebruiksrelatie[12]. De visualisatie van dit onderscheid is meestal gelaagd.

Tussen presentatie, logica en opslag. Deze rangschikking heet in het Engels een tier architecture. In de presentatieschakel zitten die diensten die de andere twee soorten diensten presenteert aan de afnemer. Bij een menselijke gebruiker gaat het om user interfacediensten, bij een andere applicatie of een bedrijfsproces, gaat het om diensten die men typisch in een adapter terugvindt.

[12]Dit is vooral van belang bij zogenaamde orkestratie van diensten, waarbij de orkestrator vaak als regisseur wordt gezien van de gebruikte diensten. Dat is niet zomaar terecht. De gebruikte diensten kunnen zich immers ook elders aanbieden. Tussen afnemer en aanbieder van een dienst bestaat niet een (bij voorbaat) hiërarchische relatie. Pas als breder gebruik wordt verhinderd door de orkestrator is er wel sprake van een besturingsrelatie.

In de logicaschakel zijn de informatie be- en verwerkende diensten ondergebracht (in termen van bijvoorbeeld bedrijfsprocessen of business rules), die op hun beurt gebruik maken van de opslagdiensten van de derde schakel. De visualisatie van dit onderscheid is meestal geschakeld.

Tussen primaire en secundaire diensten. Binnen een organisatie of een domein dragen weliswaar alle diensten uiteindelijk bij aan de doelstellingen, maar niet allemaal even direct. Primaire diensten zijn die diensten die het meest direct bijdragen aan de organisatiedoelstellingen. Daaronder vallen in elk geval de diensten aan de primaire klanten van de organisatie. Secundaire diensten doen dat minder direct. Soms worden ook nog tertiaire diensten onderscheiden. De relatie tussen deze soorten is veelal een gebruiksrelatie. De visualisatie is meestal gelaagd.



Combinaties. Vaak worden deze vormen van onderscheid in wisselende combinaties, en menigmaal ook gemengd, gebruikt in referentie-architecturen[13],[14]. De indelingen staan echter grotendeels haaks op elkaar, hoewel:
technologiediensten meestal secundair of zelfs tertiair zijn;
besturende diensten zich vaak aan het verschil tussen voor- en achterkant onttrekken.

3.4 Principes, patronen en antipatronen

Het is al gezegd: het dienst-concept biedt alleen de denk- en ontwerpvorm. Het is aan de architect of ontwerper om diensten te kiezen, op basis van belangen, behoefte, randvoorwaarden en vakmanschap. Deze vrijheid is belangrijk, maar kan ook misbruikt worden. Er zit ruimte tussen de letter en de geest van service-oriëntatie. In sommige gevallen wordt die ruimte gebruikt om andere ontwerpstijlen in een diensten-jasje te gieten, zonder de dienstverleningsgeest aan te houden: window dressing.

Daarom wordt er onder architecten wordt gesproken over ontwerppatronen en –antipatronen voor diensten. Deze patronen en antipatronen vormen de “do’s en don’ts” van ontwerp. Een bekend antipatroon is de “überdienst”, de dienst “die alles kan” en waarvan de specificatie grofweg luidt: vraag wat u wilt en de dienst geeft u wat u vroeg. Een zo generieke dienst is natuurlijk niet te bouwen. Onder zo’n “überdienst” gaat vaak functionaliteit schuil die zich niet wil openbaren, zodat de afnemer niet precies weet wat hij krijgt. En dat zondigt tegen de geest van service-oriëntatie.

[13] Een bekend voorbeeld van vermenging is het mid-officeconcept dat gangbaar is in gemeenteland. Technologiediensten maken vaak een belangrijk deel uit van de mid-office, maar vaak ook inhoudelijke voorkantdiensten.

[14] Ook wordt vaak opslag als een achterkantdienst gezien. Dit is onterecht. Opslagdiensten kunnen zowel aan de voor- als aan de achterkant bruikbaar zijn.

We willen hier geen lijst van patronen en antipatronen aan de orde stellen, maar wel een aantal principes noemen waarin de geest van service-oriëntatie wordt uitgedrukt:

- Diensten hebben precies één aanbieder en een of meerdere afnemers. Er is geen a priori hiërarchische relatie tussen deze twee rollen.
- De aanbieder kiest er zelf voor een zekere dienst aan te bieden aan een zekere groep van mogelijke afnemers. Daarin kan hij wel gestuurd of beïnvloed worden[15].
- De afnemer kiest er zelf voor een zekere dienst af te nemen van een zekere aanbieder. Daarin kan hij wel gestuurd of beïnvloed worden[16].
- Service-oriëntatie plaatst waar mogelijk afnemerssturing boven aanbiederssturing, maar dat is geen wet van Meden en Perzen.
- Diensten kennen geen eigenaars, maar aanbieders. Een eventueel besluit diensten te monopoliseren of de “dienstenmarkt te reguleren” is een separate beslissing. De aanbieder van een dienst is wel eigenaar van het proces dat die dienst realiseert, maar niet per se van de onderliggende diensten die daarvoor worden gebruikt.
- Afgeleid hiervan geldt dat publieke organisaties private diensten mogen gebruiken en omgekeerd, tenzij dit anderszins wordt bepaald.
- Diensten scheiden verantwoordelijkheid en uitvoering. De aanbieder is en blijft verantwoordelijk voor de levering. Uitvoering kan geheel of op onderdelen worden uitbesteed. Verantwoordelijkheid gaat gepaard met de vrijheid en het mandaat voor de aanbieder om de dienst ook feitelijk te kunnen realiseren en daarvoor waar nodig of handig gebruik te maken van andere diensten.
- Diensten scheiden buitenkant en binnenkant. De afnemer weet precies wat hij krijgt of kan verwachten, maar bemoeit zich niet met hoe de dienst gerealiseerd wordt. Dat is aan de aanbieder.
- Diensten zijn maximaal voorspelbaar voor de afnemer. Wat niet kan of mag wordt uitgesloten. Wat wel kan en mag wordt gespecificeerd.
- De afnemer van een dienst kent de verantwoordelijke en kan deze op deze verantwoordelijkheid in voorkomend geval altijd aanspreken.

Deze principes kunnen de indruk wekken dat het hier niet over software gaat, maar over organisaties, sturing en bedrijfsvoering. Dat is niet terecht: het gaat over allebei en dus ook over software diensten. Dat is één van de belangrijke eigenschappen van service-oriëntatie: het brengt deze twee bij elkaar en het brengt technologie onder de regie van organisatorische principes.

[15] Door de afnemer of anderen.

[16] Door de aanbieder of anderen.

3.5 Diensten en ketens

Zoals gezegd, een SOA gaat ook over de zakelijke informatiekoppeling van verschillende organisaties en dus over ketens en netwerken van organisaties. Daarbij wordt met het woord keten vaak eigenlijk niet een vast geschakelde reeks organisaties bedoeld, maar een specifieke werkstroom die deze reeks opspant; organisaties kunnen dus deelnemen in meerdere ketens.

Vaak domineert dus het werkstroomdenken de architectuuroverwegingen in ketens, waarbij op de koppelvlakken gedacht wordt in termen van overdracht: de keten als estafette. In een SOA is een keten echter niet om te beginnen een werkstroomketen, maar een dienstenketen. Elke schakel in een keten is er niet primair om een deelproces uit te voeren, maar om toegevoegde waarde te leveren. Deze toegevoegde waarde, en niet zijn activiteit, definieert de rol van elke schakel en wordt vastgelegd in dienstverleningscontracten. Het is vervolgens aan elke schakel zelf om hiervoor interne processen in te richten.

Toch is het lang niet altijd zo dat elke volgende schakel een dienst aanbiedt aan of afneemt van de vorige. Hier speelt het onderscheid tussen verantwoordelijkheid en uitvoering een grote rol. Om te beginnen moet er een reden zijn om de organisaties in een keten te schakelen. Die reden is dat de keten als geheel een dienst verleent aan zijn klanten. Als daarvan geen sprake is, is er geen reden om het geheel als keten te beschouwen. Geheel in de service-georiënteerde stijl betekent dit dat er één verantwoordelijke aanbiedende partij moet zijn.

Deze partij kan de uitvoering zelf ter hand nemen, maar ook uitbesteden, geheel of deels. Het kan zelfs zo zijn dat al het operationele contact met de afnemer (zelfs bijvoorbeeld de intake of de oplevering) wordt uitbesteed, eventueel zelfs aan verschillende partijen. Vanuit de afnemer echter is deze uitbesteding liefst onzichtbaar[17]; alle benodigde coördinatie is een interne zaak van de verantwoordelijke aanbieder. Hoe dan ook, die eindverantwoordelijkheid zelf kan sowieso niet worden uitbesteed[18]. Voor coördinatie in een keten zijn twee modellen, die ook gemengd kunnen worden toegepast: operationele sturing en tactische sturing.

Bij operationele sturing zit de aanbieder zelf in de operationele uitvoeringsketen. Hij bewaakt en regisseert de inhoud en volgorde van de bijdragen van alle schakels. De schakels behoeven eigenlijk onderling geen contact. Deze vorm kwam in sectie 3.1.2. al aan de orde onder de noemer “samenstellen van diensten” en heet ook wel orkestratie, vanwege de gelijkenis met een dirigent die bij de uitvoering van een muziekstuk continu de coördinatie voert.

---

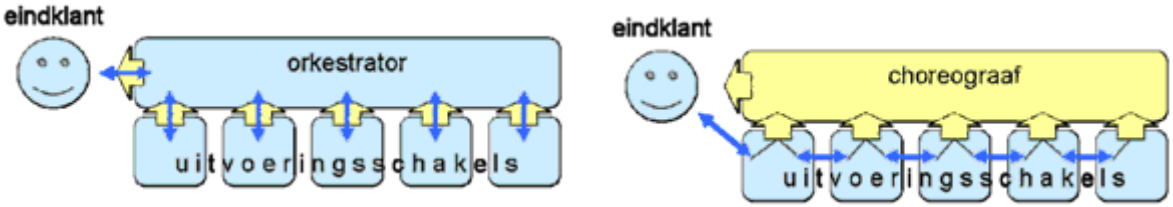
[17] Bijvoorbeeld doordat de uitvoerende partij in dezen namens de aanbieder optreedt (machtiging).

[18] Een bekend voorbeeld hiervan buiten het overheidsdomein is het transportbedrijf dat een bestelling thuis komt afleveren, in opdracht van de winkel waar de bestelling is geplaatst. De uitvoering (en zelfs een deel van het contact met de afnemer) is hier uitbesteed, maar de verantwoordelijkheid blijft integraal bij de winkel.

Bij tactische sturing besteedt de verantwoordelijke aanbieder de contacten tussen de schakels uit aan de schakels zelf. Dat heet choreografie, vanwege de gelijkenis met een choreograaf die de interactie tussen dansers weliswaar vormgeeft, maar niet deelneemt aan de dans zelf.

In beide gevallen is de eindverantwoordelijke aanbieder de eigenaar van het coördinatieproces. In SOA bestaat er niet zoiets als een proces dat zich dwars door meerdere partijen voltrekt: zo'n proces wordt altijd opgehakt in de individuele bijdragen (diensten) van de schakels. In een keten is dan altijd alleen aandacht voor de diensten en de koppelvlakken.

