



thema

De nieuwe wereld waarin de mens op veilige en betrouwbare wijze overall, op elk moment en met elk apparaat zich in verbinding kan stellen met allerhande elektronische diensten is boeiend. Maar de achterliggende technologie die dit mogelijk maakt is niet eenvoudig; vooral niet als die hoog beschikbaar, schaalbaar en eenvoudig aanpasbaar moet zijn. Als ik mijn technologiekeuze gemaakt heb, ben ik er nog niet. Want hoe zet ik die technologie het beste in?

# Microsoft patterns en practices

## *PAG bundelt ervaringskennis infrastructuur en applicaties*

Hoe bepaal ik de architectuur en infrastructuur van dat soort toepassingen, welke ontwerpkeuzes moet ik maken, hoe implementeer en beheer ik alles zo efficiënt mogelijk? Diverse leveranciers vullen hun technologieën aan met advies hoe die het beste ingezet kan worden. Microsoft doet dat ook en gaat daarin heel ver; en het is allemaal gratis te verkrijgen.

**PAG** Microsoft vult dit gebied in met wat zij noemen 'Prescriptive Architectural Guidance' of kortweg PAG. PAG is een organisatie-eenheid die zorgt voor de levering van 'Microsoft Patterns & Practices' op twee hoofdgebieden: de infrastructuur en de applicatie. Binnen elk van die hoofdgebieden worden drie niveaus van 'patterns & practices' onderscheiden: het architectuurniveau, het niveau van de bouwblokken en het operationele en beheerniveau. Inmiddels is een ware bibliotheek aan gidsen beschikbaar en er komen er elke maand nieuwe gidsen bij. Alles is online (zie [www.microsoft.NET/practices](http://www.microsoft.NET/practices); ook op cd-rom) verkrijgbaar; sommige content is ook in boekvorm verschenen via de Microsoft Press, die een apart 'practices' onderdeel kent.

Het begrip patterns komt oorspronkelijk uit de bouwkundige architectuur en is binnen de IT beroemd geworden door het boek Design Patterns van de 'Gang of Four'. Een pattern of patroon kan losjes gedefinieerd worden als 'een standaard oplossing voor een steeds terugkerend probleem'. Het begrip 'practices' moet gelezen worden als 'best practices': het zijn de beste ervaringen uit het veld of uit zware testomgevingen. De pat-

terns en practices zijn zeer inhoudelijk en diep technisch van aard. Een gedegen kennis van de producten van Microsoft en van algemene IT-architectuur beginnen zoals een meerlagenmodel wordt verondersteld. Het materiaal is echt bedoeld voor de praktijksituatie: "OK, we hebben Microsoft technologie in huis; hoe passen we die het beste toe?".

**INFRASTRUCTUUR** Bij infrastructuur moet gedacht worden aan al die zaken op en rond het Windows platform. Op architectuurniveau tref je hier allereerst de Microsoft Systems Architecture (MSA) aan. Samen met hardware partners zoals HP, Dell, EMC, Cisco en Nortel is een aantal hoog beschikbare en uiterst schaalbare technische infrastructuur omgevingen opgezet, ingericht en van voor tot achter getest, waarbij alle gebruikte materiaal is beschreven: netwerk materiaal, servers, opslagsystemen en software. De omgevingen zijn opgezet volgens een aantal patterns zoals een internet datacenter, een enterprise datacenter en een hosted datacenter.

Elk van die omgevingen stelt weer iets andere eisen aan de infrastructuur; zo bevat een enterprise datacenter wel directory services en file/print services, maar een internet datacenter niet. Van elke omgeving is een referentie architectuur of blauwdruk beschreven: een soort van toelichting op en verantwoording van de gemaakte architectuurkeuzes. Daarnaast is voor elke omgeving een of meer implementatiegidsen waarin installatie- en testscrijpts staan om een bepaald type datacenter met specifieke hardware constellaties stap-voor-stap in te richten en te testen. Tenslotte zijn er nog MSA-gidsen

