GLOBETEAM

Danmarks Miljøportal (DMP)

Vejledning til fagsystemejere omkring tilkobling af .NET-baseret web applikation

Version 1.4

Indledning

Denne vejledning beskriver hvordan man claims-enabler en .Net-baseret web applikation til brug med DMP's føderale brugerstyringsløsning ved brug af Microsoft Windows Identity Foundation (WIF).

Vejledningen skal læses i konteksten af den overordnede vejledning "Vejledning til fagsystemejere omkring forløbet for tilkobling af en applikation".

Vejledningen forudsætter at Microsoft Windows Identity Foundation (WIF) benyttes til claims-enabling af web applikationen. Denne vejledning er således ikke brugbar, såfremt man ønsker at anvende et andet API til claims-enabling af web applikationen.

Indhold

Forudsætninger for tilkobling af en .NET-baseret web applikation

Såfremt de involverede .NET-udviklere ikke allerede er fortrolige med claims og claims-enabling af web applikationer på .NET-platformen kan det anbefales at studere Microsofts onlinebog "A Guide to Claims–based Identity and Access Control" fra Patterns & Practices-serien (http://msdn.microsoft.com/en-us/library/ff423674.aspx).

WIF

Som nævnt i indledningen er vejledningen bygget på en forudsætning om at claims-enablingen sker ved brug af Windows Identity Foundation (WIF).

WIF tager rent praktisk form af en update og kan hentes på Download Center (http://www.microsoft.com/downloads/details.aspx?FamilyID=eb9c345f-e830-40b8-a5fe-ae7a864c4d76&displaylang=en). Der findes også en version af WIF til Windows 2003, der kan hentes på http://www.microsoft.com/downloads/details.aspx?FamilyID=be4db6a0-b76d-446d-810c-ea3c25b3969a&displaylang=en.

WIF kan afvikles på Windows Server 2012 R2, Windows Server 2008 R2, Windows Server 2008 Service Pack 2, Windows 2003 SP2, Windows 7 og Windows Vista Service Pack 2. WIF til Windows 2008 forudsætter at der allerede er installeret IIS 7.0 og Microsoft .NET Framework 3.5 eller højere på platformen, mens WIF til Windows 2003 forudsætter at der er IIS 6.0 og Microsoft .NET Framework 3.5 eller højere på platformen.

Det er et krav at web applikationen er bygget i .Net 3.5 SP1 samt IIS 7.0 (dog IIS 6.0 ved anvendelse af Windows Server 2003).

ASP.NET 2.0 skal således være registreret i IIS.

Eftersom WIF ikke indeholder støtte af SAML 2.0-protokollen er det forøvrigt en implicit forudsætning at web applikationen benytter WS-Federation (og dermed SAML 1.1-tokens) til kommunikationen med DMP. Dette er dog ikke noget, der er synligt for hverken udvikleren eller brugerne, idet disse teknikaliteter er gemt nede bag WIF, lige bortset fra at enkelte claims ser en smule anderledes ud (det sidste ":" i SAML 2.0 claim'ens navn er erstattet med et "/" for at gøre claim'en kompatibel med SAML 1.1).

.NET Framework 3.5 SP1

Det er et krav at anvende mindst .NET Framework 3.5 SP1. Windows Identity Foundation kræver .NET Framework 3.5 SP1.

Certifikaterne er din største fejlkilde

Når man ser bort fra deciderede kodefejl i selve web applikationen, så er brugen og konfigureringen af certifikater den absolut største kilde til fejl i forbindelse med claims-enabling af en løsning.

Derfor anbefales det på det kraftigste at at udviklere såvel som infrastrukturfolk får "Appendix D: Digital Certificates" (http://msdn.microsoft.com/en-us/library/ff359106.aspx) som pligtlæsning, idet langt størsteparten af de fejl og problemer, der forekommer i relation til claims-enabling, viser sig at udspringe i og omkring certifikaterne.

Tilkoblingen af en .NET-baseret web applikation

Sikkerheden for Web Applikationer der bruger WS-Federation passive på DMPs Safewhere*Identify STS-løsning er baseret på følgende standarder:

- WS-Federation passive requestor profile
- Kombineret transport- og message-baseret sikkerhed (dvs. HTTPS-transport med message-baseret kryptering og signering af token'et) – dvs. tokens er krypteret i http-beskederne.
- SAML 1.1 tokens

End points for DMP's STS er angivet nedenfor:

Passive log-in via WS-Federation passive

- https://log-in.miljoeportal.dk/runtime/ (produktionsmiljø)
- https://log-in.test.miljoeportal.dk/runtime/ (testmiljø)

Bemærk at samples'ne som udgangspunkt er bygget til brug med test miljøet. De vil dog naturligvis kunne tilpasses til at bruge prodsystemet blot ved at udskifte URL'erne (og evt. certifikater) på end points'ne.

Det anbefales så vidt muligt, at udviklerne tager udgangspunkt i de medfølgende samples, når der oprettes nye web applikationer. De mange opsætningsmuligheder vil ellers hurtigt kunne risikere at gøre det til en uoverskuelig opgave at konfigurere det ovenstående i hånden.

Sådan kobles web applikationen op mod DMP

Der er to fysiske forudsætninger for tilkoblingen af en .NET-baseret web applikation.

1) Der skal foreligge et X.509 SSL-certifikat fra en offentlig CA. Dette X.509-certifikat skal benyttes til transport security på web applikationen (SSL)
2) Der skal foreligge endnu et X.509 SSL-certifikat fra en offentlig CA. Dette X.509-certifikat skal benyttes til kryptering af tokens til på web applikationen

Hver enkelt web applikation, der kobles op mod DMPs Safewhere*Identify STS skal således være udstyret med disse certifikater.

