

Biztalking met SQL Server

# De voordelen van een rdbms in de nieuwe economie

Philippe Van der Stighelen

**M**icrosoft betreedt de B2B-arena met Biztalk Server. Deze software controleert elektronische berichtgeving tussen handelspartners en coördineert bedrijfsprocessen. Welke rol speelt SQL Server in dit verhaal? Gewapend met een dba-achtergrond en een jaartje bèta-testen onderzoekt Philippe Van der Stighelen de voordelen van een database bij processen in business to business (B2B).

## MESSAGING EN ORCHESTRATION

Biztalk Server valt in te delen in twee grote blokken: een *messaging*- en een *orchestration*-deel; men spreekt ook wel over enterprise application integration (EAI). Onder messaging verstaan we het ontvangen, transformeren en vervolgens opnieuw verzenden van elektronische documenten in de meest uiteenlopende formaten. Standaard ondersteunt Biztalk Server zowel XML als de Europese en Amerikaanse EDI-formaten EDIFACT en X12. Maar je kunt het zo gek niet bedenken, of Biztalk heeft wel een 'customizeerbare' parser (verwerker) die de gewenste informatie uit het document peurt. Tijdens hun reis door Biztalks communicatiekanalen kunnen documenten bovendien geconverteerd worden van een bepaald formaat naar een willekeurig ander formaat. Hiervoor zorgen mappings, die men definieert en wijzigt via de Biztalk Mapper. Door hun grote flexibiliteit vormen mappings een onmisbaar element in

het B2B-proces. Ze maken het organisaties namelijk mogelijk snel in te spelen op wijzigingen van elektronische documenten bij de handelspartners, als gevolg van fusies of acquisities. Handig is verder dat

*Door hun flexibiliteit vormen mappings een onmisbaar element in het B2B-proces*

mappings worden uitgedrukt in standaard XSLT-technologie. Je kunt dus precies nagaan wat er met je document gebeurt als je in de XSL-code duikt.

Orchestration (letterlijk: orkestleiding - of het coördineren van informatiestromen) maakt het mogelijk bedrijfsprocessen op een grafische manier weer te geven. Figuur 1 toont één van de workflow-diagrammen (die je in Visio kunt aanmaken), die de basis vormen voor orchestration. Naast het controleren van informatiestromen door middel van *if-while-then-logica*, kunnen deze workflows ook COM-componenten of scripts aanroepen. Tijdens het

compileren wordt de workflow op fouten getest, en verkrijgt men een file in XLANG-formaat, de XLANG Schedule. De XLANG Scheduler Engine zorgt vervolgens voor de uitvoering van de XLANG-schedules die men opstart door een instance van de XLANG Schedule aan te roepen in een ASP-webpagina of VBScript. Hoewel messaging en orchestration volledig onafhankelijk van elkaar kunnen functioneren, concentreert de eerste zich op het uitwisselen en routen van berichten, en zijn in orchestration bedrijfsprocessen programmatisch te definiëren.

## STRUCTURELE TRANSFORMATIES

Biztalk kan documenten van een bepaald formaat omzetten naar een ander formaat. Kijken we even naar figuur 2, dan zien we dat een bron- en doelspecificatie in de Biztalk-mapper worden geladen, waarna men via een systeem van *point-and-pull* aangeeft welke informatie uit de bronspecificatie in de doelspecificatie terecht moet

## Internetcentrische document routing

Als twee bedrijven besluiten dat zij langs elektronische weg informatie willen uitwisselen op een gestructureerde manier, moeten zij twee dingen van elkaar weten: hoe het bericht van de tegenpartij te interpreteren, en op welke manier het bericht te transporteren? Om deze informatie vast te leggen, gebruikt men een soort elektronische overeenkomst of *agreement*. Zulke agreements opstellen, beheren en verwerken doe je in een handomdraai met Biztalk, een universeel internetcentrisch document routing-systeem.

komen. Het mappen van informatie alleen volstaat echter niet om tegemoet te komen aan de specifieke vereisten van het brede B2B-spectrum. Meestal zal het ook nodig zijn de informatie uit de bronspecificatie te transformeren, en ook dit is mogelijk met het mapping-tool.

