

Applicatie ontwikkeling

Service Oriented Architecture



Version : 1.0
Plaats : Voorburg, Zuid-Holland

Inhoudsopgave

1	Delta-N applicatieontwikkeling.....	3
1.1	Service-Oriented Architecture	3
1.2	Beschrijving .NET.....	3
1.3	Webservices.....	4
1.4	Keuze voor Service-Oriented Architecture.....	4
1.5	Wat is SOA?.....	5
2	In de praktijk.....	6
2.1	Projectbeschrijving.....	6
2.2	Keuze voor SOA.....	6
2.3	Toekomstvisie	7
2.4	Veiligheid	7
2.5	Snelheid	8
2.6	De producten	9
3	Samenvatting	10
3.1	Interesse	10
3.2	Links en referenties.....	10

1 Delta-N applicatieontwikkeling

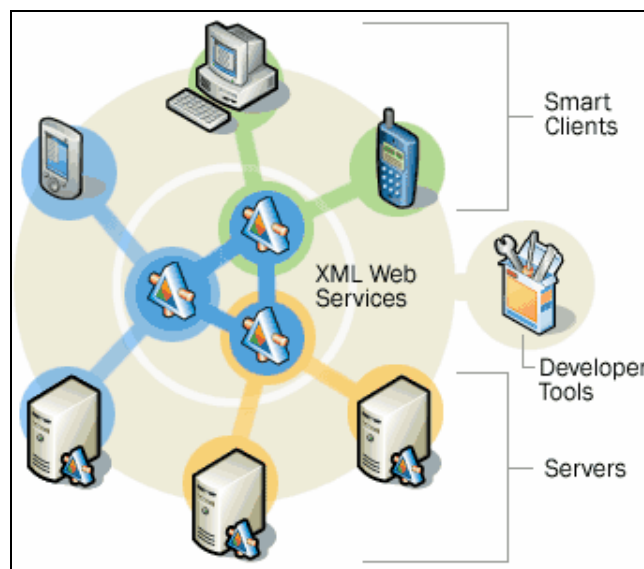
1.1 Service-Oriented Architecture

Service-Oriented Architecture (SOA) is de nieuwste methode voor de ontwikkeling van .NET-applicaties. SOA zorgt voor homogeen opgebouwde applicaties en IT-omgevingen. Dit maakt hergebruik van applicaties en code eenvoudig, zodat applicatieontwikkeling sneller, goedkoper en beter wordt en dus minder risico met zich meebrengt. Delta-N heeft SOA gekozen als dé ontwikkelmethode voor haar projecten.

In deze whitepaper wordt uiteengezet waarom Delta-N SOA gebruikt, waarom SOA in het algemeen de beste ontwikkelmethode is en, wellicht nog belangrijker, waarom dit grote voordelen biedt voor de klanten van Delta-N.

1.2 Beschrijving .NET

.NET is een set van tools en technieken ontwikkeld door Microsoft voor het bij elkaar brengen van informatie, mensen, systemen en apparatuur. Door het gebruik van webservices wordt een hoge mate van software-integratie mogelijk. Webservices zijn kleine onafhankelijke bouwstenen die via het internet aan elkaar of aan andere applicaties gekoppeld kunnen worden.



Figuur 1; .NET overzicht

De onderdelen van .NET zijn te zien in Figuur 1: 'smart clients', 'webservices', 'servers' en 'developer tools'. Via de smart clients, die bestaan uit zowel hard- als software, kunnen gebruikers toegang krijgen tot de webservices. De webservices vormen de brug tussen deze clients en de servers. De servers geven via webservices toegang tot de functionaliteit van de achterliggende applicaties. De developertools, of ontwikkeltools, kunnen worden gebruikt voor de ontwikkeling van de verschillende .NET-onderdelen, zoals webservices en smart clients. Samenvattend kan dus worden gezegd dat .NET applicaties worden gevormd door een combinatie van webservices, servers en smart clients.

1.3 Webservices

Een webservice is het communicatiemiddel tussen clients en servers. Deze webdiensten zijn gebaseerd op standaard XML (Extended Markup Language). Een groot voordeel hiervan is dat het een standaard, platformonafhankelijke en uitwisselbare communicatiemethode is. Hierdoor kunnen webservices snel en eenvoudig geheel of gedeeltelijk worden hergebruikt in een nieuwe applicatie. Deze hergebruikte onderdelen zijn bovendien van zeer goede kwaliteit, omdat ze al in een vorige applicatie zijn getest. Het schrijven van nieuwe code of het aanpassen van oude code, met alle risico's van dien, is niet meer nodig.