Trin 1: Anskaffelse af de nødvendige certifikater

Der skal erhverves to certifikater til hver web applikation:

- Et SSL certifikat. SSL certifikatet skal være udstedt til common name = applikationens url. Feks. cn=www.koreprovebooking.dk
- Et certifikat til kryptering af beskeder mellem DMP og web applikationen. Dette certifikat er ikke at forveksle med SSL certifikatet, idet certikatet benyttes til at gøre det muligt for STS'en at kryptere beskeder, som kun web applikationen kan læse.

Som nævnt i den overordnede vejledning ("Vejledning til fagsystemejere omkring forløbet for tilkobling af en applikation") skal encryption certifikatet overleveres til DMP uden den private nøgle. Dvs. DMP skal have certifikatets public key-del.

Applikationens krypteringscertifikat

Krypteringscertfikatet skal erhverves hos en offentlig og anerkendt CA. Vi anbefaler især http://www.godaddy.com, da de er godkendte i alle større browsere og er markant billigere end de store spillere såsom Verisign og Thawte. Vi anbefaler tilsvarende at dette erhverves som et almindeligt SSL-certifikat.

Det skal som nævnt ikke bruges til SSL, men SSL-certifikaterne er generelt billigst, og de dækker de behov et certifikat skal opfylde for at kunne benyttes som krypteringscertifikat. Modsat et certifikat der benyttes til SSL-trafik, så skal dette certifikat ikke være udstedt til et bestemt common name. Common name er ligegyldigt for krypteringscertifikatet. Vi anbefaler dog stadig, at der vælges et sigende common name for certifikatet (f.eks. CN=<applikationsnavn>-encryption.<organisationsnavn>.dk).

Trin 2: Konfiguration af webapplikations web.config-fil

Webapplikationens web.config-fil skal tilpasses på nogle få punkter før .NET-applikationen fungerer efter hensigten.

Det anbefales at web.config-filen tilpasses manuelt, som beskrevet i Appendiks A. Alternativt kan Microsofts FedUtil-værktøj benyttes, hvilket bliver beskrevet i Appendix C.

Trin 3: Tilpasning af global.asax

Det er endvidere påkrævet at udføre en lille ændring af global.asax-filen i relation til protected void Application_Start(object sender, EventArgs e).

Sektionen med protected void Application_Start(object sender, EventArgs e) skal helt præcist ændres til nedenstående i global.asax.cs-filen før applikationen kompileres:

```
protected void Application_Start(object sender, EventArgs e)
{
   FederatedAuthentication.WSFederationAuthenticationModule.RedirectingToIdentityProvider +=
new
EventHandler<RedirectingToIdentityProviderEventArgs>(WSFederationAuthenticationModule_Red
irectingToIdentityProvider);
}
void WSFederationAuthenticationModule_RedirectingToIdentityProvider(object sender,
RedirectingToIdentityProviderEventArgs e)
{
   e.SignInRequestMessage.HomeRealm = Request.QueryString["whr"];
}
```

Tilretningen medfører at WIF altid videresender en evt. WHR-parameter til DMPs STS, hvilket betyder at brugerne har mulighed for at angive hvilken IdP de kommer fra og således sikre at de altid bliver redirect'et til denne.

Information omkring sikkerhedsrelateret logning
Claims er konsistente og valide repræsentationer af brugeren, der er logget ind via STS'en, som i vort tilfælde er DMP's føderale brugerstyring. Claims overføres til applikationen ved hjælp af sikre offentlige standarder, der garanterer for at overleveringen af claims til applikationen er sikker og pålidelig.

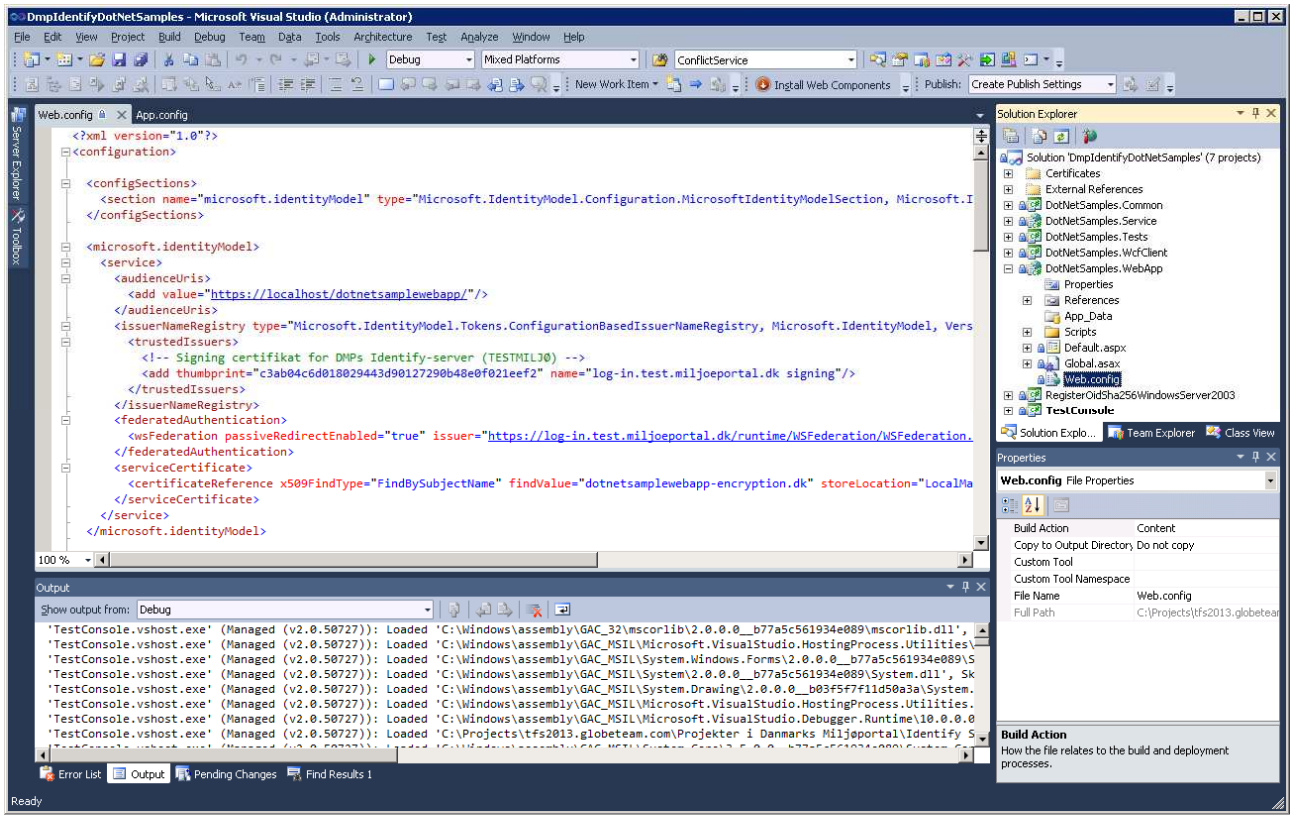
Hvis applikationen benytter sig af sikkerhedsrelateret logning kan brugerens claims således fint benyttes i denne sammenhæng. Der er med andre ord ikke nogen forskel på logning i et føderalt scenarie relativt til et traditionelt brugerlogin. Det er dog naturligvis vitalt at man tilsikrer at logningen som minimum indbefatter de claims, der sikrer at der kan ske en entydig identifikation af brugeren til et hvilke som helst senerekommende tidspunkt (dvs. logningen skal indbefatte en unik identifier såvel som de nødvendige informationer, der gør det muligt at finde tilbage til hvem denne unikke bruger var).