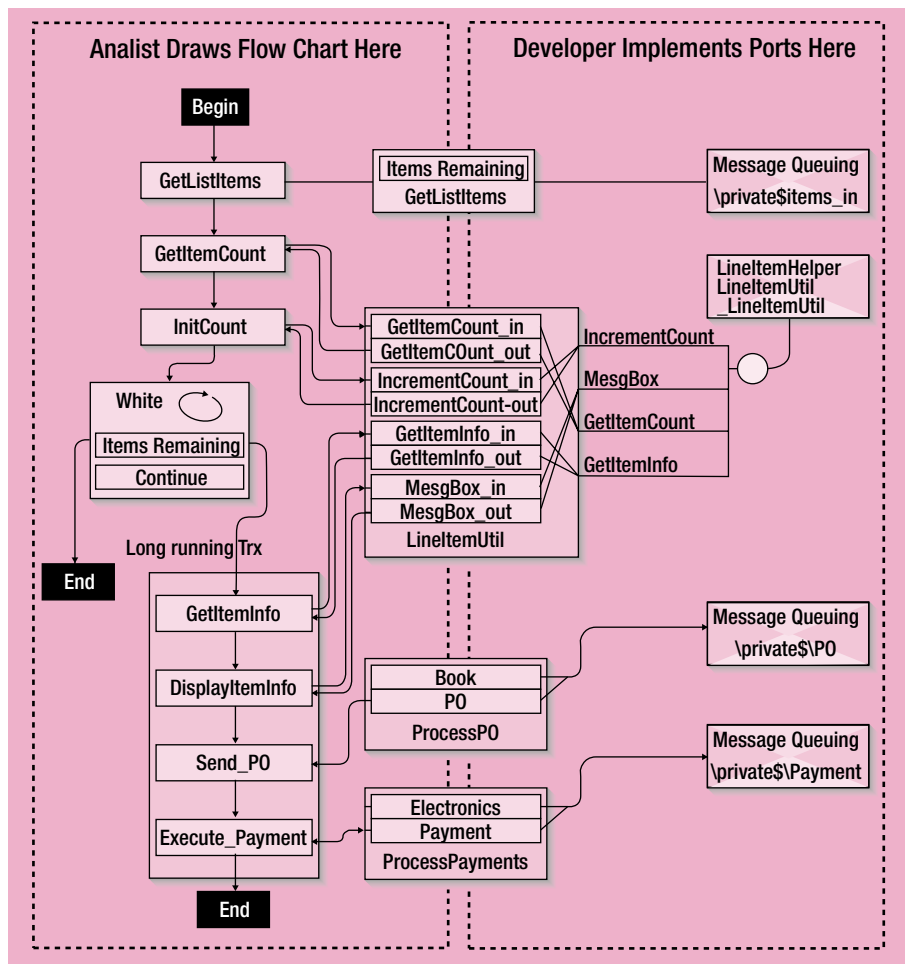
Het uitvoeren van complexe structurele transformaties van records en fields gebeurt door *functoids*. Standaard worden bij Biztalk functoids geleverd voor de meest uiteenlopende transformaties. Door een functoid uit het functoid-palet te selecteren en vervolgens naar de mapping-grid te trekken, kunnen elementen gelinkt worden. Elke functoid bevat een voorgedefinieerd VBScript, en vaak is er de mogelijkheid specifieke waarden, die men *argumenten noemt*, in te geven. Als je bijvoorbeeld een bepaalde waarde met acht wil vermenigvuldigen, kies je de Multiply-functoid en geeft als argument het getal acht in.

**DATABASE-FUNCTOIDS**

Met database-functoids kun je gegevens aan een database onttrekken. De database-Lookup-functoid die zichtbaar is in figuur 2 gaat op zoek naar een bepaalde databasetabel om een specifiek record te vinden. Je kunt de database-Lookup-functoid vergelijken met de welgekende standaard *data source name* (DSN) waarmee je een verbinding legt met een database. De output van de database-Lookup-functoid moet vervolgens verbonden worden met Value Extractor-functoids. Dit is een belangrijk punt, want database-functoids werken samen om data uit een database te halen en moeten in combinatie worden gebruikt. De Value Extractor-functoids halen de gevraagde informatie uit het lookup-record en zijn te beschouwen als de tegenhanger van SQL-query's. Met Value Extractor-functoids is namelijk precies aan te geven welke informatie je uit SQL Server wilt halen.