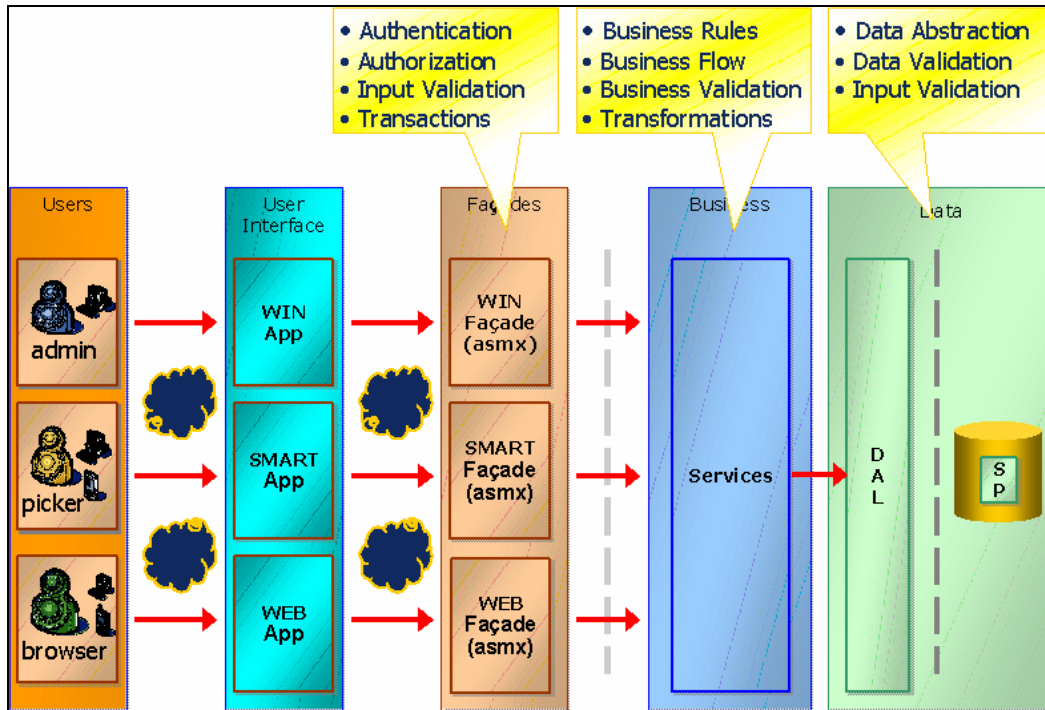
Webservices zijn zeer geschikt voor netwerk- en internetgebruik. Er hoeft geen ingewikkelde netwerkcode geschreven te worden, aangezien deze standaard in het .NET framework aanwezig is. Dit voorkomt problemen en bugs. Bovendien zorgt het ervoor dat er veel minder tijd nodig is voor de ontwikkeling van een applicatie dan met traditionele methodes. Webservices zijn te gebruiken via standaard protocollen, zoals HTTP(S) en TCP/IP, en zijn gebaseerd op SOAP (Simple Object Access Protocol), een gestandaardiseerd XML-protocol. Hierdoor is het mogelijk om op transparante wijze verschillende systemen aan elkaar te koppelen. Zo kunnen Java-applicaties die draaien op een UNIX-machine op een eenvoudige manier gebruik maken van diensten die draaien op Windows-apparatuur.

1.4 Keuze voor Service-Oriented Architecture

Het .NET framework levert net zoals alle andere ontwikkelmethoden diverse mogelijkheden om applicaties te ontwikkelen. Service-Oriented Architecture is echter dé manier om applicaties aan elkaar te koppelen en deze te laten samenwerken via een standaard communicatieprotocol. De heldere architectuur zorgt ervoor dat ontwikkelaars applicaties als netwerkdiensten kunnen gebruiken. Deze kunnen eenvoudig aan elkaar worden gekoppeld zodat ze een complex bedrijfsproces vormen. Daarnaast geeft het ontwikkelaars een gedegen framework dat per definitie zorgt voor goed gestructureerde en degelijke applicaties. SOA maakt hergebruik van webservices (code, functionaliteit en applicaties) niet alleen mogelijk, maar zorgt er bovendien voor dat dit zeer eenvoudig wordt en weinig risico's met zich meer brengt. Ontwikkelaars hoeven namelijk alleen maar de interfaces tussen bestaande applicaties te maken wanneer er een nieuwe set van business rules wordt ontwikkeld. Deze nieuwe interfaces zijn door de structuur van SOA dunner dan in andere architecturen. Hierdoor maakt SOA het mogelijk sneller betere applicaties te ontwikkelen tegen lagere kosten.

