

Welke belangrijke nieuwe J2EE platform functionaliteit komt er dit jaar in voldoende mate beschikbaar? Waarom zouden we daar in moeten investeren? Welke valkuilen en best practices kunnen we hier al onderscheiden? In dit artikel voor ontwikkelaars en hun managers wordt door Cap Gemini Ernst & Young Java community leider Henk Kolk een beeld geschetst van een aantal ontwikkelingen.



# J2EE Trends 2003

## Adaptatie door markt van nieuwe functionaliteit

Om te bepalen of een nieuwe ontwikkeling belangrijk is, wordt gekeken naar voldoende beschikbaarheid van nieuwe functionaliteit. Zo wordt nieuwe technologie pas interessant als verschillende leveranciers producten met deze functionaliteit op de markt hebben. Daarnaast is het belangrijk voor de acceptatie dat er al een aantal eerste referentie implementaties in Nederland gedaan zijn. Bovendien is het goed om een indruk te verkrijgen van de impact van deze ontwikkelingen: waarom is het belangrijk?

De volgende drie ontwikkelingen voldoen aan bovengenoemde criteria en worden in dit artikel uitgelicht:

- Presentatie tier: J2EE Portal Server producten en Portlet specificatie
- Connectivity: Worden web services eindelijk mainstream?
- Business tier: Door de container geregelde persistentie

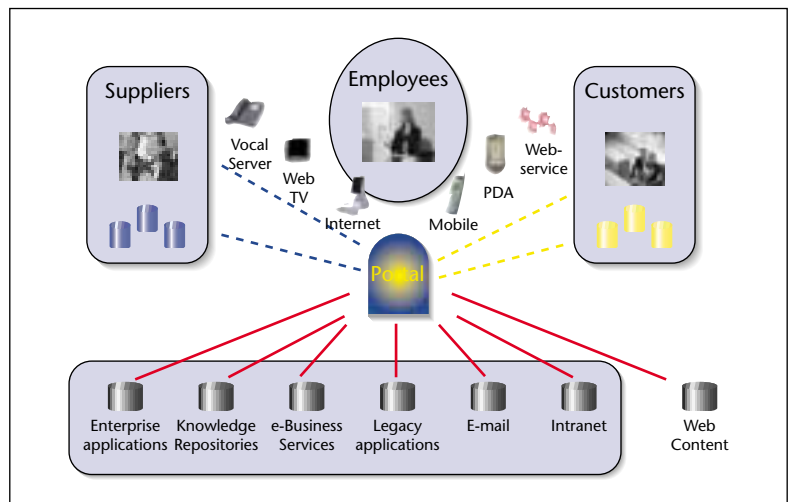
**PRESENTATIELAAG** Wat is een Java Portal Server? Waarschijnlijk de belangrijkste ontwikkeling voor Java dit jaar is de grootschalige introductie van J2EE Enterprise Information Portal Servers.

Een portaal is een centrale architecturele component die applicaties en informatie beschikbaar maakt voor eindgebruikers, onafhankelijk van het type apparaat. Typische doelgroepen voor portalen zijn toeleveranciers, werknemers en klanten. Er zijn een groot aantal portaal producten te koop op de markt, waarvan een aantal op Java gebaseerde portal servers.

In een Java Portal Server wordt de user interface samengesteld uit kleinere content - en applicatie web

componenten (Portlets). Een 'Portlet' vervult hiermee een andere rol dan de welbekende 'Servlet'. Zo kent een Portlet drie 'Window states' (normal, minimized en maximized) en een aantal Portlet Modes (view, edit, help, configure). In een aantal implementaties is de Portlet een subclass van de welbekende Servlet.

**VOORDELEN** Waarom Portlets? Met de mogelijkheid om een user interface samen te stellen uit Portlets met 'Window states' en 'Portlet Modes' speelt de Portlet een unieke rol in het realiseren van een desktop metafoor. Hiernaast heeft de Portlet een personalisatie en security interface, waardoor toegang, plaats van de Portlet op het scherm, look and feel, authenticatie en autorisatie op eenduidige wijze door de Portlet Container geregeld kunnen worden.