**UITGEKIEND MECHANISME**

Waarom heeft Biztalk eigenlijk SQL Server nodig, en welke informatie komt in de

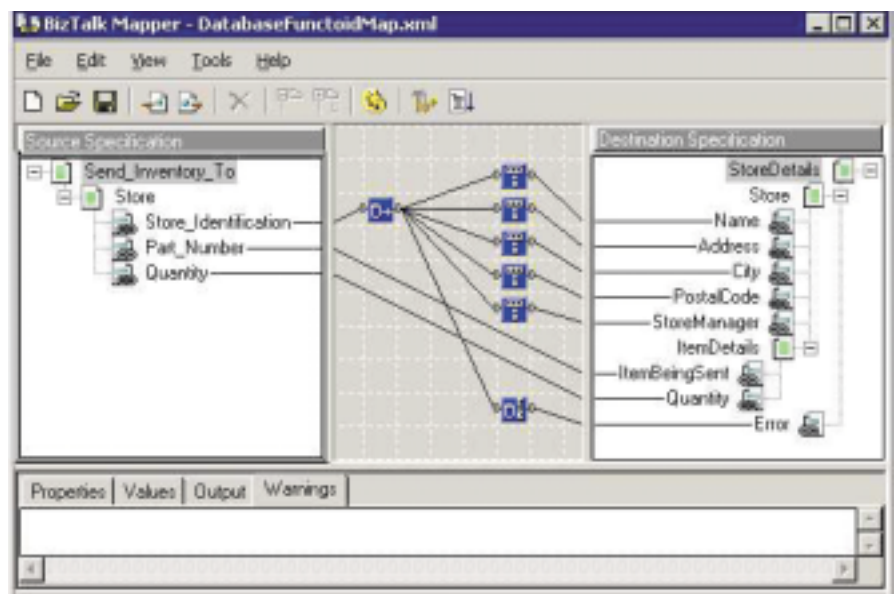


**FIGUUR 1: WORKFLOW-DIAGRAM.**

database terecht? Uit figuur 3 valt af te leiden dat Biztalk vier databases gebruikt. Tijdens het installatieproces van Biztalk worden de volgende databases aange- maakt: InterchangeBTM, InterchangeSQ,

InterchangeDTA en XLANG.

Merk op dat slechts SQL Server wordt aanvaard als dbms; we hebben tenslotte te maken met Microsoft. InterchangeBTM wordt aangewend door Biztalks



**FIGUUR 2: DATABASE-LOOKUP-FUNCTOID.**

## Enkele begrippen

- **XLANG**: een Microsoft-specifieke taal die de logische sequenties beschrijft van bedrijfsprocessen, als ook de implementatie van de bedrijfsprocessen door het gebruik van uiteenlopende applicatiediensten. XLANG wordt uitgedrukt in XML.
- **XSLT**: met extensible Stylesheet Language Transformations (XSLT) is een XML-bericht om te vormen naar een ander XML-formaat of een html-pagina. Ook bestaat de mogelijkheid informatie toe te voegen aan het resultaat.
- **Dehydrate**: het opslaan van statusinformatie voor een XLANG Schedule-instance in een SQL Server-database, terwijl de XLANG Scheduler Engine wacht op de ontvangst van een bericht, en alvorens de volgende actie in het bedrijfsproces uit te voeren.
- **Dehydrate**: het ophalen van statusinformatie voor een XLANG Schedule-instance uit een SQL Server-database, nadat een bericht is ontvangen door de XLANG Scheduler Engine.
- **XML Schemas**: schema's en DTD's (document type definitions) bevatten richtlijnen die men dient te volgen bij het opstellen van XML-documenten. Dezelfde schema's zijn vervolgens aan te wenden om de verkregen XML-files te valideren.

