

Procesintegratie in Oracle SOA Suite

R&D-traject rond ketenintegratie overheid

Tijdens een Research en Development traject naar de Oracle SOA Suite bij Capgemini is de ketenintegratie tussen (overheids)instanties vormgegeven. De focus was primair gericht op het ontwikkelen van de integratielaag. In het vorige artikel, verschenen in februari van dit jaar, zijn de data-integratie mogelijkheden van de Oracle SOA Suite behandeld. In dit vervolg artikel zal vooral worden ingaan op de mogelijkheden van procesintegratie binnen de Oracle SOA Suite.

We staat in dit artikel vooral stil bij de mogelijkheden om binnen de SOA Suite conform een Service Oriented Architecture tot een orkestratie van services, door middel van BPEL, te komen. Het artikel is opgebouwd uit twee delen: eerst behandelen we de concepten van een Service Oriented Architecture, daarna volgt een concreet voorbeeld, zijnde het orkestratie scenario uit de Proof of Concept ketenintegratie tussen (overheids)instanties.

Services

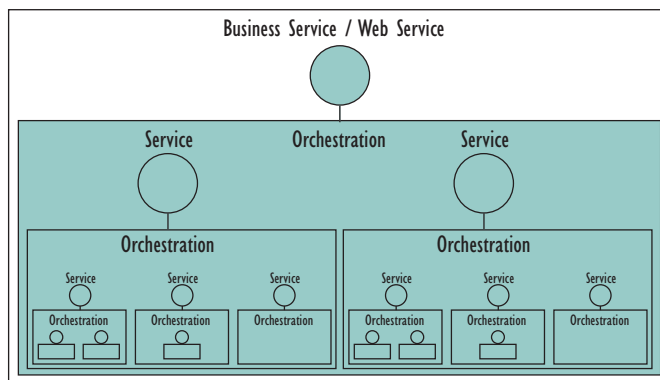
Een Service Oriented Architecture staat voor 'flexibiliteit' en voor 'ontwerpen voor verandering'. Een Service Oriented Architecture bestaat uit een hiërarchie van services. Een service heeft een duidelijk doel, is een eenheid van handelen en

heeft een duidelijk resultaat zoals beschreven in de specificatie van de service. Een service kan ook een samengestelde service zijn van meerdere elementaire services. Voor deze services gelden de volgende eigenschappen:

- **Gestandaardiseerde aanroep.** De specificaties voor het aanroepen van de service zijn hiervoor afgesproken, bijvoorbeeld in een WSDL voor een webservice. Op welke wijze de service geïmplementeerd is, is voor de buitenwereld niet van belang. De specificatie en interne werking is hierdoor strikt gescheiden;
- **Ontkoppelbaarheid.** Dankzij de gestandaardiseerde interface kan de service inhoudelijk worden gewijzigd (bijvoorbeeld bij technische upgrades) zonder dat het samenspel met andere services wordt beïnvloed;
- **Waardetoevoeging.** Services die geen waarde (meer) toevoegen of niet meer worden gebruikt, verliezen hun bestaansrecht en kunnen eenvoudig worden verwijderd;
- **Hergebruik.** Ontwerpen volgens het principe 'hergebruik' impliceert al dat rekening wordt gehouden met verandering en toekomstig gebruik;
- **Service Repository en Agreement management.** Een service heeft een specificatie met details over functionaliteit, foutafhandeling, gegevens, synchroniciteit etc. en heeft contracten met consumers over onder andere quality of service. Daarvoor dient een repository van services aanwezig te zijn evenals een SLA register waar de contracten gespecificeerd staan.

Procesintegratie

Services kunnen worden samengevoegd tot een proces, welke op haar beurt weer kan worden opgenomen in een groter proces. Op deze wijze ontstaat een hiërarchie van processen. We spreken in dit geval over procesintegratie binnen een SOA filosofie. Afbeelding 1 illustreert dit.