FIGUUR 1. Een portaal maakt applicaties en informatie beschikbaar voor eindgebruikers



FIGUUR 2. Typische doelgroepen voor portalen zijn toeleveranciers, werknemers en klanten

Vanuit business perspectief kunnen bedrijfsprocessen voor verschillende doelgroepen worden samengesteld uit 'taken' die eenvoudig vertaald kunnen worden naar Portlets. Omdat portlets los van look en feel, autorisatiemechanisme, taal en plaats op het scherm gemaakt kunnen worden, ontstaat zo een verzameling taak-componenten die herbruikbaar zijn over verschillende bedrijfsprocessen en (deel) organisaties heen. Ook leveren pakketleveranciers een verzameling portlets om pakketten te ontsluiten op taakniveau via Java Portal Servers. Voorbeelden hiervan zijn SAP en Siebel. Ook Content Management Systemen kunnen op deze wijze zowel de publicatie van content als het creatie- en beheerproces via een Java Portal ontsluiten. Een aantal CMS leveranciers stelt al portlets beschikbaar.

**STANDAARDISATIE** De meeste business cases voor Enterprise Information Portals zijn gestoeld op kosten besparing: hergebruik van componenten waaronder Portlets van pakket leveranciers, besparing in de bedrijfsprocessen door enkelvoudig invoer van gegevens en standaardisatie. Zo zijn er organisaties met over de honderd (soms duizend) verschillende intranet oplossingen met evenveel beheerders, software en hardware waarbij de implementatie van één standaard portal product een grote besparing op kan leveren. Andere business cases richten zich op afnemers en hofleveranciers en het creëren van meerwaarde in de keten, met dito hogere marges. Weer andere portalimplementaties richten zich op kostenbesparing in het HRM segment: door gepersonaliseerde pensioen-, lease- en arbeidsvoorwaardengegevens eenvoudig toegankelijk te maken kunnen vele fte's aan telefonische ondersteuning bespaard worden.

De portalserver kan tevens een belangrijke rol spelen als presentatie framework. Veel zaken die organisaties niet, of met een eigen framework geregeld hebben zijn als functionaliteit beschikbaar in een J2EE Portal product. Daarnaast is er per portal leverancier verschillende 'out of the box' functionaliteit beschikbaar. Een voorbeeld hiervan is de zeer uitgebreide set van 'collaboration' portlets van leverancier IBM.

Op dit moment zijn er een groot aantal portalserver producten in de markt, waarvan IBM WebSphere Portal Server 4.1 en 4.2, BEA Portal Server 7.0 en Oracle 9iAS Portal in Nederland wat marktaandeel betreft de belangrijkste producten lijken te zijn. Maar verrassend genoeg levert pakketleverancier SAP ook een portal die aan Java standaarden voldoet. Wellicht is er met name voor de overheid ook een rol weggelegd voor het Open Source product JetSpeed van Apache.

## BEST PRACTICES

1. *Houdt rekening met release planning van productleverancier en JSR 168*

JSR 168 Release Planning	
Community Review	apr-03
Public Review	jun-03
Release	aug-03

De Java Specification Request (JSR) 168, waarin de Portlet API eenduidig wordt vastgelegd is nog niet door de community en public review. Dit betekent dat verschillende vendors er eigen implementaties op na houden en dat tot die tijd nog wijzigingen op kunnen treden aan de Portlet API. Hoewel productleveranciers toezeggen tools te zullen leveren die migratie van reeds ontwikkelde portlets faciliteren, zal hier voor portlets die de komende maanden gebouwd worden nog altijd extra werk in zitten. Tegelijkertijd zijn er heldere business cases voor portal producten te maken, vaak met een ROI van minder dan een jaar die voldoende interessant zijn om nu al met een portal implementatie te starten. Een incrementele werkwijze die rekening houdt met de release planning van JSR 168 en die van de productleverancier is een overweging waard.

2. *Richt een aparte R&D stream in voor nieuwe release van productleverancier*

Enterprise Information Portals worden door onderzoeksbureaus gezien als belangrijke trend, met mogelijkheden voor omzetgroei voor productleveranciers.

Als gevolg hiervan is de concurrentie groot, en de snelheid waarmee nieuwe (tussen)releases op de markt komen ook. Het risico op bugs in deze portalproducten is als gevolg van deze druk ook behoorlijk aanwezig: Denk hierbij niet alleen aan de development- en productie-omgeving, maar ook aan diverse migratietools. Zo zijn er bij een aantal productleveranciers onvolkomenheden bij het deployen van Portlets vanuit de development omgeving naar de acceptatie- en productie-omgeving. Een best practice is het inrichten van een parallel R&D spoor voor iedere nieuwe productversie om te borgen dat de introductie van de (soms maandelijkse) nieuwe release het development proces zo min mogelijk verstoort.

