

Dit artikel handelt over de toepassing van applicatie integratie met behulp van een Integration Broker, een alles-regelende spin in het web, in dit geval BizTalk Server 2000 (BTS) in een Windows NT4/2000 omgeving in combinatie met Microsoft Message Queueing. Het accent in dit artikel zal liggen op MSMQ. Daarbij baseren de auteurs zich op een praktijkimplementatie van een Integration Broker omgeving bij een grote verzekeringsmaatschappij.

Allereerst wordt ingegaan op de architectuur van MSMQ. Vervolgens gaan de auteurs dieper in op het fenomeen adapter. Tenslotte behandelen zij enkele problemen die zij tegengekomen zijn en de oplossingen die daarvoor zijn ontworpen.



thema

# Het web van applicatie integratie

## *MSMQ en BizTalk beyond the basics*

In ons vorige artikel<sup>1</sup> stelden we dat Enterprise Application Integration nog steeds een belangrijk punt van aandacht hoort te zijn. Applicaties binnen een organisatie (of zelfs erbuiten) gebruiken steeds meer gegevens van elkaar. In het verleden werden daartoe vaak specifieke interfaces gemaakt tussen de applicaties. Hierdoor is veelal een wirwar van verschillende interfaces tussen systemen ontstaan. Deze wirwar is erg moeilijk te onderhouden en erg onoverzichtelijk. Door gebruik te maken van een zogenaamde Integration Broker kan al deze communicatie gestructureerd en gecentraliseerd worden. Tevens maakt het de algemene bruikbaarheid van gegevens een stuk groter. Minstens zo belangrijk zijn de communicatielijnen. Deze moeten voor een betrouwbare verbinding zorgen tussen applicaties en Integration Broker, en vice versa.

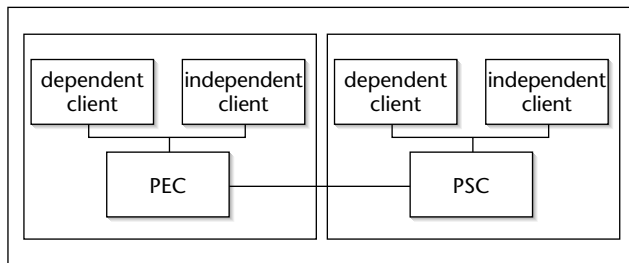
**ARCHITECTUUR VAN MSMQ** MSMQ bestaat uit een hiërarchische structuur van servers en clients<sup>2</sup>. Boven aan die structuur bevindt zich de primary enterprise controller, PEC. Het is de centrale opslag van queue locaties en de servers en clients die toegang hebben tot de queues en berichten. Merk op dat bij Windows 2000 deze informatie wordt opgeslagen in de Active Directory, terwijl bij Windows NT4 de informatie wordt opgeslagen in SQL Server. Onder de PEC bevinden zich de primary site controller, PSC's. Een site controller

heeft kopieën van delen van de enterprise informatie. Een site wordt meestal bepaald door fysieke grenzen, bijvoorbeeld een gebouw of een vestiging van een bedrijf. Onder de site-controller bevinden zich de clients. Een site bestaat dan uit een site-controller en clients.

Clients kunnen worden geïnstalleerd als independent client of als dependent client. Een dependent client kan alleen berichten versturen en ontvangen als zij toegang heeft tot haar PSC. Een dependent client kan alleen synchroon met MSMQ communiceren. In mission-critical applicaties zal over het algemeen geen gebruik worden gemaakt van deze mogelijkheid. Het moet immers mogelijk zijn ten alle tijden berichten te versturen.

Queues zijn de opvang- en verstuurbakjes voor berichten. In MSMQ kunnen deze berichten express of recoverable worden verstuurd. Zoals de benaming reeds doet vermoeden, is de ene manier snel en de andere

- 
- 1 Applicatie integratie in de praktijk, Andy Mater en Gerben Lubbers. Software Release Magazine, nr 1, Februari 2002.
  - 2 Voor de eenvoud worden de Routing Server, voor het routeren van berichten tussen queues, en de Backup Site Controller buiten beschouwing gelaten.



