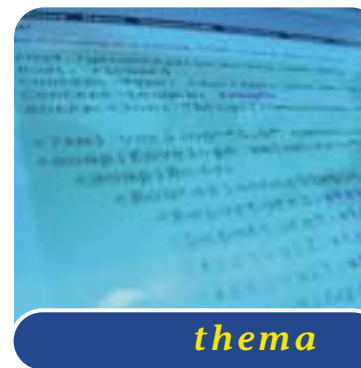


Webservices beloven de opmaat te zijn voor een wereld van servicegerichte integratie. "Dit is het land van melk en honing" hoor ik mensen roepen. Maar wacht eens even! Heb ik dit al niet eerder gehoord? Een mens krijgt toch niet zo maar grijze haren? Na zo'n twintig 'dienstjaren' in de IT-wereld, ben ik nogal grijs geworden en heb ik dit inderdaad al vaak gehoord.



Een wereld van servicegerichte integratie

Rendement en risico's van webservices

Vanuit een optimistisch pragmatisme heb ik natuurlijk wel een kijkje achter de schermen genomen. Wat is er nieuw? Biedt het rendement? En waar schuilen de gevaren? In dit artikel neem ik u mee door het webservices landschap, begeven we ons langs de randen van ravijnen en nemen we zelfs een duik in die utopische oceaan van totale integratie. Er is al genoeg geschreven over de parels op de bodem van die oceaan, dit artikel zal slechts één parel aanraken en zich verder concentreren op de gevaren.

WEBSERVICES VOOR DUMMIES Wat is een webservice? In eenvoudige bewoordingen betreft het een stuk software dat op afroepbasis informatie beschikbaar maakt middels standaard internettechnologie. Geef mij de weersvoorspelling voor Assen voor de komende drie dagen? "Regen, regen, regen", luidt het antwoord. Wat is mijn aandeel- en nabestelniveau van Bartje-beelden? "20.000 en 10", luidt het antwoord.

Maar dit werd met CORBA, DCE en DCOM jaren geleden al beloofd. Dus wat is er dan nieuw? Welnu, het enige dat nieuw is en webservices tot een onderscheidende manier van informatie uitwisseling maakt, is dat hier sprake is van wereldwijde leverancierssupport. Zelfs het 'kleine' Microsoft ondersteunt het. De industrie is de schaamte voorbij en stapt nu naakt in hetzelfde bed. Als jonge geliefden met een set standaarden die ze nu met trots dragen. Zo zijn SOAP, UDDI, en WSDL een paar van de standaarden die iedere nieuwbakken webservice programmeur met trots uitdraagt. In tegenstelling tot de kleren van de keizer dienen deze ook het doel. De indus-

trie is met een hele reeks oplossingen gekomen om snel webservice oplossingen te bouwen en te implementeren. Als u de advertenties mag geloven, dan kan zo'n beetje iedereen dit binnen een uur realiseren. Stel dat het meer dan een uur vergt. Waarom zouden we het doen? Wat zijn de voordelen? Waar ligt het rendement?

WEBSERVICE PARELS Het feit dat we over een (nogal eenvoudige) standaard beschikken, maakt het mogelijk een aantal dingen te doen:

INTEGRATIE Het integreren van ongelijksoortige systemen naar een deken van naadloze informatie is nu in theorie mogelijk. De laatste jaren is het aantal systemen en hun complexiteit dramatisch toegenomen en integratie is uitdaging nummer één op de agenda van de meeste CIO's. Integratie is nodig om het bedrijfsleven van samenhangende, kwalitatief hoogwaardige management informatie te voorzien. Twee ERP systemen binnen één organisatie koppelen, of tussen organisaties. Informatie uitwisselen tussen systemen, informatie publiceren op het web of binnen een portal, of zelfs het orkestreren van een volledig bedrijfsproces zou nu mogelijk zijn door gebruik te maken van een servicegerichte architectuur (Service Oriented Architecture ofwel: SOA). Het onmogelijke wordt nu mogelijk! Webservices zijn voor EAI (Enterprise Application Integration) wat vrachtwagens zijn voor de logistiek. Het feit dat alle leveranciers akkoord zijn gegaan met een standaard, op hoog niveau, helpt enorm. De snelwegen en verkeerstekens zijn nu hetzelfde, zodat de trucks met gemak in iedere richting op de informatiesnelweg kunnen rijden.

