

**PCM**

HET PERSONAL COMPUTER MAGAZINE

Grootbeeld voor iedereen  
**MASSATEST MONITOREN**  
 10 budget 17 inch schermen

**E-ECONOMY** "straks is er helemaal niets meer gratis..."

**PCM TEST****6 Gigahertz pc's****VERGELIJK**

Pentium 4:  
 eindelijk écht  
 multi-tasking

**INTERNET**  
 Zoeken en  
 gevonden  
 worden

**SONY**

BASS  
 Memory Stick  
 006-024.13  
 ALL REMAINING  
 PORTABLE MEMORY STICK PLAYER

**MS  
WALKMAN**

- 1 De nieuwste spelers
- 2 De platenbonzen
- 3 Maar ook: Napster

ISSN 0772-8077 • ACHTTIENDE JAARGANG



8 710294 191989

www.pcmweb.nl

EEN NIEUW BEGIN MET EEN OPEN EINDE

# Microsoft .NET

Microsoft gaat in de aanval. De geplaagde softwaregigant heeft goed naar het troetelkintje van rivaal Sun gekeken. De overeenkomsten tussen Microsoft.NET en het bestaande Java 2 zijn legio

Met .NET kondigde Microsoft afgelopen zomer - bij monde van 'chief software architect' Bill Gates - een nieuwe generatie software aan, uit te spreken als 'dot net'. Die moet een nieuwe 'gebruikersbeleving' teweeg gaan brengen. Bovendien zal .NET standaard een reeks geavanceerde softwarediensten bevatten, en het zal het op internet gebaseerd gebruik van computers en communicatie gemakkelijker, persoonlijker en productiever maken voor zowel bedrijven als voor consumenten...

Indien u zich bij dergelijke abstracte beloftes niets kunt voorstellen, geen nood. Op korte termijn zult u als eindgebruiker namelijk nog niet veel merken van Microsoft .NET. Deze nieuwe generatie web-georiënteerde software zal zeer geleidelijk zijn weg naar de consument vinden. Het verschil tussen het gevestigde Java en nieuwkomer .NET zit 'm vooral in de te volgen strategie. Microsoft lijkt bekeerd tot open standaarden en leent of deelt de onderliggende protocollen van en met anderen (zoals IBM en Lotus). Dat geldt *niet* voor Java 2. Dit platform is stevig in handen van een enkel bedrijf en het is onwaarschijnlijk dat aartsrivaal Sun deze controle ooit volledig uit handen zal geven.

## VAN DOZEN NAAR DIENSTEN

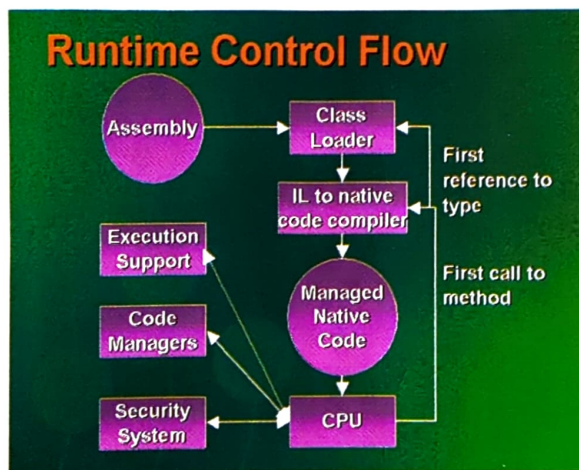
Voor eindgebruikers zal de beloofde gebruikersbeleving dankzij .NET vooral gestalte krijgen in de vorm van de zogeheten Universal Canvas. Die voorziening biedt een eenduidige omgeving voor het werken met informatie. In plaats van het zoeken naar en het lezen van informatie op het web in uw internet-bladeraar, en het aanmaken en bewerken van informatie in uw office-programma's, zult u al uw werkzaam-

heden in het nieuwe Universal Canvas kunnen uitvoeren. Verder krijgt .NET een Natural Interface via een nieuw type inbox dat onder meer spraak, handschrift en natuurlijke taal als invoer zal accepteren, en softwarediensten zoals een Information Agent om uw identiteit en andere persoonlijke gegevens te beheren. Sommige (software)-diensten zullen gratis zijn, terwijl voor anderen betaald dient te worden. Het is