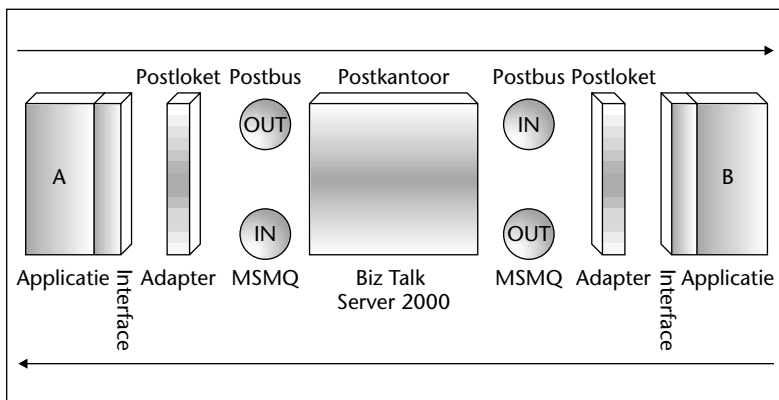
FIGUUR 1

manier kan eventuele calamiteiten overleven. Express berichten zijn RAM-based; zij worden in het werkgeheugen van de computer gehouden. Recoverable berichten zijn disk-based worden fysiek op schijf opgeslagen, deze berichten blijven in dit geval bewaard. Een voorbeeld van een calamiteit is het uitvallen van een server waar nog berichten in een queue staan. In dat geval zullen alle express-berichten ook echt verdwenen

## Door teveel onderlinge afhankelijkheden is een systeem niet toekomstvast

zijn nadat de server opnieuw in de lucht is. Daarentegen, zullen de recoverable berichten na de uitval weer gewoon worden verstuurd.

Het versturen van express berichten gaat sneller, terwijl het versturen van recoverable berichten meer zekerheid biedt. Het is afhankelijk van het soort bericht op welke manier het verstuurd dient te worden. Een voorbeeld van een toepassing van een express bericht is een vraag-antwoord bericht. In dit geval is het vaak niet erg als een bericht verloren gaat. Een typisch voorbeeld van een recoverable bericht is een informatiebericht, zoals het wegschrijven van klantinformatie in een database. In dit geval zal de klantinformatie uiteindelijk altijd in



FIGUUR 2

de database worden weggeschreven. Per soort bericht zal moeten worden bekeken wat de meest geschikte en gewenste wijze van versturen is. Er dient een afweging te worden gemaakt tussen snelheid en mate van betrouwbaarheid.

**ADAPTERS: HET CONCEPT** In het vorige artikel beschreven we reeds dat gebruik is gemaakt van een typisch *loosely coupled*, asynchrone broker omgeving. Applicaties worden middels MOM, Message Oriented Middleware, aan elkaar gekoppeld via een centrale Broker. Een dergelijk opzet maakt het mogelijk dat applicaties kunnen communiceren zonder tegelijk online te zijn. Tevens maakt deze manier van communiceren de applicaties onafhankelijk. In het ons praktijkvoorbeeld betekende dit de combinatie van MSMQ en BizTalk Server 2000.