Step-by-step eksempel på tilkoblingen af de .NET-baserede samples

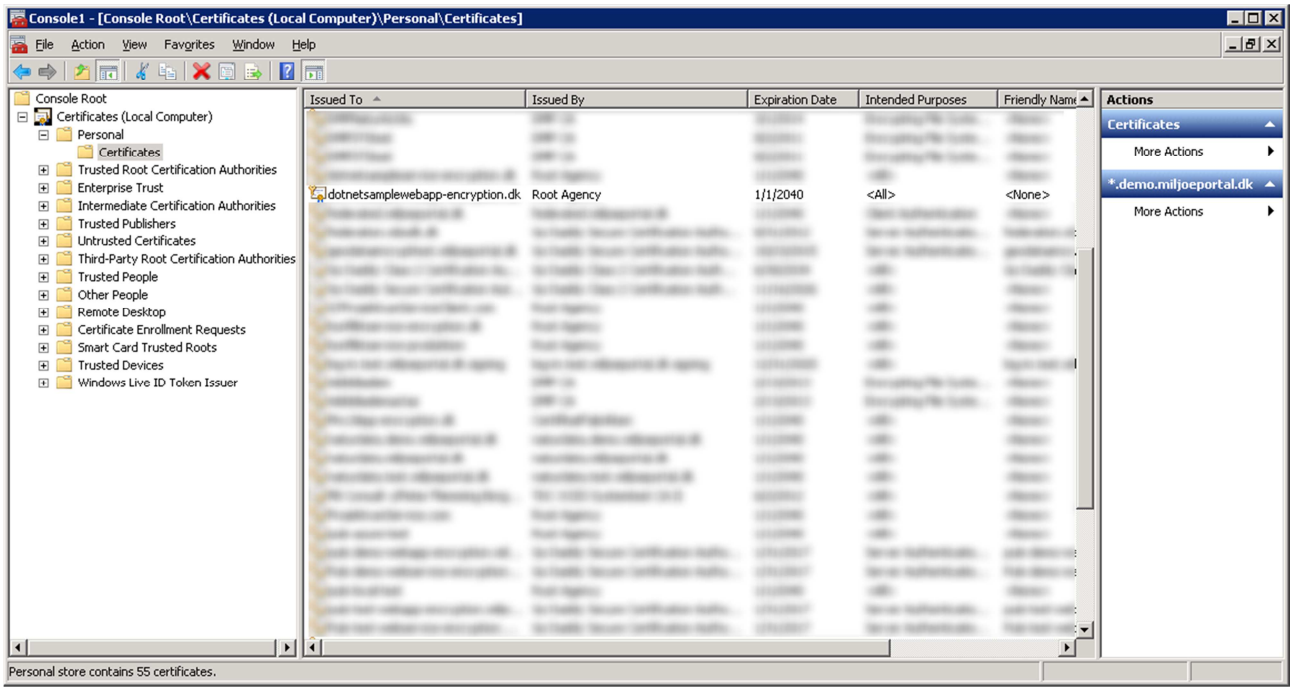
Følgende step-by-step guide tager udgangspunkt i de medfølgende samples i solution'en "DmpIdentifyDotNetSamples.sln" og beskriver, hvordan du får sample .NET web applikationen op at køre lokalt og får logget ind i web applikationen.

Når disse trin er udført vil du have en kørende .NET-baseret web applikation), der rekvirerer et token fra DMPs Safewhere*Identify STS og logger ind og viser token'ets indhold i en webside.