### 3. *Manage verwachtingen van gebruikers: Stick to the framework*

Een andere valkuil heeft te maken met het managen van de verwachting van de gebruikers. Een portalproduct heeft een vaste structuur, waaraan autorisatie, authenticatie, personalisatie en vele andere zaken zijn opgehangen. Deze structuur heeft als voordeel dat er een standaardmanier van navigeren mogelijk is, en als nadeel dat de standaardmanier dus de manier van navigeren is. Gedetailleerde gebruikerswensen op het niveau van menustructuur, navigatie en afwijkende tabblad indelingen kunnen niet altijd gehonoreerd worden zonder de onderliggende engines van het product aan te passen. Dit laatste is voorlopig onverstandig omdat er nog vele nieuwe releases van het product zullen volgen, en eigen aanpassingen aan het portal product een vorm van legacy zijn met hoge aanpassingskosten. Aangezien business cases voor Enterprise Information Portals vaak met kostenbesparing te maken hebben is het laten stijgen van de kosten niet goed. Een best practice is het vroegtijdig betrekken van eindgebruikers bij het idee dat 'Stick to the framework' uiteindelijk de laagste kosten en meeste functionaliteit oplevert. Een andere best practice is het vroegtijdig aanpassen van corporate look and feel standaarden omdat de portalstructuur en navigatie wat zullen afwijken van deze standaarden.

### 4. *Creëer een visie over de rol van de Enterprise Information Portal in je architectuur*

Een Enterprise Information Portal is een component dat de betekenis van één project verre kan overschrijden. De portal verdient een rol in de Enterprise Architectuur van de organisatie, en ondersteunende betekenis voor het realiseren van de missie van de organisatie. Het ontsluiten van primaire bedrijfsprocessen van toeleverancier tot afnemer, het ondersteunen van samenwerken (collaboration) en employee self-service modellen en de manier waarop de organisatie omgaat met informatie zijn mogelijke ingrediënten voor deze visie.

**CONNECTIVITY: WEBSERVICES** Een ander fenomeen in 2003 is de start van de grootschaligere adoptie van Microsoft .Net. Als direct gevolg hiervan zien we in grotere organisaties de eerste bedrijfsonderdelen waarin .Net de primaire rol speelt, en bedrijfsonderdelen waarin J2EE een dominante rol speelt. Budgetbeperkingen en het implementeren van een enterprise architectuur in grotere organisaties helpen ook bij het enkelvoudig implementeren van functionaliteit als services. We

Breng eindgebruikers het idee bij dat 'Stick to the framework' uiteindelijk de laagste kosten en meeste functionaliteit oplevert

zagen vorig jaar het gebruik van web services al gestaag toenemen, en verwachten dat dit in 2003 doorzet. Het gebruik van webservices is wat bouwkosten betreft relatief goedkoop: de IDE doet in de meeste gevallen al het werk.

### **BEST PRACTICES**

#### 1. *Creëer een goede test strategie op web service niveau.*

Vaak worden services en portal door verschillende partijen gebouwd. Zelfs als dit niet het geval is, is het hebben van automatische tests die controleren of de services doen wat is vastgelegd een grote tijdsbespaarder. In het Java-domein kunnen ANT, JUnit en XML Schema's hierbij een belangrijke rol vervullen.

#### 2. *Je klant is belangrijker dan het protocol*

Nieuwe technologische toeters en bellen hebben de neiging de aandacht af te leiden van business oplossing die gemaakt moet worden. Dit geldt voor webservices, maar net zo goed voor portals. Web services vertegenwoordigen slechts een protocol, waarbij het meeste werk in veel gevallen door tools gedaan kan worden.

**BUSINESS TIER** Sinds de introductie van Enterprise Java Bean Containers worstelen Java developers met design, persistentie en performance van EJB's. Het idee is vanaf het begin geweest dat de EJB container (de applicatieserver) het werk zou doen als het gaat om opslaan en ophalen van de 'state' van de business logica. De verwachtingen over de kwaliteiten van persistentie optimalisatie en caching van de container zijn zeker in het begin te hoog geweest, met als gevolg dat er veel verschillende oplossingsrichtingen gekozen zijn om de persistentie en caching goed te regelen. Bij het ontwerpen en implementeren van EJB's zijn verschillende oplossingsrichtingen ontstaan, variërend van: "Gebruik nooit EJB's", "Gebruik alleen Session Beans en geen

Entity Beans”, “Gebruik altijd SQL voor de persistentie en gebruik een DBA voor optimalisatie van de query’s”, tot aan eigen persistentie mechanismen, OO-RDB mapping frameworks zoals TopLink (Nu van Oracle) en Castor en meer ‘pure’ oplossingen als de JDO implementaties en object databases.