De figuur hieronder laat dit schematisch zien. De lichtgele pijlen staan voor (tactische) dienstverleningsrelaties, de lichtblauwe tweezijdige pijlen voor uitvoeringsrelaties. De uitvoeringsrelaties staan in beide gevallen beschreven in de dienstverleningscontracten. Vaak kan choreografie in een keten worden vergeleken met een estafette, waarin onderweg tijdens de uitvoering een stokje worden overgedragen tussen de uitvoerende schakels. Belangrijk is echter wel dat er een expliciete choreograaf is, met wie elk der uitvoerende schakels een dienstverleningscontract heeft. In die overeenkomst staat allereerst de dienst van die schakel beschreven, die voor de toegevoegde waarde van die schakel in de keten staat. Daarnaast staan ook de overdrachten met de aanpalende ketenschakels beschreven.



Choreografie is vaak favoriet boven orkestratie, omdat er geen regisseur is die zich tijdens de uitvoering met zaken bemoeit. Wat echter vaak wordt vergeten is dat ook choreografie gestoeld is op expliciete en precieze dienstverleningsovereenkomsten. Zonder dergelijke overeenkomsten is er geen sprake van een SOA, maar eerder van een werkstroomgerichte benadering. Bovendien moet er altijd een expliciete partij zijn die de verantwoordelijkheid neemt voor de einddienst. In een zuivere choreografie is dat de choreograaf. In de keuze tussen orkestratie en choreografie is dus steeds de dienst aan de eindklant van de keten het uitgangspunt. Het is niet zo dat muzikanten volgzamer zijn dan dansers; het is eenvoudig zo dat (in menige muziekvorm) de kwaliteit van het muziekstuk zou lijden onder een te losse (tactische) coördinatie. Dat geldt minder voor een dansvoorstelling. Om deze reden zou het onterecht zijn a priori voor choreografie te kiezen binnen de e-overheid, enkel en alleen om de autonomie van bestaande organisaties intact te laten.

Vooral de contacten met de eindklant zijn belangrijk om goed te coördineren, hetzij door ze als eindverantwoordelijke zelf uit te voeren, hetzij door ze operationeel aan te sturen of desnoods door ze scherp tactisch aan te sturen. Als dat niet gebeurt, doemen het kastje en de muur op voor de eindklant.

3.6 Diensten en gebeurtenissen

Gebeurtenissen (events) spelen een belangrijke rol in service-georiënteerde ontwerpstijlen, maar ook in werkstroom-georiënteerde. Zij vormen in zekere zin een brug tussen deze twee. Neem als voorbeeld de gebeurtenis “diefstal” en de dienst “aangifte van diefstal”. Voor de dienst “aangifte van diefstal” is de gebeurtenis “diefstal” om twee redenen relevant:

- Allereerst is hier de gebeurtenis het *onderwerp* van de dienst. Dus, de gebeurtenis staat gemodelleerd in het semantische domeinmodel dat bij de dienst hoort.
- Daarnaast is de diefstal in zekere zin de *trigger* van de dienst, of preciezer gezegd, van de dialoog van de dienst en daarmee ook van het interne proces van de aanbieder van de dienst.

Soms wordt naast voornoemde stijlen ook een “gebeurtenis-gerichte” stijl onderkend, waarin het waarnemen en doorgeven van gebeurtenissen een sleutelrol speelt. In zo’n zogeheten event-driven architecture (EDA) geven spelers elkaar events door, door middel van berichten, en veelal met behulp van een publish-subscribe-mechanisme.

Publish-subscribe kan worden gezien als een specifieke soort van dienst, waarin de afnemer — de “abonnee” — aan de aanbieder — de “uitgever” — meedeelt in welke gevallen (dus bij welke gebeurtenissen) hij welke gegevens wenst te ontvangen. Toch is een event-driven architecture nog niet per se een service-georiënteerde. Daarvoor moet namelijk niet alleen de uitgever zijn publish-subscribe-dienst aanbieden, maar moet de abonnee ook een dienstverleningsrelatie hebben met zijn klant of klanten.

Niettemin kan een event-driven architecture gezien worden als een tussenvorm tussen de werkstroom-georiënteerde stijl en de service-georiënteerde. De winst ten opzichte van de werkstroomgedreven stijl is namelijk dat:

- lange werkstroomslierten zijn afgebroken tot individuele gebeurtenissen
- deze gebeurtenissen niet bij voorbaat aan een actor (een uitvoerder) zijn toegewezen. Gebeurtenissen worden immers niet uitgevoerd, zij treden op. In het publish-subscribe-model neemt de uitgever het optreden van een gebeurtenis waar, maar deze gebeurtenis treedt meestal buiten de uitgever op.

3.7 Andere definities van SOA

Het dossier in uw handen kiest een vrij brede opvatting van SOA. Dat doet het bewust om te voorkomen dat SOA het zoveelste technische concept wordt in de al zeer onoverzichtelijke wereld van technologieën en drieletterafkortingen.

In plaats daarvan wil het dossier:

- SOA de synthese en verrijking laten zijn van bestaande waardevolle benaderingen en
- deze tegelijkertijd richten naar de hoofddoelstelling van de huidige e-overheidsontwikkelingen: het leveren van toegevoegde waarde en dienstverlening in een zakelijk en open arrangement

Toch kan het dossier zich niet veroorloven geen aandacht te besteden aan een aantal andere, al dan niet expliciete, opvattingen en beelden over SOA. Hieronder komen er een aantal aan de orde.

“SOA gaat alleen over software”
De opmars van SOA wordt in de praktijk voor een groot deel gedreven door de ICT-industrie en door concepten en technologieën uit de software engineering. Als boegbeeld geldt daarbij web servicetechnologie.

Deze ontwikkelingen kunnen en zullen zich echter niet kunnen beperken tot software alleen. Dat is vooral omdat software vaak niet meer kan worden gescheiden van bedrijfsvoering. Software is allang niet meer alleen een instrument, maar één van de manieren om bedrijfsvoering te realiseren. In SOA-termen: software levert allang ook businessdiensten.

Daar komt bij dat er ook aan de bedrijfsvoeringszijde een ontwikkeling is naar dienstverlening als samenwerkingsvorm. Denk aan shared dienst centers en, als allerbelangrijkste, de dienstverleningsdoelstelling van de overheid zelf.

Daarom moet SOA niet alleen over software gaan. Let wel, SOA gaat wel degelijk óók over software.

“SOA gaat alleen over informatievoorziening”
Deze lijkt op de vorige. Het komt vaak voor dat informatievoorziening wordt gescheiden van de bedrijfsvoering en daaraan instrumenteel wordt gemaakt[19]. Echter, ook dit onderscheid is aan het verslijten. Voor veel organisaties, zeker ook binnen de overheid, is informatievoorziening een primair bedrijfsproces. Daar is het bijna niet meer mogelijk informatie als een puur productiemiddel te zien: het is de dienst zelf geworden of op zijn minst een wezenlijke component van de dienst. Daarom moet SOA ook niet alleen over informatievoorziening gaan.

[19] Dit is te zien in veel drielaagsraamwerken voor architectuur, waarin achtereenvolgens business, informatievoorziening en technologie aan de orde komen.

“SOA is synoniem met web services”
Hierover spraken we hierboven al. SOA gelijkstellen met web services zou betekenen dat SOA beperkt blijft tot softwaretechnologie. Bovendien zijn er ook andere technologieën dan web services die goede ondersteuning kunnen bieden aan een SOA.

“SOA is vraag-antwoord”
Ook deze opstelling laat zich inspireren door technologie. Het vraag-antwoord-patroon is een dialoogpatroon dat bruikbaar en dominant is in veel softwarefuncties en ook in web servicetechnologie. Echter, diensten laten zich ook met andere dialoogvormen afhandelen, eenstapsdialogen of juist veel langere dialogen. De dialoogvorm is niet de kern van een dienst, maar een onderdeel van de logistiek van de dienst. De kern van een dienst is de toegevoegde waarde die de aanbieder aan de afnemer levert.

In deze lijn wordt SOA ook wel eens tegenover publish-subscribe[20] berichtenverkeer geplaatst. Echter, ook aan een dergelijk mechanisme ligt een dienstverleningscontract ten grondslag. Dat de trigger van het berichtenverkeer niet bij de afnemer ligt is daarbij van tweede orde. Wel is van belang dat de trigger en zijn herkomst in het dienstverleningscontract gespecificeerd staan. Deze trigger kan:
van de afnemer komen, zoals bij request response
intern worden waargenomen door de aanbieder, zoals bij een tijdstrigger
door de aanbieder van elders worden betrokken, bijvoorbeeld door zich te abonneren op een dienst die hem op de hoogte brengt van een dergelijke gebeurtenis.

“SOA is hetzelfde als berichtenverkeer”
Veel informatie-uitwisseling wordt ontworpen in termen van berichten. Het bericht is de eenheid van beschouwing en ontwerp, vaak vanuit een gegevens-gerichte ontwerpstijl. Zoals hiervoor beschreven, nuanceert en verrijkt SOA deze benadering. Berichten zijn niet de kern van het ontwerp. Dat wil echter niet zeggen dat berichten daarmee overbodig zijn geworden. Integendeel, de uitwisseling van berichten is een onmisbaar logistiek middel in een SOA. Het bericht is alleen niet meer het centrum van het ontwerp.

“Als ik een ESB heb, heb ik een SOA”
SOA is geen softwareproduct. Slechts één van de vier spaken in het SOA-wiel duidt wel op softwareproducten. De andere drie zijn minder tastbaar. Echter, een ESB maakt nog geen SOA.

---

[20] Publish-subscribe is een mechanisme waarin een afnemer zekere informatie van een aanbieder ontvangt, telkens als er een bepaalde gebeurtenis optreedt. Deze gebeurtenis kan bijvoorbeeld zijn:
– het verstrijken van een tijdstip (bijvoorbeeld: elke dag)
– het veranderen van een bepaald gegeven
– het optreden van externe gebeurtenis, zoals een verhuizing

“SOA impliceert (centrale) orkestratie”
Hiervoor zij verwezen naar paragraag 3.5. Centrale orkestratie is niet verplicht in een SOA, ook al positioneren aanbieder van BPM- en workflowtechnologie hun producten ook graag in de SOA-trend. Wat een SOA wel vraagt is dat elke dienst, ook als daarbij een hele keten wordt ingeschakeld, een duidelijke eindverantwoordelijke heeft die de keten zodanig coördineert dat de dienst naar behoren en volgens expliciete afspraak wordt geleverd aan de eindklant.

“SOA maakt procesontwerp overbodig”
In sectie 3.1.1. is al aangegeven dat SOA weliswaar een verrijking is van het procesdenken, maar dit zeker niet overbodig maakt. Echter, daar waar werkstromen de dominante ontwerpbenadering zijn, maakt dat wel plaats voor de dienst. Er is echter nog steeds een duidelijke rol voor werkstromen in een SOA, vooral:

- als *intern* middel van een aanbieder van een dienst om die dienst te realiseren, al dan niet gebruik makend van diensten van derden (orkestratie);
- als *dialoog* op een koppelvlak waarmee een dienst wordt afgehandeld.

