

.NET developers in de wolken

Bram Veenhof

Op de Professional Developer Conference in Los Angeles stond de keynote helemaal in het teken van cloud computing. Ray Ozie onthulde het Azure Services Platform. In dit artikel kijk ik naar de Software + Services-visie waar het Azure Services Platform onderdeel van uitmaakt. Vervolgens duid ik de verschillende componenten van het Azure Services Platform.

De PDC (27 - 30 oktober) was natuurlijk de plek waar Steven Sinofski Windows 7 voor het eerst presenteerde. Waar Microsoft voor de eerste keer openlijk praatte over het Oslo modelling framework. Het was de PDC waar de bezoekers een 160Gb HD kregen vol met technische content. Daar wil ik het hier nu niet over hebben. Wel over het Azure Services Platform, een development-platform op het internet dat het ontwikkelen van cloud-applicaties voor .NET developers zeer moet vereenvoudigen.

Het Azure Services Platform maakt onderdeel uit van de Software + Services (S+S)-visie van Microsoft. Deze visie omvat trends als Service Oriented Architecture, Web 2.0, Rich Internet Applications, Software as a Service en Cloud computing, die het landschap drastisch zullen veranderen. Microsoft verwacht niet dat alle software de browser als client gaan gebruiken. Dat zou zonde zijn van de kracht van de PC's en andere devices waar de browser op draait en doet ook geen recht aan het unieke karakter van die apparaten. Het gaat bij S+S om het creëren van ervaringen die de magie van software combineert met de kracht van internetdiensten over een wereld van devices.

Een belangrijk onderdeel van deze visie is dat software niet meer standaard gedistribueerd wordt met een perpetual license op een dvd. Maar dat we software ook op andere manieren kunnen verkrijgen. Daarbij kun je denken aan het huren van software, een abonnementvorm, prepaid of betaald door advertenties. Bovendien hoeft software niet altijd meer te draaien in de datacenters van de eindgebruiker. Partners leveren diensten uit hun datacenters in hosting-, housing- of outsourcing-constructies. Daarnaast ziet Microsoft dat leveranciers software als een dienst (SaaS) gaan aanbieden vanuit eigen datacenters. Tot 27 oktober was deze visie voor ontwikkelaars een theoretische exercitie. Microsoft had geen eigen datacenters waar ze het application-platform aanbiedt als een service. Met Business Productivity Online was Microsoft wel voorzichtig begonnen met wat diensten aan te bieden. Tijdens de keynote van Ray Ozie kwam deze visie een grote stap dichterbij met Azure Services Platform als hét platform.

Besturingssysteem in de wolken

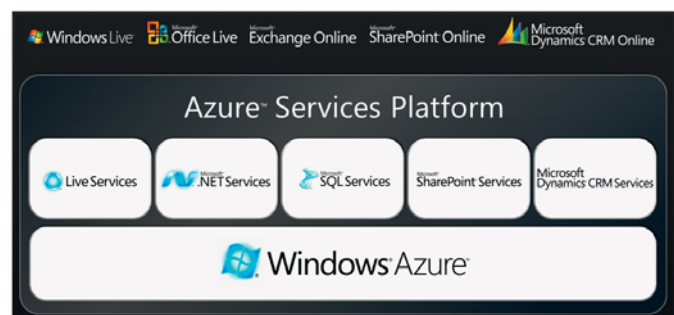
Stel je voor dat je op de volgende manier een applicatie gaat bouwen. Allereerst koop je de benodigde hardware. Je verzamelt alle

drivers om die hardware aan de praat te krijgen. Vervolgens schrijf je een filestelsel om bestanden op te slaan. Als je efficiënt bent, zoek je een filestelsel dat iemand al eerder schreef. De applicatie die je gaat schrijven, moet meer dingen tegelijk kunnen, dus zorg je voor een job scheduler. Verder gaat de applicatie memory gebruiken, dus schrijf je je eigen memory-manager.

Dat is niet de manier waarop we momenteel applicaties schrijven voor PC of mobile. De reden hiervoor vloeit voort uit het besturingssysteem. Een besturingssysteem maakt al deze zaken doorzichtig voor de applicatie, zodat je je als programmeur alleen nog hoeft te focussen op de businesslogica die de applicatie uniek en waardevol maakt.