Het is waarschijnlijk een illusie te denken dat al deze sporen weer eenvoudig bij elkaar gebracht kunnen worden, maar feit is dat huidige applicatieservers zoals BEA WebLogic Server en Jboss en IBM WebSphere 5.0 zeer snel en flexibel zijn in het persistieren van objecten en hun relaties (Container Managed Persistence (CMP) en Container Managed Relations (CMR)). In de meeste gevallen zien we bij de persistentie door de container een veel hogere performance dan bij handmatig SQL coderen, en vergelijkbare performance als bij OO-RDB mapping frameworks. De meest optimale keuze hangt natuurlijk altijd af van het toepassingsgebied. Wat we ook zien is dat CMP en CMR veel bouw – en debug tijd scheelt.

## REFERENTIES

*JSR 168 (Portlet Specificatie):*

<http://www.jcp.org/en/jsr/detail?id=168>

*JSR 167 (Java Server Faces):*

<http://www.jcp.org/en/jsr/detail?id=127>

*IBM WPS:*

<http://www7b.software.ibm.com/wsdd/zones/portal/>

*BEA WLP:*

<http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/portal>

*Oracle 9iAS Portal:*

<http://oracle.com/ip/deploy/ias/portal/index.html>

*JetSpeed Portal (Open Source):*

<http://jakarta.apache.org/jetspeed/site/index.html>

*Henk Kolk is leider van de Java community bij Cap Gemini Ernst & Young. Voor vragen en opmerkingen over dit artikel is hij bereikbaar via e-mail: [henk.kolk@cgey.nl](mailto:henk.kolk@cgey.nl)*

## PATCHES Patches PATCHES Patches PATCHES Patches PATCHES

### **Systeemintegratie: iWay Software introduceert ‘broker independent integration’**

iWay Software, dochterbedrijf van Information Builders en leverancier van middleware voor de integratie van data en applicaties, introduceert broker independent integration, een reeks softwaretoepassingen die een alternatief bieden voor de traditionele op integration brokers gebaseerde EAI-technieken en tools. Het gebruik van integration broker-producten heeft een aantal grote nadelen. Er ontstaat een uitermate ingewikkelde infrastructuur, gebruikers leggen zich vast in leveranciersgebonden technologie en het gebruik ervan vraagt een dwingende aanpak. iWay Software’s oplossingen geven daarentegen de benodigde flexibiliteit aan EAI zonder dat de gebruikelijke investeringen in tijd en geld nodig zijn. Broker independent integration richt zich op de

oplossing van de moeilijkste vraagstukken van EAI projecten – de technische aansluiting op applicaties en data, en de transformaties die nodig zijn om systemen aan elkaar te plakken. iWay Software’s aanpak omvat XML, webservices en intelligente adapters, die een universele vertaler vormen tussen applicaties. Deze heeft lang niet alle infrastructuur nodig die gewoonlijk in integration brokers productsuites te vinden zijn.

Veel IT organisaties die voor de uitdaging staan om integratie van systemen te realiseren, kiezen voor XML als dataformaat. iWay Software’s broker independent integratieaanpak maakt het mogelijk elk IT-element te benaderen via XML-documenten. Dit heeft tot gevolg dat tools zoals iWay’s XML Transformation Engine informatie uit het ene XML-document kunnen overzetten in het andere zonder dat het nodig is transformatiecode op

maakt te schrijven voor niet-XML API’s. Zelfs niet-standaard systemen zoals legacy databases en ERP-applicaties interacteren met gestandaardiseerde XML-documenten die kunnen zijn ingebed in elke applicatie of bedrijfsproces.

Webservices hebben een centrale plaats ingenomen in de gedachten van veel informatiearchitecten vanwege hun gestandaardiseerde aanpak van het gebruik van XML over het Internet. De XML-functionaliteit van iWay’s broker independent integratietechnologie is conform de laatste standaards voor SOAP and WSDL, zonder dat specifieke implementaties in J2EE of .NET vereist zijn. Dit geeft gebruikers van nieuwe of legacy systemen de vrijheid om onmiddellijk te migreren naar webservices zonder dat significante aanpassingen aan hun architectuur, dure consultancy of training nodig zijn.

iWay’s adapters bieden bovendien technische connectiviteit met in principe elk informatiesysteem, inclusief softwarepakketten zoals SAP en Siebel, legacy databronnen zoals IMS en VSAM, en transactiesystemen zoals CICS en IMS/TM. De iWay adapters valideren en transformeren elektronische handelsdocumenten zoals EDI en XML (inclusief branchespecifieke formats zoals HIPAA, HL7, SWIFT en FIX) in XML-documenten. Het bedrijf concentreert zich op broker independent integratietechnologie, waardoor de adapters toekunnen met veel minder softwarecode dan die van de broker-leveranciers op de markt. iWay’s werkt hiertoe in partnerships samen met onder meer BEA, IBM en Microsoft.