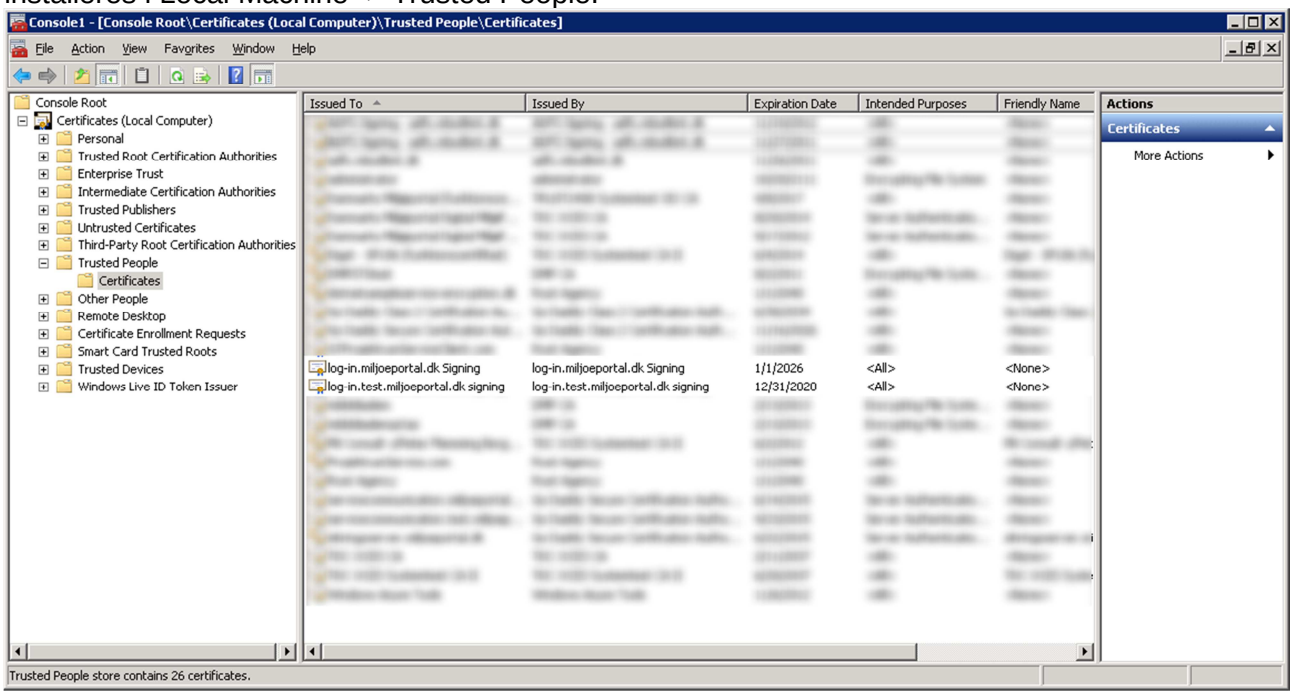
1. Åbn solution'en i Visual Studio 2010.



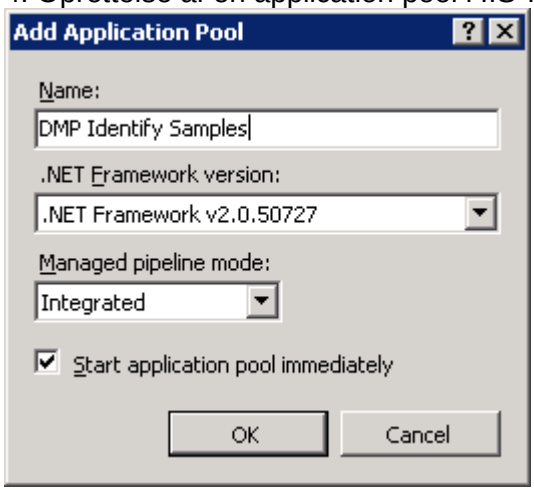
2. Certifikatet “dotnetsamplewebapp-encryption.dk.pfx” (password: pw) installeres i Local Machine -> Personal.