## NIEUW PLATFORM

Microsoft .NET is niets meer en minder dan een nieuw platform voor het ontwikkelen en uitvoeren van softwaretoepassingen. Het biedt ontwikkelaars ondersteuning voor op internet gebaseerde systemen, een eenduidig en mede daardoor vereenvoudigd programmeringsmodel, automatisch beheer van geheugen en systeembronnen, een uniform foutafhandelingsmechanisme en geïntegreerde beveiliging. Bovendien is .NET niet gebonden aan één programmeertaal of ontwikkelomgeving en biedt het uitstekende ondersteuning voor het opsporen van fouten in de broncode. Het is zelfs mogelijk om een klasse geschreven in taal A te overerven in taal B. Tenslotte wordt ook het distribueren, installeren en configureren van .NET-toepassingen aanmerkelijk eenvoudiger. U kunt typische Windows-zaken als het register en zelfs COM terzijde schuiven, aangezien alle applicatiediensten aangeboden worden via de veelomvattende .NET-klassenbibliotheek en de onderliggende .NET Runtime.

Overigens is .NET niet een nieuwe naam voor, of een nieuwe versie van Windows DNA. Het is wél de uiteindelijke benaming voor de Next Generation Windows Services (NGWS). Nog dit jaar zal de officiële bèta van Microsoft Visual Studio.NET het licht zien. Tot dan zult u het moeten doen met de .NET Framework SDK Technology Preview, die gratis van de bekende website ([www.microsoft.com](http://www.microsoft.com)) is op te halen. Deze 'prebèta' bevat onder meer de benodigde command-line compilers. Over de verschijningsdatum van .NET 1.0 doet Microsoft geen uitspraak, maar de verwachting is dat dit nieuwe platform pas in de tweede helft van 2001 zal worden uitgebracht. Geen haastige spoed dus.



Een overzicht van het uitvoeringsproces van een .NET-assembly, zoals gepresenteerd tijdens de .NET-aankondiging.

deze verschuiving - van aan te schaffen applicaties in een kartonnen doos naar softwarediensten waarop u zich via internet kunt abonneren - waarop nu ook Microsoft afstevent. Het .NET-platform zal een coherente infrastructuur verschaffen voor dit toekomstmodel.

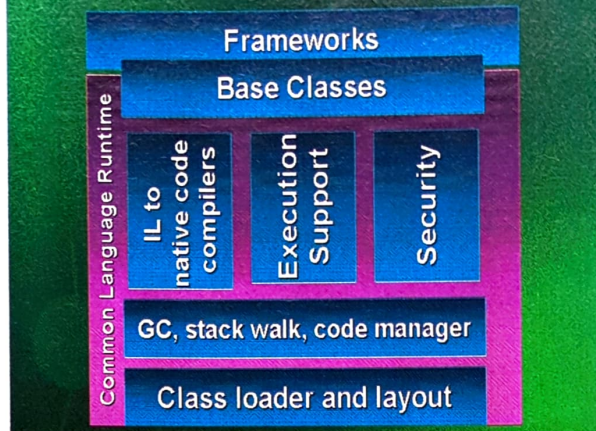
Maar een digitale revolutie of niet, .NET zal eerst het hart van de schare Windows-ontwikkelaars moeten overerven om door te kunnen breken. Een nieuw begin achter de schermen dus, waar de consument op de lange duur wel degelijk de vruchten van kan gaan plukken.

## IN ALLE OPENHEID

In Microsoft .NET worden gegevens in XML verpakt en door middel van SOAP-boodschappen via http over netwerken uitgewisseld. Extensible Markup Language (XML) is een deelverzameling van Standard Generalized Markup Language (SGML) dat geoptimaliseerd is voor gebruik op internet. Het wordt gedefinieerd door het World Wide Web Consortium (W3C) zodat gegevens uniform en onafhankelijk van bedrijven zijn. Dit is één van de redenen waarom XML als formaat voor het representeren van allerlei gegevens steeds meer navolging vindt.