1.5 Wat is SOA?

SOA is een standaard architectuur voor softwareontwikkelaars die definieert waar welke functionaliteit van een applicatie wordt geplaatst. In Figuur 2 is te zien dat elke applicatie kan worden opgedeeld in vijf lagen, te weten een users-, user interface-, facades-, business- en data laag.



Figuur 2; Overzicht van Service-Oriented Architecture

De eerste laag (users) bestaat uit de gebruikers. De tweede laag (user interface) biedt deze gebruikers een weergave van de applicatie. Door de soort weergave af te stemmen op het apparaat (de hardware) die de gebruiker op dat moment in handen heeft, biedt SOA te allen tijde een voor deze hardware optimale interface. De facades vormen de brug tussen de user interfacelaag en de businesslaag. Omdat een subset van de functionaliteit van de businesslaag wordt aangeboden aan de users interfacelaag, zijn de mogelijkheden van webapplicaties in een browser per definitie beperkter dan op de desktop van een rich client.

In de businesslaag wordt alle business logic opgeslagen, zoals datatransformatie, koppeling en andere business rules die relevant zijn voor de applicatie. De businesslaag communiceert op zijn beurt weer met de data laag. De gebruikte data wordt door middel van stored procedures of SQL statements beschikbaar gesteld in de data access layer.

Zoals de naam SOA al zegt is de architectuur gericht op het leveren van diensten. Bij het ontwikkelen van een applicatie met SOA is het doel om de gedeelde functionaliteit of dienst beschikbaar te maken voor verschillende clients, zoals een PC of PDA. Hierbij wordt gebruik gemaakt van de gedeelde onderliggende functionaliteit en de data die in onderliggende lagen ligt opgeslagen.

2 In de praktijk

Delta-N heeft gekozen om bij de ontwikkeling van .NET-applicaties in de meeste gevallen volgens het SOA-principe te werken omdat uit de praktijk blijkt dat dit het beste resultaat oplevert voor de klant.

Een door Delta-N bij Displays-ICT uitgevoerd project laat zien waar de voordelen van SOA in de praktijk liggen.

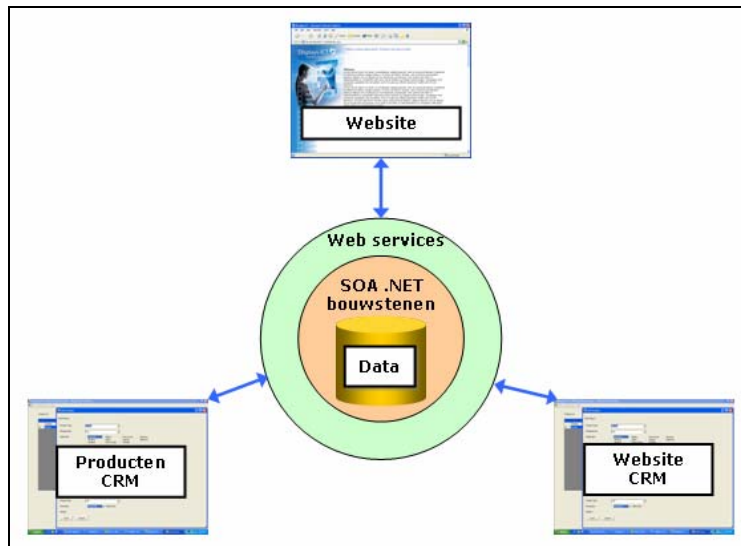
2.1 Projectbeschrijving

Displays-ICT is gespecialiseerd in het leveren van monitoren, projectoren en toebehoren aan de zakelijke markt. Zij verkoopt en verhuurt producten en biedt consultancydiensten en ondersteuning. Het bedrijf biedt haar klanten hoogwaardige, maar voordelige oplossingen. Enkele recente opdrachtgevers zijn: Aselsan Turkije (shockproof LCD-panelen), Shell (officemonitoren voor het gehele hoofdkantoor), Siemens Nederland (monitoren voor Special Products)

Displays-ICT vroeg Delta-N een nieuwe website te maken, omdat de functionaliteit van de oude te wensen overliet. Daarnaast moest er een content managementsysteem (CMS) worden opgezet zodat de inhoud van de nieuwe website op eenvoudige wijze kan worden aangepast. In het CMS bouwde Delta-N bovendien een uitgebreid product managementsysteem. De webpagina's, producten en instellingen werden opgeslagen in een database.