IT KOSTENREDUCTIE Integratie en interfaces vormen een grote post binnen elk IT budget. Vaak vallen ze formeel echter niet onder die noemer. Dat maakt ze zo gevaarlijk en moeilijk te reduceren. Zelden zult u een kopje tegenkomen zoals "integratiekosten: 300.000 Euro". Maar we weten allen dat het een groot deel van onze tijd in beslag neemt. Kauwgom en cellotape interfaces zijn de vloek binnen IT-afdelingen in termen van onderhoudskosten. En dan zijn daar nog de kosten van de ontwikkeling van de interfaces. 'Quick & dirty' interfaces bezaaien de straten van ons IT-landschap zoals plastic biercups in Amsterdam op de ochtend na Koninginnedag.

Met webservices is het mogelijk om een dienst één keer te ontwikkelen en deze vervolgens te laten gebruiken door een aantal andere applicaties. Jawel, daar zijn weer de zalige voordelen van hergebruik! Zal het voordeliger worden om nieuwe applicaties te ontwikkelen met gebruik van webservices? Ik betwijfel het. Wel kunnen onderhoudskosten worden gereduceerd als gevolg van de centrale positie van webservices. En integratiekosten zullen beslist afnemen.

VERLENGING LEVENSDUUR VAN LEGACY Het is nu mogelijk om met gebruik van dit webservice model de levensduur van bestaande legacyapplicaties te verlengen, waarbij het rendement van deze oude systemen toeneemt (legacy is alles ouder dan twee jaar...uw ERP systeem zou dus eveneens binnen deze groep kunnen vallen). Complexe functionaliteit die ooit binnen een mainframe is begraven en achter een groen scherm verborgen zit, kan nu aan andere toonaangevende web applicaties binnen en buiten de organisatie worden aangeboden.

FLEXIBILITEIT De mogelijkheid van 'plug & play', het verplaatsen en hergebruiken van functionaliteit in verschillende applicaties verhoogt de flexibiliteit van

matieservice bieden? Wie wil er niet makkelijke toegang verlenen tot informatie vanaf een willekeurige plek naar een willekeurige plek? De early adopters zijn al vooruit gerend en hebben een weg door de jungle gekapt. De sceptici volgen voorzichtig. Nu het pad iets duidelijker is geworden en de opgewonden mist rond de hype langzaam optrekt, kunnen we het bos van de bomen onderscheiden.

DE RISICO'S Waar moeten we dan bang voor zijn? Waarom gevreesd, er is toch een industrie standaard! Niet zo snel. De weg is nog steeds moeilijk begaanbaar en de berm begint bezaaid te raken met lijken.

WEBSERVICE SPAGHETTI Met slechts tien webservices in gebruik zal er geen probleem zijn. Zet de klok nu eens drie jaar vooruit. U hebt tien systemen in huis die ieder tien webservices bieden plus tien externe systemen die ieder twintig webservices bieden. Iedere webservice op een iets andere manier geïmplementeerd, met een verschillende beveiligingsconfiguratie, toegangsbeheer en monitoring mechanisme. Het gebrek aan administratieve consistentie tussen providers en de 'overall' complexiteit zal het risico van mislukken vertienvoudigen. Alleen voor het in de lucht houden van dit alles is al een tovenaars nodig. HELP!

De kosten zullen de neiging hebben om te stijgen op een niet-lineaire wijze aangezien ieder nieuw systeem afhankelijk wordt, direct of indirect, van een set van bestaande systemen.

SLECHTE PERFORMANCE Met zoveel links/koppelingen in de schakel zullen 'tracking & tracing' problemen met performance en crashes een nachtmerrie worden. Het met de vinger naar elkaar wijzen tussen partijen zal politieke proporties aannemen. Denk maar eens aan dit scenario: U start uw webapplicatie en klikt op de functie om een offerte te maken. Het duurt drie minuten voordat u enige informatie heeft! Of nog erger: u krijgt een time-out waarschuwing. U baalt deze keer nogal, omdat het inmiddels de zevende keer is dat zich dat die dag voordoet. U belt de helpdesk. Zij bellen de applicatie manager. Hij belt de technische persoon verantwoordelijk voor de applicatie. Na een uur 'tracking & tracing' laat de technische persoon weten niets te kunnen doen, aangezien de webservice, die de prijsinformatie levert, niet werkt. Er wordt contact opgenomen met de eigenaar van de 'pricing' webservice. Die zegt dat zijn webservice prima werkt voor andere applicaties en dat hij er niets aan kan doen. De systeembeheerder wordt gebeld en zegt op zijn beurt dat het netwerk prima functioneert. Hij kan er verder ook niets aan doen! HELP!