Management Desk, die hierin informatie opslaat in verband met de gebruikte transporttechnologie (http, smtp, msmq) en de organisaties die betrokken zijn bij het uitwisselen van documenten. Om na te gaan welke weg een bepaald document heeft afgelegd tijdens de reis door Biztalk, wordt gebruik gemaakt van document tracking, en deze trackinginformatie kun je vinden in de InterchangeDTA-database. XLANG is de enige database die door Orchestration wordt gebruikt, met name door de XLANG Scheduler-engine, die hierin statusinformatie wegschrijft met betrekking tot bedrijfsprocessen die wachten op een bericht.

Als laatste is er de InterchangeSQ (Shared Queue) -database; deze bevat vier SQL Server-queues die worden gebruikt tijdens de verwerking van documenten: de Scheduled queue, Work queue, Retry queue en Suspended queue.

Figuur 4 toont de Microsoft Management Console (MMC) die is aan te wenden om deze queues te beheren. Documenten worden uit de Scheduled queue gehaald voor verwerking door de eerst beschikbare Biztalk-server en geplaatst in de Work queue. De Scheduled queue verzekert een hoge verwerkingsnelheid door het ter beschikking stellen van server resource management. Is een benodigde service niet beschikbaar voor

de verwerking van een document, dan wordt het document in de Scheduled queue geplaatst, zodat de servers verder kunnen werken met een andere uitwisseling. Documenten die falen wegens verwerkingsfouten, zoals netwerkproblemen of validatiefouten, worden in de Suspended queue of Retry queue geplaatst voor latere pogingen.

SQL Server zorgt dus voor het opslaan van gegevens met betrekking tot messaging en orchestration, en uiteraard maken we handig gebruik van de typische voor-

### **B2B stelt specifieke eisen aan een database en vereist dat statusinformatie van processen wordt beheerd**

delen die een relationele database biedt: beschikbaarheid van de data dankzij regelmatige backups, beveiliging van de gegevens tegen ongewenste toegang en betrouwbaarheid van de data dankzij transactionele integriteit. B2B-processen stellen echter specifieke eisen aan een database: ze zijn van nature langdurig en bestaan vaak uit meerdere componenten. We hebben bijgevolg behoefte aan een uitgekend mechanisme dat statusinformatie

van processen beheert. Dit mechanisme noemt men transactional messaging.

## TRANSACTIONAL MESSAGING

Transacties zijn SQL-commando's die niet noodzakelijk gerelateerd zijn, maar uitgevoerd moeten worden op een manier van alles of niets. Zodra er iets misloopt met een van de commando's die deel uitmaken van een transactie, moet de volledige transactie worden teruggedraaid naar de beginstatus. Op deze manier blijven de gegevens in de database te allen tijde consistent. B2B-processen waarbij bedrijfskritische communicatie via XML en EDI tussen handelspartners centraal staat, vertrouwen uiteraard op de transactionele integriteit die een database hen biedt. Transactionele berichten worden verzonden en ontvangen binnen één en dezelfde transactie. Dit houdt in dat deze berichten hetzij samen worden verzonden, en in dezelfde volgorde (*committed transaction*), of helemaal niet: *aborted transaction*. Men geeft er immers de voorkeur aan dat bij een probleem de informatie verwijderd en opnieuw verzonden wordt, boven de ontvangst van onjuiste of onvolledige informatie.

Maar er is meer: Biztalk maakt een onderscheid tussen *short-lived transactions* en *long-running transactions*. Een bericht verzenden tussen twee bedrijven is een short-lived transaction: deze transactie moet absoluut slagen of falen. Een bedrijfsproces daarentegen is een long-running transaction; een volledige order in een geautomatiseerd aankoopstelsel bijvoorbeeld.