Op dit moment lijkt het bouwen van een internetservice nog wel erg op onze beschrijving. Voordat de service live kan gaan, moet je eigenlijk de volgende onderdelen geregeld hebben. Allereerst een datacenter; gelukkig kun je tegenwoordig een rack of een deel ervan ergens huren. Vervolgens komt daar hardware in en die moet onderhouden worden. En je service dient er rekening mee te houden dat de hardware stuk gaat. Als je service in populariteit groeit en meer verkeer af te handelen krijgt, is het tijd hardware toe te voegen en load balancers in te zetten. Het besturingssysteem dat op de hardware staat, vergt onderhoud. Je moet patches installeren en af en toe upgraden. In de service moet je met een hoop zaken rekening houden. Als het niet om een service maar om een desktop-applicatie zou gaan, behoort dit tot de taken van het besturingssysteem.

Het Azure Services Platform fungeert als besturingssysteem voor internetservices. Het komt neer op een abstractielaag tussen de complexiteit van een datacenter en de code van de applicatie.



Het gaat bij S+S om creëren van ervaringen die de magie van software combineert met de kracht van internetdiensten over een wereld van devices

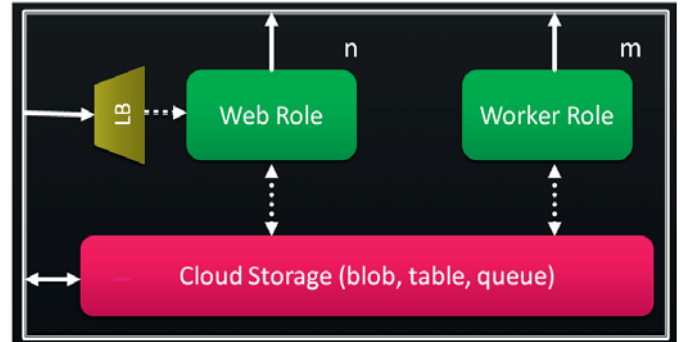
Ontwikkelaars moeten zoveel mogelijk tijd besteden aan het schrijven van de code die de service uniek en waardevol maakt en zo min mogelijk bezig zijn met omzeilen van complexiteit. Het platform levert een execution environment, een 'oneindig' elastisch filesysteem dat schaalbaar naar de behoefte van de service. Het is bijna net zo eenvoudig een service te starten als een executable op een desktop. Resources uit de Azure-datacenters kunnen we opschakelen en terugbrengen als de belasting van de applicatie dit vraagt. Schrijven en debuggen van services verloopt via Visual Studio.

'Windows' van het platform

Het Azure Services Platform bestaat uit verschillende lagen. Helemaal onderaan liggen de foundation-services. Windows Azure functioneert als de 'Windows' van het platform, het levert computation, storage en management van het platform. De bovenliggende building block-services gebruiken zelf weer Windows Azure. De building block-services leveren mogelijkheden die een developer gewend is te gebruiken in het Microsoft Application Platform. Daarnaast krijgen we een aantal nieuwe mogelijkheden. Sharepoint en CRM-services fungeren als de APIs van het business productivity online-platform. Via deze services integreert een klant de line of business applications met het information worker-platform (Sharepoint, CRM, Exchange) in Microsoft Datacenters. SQL Services levert relationele databasefunctionaliteit, daar waar Windows Azure meer de file-systeemachtige storage biedt. .Net services verschaft workflow, service bus en access control uit de cloud. Allemaal geen typische diensten waar een eindklant meteen mee aan de slag kan, maar zeer nuttig als building blocks voor nieuwe services. Daarboven liggen de finished services. Dit zijn juist wél de diensten die de eindgebruiker meteen kan gebruiken. Exchange, Sharepoint en CRM online en het Live Platform met diensten als Live hotmail en Live maps. Hierna kijken we vooral naar voor ontwikkelaars interessante foundation- en building block-services.

Windows Azure

Met het bieden van zeer rudimentaire diensten (compute, storage en management) fungeert Windows Azure als het Microsoft Datacenter-besturingssysteem. De computation-diensten mag je vergelijken met zeer gelimiteerde virtuele machines. Deze hebben geen desktop, geen filesysteem, geen Internet Explorer of notepad maar draaien alleen de code van de service waar ze aan zijn toegewezen. Er bestaan in principe twee type virtuele machines: een web node en een worker node. Een web node, de naam zegt het eigenlijk al, onderhoudt een http(s)-connectie met het internet. De web node kan momenteel WCF, ASP.NET, HTML en Silverlight aanbieden (op den duur komen ook native-code en daar



mee andere platformen als PHP beschikbaar). Daar waar de web node de ingang vormt van de service tot het internet, onderhoudt de worker node daarentegen geen connectie met het net en doet, zoals ook de naam al doet vermoeden, het werk. Beide type nodes komen in verschillende specificaties (RAM, CPU) beschikbaar. Beide hebben wel toegang tot de Windows Azure storage-service. Het gaat om een zeer rudimentaire opslag die je het best kunt vergelijken met een filesysteem. Er bestaan drie soorten opslag, namelijk blob (het opslaan van binary data), table (platte tabellen) en queue. Azure storage schaalbaar met de behoefte van de service zodat de service onbeperkt kan groeien.

De managementservices richten zich vooral op het beheer van de nodes. Op het moment dat een service veel verkeer te verwerken krijgt, wil je snel web nodes toevoegen. Met Windows Azure beschik je over een web portal en API's die deze processen vereenvoudigen. Het komt feitelijk neer op het verhogen van de hoeveelheid nodes. Windows Azure verzorgt de provisioning en loadbalancing.

Ontwikkelen voor het Windows Azure framework gaat zeer eenvoudig door de Cloud Services templates en development fabric





voor Visual Studio 2008. Na installeren van de Azure SDK ben je in staat services offline te bouwen en te debuggen, alvorens ze naar Windows Azure te deployen.

Hoe ziet een Windows Azure-service er uit? Als simpel voorbeeld noemen we een fotosite. Via de front-end loaden gebruikers foto's up. Deze krijgen een watermerk en er worden thumbnails van gemaakt, om deze vervolgens via de webwinkel van de site te verkopen. De web node vangt de foto's op, plaatst ze in de de blob storage en zet een job in de queue. Een worker node haalt de job uit de queue. Vervolgens past de worker node de foto aan en maakt een thumbnail. Dit model van een heel basic Windows Azure-applicatie geeft een aantal 'regels' aan. De worker nodes communiceren alleen via de queue met de web nodes. Elke node is stateless. Het model kunnen we uitbreiden tot zeer complexe services.

SQL Services

Zoals Windows Azure functioneert als besturingssysteem voor internetservices, doet SQL Services dit als database. Waar Windows Azure storage unstructured data opslaat, vormt SQL Services de plek voor structured data. SQL Services is gebaseerd op SQL Server. Je maakt in SQL Services bijvoorbeeld joins, wat in Windows Azure niet gaat. Belangrijk daarbij is momenteel dat SQL Services schaalbaar met de service (hoe hard die ook groeit) en via het sync-framework data van on-premises SQL servers repliceert naar de cloud en terug. Dit maakt business intelligence en reporting op internet data stores mogelijk.

.NET Services

De door .NET Services geleverde diensten halen een hoop complexiteit uit je code. De nu in de CTP aangeboden services bestaan uit een Service Bus, Workflow en Access control. Andere diensten volgen in de loop van de tijd. De service bus maakt het mogelijk op een veilige, flexibele en firewall-vriendelijke manier applicaties met elkaar te laten communiceren. Sommigen vertonen overeenkomsten met Biztalk-services. Het is complex om een service te bouwen die moet integreren met applicaties achter de firewall bij klanten. In de nieuwe situatie hoeft je alleen te communiceren met de service bus, die op deze complexiteit berekend is. Als de on-premises applicatie dit ook kan, regelt de service bus de security, messaging en eventing. Praten met de service bus biedt

als ander voordeel het zeer eenvoudig vervangen van de applicatie aan de ontvangende kant. Er zit namelijk geen service-specifieke code in.

Workflow-services zijn onmisbaar, zeker in een platform met 'loosely coupled'-componenten die asynchroon werken. .NET Workflow Services brengt .NET workflow naar de cloud zodat je service er op dezelfde schaalbare manier gebruik van kan maken als alle diensten van het Azure Services Framework.