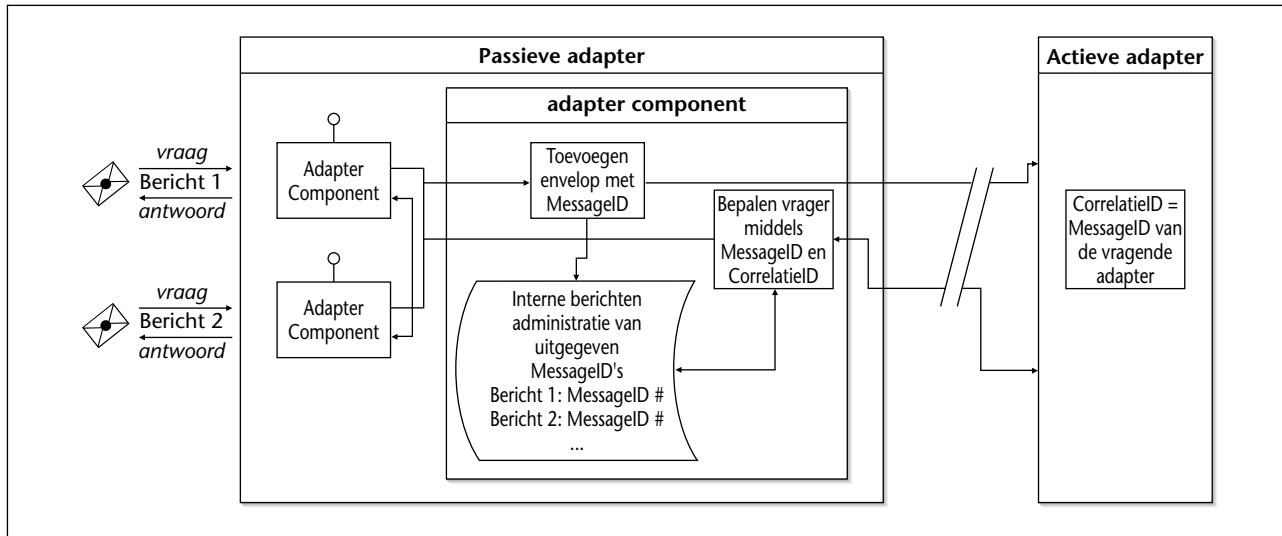
Om complete onafhankelijkheid te garanderen, is gekozen voor het fenomeen adapters. Indien we de integration broker, i.c. BizTalk Server, vergelijken met een postkantoor en de queues met postbussen of postbakjes, kunnen we de adapters zien als postloketten. Een applicatie A stuurt zijn berichten naar een adapter. De adapter zorgt ervoor dat het bericht in het uitbakje wordt geplaatst. Het postkantoor leegt het uitbakje en bepaalt aan de hand van de afzender en het type bericht in welk 'in'-bakje het bericht moet worden afgeleverd. De adapter aan de ontvangende kant leegt het 'in'-bakje en levert het bericht af aan applicatie B.

In ons voorbeeld zijn er twee typen adapters:

- Passieve adapter:
  - Passief wil zeggen dat de adapter slechts dan activiteiten gaat verrichten indien de adapter een bericht krijgt aangeleverd.
- Actieve adapter:
  - Actief wil zeggen dat de adapter 'actief' de betreffende queue staat te pollen op inkomende berichten. De reden voor een actieve adapter wordt verderop in het artikel uitgelegd.

Applicatie onafhankelijkheid is gegarandeerd, omdat applicaties niet direct met elkaar communiceren. Waar de gegevens vandaan komen is voor een applicatie onbekend. Het kan zelfs inhouden dat de broker een deel van de gegevens haalt uit applicatie C en een deel uit applicatie D.

Complete onafhankelijkheid wordt gegarandeerd, omdat applicaties slechts met de adapters communiceren. Mocht in de toekomst voor een andere broker worden gekozen, of zelfs een ander communicatie mechanisme, merken de applicaties daar niets van. De inter-



FIGUUR 3

face naar de adapters blijft immers hetzelfde. De adapter heeft de kennis om met MSMQ te communiceren.

De berichten die worden verstuurd zijn XML berichten. Net als in de 'real world' worden de berichten voorzien van een envelop. De adapters plaatsen de envelop om het bericht. De applicaties leveren de informatie die 'op' de envelop wordt geplaatst. Denk hierbij aan afzender en het soort bericht. De berichten worden echter zonder envelop aan de ontvangende applicatie afgeleverd. Immers, waar het bericht vandaan komt, is voor de applicatie niet van belang.

Hoewel zojuist is gesproken van complete onafhankelijk, is een kanttekening op zijn plaats. Er dienen uiteraard wel afspraken te worden gemaakt omtrent de berichtstructuren die worden uitgewisseld. Applicaties moeten weten welke XML-berichten zij dienen te versturen en welke XML-berichten zij kunnen verwachten. De structuur van die geldige XML-berichten waarmee onderling wordt gecommuniceerd, wordt vastgelegd in BizTalk Server die tevens de controle op zich neemt.