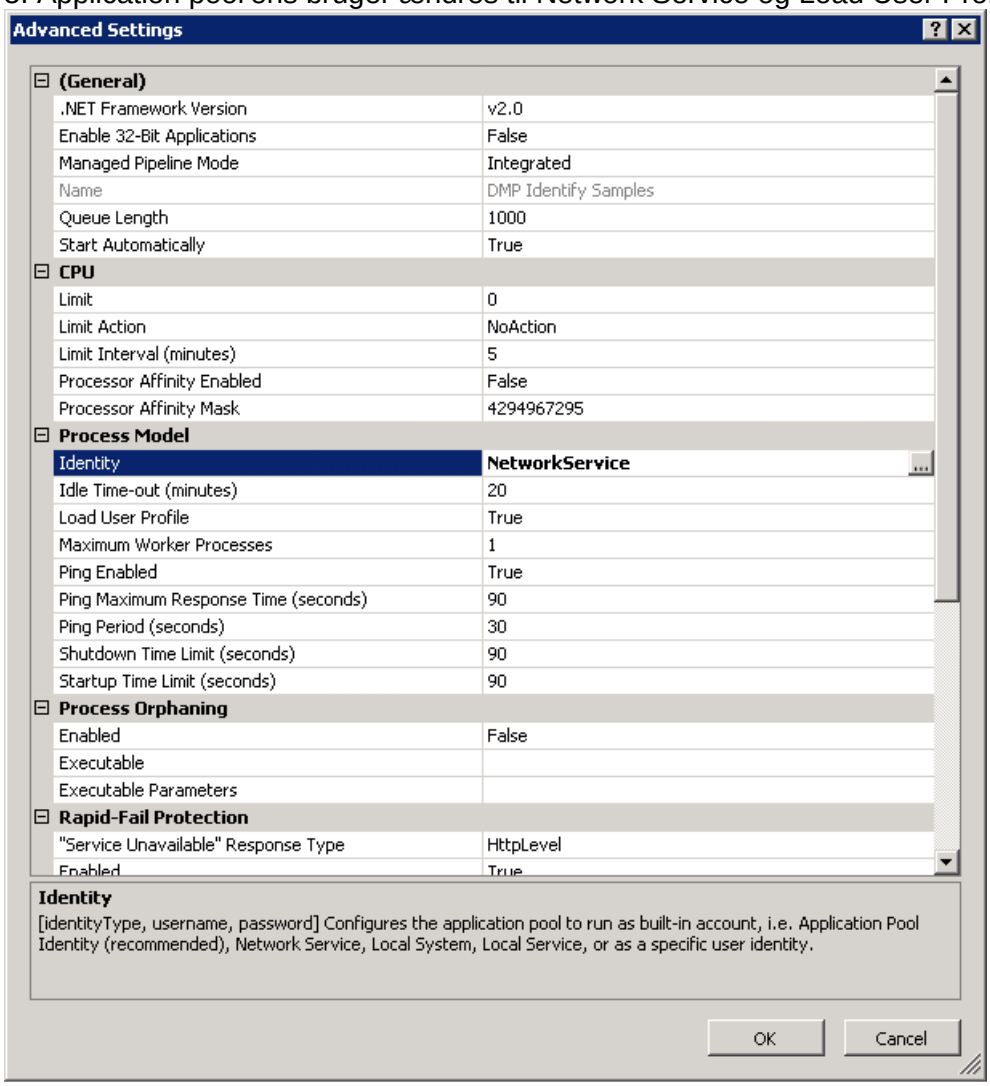
3. Certifikatet: “log-in.miljoeportal.dk signing.cer” og ”log-in.test.miljoeportal.dk signing.cer” installeres i Local Machine -> Trusted People.



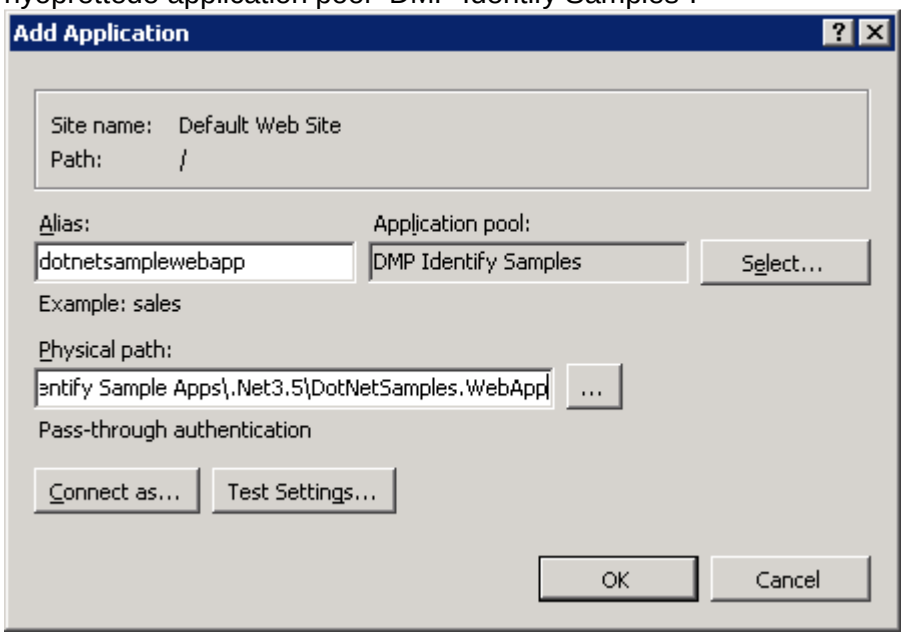
4. Oprettelse af en application pool i IIS 7 ved navn ”DMP Identify Samples”.



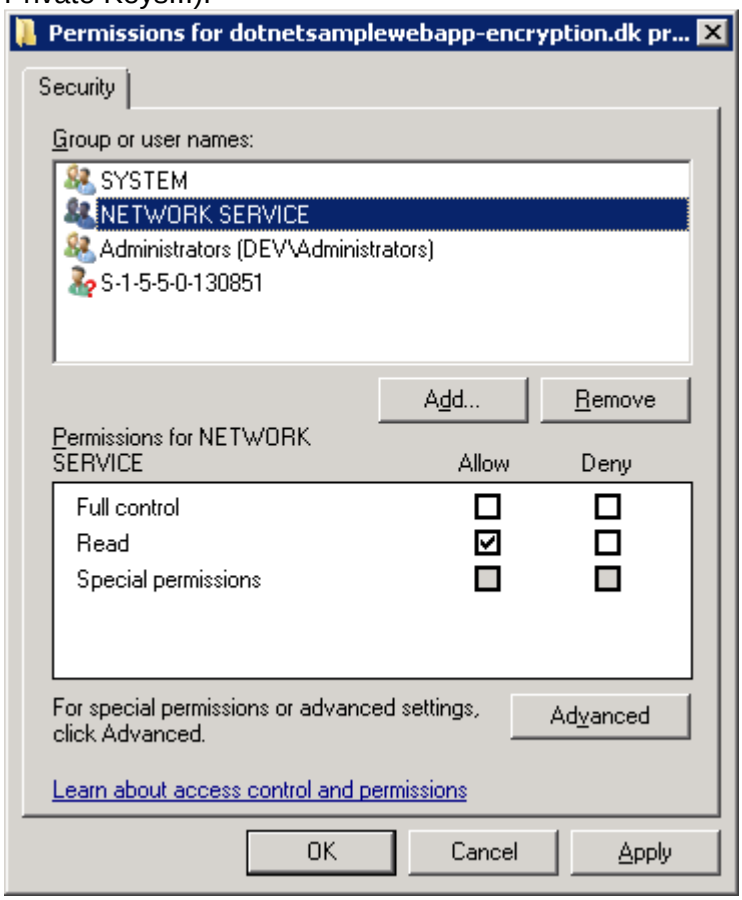
5. Application pool’ens bruger ændres til Network Service og Load User Profile sættes til True.