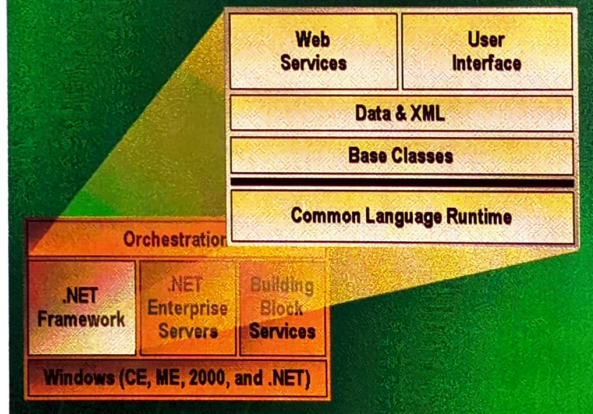
Het Simple Object Access Protocol (SOAP) is een op XML gebaseerde specificatie voor het uitvoeren van Remote Procedure Calls (RPC's), en dus de componentenlijm voor het ontwikkelen van gedistribueerde systemen. SOAP is door Microsoft in samenwerking met onder meer IBM en Lotus ontwikkeld en vervolgens overgedragen aan de Internet Engineering Task Force (IETF). In de huidige SOAP-specificatie wordt alleen over de toepassing van SOAP-boodschappen in de context van het Hypertext Transfer Protocol (http) als transportmechanisme gesproken. Http is niet alleen een alom gerespecteerde industriestandaard, maar bovendien één van de weinige protocollen die firewalls toelaten. Overigens blijft het mogelijk voor systeembeheerders om in een firewall SOAP-boodschappen te onderscheiden van de overige boodschappen die over de http-poort binnenkomen. Het grote verschil met vergelijkbare technologieën als COM Internet Services (CIS), Remote Data Services (RDS) en Remote Scripting is dat SOAP platform- en producentonafhankelijk is. SOAP is dus geen nieuw concept, maar biedt dezelfde functionaliteit in een relatief eenvoudig, generiek en op open standaarden gebaseerd protocol.

## Common Language Runtime



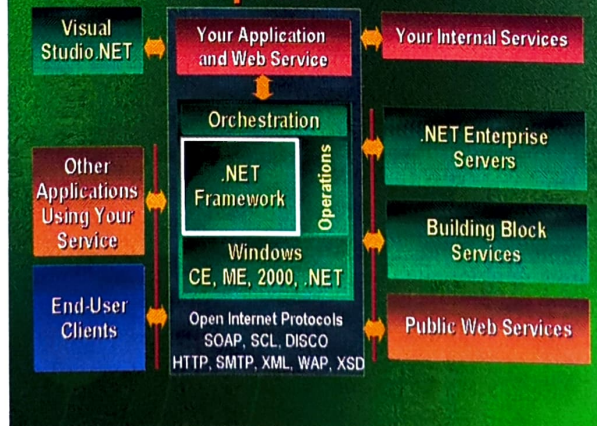
Een overzicht van de onderdelen waaruit de .NET Runtime is opgebouwd (aldus de .NET-aankondiging).

## .NET Framework



De .NET architectuur in context, zoals door Microsoft gepresenteerd tijdens de aankondiging.

## .NET Blueprint



## EEN OPEN EINDE

Veel is nog onduidelijk over .NET, zeker is wel dat Microsoft met .NET alles of niets speelt. Allereerst zal het nog minimaal een half jaar duren voordat de eerste officiële versie uitkomt, en zelfs dat is koffiedik kijken. Van cruciaal belang zal de ondersteuning van het enorme leger Windows-ontwikkelaars voor het .NET-platform zijn. Ontwikkelaars van Visual Basic hebben gewoonweg geen keus,

De overeenkomsten tussen Microsoft .NET en het bestaande Java 2-platform zijn natuurlijk overduidelijk (zie kader). De CLR versus de Java Virtual Machine (JVM), de verwantschap tussen de Java- en .NET-klassenbibliotheken, de overeenkomsten tussen de programmeertaal Java en het nieuwe C#, automatisch beheer van geheugen en systeembronnen: Microsoft heeft goed naar het troetelkinder van rivaal Sun gekeken. Maar Java is een niet-gestandaardiseerde programmeertaal. Microsoft daarentegen heeft in ieder geval reeds de intentie uitgesproken dat het C# zal standaardiseren via ECMA Technical Committee (TC) 39, en het onderliggende SOAP-protocol is zelfs nooit in bezit geweest van Microsoft. U hoeft maar te denken aan de Java-soap van rechtszaken tussen deze twee zwaargewichten om te bedenken hoe het kan uitpakken indien technologie niet onder controle van een onafhankelijk comité staat...