Afbeelding 1. Procesintegratie

Doordat processtappen op een gestandaardiseerde wijze met elkaar zijn te koppelen, wordt het eenvoudig om te komen tot een geïntegreerd totaalproces. Wanneer bedrijfsprocessen of administratieve handelingen wijzigen ('organisatie flexibiliteit')

Welke componenten zijn noodzakelijk om een SOA omgeving mogelijk te maken, en wat is de implementatie van Oracle van deze componenten.

Providers /Consumers

Services worden aangeboden door Providers en afgenomen door Consumers. Services kunnen door een verscheidenheid van bronnen aangeboden worden. Web Services is momenteel de meest gebruikte implementatie.

Development

In de Oracle Fusion omgeving worden Services ontwikkeld met JDeveloper.

Service Registry

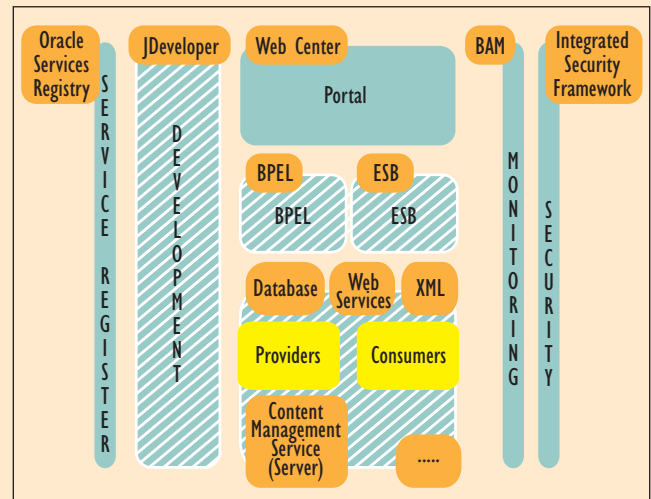
Een van de hoofddoelen van Service Orientatie is hergebruik van bestaande functionaliteit. Een Service Registry helpt bij het (terug)vinden van aanwezige Services.

ESB/BPEL

De integratie componenten ESB en BPEL zijn essentieel in de SOA architectuur. Zaken als garantie van aankomst, transformatie, routing, proces integratie worden met deze tooling verzorgd.

Monitoring

Bij het ontwerpen van een Service gaat het niet alleen om de functionele aspecten maar ook om de niet-functionele, zoals



De Proof of Concept gaat in op de gearceerde gebieden.

beschikbaarheid, transport details, performance en dergelijke. De Monitoring component controleert deze aspecten, en kan real-time ook uitzonderingssituaties melden.

Security

Door wie en hoe mag een Service behandeld worden? Deze functionaliteit wordt los van de Service bepaald en uitgevoerd.

Portal

Een portal vormt de toegang tot interne en/of externe informatievoorziening.

hoeft in deze opzet niet het hele systeem te worden aangepast maar volstaat het wijzigen van de services in het grotere geheel. Technologisch gezien biedt dit een hogere mate van flexibiliteit.

BPEL als orchestrator

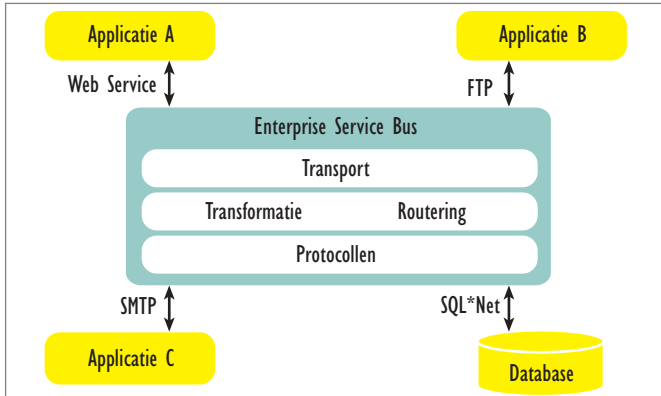
Een service dient bij voorkeur niet zelf te zorgen voor de samenwerking met andere services. Een service is zoveel mogelijk 'Stateless', dat wil zeggen dat ze bestuurd moeten worden en niet zelf moet sturen op de wijze waarop meerdere services (of een samengesteld proces van services) onderling samenwerken. Om dit te verduidelijken wordt vaak de vergelijking met een orkest gemaakt waarbij de muzikanten de aanwijzingen van de dirigent opvolgen. Het besturen van services wordt daarom ook wel aangeduid als orchestreren.