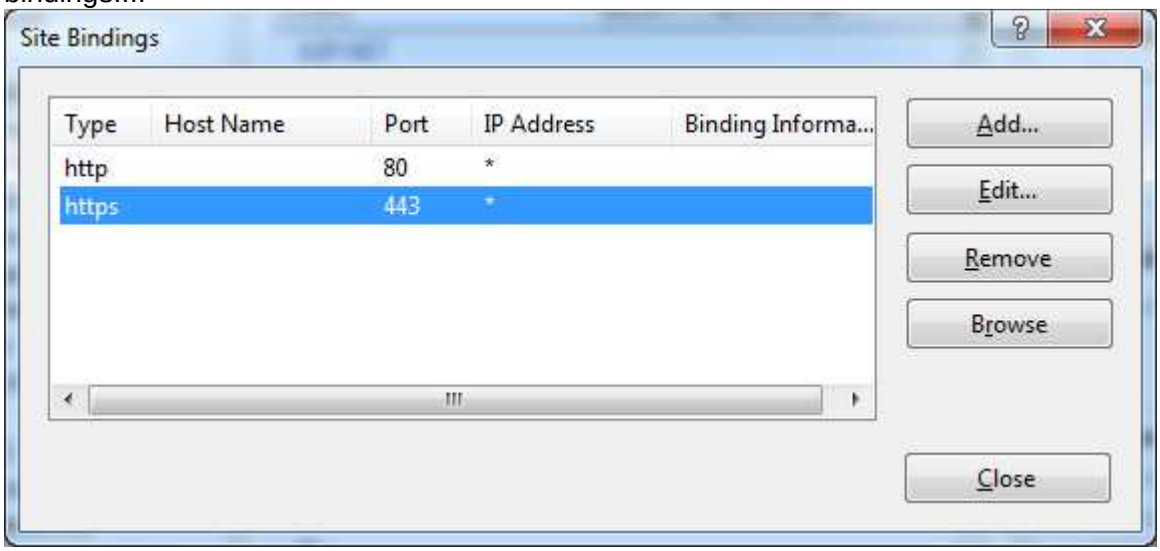
6. Der oprettes en applikation i IIS 7 ved navn ”dotnetsamplewebapp”. Denne skal pege ned på mappen ”DotNetSamples.WebApp” i de medfølgende samples. Applikationen tildeles den nyoprettede application pool ”DMP Identify Samples”.



7. Network Service tildeles Read-rettighed til private key på dotnetsamplewebapp-encryption.dk-certifikatet (højreklik på certifikatet under Local Machine -> Personal og vælg All Tasks -> Manage Private Keys...).