Ondanks de constatering dat webservices snel volwassen lijken te worden, hebben we het 'wilde westen' nog lang niet verlaten

het bedrijfsleven en reactiesnelheid van de IT-afdeling op nieuwe informatiebehoefte. De mogelijkheid om verbinding te maken met een externe bron en de 'real-time' grondstofprijs via een webservice op te vragen, zou de wijze waarop een prijsmodel werkt, kunnen veranderen. Dat klinkt goed! Waarom beginnen we dan niet meteen webservices te bouwen? Wie wil er niet kosten reduceren? Wie wil er niet een meer flexibele infor-

Vermenigvuldig dit typische probleem nu eens met vijftig applicaties, ieder met tien webservices, en vermenigvuldig dit vervolgens met twintig externe providers met ieder nog eens twintig webservices. Opnieuw HELP!

Webservice transacties verschillen van traditionele transactie omdat de uitvoering ervan zich over langere tijd kan uitstreken. Dit maakt het garanderen van transactionele integriteit een behoorlijke uitdaging. De semantiek van traditionele ACID (atomicity, consistency, isolation, durability) transacties zijn niet geschikt voor exploitatie op web-schaal.

Tot op heden werken er slechts een paar bedrijfskritische applicaties op webservices of op een Service Oriented Architecture (SOA) en zijn er geen richtlijnen of performance maatstaven. De early adopters banen zoals gebruikelijk een pad in niemandsland, dus dienen de meer behoedzamen onder ons te luisteren naar de berichten die uitgezonden worden. HELP!

BEVEILIGINGSRISICO'S Hierin ligt het voornaamste risico van webservices. De webservice architectuur is een inherent andere architectuur en heeft daardoor verschillende uitdagingen. De omschakeling van client-server naar web gebaseerd was niet substantieel verschillend in termen van beveiliging, maar met webservices is het een totaal nieuw spel geworden. Traditionele applicaties zijn connectiegericht, maar webservices zijn "berichtgericht" en missen de garantie van een koppeling tussen partijen. Daardoor werken de traditionele connectiegerichte beveiligingsmethoden niet of onvoldoende. U moet alle aspecten van beveiliging zoals infrastructuur, integriteit, authenticiteit, cohesie en geheimhouding weer opnieuw aanpakken. Voeg daar de virussen en hackers aan toe en u hebt een belangrijke uitdaging in uw portefeuille! HELP!

BEPERKTE REIKWIJDTE Terwijl er webservice standaarden binnen de communicatielaag van iedere organisatie bestaan, kan iedere organisatie de standaarden verschillend interpreteren binnen de taxonomielaag. Een jungle van ietwat verschillende standaard interpretaties kan het bereik van webservices beperken. Variërende semantiek en schema's tussen applicaties kunnen het bereik eveneens beperken. Mijn werkorder is uw inkooporder, maar de twee formaten zijn verschillend. "Mijn systeem vraagt om werkorders van het formaat X, maar uw inkooporder is van het formaat Y". HELP!

GEBREK AAN FLEXIBILITEIT Een kleine verandering in een webservice kan zomaar negen applicaties om zeep helpen. Het kennen van de impact en het

managen van de verandering van één enkele webservice zal een uitdaging zijn, laat staan het proberen dit te doen voor vele webservices. En als webservices niet alleen worden gebruikt om 'tegenaan te praten', maar ook om applicaties mee te bouwen, wordt het probleem honderdmaal verergerd. Elke noodzakelijke verandering in de applicatie kan dan impact hebben op externe communicatie en vice versa. Verder zal een migratie naar nieuwe releases een nachtmerrie worden als het

De industrie is de schaamte voorbij en stapt nu naakt in hetzelfde bed

niet juist beheerd en bestuurd wordt. In de wereld van vandaag zijn applicaties veelal op zichzelf staand en zijn de externe interfaces helder gedefinieerd. In de wereld van webservices zijn de mogelijkheden van calls en externe connecties oneindig en het gebruik ervan of de impact op de omgeving moeilijk of onduidelijk. HELP!

