

Vernieuw waar nodig en behoud het goede. Die stelregel wordt in praktisch elke branche gekoesterd, maar niet in de IT-industrie. Omwille van de hype vervangen we de bewezen principes door concepten, waar de nestgeur nog aanzit. SOA, geïmplementeerd als webservice op een ESB creëert één grote monoliet, terwijl we al lang weten dat de doelmatigheid en beveiliging is gediend met ont koppeling van applicaties en data.

opinie

SOAP van real time drukt prestaties

De vergeten principes van message queuing

Het Internet is langzamerhand het algemeen geaccepteerde overslagterrein voor het lossen en laden van informatie. Zo is vanaf dit kwartaal ook het register met naw-gegevens van alle Nederlandse burgers - voorheen aangeduid als GBA (gemeentelijke basisadministratie) - als landelijk raadpleegbare deelverzameling (LRD) op het web te vinden. Elke organisatie kan aan de hand van een sofi-nummer of burgerservicenummer het register raadplegen. Wie een actie voorbereid voor een mail naar 600.000 personen, weet dat het even tijd kost om het juiste adressenbestand samen te stellen uit dit remote register. Door te repliceren, ont koppelen we van het hoofdbestand. We nemen op de koop toe dat de informatie niet op elk moment actueel is. Maar omdat we een abonnement nemen op een update-service, is de periode van onvolledigheid van korte duur.

In het bovenstaande voorbeeld passen we bewust een ont koppeling toe volgens het master-data principe. We nemen de tijd voor synchronisatie van de data. In veel nieuwe systemen, geënt op de ESB (Enterprise Service Bus) is niets meer terug vinden van die ont koppeling. Ergens op het Internet of intranet bevinden zich de gedistribueerde applicaties of data en daar 'callen' we vervolgens synchroon lustig op los via SOAP (Simple Object Access Protocol), Ondanks garanties van het transportprotocol (garanteed delivery) verwachten we dat elke aangeroepen business functie onmiddellijk wordt uitgevoerd. De klant vraagt en de service geeft direct antwoord. Als het dan even tegenzit omdat er niet voldoende verwerkingscapaciteit voorhanden is, of omdat een service niet beschikbaar is, zitten we gelijk met de handen in het haar. De klant ontevreden, de organisatie blameert zich en lijdt reputatieschade.

WEER TERUG NAAR AF Met de SOA (service oriented architecture), geïmplementeerd als webservices op een ESB, zijn we weer terug naar af. In het verleden werden bedrijfsfuncties aan de gebruikers ter beschikking gesteld vanuit grote centrale computersystemen. Ondanks de aanwezigheid van krachtige processoren, een omvangrijke opslagcapaciteit en veel werkstations vormde deze oplossing een monoliet. Viel er ergens een

ESB is in feite niet anders dan een berichtenbus, waarop remote procedure calls worden losgelaten

schakel uit dan was iedereen de sigaar. De oplossing zochten we in ont koppeling, zowel op 'design' tijd als op 'run' tijd. Voor dit laatste werd een queuing-systeem gebruikt. De berichtenqueue biedt een ont koppeling in de tijd. Daarmee scheppen we ruimte voor het uitvoeren van bijvoorbeeld synchronisatietaken in een asynchrone verwerkingsproces. De applicatie krijgt meer scheduling-vrijheid, waardoor taken worden uitgevoerd wanneer er voldoende 'cycles' voorhanden zijn. De computer wordt beter benut en de gebruiker merkt daar zelden iets van. Volgens het 'fire and forget'- principe vuren zij hun vragen af en gaan er vanuit dat hun verzoek wordt uitgevoerd. Wie loopt er direct naar de printer om zijn documenten op te halen?

Helaas, batch is uit. Lang leve SOAP! De SOAP-adepten zijn echter vergeten dat asynchrone verwerking één van de pijlers is van de moderne architectuur. Fraaie

concepten als de druppelbatch of sudderbatch zijn uit de gratie. Doch wat is er mis met een batch-run in een prolongatietoepassing van hypotheek- of verzekeringspolissen aan het begin van de maand? Zolang de batch het online verkeer niet in de weg zit, is er niks aan de hand. Laat het systeem maar lekker aansudder. Een doorlooptijd van dertig dagen moet voldoende zijn. Het batchproces mag natuurlijk niet, zoals in het verleden,

Het gevolg van een dergelijke hapering kan het gehele systeem als een dominospel in elkaar laten storten

alles terroriseren. De batchverwerking moet anderen naast zich dulden, zoals online verkeer en andere asynchrone verwerking. Door middel van 'listeners' luisteren we naar wat er in de queue zit. Zijn er geen vrije 'cycles' voorhanden dan sluiten we de queue af. Wordt het even wat stiller, dan openen we de queue en levert het systeem volle prestaties. Opschaling in hardwarecapaciteit kan achterwege blijven. Ontkoppeling van online-verwerking door het toepassen van inmiddels genegeerde *de facto* standaard middleware als MQseries (IBM) en MSMQ (Microsoft) opent de weg voor 'straight through processing'.

AFSCHIEDEN VAN BOZE BUITENWERELD

Juist nu we vanwege de boze buitenwereld met het ongrijpbare Internet de backoffice met de operationele bedrijfsinformatie graag zo goed mogelijk willen scheiden van de frontoffice, valt aan asynchrone verwerking niet te ontkomen. We creëren immers tussen de twee een gedemilitariseerde zone. Voortdurend zal de voorkant moeten afstemmen met de masterdata aan achterkant. Indien om welke reden dan ook de backoffice niet in de lucht is, blijven de transacties keurig in een queue staan. Als de achterkant stoort, draait de voorkant gewoon door en groeit de queue. Daar is niks mis mee. Telebankieren werkt al jaren zo tot volle tevredenheid.

ESB is in feite niet anders dan een berichtenbus, waarop remote procedure calls worden losgelaten. Als de aange-roepen service niet beschikbaar is, dan laten we de gebruiker nodeloos wachten op een business-functie, waarop hij of zij mocht vertrouwen dat die goed zou worden uitgevoerd. Het gevolg van een dergelijke hapering kan het gehele systeem als een dominospel in elkaar laten storten. We zijn vergeten de ooit zo gewaardeerde principes van 'loose coupling' toe te passen. In het woordenboek van SOA komt ontkoppeling niet meer voor. Vrijwillig laten we net als in het verkeer ons lamleggen door de pieken. Wie 's ochtends eerst thuis zijn mail doorneemt, rijdt bui-

ten de spits naar kantoor. Het is nergens voor nodig dat een gebruiker van een business-applicatie zich opwindt over de prestaties van het systeem.

SOAP is een gestandaardiseerde definitie voor het benaderen van webservices. Achter die façade zit een black box. Daar zouden we echter heel gemakkelijk een queue kunnen plaatsen. Het lijkt erop alsof de wereld real time nastreeft, doch 'near' real time blijkt bij de meeste bedrijfsapplicaties ruimschoots voldoende. Gebruikers zitten niet altijd te wachten op een directe bevestiging. Ook bij het toepassen van een queue vindt verwerking nog altijd vaak binnen luttele seconden plaats. Gaat er iets fout, dan escaleren we naar een hoger niveau zonder dat het slechte nieuws direct bij de gebruikers – of nog erger – buiten de organisatie terechtkomt. Alle focus in ons SOAP-tijdperk is evenwel komen te liggen op de distributie van services die ergens op het web leven. We vergeten te ontkoppelen en maken de hele wereld tot één monolithische klont.

André Rottier, Principal Consultant /Architect Capgemini