Voor de wijze waarop services onderling georchestreerd worden, is de Business Process Execution Language (BPEL) bij uitstek geschikt. Zoals in afbeelding 1 al werd geïllustreerd, kun-

nen door deze orchestratie van services tot grotere procesdelen, uiteindelijk bedrijfsprocessen worden samengesteld. BPEL is een op XML gebaseerde taal die de flow binnen een bedrijfsproces beschrijft om zo de interactie tussen één of meer services aan te sturen.

De rol van de ESB

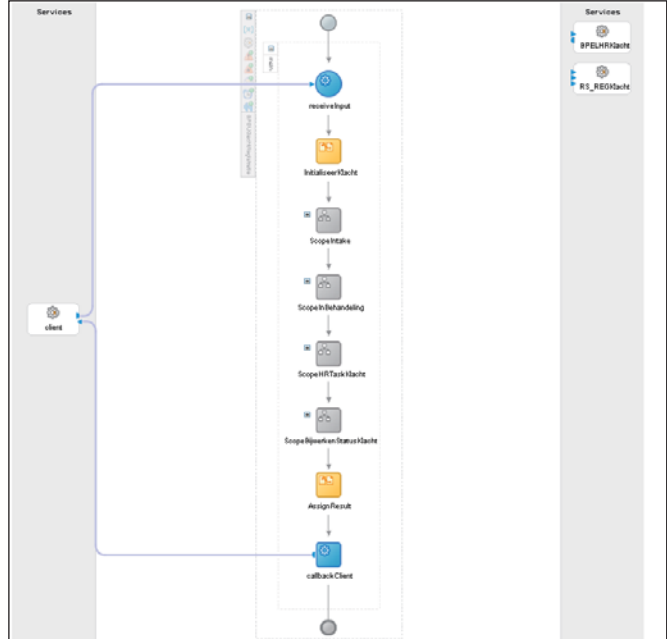
Een individuele service (of georchestreerd samenspel van services), kan op een aantal manieren ter beschikking worden gesteld aan externe services of applicaties: door middel van een UDDI registry, een rechtstreekse aanroep via een endpoint, of door middel van een aanroep via een Enterprise Service Bus (ESB). Een van de belangrijkste redenen om een ESB in te zetten, is dat het daarmee mogelijk wordt om bestaande applicaties of software componenten direct als een service aan te spreken. Daarbij ontstaat de mogelijkheid om nieuwe processen die geheel zijn opgebouwd uit (web)services, te integreren. De ESB bevat alle noodzakelijke functionaliteiten voor uitwisseling tussen applicaties zoals afbeelding 2 laat zien.



Afbeelding 2. Functionaliteit ESB

Proof of Concept

Zoals in het vorige artikel beschreven, verschenen in februari van dit jaar, is tijdens een Research en Development traject bij Capgemini de ketenintegratie tussen (overheids)instanties vormgegeven in een Proof of Concept (PoC). Het doel hiervan was tweeledig: 1. om kennis te krijgen van de Oracle SOA Suite en 2. om te bepalen of de Oracle SOA suite geschikt is voor integratie in een SOA omgeving zoals die ons voor ogen staat. In ons PoC traject is er uitgegaan van een fictieve organisatie die integratiediensten verzorgt voor aanbieders en afnemers van services: lokale, regionale, landelijke overheid, instanties en bedrijven. Hierbij zijn een viertal integratiescenario's uitgewerkt (zie afbeelding 3), waarbij in dit artikel de focus verder zal liggen op het