Sun's Java 2 heeft natuurlijk wel een flink aantal streepjes voor op Microsoft .NET. Zo bestaat het reeds jaren en is het intussen duidelijk waarvoor het wel en niet geschikt is. Bovendien wordt Java door vrijwel alle grote namen in de software-industrie ondersteund. Tenslotte is de JVM voor vrijwel ieder besturingssysteem beschikbaar en het is langzamerhand het stadium van versie 1.0 - met de vele kinderziekten - ontgroeid.

aangezien met Visual Basic 7 (VB.NET) alleen nog .NET-toepassingen zijn te produceren. Dat zijn alvast zo'n vier miljoen potentiële ontwikkelaars. Ook zal .NET web-programmeurs niet al te veel hoofdbreken veroorzaken. Voor ontwikkelaars van Visual C++ die onder meer toepassingen voor eindgebruikers produceren, is echter nog veel onduidelijk. C++ met beheerde extensies zal in de eerste versie bijvoorbeeld niet een door de Runtime te controleren IL-code genereren. De kracht van C++ laat zich moeilijk vertalen. Een eventuele overstap naar C# betekent voornamelijk het opgeven van platformafhankelijkheid op het niveau van de broncode zolang de .NET Runtime niet beschikbaar is voor niet-Windows-systemen.

Tenslotte zal ook de uiteindelijke uitvoeringssnelheid van .NET-toepassingen van doorslaggevend belang zijn voor het slagen ervan. Een tekortkoming op dit gebied neemt namelijk de grootste bron van ergenis over Java 2 weg. En Java is wél beschikbaar voor meerdere besturingssystemen. Overigens heeft Microsoft reeds statistieken aangehaald van interne testen op uitvoeringssnelheid die bijna ongeloofwaardig rooskleurig klinken... Aanmerkelijk sneller dan Visual Basic 6, slechts tien tot veertig procent langzamer dan C++? Het wachten is op de eerste officiële uitgave van de .NET Runtime. Nog dit jaar zal naar alle waarschijnlijkheid de officiële bèta van Microsoft Visual Studio.NET het licht zien, en zult u in PCM meer kunnen lezen over .NET in de praktijk. Hoe dan ook, met Microsoft in de aanval valt er het komende jaar weer veel te beleven. **PC**

Marco René Spruit

## .NET IN VOGELVLUCHT

### TAAL-NEUTRAAL

Aan de basis van .NET bevinden zich de Common Language Runtime (CLR, of Runtime) en een standaard klassenbibliotheek. De Runtime is taal-neutraal, dus u kunt uw favoriete programmeertaal kiezen om een bepaalde taak te implementeren voor het .NET-platform. Microsoft zal in de volgende versie van de Visual Studio suite van programmeeromgevingen, Visual Studio.NET, vier compilers leveren waarmee .NET-componenten kunnen worden ontwikkeld, te weten C++ met beheerde extensies (MC++), Visual Basic, JScript, en de speciaal voor het .NET-platform ontwikkelde programmeertaal C-Sharp (C#). Bovendien werken andere bedrijven aan .NET-compatibele compilers voor onder meer Pascal, Perl, SmallTalk en Cobol. De Common Language Specification (CLS) beschrijft de minimale eisen die een compiler moet ondersteunen om Runtime-compatibele code te kunnen genereren. Aan de andere kant van het spectrum bevindt zich Common Type System (CTS), dat het gehele bereik van types specificeert die de Runtime begrijpt. Hoewel .NET-toepassingen alle types kunnen gebruiken, ondersteunen niet alle .NET-programmeertalen ze.

### ZELFBESCHRIJVEND

De broncode van .NET wordt gecompileerd tot Intermediate Language (MSIL of IL), dus niet in direct uitvoerbare machinecode. Bovendien voegt een .NET-compiler metadata toe aan ieder IL-bestand dat de Runtime gebruikt om de ingevoegde klassen te kunnen lokaliseren, laden en activeren. Een verzameling van .NET-componenten wordt een assembly genoemd, die u zowel kunt vergelijken met een uitvoerbaar programma als met een bibliotheekmodule, aangezien er geen fysieke verschillen zijn tussen deze twee implementatievormen in het .NET-componentenconcept. De verzameling componenten wordt beschreven in een manifest. Deze bevat onder meer informatie over de bestanden waaruit de assembly bestaat en beschrijft eventuele afhankelijkheden. Een assembly is dus zelfbeschrijvend en bepaalt tevens het bereik van de naamsovername van de opgenomen types. Die zelfbeschrijvendheid maakt het voor zowel ontwikkelaars als systeembeheerders mogelijk om het gebruik van een bepaalde versie van een component af te dwingen, zodat installatieproblemen gerelateerd aan de beruchte 'DLL Hell' tot het verleden gaan behoren. Tel hierbij het ontbreken van een centraal registratie-