**CORRELATIE VAN BERICHTEN** Een typisch probleem bij simultane vraag-antwoord berichten is om het juiste antwoord aan de juiste vraag te koppelen. Als je een vraag stelt, wil je niet het antwoord op de vraag van je buurman terug krijgen, maar het antwoord op je eigen vraag. Een goed voorbeeld van een applicatie die meerdere simultane vragen kan versturen, is een internet site. Meerdere mensen vragen hun gegevens op via een internet site, maar willen natuurlijk wel hun eigen gegevens als antwoord krijgen.

Reeds eerder is gesproken over een envelop die door de adapter aan het bericht wordt toegevoegd. Door gebruik te maken van die envelop bij elk bericht, worden aan het bericht een aantal kenmerken toegevoegd.

Als door applicatie A een bericht verstuurd wordt zal dit via de passieve adapter gaan. Deze adapter krijgt informatie, zoals de inhoud van het bericht, het soort bericht en de afzender.

De adapter plaatst deze informatie in de envelop. Daarnaast genereert de adapter een unieke GUID code. Een GUID code is een zogenaamde "Globally Unique Identifier". Het is een getal van 128-bits (16 bytes), waarbij een algoritme steeds een uniek getal genereert. Het algoritme is een standaard van de "Open Software Foundation (OSF), Distributed Computing Environment (DCE)". Deze code wordt voor het bericht de message-identificatie (var1), messageID. De adapter plaatst vervolgens het complete bericht, inclusief envelop in een MSMQ queue en onthoudt wat is verstuurd: de interne messageID administratie. Indien de antwoordende applicatie een antwoord heeft op het bericht, zal dit bericht terug worden gestuurd via de actieve adapter. Deze adapter gebruikt dezelfde envelop, maar plaatst de originele unieke messageID in een ander veld, de correlationID. Zodra het bericht terug is bij de passieve adapter, zal deze met behulp van zijn interne adminis-

Voor het versturen van berichten dient men een afweging te maken tussen snelheid en betrouwbaarheid

tratie exact kunnen bepalen aan welke vraagsteller (messageID) het betreffende antwoord gekoppeld moet worden.

De interne messageID administratie kan in de adapter in geheugenvariabelen worden opgeslagen, maar



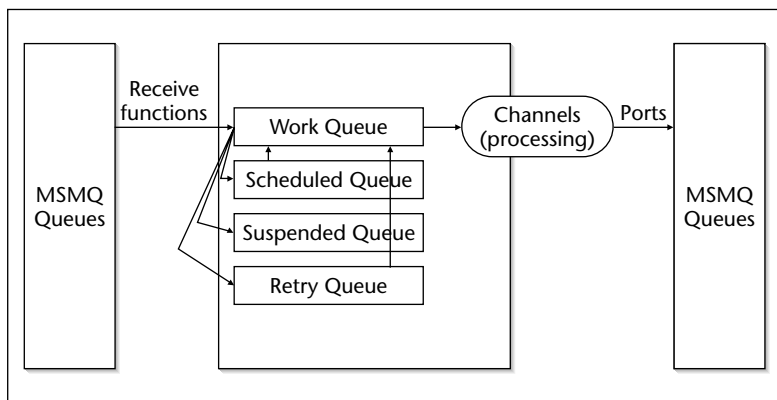
FIGUUR 4

(tijdelijk) opslaan in een databasetabel behoort uiteraard ook tot de mogelijkheden.

Het bericht is opgebouwd met XML (eXtensible Markup Language). XML wordt steeds meer gebruikt als

## Als MSMQ als PSC wordt geïnstalleerd, zet deze zijn IP adres in een tabel van SQL Server

manier van gegevensstructurering. Een Integration Broker is als een spin in het web van diverse verschillende systemen. Al die verschillende systemen hebben hun eigen manier van structuren en opslaan van gegevens. Daarom is het noodzakelijk om bij communicatie tussen deze systemen een algemene manier van structurering te gebruiken, die tevens platform onafhankelijk is. Praktisch alle systemen kunnen met XML overweg (al dan niet via een interface). Binnen BizTalk wordt XML ondersteund, sterker nog: het is binnen BizTalk Server de standaard structureringmethode.



FIGUUR 5

**ADAPTERS: HET ALTERNATIEF** Het is goed mogelijk voor een applicatie om geen gebruik te maken van adapters en berichten direct aan BizTalk aan te bieden. Waarom zouden we deze extra schakel in de communicatie plaatsen?

Met een dergelijke koppeling wordt dan mogelijk niet voldaan aan asynchrone communicatie en het *loosely coupled* zijn van systemen. Alhoewel BTS server over een intern queueing mechanisme beschikt, zal een dergelijke koppeling risicovol zijn. Deze zal mogelijkwijs moeten worden aangepast bij het migreren naar een andere versie of andere leverancier van de Integration Broker. Daarnaast wordt in dat geval het interne queueing mechanisme "misbruikt" voor de opslag van berichten indien en (ontvangende) applicatie niet beschikbaar is.

Asynchroon versturen kan ook door direct middels de Interchange interface die BTS ter beschikking stelt<sup>3</sup>. Via deze COM-interface zijn de functionaliteiten van BTS met behulp van bijvoorbeeld Visual Basic te benaderen. Zo kan met de Submit-functie een bericht naar BizTalk worden verstuurd voor verdere verwerking. Hoewel, indien een applicatie niet beschikbaar is, door BTS een in te stellen aantal malen wordt geprobeerd de berichten opnieuw te versturen, komen deze berichten uiteindelijk in de suspended queue terecht. Alhoewel met dezelfde COM-interface ook de suspended queue van BizTalk kan worden benaderd, zal hiervoor wel een apart mechanisme moeten worden ontworpen om de "wachtende" berichten te kunnen scheiden van de foutieve berichten. Dit kan leiden tot een groot beheersprobleem. Dit probleem treedt niet meer op door gebruik te maken van een extern queueing mechanisme.

Om deze afhankelijkheden zoveel mogelijk te beperken wordt gebruik gemaakt van adapters en een queueing mechanisme, in casu MSMQ.

**LESSONS LEARNED** Om de adapters te laten weten dat nieuwe berichten in de queue van MSMQ aanwezig zijn, werd in eerste instantie de standaard MSMQ arrived event gebruikt om een event te laten triggeren. Binnen de adapter werd vervolgens het event opgevangen en het bericht van de queue gelezen. Gebleken is dat dit onder bepaalde omstandigheden niet goed werkt. Indien binnen de gebruikte omgeving de PSC van MSMQ (de domein-controller binnen een Windows 2000 domein) crasht, is het vervolgens noodzakelijk om de MSMQ service van de server waar de adapter op draaide opnieuw te starten, alsmede de adapter zelf. Dit

3 We spreken dan van een AIC: application integration component.

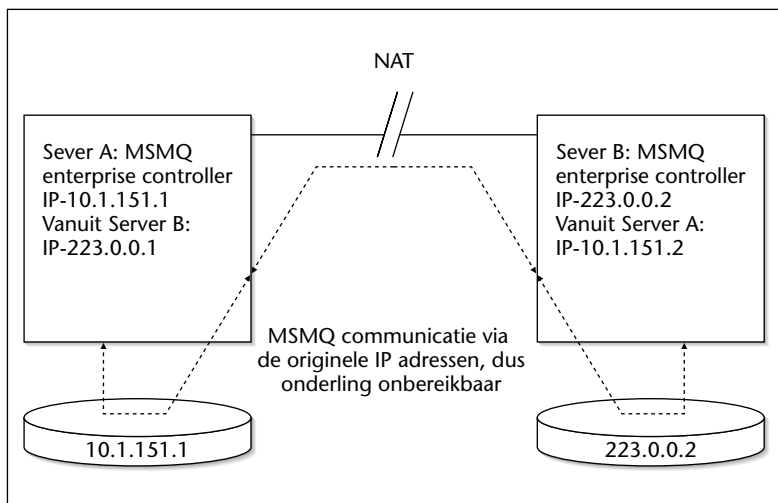
probleem is opgelost door MSMQ geen events te laten triggeren, maar door de adapter periodiek actief te laten pollen op de betreffende MSMQ queue. De frequentie waarmee wordt gepollt is configureerbaar.