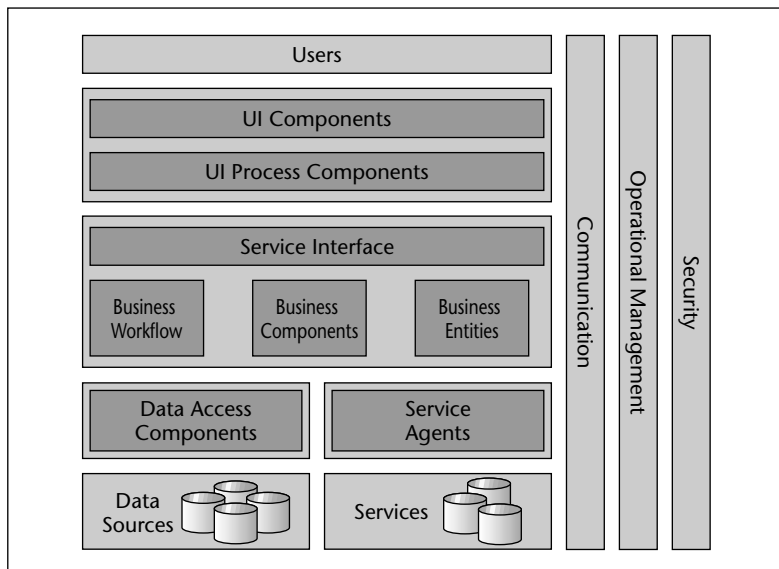
voor het beheer en de ondersteuning van de ingerichte omgeving.

Behalve voor complete datacenters is er ook een aantal gidsen ontwikkeld voor de architectuur rondom specifieke producten zoals Exchange 2000 en SQL Server 2000. Op operationeel-, beheerniveau kom je modules als 'Backup and Restore for Internet Data Center' tegen, waarin het ontwerp, de implementatie en het gebruik van back-up en restore faciliteiten stap-voor-stap wordt beschreven. Een ander voorbeeld op dit niveau is 'Security Operations for Microsoft Windows 2000 Server'. Op dit moment wordt door de PAG-groep hard gewerkt aan een update van een groot aantal van deze gidsen in verband met de komst van Windows Server 2003.

**APPLICATIE ARCHITECTUUR** De gemiddelde lezer van Software Release Magazine zal weliswaar erkennen dat een goede inrichting en operatie van de infrastructuur zeer belangrijk is, zijn of haar interesse gaat echter meer uit naar de applicatiekant. Figuur 1 geeft de applicatie architectuur voor .NET weer. Daarin is zowel aandacht voor (traditionele) meerlagen, componentgebaseerde applicaties als de meer recente service gerichte oplossingen. De 150 pagina tellende gids beschrijft op relatief hoog niveau de te nemen ontwerpbeslissingen en de diverse technologiekeuzes voor moderne service-gebaseerde toepassingen. PAG heeft reeds de exercitie uitgevoerd om de applicatie architectuur voor elke laag met producten in te vullen. Eind van dit jaar zal PAG met een referentie-applicatie op de proppen komen die de besproken applicatie architectuur volledig implementeert.

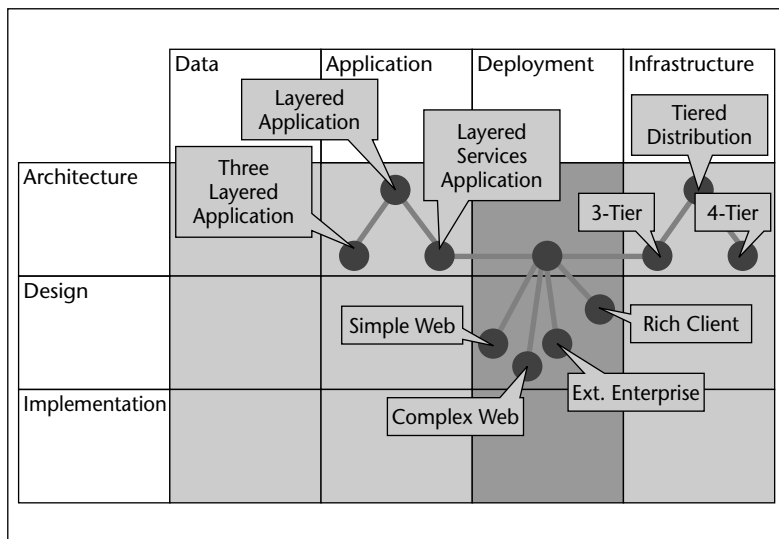
**BOUWBLOKKEN** Een niveau beneden de applicatie architectuur treffen we de bouwblokken aan die dieper – vaak tot op codeniveau – op specifieke onderdelen ingaan. Stel dat je op basis van de applicatie architectuur hebt besloten ASP.NET te gebruiken, dan is een volgende vraag wellicht: "Maar hoe maak ik mijn ASP.NET applicaties veilig?". Daarvoor heeft PAG een gids gemaakt die "Building Secure Microsoft ASP.NET Applications" heet. Dit 600 pagina's tellende boekwerk bespreekt hoe je diverse technologieën kunt combineren om veilige applicaties te maken. Het bespreekt de authenticatie, autorisatie en veilige communicatie (privacy, integriteit) van applicaties over de diverse lagen van die applicatie heen.