Het beeld echter van een lange werkstroomketen die zich dwars door meerdere ketenschakels voltrekt past niet in een SOA. Van belang is ook het onderscheid tussen product en dienst (of dienst). De term “product” staat voor het eindresultaat van een proces of werkstroom. De term *dienst* in relatie tot *proces* staat voor de *toegevoegde* waarde, dat wil zeggen, voor het effect of het product van het proces *ten opzichte van de beginsituatie en de input*[21].

“SOA is gelaagdheid”
Ten slotte wordt een SOA wel eens geïdentificeerd met één van de gelaagde of geschakelde modellen uit sectie 3.3. Echter, diensten zijn op vele manieren te categoriseren; vaak ook moeten meerdere categoriseringen naast elkaar worden toegepast. Eén enkele categorisering kan daarmee niet een SOA definiëren.

Een typisch voorbeeld hiervan is de karakterisering van SOA met een plaat waarin “bedrijfsprocessen” op een bovenlaag liggen en “diensten” daaronder. Zulke platen vind men vooral in de BPM- en workflow-wereld. Zoals dit dossier al eerder zei:

- zijn diensten liefst ook van toepassing op de business-laag en
- spelen werkstromen niet alleen de rol van *orkestrator* van onderliggende diensten, maar vooral ook als *middel* voor de uitvoering van bovenliggende diensten

[21] Bijvoorbeeld. Men stelle zich twee ondernemingen voor: een klassieke fietsenhandel en een onderneming die fietsen samenstelt uit door de klant zelf aangeleverde onderdelen. Van deze ondernemingen kan gezegd worden dat zij dezelfde producten leveren, maar niet evenveel toegevoegde waarde. De diensten verschillen duidelijk.

3.8 Betekenis en context

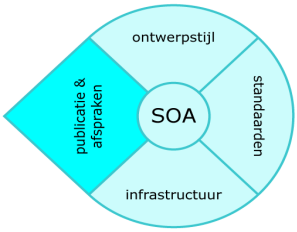
Na lezing van bovenstaande is misschien het beeld ontstaan dat SOA de overkoepeling kan zijn van álle architectuurvraagstukken. Dat is zeker niet zo. Vooral willen we hier noemen dat SOA niet gaat over semantiek, betekenis, bedoeling en context van informatie en diensten. Dat zijn wel degelijk cruciale aspecten in architectuur, waarbij internationale ontwikkelingen zoals Semantic Web en MDA een grote rol spelen. Zonder semantiek is ook een dienst betekenisloos. Hieraan zal dit dossier echter maar beperkt aandacht schenken

Wel heeft het Forum Standaardisatie[22] in 2009 het onderwerp semantische interoperabiliteit voorrang gegeven in haar Interoperabilitetsagenda. Het Forum Standaardisatie onderzoekt een benadering voor semantischeinteroperabiliteit die goed past bij service-oriëntatie.

[22] www.forumstandaardisatie.nl

4 Publiceren en afspraken in een SOA

4.1 De SOA-cyclus



Het is gebruikelijk om in software- en andere ontwerpkringen onderscheid te maken tussen twee globale stadia in de levenscyclus van een systeem (of bedrijfsproces): het ontwikkelstadium en het gebruiksstadium. In het ontwikkelingsstadium wordt functionaliteit gecreëerd of veranderd, in de gebruiksfase wordt deze gebruikt. De stadia kunnen elkaar afwisselen, bijvoorbeeld als tijdens een periode van gebruik nieuwe functionele wensen door gebruikers of afnemers naar voren zijn gebracht, of fouten aan het licht zijn getreden. Als de ontwikkeling betrekking heeft op kleinere of minder ingrijpende wijzigingen wordt dit ook wel onderhoud genoemd.

De term *beheer* kent meerdere interpretaties. Vaak duidt men hiermee op het volgens afspraak operationeel houden van systemen en diensten[23]. Hiervoor wordt ook wel de term *technisch beheer* gebruikt. Andere vormen van beheer, zoals applicatiebeheer maar vooral functioneel beheer[24], liggen eerder in het ontwikkelstadium. SOA voegt een stadium toe tussen het ontwikkel- en het gebruiksstadium. De belangrijkste redenen daarvoor zijn dat:

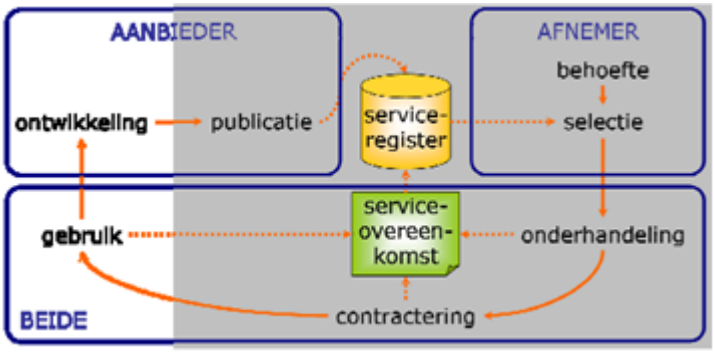
- de ontwikkelde dienst ter beschikking moet worden gesteld aan meerdere (potentiële) gebruikers, die niet bij voorbaat bekend zijn;
- aanbieder en afnemer van de dienst precies moeten zijn in de wederzijdse verwachtingen.

Om dit mogelijk te maken:

- publiceert de aanbieder de dienst, zodat elke (potentiële) gebruiker hiervan kennis kan nemen;
- gaan aanbieder en (elke) afnemer een expliciete gebruiksovereenkomst aan.

[23] Zoals typisch behandeld door de ITIL-methodiek.

[24] Zoals typisch bedoeld door de ASL- en BiSL-methodieken.



De publicatie van een dienst en de dienstverleningsovereenkomst vinden hun neerslag in een document. De dienstpublicatie is eigendom van de aanbieder en gericht aan gemeenschap van alle potentiële gebruikers, door middel van publicatie in een dienstenregister. De dienstverleningsovereenkomst is eigendom van de beide contractanten, die in beginsel de enigen zijn die erover beschikken. Het schema hiernaast laat preciezer zien hoe publicatie en het maken van afspraken (het blok met de grijze achtergrond) in hun werk gaan. De aanbieder publiceert een ontwikkelde dienst in een dienstenregister. Op basis van een gevoelde behoefte zoekt de potentiële afnemer in het register. Met een gevonden passende dienstpublicatie benadert deze vervolgens de aanbieder en onderhandelt daarmee. Dat leidt tot een dienstverleningsovereenkomst, die de basis is voor het gebruik van de dienst.

4.2 Registers en overeenkomsten

De dienstverleningsovereenkomst verwijst idealiter naar de geselecteerde dienst(en) in het register en regelt die zaken die nog niet (eenduidig) in de dienstpublicatie zijn geregeld. In een dienstverleningsovereenkomst kunnen meerdere diensten aan de orde zijn. Zo is het bijvoorbeeld mogelijk om in één overeenkomst twee wederzijdse diensten op te nemen, die als elkaars wederprestatie dienen of om een andere reden alleen in combinatie mogen voorkomen. Hoe meer er in de dienstpublicatie(s) is geregeld, des te minder hoeft nog in de overeenkomst te worden geregeld. In dat geval is er sprake van een standaard overeenkomst. Als de dienstpublicatie veel vrijheidsgraden biedt, die dus nog in de overeenkomst moeten worden ingeperkt, is er sprake van een maatwerkovereenkomst.

Dienstenregisters zijn er in soorten en maten. Het zal niet zo zijn dat er één wereldwijd publiek dienstenregister zal zijn voor alle diensten. Eerder zal elke “gemeenschap van partijen” die onderling diensten verlenen een dienstenregister hanteren. Omdat een partij lid kan zijn van meerdere gemeenschappen, kan het zijn dat een afnemer meerdere dienstenregisters raadpleegt, zelfs voor eenzelfde behoefte. Ook binnen een dienstenregister kunnen overigens concurrerende diensten voorkomen.

4.3 De dienstpublicatie

Onderstaand figuur schetst welke hoofdstukken kunnen voorkomen in een dienstpublicatie.

Naam
De unieke naam van de dienst.

Aanbieder
De naam van de aanbieder van de dienst, alsmede de identificatie van de (elektronische) lokatie waar de dienst afgenomen kan worden (in de vorm van bijvoorbeeld een URL).

Afnemer
Een beschrijving van de mogelijke afnemers van de dienst, hetzij door hen expliciet op te sommen, hetzij door de eisen te beschrijven waaraan een afnemer zou moeten voldoen. Kortom, in dit hoofdstuk wordt de autorisatie geregeld.

Semantiek
Dit onderdeel beschrijft de kerneigenschappen van de dienst zelf en wel vooral:

- het *domein* waarover de dienst gaat, liefst door middel van een verwijzing naar een (semantisch) domeinmodel
- het *effect* of de *prestatie* van de dienst, in termen van
  - de precondities, dat wil zeggen, een precieze beschrijving van de toestand van waaruit de dienst zinvol kan en mag worden aangeroepen
  - een beschrijving van het resultaat en/of de postconditie van de dienst

Een voorbeeld van een preconditie is dat bij een melding van een huwelijk elk der echtelieden niet reeds gehuwd mag zijn. Als een preconditie genoemd staat, is het de verantwoordelijkheid van de afnemer om zich van de geldigheid van de preconditie te vergewissen, alvorens de dienst te gebruiken. Natuurlijk zou ook de aanbieder een dergelijke controle kunnen doen, maar in dat geval is dat opgenomen in de dienst, en dus geen preconditie voor de dienst. Het weigeren van de huwelijksmelding is in dat geval één van de mogelijke resultaten van de dienst geworden.

Maar, de crux van een dienst is gelegen in het resultaat of het effect van de dienst, omdat dat de belichaming is van de waarde die de dienst heeft voor de afnemer. Sommige diensten hebben als enige resultaat dat zij gegevens of een waarde opleveren, zonder zelf iets aan de toestand te veranderen. Typisch zijn dit bevragingsdiensten. Zij hebben dus niet zozeer een effect als wel een resultaat.

Diensten die de toestand wel veranderen heten vaak transacties. Een voorbeeld van een transactie is het verlenen van een vergunning. Na afloop van een dergelijke dienst is er iets wezenlijks veranderd tussen de betrokken partijen. In het specifieke geval van databasediensten heten transacties ook wel mutaties. Eén van de manieren om het effect te beschrijven is om de toestand van na de afronding van de dienst te beschrijven ten opzichte van de toestand ervoor. Dat wordt wel een postconditie genoemd.

Veelal echter zullen diensten zowel een resultaat als een effect hebben, al is het maar dat de afnemer op de hoogte wordt gesteld van het succes of falen van de transactie. Voor een goede beschrijving van precondities, effecten en resultaten, is een goede definitie nodig van de begrippen waarover zij handelen. Een huwelijksdienst zal zich moeten kunnen baseren op een definitie van huwelijk, van echtgenoten, et cetera. Deze zogenaamde semantische beschrijvingen worden liever echter niet in de dienstbeschrijving zelf opgenomen, maar — omdat ze van toepassing zijn op veel meer diensten — separaat vastgelegd.