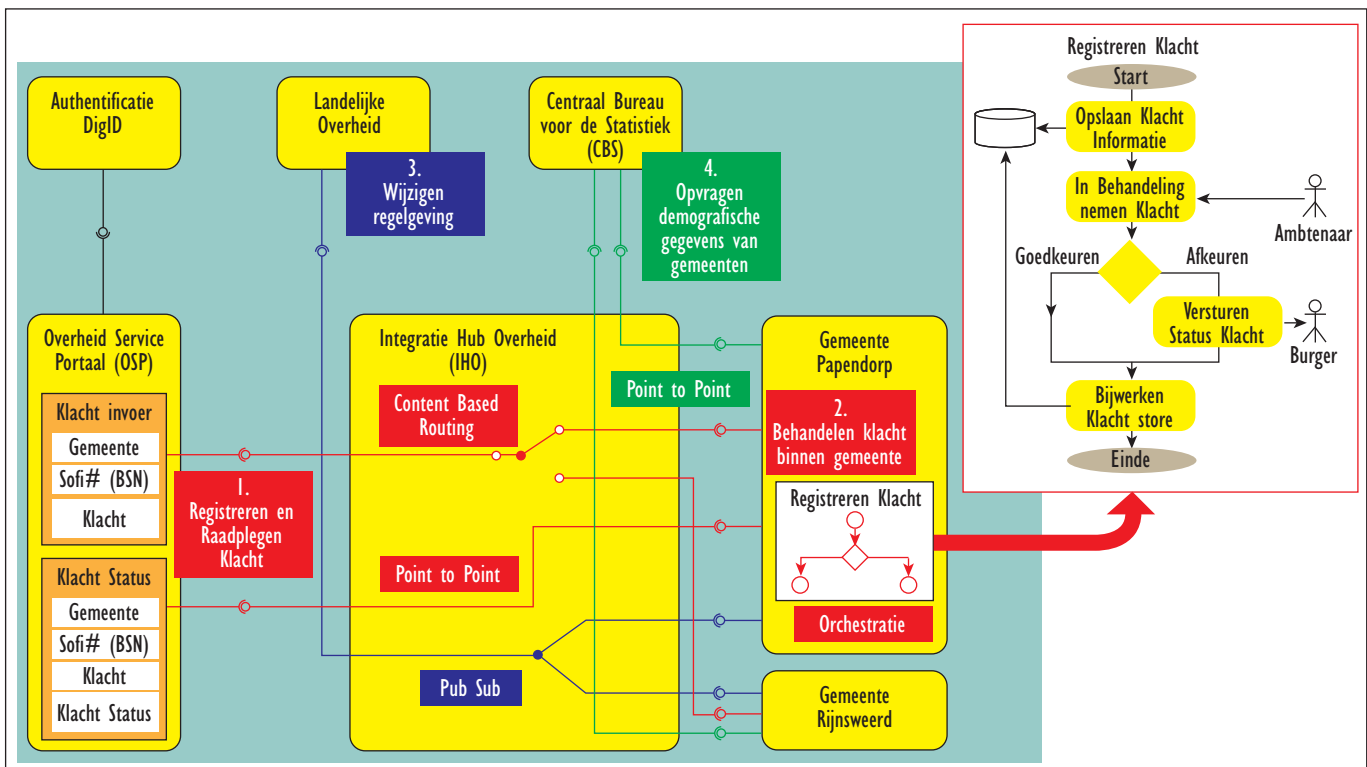


Afbeelding 4. Bedrijfsproces 'Registreren Klacht'