2.2 Keuze voor SOA

Het CMS moest als een windows applicatie ontwikkeld worden, omdat deze een complexe userinterface heeft en veel validatie functionaliteit vereist. Er is bewust niet gekozen voor de ontwikkeling van een webapplicatie vanwege een langer traject met een duurder prijsplaatje. De website wordt uiteraard ontwikkeld voor een browserinterface. SOA is uitstekend geschikt voor dit soort hybride oplossingen. Beide applicaties maken namelijk gebruik van gemeenschappelijke diensten die worden aangeboden via webservices. Daardoor hoefden de business rules slechts één keer te worden geïmplementeerd, om vervolgens door de beide applicaties te worden gebruikt.



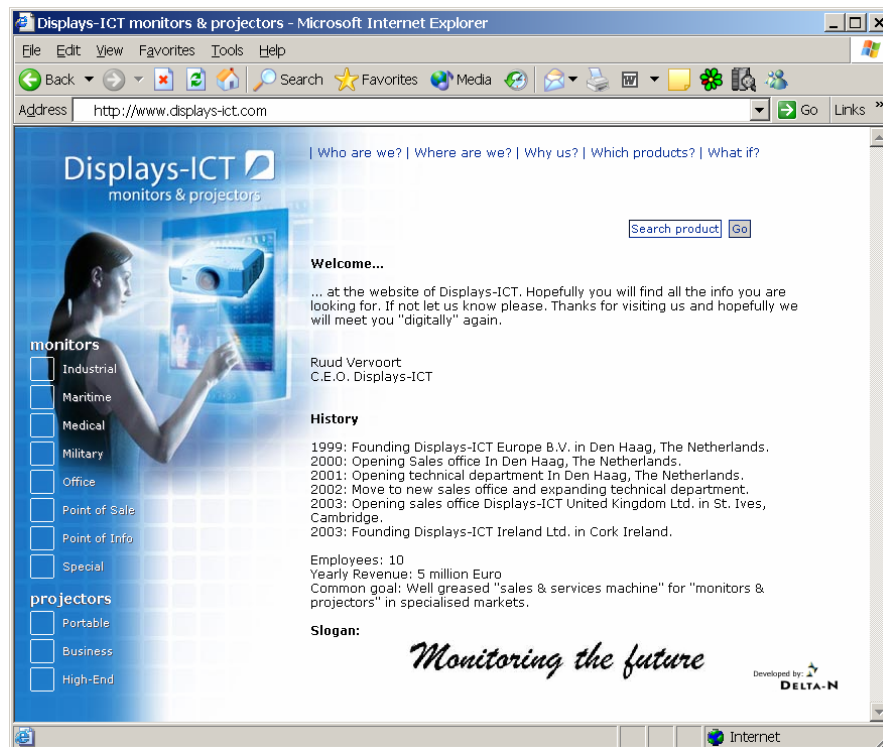
Figuur 3; Overzicht architectuur van het project

2.3 Toekomstvisie

Met het oog op mogelijke toekomstige uitbreiding van de activiteiten van Displays-ICT, was de ontwikkeling met een SOA-architectuur een verstandige keuze. Bij uitbreiding van de business rules (zoals nieuwe kortingsregelingen), kunnen deze op zeer eenvoudige wijze in de businesslaag worden geplaatst en vervolgens in de verschillende applicaties worden gebruikt. Wanneer een nieuwe applicatie of website nodig is, bijvoorbeeld een nieuwe webwinkel, kunnen de business rules eenvoudig worden overgenomen. Ook wanneer een website zwaar belast wordt door veel gebruikers, biedt SOA een eenvoudige oplossing. Zo kan zonder veel moeite een extra server worden geplaatst. Het plaatsen van extra servers kan ook nodig zijn wanneer van een website hoge stabiliteit en robuustheid gevraagd wordt.