Op deze plaats is ook een opmerking over zogenaamde doelbinding op haar plaats. Onder doelbinding wordt verstaan dat uitgewisselde gegevens alleen voor een nader gespecificeerd doel mogen worden gebruikt. Omdat in een SOA gegevensverkeer niet op zichzelf staat, maar altijd ten dienste van een specifieke en expliciete dienst, kan desgewenst worden bepaald dat alle in het kader van een dienst uitgewisselde gegevens alleen mogen worden gebruikt in het kader van de afhandeling van deze specifieke dienst. Een dergelijke bepaling kan gelezen worden als: “De dienst heeft bovenbeschreven effect en resultaat en zal daarnaast geen andere effecten of resultaten kennen.”

Syntax

In dit hoofdstuk staan de uitvoeringsaspecten van de dienst beschreven. Daarbij gaat het om:

- een beschrijving van de (deels) elektronische dialoog of dialogen die moeten worden gevolgd om de dienst af te nemen/te leveren;
- een beschrijving van de uitingen (in het algemeen: berichten) die in die dialoog of dialogen worden uitgewisseld;
- de technologische middelen (protocollen en voorzieningen) waarmee de dienst is te gebruiken
- een beschrijving van de benodigde beveiliging of beveiligingsmaatregelen.

Een dialoog kan een éénstaps-dialoog zijn, bestaande uit maar één bericht bijvoorbeeld, maar vaak zijn zij langer of complexer. Een populair dialoogpatroon is request-response. In geval van een raadpleging staat de request voor het bevragingsbericht en de response voor de gevraagde gegevens. In geval van een transactie is de request typisch het verzoek tot de transactie en omvat de response de bevestiging of afwijzing.

Overigens: de dialoog hoeft niet beperkt te blijven tot de dienst-afnemer en –leverancier, oftewel de contractanten. Het is goed mogelijk dat in de dialoog derden zijn betrokken. Dit is typisch het geval in het choreografiemodel, zoals we eerder bespraken. Het is mogelijk dat de dienstbeschrijving op deze uitvoeringsaspecten vrijheidsgraden toestaat, waartussen de afnemer kan kiezen.

Dienstverleningsniveau
Dit onderdeel beschrijft kwaliteitsaspecten van de dienst, zoals:

- inhoudelijke kwaliteit van de dienst;
- dienstverleningstijden (normale tijden, weekends, feestdagen en vakantiedagen);
- beschikbaarheid van de dienst (minimumpercentages, gemiddelde, maximaal aantal verstoringen per periode, meetperiode);
- snelheid, zoals de responstijd.

Dit onderdeel staat het toe af te dingen op het perfecte functioneren van de dienst, omdat 100% perfectie zelden haalbaar is. In dat geval is het maar beter om hierover open te zijn en scherpe afspraken te maken. Mogelijkerwijs worden ook verschillende dienstverleningsniveaus afgesproken, die van toepassing zijn op verschillende afnemersgroepen of verschillende prijzen kennen.

Merk op dat ook inhoudelijke kwaliteitscriteria deel kunnen uitmaken van een beschrijving van het dienstverleningsniveau. Zo kan een aanbieder van een bevragingsdienst precies aangeven wat de geboden (gemiddelde) kwaliteit of actualiteit van de geleverde gegevens is.

Policy
Dit onderdeel beschrijft de voorwaarden die, technisch of anderszins worden gesteld aan de afnemer van de dienst. Een belangrijke component hierin is de autorisatie. Maar het kan ook gaan om het een keuze vrijheid tussen twee onderliggende logistieke protocollen. In het algemeen bevat dit onderdeel de nodige vrijheidsgraden, waartussen een afnemer kan kiezen.

Wederprestatie
Dit is niet zozeer een beschrijving van de dienst zelf, maar van de wederprestaties die de aanbieder verwacht van de afnemer. Een voor de hand liggende wederprestatie is een geldelijke tegenprestatie (een prijs), maar het kan ook zijn dat de aanbieder verwacht dat de afnemer andersom ook een dienst verleent. Van een voorbeeld hiervan is sprake bij afnemers van basisregistratiediensten, die het op zich nemen in voorkomend geval vermoedens van fouten terug te melden.

4.4 De dienstverleningsovereenkomst

De dienstverleningsovereenkomst kent bijna dezelfde structuur als de publicatie. Dat komt doordat zij op te vatten is als een nadere invulling van de dienstpublicatie, als volgt:

- Natuurlijk moet bij de afnemers nu de naam van de specifieke afnemer worden vermeld.
- Bij de uitvoering is door de afnemer een keuze gemaakt voor de specifieke uitvoerinsgvariant, voorzover de dienstpublicatie daarvoor ruimte bood.
- Datzelfde geldt voor de dienstverleningsniveaus.
- Datzelfde geldt voor de wederprestaties.
- Datzelfde geldt voor de policy. Hier hoeft autorisatie niet meer aan bod te komen. Immers, als er een dienstverleningsovereenkomst komt, mag ervan worden uitgegaan dat de afnemer geautoriseerd is. Maar het kan nog wel zinvol zijn om hier bijvoorbeeld de gemaakte keuze uit een aantal technische varianten te noemen, die in de dienstpublicatie werden aangeboden.

Liefst beschrijft de dienstverleningsovereenkomst alleen de aanvullende keuzes die gemaakt zijn ten opzichte van de dienstpublicatie, waarna een verwijzing naar de dienstpublicatie volstaat om het een volledige overeenkomst te maken.

Natuurlijk kan het zijn dat in de onderhandelingen tussen aanbieder en afnemer uiteindelijk wordt afgedongen op de grenzen van wat de dienstpublicatie stelde. In dat geval is dus eigenlijk geen sprake meer van de oorspronkelijke dienst en zou de aanbieder hetzij een specifieke “uitzonderingsdienst” moeten formuleren of de oorspronkelijke dienst moeten “oprekken”.

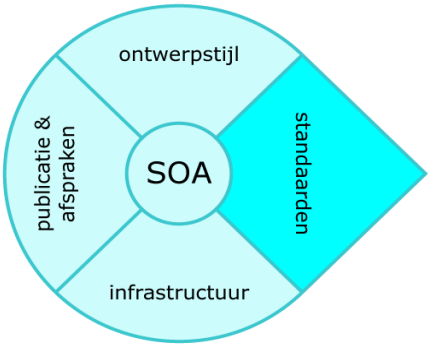
Veel informatie over dergelijke overeenkomsten kan worden gevonden in de literatuur over SLA’s of SNO’s[25]. Een dienstverleningsovereenkomst is een overeenkomst en zal daarom ook allerlei algemene contractparagrafen kunnen kennen, zoals bepalingen rondom wijziging van de overeenkomst, rondom aansprakelijkheid, et cetera.

[25] Zie bijvoorbeeld de checklist op www.ictforyourbusiness.nl/main.asp?ChapterID=1026.

5 Standaarden voor SOA

5.1 Interoperabiliteit

In dit dossier wordt bepleit dat SOA de zakelijkheid, de externe oriëntatie, de openheid en het hergebruik bevordert binnen de e-overheid. Maar, ook met een service-georiënteerde ontwerpstijl en expliciete publicatie en overeenkomsten blijven er nog flinke obstakels over.



Want, wat nu als alle aanbieders volledig maatwerk zouden maken van al hun diensten? Of, als alle diensten hun inhoudelijke begrippen anders zouden definiëren? Of, als alle diensten met andere talen zouden worden beschreven? Of, als alle diensten andere technische protocollen zouden vragen? Dat zou zeker niet bijdragen aan de mate waarin dienst-afnemers en –aanbieders elkaar eenvoudig, snel en goedkoop zouden vinden, tot een vergelijk zouden komen en wederzijds diensten zouden verlenen. Anders gezegd, dat zou afbreuk doen aan de *interoperabiliteit*.

Gelukkig is een groot deel van deze obstakels te bevechten door te standaardiseren, dat wil zeggen, door op allerlei aspecten en onderdelen ervoor te kiezen overeenkomstige behoeftes en problemen op overeenkomstige wijze te lijf te gaan. Dat geldt zeker voor een hele reeks van secundaire aspecten, waarop grote delen van de e-overheid niet wezenlijk zullen verschillen. Voor de primaire aspecten (namelijk de inhoudelijke semantiek en de dienstinhoud zelf, alsmede de inhoud van dienstpublicaties en overeenkomsten) is voorzichtigheid geboden met al te strenge standaardisatie, maar ook daar kan veel worden gedaan, bijvoorbeeld door te werken met sjablonen en patronen.

Kort gezegd:

- zijn dienstenpublicaties principieel *eenzijdige* documenten, waarin een aanbieder van een dienst aan potentiële afnemers duidelijk maakt wat er geboden wordt;
- zijn dienstverleningsovereenkomsten *bilaterale* overeenkomsten tussen een aanbieder van een dienst en één van de afnemers;
- zijn standaarden *multilaterale* afspraken waarin een zekere gemeenschap van partijen bepaalt hoe zij omgaan met een zeker aspect van een dienst.

De relatie tussen deze drie soorten documenten is dat:

- standaarden worden toegepast in (o.a.) dienstenpublicaties;
- dienstpublicaties de inhoudelijke grenzen bepalen van dienstverleningsovereenkomsten.

5.2 Wegwijs in het standaardenwoud

Het is niet mogelijk — laat staan wenselijk — om één grote standaard te maken of te gebruiken voor alle denkbare aspecten van alle denkbare diensten. Daarvoor zitten er teveel verschillen tussen diensten, zowel inhoudelijk als in hun levensfase. Bovendien is de internationale standaardisatiewereld nog lang niet zo volwassen dat er voor elk generiek aspect een goed uitontwikkelde standaard bestaat.

Zonder de standaardisatie-omgeving in haar volle omvang en complexiteit te behandelen, wil dit dossier hier vooral twee dimensies gebruiken om structuur te geven aan standaarden voor SOA.

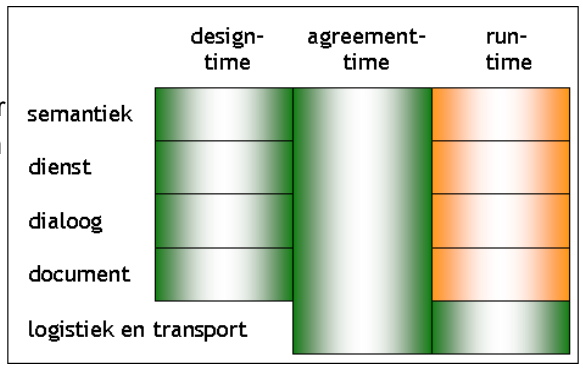
De eerste dimensie (beschreven in 4.1) is aan de orde in drie levensstadia van een dienst: ontwikkeling, publicatie en overeenkomst en gebruik. Ontwerpkundigen zullen hier de drie fasen “design-time”, “agreement-time” en “run-time” herkennen. In elk van deze fasen zijn specifieke standaarden aan de orde.