Daarnaast is gebleken dat een server waarop het besturingssysteem Windows NT4 in combinatie met service pack 6 is geïnstalleerd onverwachte resultaten kan opleveren binnen MSMQ. Het is beter om in dat geval Windows NT4 en service pack 5 te gebruiken. Voor het voorgaande probleem heeft Microsoft inmiddels een hotfix<sup>4</sup> voor SP 6 ter beschikking gesteld. Deze hotfix dient te worden toegepast indien bij het lezen van de queues resulteert in de volgende melding: "Cannot open queue: The Message Queue Service is not available" of "Cannot open queue: An invalid handle passed to a function".

Ook bleken problemen te ontstaan indien meerdere MSMQ sites in combinatie met Network Address Translation, NAT, wordt gebruikt. In ons voorbeeld werd tussen de twee sites gecommuniceerd over een WAN middels NAT. In een Windows NT4 omgeving geeft dit problemen. Communicatie over netwerken, zowel LAN – LAN als WAN – WAN, gaat over het algemeen via het IP-protocol. Elke computer heeft daarbij een uniek adresnummer; IP-nummer, om elkaar te kunnen zien en met elkaar te kunnen communiceren.

MSMQ op Windows NT4 is afhankelijk van SQL Server. Diverse configuratie gegevens worden in tabellen van SQL Server opgeslagen. Op elke site moet, in geval van NT4, SQL Server worden geïnstalleerd. Als MSMQ als PSC wordt geïnstalleerd, zet deze zijn IP adres in een tabel van SQL Server. Voor replicatie van de MSMQ omgeving dient deze site verbinding te krijgen met de PEC. Indien vervolgens de PSC verbinding probeert te maken met de PEC, wordt het adres van de PSC doorgegeven. Dit adres is het adres dat in de SQL Server tabel staat. Het probleem is dat door het gebruik van NAT het IP adres vanuit de PEC niet zichtbaar is en de verbinding tussen deze twee onmogelijk wordt. Er is daarom besloten versneld over te stappen op Windows 2000.

In de laatste versie van Biztalk Server (2002) zijn een aantal verbeteringen aangebracht. Zo wordt er gebruik gemaakt van Integrated Windows Security bij het benaderen van SQL Server. Verder is het beheer van Biztalk server gemakkelijker gemaakt door gebruik te maken van remote administration. Ook zijn de tools "rondom Biztalk", zoals de Biztalk Mapper en de Biztalk Editor, verbeterd. De eindconclusie voor het gegeven



FIGUUR 6

praktijkvoorbeeld moet luiden, dat MSMQ in combinatie met BizTalk Server a near perfect match vormen.

**TOT SLOT** Een belangrijke conclusie is dat men, alvorens BizTalk Server te gebruiken, een gedegen ontwerp moet maken van de gewenste situatie. Voor dit ontwerp is gedegen kennis van alle onderdelen van de

## Binnen BizTalk Server is XML de standaard structureringsmethode

omgeving een vereiste. Eerst denken en dan pas doen.

Minstens net zo belangrijk als de Integration Broker zelf, is de transportmethode van en naar de Integration Broker. MSMQ is een goed te gebruiken methode. Wel zal een opzet en een architectuur moeten worden gekozen die geheel *loosely coupled* blijft. Door teveel onderlinge afhankelijkheden is een systeem niet toekomstvast.

Ons derde en laatste artikel staat geheel in het teken van de performance en infrastructuur van BizTalk Server en MSMQ. In dat artikel gaan we dieper in op het installeren van een BizTalk/MSMQ omgeving binnen een mission-critical bedrijfsomgeving en het tunen van de omgeving. Tenslotte tonen we de resultaten van ons performance onderzoek.

drs. Gerben Lubbers en drs. Andy Mater  
Lubbers en Mater zijn beiden werkzaam als application engineer voor Cap Gemini Ernst & Young binnen de practice Warp11 (e-mail: gerben.lubbers@cgey.nl / andy.mater@cgey.nl).

<sup>4</sup> Zie ook Microsoft artikel Q251294.