De gevaren lijken serieus, maar zijn niet onoverkomelijk. Ik denk dat de voordelen van standaardisering van informatie-uitwisseling en het beschikbaar maken van "services" voor eindgebruikers sterk tegen de risico's opwegen. Bovendien kunnen risico's altijd worden gemanaged. Mits op een gestructureerde, professionele en serieuze manier aangepakt, kunnen webservices eenvoudig hun nut bewijzen voor de organisatie. Hier volgen een paar stappen om op het juiste pad te blijven en de eerder genoemde ravijnen te vermijden.

UITDAGINGEN BEGRIJPEN Ja, er zijn veel uitdagingen, maar ieder groot brein in ons IT universum draait juist hier op. Met het industriebrede akkoord ten aanzien van webservices in het algemeen en een gecoördineerde drive voor standaarden bestaan er reeds oplossingen om de uitdagingen aan te gaan. Het meest belangrijke is de uitdagingen te begrijpen en vervolgens in staat te zijn een gestructureerde aanpak in gang te zetten om deze uitdagingen aan te gaan.

BEVEILIGING Zoals reeds eerder genoemd vormt beveiliging rondom webservices een totaal nieuwe wereld. De 'messaging'-aard van webservices vraagt om totaal verschillende oplossingen. Beveiligingsmedewerkers en zelfs architecten binnen IT-afdelingen tasten vaak in het duister en kunnen wel eens verrast worden als zaken in het donker beginnen te exploderen! De drie aspecten die dienen te worden geadresseerd zijn:

Aanpak

Cohesie is de sleutel naar het managen van webservice beveiliging. Dit betekent dat de verschillende beveiligingssystemen en architecturen binnen een heterogene omgeving op een gecentraliseerde wijze is geor-

en biedt een solide basis voor message integriteit, authenticatie en geheimhouding. Het implementeren van deze standaarden is een must, willen we in staat zijn om beveiligingsinformatie uit te wisselen tussen communicerende applicaties.

Eén ding lijkt zeker: welke standaard u ook verkiest, u kiest zeer waarschijnlijk niet de juiste

ganiseerd en gecoördineerd moeten worden. Iedere organisatie zal vrijwel zeker over legacy applicaties beschikken, die geschikt zijn gemaakt voor webservice:

- enterprise applicaties zoals ERP;
- CRM-oplossingen die webservice interfaces hebben
- externe webservices.

Al deze applicaties zullen hoogstwaarschijnlijk verschillende beveiligingsschema's, gebruikers, rollen, privileges en domeinen hebben. Een raamwerk voor security-beheer en gestructureerde aanpak om al deze beveiligingssystemen te coördineren is van vitaal belang.

Standaarden

De OASIS groep coördineert de ontwikkeling van nieuwe standaarden. De WS-Security specificatie die gestart is door IBM, Microsoft en VeriSign omvat de SAML (Security Assertion Markup Language) standaard