systeem zoals het Windows-register binnen het .NET-platform, en u begrijpt dat het installeren van een toepassing net zo eenvoudig is als het kopiëren van een map.

### KLASSE

Ontwikkelaars gebruiken de standaard klassenbibliotheek om systeemdiensten van de Runtime aan te spreken. De klassen zijn ontworpen voor het ontwikkelen van een breed scala aan toepassingsfuncties, zoals webdiensten, desktop-programma's en componenten. Alle functionaliteit is alleen beschikbaar via klassen die van Object zijn afgeleid. Gezien de diverse functionaliteit die de bibliotheek dient te bieden, is het niet verwonderlijk dat deze is opgebouwd uit honderden klassen, gegroepeerd in een aantal namespaces. De belangrijkste is de System namespace, waarin onder meer de Object-basisklasse is opgenomen, waarvan alle andere klassen zijn afgeleid. Door uw eigen namespace te definiëren, kunt u uw eigen klassen op transparante wijze integreren in het .NET-platform. Het Virtual Object System (VOS) specificeert hoe een .NET Object-klasse, ofwel een type, gedefinieerd moet zijn om door de Runtime gebruikt te kunnen worden.

### UITVOERING

Een .NET-toepassing kan in principe op ieder besturingssysteem worden uitgevoerd, aangezien het IL-bestandsformaat processor-onafhankelijk is. (Mits natuurlijk .NET Runtime op het OS aanwezig is.) Microsoft belooft de Runtime in ieder geval beschikbaar te maken voor de Windows 9x-familie (ook ME), NT 4 en hoger, en CE. Over versies voor Mac en Unix/Linux geen woord. Het starten van een .NET-programma heeft tot gevolg dat de IL naar processorinstructies wordt vertaald door de Just-In-Time (JIT) compiler van de Runtime. Na het JIT-compileren van het aangeroepen fragment wordt het beginadres ervan in het geheugen vervangen door dat van de gecompileerde code, zodat bij elke volgende aanroep van het fragment automatisch de snelle processorinstructies uitgevoerd worden. De snelheid van de JIT-compiler en vooral de snelheid van de geproduceerde processorinstructies zou wel eens van grote invloed blijken te zijn op het welslagen van het .NET-platform. Daarom biedt Microsoft een PreJIT-programma waarmee een volledige assembly gecompileerd kan worden. Hiermee verliest u de processoronafhankelijkheid, maar uw .NET-programma zal bij gebruikers een beduidend snellere eerste indruk achterlaten. Bovendien maakt het toepassing in de vorm van processorinstructies veel moeilijker te decompileren, een belangrijke factor in verband met intellectueel eigendomsrecht.

## LINKS

Startpagina: [www.microsoft.com/net](http://www.microsoft.com/net)

Voor ontwikkelaars: <http://msdn.microsoft.com/net>

Algemene FAQ: [www.andymcm.com/dotnetfaq](http://www.andymcm.com/dotnetfaq)

Web-geïntereerde FAQ: [www.4guysfromrolla.com/webtech/071900-1.shtml](http://www.4guysfromrolla.com/webtech/071900-1.shtml)

Overige .NET-informatie: [www.devx.com/dotnet/resources](http://www.devx.com/dotnet/resources)

Visual Studio.NET-startpagina: <http://msdn.microsoft.com/vstudio/nextgen/default.asp>

Visual Studio.NET FAQ: [www.devx.com/free/press/2000/vs-qalist.asp](http://www.devx.com/free/press/2000/vs-qalist.asp)

SOAP: <http://msdn.microsoft.com/xml/generals/soap-spec.asp>

XML: [www.w3.org/tr/rec-xml](http://www.w3.org/tr/rec-xml)