8. Verificer, at der kører SSL (HTTPS) på web sitet ved at højreklikke på web sitet og vælge Edit bindings...:

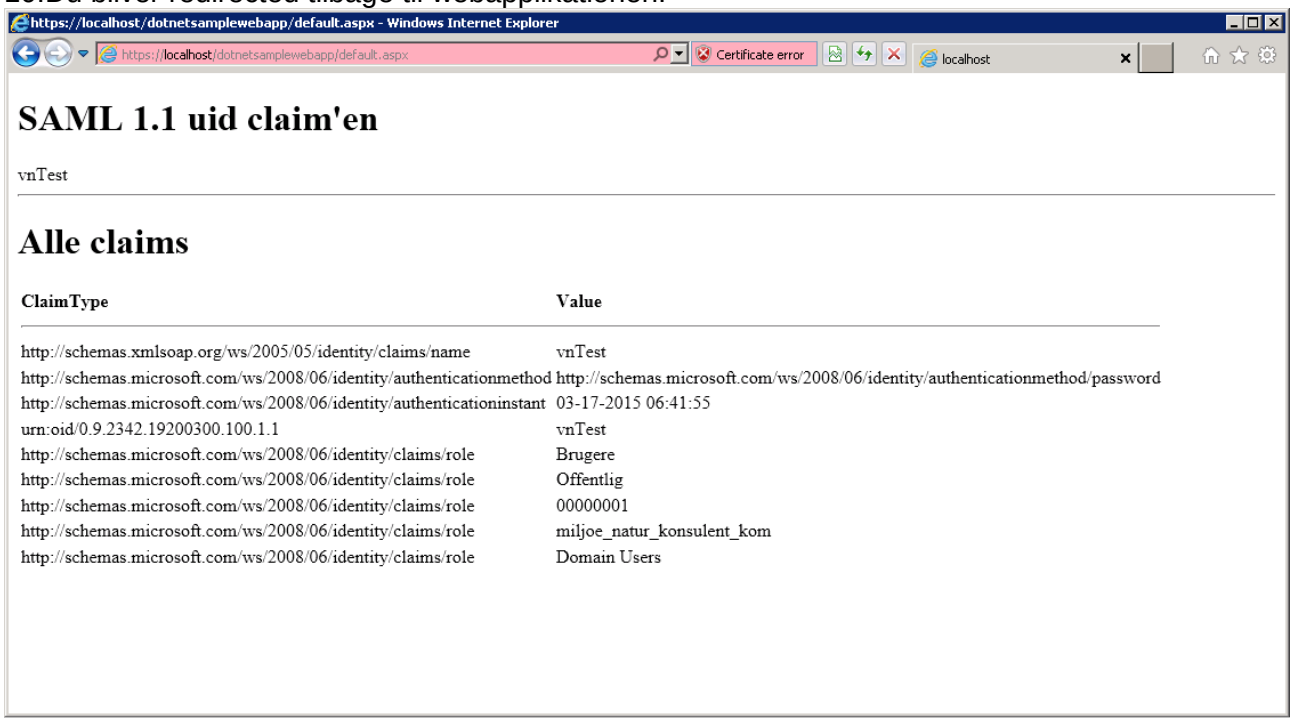


Såfremt der ikke kører SSL på web sitet kan du installere et testcertifikat via følgende guide: http://weblogs.asp.net/scottgu/archive/2007/04/06/tip-trick-enabling-ssl-on-iis7-using-self-signed-certificates.aspx

9. Kompiler løsningen og åbn din browser på https://localhost/dotnetsamplewebapp/. Du vil blive redirected til loginbilledet:

Indtast brugernavn: **vnTest** og password: **HejDMPtest123!** og tryk enter

10.Du bliver redirected tilbage til webapplikationen:



Appendiks A. Beskrivelse af webapplikationens web.config-fil

Dette appendix indeholder en uddybning af indholdet af web applikationens web.config-fil.

Bemærk at sektioner der ikke skal ændres for at bygge en ny webapplikation er udeladt af hensyn til overblikket.

```
<?xml version="1.0"?>
<configuration>

  <microsoft.identityModel>
    <service>
      <audienceUris>
        <add value="https://localhost/dotnetsamplewebapp/"/>
      </audienceUris>
<issuerNameRegistry
type="Microsoft.IdentityModel.Tokens.ConfigurationBasedIssuerNa
meRegistry, Microsoft.IdentityModel, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35">
        <trustedIssuers>
          <!-- Signing certifikat for DMPs Identify-server (TESTMILJØ) -->
          <add thumbprint="c3ab04c6d018029443d90127290b48e0f021eef2" name="log-in.test.miljoeportal.dk
signing"/>
        </trustedIssuers>

      <federatedAuthentication>
        <wsFederation passiveRedirectEnabled="true" issuer="https://log-
in.test.miljoeportal.dk/runtime/WSFederation/WSFederation.idp"
realm="https://localhost/dotnetsamplewebapp/"></wsFederation>
      </federatedAuthentication>
      <serviceCertificate>
        <certificateReference x509FindType="FindBySubjectName" findValue="dotnetsamplewebapp-encryption.dk"
storeLocation="LocalMachine" storeName="My" />
      </serviceCertificate>
    </service>
  <system.serviceModel>

</configuration>
```

Der angives webapplikationens præcise URL i audienceUris og under realm.

Appendiks B. Brug af ActAs til at kalde en service på vegne af en bruger

Der henvises til vejledningen ”DMP - Vejledning til fagsystemejere omkring tilkobling af .NET-baseret web service” for hvordan token’et sendt til en web applikation bruges til kald til bagvedliggende services via ActAs.

Appendix C. Konfiguration af webapplikations web.config-fil ved hjælp af FedUtil

Webapplikationens web.config-fil skal som nævnt tidligere i dette dokument tilpasses på nogle få punkter før .NET-applikationen fungerer efter hensigten.

Det er som nævnt vores anbefaling at man gennemfører opgaven manuelt, da dette er en forholdsvis hurtig og nem operation (se Appendix A).

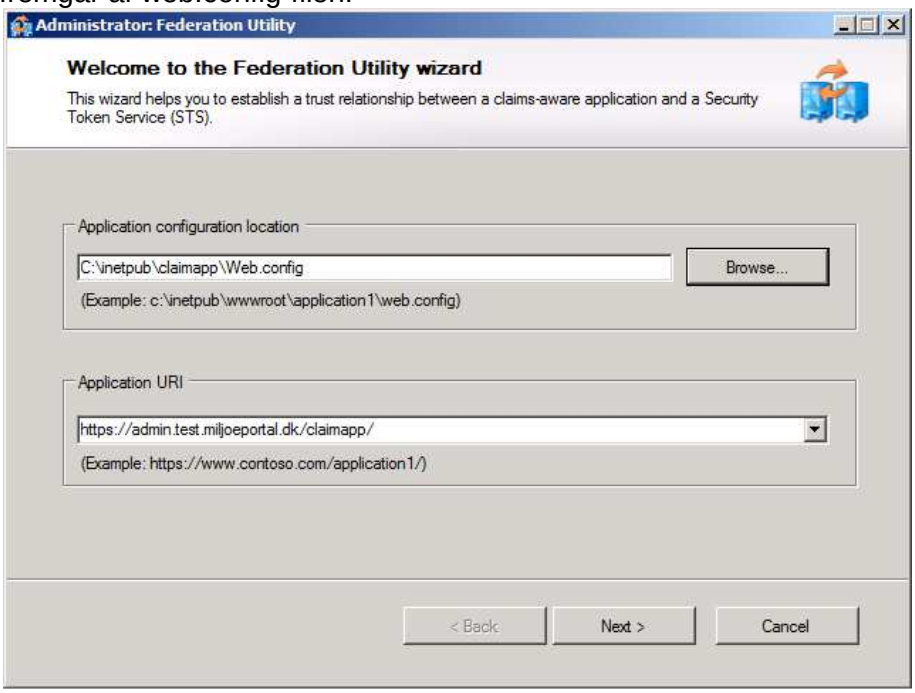
Alternativt kan man anvende værktøjet FedUtil, der er udviklet af Microsoft til formålet. FedUtil genererer imidlertid nogle meget store og komplekse web.config-filer, hvilket nedsætter læseligheden af web.config-filen ganske dramatisk.

