.net column

.NET Remoting

Aanleiding voor een nieuwe generatie gedistribueerde applicaties?

Voor de komst van het Microsoft .NET framework, werd DCOM als onderliggende technologie gebruikt voor de communicatie van gedistribueerde Windows applicaties. Echter, DCOM was gebaseerd op een proprietary binair protocol en kent een aantal nadelen, voornamelijk in het setup- en configuratieproces.

Toen er behoefte ontstond aan de fysieke distributie van op COM gebaseerde applicaties, was het voor de hand liggende antwoord de toevoeging van een netwerk extensie. DCOM werd deze netwerkextensie en voegde een nieuwe infrastructuur toe aan de zeer succesvolle componenten technologie.

DCOM is een sterk connection oriented protocol dat onder water gebruikt maakt van Remote Procedure Calls (RPC) voor de overdracht van informatie. Een belangrijke eigenschap van DCOM is het vermogen om automatisch proxy- en stubmodules te creëren, benodigd voor de interproces- en intermachinecommunicatie, waardoor een transparant programmeermodel ontstaat voor zowel in-process, out-of-process als remote calls.

De .NET Remoting-architectuur biedt een grote mate van flexibiliteit en maakt het mogelijk om gebruik te maken van verschillende transportprotocollen, serialisatieformaten, objectlifetime technieken en methoden van objectcreatie. Twee belangrijke onderdelen van .NET Remoting zijn Channels en Formatters. Een channel is een element in de .NET Remoting-architectuur dat fysiek bytes transporteert van het ene endpoint naar het andere. Een channel ontvangt een array van bytes, creëert een package volgens een bepaald protocol (formatter) en routeert naar een endpoint over een remotinggrens. In het .NET framework worden twee channel classes meegeleverd, TcpChannel en HttpChannel. Het TcpChannel maakt gebruik van een binaire formatter om de data te serialiseren en transporteert deze data over het TCP/IP protocol.

Het HttpChannel transporteert berichten via het HTTP protocol en maakt gebruik van een SOAPformatter.

De flexibiliteit van de .NET Remoting-architectuur zit in de mogelijkheid om eigen channels, formatters en channel sinks te bouwen, waarbij channel sinks de mogelijkheid bieden out-of-band data mee te sturen of om berichten te onderscheppen (interception), iets wat met DCOM zeker niet eenvoudig te realiseren is.

Ondanks de enorme flexibiliteit van de .NET Remoting-architectuur, heeft DCOM remoting op dit moment meer functionaliteit. In grote enterprise systemen wordt er vaak gebruik gemaakt van COM+ services bestaande uit: transaction management, security, synchronisatie, *loosely coupled events*, queued components, compensation resource manager en object pooling. Voor remoting tussen gedistribueerde COM+ applicaties wordt er nog DCOM gebruikt, dat geheel transparant transactie- en security context propagatie kan verrichten, iets wat niet mogelijk is met de standaard meegeleverde .NET channels. Natuurlijk is het mogelijk om met *custom channel sinks* authenticatie services (security) in te bouwen, maar de vraag is of dat niet tot de standaard infrastructuur zou moeten behoren.

Met de nieuwe .NET Remoting-architectuur speelt Microsoft in op de behoefte om gedistribueerde applicaties, gebaseerd op standaard en open protocollen te kunnen realiseren. DCOM is hiermee vergeleken een statisch, proprietary protocol met veel nadelen qua configuratie en deployment, maar met (op dit moment) meer functionaliteit.

Xander Buffart is trainer en ICT-Architect van Info Support .