2.4 Veiligheid

Voor het .NET-tijdperk werd de beveiliging geregeld in het front-end van de applicatie. Dit maakte onderliggende onderdelen van de applicatie kwetsbaar. Het SOA-concept vereist echter beveiliging in alle lagen van het systeem. Daarnaast kan door gebruik te maken van de standaard beveiliging van het HTTP-protocol, eenvoudig het gewenste veiligheidsniveau worden bereikt.



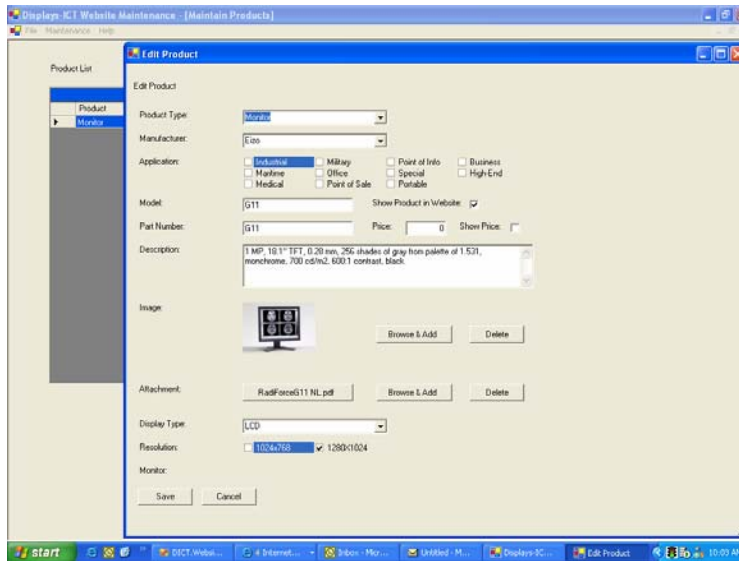
Figuur 4; Screenshot van de nieuwe website van Displays-ICT

2.5 Snelheid

Het gehele systeem, inclusief website, CMS en databases, is in zeer korte tijd (6 weken) ontwikkeld. Dit was mogelijk omdat Delta-N een goed gevulde bibliotheek heeft met eenvoudig in te zetten SOA-bouwstenen (webservices). Functionaliteiten zoals het presenteren, opslaan en plaatsen van plaatjes op een website kunnen worden gecreëerd met deze bouwstenen.

Degelijkheid

Door de inzet van deze bouwstenen, wordt de kans op fouten in nieuwe applicaties geminimaliseerd. De bouwstenen zijn niet alleen grondig getest, maar worden ook al met succes gebruikt in andere applicaties. Doordat de interfaces relatief dun zijn in vergelijking met traditioneel ontwikkelde onderdelen en doordat SOA voorschrijft wat waar in het systeem hoort te staan, zijn nieuw ontwikkelde onderdelen bovendien eenvoudiger te testen. Deze heldere architectuur maakt het mogelijk applicaties sneller en op een eenvoudige manier aan te passen. Deze voordelen zullen naar voren komen in de vervolgoopdracht die Delta-N voor Displays ICT zal uitvoeren.



Figuur 5; Screenshot van de ontwikkelde product managementapplicatie

2.6 De producten

De opgeleverde producten, zowel de website als het content managementsysteem, zijn zeer degelijke, snel opgeleverde en goedkope applicaties geworden waar opdrachtgever Displays-ICT zeer tevreden mee is.

3 Samenvatting

Service-Oriented Architecture is de nieuwste en meest geavanceerde methode voor het ontwikkelen van zowel internet-, intranet- als desktopapplicaties. SOA maakt snellere, betere en goedkopere applicatieontwikkeling mogelijk. Daarom gebruikt Delta-N SOA voor de ontwikkeling van haar applicaties. Delta-N kan haar klanten hierdoor snel goedkope en kwalitatief hoogstaande applicaties leveren.

3.1 Interesse

Wilt u meer weten over Delta-N, applicatieontwikkeling, .NET of Service-Oriented Architecture, neemt u dan contact op met Marcel Timmers (marcelt@delta-n.nl), telefoon: 070-3575700

3.2 Links en referenties

Contactinformatie en meer informatie over onze bezigheden, als ook referenties vindt u op de Delta-N website;
<http://www.delta-n.nl>

Meer informatie over .NET vindt u op de website van Microsoft:
<http://www.microsoft.com/net/basics>

Meer informatie over Service-Oriented Architecture vindt u op:
<http://www.developer.com/tech/article.php/1041191>