For komplethedens skyld gennemgår vi dog brugen af FedUtil i dette appendix.

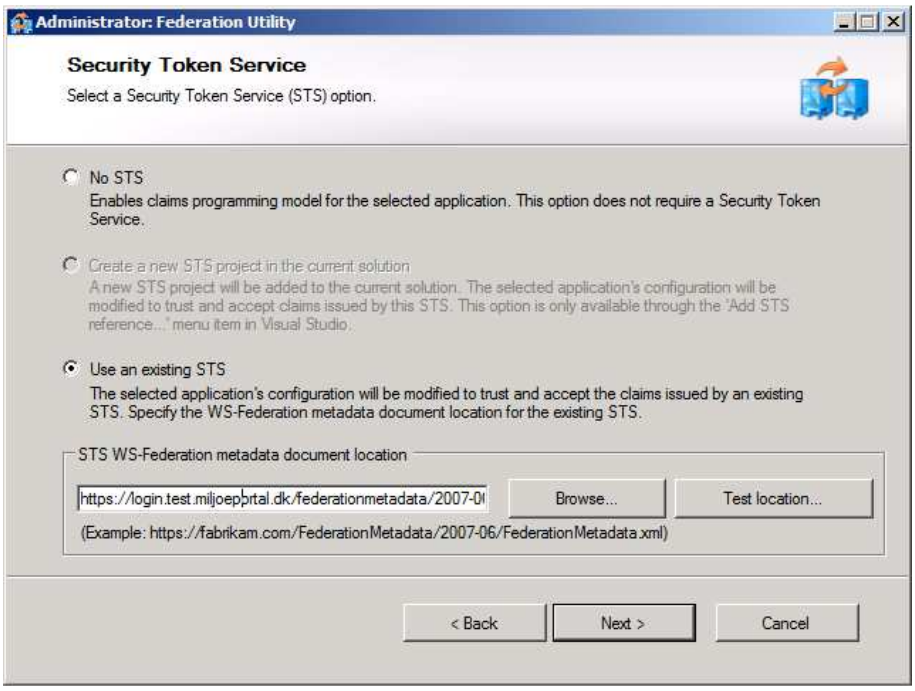
Konfigurering med FedUtil

WIF SDK'et (http://www.microsoft.com/downloads/details.aspx?FamilyID=c148b2df-c7af-46bb-9162-2c9422208504&displaylang=en) indeholder et værktøj ved navn FedUtil, som har til hensigt at gøre det lidt nemmere at gennemføre tilpasningen af web.config-filen end hvis man arbejder direkte i filen.

På første skærmbillede vælges web.config-filen og applikationens navn, hvis ikke det allerede fremgår af web.config-filen:

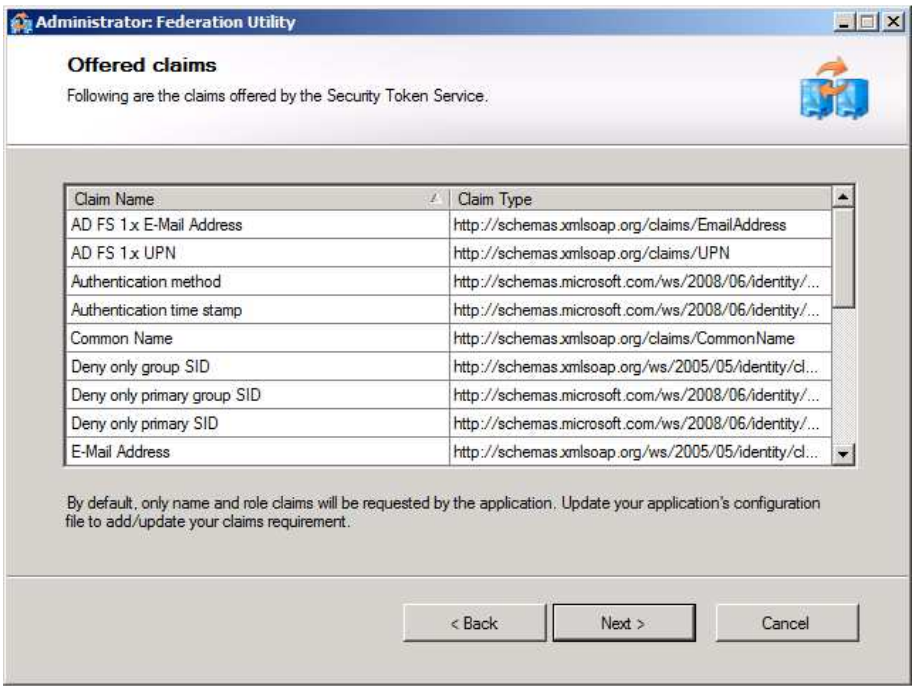


På andet skærmbillede vælges STS'en (dvs. DMP's føderale brugerstyring) og der angives dennes federation metadatafil (enten direkte eller indirekte ved hjælp af dets webadresse):



På det tredje skærmbillede angives krypteringscertifikatet:

På det fjerde skærmbillede fremvises de claims, som STS'en oplyser er til rådighed via dets federation metadata-fil (det er desværre ikke muligt at angive hvilke af disse claims, der skal benyttes af applikationen; dette kan kun ske ved at redigere i web.config-filen):



På det femte skærmbillede opsummeres opsætningerne og der er mulighed for at tilvælge opsætningen af et job, der automatisk checker for evt. ændringer af STS'ens federation metadat-fil en gang i døgnet således at et evt. skift af token signing-certifikat også bliver automatisk ændret i applikationens web.config-fil: