

VELDWIJK

SOA? DOA!



Het moest er eens van komen: een column over Service Oriented Architecture(s). Ik hoor dat SOA passé is, dus het heeft haast. SOA gaat worden vervangen door een nieuwe afko – *cloud computing*, hoor ik nu. Welnu, ik vind SOA geweldig, zoals ik Internet geweldig vind. Ik vind het fantastisch dat we gebruikers op één html-pagina kunnen voorzien van gegevens over iemands maandinkomen, de WOZ-waarde van zijn huis, de AEX-ticker en een buitenradarfilmpje. Mijn probleem met SOA is wat ik met zoveel *technohypes* heb: nuttig mits niet gepositieerd als de oplossing voor problemen met een gegevensinfrastructuur. Wie denkt dat SOA gegevensintegratieproblemen oplost, werkt zoals altijd aan een ramp: minder functionaliteit, beroerde performance, meer tools en interfaces en vooral veel meer code. SOA is een waardige afstammeling in een lijn van verlossertechnologieën die begon met gedistribueerde databases en EDI en via *datawarehouses* en *scraping* nu uitkomt bij SOA en de '*computing cloud*'. Onlangs nog hoorde ik op een seminar Jan Baan nog beweren dat databases passé zijn en de 'wolk' alle processen binnen en tussen organisaties naadloos gaat integreren. "*Lamaar*", zo dacht ik.

Toch verdient de SOA van vandaag een trap na en tegelijk een schop vooruit. Ik zie namelijk met eigen ogen hoezeer de omgang met SOA het allerslechtste uit ICT-faciliteiten, budgetten en mensen haalt terwijl ik SOA als zodanig – anders dan voorgaande hypes – niet zie verdwijnen.

Allereerst valt op dat gebruikers bij SOA overgaan van rijke GUI's naar html-schermpjes, waarbij het summum van hi-tech het slepen aan de *scrollbar* is. Het is niet zo dat deze teruggang naar het stenen tijdperk inherent is aan SOA, maar het is wel de praktijk. Daarbij blijkt SOA in bouw en exploitatie altijd duur en vaak traag en lijkt er een hoge correlatie tussen SOA en projectfalen. SOA zonder een extreem gedisciplineerde werkwijze is DOA: *Dead On Arrival*. Hoe dat komt? Allereerst leidt het complex van SOA, event gedreven systemen en jonge SOA-programmeurs ertoe dat databases onnodig zwaar worden belast door grote aantallen atomaire opvragingen. Wederom komt dat niet dóór SOA maar mét SOA. Onze oplossing: webservice programmeurs laten werken op door database programmeurs gemaakte *compound views* en afnemers melden dat wie onnodige atomaire request/response programmatuur schrijft, getraceerd wordt op een wachttijd van tien seconden na elke request. Interne afnemers die grote hoeveelheden data afnemen hebben we verrast met database links en een uitleg van wat een database cursor is – heel *low-tech*. Zo hebben we waar nuttig een

SOA gemaakt die werkt: dagelijks leveren we tot 50.000 dikke persoonsdossiers uit via een webservice aan vele tienduizenden gebruikers in het land en het systeem merkt het nauwelijks. Nogmaals, in *onthypede* vorm is SOA in concept prachtig, maar zelfs dan is het geen pretje om uit te zoeken waar bijvoorbeeld een probleem met responstijden precies vandaan komt. Ook heel vervelend is dat de stand van de SOA-technologie zodanig is dat er veel informatie in programma's moet worden gestopt om SOA te laten werken – en dat aan twee kanten! Dat gaat vaker verkeerd dan goed. Meestal gaat het om kleinheden, maar als SOA wordt toegepast tussen organisaties of binnen organisaties met de ICT-leverancier in Mumbai betekent ook een kleine SOA *mismatch* zomaar een kwartaal uitloop. Dan is er een bezwaar dat volgens mij zelden wordt genoemd: SOA leidt in de praktijk vaak tot gesjacher en gesleep en soms tot misleiding en bedrog. De oorzaak is dat er bij SOA vaak twee partijen zijn die elk een nontriviale hoeveelheid software moeten ontwikkelen en elkaar niet direct in de ogen kijken. Complete specificaties zijn dan belangrijk maar onderlinge openheid nog meer. Een collega kwam recent een situatie tegen waarin bedrijf A bedrijf B onder druk zet om een afgesproken webservice te bouwen. Bedrijf B zet onder druk alles op alles om aan zijn verplichtingen te voldoen waarna bedrijf A zelf helemaal geen software blijkt te hebben gebouwd. Die belangrijke webservice is er na een jaar nog steeds niet. Zo wordt SOA dus DOA.

Maar het meest trieste van alles is dat de Internet infrastructuur die SOA mogelijk maakt SOA vaak ook overbodig maakt. Waarom bouwen we point-to-point webservices in weken of maanden waar we complete client-server functionaliteit over het Internet kunnen realiseren in uren of dagen? Wie het weet mag het zeggen. Ondertussen zijn mijn collega's en ik aan het nadenken over hoe we de belangrijkste problemen rond SOA kunnen oplossen in die gevallen waarin SOA nuttig is. Concreet denken we na over een maximaal declaratieve toepassing van SOA implementaties, geheel in de lijn van het artikel in de vorige DB/M over schermontwikkeling zonder programmeren. Want welk label we er ook op plakken, SOA is een blijvertje en we kunnen niet volstaan met de toepassing ervan te minimaliseren en gedisciplineerd met SOA te werken. Ik hoop binnenkort op dit onderwerp terug te komen, niet als columnist maar als ontwerper.

René Veldwijk (rene.veldwijk@faapartners.com) is partner bij FAA Partners, een onderdeel van de Ockham Groep.