

REAAL migreert B-2-B gateway met WCF

Leon Bakker

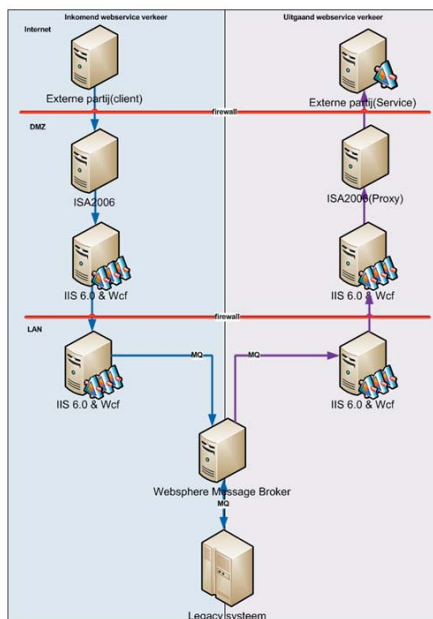
Sinds 2002 beschikt REAAL Verzekeringen over een Business-To-Business (B2B) Internet Gateway die alle in- en outbound webservice communicatie met externe partijen afhandelt. Deze B2B is gebouwd in Java en draaide sindsdien binnen een infrastructuur met Apache Tomcat. In 2007 is begonnen met de migratie van deze B2B naar het .NET platform.

Voor deze migratie hebben we gekozen voor is het gebruik van Windows Communicatie Foundation (WCF). Het uiteindelijke doel is een hogere mate van flexibiliteit en een hogere ontwikkelsnelheid bij het koppelen van een externe partij. Flexibiliteit moet gewaarborgd worden door het gebruik van WCF extensions voor het koppelen van de bestaande legacy (oude versies) en vermindering van afhankelijkheid van het onderliggende transportprotocol (BasicHTTP, MSMQ, MQ etc.). Het gebruik van de laatste standaarden op het gebied van webservices, wsdl en ws* moeten de ontwikkelsnelheid waarborgen. Hoe heeft het WCF framework REAAL Verzekeringen geholpen bij het realiseren van een vernieuwde Business-2-Business gateway?

Historie architectuur oude B2B

Voor inkomende webservices bestond de oude applicatie uit een webserver en een applicatieserver. De webserver bestond uit twee applicaties namelijk IIS 6.0 en Apache Tomcat, waarbij in Apache Tomcat werd gewerkt met een op maat gemaakte Java-implementatie die ervoor zorgde dat er gecommuniceerd kon worden met de applicatieserver. IIS zorgde in deze configuratie voor de SSL afhandeling en Apache Tomcat voor de berichtverwerking waarbij de SOAP envelope werd verwijderd. De applicatieserver ontvangt het bericht van de webserver door middel van RMI (Remote Method Invocation). De applicatieserver zet vervolgens het bericht op MQSeries van IBM en wacht op de reply.

Voor uitgaand webserviceverkeer hebben we een Java-applicatie ontwikkeld. Deze draait in een Windows service waarbij de Windows service luistert op een MQ queue. Via een proxyserver wordt de webservice bij de externe partij aangeroepen.



FIGUUR 1: ARCHITECTUUR OUDE B2B PORTAAL.

Nieuwe architectuur

In de nieuwe architectuur is er een plaats ingeruimd voor ISA Server 2006. Dit serverproduct van Microsoft handelt de SSL-verbinding vanuit het internet af. Inkomende berichten zijn voorzien van een clientcertificaat. Dit certificaat wordt in de ISA Server omgezet naar een Active Directory user.

Hierna wordt het bericht gerouteerd naar een server in de DMZ-omgeving. In deze

omgeving draait een WCF-service die gehost wordt in IIS 6.0. In deze WCF-service wordt het bericht gevalideerd en wordt bepaald naar welke WCF-service het bericht moet worden gestuurd in het LAN. Indien noodzakelijk wordt een XML-mapping uitgevoerd, zodat het inkomende bericht wordt vertaald naar een voor REAAL bekend formaat. De WCF-service in het LAN wordt gehost in IIS, deze WCF-service heeft als belangrijkste functionaliteit het bericht op MQ zetten en hierna wachten op het antwoord van de back-office.

Voor de applicatie voor uitgaand verkeer hebben we een extra webservicelaag geplaatst binnen het LAN. In de oude situatie waren de externe webservices voor onze interne applicaties alleen te benaderen via MQ. Alle externe services zijn nu intern aangeboden als WCF-service waarbij we een opdeling hebben gemaakt naar functioneel domein. Vanuit de LAN-omgeving wordt een WCF-service in de DMZ-omgeving aangeroepen. De WCF-service voert een XML-mapping uit van REAAL's canonisch XML-formaat naar het formaat van de externe service. Hierna wordt de aanroep naar de externe webservice gedaan.

Er was nog een aantal interne applicaties die alleen via MQ konden communiceren met de buitenwereld. Voor deze applicaties hebben we een component ontwikkeld die de protocol transformatie van MQ naar webservices uitvoert (de WCF-MQAdapter). Hierdoor kunnen deze applicaties ook met de externe service communiceren.

WCF in de nieuwe architectuur

In onze vernieuwde architectuur neemt WCF een belangrijke plaats in. Alle aangeboden webservices worden gerealiseerd met de standaardfunctionaliteit die het WCF-framework biedt. Hierbij werken we met ServiceContracten, MessageContracten, DataContracten en FaultContracten. Voor de exception handling wordt Enterprise Library 3.5 gebruikt. Daarnaast hebben we op maat gemaakte functionaliteit toegevoegd.

Het project was voorzien van een aantal requirements. Ten eerste hebben we elk bericht dat door ons webservicelandschap loopt, voorzien van een transaction id. Er is een behavior ontwikkeld die controleert of het bericht voorzien is van een transaction id ten behoeve van correlatie van alle berichten die binnen één transactie aangeschopt worden. Wanneer het id aanwezig is, doet de behavior niks, wanneer het id ontbreekt wordt deze toegevoegd in de vorm van een guid. Alles wat in de database wordt wegschreven, wordt voorzien van hetzelfde transaction id als het bericht waardoor berichten makkelijk terug te vinden zijn in de database. Ten tweede moesten de berichten die gecommuniceerd worden met externe partijen opgeslagen worden in een database. Hiervoor is de AuditBehavior gemaakt. Ten derde moesten alle berichten geteld worden. De ServiceAccountBehavior is hiervoor ontwikkeld.

WCFMQAdapter

De WCFMQAdapter is ontwikkeld om een brug te slaan tussen aan de ene kant MQSeries van IBM en aan de andere kant webservices gebouwd met het WCF-framework. De legacy systemen werken met een request/reply communicatiepatroon en zijn alleen te benaderen via MQ met zijn centrale Websphere Message Broker (Hub-and-Spoke). Tevens moest de WCFMQAdapter functionaliteit ondersteunen om vanuit een legacy systeem een WCF-service aan te roepen.

Om de communicatie tussen het WCF-framework en MQ mogelijk te maken, is een eigen MQ transport channel gebouwd. Hierbij is een implementatie gebouwd met de daarvoor bestemde interfaces IRequestChannel en IReplyChannel (zie: <http://msdn.microsoft.com/es-us/library/ms729840.aspx>). In deze implementatie is gebruik gemaakt van amqmdnet.dll assembly welke wordt meegeleverd door

het MQ product. In deze assembly zit onder andere functionaliteit om bijvoorbeeld een bericht op een Queue te zetten of eraf te halen.