Het behandelt de veiligheidsaspecten in bepaalde scenario's en settings. Zo kan in een bepaald scenario het advies zijn voor veilige communicatie tussen de diverse lagen om SSL te gebruiken tussen de web browser en web server alsmede tussen web server en applicatie server en IPSec tussen applicatieserver en database server. Vervolgens wordt verteld hoe een en ander geconfigureerd kan worden.

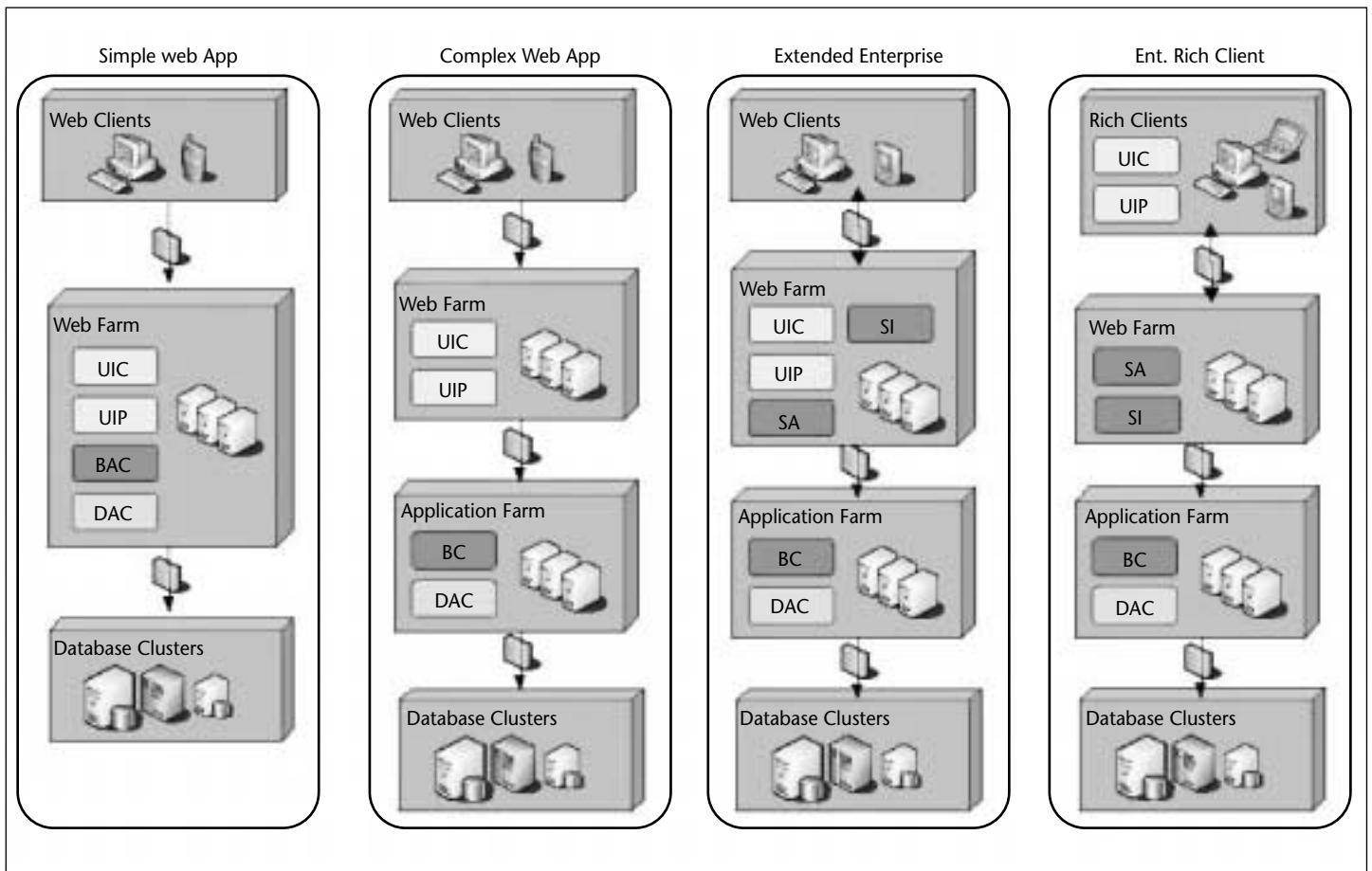


FIGUUR 1. De applicatie-architectuur van .NET

**MEEGELEVERDE CODE** Een ander voorbeeld is "Data Access Application Block for .NET" waarin de best practices rond ADO.NET vervat zijn en dat zich onder meer richt op het zo efficiënt mogelijk benaderen van de SQL Server database. De meegeleverde code kun je zo in je eigen applicatie plakken. Binnen de Nile-applicatie, een e-commerce performance benchmark, verving men de eigen reeds geoptimaliseerde data access laag door die van PAG. Toen bleek dit de hoeveelheid eigen code met een derde te reduceren, terwijl er geen performanceverlies optrad. Op het operationele niveau rondom de applicatie zijn er gidsen ontwikkeld zoals "Team Development with Visual Studio .NET and Visual Sourcesafe" en "Deploying .NET Applications: Lifecycle Guide". Hier gaat het dus meer om de best practices rondom de inrichting en het gebruik van ontwikkelomgevingen.



FIGUUR 2. De relaties tussen de verschillende niveaus en aandachtsgebieden, zal leiden tot clusters van patterns.



FIGUUR 3. Applicaties en infrastructuur van een web deployment omgeving

**PATTERN MODEL** Om de patterns op de diverse gebieden te ordenen is door PAG een pattern model ontwikkeld. Figuur 2 laat dat model zien. Men onderscheidt patterns naar twee gezichtspunten: enerzijds het niveau (architectuur, ontwerp en implementatie) en anderzijds het aandachtsgebied (database, applicatie, deployment en infrastructuur).

Er bestaan relaties tussen die verschillende niveaus en aandachtsgebieden, hetgeen zal leiden tot clusters van patterns. Figuur 2 geeft daarvan een voorbeeld. De hiervoor besproken applicatie architectuur voor .NET moet uiteindelijk landen op een infrastructuur. Daarbij wordt dan tevens een keuze voor het type deployment gemaakt en dat heeft gevolgen voor waar men welke onderdelen van de applicatie laat landen. Zo zal een gelaagde applicatie-architectuur voor een relatief simpele web deployment omgeving landen op een drielagen infrastructuur waarbij feitelijk alle onderkende applicatie architectuur componenten op de gecombineerde web- of applicatieserver terecht komen (figuur 3). In die simpele omgeving is geen plaats voor service interfaces en service agents en ook niet voor business workflows. Dat type componenten komt men wel tegen op de extended enterprise deployment omgeving, die dan ook uit een vierlagen infrastructuur bestaat.

Het is niet de bedoeling van Microsoft om alle patterns binnen het model zelf te ontwikkelen. De specificaties van de meeste patterns zullen wel van de hand van Microsoft zijn, maar voor de implementatie ervan verwacht men veel van de Microsoft partners en de .NET community ([www.gotdotnet.com](http://www.gotdotnet.com)).

**TOT SLOT** Binnen Microsoft zelf wordt veel belang gehecht aan het werk van de PAG-groep. De IT-wereld wordt immers almaar complexer. Het is in de eerste plaats in het belang van de klanten, maar daarnaast ook in het belang van Microsoft dat de door hen geleverde technologie op de best mogelijke wijze wordt ingezet. Dat leidt niet alleen tot kwalitatief betere resultaten, maar ook tot enorme kostenbesparingen omdat men kiest voor bewezen oplossingen.

Ook de system integrators kunnen van het werk van PAG profiteren. Ze hoeven het wiel niet opnieuw uit te vinden en kunnen hun eigen diensten aanvullend op de patterns en practices leveren. Voor de nabije toekomst kunnen we nog meer werk van de PAG-groep verwachten. Liefhebbers dienen de website in de gaten te houden.

*Dik Bijl, enterprise architect*