Identiteit naar de Cloud met WIF

WCF services veilig migreren naar Windows Azure

Erwyn van der Meer

Windows Identity Foundation (WIF) maakt het mogelijk om authenticatie los te weken uit WCF services en ASP.NET applicaties zodat deze gemigreerd kunnen worden naar Windows Azure zonder gevoelige inloggegevens in de cloud op te hoeven slaan.

Veel applicaties maken gebruik van het invoeren van gebruikersnaam/wachtwoord op een formulier of van Integrated Windows Security om gebruikers te identificeren. Als een applicatie zelf de gebruikersnaam en het wachtwoord moet kunnen controleren, dan heeft deze applicatie toegang nodig tot bijvoorbeeld een database waarin deze gegevens zijn opgeslagen. Voor Integrated Windows Security is het nodig dat de server waarop de applicatie draait onderdeel uitmaakt van een Active Directory domein. Dit geeft complicaties in een scenario waarin de applicatie gehost wordt buiten de bedrijfsmuren. Dit artikel is gebaseerd op een daadwerkelijk uitgevoerd voorbeeld uit de praktijk en toont hoe Windows Identity Foundation flexibiliteit geeft aan hoe en waar er geauthenticeerd wordt. Bijvoorbeeld on premises met een wachtwoord dat niet verstuurd hoeft te worden aan de applicatie die off premises draait.

Probleemstelling

Stel je een bedrijf voor met locaties verspreid over de hele wereld van waaruit medewerkers toegang nodig hebben tot dezelfde data. Die medewerkers gebruiken daarvoor een .NET based client applicatie die via een webservice connecteert naar data in een SQL Server database. Die database bevindt zich op een server in een datacentrum op één locatie en daar wordt ook de webservice gehost. Veel van die wereldwijde locaties zijn dus ver verwijderd van de data en de gebruikers lopen daar tegen latency issues aan. Het kan honderden milliseconden duren voordat TCP/IP pakketjes de andere kant van de wereld bereiken. Het is voor het bedrijf erg kostbaar om wereldwijd datacentra in te richten om servers dichter bij de gebruikers te plaatsen en aldus de performance te verhogen. Microsoft heeft die datacentra al en biedt de Windows Azure Platform dienst aan vanuit zes datacentra op drie continenten. Misschien wil het bedrijf niet dat de gegevens waarmee een gebruiker zich identificeert buiten het eigen datacentrum worden verstuurd en/of opgeslagen. Hoe is het mogelijk om dan toch veilig een webservice aan te roepen die zich buiten de bedrijfsmuren bevindt? Om die vraag uit te werken, introduceren we eerst kort de benodigde concepten en technologieën.
In het dagelijks leven identificeren we ons bijvoorbeeld met een rijbewijs. Eigenlijk is dat niet meer dan een verzameling claims zoals naam, geboortedatum, burgerservicenummmer, welke typen voertuigen je mag besturen en echtheidskenmerken. Die echtheidskenmerken maken het mogelijk om als vertrouwende partij te controleren dat het geen nagemaakt rijbewijs is en dat je de claims kunt vertrouwen als je de uitgevende partij vertrouwd. Een claims gebaseerde identiteit kan gezien worden als een digitale vorm van een rijbewijs, maar dan met rijkere mogelijkheden voor het type claims dat daarin is opgenomen.

Windows Identity Foundation

Windows Identity Foundation (WIF) is een uitbreiding op het .NET Framework om programmatisch om te kunnen gaan met claims gebaseerde identiteit. Met WIF kan zo'n identiteit opgebouwd, uitgewisseld, gecontroleerd en uitgelezen worden. WIF breidt het bestaande identiteitsmodel van het framework, de IPrincipal en IIdentity interfaces, uit tot IClaimsPrincipal en IClaimsIdentity en heeft een Claim klasse. Zie codevoorbeeld 1. Het belangrijkste onderdeel van WIF is de assembly Microsoft.IdentityModel.dll waarin deze interfaces en klasse zich bevinden.

```
public interface IClaimsPrincipal : IPrincipal
{
    …
    // Properties
    ClaimsIdentityCollection Identities { get; }
}

public interface IClaimsIdentity : IIdentity
{
    …
    ClaimCollection Claims { get; }
}

public class Claim
{
    …
    // Properties
    public virtual string ClaimType { get; }
    public virtual string Issuer { get; }
    public virtual IClaimsIdentity Subject { get; }
    public virtual string Value { get; }
}
```

CODEVOORBEELD 1: ICLAIMSPRINCIPAL EN IIDENTITY INTERFACES EN CLAIM KLASSE.

Type claims die we bijvoorbeeld kunnen gebruiken zijn de naam van de gebruiker en zijn rollen.

De verantwoordelijkheid voor authenticatie van de gebruiker wordt overgenomen door een identity provider. Deze geeft een security token uit met daarin claims. Dit token wordt voorzien van een digitale handtekening. Dat is het echtheidskenmerk waarmee de webapplicatie kan verifiëren dat de claims uitgegeven zijn door een vertrouwde partij. Er zijn verschillende formaten tokens. Een voorbeeld is een SAML 2.0 token (SAML staat voor Security Assertion Markup Language). Een identity provider wordt ook wel een Security Token Service genoemd (STS). WIF ondersteunt standaard een aantal formaten en protocollen voor het uitwisselen van security tokens en deze ondersteuning kan uitgebreid worden door op de goede punten op het framework in te haken.

Windows Identity Foundation integreert met ASP.NET via een aantal HTTP modules. Na installatie van de WIF SDK kunnen deze modules door middel van een wizard in Visual Studio aan de web.config toegevoegd worden. Het resultaat staat in codevoorbeeld 2. Hierin staat ook een endpoint behavior opgenomen voor de integratie met een WCF service.

```
<system.webServer>
    …
    <modules>
        <add name="WSFederationAuthenticationModule"
            type="Microsoft.IdentityModel.Web.WSFederation-
AuthenticationModule,
            Microsoft.IdentityModel, Version=3.5.0.0, …"/>
        <add name="SessionAuthenticationModule"
            type="Microsoft.IdentityModel.Web.Session-
AuthenticationModule,
            Microsoft.IdentityModel, Version=3.5.0.0, …"/>
    </modules>
</system.webServer>

<system.serviceModel>
    …
    <extensions>
        <behaviorExtensions>
            <add name="federatedServiceHostConfiguration"
                type="Microsoft.IdentityModel.Configuration.
ConfigureServiceHostBehaviorExtensionElement,Microsoft.
IdentityModel,
Version=3.5.0.0, … "/>
        </behaviorExtensions>
    </extensions>
</system.serviceModel>
```

CODEVOORBEELD 2: WIF INTEGRATIE VIA WEB.CONFIG.

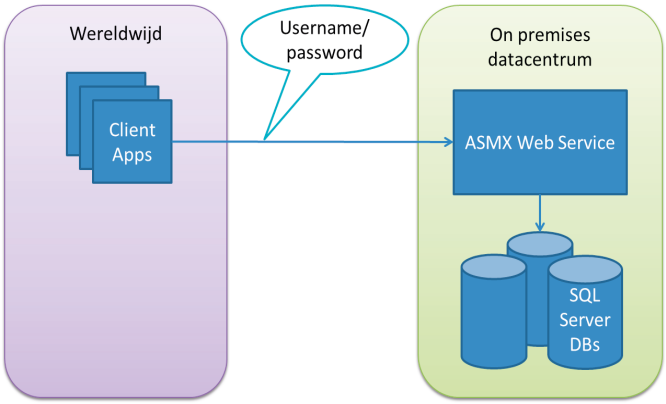
Het zou teveel ruimte innemen om in te gaan op hoe deze integratie precies werkt. Als WIF juist geconfigureerd is, zijn de claims als volgt uit te lezen in een ASP.NET pagina:

```
<asp:Repeater runat="server" ID="Claims">
    <HeaderTemplate><table></HeaderTemplate>
    <ItemTemplate>
        <tr>
        <td><%# Eval("ClaimType") %></td>
        <td><%# Eval("Value") %></td>
        </tr>
    </ItemTemplate>
    <FooterTemplate></table></FooterTemplate>
</asp:Repeater>

IClaimsPrincipal icp = Thread.CurrentPrincipal
asIClaimsPrincipal;
IClaimsIdentity ici = icp.Identity as IClaimsIdentity;
Claims.DataSource = from Claim c in ici.Claims select c;
Claims.DataBind();
```

CODEVOORBEELD 3: CLAIMS UITLEZEN.

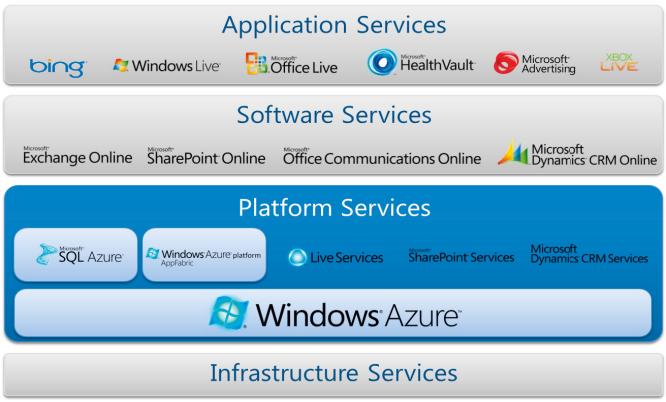
In afbeelding 1 staat de uitgangssituatie weergegeven waarbij een informatieleverende ASMX webservice zelf de aanroeper authenticeert door een opgegeven gebruikersnaam en wachtwoordcombinatie te vergelijken met in een database opgeslagen waarden. Als de aanroeper gerechtigd is, wordt informatie uit een andere database geretourneerd. We introduceren nu het Windows Azure Platform om deze webservice naar toe te kunnen migreren.



AFBEELDING 1: UITGANGSSITUATIE.

Windows Azure Platform

Het Windows Azure Platform is een zogenaamde Platform as a Service dienst waarbij je op een zeer schaalbare manier .NET applicaties kunt hosten en data kunt opslaan bij Microsoft. De afnemer betaalt alleen voor verbruik, dus geen opstartkosten. SQL Azure is de relationele database dienst binnen het Windows Azure Platform en maakt het mogelijk om on premises SQL Server databases naar de cloud te migreren.
Zie afbeelding 2 voor de positionering van het Windows Azure Platform binnen het cloud diensten portfolio van Microsoft.



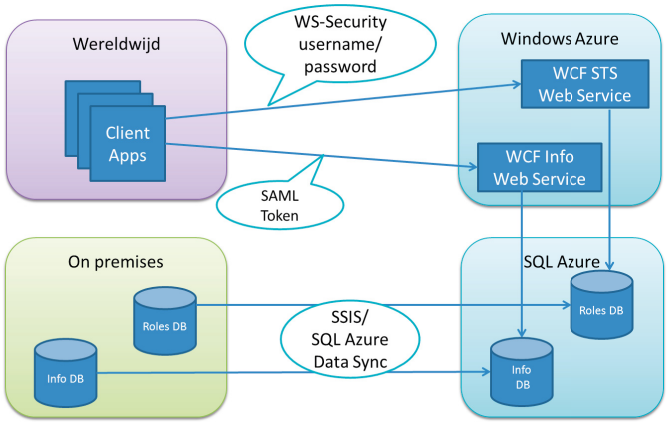
AFBEELDING 2: MICROSOFT CLOUDDIENSTEN.

Windows Azure ondersteunt onder andere .NET 2.0, 3.0, 3.5 en .NET 4 applicaties. Applicaties hoeven niet persé te migreren naar de laatste versie van het .NET Framework om op Windows Azure te kunnen landen, maar dit biedt wel voordelen. WIF kan goed gebruikt worden WCF web services, maar niet zo goed vanuit de oudere ASMX technologie voor web services en niet direct vanuit Java en PHP.

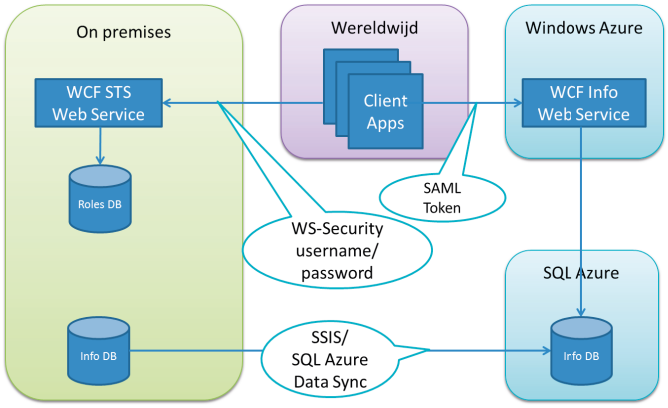
De oplossing

In afbeelding 3 staat de oplossing weergegeven na migratie naar Windows Azure en SQL Azure met gebruik van WCF en WIF.

De client applicatie verstuurt de gebruikersnaam en wachtwoord nu naar een Security Token Service (STS) in een bericht met beveiliging die voldoet aan de WS-Security standaard. De STS verifieert de informatie en verstrekt een SAML token. Dat SAML token wordt vervolgens door de client applicatie verstuurd naar de



AFBEELDING 3: OPLOSSING NA MIGRATIE.



AFBEELDING 4: ALTERNATIEVE DEPLOYMENT.

informatieleverende webservice. Deze verifieert het token en verstrekt informatie indien de gebruiker geautoriseerd is. De services identificeren zich met X509 certificaten daarmee worden ook de tokens en de communicatie beveiligd.
De SQL Azure databases worden via SQL Azure Data Sync periodiek gevuld met data vanuit databases die on premises gehost worden. De initiële vulling is gedaan met SQL Server Integration Services (SSIS).
Er zijn deployments van de services en databases in meerdere Windows Azure datacentra zodat de latency voor de gebruikers verkleind wordt.
Nu de authenticatie ontkoppeld is kunnen we een stap terug zetten en de STS on premises hosten zodat de databases met gevoelige gebruikersgevens niet in de cloud terecht komen. De latency bij de eerste aanroep wordt weer groter. Zie afbeelding 4 voor dit alternatieve deploymentscenario. Er zijn hiervoor geen codeaanpassingen nodig.

De Security Token Service kan gerealiseerd worden met custom code die WIF gebruikt. Het is ook mogelijk om een standaard STS te gebruiken zoals het product Microsoft Active Directory Federation Services (ADFS) 2.0 of zelfs een STS die op een ander platform dan Windows draait. De kracht van het externaliseren van authenticatie is dat het voor de informatie leverende webservice niet meer uitmaakt hoe en waar er geauthenticeerd wordt, zolang er maar standaarden gevolgd worden voor de security tokens die uitgewisseld worden. De service ontvangt claims voor de gebruikersnaam en de rollen en hoeft slechts de uitgevende partij te vertrouwen.

Conclusie

Windows Identity Foundation is krachtige technologie voor het .NET platform om via claims gebaseerde identiteit de verantwoordelijkheid voor authenticatie los te kunnen koppelen van applicaties. Dit geeft meer deploymentflexibiliteit en werkt goed samen met hosting op het Windows Azure Platform. Dit artikel geeft slechts een klein gedeelte van de mogelijkheden van WIF en Windows Azure weer. Hieronder staan goede bronnen vermeld om meer te leren over WIF.

Links

Windows Identity Foundation home on MSDN
http://msdn.microsoft.com/en-us/security/aa570351.aspx
Windows Identity Foundation SDK
http://www.microsoft.com/downloads/en/details.aspx?FamilyID=c148b2df-c7af-46bb-9162-2c9422208504&displaylang=en
Identity Training Toolkit
http://www.microsoft.com/downloads/en/details.aspx?displaylang=en&FamilyID=c3e315fa-94e2-4028-99cb-904369f177c0
Boek “Programming Windows Identity Foundation” van Vittorio Bertocci, Microsoft Press, ISBN 978-0735627185
Windows Azure Developer Center
http://msdn.microsoft.com/en-us/windowsazure/default.aspx

Erwyn van der Meer, werkt als application platform consultant bij Microsoft Services en specialiseert zich in het Windows Azure Platform. Hij is te bereiken via erwyn.van.der.meer@microsoft.com