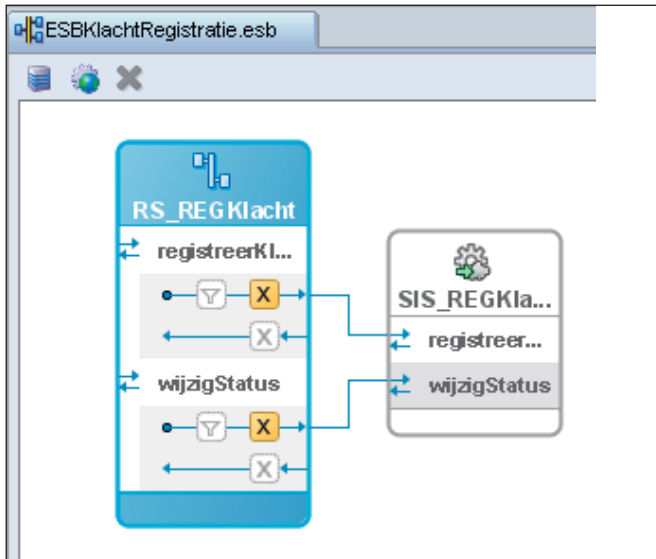
ontwikkelen van de integratielaag, voor scenario 'Orchestratie' (de afhandeling van een klacht binnen de gemeente).

Orchestratie scenario

Bij het integratiescenario 'Orchestratie' zijn een tweetal services binnen JDeveloper aangemaakt, service 'REGKlacht' voor



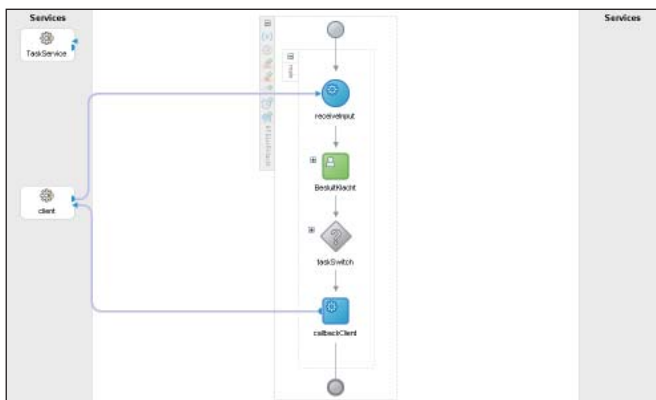
Afbeelding 3. Ketenintegratie overheid



Afbeelding 5. ESB Designer

het registreren en wijzigen van een klacht in een database, en service 'BPELHRKlacht' voor het in behandeling nemen van een klacht door een ambtenaar van een gemeente. Het bedrijfsproces 'Registreren Klacht' is binnen JDeveloper gemodelleerd met de BPEL Proces Designer (zie afbeelding 4) en bestaat uit een aantal processtappen (Process Activities), zoals 'receiveInput' en 'AssignResult'. Processtappen kunnen binnen een scope gegroepeerd worden (bijvoorbeeld 'ScopeBijwerkenStatusKlacht'). Hierbij kan het detail in het getoonde schema beperkt worden, waardoor bijvoorbeeld het hoofdproces beter leesbaar blijft (verminderen van complexiteit).

De service 'REGKlacht' wordt via de Enterprise Service Bus aangeroepen waarmee zaken als transport, transformatie en routing gemakkelijk te regelen zijn. Binnen JDeveloper is met de ESB Designer een routing service 'RS_REGKlacht' gemodelleerd (zie afbeelding 5) welke de service 'REGKlacht' aanroept (via een zogenaamde SOAP Service, genaamd 'SIS_REGKlacht').



Afbeelding 6. Service 'BPELHRKlacht'

De routing service wordt vanuit het bedrijfsproces 'Registreren Klacht' aangeroepen.

De service 'BPELHRKlacht', die binnen het bedrijfsproces 'Registreren Klacht' wordt aangeroepen, is op haar beurt ook weer gemodelleerd met de BPEL Proces Designer (zie afbeelding 6), en roept zelf weer een andere service aan, zijnde TaskService.

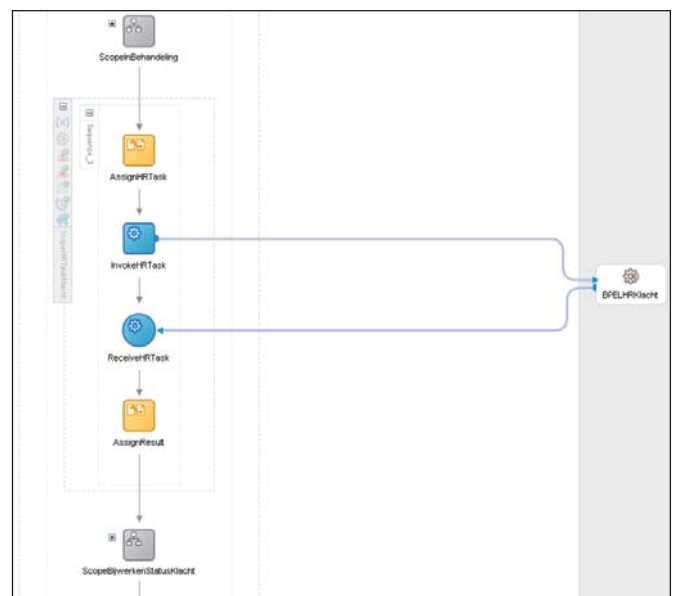
In afbeelding 7 zijn in meer detail de processtappen binnen de scope 'ScopeHRTaskKlacht' van bedrijfsproces 'Registreren Klacht' te zien, waarbij ook duidelijk de aanroep naar service 'BPELHRKlacht' zichtbaar is. Het betreft hierbij een zogenaamde asynchrone service, wat betekent dat de taak die deze service uitvoert, minuten of zelfs dagen kan duren. Het antwoord (response) kan met ander woorden even op zich laten wachten.

ESB Console

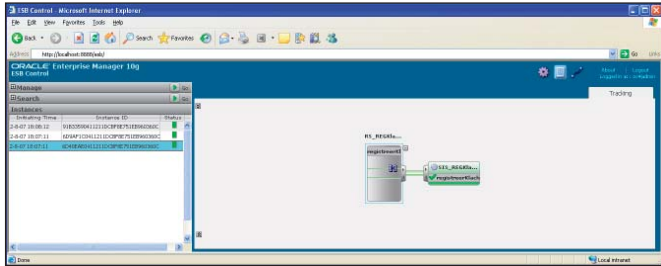
Via de ESB Console kunnen de instances (aanroepen) van de diverse services bekeken worden. Zo is te zien dat de routing service 'RS_REGKlacht', aangeroepen vanuit het bedrijfsproces 'Registreren Klacht' (BPEL), zonder fouten is verlopen, m.a.w. de klacht is geregistreerd in de database.

BPEL Console

Via de BPEL Console kunnen de BPEL processen die op de applicatieserver zijn gedeployed onder andere worden beheerd en getest. Tijdens het testen kan de flow door de processtappen exact worden gevolgd (zie afbeelding 9). Door te klikken op een processtap (bijvoorbeeld 'ReceiveHRTask' binnen de



Afbeelding 7. Aanroep van service 'BPELHRKlacht'



Afbeelding 8. ESB Console

scope 'ScopeHRTaskKlacht') kunnen de details daarvan worden bekeken. Zo is te zien dat er 1 minuut na een asynchrone aanroep van de 'BPELHRKlacht' service een antwoord is ontvangen (zie afbeelding 10).

Door te klikken op de 'View xml document' link is het bijbehorende bericht te achterhalen:

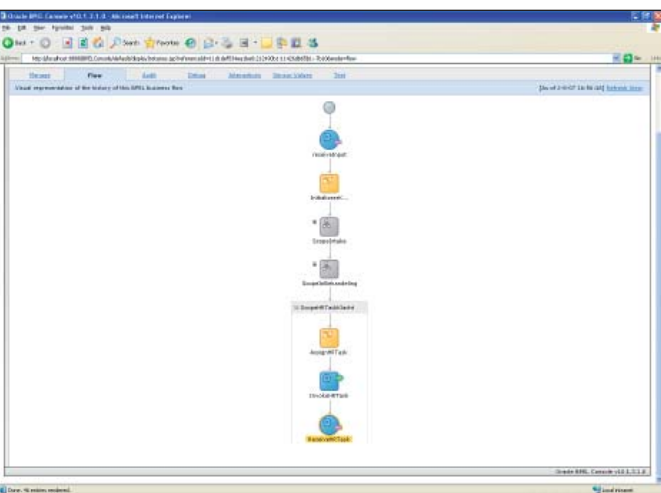
```
<?xml version="1.0" encoding="UTF-8" ?>
<BPELHRKlachtProcessResponse xmlns="http://xmlns.oracle.com/BPELHRKlacht">
  <result>Goedgekeurd</result>
</BPELHRKlachtProcessResponse>
```

Hieruit is op te maken dat de ambtenaar van de gemeente de klacht heeft goedgekeurd.

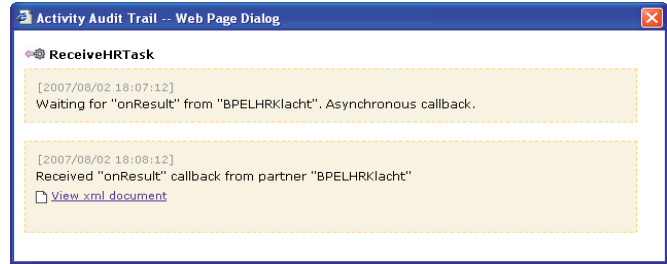
Samenvatting

In dit artikel wordt het procesintegratie deel van de SOA Architectuur van Oracle getoetst.

Eerst wordt gekeken wat de concepten van een Service Oriented Architecture zijn.



Afbeelding 9. BPEL Console



Afbeelding 10. Processtap details

De basiskennmerken van een Service Oriented Architecture (SOA) zijn flexibiliteit en ontwerpen met het oog op veranderingen. Om dit te bewerkstelligen wordt gebruik gemaakt van services, die gebaseerd zijn op standaardisatie, ontkoppelbaarheid, waarden toevoeging en hergebruik. De services worden daarna gebruikt om andere services en processen samen te stellen. Er wordt zo een hiërarchie van services samengesteld. Daarna wordt gekeken welke componenten noodzakelijk zijn om een SOA omgeving mogelijk te maken, en wat de implementatie van Oracle is van deze componenten.

Dit artikel toetst de BPEL en ESB componenten. Een van de belangrijkste redenen om een ESB in te zetten, is dat het daarmee mogelijk wordt om bestaande applicaties of software componenten aan te spreken als service.

De toetsing is uitgevoerd met een integratiescenario uit de ketenintegratie overheid. De door Capgemini uitgevoerde Proof of Concept is goed uit voeren met de Oracle SOA Suite. Oracle is enorm hard aan het inzetten om een van de voorlopers in de SOA markt te worden, door initiatieven te nemen in het definiëren van onder andere de Service Component Architecture. De SOA Suite zal de komende jaren daarmee in continue beweging zijn, waarmee de mogelijkheden voor de gebruikers eindeloos lijken. Risico van deze snelle beweging is echter de stabiliteit van aanwezige functionaliteit, nog sneller dan vroeger zal een upgrade van een ontwikkelde applicatie nodig zijn. Hoe goed zal Oracle de verandering van de verandering ondersteunen?

Referenties

- Oracle Enterprise Service Bus Quick Start Guide, 10g Release 3 (10.1.3), Vimmika Dinesh, Oracle Corporation, August 2006.
- Oracle SOA Suite 10g: Services Orchestration, Glenn Stokol, Bijoy Choudhury, Deepa Pottangadi, Oracle Corporation, March 2007.

De auteurs zijn werkzaam bij Capgemini:

Marc Lameriks

marc.lameriks@capgemini.com

Léon Smiers

leon.smiers@capgemini.com

Alexander Bijl

alexander.bijl@capgemini.com

Joost van der Vlies

joost.vander.vlies@capgemini.com