- In de ontwikkelingsfase (“at design-time”) zijn dat specificatiestandaarden, dat wil zeggen, talen die gebruikt worden om een aspect van een dienst te beschrijven.
- In de publicatie- en overeenkomstfase (“at agreement-time”) zijn dat standaarden voor dienstenregisters, -publicaties en -overeenkomsten.
- In de gebruiksfase (“at run-time”) zijn dat uitwisselstandaarden, dat wil zeggen, standaarden voor aspecten van de diensten, die feitelijk “over de lijn gaan” als de dienst wordt gebruikt.

De tweede dimensie beslaat de verschillende aspecten van een dienst die grotendeels al in 4.3 aan bod kwamen, en wel vooral:

- de semantiek (in termen van een domeinmodel)
- de dienst zelf (preconditie en effect/resultaat)
- de dialoog
- de uitingen/berichten
- de protocollen voor logistiek en transport

Deze twee dimensies spannende structuur op die weergegeven staat in de matrix hieronder. In de rechterkolom staan de uitwisselstandaarden, die bedoeld zijn voor specifieke domeinen of specifieke sets van diensten. Zo behoren hier bijvoorbeeld de uitwisselstandaarden die gebruikt worden voor GBA-diensten, om maar eens wat te noemen, of een specifieke berichtenset voor een ander domein.



In de linkerkolom staan de talen die gebruikt kunnen worden om deze te beschrijven. De rechterkolom is dus domeinspecifiek[26], de linker is generiek[27].

Hierop is één uitzondering: de uitwisselstandaarden voor logistiek en transport. Deze zijn zo generiek, dat een beschrijvingstaal voor hen niet zinvol is. De middelste kolom onttrekt zich grotendeels aan de indeling in vijf aspecten. Het gaat hier immers om generieke standaarden voor dienstenregisters, -publicaties en overeenkomsten die voor de beschrijving van diensten zullen terugvallen op de specificatiestandaarden uit de linkerkolom.

Hieronder bespreekt sectie 5.3 de generieke standaarden voor een SOA en sectie 5.4 de domein-specifieke.

5.3 Generieke standaarden voor SOA

Deze sectie noemt kort een aantal standaarden, beschikbaar of nog in ontwikkeling, die de generieke basis kunnen vormen onder een service-georiënteerde architectuur. Anders gezegd, een organisatie of een groep organisaties die een service-georiënteerde architectuur ambieert, ziet zich gesteld voor de keuze of zij deze standaarden zullen adopteren en welke dan.

Omdat het generieke standaarden zijn, is het niet verstandig op deze aspecten het wiel opnieuw uit te vinden, niet als organisatie, noch als keten of land: de internationale professionele gemeenschap heeft in het algemeen deze uitdaging al aangenomen en zal vaak met betere en duurzamere oplossingen komen.

Toch is het niet makkelijk om een keuze te maken uit de beschikbare standaarden. Dat komt doordat de standaardisatiewereld continu in beweging is, gedreven door technologische ontwikkelingen en strategisch-commerciële drijfveren. Kwaliteit en volwassenheid van standaarden is dan niet altijd duidelijk, noch de mate van ondersteuning in de markt. Ook is concurrentie tussen standaarden eerder regel dan uitzondering. Dit dossier wil deze problematiek echter niet uitgebreid behandelen en beperkt zich tot een inhoudelijke vogelvlucht over de soorten standaarden die hier aan de orde zijn.

5.3.1 Design-time standaarden voor SOA

Bij design-time standaarden voor SOA gaat het om standaardtalen die gebruikt kunnen worden om een aspect van een dienst uit te drukken. Deze talen worden gebruikt door:

- ontwerpers en bouwers van de dienst
- aanbieders van de dienst, voor de dienstpublicatie en dienstverleningsovereenkomst
- afnemers van de dienst, voor de dienstverleningsovereenkomst

[26] Daarom oranje gekleurd.

[27] Daarom groen gekleurd.

Zoals hierboven aangegeven, moeten vier van de vijf gelaagde aspecten van een dienst kunnen worden uitgedrukt in een taal:

- semantiek
- dienst
- dialoog
- bericht

Standaarden hiervoor zijn nog volop in ontwikkeling. Daarbij ontmoeten uiteindelijk drie disciplines elkaar die van oudsher nogal gescheiden zijn geweest:

- de kenniswereld, die zich ten doel stelt “menselijke” kennis te vangen, te ontsluiten en ermee te redeneren;
- de bedrijfssoftwarewereld, die zich ten doel stelt bedrijfsprocessen te ondersteunen of automatiseren met software;
- de softwaretechnologiewereld, die zich ten doel stelt softwareplatformen te ontwikkelen waarop toepassingen (of diensten) kunnen draaien.

Semantiek

Het *domeinmodel* beschrijft die delen en aspecten van de “werkelijke wereld”, waarover een dienst gaat. Het beschrijft het “gespreksonderwerp” van een dienst. Bijvoorbeeld, een huweijksaangiftedienst gaat over een *huwelijk* en over *echtelieden*. De betekenis van huwelijk en echtelieden moet dus voldoende duidelijk zijn om te begrijpen wat de dienst is en wat een afnemer eraan heeft. Bovendien moet deze betekenis worden begrepen voordat er zoiets als een huwelijksbericht kan worden ontworpen. In het overheidsdomein is wetgeving vaak een dominante bron van dit soort semantiek.

Semantische modellen worden vaak conceptuele modellen genoemd. Talen daarvoor onderscheiden zich in de rijkheid van constructies en relaties die zij tussen termen kunnen aanbrengen. Bekende soorten semantische uitdrukkingsmiddelen zijn:

- een lexicon: een lijst van termen;
- een controlled vocabulary: een limitatieve en dus beheerde lijst van toegestane termen;
- een glossary: een verklarende woordenlijst, met per term een definitie, vaak in natuurlijke taal;
- een thesaurus: een verzameling termen waartussen een rijker scala aan relaties is aangebracht, zoals synonymie, antonymie, et cetera;
- een taxonomie: een meestal hiërarchische classificatie van termen, waarbij de “vaderterm” een klasse (categorie) aangeeft en de “kindterm” een deelklasse of instantie;
- een ontologie: een nog rijker semantisch uitdrukkingsmiddel;
- een gegevens- of objectmodel, zoals we die kennen uit de informatiekunde;
- een object-eventmodel, waarin naast statische concepten (objecten) ook dynamische verschijnselen (gebeurtenissen, events) een expliciete plaats hebben.

Semantische uitdrukkingsmiddelen komen meestal uit hetzij de kenniswereld (zoals lexicons, thesauri, glossary's, thesauri, taxonomieën en ontologieën) of de bedrijfssoftwarewereld (gegevensmodellen, objectmodellen, object-eventmodellen), waarin men de inhoud en het gedrag van bedrijfsinformatiesystemen van een betekenisvolle basis wil voorzien. Bekende standaardtalen voor het uitdrukken van domeinmodellen zijn:

- de UML-familie (voor o.a. objectmodellen)
- OWL en Topic Maps(voor ontologieën)

Zoals gezegd, over dit onderwerp is nog veel meer te zeggen, maar dat kan beter belegd worden in een apart dossier over domeinmodellering en semantiek.

Dienst

De lezer zou misschien verwachten dat WSDL hier wordt genoemd als taal om diensten te beschrijven, maar nee! Dat komt doordat WSDL niet de kern van een dienst kan vatten. Die kern is namelijk de prestatie of het effect dat de dienst voor de afnemer bewerkstelligt. In plaats daarvan gaat WSDL vooral over de dialoog; daarmee is WSDL een syntactische standaard, geen semantische. Dat WSDL toch voor "Web Service Description Language" staat, verraadt zijn afkomst: de softwaretechnologiewereld. In deze wereld staat de werking van een systeem vaak eerder in de belangstelling dan het inhoudelijke effect of de betekenis van die werking.

Zowel vanuit de semantische bovenlaag als de syntactische onderlaag wordt gepoogd dit gat te dichten, met bijvoorbeeld OWL-S, respectievelijk het SWSF en WSMO. Deze initiatieven hebben echter nog niet tot volwassen standaarden geleid. Daarnaast kent ook de softwarewereld al lange tijd allerlei manieren om het resultaat of effect van diensten te beschrijven. Bekende formele talen daarvoor zijn Z en VDM. Verder herbergt de UML-familie van standaarden de taal OCL, waarmee predicaten (logische uitspraken) over objecten te doen zijn. Pre- en postcondities zijn van dit soort logische uitspraken.

Samenvattend kan worden gesteld dat de standaardisatiewereld op dit gebied zich nog niet heeft gezet, hoewel het onderwerp al geruime tijd goed begrepen wordt, vooral vanuit de softwarewereld. Het is echter niet gezegd dat daarom maar formele talen zoals Z en VDM zouden moeten worden gekozen. Deze zijn waarschijnlijk te ontoegankelijk voor gebruikers buiten de softwarewereld. Dat ze formeel en dus precies zijn is een prettige eigenschap, maar met natuurlijke taal is ook veel te bereiken. Belangrijke vuistregels daarbij zijn:

- relateer de beschrijving steeds aan een expliciet domeinmodel
- beschrijf het effect of het resultaat zonder te spreken over de interne activiteiten van de dienstenleverancier
- beschrijf het effect of het resultaat zonder te spreken over de dialoog of berichtenstructuren

Dialoog
Met de dialooglaag zijn we bij de syntax van een dienst aangeland. Een dialoogmodel beschrijft de stapsgewijze voortgang van de onderling uitgewisselde “uitingen”, meestal berichten. Bekende grafische uitdrukkingsmiddelen hiervoor zijn MSD’s. Bekende standaarden hiervoor komen ook uit twee belangrijke standaardenfamilies: ebXML en web services. Daar worden dit wel choreografietalen genoemd, omdat een dialoog twee (interne) bedrijfsprocessen aan elkaar koppelt, zonder daarbij een centrale orkestrator te impliceren.

De ebXML-familie biedt hiervoor BPSS. De web service-familie biedt vooral WSCL en WSDL, waarbij WSDL gebruikt wordt voor elementaire dialoogpatronen ─ zoals vraag-antwoord ─ en WSCL voor samengestelde dialogen.

Orkestratietalen, zoals BPEL, blijven hier onbesproken, omdat zij bedoeld zijn voor de interne werking van een dienst, niet voor de externe. Orkestratie wordt gebruikt als een dienstaanbieder, voor het leveren van zijn eigen diensten, andere diensten stapsgewijs aanroept.

Bericht
Van berichten is natuurlijk de structuur van belang. In de XML-familie van standaarden wordt hiervoor vooral XML Schema gebruikt, ooit een opvolger van DTD’s. In XML Schema kan een documenttype worden beschreven, in de boomstructuur die zo typerend is voor XML-documenten.

Met bijvoorbeeld Schematron kunnen vervolgens additionele condities op het XML-document worden beschreven. Bijvoorbeeld, met XML Schema kan worden beschreven hoe de structuur van een XML-document voor een financiële balans eruit ziet. Met Schematron kan daarna worden vastgelegd dat de credit- en de debet-zijde van zo’n balans dezelfde totaalbedragen moeten vertonen. Schematron kan worden gezien als een syntactische tegenhanger van bijvoorbeeld OCL, net zoals XML Schema kan worden gezien als een syntactische tegenhanger van bijvoorbeeld UML-objectmodellen[28].

5.3.2 Agreement-time standaarden voor SOA
Op agreement-time spelen drie soorten standaarden een hoofdrol:
- standaarden voor de structuur van dienstenpublicaties
- standaarden voor de structuur van dienstverleningsovereenkomsten
- standaarden voor de toegang tot dienstenregisters

[28]Schemavalidators kunnen een bericht controleren op structurele juistheid (voldoen aan de structuur), terwijl Schematronvalidators een bericht kunnen controleren op aanvullende voorwaarden. Daarvoor maakt Schematron gebruik van XSL-T, een standaard uit de XML-familie voor het vertalen van het ene XML-document uit de andere.

Een groot deel van de structuur van dienstenpublicaties en -overeenkomsten wordt ingenomen door de beschrijving van de aspecten van een dienst. Hiervoor kunnen de hierboven genoemde design-time standaarden worden gebruikt. Maar, voor vooral de beschrijving van de aanbieder (in geval van een dienstpublicatie) en de contracten (in geval van een dienstverleningsovereenkomst) is een nadere standaard nodig. Het is vooral de ebXML-familie die daarvoor standaarden biedt, namelijk CPA. Maar ook sjablonen voor SLA's kunnen hier een standaardiserende werking bieden.

De belangrijkste standaarden voor dienstenregisters beschrijven de diensten die door deze registers zelf worden geboden. De bekendste standaard daarvoor heet UDDI en komt uit de web-servicefamilie. De ebXML-familie biedt hiervoor RegRep.

5.3.3 Run-time standaarden voor logistiek en transport voor een SOA

De meest generieke standaarden op run-time zijn bedoeld voor het verzorgen van de logistiek en het transport van standaarden. Ook hier biedt zowel de web servicefamilie als de ebXML-familie internationale open standaarden[29]. SOAP speelt hierbij de hoofdrol in de web servicefamilie en ebMS in de ebXML-familie.

Voor een nadere beschrijving verwijzen we naar de lijst van open standaarden voor "pas-toe-of-leg-uit", zoals die wordt beheerd door het Forum Standaardisatie[30]. Op die lijst staan namelijk de OSB-standaarden, die logistiek en transport regelen.

5.4 Domein-specifieke standaarden in een SOA

Hoe moeilijk ook een passende keuze uit de beschikbare generieke SOA-standaarden mag zijn, het is nog maar een eerste stap: binnen een specifieke sector, organisatie, keten of netwerk moet ook nog gewerkt worden aan nadere afspraken over het onderlinge elektronische verkeer. Het gaat daarbij om run-time standaarden om:

- specifieke domeinmodellen
- specifieke dienstenspecificaties
- specifieke dialoogmodellen
- specifieke berichtenschema's

Deze volgorde is ook de principiële ontwerpvolgorde:

- domeinmodellen beschrijven de wereld waarover de andere drie gaan
- dienstenspecificaties beschrijven de bijdrage van de aanbieder, in termen van het domeinmodel
- dialoogmodellen beschrijven met welke dialoog deze bijdrage wordt geleverd
- berichtenschema's beschrijven het formaat van de uitingen die in de dialoog worden gedaan

[29] Naast nog enkele andere, die we hier niet zullen noemen.

[30] http://www.open-standaarden.nl/open-standaarden/lijsten-met-open-standaarden/lijst-voor-pas-toe-of-leg-uit/

We kunnen dus de volgende “contextladder” vormen:

- het domein is de context voor de dienst
- de dienst is de context voor de dialoog
- de dialoog is de context voor het bericht

Dat is van groot belang voor standaardisatie. Het heeft immers geen zin een standaard van toepassing te verklaren zonder daarbij de gebruikscontext af te bakenen. Andersom is het niet nodig een standaard te hebben voor twee geheel uiteenlopende gebruikscontexten: die kan toch niet zinvol worden toegepast.

Berichten standaardiseren

Vaak zijn berichten het onderwerp van standaardisatie, waarbij vooral veel aandacht is voor de berichtenstructuur: de velden die al dan niet verplicht voorkomen in het bericht en hun onderlinge plaatsing.

Vaak wordt daarbij niet expliciet genoemd in welke dialoog het bericht van toepassing wordt geacht. Meestal wordt dan geïmpliceerd dat het om een éénstaps-dialoog gaat die alleen uit dat ene bericht bestaat.

Die éénstaps-dialoog is vervolgens bedoeld op een dienst te bieden. Ook deze dienst lijkt vaak ongenoemd te blijven. Toch wordt deze wel vaak benoemd, in de vorm van het *berichttype* – bijvoorbeeld: verhuisbericht. Dit berichttype staat min of meer voor het effect van het versturen van het bericht, ofte wel voor het effect van deze “éénstaps-dialoogdienst”.

Soms staat dat effect nadrukkelijk beschreven, maar vaak ook niet. De allerhoogste context is vervolgens het domeinmodel. Vaak ontbreekt dit, maar regelmatig is een zeker domeinmodel wel aanwezig, bijvoorbeeld in de vorm van een gegevenswoordenboek.

Context en standaardiseerbaarheid

Dus, hoewel standaarden niet zonder een gebruikscontext kunnen, worden berichtenstandaarden vaak vanaf de laagste trede van de contextladder ontworpen. Toch is dat niet zo raar: hoe hoger op de contextladder, des te specifieker en groter het model en des te lastiger het dus is om het te standaardiseren. Dus: berichten zijn “kleiner” en “generieker” en daarom makkelijker standaardiseerbaar.

Toch is het uitkijken geblazen: het niet noemen van relevante context maakt een standaard alleen op het oog breder toepasbaar: in de praktijk is de passing in de gebruikscontext toch de toetssteen.

Vaak ook worden berichten “geschikt gemaakt” voor meerdere gebruikscontexten door bepaalde velden optioneel te maken. Daarmee wordt de standaard vrijer, maar is de kans op gebrekkige interoperabiliteit natuurlijk ook groter.

Dienst-aanbieders zullen in specifieke gevallen toch moeten kiezen hoe zij met deze velden omgaan. Een goede berichtenstandaard geeft dus zo veel mogelijk aan in welke gevallen een optioneel veld wél of juist niet moet worden gebruikt.

Dialogen en diensten standaardiseren
Standaardiseren van een dialoog is aan de orde als meerdere diensten dezelfde dialoog of hetzelfde dialoogfragment zouden moeten toepassen. Het is de vraag hoe zinvol dit is als het gaat om sterk verschillende diensten. Waarom zouden twee sterk verschillende diensten eenzelfde dialoog moeten hanteren? Als diensten sterk op elkaar lijken of zelfs hetzelfde zijn, is dat natuurlijk wel zinvol, maar dan is er sprake van dienstentandaardisatie.

Toch lijken veel dialogen van verschillende diensten vaak wel op elkaar, op een sjabloonachtige manier. Daarbij geldt vaak dat een dialoogpatroon hoort bij een type dienst. Voorbeelden daarvan zijn:

- vraag-antwoorddialogen, bijvoorbeeld voor bevragingen
- éénstaps notificatiedialogen

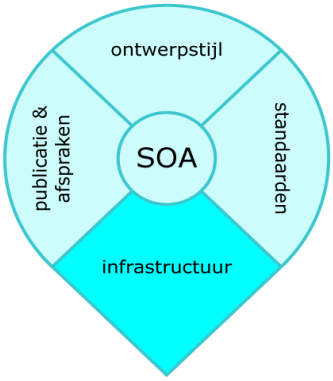
Het komt de interoperabiliteit binnen een zekere gemeenschap (zoals de e-overheid) ten goede als zij kan beschikken over een set van:

- veel voorkomende types diensten
- met bijbehorende dialoogpatronen
- met bijbehorende berichtenpatronen

zodat ontwikkelaars van diensten en van standaarden deze kunnen hergebruiken. Dat scheelt niet alleen tijd en kosten in het ontwerp, maar verhoogt en versnelt ook het wederzijds begrip en de interoperabiliteit binnen de e-overheid.

6 Servicebussen

Met servicebussen zijn we aangeland bij het vierde en laatste — en meest technologische — aspect van SOA. Servicebussen zijn voorzieningen die de logistieke afhandeling van dienstenverkeer tussen betrokken partijen, door middel van berichten. Daarom kunnen we voor servicebussen wél de logistieke beeldspraak gebruiken die voor andere aspecten van SOA zo gevaarlijk is. Servicebussen zijn de logistieke dienstverleners in een SOA.



Logistiek
Alle partijen die aangesloten zijn op een servicebus vormen daarmee een logistieke gemeenschap, maar dat wil niet zeggen dat zij altijd zomaar van elkaars diensten gebruik kunnen maken, noch dat zij geen gebruik zouden kunnen maken van diensten van partijen buiten deze logistieke gemeenschap (en andersom).

Vergelijk het met alle bedrijven die een eigen aansluiting hebben op het spoorwegnet, zoals dat een eeuw geleden vrij gebruikelijk was. Zij zijn logistiek verbonden en kunnen dus in principe goederenverkeer met elkaar aangaan. Echter, zij hebben niet per se allemaal een zakelijke relatie. Andersom kunnen zij ook zaken doen met bedrijven die geen directe aansluiting hebben op het spoorwegnet. Daarvoor zijn overslagpunten ingericht, waar goederen kunnen wisselen van transportmiddel.

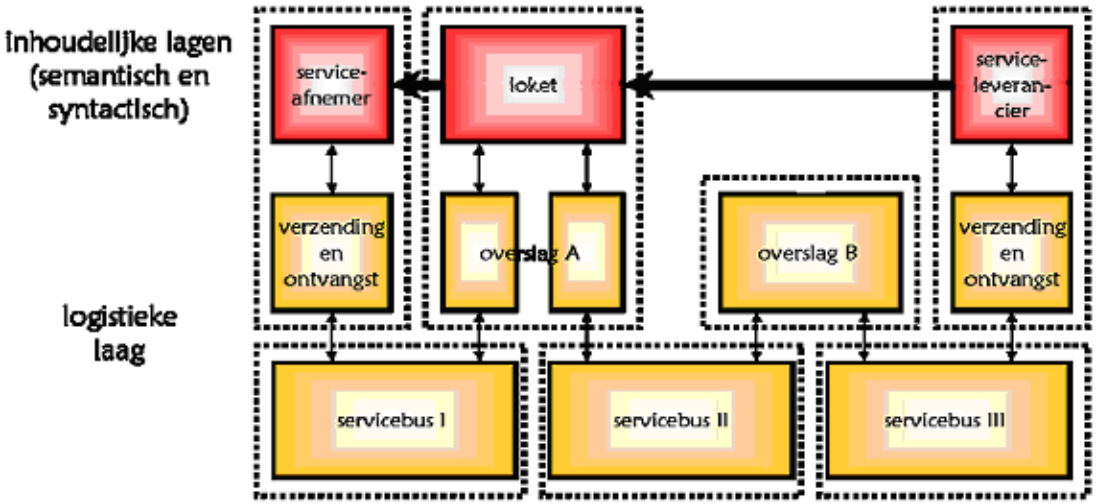
In een SOA bestaan ook van deze overslagpunten. Let wel, overslag is ook een logistieke activiteit, die principieel geen “boodschap heeft aan de boodschap”, dat wil zeggen, die zich geen toegang verschaft tot — of een bewerking uitvoert op — de inhoud van de berichten.

