



# Communicatie

Bijna zes jaar geleden, toen Microsoft .NET introduceerde, was de meest in het oog springende, alom gedemonstreerde oplossing voor alle problemen de webservice. Het klonk ook erg mooi: platform onafhankelijk met standaard protocollen over het internet alle mogelijk denkbare services aanspreken. Te mooi om waar te zijn. Nadat de initiële hype wat bedaard was en er nagedacht werd over het serieus (lees: in productie omgeving) toepassen van deze technologieën, bleken er redelijk wat haken en ogen aan het hele verhaal te zitten.

Het bouwen en aanspreken van een Web Service die een enkele string 'Hello World' teruggeeft is binnen een minuut gedaan met behulp van Visual Studio. Echt spannend wordt het als er wat extra eisen worden gesteld: de tekst moet versleuteld worden, de tekst moet van een digitale handtekening voorzien worden of niet iedereen mag de tekst opvragen. In een voorbeeld op het internet, in een boek of presentatie worden deze dingen eenvoudig overgeslagen maar in de praktijk zijn ze de gewoonste zaak van de wereld.

In samenwerking met andere partijen heeft Microsoft een verzameling van extra specificaties opgesteld zodat deze extra eisen, binnen de gekozen standaarden, op een platform onafhankelijke manier gerealiseerd kunnen worden. Deze specificaties zijn bekend onder de naam WS-\* Het heeft natuurlijk geen zin om de tekst die verstuurd wordt te versleutelen met behulp van een platform afhankelijk algoritme. Ondanks het gebruik van HTTP en SOAP zal het bericht dan niet door iedereen (die daartoe gerechtigd is) gelezen kunnen worden.

Microsoft heeft vanaf het beschikbaar komen van deze specificaties er

zorg voor gedragen dat het relatief eenvoudig is om Web Services, die gebouwd worden met Visual Studio, te laten conformeren aan deze afspraken. De toolkit met de naam Web Service Enhancements (WSE) bestaat uit een uitbreiding voor Visual Studio en uit een run-time omgeving die de berichten, voordat ze verstuurd of ontvangen worden, onderschept en voorziet van extra informatie.

Ondertussen is er bij Microsoft het besef ontstaan dat het probleem dat met Web Services opgelost wordt binnen de Microsoft technologieën ook op andere manieren opgelost kan worden. Remoting, Message Queueing en Serviced Components (COM+) bieden vergelijkbare oplossingen aan die door veel ontwikkelaars worden toegepast. Wanneer we kijken naar de code die geschreven wordt in de service en in de client dan valt op dat daar vaak veel geprogrammeerd moet worden om de functies van de gekozen technologie aan te spreken. Zo zal, wanneer een Service gebruik maakt van Message Queueing, de client code bevatten die een bericht stuurt naar een Message Queue. Dat lijkt vanzelfsprekend maar het is eigenlijk niet logisch dat de client weet op

welke manier de Service een technisch probleem oplost. Het enige dat de client hoeft te weten is waar de Service zich bevindt, welke functionaliteit daar is en welke taal de Service spreekt.

Microsoft heeft er voor gekozen de technologieën te verstoppen in configuratie en een run-time. Met als gevolg dat in de code van de client en de Service geen technologie specifieke code geschreven hoeft te worden en dat het makkelijker is geworden om via het wijzigen van de configuratie een andere technologie te gebruiken. Deze run-time heet Windows Communication Foundation (WCF) en deze zal tegelijkertijd met Windows Vista beschikbaar zijn. Heel prettig is om te weten dat WCF volledig compatibel is met WSE en dat het migreren of integreren van de 'oude' en nieuwe communicatielaag daardoor relatief eenvoudig zal zijn.

Misschien dat we nu eens kunnen gaan nadenken over welke functies we gaan aanbieden...

*Erno de Weerd – Info Support. Over deze column kan verder worden gediscussieerd op <http://blogs.infosupport.com/ernow>.*