Als services en applicaties niet meer in hetzelfde domein leven, vormen Identity en Access control (verificatie en autorisatie) het grootste integratievraagstuk. Voor de meeste eindklanten is het onacceptabel dat eindgebruikers een andere gebruikersnaam en wachtwoord moeten onthouden, als ze gebruik willen maken van een andere dienst. Microsoft zet zwaar in op claims en claims aware-applicaties. Door gebruik te maken van claims voor autorisatie wordt het mogelijk de verificatie te federeren zonder de applicatie daarvoor aan te passen. Het nieuw ontwikkelde Geneva framework (www.microsoft.com/geneva) maakt het de service-ontwikkelaar eenvoudig claims aware-services te schrijven. Als de service

Het Azure Services Platform is een besturingssysteem voor internetservices

zich daaraan confirmeert, is het wisselen tussen Windows Integrated Authentication naar Live ID en naar gefedereerde verificatie een kwestie van configureren. De .NET Access-service maakt het vervolgens mogelijk de service in het Azure Services Platform te laten verifiëren door een identity provider die bij de klant achter de firewall staat (bijvoorbeeld Active Directory). Het federeren van Active Directory aan de .Net Access Services is door de Microsoft Services-connector een kwestie van een aantal kliks.

Live Services

Alle tot nu toe beschreven services vinden een logisch equivalent in het traditionele Microsoft Application Platform. Dit geldt niet voor



Live Services. Toch vormt Live Services een essentieel onderdeel van de Software + Services-visie: "Het creëren van ervaringen die de magie van software combineert met de kracht van internetdiensten over een wereld van devices". Alle services die we tot nu toe zagen hadden te maken met de magie van software en gebruikte de kracht van het internet. Maar waar blijven de devices? Live Services brengt Software + Services naar de device. Allereerst is Live Services de software die de verschillende devices met

In dit artikel heb ik Windows Azure toegelicht. In toekomstige edities van .NET Magazine mag je meer informatie verwachten over Windows Azure.

Blog: <http://blogs.microsoft.nl/bramveen>



Live Services brengt Software + Services naar de device

elkaar verbindt in een 'Mesh'. Een Mesh bestaat uit een verzameling devices die bij elkaar horen. Bijvoorbeeld mijn laptop, mijn netbook, mijn telefoon, mijn live desktop en mijn mediacenter-pc vormen mijn Mesh. Deze devices horen bij elkaar omdat ik ze gebruik. Het aan elkaar koppelen van devices in een Mesh en een framework om meer devices (denk aan printers, digitale fotolijstjes, digitale camera's) aan de Mesh toe te voegen, maken onderdeel uit van Live Services.

Als de devices in een Mesh zitten is de kans heel groot dat ik gegevens op het ene device op het andere wil gebruiken. Live Services maakt het mogelijk en zeer eenvoudig om bestanden (bijvoorbeeld foto's) van het ene device naar het andere te repliceren. Doordat de data is gerepliceerd, kan ik die gegevens ook gebruiken als het device niet verbonden is met het internet. Dus een data-replicatie-framework maakt ook onderdeel uit van Live Services.

Als we kijken naar het voorbeeld van mijn Mesh bestaat het verbindende element van de verschillende devices uit 'mijn' devices. Live Services heeft dus ook weet van een user en die heeft contacts waarmee hij data en devices wil delen. Tot zover lijkt het nog steeds meer op een finished-service dan op een building block-service, in principe een datasynchronisatie-framework. Het Live Services Framework is echter ook een applicatieplatform. Het maakt het namelijk mogelijk applicaties te schrijven die dan op elk device in de Mesh functioneert. Een applicatie die zowel in de browser als op de desktop werkt. Een applicatie die online en offline beschikt over de applicatiedata en deze weer synchroniseert met de Mesh zodra het device online komt.

Naast het gebruikmaken van het Live Services Framework is het ook mogelijk de data vanuit andere web- of client-applicaties te benaderen. Persoonlijk vind ik Live Services het meest fantastische onderdeel van het Azure Services Platform omdat het scenario's mogelijk maakt die hiervoor niet denkbaar waren of praktisch niet uitvoerbaar.

Ten slotte

Alle componenten van het Azure Services Framework zijn gebouwd om optimaal te laten werken met open standaarden. Alle services hebben SOAP- en/of REST-interfaces zodat het platform eveneens ter beschikking staat aan niet-.NET-programmeurs.

Bram Veenhof (bram.veenhof@microsoft.com) is Web Platform Architect bij DPE Microsoft Nederland.

(Advertentie)