Zoals figuur 1 laat zien, fungeert een long-running transaction als een container voor short-lived transactions, en soms kan men zich genoodzaakt zien een dit soort transacties te stoppen voordat zij volledig zijn. Gebeurt dit, dan is het nodig de short-lived transactions die deel uitmaken van de long-running transaction te compenseren. Een XML-document kan bijvoorbeeld gemapt en omgezet zijn naar EDI, waarna het automatisch is verzonden naar een leverancier, die daarop materiaal

en een factuur heeft teruggestuurd en op betaling wacht. Indien nu een vraag tot annulering volgt, moet de long-running transaction worden geannuleerd, en zijn compensatie-acties te nemen voor alle reeds uitgevoerde short-lived transactions. Bijvoorbeeld: men moet materiaal terugsturen en een creditnota opstellen.

Langzamerhand wordt nu ook duidelijk waarom de database zo belangrijk is als

**Biztalk loopt probleemloos en ziet er goed uit, maar daaraan zijn enkele zenuwslopende bètaversies voorafgegaan**

enabler voor deze langdurige transacties. SQL Server wordt namelijk gebruikt om statusinformatie op te slaan (*hydrate*) voor XLANG Schedule-instanties die wachten op een XML- of EDI-bericht. De statusinformatie, opgeslagen in de database, wordt later gebruikt om de XLANG Schedule-instances te activeren (*dehydrate*), zodra het bericht beschikbaar is.

**OPTIMALISATIE**

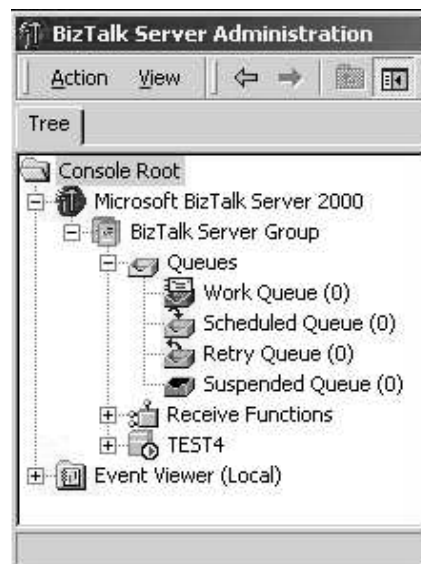
Een door Biztalk behandeld bericht brengt een kettingreactie teweeg van verschillende processen:

- er wordt een XML-schema uit de repository gehaald voor validatie;
- een map wordt gebruikt voor transformatie;

- en hierbovenop komt dan nog uitgebreide logging en tracking, om op elk moment te weten waar het bericht zich bevindt, en in welke staat.

Zo kan bij een drukke uitwisseling van berichten de load op SQL Server best hoog zijn. De Biztalk-databases functioneren optimaal indien ze elk afzonderlijk geplaatst worden op een server met een processor van 733 MHz of meer. Zo'n opzet zorgt ervoor dat alle resources van de server aan de databasetransacties worden gewijd, maar impliceert ook een belangrijke investering, aangezien vijf verschillende servers nodig zijn. Dit druijft uiteraard in tegen de consolidatietrend die vele bedrijven toepassen. Hoe dan ook, in een productie-omgeving plaats je liever geen enkele database op de server waarop Biztalk Server 2000 geïnstalleerd staat.

Ook toekomstige uitbreidingsmogelijkheden moeten beschouwd worden bij de initiële opzet van Biztalk. Een organisatie die slechts een klein aantal documenten verwerkt, zal idealiter een server kiezen die gemakkelijk valt uit te breiden. Indien de verwerkingscapaciteit nadien uitgebreid moet worden, kan de Shared Queue-, Tracking- en Orchestration-database worden verplaatst naar een andere server. Het wordt echter ten stelligste afgeraden de Biztalk Messaging Management-database te verplaatsen. Het is dus zaak ervoor te zorgen dat je de Biztalk Messaging Management-database bij de installatie op een server plaatst met voldoende opwaarderingsmogelijkheden. Dit zowel als het gaat om schijfruimte als geheugen, om performance-caching mogelijk te maken



**FIGUUR 4: QUEUES IN BIZTALK-ADMINISTRATOR.**

wanneer de hoeveelheid transacties stijgt.

De Tracking-database vraagt een snelle netwerkkaart, 100 Mbps of meer, vanwege het hoge volume aan ongecomprimeerde data die worden uitgewisseld tussen Biztalk Server en de database. En nu we het toch hebben over uitwisseling van data, een laatste tip: houd rekening met de beperking van 20 Mb bij documenten die Biztalk moet verwerken. Wordt dit maximum overschreden, dan heeft dit een negatieve invloed op de prestatie van Biztalk. Rekening houdend met de gemiddelde grootte van XML- en EDI-documenten die men inmiddels dag gebruikt bij B2B-transacties, zal deze grens echter niet meteen worden bereikt.

**DOEMSCENARIO**

Fouttolerantie is extreem belangrijk voor een allesomvattend systeem als Biztalk. Bij het uitvallen van deze laatste verlies je namelijk niet alleen de uitwisseling van elektronische documenten met je handelspartners, maar ook de interne bedrijfsprocessen gemodelleerd in XLANG vallen stil. Het is dan ook niet moeilijk hierbij de meest afschrikwekkende doemscenario's te bedenken.

Voor bedrijven die Biztalk hebben gekozen voor messaging, orchestration of beide, is het noodzakelijk de databases te

**Onderhoud noodzakelijk**

Twee van de vier databases vragen van de dba wat meer aandacht dan gebruikelijk: de Tracking- en de Orchestration-database. Beide kunnen zeer snel in omvang toenemen, en onderhoud is hier noodzakelijk. Als bij regelmatige tussenpozen het meegeleverde script `DTA_SampleJobs.sql` wordt uitgevoerd, blijft de Tracking-database gezond. Dit script voert namelijk een soort grote schoonmaak door, waarbij onder meer tussentijdse records die wachten op een bericht maar waarvan de wachttijd is verstreken, worden verwijderd. Ook de Orchestration-database kan snel opzwellen tot een omvang die de performance bedreigt. Jammer genoeg heeft Microsoft hiervoor geen scripts meegeleverd. Dit wordt echter wel voorzien bij toekomstige versies van het product. En in afwachting daarvan kun je terecht op [www.microsoft.com/biztalk](http://www.microsoft.com/biztalk) voor richtlijnen voor het onderhoud van de Orchestration-database.

# Systeembenodigheden

Om Biztalk Server te installeren is minimaal de volgende hard- en software nodig:

- 300 Mhz processor en 128 Mb geheugen;
- minimaal 6 Gb aan schijfruimte;
- Windows 2000 Server met SP1;
- SQL Server 7.0 met SP2 of SQL Server 2000;
- Visio 2000 SR-1a;
- Internet Explorer 5.0 of hoger.

repliseren naar een backup-server. De procedure is eenvoudig: maak een kopie van de databases, verplaats deze naar een andere server, en configureer automatische replicatie tussen de SQL- en de backup-server. Met dit mechanisme is er steeds een consistente kopie beschikbaar die men kan gebruiken bij datacorruptie in een van de productie-databases. Ook de reparatieprocedure ligt voor de hand: Biztalk-messaging en XLANG engine services moeten stoppen, waarna de corrupte database kan worden vervangen door de replica. Na het opnieuw opstarten van de services kan Biztalk-server opnieuw aan de slag, en bovendien zal het eerder beschreven mechanisme van transactional messaging ervoor zorgen dat half afge-

werkte berichten worden verwijderd en opnieuw verzonden.

Voor wie naast datareplicatie ook een uptime van 99,99 procent wenst te benaderen, kan clustering uitkomst bieden. Biztalk ondersteunt clustering en load balancing in de enterprise-versie. Je dient dan wel de software op Windows 2000 Advanced Server te installeren, samen met SQL Server Enterprise-editie. Met deze opzet kun je tot 32 machines connecteren voor load balancing, en zodra een van de Biztalk-servers in het cluster uitvalt, neemt de andere clusternode in een fractie van een seconde de Biztalk-diensten over. Hier blijft echter gelden dat de databases die zich bij een clusterconfiguratie normaal gesproken op de shared RAID-array bevin-

den, bij voorkeur gerepliceerd worden naar een backup-server.

## UITDAGING

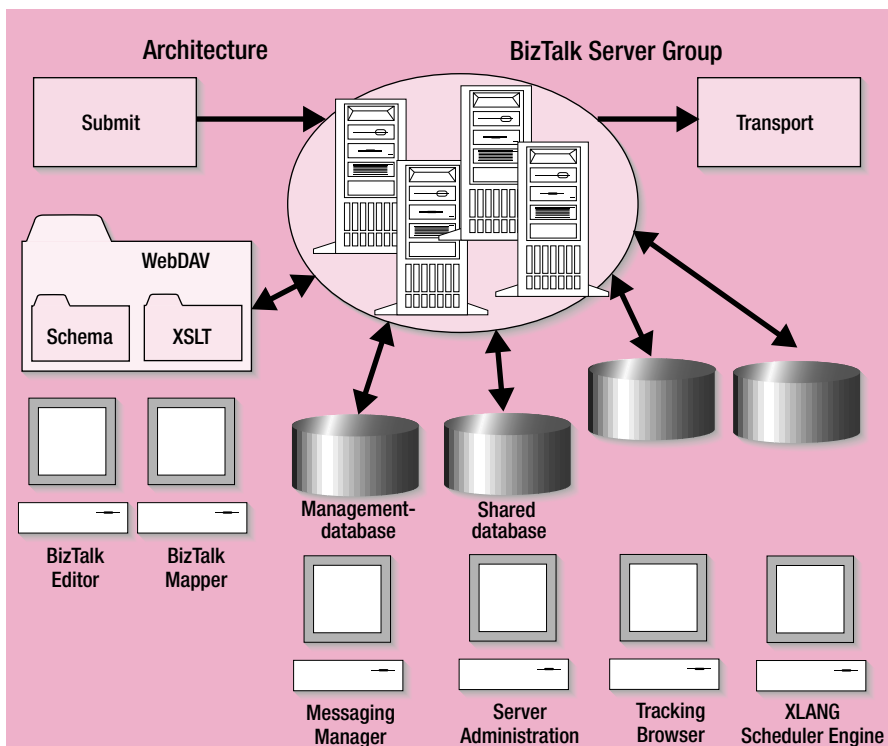
Biztalk werd al twee jaar geleden aangekondigd. Microsoft heeft het geduld van menigeen dus enigszins op de proef gesteld. Er zijn enkele zenuwslopende bètaversies aan voorafgegaan, die opvielen door hun gebrek aan stabiliteit en cryptische foutmeldingen, met ontelbare herinstallaties als gevolg. Maar de uiteindelijke versie heeft deze kinderziekten overwonnen, loopt probleemloos en ziet er goed uit.

*Microsoft voegt met Biztalk aan zijn .Net-gamma een belangrijke en zelfs onmisbare component toe*

Tot nog toe was Microsofts software gericht op interne bedrijfsinfrastructuur en ontwikkeling. Biztalk is Big Greens eerste product dat zich toespitst op elektronische handel tussen bedrijven. Het wordt een hele uitdaging om een plaatsje te bemachtigen op de B2B-markt, waar sinds enkele jaren grote spelers zoals webMethods of Silverstream met succes aan het werk zijn. Maar daar staat een sterk punt tegenover: terwijl die leveranciers in hun productgamma een scheiding maken tussen messaging en applicatie-integratie, komt Biztalk geleverd met beide out-of-the-box.

Microsoft voegt met Biztalk aan zijn .Net-gamma een belangrijke en zelfs onmisbare component toe, en men kan zonder overdrijven stellen dat dit product behoort tot de meest complexe software die het bedrijf ooit heeft ontwikkeld. ●

Philippe Van der Stighelen (pvanderstighelen@vintura.com) is MCDBA en werkzaam bij Vintura als e-business solution architect. Binnen de solutions-groep houdt hij zich bezig met B2B-integratieprojecten gebaseerd op Biztalk en webMethods.



FIGUUR 3: BIZTALK GEBRUIKT VIER DATABASES: INTERCHANGEBTM, -SQ EN -DTA EN XLANG.