In de wereld van de fysieke logistiek komt het ook voor dat, als men toch bezig is de goederen over te slaan, men tóch ook bewerkingen op de goederen uitvoert. Veelzeggend is dat men hiervoor de term “value-added logistics” gebruikt. Ook hiervoor is een mooie parallel in de SOA, namelijk het (diensten)loket. Een loket is een voorziening die diensten uit haar eigen “achterland” ontsluit voor een buitenwereld. Dat bij — maar niet ín — een loket ook overslag van berichten plaatsvindt, is logisch. Soms zelfs verrijzen in de buurt van logistieke overslagpunten ook fabrieken waar werkelijke productie plaatsvindt. Verder maken loketten vaak gebruik van een dienstenregister om de door hen geleverde diensten te publiceren.

Toch is het belangrijk deze zaken uit elkaar te halen: berichtenoverslag, loketten en feitelijke productie, ook al zijn ze dicht bij elkaar gebouwd. In de e-overheid komt het regelmatig voor dat “centrale” voorzieningen worden ontwikkeld, die een mengeling van overslag, loket en productie voor hun rekening nemen. Dat is niet principieel af te raden, maar het is wel zaak deze drie onderdelen logisch, en qua verantwoordelijkheid, te blijven scheiden.

Typische voorbeelden van loketten zijn natuurlijk de burger- en bedrijvenloketten die voor Internet-dienstverlening zijn of worden ontwikkeld, maar ook bijvoorbeeld de Digipoort voor dienstverlening tussen applicaties van bedrijven en overheden.

Aanbieders en afnemers van diensten, die op één of meerdere servicebussen zijn aangesloten moeten natuurlijk ook zelf de logistieke ontvangst en verzending van hun berichten op orde hebben. Deze functionaliteit wordt bij berichtenverkeer vaak adapter genoemd.



Bovenstaand schema geeft een voorbeeld van een dienst-afnemer die een dienst gebruikt van een dienstenleverancier, waarbij een geschakelde keten van drie servicebussen wordt ingeschakeld, die op de ene plaats met een puur overslagpunt zijn gekoppeld en op de andere plaats via een loket. De bovenste, rode laag staat voor de inhoudelijke (semantische en syntactische) aspecten en toont dus de dienstverleningsrelaties (met de dikke pijlen). De afnemer neemt een dienst af van het loket, die die dienst betrekt bij de uiteindelijke leverancier. De dunne pijlen laten zien hoe het berichtenverkeer logistiek verloopt.

De gestippelde lijnen suggereert een gebundelde plaatsing van de verschillende functionaliteiten bij spelers. Dit is een voorbeeld: de bundeling kan ook heel goed anders. Ook is het zo dat dienst-aanbieders en afnemers op meerdere bussen kunnen zijn aangesloten, net als overslagpunten en loketten.

Een stelsel van servicebussen
Het is prettig om servicebussen te kunnen schakelen in “logistieke ketens”. Het is immers niet alleen onhaalbaar, maar zelfs onwenselijk om alle e-overheidsapplicaties rechtstreeks op één enkele servicebus aan te sluiten. Daarvoor is de schaal te groot, de logistieke wensen te divers en de ontwikkelingsvoortgang van verschillende overheidsonderdelen te uiteenlopend.

Daarom wordt binnen de e-overheid een stelsel van servicebussen voorzien die met overslagvoorzieningen gekoppeld zijn, al dan niet verrijkt met loketten. Grofweg zal zo’n stelsel er als volgt uit zien.

- Elke overheidsorganisatie zelf kent zijn eigen interne servicebus[31]. Grote organisaties kunnen er zelfs eentje per business unit kennen. Elke overheidsorganisatie kent een loket via welke eigen diensten naar buiten worden ontsloten en externe diensten naar binnen.
- Elke overheidssector en –laag kent een sectorale servicebus. Soms wordt dit gecombineerd met een sectorloket, maar in veel gevallen zal er op overslagniveau worden gekoppeld.
- De Nederlandse overheid kent een overheidsbrede servicebus, die Digikoppeling heet. Deze speelt een hoofdrol in het onderling verbinden van de sectorale servicebussen, maar een aantal grote overheids-brede diensten en een aantal van hun afnemers kunnen ook rechtstreeks op Digikoppeling zijn aangesloten.

[31] Zo’n servicebus wordt vaak een Enterprise Service Bus genoemd (ESB).

7 Hoe krijgt de e-overheid een SOA?

In dit laatste hoofdstuk wil dit dossier in hoofdlijnen aangeven welke ontwikkelingen nodig zijn om de Nederlandse e-overheid zich te laten ontwikkelen naar een SOA. Daarbij wordt opnieuw het SOA-wiel gebruikt om de ontwikkelingslijnen te structureren.

Moge het, om te beginnen, duidelijk zijn dat service-oriëntatie een onderwerp is met een reikwijdte en diepte die zo groot is, dat de e-overheid gezamenlijk moet optrekken om het te realiseren. Die gezamenlijkheid moet bovendien aan de dag gelegd worden op alle besproken aspecten: semantiek, syntax en technologie. Het is niet mogelijk om SOA als een technologische ontwikkeling af te doen en haar dienovereenkomstig te beleggen of uit te besteden. Dus:

> Om tot een SOA te geraken, moet de Nederlandse e-overheid werken onder architectuur, voortbouwend op wat NORA op dat terrein in gang heeft gezet.

7.1 Ontwerpstijl

In een SOA zijn de werkzame onderdelen van de e-overheid allemaal dienst-afnemers en –aanbieders. Dat geldt zowel voor organisaties, hun afdelingen en bedrijfsprocessen, als voor applicaties en de belangrijkste hoofdcomponenten van die applicaties.

> Om tot een SOA te geraken, moet van alle huidige en toekomstige werkzame onderdelen van de e-overheid een precieze dienstenbeschrijving gaan bestaan, die gesteld is in termen van een passend domeinmodel.

De hoofdverantwoordelijkheid daarvoor ligt bij de aanbieder van de dienst. In uitvoerende zin ligt deze taak echter meestal bij architecten en ontwerpers, die zich in dat geval met allerlei vragen geconfronteerd zien, zoals die van de granulariteit. Daarom formuleren we bovendien:

> Voor de ontwikkeling van diensten en dienstenbeschrijvingen moeten de ontwerpers en opstellers kunnen beschikken over gezamenlijke handreikingen, conventies, richtlijnen, patronen en anti-patronen voor dienstenontwerp.

Voor een belangrijk deel wordt de ontwikkeling en bouw van applicaties belegd bij externe commerciële partijen. Ook in dat geval moeten de diensten van de geleverde applicaties duidelijk zijn.

> Alle door derden geleverde software moet voorzien zijn van precieze en voor de aanbieder en –gebruikers vrij toegankelijke dienstbeschrijvingen.

Onder de werkzame onderdelen van de e-overheid behoren zeker ook de organisaties en hun onderdelen, die wederzijds diensten verlenen en aan burger en bedrijf. Normaliter wordt het arrangement van de Nederlandse overheid uitgedrukt in de verdeling van — in dit geval uitvoerende — taken over organisaties. Daarbij wordt snel gedacht in het claimen van eigenaarschap van activiteiten. Service-oriëntatie begint anders, namelijk bij het nemen van verantwoordelijkheid voor te bieden diensten en vervolgens verkrijging van het mandaat om de middelen (waaronder eventueel andere diensten) daarvoor aan te spreken.
Een dergelijke manier van denken zal niet één-twee-drie zijn beslag krijgen. Toch blijft de vuistregel: “Services are not owned, they are used.”

> Voor de ontwikkeling van een service-georiënteerde e-overheid moeten overheidsorganisaties kunnen beschikken over richtlijnen en handreikingen over hoe wettelijke taken zich laten vertalen naar bedrijfsdiensten en -diensten.

Ten slotte, waar de inhoud van diensten grotendeels sector- of organisatie-specifieke zal zijn, geldt dat niet voor de uitdrukkingsmiddelen en beschrijvingstalen daarvoor. Tegelijkertijd beperkt de diversiteit daarin wel de interoperabiliteit en de koppelingskosten binnen en met de e-overheid. Daarom formuleren we het volgende.

> Voor de ontwikkeling van een service-georiënteerde e-overheid moet harmonisatie in gang worden gezet voor de uitdrukkingsmiddelen en beschrijvingstalen voor alle syntactische en semantische aspecten van diensten, op basis van volwassen, open en internationale standaarden.

7.2 Publiceren en afspraken

Dienstenregisters zijn sleutelvoorzieningen in een SOA. Zij staan het toe dat dienst-afnemers diensten en hun aanbieders ontdekken. Bovendien vormen de dienstenbeschrijvingen in die registers de precieze basis voor dienstverleningsovereenkomsten en voor de uiteindelijke koppeling tussen de partijen.

Net zoals er niet slechts één servicebus zal kunnen en moeten komen voor de gehele overheid – maar een gekoppeld stelsel van servicebussen – zo zal er ook niet slechts één servicebus zijn voor de gehele overheid. In plaats daarvan zullen groepen van onderlinge dienstenaanbieders en – afnemers gemeenschappen vormen die gezamenlijk een dienstenregister onderhouden.

Voor de hand liggend- maar niet noodzakelijk – is dat individuele organisaties hun eigen dienstenregister hebben, net als bijvoorbeeld sectoren en ketens. Omdat echter de overheid niet is op te delen in een aantal geïsoleerde diensteneilanden, zullen die registers onderling dienstenbeschrijvingen moeten kunnen kopiëren of delen of onderling kunnen doorverwijzen.

> Voor een service-georiënteerde e-overheid moet een geschakeld stelsel van diensten registers worden ontwikkeld.

Ook hier geldt dat de toepasselijke standaarden generiek zijn. Dus:

> Voor de ontwikkeling van een service-georiënteerde e-overheid moet harmonisatie in gang worden gezet voor de toegang en koppeling van dienstenregisters en voor dienstpublicaties en -overeenkomsten, op basis van volwassen, open en internationale standaarden.

Verder zouden diensten-aanbieders en –afnemers geholpen zijn met richtlijnen voor het opstellen van dienstpublicaties en –overeenkomsten, bijvoorbeeld rondom:

- hoe in dienstpublicaties de balans tussen standaard en maatwerk moet worden gekozen
- hoe op basis van de dienstpublicatie en de wensen van betrokken partijen gekomen kan worden tot een dienstverleningsovereenkomst

Sjablonen voor (archetypische) dienstpublicaties en –overeenkomsten zullen hiervan deel uitmaken.

> De ontwikkeling van een service-georiënteerde e-overheid kan profiteren van richtlijnen voor het opstellen van dienstpublicaties en -overeenkomsten.

7.3 Standaardiseren

Standaardisatie voor een service-georiënteerde e-overheid valt uiteen in:

- het kiezen van open internationale standaarden voor de generieke aspecten
- het adopteren van een service-georiënteerde benadering voor het standaardiseren van de specifieke aspecten.

Het eerste punt stelden we al eerder in dit dossier aan de orde. Voor de specifieke aspecten formuleren we:

> Voor een service-georiënteerde e-overheid is het van belang tot een gezamenlijke methode te komen voor het service-georiënteerd ontwikkelen en beheren van domein-specifieke standaarden.

Hoe domein-specifiek echter ook, diensten delen onderling vaak toch nog het semantische domein of een onderdeel daarvan. Om over diensten heen de interoperabiliteit te bevorderen, formuleren we:

> Voor de semantische interoperabiliteit in een service-georiënteerde e-overheid is het van belang dat gestandaardiseerde (fragmenten van) domeinmodellen worden ontwikkeld en ontsloten in een semantische repository.

7.4 Servicebussen

In navolging van de eerdere beschrijving van servicebussen formuleren we:

> Een service-georiënteerde e-overheid vereist een samenhangend stelsel van servicebussen. Een dergelijk stelsel kan niet functioneren zonder de logistieke afhandeling van diensten te standaardiseren.

Voor een service-georiënteerde e-overheid is het van belang een beperkt aantal standaarden te hanteren voor de logistieke afhandeling van dienstenverkeer, op basis van volwassen, open en internationale standaarden.

7.5 Valkuilen

Een goede advertentie eindigt met een pakkende pay-off. Maar, zonder het nut en misschien zelfs wel de noodzaak van service-georiënteerde architecturen te willen onderschatten, wil dit dossier afsluiten met een aantal waarschuwingen voor valkuilen. Daarmee wil het het zilvergehalte van de SOA-kogel relativeren en tegelijkertijd helpen voorkomen dat de ontwikkeling van een service-georiënteerde e-overheid gaat lijden onder teleurstellingen ten gevolge van al te overspannen verwachtingen of oversimplificaties.

Valkuil 1 - Denken dat SOA te koop is

Er bestaan geen SOA-leveranciers. SOA is geen product en geen technologie. Het binnenhalen van een servicebus is niet voldoende en zeker ook niet het binnenhalen van een BPM-oplossing. Een service-georiënteerde e-overheid vraagt om maatregelen op alle vier de spaken in het SOA-wiel.

Valkuil 2 - Denken dat SOA alles regelt

Een SOA biedt u een ontwerpstijl voor zakelijke en transparante architecturen, die goed presteren op beheersbaarheid, toekomstvastheid, koppelbaarheid en flexibiliteit. Maar het geeft geen pasklaar antwoord op vragen als:

- Welke diensten moet ik precies kiezen, hoe groot, hoe klein?
- Waar leg ik de balans tussen maatwerk- en standaard-diensten?
- Hoe zorg ik dat mijn diensten de goede kwaliteit, performance en beveiliging aan de dag leggen?
- Hoe beheers ik de afhankelijkheid van diensten van derden?
- Hoe beheers ik doorlopend het portfolio aan diensten dat ik aanbiedt en afneem?
- Hoe beschrijf en standaardiseer ik mijn domeinmodellen?

Valkuil 3 - Denken dat SOA een onderwerp is voor technologen

Het kan niet genoeg gezegd worden: SOA raakt de organisatie op zijn minst evenzeer als de technologische infrastructuur. Software is allang niet meer alleen een instrument, maar een uitvoerder van bedrijfsprocessen. Service-georiënteerde software vraagt om een service-georiënteerde organisatie en omgekeerd.

Valkuil 4 - Denken dat SOA vrijheid-blijheid is
Eén van de belangrijkste pluspunten van SOA is dat het de werkzame onderdelen van de e-overheid een zekere vrijheid biedt. De bedoeling is dat deze vrijheid groot genoeg is om deelnemende partijen de bewegingsruimte te geven die zij nodig hebben. Maar, ook in een SOA is sprake van gebondenheid. Zonder gebondenheid is samenhang immers ondenkbaar. Die gebondenheid zit hem in:

- de dienstverleningsovereenkomsten die tussen aanbieders en afnemers van diensten worden aangegaan
- de *standaardisatie* en *harmonisatie* die nodig is voor een soepel functionerende SOA
- het gebruik van *gemeenschappelijke basisvoorzieningen* die nodig zijn voor een soepel functionerende SOA, zoals servicebussen en dienstenregisters

Ook een service-georiënteerde e-overheid vraagt dus om governance.

Valkuil 5 - Window dressing
SOA is niet makkelijk en SOA kan ingrijpend zijn. Het gevaar is daarom groot dat oude wijn in SOA-zakken wordt gestopt of dat het SOA-label wordt gebruik voor producten die dat kaartje niet of maar beperkt verdienen. Dus — als variant op een klassieke slogan — een SOA is nog geen SOA als er “SOA” op staat.

Met de elementen van het SOA-wiel en de uitwerking van die elementen heeft dit dossier houvast willen geven in wat de kern is van een SOA.

Valkuil 6 - Overhaasten
De verstrekkendheid van service-oriëntatie brengt bovendien met zich mee dat een SOA alleen stapsgewijs en in kleinere onderdelen kan worden ingevoerd. De maatregelen die hierboven in dit hoofdstuk staan genoemd zijn per stuk al een project of programma op zichzelf. Een keuze voor een SOA is dus een keuze voor langere tijd.

Valkuil 7 - Onderschatting
Ten slotte, hopelijk heeft dit dossier u overtuigd van de waarde van service-oriëntatie in de e-overheid. Relativering van de hype is zeker op zijn plaats, maar moet niet omslaan in onderschatting. SOA, of welke naam het in de toekomst ook van marketeers moge krijgen, is een structurele ontwikkeling.

En zo krijgt dit dossier toch nog zijn pay-off.

8 Afkortingen

ASL	Application Services Library
BiSL	Business information Services Library
BPE	Business Process Engineering
BPEL	Business Process Execution Language
BPM	Business Process Management
BPR	Business Process Redesign
BPSS	Business Process Specification Scheme
CPA	Collaborating Partner Agreement
CPP	Collaborating Partner Profile
CRUD	Create, Read, Update, Delete
DFD	Data Flow Diagram
DTD	Document Type Definition
ebMS	ebXML Messaging Service
ebXML	electronic business XML
EDA	Event-Driven Architecture
ERD	Entity Relation Diagram
ESB	Enterprise Service Bus
FTP	File Transfer Protocol
GBA	Gemeentelijke Basisadministratie Persoonsgegevens
HTTP	HyperText Transfer Protocol
ICTU	ICT Uitvoeringsorganisatie
ITIL	Information Technology Infrastructure Library
MDA	Model-Driven Architecture
MSD	Message Sequence Diagram
NORA	Nederlandse Overheids Referentie-Architectuur
OCL	Object Constraint Language
OSB	OverheidsServiceBus
OTP	OverheidsTransactiePoort
OWL	Web Ontology Language
RDF	Resource Description Framework
RENOIR	Regie E-overheid NUP Ondersteuning Implementatie en Realisatie
RegRep	Registry/Repository
SOA	Service-georiënteerde Architectuur
SLA	Service-Level Agreement
SNO	Serviceniveau-Overeenkomst
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SWSF	Semantic Web Service Framework
UDDI	Universal Description, Discovery and Integration

UML	Unified Modeling Language
VDM	Vienna Development Method
WSCL	Web Service Choreography Language
WSDL	Web Service Definition Language
WSMO	Web Service Modelling Ontology
XML	eXtensible Mark-up Language
XSL-T	eXtensible Stylesheet Language — Transformations

9 Document informatie

Colofon

Auteur Paul Oude Luttighuis	Status Definitief
Project -	Datum 1-11-2010
Organisatie NORA	Classificatie -
Titel Service Georiënteerde Architectuur	Locatie -
Versie 1.0	

Historie

Datum	Versie	Wijziging	Status	Verwerkt door

Distributielijst

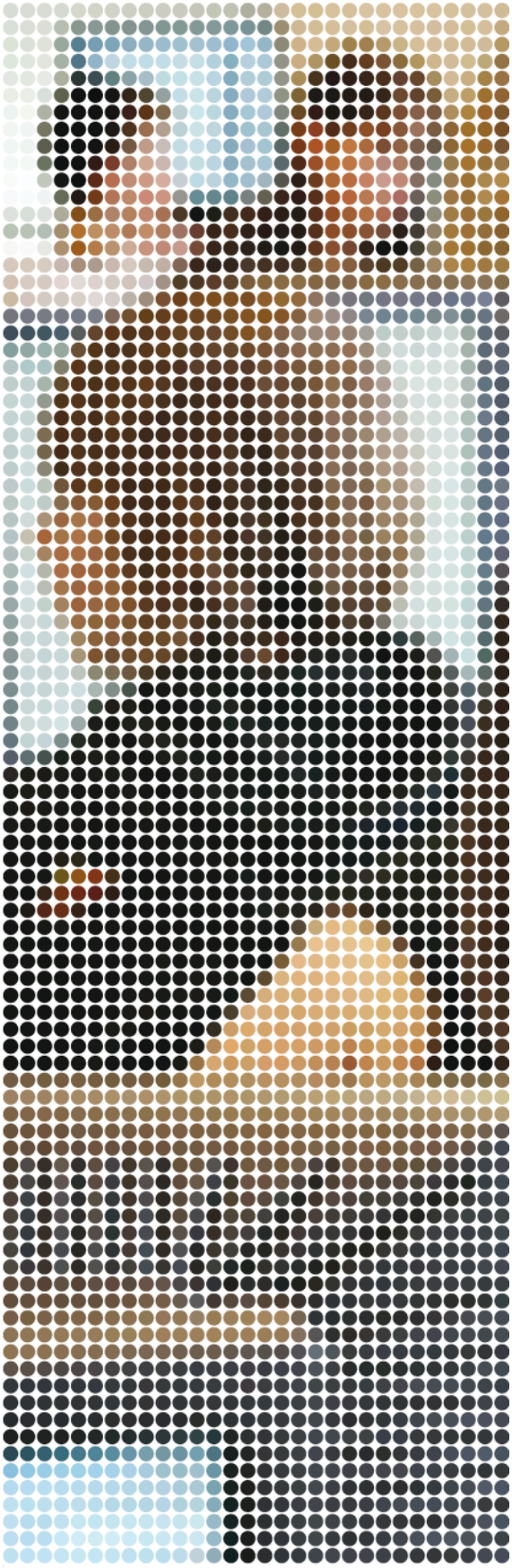
Datum	Distributie	Presentatie	Versie

Goedkeuring

Datum	Naam	Versie

Dit document is beschikbaar onder de volgende Creative Commons licentie:

http://creativecommons.org/licenses/by-nd/3.0/nl/



Wilhelmina van Pruisenweg 104
2595 AN Den haag
Postbus 84011
2508 AA Den Haag

T (070) 888 78 20
F (070 888 78 81
www.e-overheid.nl