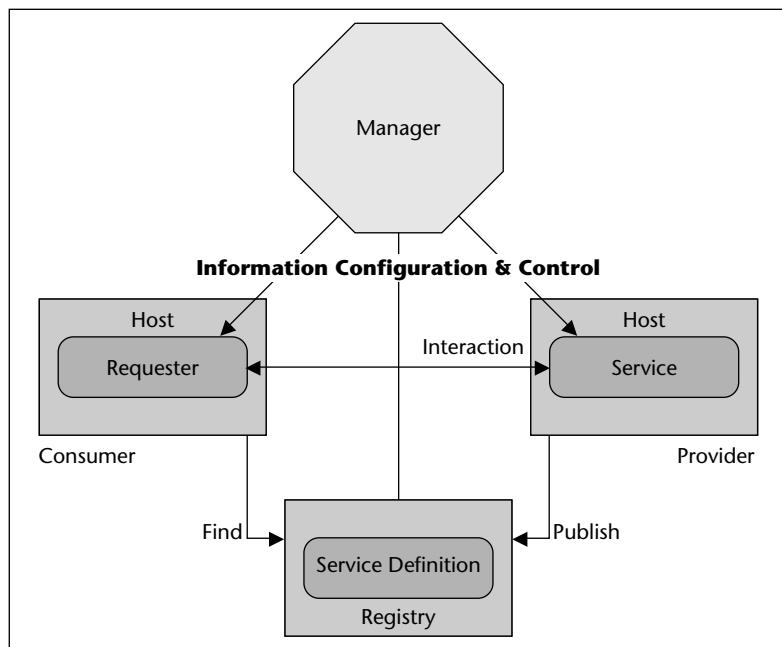
Beveiligingsarchitectuur

Een complexe beveiligingsarchitectuur kan worden opgezet door de semantische beschrijving van de service los te koppelen van het feitelijk leveren van deze service, met andere woorden: het loskoppelen van de dienstverlening van het dienstbeheer. U kunt de toegang tot uw webservices op verschillende manieren beheren voor verschillende afnemers. U kunt de service eenmalig in bedrijf stellen en managen op basis van de klanten en niet op basis van de content.

Een goede beveiligingsarchitectuur dient het volgende te ondersteunen:

- Message geheimhouding door middel van encryptie en geheimhouding schema's op transportniveau;
- Message authenticatie door middel van digitale handtekeningen, eigendomsbewijs, authenticatie op transportniveau en beveiligingsverklaringen en
- Sleutelmanagement door middel van interne sleutels, XMKS services en 'Java key store architectures'.

De beveiliging van webservices is een uitdaging op een nieuw gebied, maar het is robuust en volwassen genoeg om te beginnen met een organisatiebrede toepassing ervan. Bovendien wordt de weg die voor ons ligt verlicht door briljante geesten, die iedere dag met dit onderwerp bezig zijn.



FIGUUR 1. Binnen webservices bestaan drie, aan elkaar gerelateerde verantwoordelijkheidsgebieden

MANAGEMENT Het succesvol realiseren van nieuwe stadswijken vereist een goede planning en een strakke besturing van alle elementen in de procesketen. Het realiseren van een webservice infrastructuur lijkt veel op het bouwen van een stadswijk, inclusief het verbinden daarvan met de bestaande stad en het langzaam verplaatsen van functies naar die nieuwe stadswijk. Het toepassen van zogenaamde 'frameworks' voor webservice management zijn hierbij cruciaal. Traditioneel meten management tools de toestand van het netwerk en de prestaties van afzonderlijke onderdelen. Ze zijn nauwelijks in staat om oorzakelijke verbanden te leggen. In het geval van webservices is het kunnen meten van service levels van vitaal belang.

Zo kunnen losse componenten geheel in orde zijn, terwijl de beoogde service (als keten) niet tot stand komt. Management tools voor de monitoring van 'services' moeten deze top down kunnen besturen. Ze moeten de kwaliteit van de geleverde service meten, en niet alleen de beschikbaarheid van het netwerk. Ze moeten

derhalve informatie kunnen interpreteren en combineren over de gezondheid van de infrastructuur én van de applicaties.

De mate van succes van webservices binnen een organisatie is recht evenredig met de mate van adaptatie van de tools om deze goed te kunnen beheren.

Het beheer van webservices kan als volgt worden onderverdeeld:

System Management

Dit omvat monitoring en logging, alarmering, verbruiksmeting en facturatie, controle en rapportage, SLA management, exception management, verkeersmanagement, uitvoer van security-beleid en oorzaak-gevolg analyses.

Life-cycle Management

Dit omvat versiebeleid, afschrijvingsmodellen, afhankelijkheden en configuratie management. Daarnaast bestaan er drie, aan elkaar gerelateerde verantwoordelijkheidsgebieden binnen webservices:

1. Service monitoring en rapportage
2. Service uitvoering en management
3. Management van de service omgeving

Met de huidige beschikbare gereedschappen is het bouwen en uitbaten van webservices niet erg ingewikkeld. Totdat je simpele vragen gaat stellen als:

- Wat gebeurt er als de service 'crashed'?
- Wat gebeurt er als alle bronnen, benodigd voor de service, in gebruik zijn?
- Wat gebeurt er als de performance van de service daalt tot onder de gestelde normen?

Het is van cruciaal belang dat ontwerpers hun applicaties en oplossingen vanaf het begin ontwerpen inclusief aspecten rondom management. Daarmee wordt management een integraal en getest onderdeel.

Er bestaat een raamwerk van acht principes dat kan worden toegepast bij het ontwerpen en exploiteren van een webservice framework.

- 1) Weet welke informatie je van het service platform verlangt en realiseer deze vanaf het prille begin;
- 2) Dwing een scheiding af tussen het beheer van webservices en de zakelijke drijfveren achter de realisatie ervan;
- 3) Adopteer SOAP-berichten voor service management in plaats van een scala aan messaging protocollen;
- 4) Realiseer een alternatieve route naar het webservices platform;

- 5) Creëer een centraal beeld van de gehele operatie en performance;
- 6) Maak onderscheid tussen vraagstukken over management en vraagstukken over webservices;
- 7) Geef de hoogste voorrang aan management vraagstukken;
- 8) Minimaal de invloed van webservices in het besturen van die webservices.

Gelukkig werken leveranciers van service tools samen met die van service management tools om te komen tot goede oplossingen voor het beheren van webservices. Op dit moment zijn er een aantal tools op de markt, die een goed framework bieden voor webservices management. Bijvoorbeeld Computer Associates en Blue Titan.

Het succes van een webservices initiatief wordt in belangrijke mate bepaald door het op voorhand onderkennen van de noodzaak van dergelijk gereedschap.

ARCHITECTUUR Een van de belangrijkste peilers onder succesvolle webservices is het werken vanuit een doordachte Service Oriented Architecture, ofwel SOA. Momenteel is SOA de implementatiewijze, die veel organisaties toepassen om hun integratiedoelstellingen te realiseren. Bij de opzet van een robuuste SOA dient men over vier aspecten te beslissen:

1. Service transport

Dit is de laag waarop verschillende services met elkaar communiceren. De twee keuzen hier zijn HTTP of communicatie op basis van messaging, middels daarvoor geschikte middleware. HTTP wordt voornamelijk gebruikt voor de communicatie binnen één organisatie. Voor separaat te koppelen B-to-B toepassingen biedt vooral de message-gebaseerde opzet de robuustheid die wordt verlangd bij 24x7 werken de enterprise applicaties.

2. Service protocol

Dit is de 'taal' die door services wordt gebruikt om met elkaar te communiceren. In geval van een eigen, vaste semantiek ligt de voorkeur bij SOAP. Vaak biedt SOAP echter onvoldoende mogelijkheden: het lijkt nogal eens op een discussie over quantum mechanica via het gebruik van middelbare school Engels. Afhankelijk van de verlangde interactie kan ook gebruik worden gemaakt van meer geavanceerde protocollen als ebXML of RosettaNet. En binnen organisaties zelf blijken nogal eens maatwerk-protocollen nodig.

3. Service interface

Deze interface bepaalt hoe een service er uitziet voor hen die deze willen gebruiken. API-gebaseerde interfaces bieden mogelijkheden voor 'early binding', ter-

wijl documentgebaseerde interfaces 'late binding' mogelijk maken. Daarnaast blijkt het verstandig om de service interface ook te kunnen bepalen buiten de implementatie.

4. Service interactie patroon

Dit aspect definieert de wijze waarop services samenwerken. Hoe worden verschillende services bestuurd en hoe verloopt de communicatie tussen de bestuurde en besturende services?

Deze vier aspecten vormen het fundament van de SOA-architectuur. Vervolgens bouwt men voort naar de keuzen die worden gemaakt op applicatieniveau over zaken als 'session management', security, transactie en 'context sharing'.

*In de drie weergegeven gebieden **security, management** en **architectuur** schuilen vrijwel alle risico's van webservices. Bij een gestructureerde aanpak hiervan zijn organisaties veelal goed in staat om de voordelen van webservices te plukken.*

CONCLUSIE Ondanks de constatering dat webservices snel volwassen lijken te worden, hebben we het 'wilde westen' nog lang niet verlaten. Er dient nog veel

te gebeuren voordat deze technologie haar beloften kan waarmaken. Het grootste obstakel is dat er zich teveel organisaties bezig houden met standaardisatie. Iedere van deze organisaties produceert 'standaarden', met als gevolg dat er zelfs een organisatie zich bezig houdt met de definitie van de standaard waarop standaarden dienen te worden geïntegreerd. Dit maakt het beeld wel erg troebel. Toch vormen de voordelen van webservices het fameuze 'low hanging fruit' in B-to-B toepassingen, complexe integraties en informatieverbreiding. In een gecontroleerde omgeving kunnen webservices tegemoet komen aan de vraag om hogere service levels en verlaging van de kosten van het beheer van complexe omgevingen.

Eén ding lijkt zeker: welke standaard u ook verkiest, u kiest zeer waarschijnlijk niet de juiste.

Toch is dat beter dan niets te ondernemen en daarmee voorbij te gaan aan kostenbesparing enerzijds en nieuwe zakelijke kansen anderzijds.

Philip Abernathy

De heer Abernathy is medeoprichter en algemeen directeur van Wisdom Information Consultants.



BEA Systems introduceert WebLogic Platform 8.1

BEA Systems, leverancier van applicatie-infrastructuursoftware, benadrukt haar sterke positie in de markt met de volledige beschikbaarheid van WebLogic Platform 8.1, WebLogic Workshop 8.1, WebLogic Integration 8.1 en BEA WebLogic Portal 8.1. Het platform en alle afzonderlijke producten zijn, evenals de eerder al beschikbare WebLogic Server 8.1 en WebLogic JRockit 8.1, via het web beschikbaar (www.bea.com). De nieuwe producten zijn gebaseerd op de door BEA ontwikkelde softwarearchitectuur ontwikkeld op J2EE, XML en web-services-standaards. Deze architectuur convergeert applicatieontwikkeling met applicatie-integratie tot een eenvoudig en flexibel platform voor applicatie-

infrastructuur. Het WebLogic Platform 8.1 van BEA is gebaseerd op de nieuwste applicatieserver van BEA en BEA's Java Virtual Machine. Het Platform bevat een framework voor enterprise portals en applicatieontwikkeling van BEA, en een zeer innovatieve integratieoplossing. Het WebLogic Platform 8.1 maakt gebruik van wereldwijd toegepaste technologie voor applicatie-infrastructuur en een eenvoudig te gebruiken infrastructuurplatform voor applicatieontwikkeling, integratie, uitrol en beheer. Met WebLogic Integration 8.1 introduceert BEA een nieuwe benadering van applicatie-integratie (EAI), data-integratie en business process management (BPM). WebLogic Integration 8.1 voorziet klanten van een nieuwe architectuur voor snelle

assemblage en integratie van applicaties, data, bedrijfsprocessen en partnernetwerken. WebLogic Integration 8.1 levert met de WebLogic Workshop-ontwikkelomgeving en een reeks standaardadapters, een goede productieomgeving voor integratiespecialisten en applicatieontwikkelaars. WebLogic Workshop 8.1 levert een servicegerichte ontwikkelomgeving voor het bouwen, uitbreiden en koppelen van webapplicaties aan andere toepassingen. WebLogic Workshop biedt een nieuwe manier van flexibele applicatieontwikkeling met een uitgebreide beheeromgeving voor snelle applicatieopbouw en de inzet van best practices.

WebLogic Portal 8.1 is een van de eerste portals met de nieuwe Java portletstandaarden voor ontwikkeling voor de inte-

gratie van contentmanagementsystemen. WebLogic Portal 8.1 biedt een nieuwe ontwerp- en ontwikkelomgeving voor portals, die gebaseerd is op WebLogic Workshop. Daarbij biedt WebLogic Portal 8.1 nieuwe functionaliteiten voor contentmanagement, collaboration, search, webintegratie en ondersteuning van draadloze toepassingen. Bezoek www.bea.com/products voor meer productinformatie.

Tegelijkertijd brengen Information Builders en dochteronderneming iWay Software een nieuwe reeks producten uit voor het vernieuwde WebLogic platform. Het betreft WebFOCUS voor BEA WebLogic, WebFOCUS Open Portal Services for BEA WebLogic

Lees verder op pagina 31