Messages die worden verstuurd binnen het WCF framework hanteren het SOAP-protocol, waarbij SOAP de envelope is waarin het datacontract wordt verzonden. Echter binnen de Websphere Message Broker wordt een andere envelope gehanteerd. Voor de conversie van envelope is gebruik gemaakt van een zelf ontwikkelde MessageEncoder. Deze MessageEncoder heeft buiten de envelope conversie ook functionaliteit om de foutafhandeling te regelen tussen de twee protocollen. De transport channel en de message encoder vormen samen een nieuwe binding, de 'BusMQBinding'. Deze binding is binnen de WCF-configuratie op dezelfde manier te gebruiken als bijvoorbeeld de standaard basicHttpBinding. Daarom heeft ook de BusMQBinding een aantal andere properties om bepaalde zaken binnen de binding instelbaar te maken, zoals bijvoorbeeld Queues en time-out.

```
<?xml version="1.0"?>
<configuration>
  <ServiceModel>
    <services>
      <service
name="HelloWorld">

<endpoint address="MQ://QueueManager
Name//Queue_Name" behaviorConfiguration="
ServerEndpointBehavior"

binding="busMQBinding" bindingConfiguratio
n="busMQBindingConfig"
name="MQHelloWorldService"
contract="IHelloWorld" />
      </service>
    </services>
    <Bindings>
      <busMQBinding>

<binding name="busMQBindingConfig" closeT
imeout="00:00:15" replyQueueName="Reply_
Queue_Name" mqPollingInterval="100" />
      </busMQBinding>
    </Bindings>
    <extensions>
      <bindingExtensions>

<add name="busMQBinding"
type="Verzekeraar.Framework.Esb.WcfMQAdap
ter.BusMQBindingCollectionElement, Verze
keraar.Framework.Esb.WcfMQAdapter, Version
=1.0.0.0, Culture=neutral, PublicKeyToken=
470bd532707c672d" />
      </bindingExtensions>
    </extensions>
  </ServiceModel>
</configuration>
```

FIGUUR 2: BUSMQBINDING CONFIGURATIE.

Voordelen WCF

Het WCF-framework is op elke plek wel op maat te maken waardoor het heel flexi-

bel wordt en waardoor alle functionaliteit die niet door het framework wordt geleverd, zelf toe te voegen is.


Het voordeel voor REAAL verzekeringen is het feit dat het WCF-framework een gestandaardiseerde manier biedt voor het aanbieden van de webservices aan externe partijen. Door het overleggen van een correcte wsdl moet het voor externe partijen makkelijker worden om aan te sluiten op de webservice bij REAAL Verzekeringen.

Nadelen WCF

Tijdens het project was het de bedoeling om de WCF-services gebruik te laten maken van de wsHttpBinding, om zo makkelijk aan de WS* standaarden te kunnen voldoen. Echter tijdens de implementatie van deze wsHttpBinding is een aantal problemen gevonden in combinatie met de infrastructuur.

- Certificaat afhandeling door ISA 2006 server geeft problemen in combinatie met wsHttpBinding. ISA 2006 mist de SOAP header waardoor de service het bericht niet meer begrijpt.
- Bij gebruik van wsHttpBinding en message security wordt een aantal berichten tussen de client en service uitgewisseld. Network Load Balancing zorgt ervoor dat de berichten worden verdeeld over de servers, hierdoor kan de wsHttpBinding zijn communicatie protocol niet hanteren want de handshake wordt nooit voltooid.

Conclusie

Windows Communication Foundation heeft REAAL Verzekeringen ondanks wat kleine minpunten enorm geholpen bij het realiseren van dit project. Door middel van de diverse extensions heeft WCF ervoor gezorgd dat we de bestaande functionaliteit van de java gateway één-op-één konden aanbieden, zodat de bestaande externe partijen geen wijzigingen in de programmatuur hoefden te doen. Tevens hebben we extra functionaliteit kunnen toevoegen door middel van de extensions, waarbij we dus de ServiceInterface gelijk konden houden. In de toekomst wordt het voor REAAL Verzekeringen makkelijker om externe koppelingen te realiseren waarbij door middel van het WCF framework aan de marktstandaarden voldaan kan worden. 

Leon Bakker, is Software Architect bij de afdeling .NET van REAAL Verzekeringen. Hij is bereikbaar via email leon.bakker@reaal.nl.